

Ekonomia

Układanie tras pojazdów w łańcuchu dostaw

Modele, metody, zastosowania

Radostaw Jadczyk



Układanie tras pojazdów w łańcuchu dostaw

Modele, metody, zastosowania



WYDAWNICTWO
UNIWERSYTETU
ŁÓDZKIEGO

Ekonomia

Układanie tras pojazdów w łańcuchu dostaw

Modele, metody, zastosowania

Radostaw Jadczyk

Radosław Jadczak – Uniwersytet Łódzki, Wydział Ekonomiczno-Socjologiczny
Katedra Badań Operacyjnych, 90-214 Łódź, ul. Rewolucji 1905 r. nr 39

RECENZENT

Maciej Nowak

REDAKTOR INICJUJĄCY

Monika Borowczyk

REDAKTOR

Aneta Tkaczyk

SKŁAD I ŁAMANIE

AGENT PR

KOREKTA TECHNICZNA

Leonora Gralka

PROJEKT OKŁADKI

Katarzyna Turkowska

Zdjęcie wykorzystane na okładce: © Depositphotos.com/candy18

© Copyright by Radosław Jadczak, Łódź 2019

© Copyright for this edition by Uniwersytet Łódzki, Łódź 2019

Wydane przez Wydawnictwo Uniwersytetu Łódzkiego

Wydanie I. W.08628.18.0.M

Ark. wyd. 14,0; ark. druk. 18,25

ISBN 978-83-8142-140-9

e-ISBN 978-83-8142-141-6

<https://doi.org/10.18778/8142-140-9>

Wydawnictwo Uniwersytetu Łódzkiego

90-131 Łódź, ul. Lindleya 8

www.wydawnictwo.uni.lodz.pl

e-mail: ksiegarnia@uni.lodz.pl

tel. (42) 665 58 63

Spis treści

Wprowadzenie	9
Rozdział 1	
Procesy transportowe w zarządzaniu łańcuchem dostaw	13
1.1. Łańcuch dostaw	13
1.2. Kanały dystrybucji	18
1.3. Procesy transportowe	22
1.4. Organizacja procesów transportowych	27
Rozdział 2	
Klasyczny problem transportowy	33
2.1. Sformułowanie problemu decyzyjnego	34
2.2. Warianty klasycznego problemu transportowego	35
2.2.1. Zadanie transportowe otwarte	36
2.2.2. Zadanie transportowe z kryterium czasu	36
2.2.3. Zadanie transportowe z blokadą tras	37
2.2.4. Wielokryteriove zadanie transportowe	38
2.2.5. Wieloszczeblowe zadanie transportowe	39
2.2.6. Zadanie transportowo-produkcyjne	44
2.2.7. Zagadnienie przydziału	45
2.2.8. Minimalizacja pustych przebiegów	46
2.3. Algorytm transportowy	48
Rozdział 3	
Układanie trasy jednego pojazdu	51
3.1. Sformułowanie problemu decyzyjnego	52
3.2. Warianty problemu komiwojażera	59
3.2.1. Symetryczny i niesymetryczny problem komiwojażera	59
3.2.2. Problem komiwojażera w grafie niepełnym	60
3.2.3. Problem komiwojażera z minimalizacją czasu	62
3.2.4. Problem chińskiego listonosza	63
3.2.5. Problem m -komiwojażerów	67
Rozdział 4	
Układanie tras wielu pojazdów	69
4.1. Sformułowanie problemu decyzyjnego	71
4.2. Warianty problemu wielu komiwojażerów	75

6 Spis treści

4.2.1. Problem wielu komiwojażerów z ograniczeniami zasobowymi	75
4.2.2. Problem wielu komiwojażerów z oknami czasowymi	80
4.2.3. Problem wielu komiwojażerów z dostawą i/lub odbiorem dóbr	83
4.2.4. Problem wielu komiwojażerów z obsługą rozdzieloną	89
4.2.5. Problem wielu komiwojażerów z uzupełnieniami ładunku	92
4.2.6. Problem wielu komiwojażerów z obsługą okresową	97
4.2.7. Stochastyczny problem wielu komiwojażerów	103
4.2.8. Dynamiczny problem wielu komiwojażerów	110
4.2.9. Wielokryterialny problem wielu komiwojażerów	116
4.2.10. Wielobazowy problem wielu komiwojażerów	126
Rozdział 5	
Metody dokładne układania tras pojazdów	141
5.1. Podstawowe założenia metody podziału i ograniczeń	141
5.2. Metody dokładne a zagadnienie przydziału	147
5.3. Metody dokładne a minimalne drzewo rozpinające	154
Rozdział 6	
Heurystyki układania tras pojazdów	163
6.1. Heurystyki konstrukcyjne	164
6.2. Heurystyki dekompozycyjne	177
Rozdział 7	
Metaheurystyki układania tras pojazdów	193
7.1. Jednoagentowe przeszukiwanie lokalne	194
7.1.1. Algorytmy lokalnej optymalizacji	195
7.1.2. Algorytmy symulowanego wyżarzania	207
7.1.3. Algorytmy przeszukiwania tabu	219
7.2. Metaheurystyki wieloagentowe	233
7.2.1. Algorytmy mrówkowe	235
7.2.2. Algorytmy ewolucyjne	248
Zakończenie	269
Spis rysunków i tabel	273
Bibliografia	281

Table of Contents

Introduction	9
Chapter 1	
Transport processes in supply chain management	13
1.1. Supply chain	13
1.2. Distribution channels	18
1.3. Transport processes	22
1.4. Transport processes organization	27
Chapter 2	
Classic transportation problems	33
2.1. Decision problem formulation	34
2.2. Classic transportation problem variants	35
2.2.1. Open transportation problem	36
2.2.2. Transportation problem with time criterion	36
2.2.3. Transportation problem with route blocking	37
2.2.4. Multiobjective transportation problem	38
2.2.5. Multilevel transportation problem	39
2.2.6. Transportation-production problem	44
2.2.7. Assignment problem	45
2.2.8. Minimization of empty routes	46
2.3. Transport algorithm	48
Chapter 3	
Travelling salesman problem	51
3.1. Decision problem formulation	52
3.2. Travelling salesman problem variants	59
3.2.1. Symmetric and asymmetric travelling salesman problem	59
3.2.2. Traveling salesman problem in incomplete graph	60
3.2.3. Traveling salesman problem with time criterion	62
3.2.4. Chinese postman problem	63
3.2.5. Multiple travelling salesman problem	67
Chapter 4	
Vehicle routing problem	69
4.1. Decision problem formulation	71
4.2. Vehicle routing problem variants	75

8 Table of Contents

4.2.1. Capacitated vehicle routing problem	75
4.2.2. Time windows vehicle routing problem	80
4.2.3. Pich-up and delivery vehicle routing problem	83
4.2.4. Split delivery vehicle routing problem	89
4.2.5. Vehicle routing problem with satellite facilities	92
4.2.6. Period vehicle routing problem	97
4.2.7. Stochastic vehicle routing problem	103
4.2.8. Dynamic vehicle routing problem	110
4.2.9. Multiobjective vehicle routing problem	116
4.2.10. Multi-depot vehicle routing problem	126
 Chapter 5	
Exact methods for vehicle routing problems	141
5.1. Branch and bounds basics	141
5.2. Exact algorithms and assignment problem	147
5.3. Exact algorithms and minimum spanning tree	154
 Chapter 6	
Heuristics for vehicle routing problems	163
6.1. Construction heuristics	164
6.2. Decomposition heuristics	177
 Chapter 7	
Metaheuristics for vehicle routing problems	193
7.1. Single-agent local optimization	194
7.1.1. Local search algorithms	195
7.1.2. Simulated annealing algorithms	207
7.1.3. Tabu search algorithms	219
7.2. Multi-agent metaheuristics	233
7.2.1. Ant algorithms	235
7.2.2. Evolutionary algorithms	248
 Conclusion	269
List drawings and tables	273
Bibliography	281

Wprowadzenie

Dzisiejsze rozumienie procesów logistycznych jest niewspółmiernie szersze niż blisko siedemdziesiąt lat temu, kiedy logistyka zaczynała być postrzegana jako niezwykle użyteczna koncepcja biznesowa. Ciągły rozwój procesów gospodarczych, postępująca globalizacja rynku, a także rozkwit i doskonalenie istniejących technik zarządzania sprawiły, że konieczne stało się wypracowanie nowego spojrzenia na logistykę, wykraczającego daleko poza granice pojedynczego przedsiębiorstwa.

Obecnie z zagadnieniem logistyki nierozzerwalnie łączy się pojęcie łańcucha dostaw, który według wielu badaczy jest częścią łańcucha logistycznego, a nawet sieci logistycznych. Łańcuch dostaw – w dużym uproszeniu – to zbiór wzajemnie ze sobą powiązanych podmiotów, których nadrzędnym celem jest dostarczenie finalnemu nabywcy pożądanego przez niego produktu z jak największą wartością dodaną. Procesy logistyczne natomiast stanowią zbiór wszelkich działań służących osiągnięciu tego celu przy jak najwyższym poziomie obsługi konsumenta finalnego.

Globalizacja procesów gospodarczych oraz nieustannie zmieniające się potrzeby rynku stawiają przed decydentami nowe wyzwania nie tylko w kwestii poszukiwania nowych rynków zbytu, ale także (jeżeli nie przede wszystkim) poprawy konkurencyjności na rynkach istniejących. Wyzwania te dotyczą ciągłego doskonalenia działań zarówno w zakresie modyfikacji istniejących i konstruowania nowych łańcuchów dostaw, jak i obsługujących je procesów logistycznych. Ponadto bardzo szybki i wciąż postępujący rozwój systemów informacyjnych (w tym informatycznych i telekomunikacyjnych) stawia decydentów przed koniecznością podejmowania decyzji właściwie w czasie rzeczywistym. Powoduje to, że coraz wyraźniejszych kształtów nabiera i coraz bardziej doceniana staje się logistyka o charakterze operacyjnym, polegająca na nieustannym identyfikowaniu, formułowaniu i rozwiązywaniu problemów decyzyjnych w celu poszukiwania optymalnych decyzji.

Niewątpliwie kolejnym sporym wyzwaniem jest identyfikacja i rozwiązywanie problemów w zakresie dystrybucji produktów w łańcuchu dostaw. Chodzi tu między innymi o problemy transportowe, a w szczególności te, które dotyczą układania tras jednego lub wielu pojazdów. W literaturze przedmiotu znane są one od dawna pod nazwą zagadnienia jednego lub wielu komiwojazerów. Znacznie dłuższą historię ma problem jednego komiwojazera, którego początki sięgają połowy

XIX stulecia. Problem wielu komiwojazerów z kolei po raz pierwszy został sformułowany przez G.B. Dantziga i J.H. Ramsera w 1959 roku jako problem wysyłania ciężarówek (*Truck Dispatching Problem*).

Cechą charakterystyczną zagadnień układania tras dla pojazdów jest łatwość ich sformułowania, w przeciwieństwie do znalezienia optymalnego rozwiązania. Pojazd lub pojazdy, które wyruszają z miejsca startowego (bazy), mają za zadanie odwiedzić każdy z zadanej liczby punktów zlokalizowanych na określonym obszarze geograficznym, spełniając przy tym wszystkie nałożone na problem ograniczenia (np. dotyczące czasu pracy pojazdu, ładowności, godzin pracy odwiedzanych punktów itp.) w taki sposób, aby szeroko rozumiany łączny koszt tras wszystkich wykorzystanych pojazdów był jak najniższy.

Jest to wysoce skomplikowane zagadnienie, a poszukiwanie optymalnego układu tras oraz problem jednego i wielu komiwojazerów wciąż przyciągają uwagę wielu badaczy oraz praktyków, czego najlepszym dowodem jest ogromna i wciąż powiększająca się liczba publikacji na ten temat. Można w nich znaleźć zarówno próby identyfikacji i opisu modeli nowych odmian podstawowego problemu decyzyjnego, jak i prezentację coraz to nowszych metod ich rozwiązywania. Proponowane metody poszukiwania optymalnych tras charakteryzują się z jednej strony prostotą, a z drugiej strony wysokim stopniem złożoności podejmowanych kroków. Jedne gwarantują znalezienie rozwiązań optymalnych, podczas gdy inne tego nie zapewniają, dając rozwiązania bardzo dobre, ale nie najlepsze z możliwych. Niestety zazwyczaj decyzje optymalne okupione są dużym wysiłkiem obliczeniowym.

Nie sposób przedstawić w tym miejscu wszystkich istniejących metod odnoszących się do problemów układania tras pojazdów. Dlatego celem niniejszej pracy jest próba zebrania oraz usystematyzowania wiedzy i doświadczeń z ostatnich blisko pięćdziesięciu lat badań nad zagadnieniami logistyki operacyjnej w zakresie transportu. Motywacją do jej napisania było wypełnienie luki w piśmiennictwie w tej dziedzinie, gdyż i w fachowej literaturze obcojęzycznej, i w literaturze polskiej brakuje pozycji podejmujących tak ważne z punktu widzenia logistyki operacyjnej zagadnienia.

Książka podzielona została na siedem rozdziałów, które układają się w trzy nieformalne części. Rozdział pierwszy to część wprowadzająca. Przedstawiono w nim podstawowe zagadnienia związane z łańcuchem dostaw oraz procesami transportowymi. Zawiera definicje łańcuchów dostaw, kanałów dystrybucji, procesów transportowych i przewozowych oraz ich najważniejsze klasyfikacje. Omówiono tam także rolę procesów transportowych w obsłudze łańcucha dostaw ze szczególnym uwzględnieniem najważniejszych form i modeli ich organizacji.

Część drugą stanowią kolejne trzy rozdziały, których celem jest prezentacja podstawowych modeli i ich odmian dotyczących zagadnień układania tras pojazdów. Rozdział drugi poświęcony został najbardziej znanemu wśród logistyków klasycznemu problemowi transportowemu. Oprócz sformułowanej tam podstawowej postaci problemu przedstawiono również najważniejsze jego

odmiany i modyfikacje, a także algorytm transportowy służący znalezieniu rozwiązania optymalnego. Klasyczny problem transportowy jest nie tylko ważny z punktu widzenia problematyki logistyki operacyjnej, ale może też stanowić niezwykle cenne narzędzie w poszukiwaniu rozwiązań zadań, o których będzie mowa w dalszej części książki.

W rozdziałach trzecim i czwartym zdefiniowano zagadnienia układania tras – odpowiednio – jednego i wielu pojazdów. Rozdział trzeci omawia problem komiwojażera. Poza tym, że znajduje się on w centrum zainteresowań logistyków, stanowi także problem o charakterze kombinatorycznym, chętnie podejmowany jako pole zastosowań do wielu nowych narzędzi optymalizacji.

Z kolei rozdział czwarty skupia się stricte na problematyce układania tras pojazdów i sformułowano w nim podstawową postać problemu wielu komiwojażerów. Ponadto przedstawiono w tym miejscu najważniejsze, zdaniem autora, warianty i modyfikacje problemu podstawowego, które uwzględniają występujące w praktyce gospodarczej dodatkowe założenia i warunki związane z optymalizacją tras pojazdów (tj. ograniczenia czasowe, ograniczenia związane z zasobami transportowymi czy założenia związane z samą strukturą rozwiązywanego zadania).

Trzecia (nieformalna) część pracy poświęcona została prezentacji metod służących do poszukiwania optymalnych tras pojazdów. Jako że w ciągu ostatnich kilkudziesięciu lat pojawiło się ich bardzo wiele, zdecydowano się wybrać spośród nich te najważniejsze i najbardziej charakterystyczne. Ich opis znajduje się w trzech osobnych rozdziałach.

Rozdziały piąty i szósty stanowią zbiór klasycznych podejść do rozwiązania problemów układania tras jednego i wielu pojazdów. Rozdział piąty zawiera prezentację metod dokładnych, czyli takich, które gwarantują znalezienie rozwiązania optymalnego. Jednakże metody te charakteryzują się znacznym stopniem złożoności i trudności, a ponadto bardzo często korzystają z przekształconej wyjściowej postaci problemu podstawowego (np. zagadnienia przydziału czy minimalnego drzewa rozpinającego), dlatego w rozdziale szóstym przedstawiono najchętniej stosowane przez praktyków strategie poszukiwania optymalnych tras, jakimi są metody heurystyczne. W przeciwieństwie do metod dokładnych nie zapewniają one, że znaleziony przez nie układ tras jest najlepszy, lecz cechują się o wiele niższym stopniem złożoności, co powoduje, że są łatwe w użyciu.

Ostatni rozdział, omawiający metody optymalizacji układania tras pojazdów, jest najobszerniejszy, ponieważ zawiera nowatorskie, a zarazem nieklasyczne podejścia do układania tras pojazdów. Metody tam przedstawione czerpią motywację nie tylko z cech i konstrukcji samego problemu układania tras (metody przeszukiwania lokalnego), ale także z procesów zachodzących w naturze (algorytmy metaheurystyczne). W rozdziale tym zaprezentowano zastosowanie do problemów jednego i wielu komiwojażerów: algorytmów symulowanego wyżarzania, algorytmów przeszukiwania tabu, algorytmów mrówkowych oraz algorytmów ewolucyjnych. Przybliżenie ich podstawowych założeń i procesów działania może

też stanowić inspirację do budowy konkretnych metod poszukiwania najlepszych rozwiązań innych problemów optymalizacji.

Niniejsza pozycja może być adresowana do szerokiego grona czytelników. Z jednej strony mogą to być praktycy (analitycy i decydenci) poszukujący najlepszych, a zarazem korzystnych ekonomicznie rozwiązań służących optymalizacji problemów transportowych (i nie tylko) w ramach logistyki operacyjnej w swoich przedsiębiorstwach. Z drugiej strony publikacja ta może być także wykorzystywana jako zwarty materiał źródłowy stanowiący punkt wyjścia do dalszych badań zarówno nad identyfikacją nowych odmian problemów układania tras pojazdów, jak i poszukiwaniem nowych, lepszych sposobów ich rozwiązywania. Ponadto w intencji autora powinna ona stanowić interesujące uzupełnienie literatury logistycznej, a w szczególności w zakresie logistyki operacyjnej.

Rozdział 1

Procesy transportowe w zarządzaniu łańcuchem dostaw

W dobie gospodarki globalnej logistyka jest terminem rozumianym o wiele szerzej niż w drugiej połowie ubiegłego stulecia. Pojęcie to ewaluowało od prostego fizycznego ruchu produktu pomiędzy wytwórcą a konsumentem do dziedziny wiedzy, której jednym z celów jest optymalne kształtowanie łańcuchów dostaw.

Dzisiejsze znaczenie logistyki rozumiane jest jako proces polegający na planowaniu, realizowaniu oraz kontrolowaniu przepływu dóbr i informacji pomiędzy jego interesariuszami, w szczególności pomiędzy dostawcami i odbiorcami. Procesy fizycznego przepływu dóbr oraz przepływu informacji nie odnoszą się tylko do wąsko pojmowanej bezpośredniej relacji dostawca–odbiorca (np. producent–konsument), lecz obejmują wszystkie relacje, począwszy od naturalnych źródeł wytworzenia, na punkcie konsumpcji skończywszy. Podmiotem szeroko rozumianych działań logistycznych jest łańcuch dostaw, który może być definiowany w ramach pojedynczego przedsiębiorstwa, ale i pewnego zbioru organizacji, znacznie się od siebie różniących pod względem np. swojej podstawowej działalności.

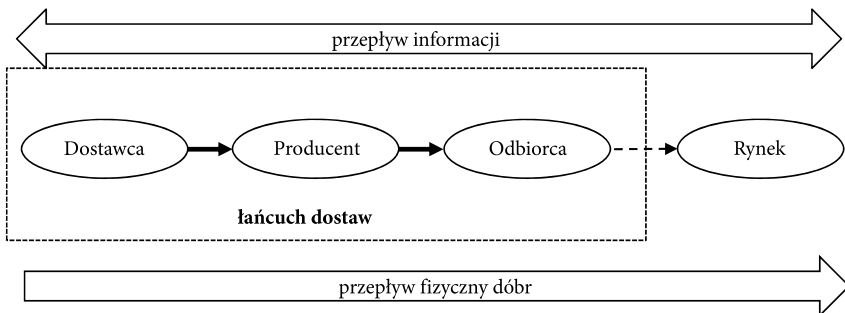
1.1. Łańcuch dostaw

W literaturze związanej z logistyką trudno znaleźć jedną uniwersalną definicję pojęcia łańcucha dostaw. Przedstawia się je na rozmaite sposoby: jako pewną strukturę podmiotów gospodarczych, jako proces gospodarczy lub filozofię zarządzania. Według jednej z definicji łańcuch dostaw to zbiór różnych organizacji, które zaangażowane są w różnego rodzaju procesy i działania mające na celu dostarczenie finalnemu konsumentowi produktów lub usługi (Christopher, 2000, s. 14). To zaangażowanie polega na różnego rodzaju powiązaniach z dostawcami i odbiorcami. Z kolei w Słowniku *terminologii logistycznej* (Fertsch, 2006) łańcuch

logistyczny rozumiany jest dwojako. Po pierwsze, jest to proces, który stanowi sekwencja zdarzeń dotyczących przemieszczenia dóbr, aby zwiększyć ich wartość. Po drugie, łańcuch dostaw to pewna struktura (organizacja), którą stanowi grupa przedsiębiorstw mająca na celu zaspokojenie popytu na produkty w całym łańcuchu fizycznego przepływu dóbr. Grupa przedsiębiorstw osiąga ten cel przez podejmowanie wspólnych działań, które w szczególności dotyczyć mogą: rozwoju, produkcji, sprzedaży, zaopatrzenia, serwisu, dystrybucji, zarządzania zasobami oraz czynności wspierających.

Łańcucha dostaw nie można jednak postrzegać tylko w kategoriach określonej pionowej struktury, w której np. dystrybutorzy produktu należą do producenta (są jego własnością). Coraz częściej przedsiębiorstwa doceniają możliwość korzystania z rozmaitych usług świadczonych przez osobne podmioty gospodarcze (outsourcing), dzięki czemu mogą się skupić na swojej podstawowej działalności. Zjawisko to nie odnosi się zresztą tylko do usług. Formą outsourcingu może być także nabywanie różnego rodzaju dóbr (np. surowców lub materiałów) niezbędnych do funkcjonowania przedsiębiorstwa, które dotąd dane przedsiębiorstwo musiało wytwarzać we własnym zakresie. Taki sposób inicjowania i rozwijania powiązań pomiędzy podmiotami gospodarczymi jest charakterystyczny właśnie dla łańcucha dostaw.

Łańcuch dostaw może mieć charakter zarówno wewnętrzny, jak i zewnętrzny (rys. 1.1 i 1.2) (Fechner, 2007).

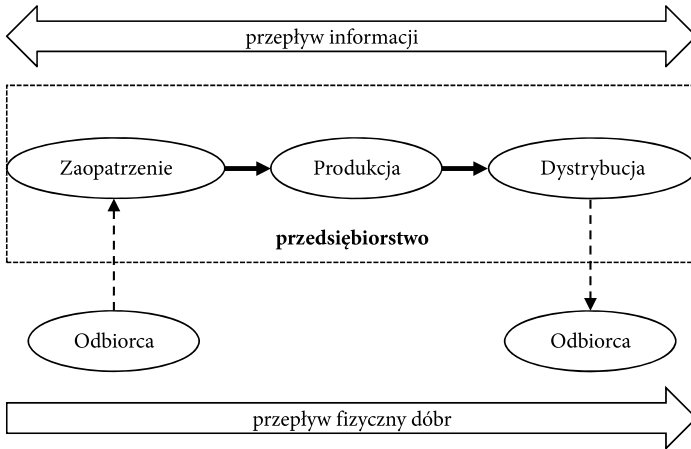


Rys. 1.1. Zewnętrzny łańcuch dostaw

Źródło: opracowanie własne na podstawie (Fechner, 2007).

O łańcuchu zewnętrznym można mówić wtedy, gdy tworzą go podmioty od siebie niezależne (w sensie prawnym i gospodarczym), ale nastawione na realizację wspólnego celu. Z kolei w wypadku łańcucha wewnętrznego jego strukturę tworzą jednostki organizacyjne (np. działy, przedstawicielstwa, oddziały itp.) należące do danego przedsiębiorstwa. Ich zadaniem jest osiągnięcie celu, jaki stoi przed danym przedsiębiorstwem. Należy tutaj zauważyć, że zarówno kształt, jak i stopień złożoności konkretnego łańcucha dostaw (zewnętrznego i wewnętrznego) nie są zawsze jednakowe. Wręcz przeciwnie – są one bardzo zróżnicowane i uzależnione od wielu czynników, do których należą między innymi:

- cel, jaki jest postawiony przed łańcuchem dostaw;
- liczba i charakter podmiotów tworzących łańcuch dostaw;
- rodzaj produktu, dla którego funkcjonuje łańcuch dostaw;
- zasięg działania łańcucha dostaw;
- wzajemne powiązania pomiędzy podmiotami tworzącymi łańcuch dostaw.



Rys. 1.2. Wewnętrzny łańcuch dostaw

Źródło: opracowanie własne na podstawie (Fechner, 2007).

W związku z tym potencjalne konfiguracje łańcucha dostaw (w szczególności zewnętrzne) mogą mieć charakter:

- bardzo prosty (bezpośrednie relacje odpowiednio producenta z dostawcą i producenta z odbiorcą); konfiguracje te zazwyczaj dotyczą bardzo prostych produktów i cechują się niewielkim zasięgiem działania;
- złożony (rozbudowana struktura zarówno po stronie dostawców, jak i po stronie odbiorców, np. dostawcy niższego rzędu i dostawcy wyższego rzędu); konfiguracje te dotyczą bardziej złożonych produktów oraz cechują się dużym zasięgiem działania np. przez rozbudowaną formę dystrybucji;
- kompleksowy (struktura rozbudowana o dodatkowe podmioty świadczące np. usługi informatyczne, logistyczne, finansowe, telekomunikacyjne); w konfiguracjach tych widoczny jest największy stopień integracji podstawowych podmiotów (dostawców i odbiorców), działających na podstawie wypracowanych takich samych standardów dotyczących np. przepływu informacji lub zarządzania konkretnymi operacjami logistycznymi.

Istotę każdego łańcucha dostaw stanowią jego poszczególne ogniwa oraz kształt wzajemnych relacji pomiędzy nimi. Ogniwami łańcuchów dostaw są osobne i niezależne od siebie przedsiębiorstwa, jak również podmioty gospodarcze, które mogą należeć do jednego właściciela. Wyróżniamy ogniwa: początkowe (zaopatrujące

w materiały i surowce niezbędne do produkcji), pośrednie (wytwarzające półprodukty i wyroby gotowe) oraz końcowe (dostarczające produkty do klientów końcowych). Pomędzy nimi mogą występować odrębne podmioty zapewniające sprawny przepływ fizyczny produktów, do których można zaliczyć operatorów logistycznych (np. centra dystrybucyjne i logistyczne).

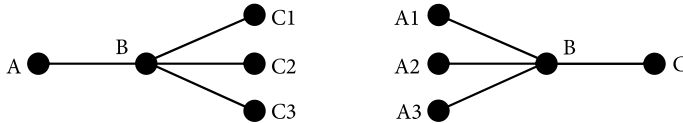
Z kolei wzajemne powiązania pomiędzy poszczególnymi ogniwami każdego łańcucha dostaw pozwalają na wyodrębnienie różnych ich konfiguracji, których kształt w sposób bezpośredni wpływa na sposób (metody) realizacji przepływów fizycznych produktów oraz związanych z nimi przepływów informacyjnych. W związku z tym wyróżnić można następujące przykładowe konfiguracje spotykane w łańcuchach dostaw:

- bezpośrednio (rys. 1.3a);
- pośrednio dwuetapowe (rys. 1.3b);
- pośrednio wieloetapowe (rys. 1.3c);
- mieszane (rys. 1.3d).



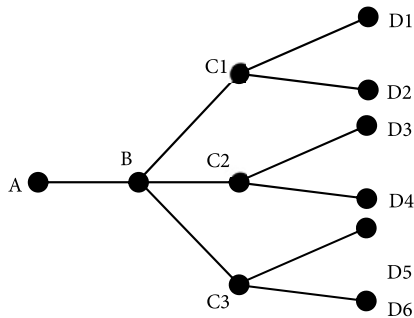
Rys. 1.3a. Przykłady konfiguracji bezpośredniej w łańcuchu dostaw

Źródło: opracowanie własne na podstawie (Szatek, 1992).



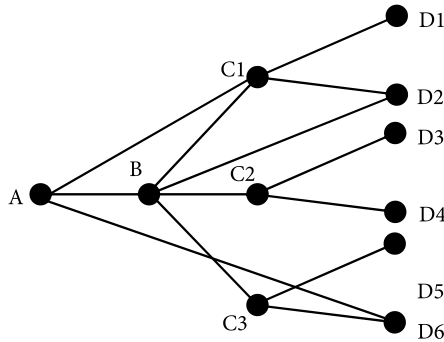
Rys. 1.3b. Przykłady konfiguracji dwuetapowej pośredniej w łańcuchu dostaw

Źródło: opracowanie własne na podstawie (Szatek, 1992).



Rys. 1.3c. Przykład konfiguracji wieloetapowej pośredniej w łańcuchu dostaw

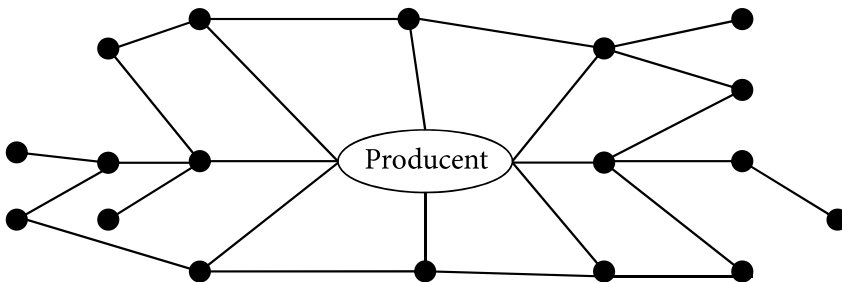
Źródło: opracowanie własne na podstawie (Szatek, 1992).



Rys. 1.3d. Przykłady konfiguracji mieszanej w łańcuchu dostaw
Źródło: opracowanie własne na podstawie (Szalek, 1992).

Do czynników kształtujących konfigurację danego łańcucha dostaw zaliczyć można: rodzaj produktu (w szczególności jego stopień złożoności), lokalizację przestrzenną dostawców oraz odbiorców, lokalizację infrastruktury operatorów logistycznych (magazyny, terminale przeładunkowe), przyjętą strategię współpracy z jednym lub wieloma dostawcami czy jednym lub wieloma operatorami logistycznymi.

Pojęciem, które jest naturalnym rozwinięciem łańcucha dostaw, jest „sieć dostaw” (Łupicka, 2006, s. 11). Może być ona traktowana także jako złożenie wielu łańcuchów dostaw. Termin ten wynika bezpośrednio z samej postaci, jak również z możliwości dowolnego rozbudowania przedstawionej powyżej konfiguracji mieszanej łańcucha dostaw (rys. 1.4), co powoduje, że przyjmuje on postać sieci. Jest to odzwierciedleniem rzeczywistości gospodarczej, w której nierzadko mamy do czynienia z dostawcami dostawców lub klientami klientów tworzącymi wspólnie bardzo rozbudowany system dostaw. Ponadto tak jak w łańcuchach dostaw, tak i w sieciach dostaw ogniwami mogą być operatorzy logistyczni, świadczący usługi wszystkim pozostałym podmiotom (dostawcom i odbiorcom).



Rys. 1.4. Sieć dostaw
Źródło: opracowanie własne na podstawie (Łupicka, 2006, s. 11).

Z pojęciami łańcucha dostaw lub sieci dostaw w sposób nierozzerwalny wiąże się termin zarządzania łańcuchem dostaw. Można go zdefiniować jako pewnego rodzaju koncepcję zarządzania relacjami z dostawcami i odbiorcami, a także z klientami w celu dostarczenia produktu we właściwym miejscu i czasie, przy jednoczesnej minimalizacji kosztów związanych z funkcjonowaniem łańcucha dostaw.

Wreszcie, patrząc całościowo na łańcuch logistyczny, począwszy od źródeł wytwarzania podstawowych surowców niezbędnych do powstania wyrobów gotowych, a skończywszy na konsumentach finalnych, wyróżnić można dwie części łańcucha dostaw z punktu widzenia każdego ogniwa (przedsiębiorstwa) pośredniego: łańcuch zaopatrzenia oraz łańcuch dystrybucji. Dotyczy to także łańcucha wewnętrznego w ramach konkretnego przedsiębiorstwa, gdzie wskazuje się podsystem logistyki zaopatrzenia i podsystem logistyki dystrybucji. W obrębie obu tych części bardzo istotną rolę odgrywają kanały zaopatrzenia i dystrybucji kształtujące bezpośrednio sposób przepływu fizycznego produktów.

1.2. Kanały dystrybucji

Choć można dokonać podziału łańcucha dostaw na część dotyczącą zaopatrzenia i dystrybucji, to z perspektywy każdego przedsiębiorstwa stanowiącego jego ogniwo wszystkie ogniwa (podmioty) znajdujące się po nim tworzą łańcuch dystrybucji. Szczególnym pojęciem związanym z logistyką dystrybucji jest „kanał dystrybucji”. Najprościej rzecz ujmując, jest to jeden podmiot lub większa liczba podmiotów uczestniczących w procesie przepływu fizycznego dóbr od producenta do użytkownika końcowego (Coyle i in., 2002). Z punktu widzenia szeroko rozumianego łańcucha dostaw kanał dystrybucyjny może mieć swój początek w ogniwie, które stanowi przedsiębiorstwo produkcyjne, a koniec u konsumenta finalnego. Pośrednimi punktami kanału mogą być np. centra dystrybucji, hurtownie, punkty sprzedaży detalicznej, stanowiące własność producenta lub będące niezależnymi od niego podmiotami gospodarczymi. Z kolei z perspektywy dostawcy, który dostarcza producentowi niezbędne materiały i surowce do produkcji, jego kanał dystrybucji kończyć się może właśnie na nim, gdzie konsumentem finalnym jest dla niego producent wyrobu gotowego. Poszczególne kanały dystrybucji składać się mogą na łańcuch dostaw lub szerzej – sieć dostaw.

W literaturze przedmiotu rozróżnia się kilka rodzajów kanałów dystrybucyjnych, do których zaliczyć można:

- kanały dystrybucji pośrednie i bezpośrednie;
- kanały dystrybucji krótkie i długie;
- kanały dystrybucji wąskie i szerokie.

Podział ten warunkowany jest przede wszystkim przyjętą strukturą dystrybucji w łańcuchu dostaw, a w szczególności liczbą pośredniczących ogniw pomiędzy dostawcą a finalnym nabywcą produktu. Kanały bezpośrednie to takie, które dotyczą bezpośrednich powiązań producenta z finalnymi nabywcami, w przeciwieństwie do kanałów pośrednich, w których zaangażowane są podmioty pośrednie łączące producenta z finalnym odbiorcą.

Mianem krótkich kanałów dystrybucji zwykle określa się te, które składają się z co najwyżej jednego podmiotu pośredniczącego, w przeciwieństwie do kanałów długich zawierających większą liczbę pośredników. Z kolei podział kanałów dystrybucji na wąskie i szerokie uwarunkowany jest stopniem złożoności ogniw wszerek łańcucha dostaw. Im większa liczba pośredników tego samego typu na poszczególnych etapach dystrybucji, za tym szerszy można uznać kanał dystrybucji.

Przedstawione powyżej rodzaje kanałów dystrybucji mają swoje wady i zalety. Mniej rozbudowane kanały dystrybucji powodują, że różnice w cenach tego samego produktu między producentem a punktem sprzedaży detalicznej są znacznie mniejsze niż w wypadku kanałów bardziej złożonych. Z kolei kanały bardziej rozbudowane oferują większe możliwości w dotarciu do dużej liczby potencjalnych odbiorców. Ponadto szybciej i elastyczniej reagują one na zmieniający się popyt konsumentów finalnych ze względu na możliwość zdecentralizowania poziomu zapasów w poszczególnych ogniwach łańcucha dostaw. Ale też budowa złożonych kanałów dystrybucji wymaga także właściwej organizacji i ciągłej kontroli procesów związanych z przemieszczaniem fizycznym produktów w dół kanału (od producenta przez pośredników do klienta finalnego) oraz przepływem informacji w górę kanału (od uczestników rynku przez pośredników do producenta).

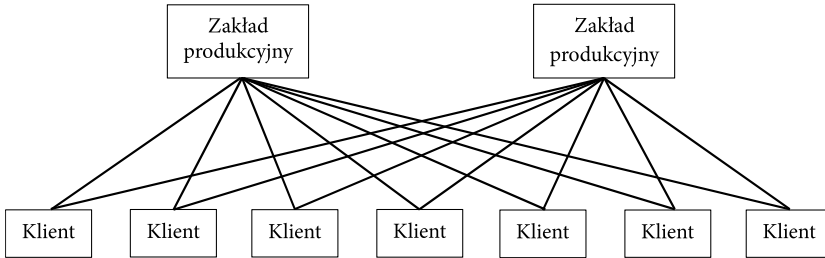
Konstrukcja konkretnego kanału dystrybucji wynika wprost z przyjętej strategii dystrybucji fizycznej produktów. Wśród znanych modeli systemów fizycznej dystrybucji produktów wyróżnić można trzy podstawowe (Rutkowski, 2005, s. 55–62):

- 1) model dostaw bezpośrednich;
- 2) scentralizowany model dystrybucji;
- 3) zdecentralizowany model dystrybucji.

Model dostaw bezpośrednich odnosi się do sytuacji, w której dostawy są realizowane bezpośrednio od przedsiębiorstwa produkcyjnego (rys. 1.5). Dotyczyć to może produkcji na zamówienie, w szczególności dóbr o charakterze luksusowym lub produktów nienależących do luksusowych, ale zamawianych przez klientów finalnych w dużych ilościach. Model ten może być stosowany także wobec produktów wysoce wyspecjalizowanych lub łatwo się psujących. Jego zalety to:

- możliwość redukcji poziomu zapasów w łańcuchu dostaw, a tym samym niższe koszty z nimi związane;
- ograniczenie liczby operacji transakcyjnych i ryzyka związanego z uszkodzeniem produktu;

- skrócenie czasu pomiędzy produkcją a momentem udostępnienia produktu do sprzedaży finalnemu nabywcy;
- możliwość optymalizacji poziomu produkcji;
- uzyskiwanie informacji o popycie u źródeł jego powstawania.



Rys. 1.5. Model dostaw bezpośrednich

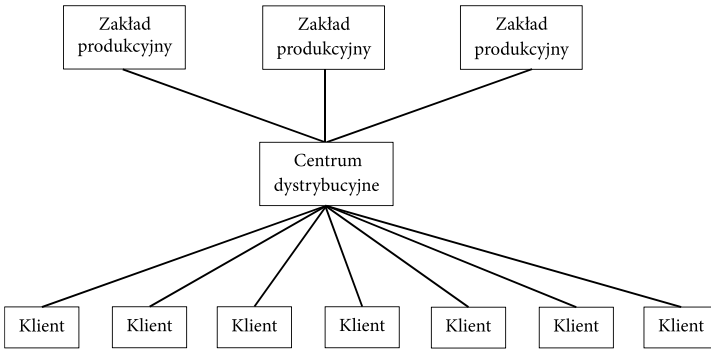
Źródło: opracowanie własne na podstawie (Rutkowski, 2005, s. 55–62).

Model ten nie jest bynajmniej pozbawiony wad i z jego stosowaniem wiążą się następujące problemy:

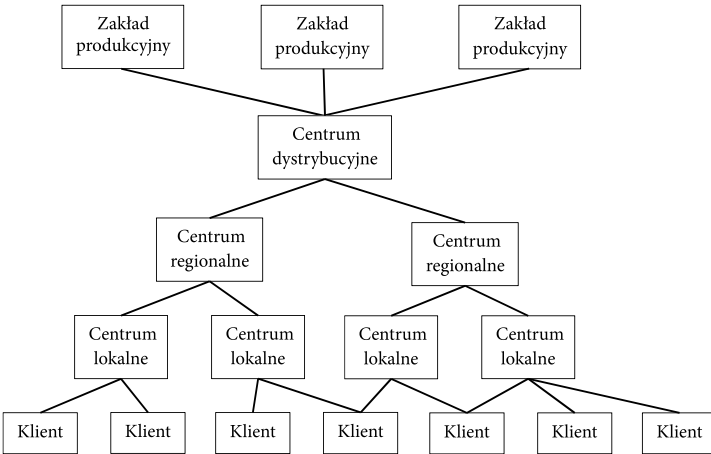
- większe obciążenie punktów sprzedaży detalicznej związane z procesami zamówień;
- brak zapasów bezpieczeństwa w razie pojawienia się problemów u producenta;
- większe koszty związane z transportem;
- większe koszty związane z obsługą transakcyjną u producenta (przy dużej liczbie małych odbiorców);
- mniejsza elastyczność w dostosowaniu się do zmian popytu u nabywców (np. okresy sezonowe, wakacyjne).

Drugi z wymienionych modeli dystrybucji – model scentralizowany – wprowadza do łańcucha dostaw dodatkowe ogniwo w postaci centrum dystrybucji (rys. 1.6). Nabywcy finalni obsługiwani są z jednego miejsca, które przejmuje od producenta na siebie ciężar realizacji ich zamówień. W miejscu tym utrzymywany jest zapas produktu, którego poziom dostosowywany jest do bieżących potrzeb rynku. Funkcję centrów dystrybucyjnych pełnią bardzo często parki magazynowe lub wyspecjalizowane centra logistyczne oferujące znacznie szerszy zakres usług. Podstawową zaletą centrów dystrybucyjnych jest obniżenie kosztów funkcjonowania i obsługi łańcucha logistycznego, w szczególności w sytuacji, kiedy nabywcy finalni są rozproszeni oraz znajdują się w znacznej odległości od producenta.

Przeciwnieństwem systemu scentralizowanego jest model zdecentralizowany (rys. 1.7), w którym dostawy produktów do nabywców finalnych realizowane są z wykorzystaniem składów i centrów dystrybucyjnych nie tylko producenta, ale także poszczególnych pośredników.



Rys. 1.6. Model scentralizowany
Źródło: opracowanie własne na podstawie (Rutkowski, 2005, s. 57).



Rys. 1.7. Model zdecentralizowany
Źródło: opracowanie własne na podstawie (Rutkowski, 2005, s. 58).

Omawiając wady i zalety dwóch ostatnich modeli dystrybucji, należy zauważyć, że bardzo często to, co jest korzyścią wypływającą z jednego z nich, jest jednocześnie wadą drugiego. Do podstawowych korzyści zastosowania modelu scentralizowanego zaliczyć można:

- możliwość konsolidacji przesyłek transportowych, a tym samym generowania oszczędności na transporcie;
- znaczną redukcję poziomu zapasów (w przeciwieństwie do systemu zdecentralizowanego);
- większą precyzję w badaniu popytu nabywców finalnych;
- niższe koszty stałe i zmienne związane z zaangażowaniem infrastruktury logistycznej;

- większą integrację łańcucha dostaw w wyniku centralnie sterowanego przepływu informacji.

Z kolei wśród zalet systemu zdecentralizowanego wymienia się między innymi:

- minimalizację globalnych kosztów fizycznej dystrybucji produktów (w szczególności przez producenta);
- wysoki poziom obsługi klienta (szybkie reakcje na zmieniające się potrzeby nabywców finalnych dzięki utrzymywanym zapasom w każdym punkcie pośrednim);
- krótki czas dostaw (im dłuższy kanał dystrybucji, tym bliżej klienta jest ostatnie jego ogniwo);
- relatywnie niższe koszty transportu na ostatnim etapie dostaw do klienta finalnego.

1.3. Procesy transportowe

Wśród procesów związanych z fizycznym przepływem produktów w łańcuchach dostaw jednym z najważniejszych jest proces transportowy. Należy go rozumieć jako zbiór wszelkiego rodzaju działań, których celem jest dokonanie przemieszczenia produktu z jednego lub kilku punktów początkowych (punktów nadania) do jednego lub kilku punktów końcowych (punktów odbioru) (Fertsch, 2006, s. 107). Na działania te składają się czynności organizacyjne, wykonawcze, a także transakcyjne (handlowe). Czynności organizacyjne to inaczej przygotowanie realizacji procesu transportowego (np. sporządzenie stosownej dokumentacji, wybór formy transportu i podmiotu, który się go podejmie, ustalenie tras przewozu). Wśród czynności wykonawczych wyróżnia się te, które są w bezpośredni sposób związane z fizycznym przemieszczeniem produktów od nadawcy do odbiorcy, oraz czynności manipulacyjne: załadunek i wyładunek, a także dojazd środka transportowego do miejsca załadunku, a następnie jego powrót do bazy (lub do kolejnego miejsca załadunku). Czynności transakcyjne natomiast związane są z zawieraniem umów zakupu usług transportowych (negocjacje z podmiotami oferującymi usługi transportowe, dokonywanie płatności itp.) (Mendyk, 2009, s. 189–195).

Ostateczny kształt procesu transportowego, w tym jego poszczególnych czynności, przeważnie jest wynikiem przyjętego w łańcuchu dostaw modelu dystrybucji. Poza tym każdy decydent musi zdawać sobie sprawę z tego, że proces transportowy składa się właśnie z wyżej wymienionych etapów, które bezpośrednio przekładają się na czas jego realizacji. Sam przewóz z punktu *A* do punktu *B* wcale nie musi być najdłuższą trwającą jego częścią. Analizując, ile czasu zajmuje każda z czynności poza fizycznym przewozem, mamy szansę poszukać ewentualnych

możliwości przyspieszenia całego procesu. Rozpatrując jednak pod tym kątem sam proces przewozowy, można w nim wskazać:

- czas przygotowania produktu u nadawcy (kompletacja, przygotowanie jednostki ładunkowej, czynności zdawczo-odbiorcze);
- czas załadunku (czynności manipulacyjne);
- czas przewozu do odbiorcy;
- czas wyładunku (czynności manipulacyjne);
- czas czynności zdawczo-odbiorczych (czynności kontrolne, umieszczenie produktu w miejscu przeznaczenia odbiorcy).

Czas przewozu niekiedy rozbija się na dodatkowe czynności, gdy np. trzeba dokonać przeładunku produktu z jednego środka transportowego na inny, co jest nieuniknione przy wykorzystaniu różnych form transportu.

Jeśli chodzi o funkcjonowanie łańcuchów dostaw, procesy transportowe są w nich kluczowe i nie ograniczają się tylko do zapewnienia fizycznego przemieszczenia produktu w ramach ich struktury. Przede wszystkim transport odgrywa rolę czynnika integrującego poszczególne ogniwa łańcucha dostaw, który stanowi o istnieniu oraz formie współpracy pomiędzy dostawcami a odbiorcami. Ponadto to między innymi dzięki odpowiedniej organizacji procesów transportowych możliwa jest właściwa realizacja powszechnie stosowanej strategii zarządzania logistycznego *just-in-time*. Optymalizacja czasu i kosztu związanego z procesami transportowymi w sposób bezpośredni przekłada się np. na poziom zapasów gromadzonych w poszczególnych ogniwach łańcuchów dostaw. Pamiętać przy tym należy, że tylko te dwa procesy fizycznego przepływu produktów (transport i magazynowanie) generują razem bardzo znaczący udział kosztów dystrybucyjnych¹.

W ujęciu przedmiotowym podstawowym pojęciem związanym z procesami transportowymi jest „ładunek”. Zgodnie z definicją zamieszczoną w *Słowniku terminologii logistycznej* jest to dobro materialne, które jest przemieszczane w łańcuchu dostaw (Fertsch, 2006). Dobrami materialnymi mogą być: surowce, materiały do produkcji, półprodukty, wyroby gotowe. Przy czym pojęcie ładunku nie musi być tożsame z partią produktu wysłaną od dostawcy do odbiorcy np. w ramach przewozu jednym środkiem transportowym. Ładunek należy traktować raczej jako pewien strumień produktów, który może być podzielony na poszczególne przesyłki z wykorzystaniem jednostek ładunkowych. Jednostką ładunkową zaś jest określona partia ładunku (np. wyrażona w sztukach lub kilogramach), która zachowuje swą formę podczas przeprowadzania poszczególnych czynności procesu przewozowego (np. załadunku, przewozu, wyładunku czy składowania). Wśród najczęściej stosowanych jednostek ładunkowych można wyróżnić:

¹ W literaturze przedmiotu uważa się, że koszty związane z procesami transportowymi i zarządzania zapasami stanowią co najmniej 50–70% całkowitych kosztów dystrybucji produktów w łańcuchach dostaw.

pojemniki i opakowania, palety oraz kontenery. Ze względu na wielkość ładunku procesy przewozowe można podzielić na drobne, całopojazdowe i ponadgabarytowe. Te dwa ostatnie dotyczą przewozów pomiędzy jednym dostawcą a jednym odbiorcą. W wypadku ładunków o mniejszych wielkościach możliwe jest przeprowadzanie procesów przewozowych od jednego lub kilku dostawców do jednego lub kilku odbiorców jednocześnie.

Ponadto mówiąc o ładunkach, można dokonać ich rozróżnienia także według innych kryteriów, tj.: podatności przewozowej, sposobów załadunku oraz wagi i objętości. Klasyfikacje te mają bardzo istotne znaczenie z punktu widzenia samej organizacji konkretnych procesów przewozowych. Przynależność ładunku do określonej grupy stanowi wyznacznik do formułowania konkretnych warunków, jakim podlegać muszą konkretne zadania przewozowe.

Biorąc pod uwagę podatność przewozową ładunku, wymienia się podatność naturalną, techniczną i ekonomiczną. Podatność naturalna odnosi się do rozróżnienia ładunków, które są z jednej strony niebezpieczne (stanowią bezpośrednie zagrożenie dla ludzi i/lub środowiska), a z drugiej strony są wrażliwe na warunki transportowe (są podatne na zmiany temperatury i wilgotności powietrza, czas trwania transportu, działanie światła, wstrząsy i uderzenia). Z kolei podatność techniczna rozpatruje ładunki według ich podstawowych cech fizycznych (ciała stałe, postać gazowa, postać płynna). Wreszcie podatność ekonomiczna dzieli ładunki na te o niskiej, średniej lub wysokiej wartości.

Kryterium sposobów załadunku uwzględnia takie czynności jak: napełnianie, nasypywanie, umieszczanie (np. podnoszenie, wtaczanie, wciąganie), które stanowią podstawę podziału ładunków. Ostatnie kryterium, dotyczące wagi i objętości, rozróżnia ładunki: całkowite, częściowe, drobnicowe i paczkowane.

Struktura łańcuchów dostaw oraz funkcjonujących w ich ramach systemów dystrybucji, a także rodzaj przewożonego ładunku determinują wybór środków transportu wykorzystywanych w procesach przewozowych. W procesach transportowych wykorzystywane są następujące podstawowe formy transportu (Stajniak i in., 2008, s. 16):

- transport lądowy (samochodowy i kolejowy);
- transport wodny (morski i śródlądowy);
- transport powietrzny (lotniczy);
- transport przewodowy (przesyłowy, rurociągowy);
- transport intermodalny (kombinowany).

Ostatni rodzaj transportu (intermodalny) polega na połączeniu przynajmniej dwóch spośród pozostałych czterech form. Najczęściej składają się na niego transport samochodowy i kolejowy lub kolejowy i morski i jest on charakterystyczny w szczególności dla procesów transportowych, w których odległości pomiędzy dostawcą a odbiorcą są znaczne. Korzystanie z transportu kombinowanego (np. w formie: transport samochodowy i kolejowy) ma także uzasadnienie

ekonomiczne w sytuacji, kiedy konieczne lub opłacalne jest skonsolidowanie przesyłek w większe jednostki ładunkowe.

Każdy z wymienionych podstawowych rodzajów transportu ma swoje wady i zalety. Jeśli chodzi o transport samochodowy, będą to:

- bardzo duża dostępność przestrzenna;
- duża dyspozycyjność;
- względna niezależność od warunków pogodowych;
- względna niezależność od innych form transportu;
- wysoki stopień zróżnicowania taboru dla różnych form i jednostek ładunkowych;
- ograniczone możliwości przewozów masowych;
- relatywnie wysokie koszty jednostkowe przewozu;
- relatywnie niski poziom bezpieczeństwa transportu.

W wypadku transportu kolejowego wady i zalety przedstawiają się następująco:

- stosunkowo duża dostępność przestrzenna, w tym także do innych form transportu;
- duże możliwości wygenerowania niskich kosztów w przeliczeniu na jednostkę ładunkową;
- wysoki stopień zróżnicowania taboru dla różnych form i jednostek ładunkowych;
- wysoki poziom niezawodności i regularności połączeń;
- relatywnie długi czas przewozu;
- brak dyspozycyjności czasowej;
- przewóz na ściśle określonych szlakach transportowych;
- relatywnie niski poziom bezpieczeństwa transportu.

Transport morski i śródlądowy charakteryzuje się natomiast:

- bardzo dużą zdolnością do przewozów masowych;
- bardzo dużymi możliwościami wygenerowania niskich kosztów jednostkowych transportu;
- globalnym zasięgiem;
- bardzo długim czasem przewozu;
- niską punktualnością i regularnością;
- zależnością od innych form transportu (w szczególności samochodowego i kolejowego);
- zależnością od zmieniających się warunków pogodowych;
- niskim poziomem bezpieczeństwa;
- znacznie ograniczoną dostępnością przestrzenną.

Do zalet i wad transportu lotniczego zaliczyć można:

- bardzo wysoką prędkość przewozu;
- wysoki poziom bezpieczeństwa;

- punktualność i rytmiczność transportu;
- bardzo wysokie koszty jednostkowe;
- zależność od innych form transportu (w szczególności samochodowego);
- dużą wrażliwość na warunki klimatyczne i pogodowe;
- ograniczoną dostępność przestrzenną.

Wreszcie ostatnia z przedstawionych wcześniej form transportu – transport przesyłowy – ma następujące wady i zalety:

- największą przepustowość (w szczególności gazów i cieczy);
- bardzo wysoką niezawodność;
- rytmiczność (regularność) transportu;
- niski i stały poziom kosztów jednostkowych;
- ograniczoną podatność asortymentową;
- relatywnie niską dostępność przestrzenną w transporcie na duże odległości.

Tabela 1.1. Porównanie form transportu

Kryterium	Forma transportu				
	samochodowy	kolejowy	wodny	lotniczy	przesyłowy
Koszt	4	3	2	5	1
Czas	2	3	4	1	–
Niezawodność	1	2	4	3	–
Zdolność transportowa	2	1	4	3	5
Dostępność przestrzenna	1	2	4	3	–
Bezpieczeństwo	2	3	4	1	–

1 – najlepiej, 5 – najgorzej

Źródło: (Coyle i in., 2002, s. 427).

Z przedstawionego powyżej zbioru wad i zalet, jakimi charakteryzują się poszczególne rodzaje środków transportu, można wywnioskować, że dotyczą one kilku najczęściej branych pod uwagę przez decydentów podstawowych cech, takich jak: koszt, czas, niezawodność (w szczególności regularność), zdolność transportowa (przepustowość, możliwości transportu masowego), dostępność przestrzenna oraz bezpieczeństwo. W tabeli 1.1 zestawiono wszystkie podstawowe rodzaje transportu i porównano je ze sobą z punktu widzenia wymienionych kryteriów, którymi kierują się decydenci, organizując i realizując procesy transportowe.

Na koniec rozważań dotyczących podstawowych zagadnień związanych z procesami transportowymi należy jeszcze nawiązać do aspektu podmiotowego, czyli wskazać głównych jego uczestników. W wypadku realizowania procesów

transportowych własnymi siłami uczestnikami procesu są tylko dostawca oraz odbiorca. Każdy z nich może jednak powierzyć wykonywanie tych czynności podmiotom zewnętrznym. Pierwszym z takich podmiotów są przewoźnicy, czyli podmioty gospodarcze realizujące fizyczny przewóz ładunku z punktu nadania do punktu odbioru. W takiej sytuacji mówimy o transporcie obcym. Z kolei przewoźnicy w łańcuchach dostaw mogą być usługodawcami zarówno dla dostawców lub odbiorców, jak i dla innego rodzaju podmiotów gospodarczych, czyli dla spedytorów. Rola spedytora w porównaniu do roli przewoźnika jest znacznie szersza. Jego zadaniem jest nie tylko realizacja fizycznego procesu przewozowego, ale całościowe przeprowadzenie procesu transportowego, a więc w pierwszej kolejności jego przygotowanie, a na końcu rozliczenie. Tym samym zbiór zadań spedytora jest o wiele większy i obejmuje między innymi (Rydzkowski i Wojewódzka-Król (red.), 2009, s. 254):

- wybór form i środków transportowych;
- ustalenie form i tras przewozów;
- wybór przewoźników i zakontraktowanie ich usług;
- odbiór przesyłek od dostawcy i przekazanie ich odbiorcy;
- wybór i dokonywanie operacji finansowych (płatności, ubezpieczenia, cła);
- przygotowywanie i prowadzenie dokumentacji związanej z transportem.

Wszystkie wymienione w tym podrozdziale podstawowe elementy procesów transportowych (ładunki transportowe, środki transportu, podmioty realizujące proces transportowy), a także przyjęty w łańcuchach dostaw model (system) dystrybucji stanowią bardzo istotne założenia do właściwej i zarazem optymalnej organizacji procesów transportowych, a w szczególności procesów przewozowych.

1.4. Organizacja procesów transportowych

Konkretna organizacja procesu transportowego w łańcuchu dostaw uzależniona jest od przyjętego systemu transportowego, którego podstawowym celem jest zapewnienie optymalnego poziomu obsługi odbiorców. Nierzadko systemy transportowe są tworzone specjalnie dla konkretnych łańcuchów dostaw lub wręcz konkretnych odbiorców funkcjonujących w ich ramach. Kształtowanie takiego systemu polega przede wszystkim na umiejętnym wykorzystaniu posiadanych zasobów i dopasowaniu oferowanych usług do potrzeb usługobiorców. Systemy przeznaczone dla konkretnego klienta są bardzo często droższe od standardowych rozwiązań, co wiąże się z brakiem możliwości optymalnego wykorzystania posiadanych zasobów transportowych. Jednakże są one zarazem lepiej dopasowane do potrzeb klienta, co podnosi poziom jego obsługi. Należy pamiętać, że jak najlepsze

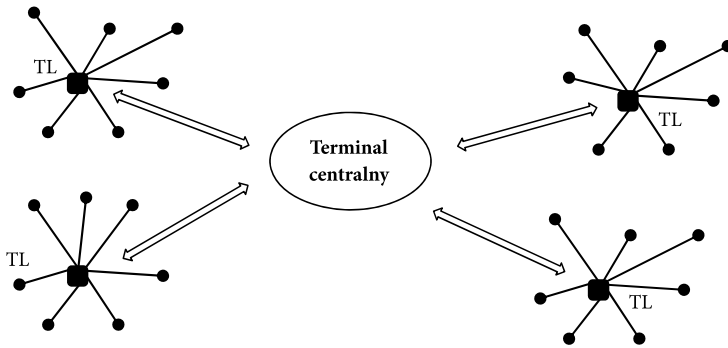
wykorzystanie zasobów, którymi dysponuje przedsiębiorstwo realizujące procesy transportowe, stanowi klucz do wysokiego poziomu efektywności jego działania. Z reguły wymienia się trzy podstawowe czynniki przyczyniające się do podniesienia tej efektywności (Tarkowski i in., 1995, s. 169):

- 1) zwiększenie wydajności jednostek ładunkowych (lepsze ich wykorzystanie);
- 2) zwiększenie w procesach transportowych udziału ładunków powrotnych (minimalizacja pustych przebiegów) od miejsca rozładunku do miejsca załadunku;
- 3) optymalne wykorzystanie czasu postoju środków transportowych zarówno pomiędzy kolejnymi zadaniami przewozowymi, jak i podczas obsługi dostawców i odbiorców.

Oprócz optymalnego wykorzystania zasobów w postaci środków transportu ważne jest także jak najlepsze wykorzystanie jednostek ładunkowych (np. naczep, kontenerów). Właściwa organizacja systemu transportowego może polegać np. na rozlokowaniu przez operatora logistycznego jednostek ładunkowych wśród dostawców i odbiorców w taki sposób, aby zminimalizować czas postoju u każdego z nich (puste kontenery lub naczepy są załadowywane przez dostawcę samodzielnie). Obok optymalnego wykorzystania zasobów jednym z najważniejszych elementów bezpośrednio wpływających na właściwą organizację procesu transportowego jest właściwe planowanie samych procesów przewozowych (stosowanie określonych modeli i metod rozwiązywania zadań przewozowych).

Systemy transportowe w pierwszej kolejności można podzielić z punktu widzenia geograficznego i organizacyjnego. Tego rodzaju klasyfikacje wynikają z konstrukcji łańcuchów dostaw, które one obsługują. Według kryterium geograficznego wyróżnia się systemy lokalne obejmujące np. aglomerację miejską lub inny niewielki wyodrębniony geograficznie region. Charakteryzują się one przewozami na niewielkie odległości (np. przewozy kurierskie, dostawy z centrum lokalnego dystrybucji do sieci sprzedaży detalicznej). Systemy transportowe o szerszym zasięgu to systemy ogólnokrajowe i międzynarodowe, w których ramach mogą być realizowane zarówno przewozy na niewielkie, jak i znaczne odległości. Systemy ogólnokrajowe, a w szczególności międzynarodowe mogą wymagać funkcjonowania dodatkowych punktów między dostawcami a odbiorcami w postaci terminali przeładunkowych. Bardzo często znajdują się one w miejscach, gdzie jest dostęp przynajmniej do dwóch form transportu (terminale morskie, lotnicze i kolejowe).

Z punktu widzenia organizacyjnego oraz pod względem liczby wykorzystywanych terminali systemy transportowe można podzielić na jedno- i wieloterminalowe, w tym wśród tych drugich wyodrębnić można systemy równorzędne i hierarchiczne (Tarkowski i in., 1995, s. 173). System jednoterminalowy charakteryzuje się jednym centralnie zlokalizowanym terminalem, stanowiącym punkt centralny systemu (rys. 1.8). Taki system transportowy może być stosowany w prezentowanym wcześniej scentralizowanym modelu dystrybucji.



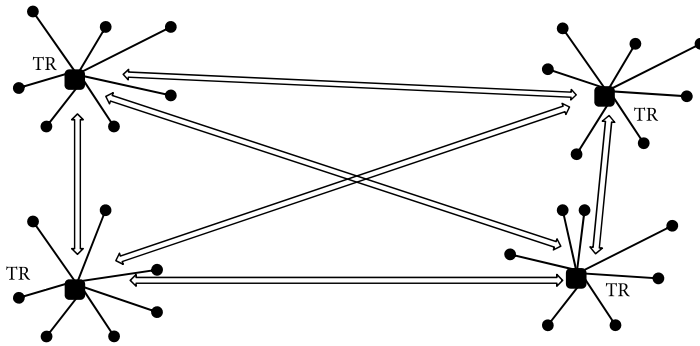
Rys. 1.8. System jednoterminalowy (TL – terminale lokalne)
Źródło: opracowanie własne na podstawie (Tarkowski i in., 1995, s. 173).

W terminalu centralnym odbywa się rozdział strumienia ładunku od np. producenta do terminali lokalnych przy zachowaniu tej samej jednostki ładunkowej. W celu obsługi punktów odbioru mogą współpracować z nim dodatkowo terminale lokalne, dla których budowane są konkretne zadania przewozowe (trasy dostaw i/lub odbioru produktów). W terminalu lokalnym następuje dekompletacja ładunków zbiorczych i rozdzielanie ich na ładunki mniejsze przeznaczone do docelowych miejsc przeznaczenia. Do zalet systemu jednoterminalowego zaliczyć można jednolitość podejmowanych czynności w całym systemie transportowym, a także możliwość wykorzystania efektów skali w procesach przewozowych, w szczególności jeżeli zastosowany zostanie transport skonsolidowanych przesyłek (np. kolejowy).

Bardziej rozbudowanym systemem transportowym jest system wieloterminalowy, który zdaje egzamin zwłaszcza przy dużej liczbie przesyłek drobnych (np. przewozy kurierskie), gdy trzeba pokryć obszar geograficzny o znacznych rozmiarach. Może mieć on charakter systemu równorzędnego (rys. 1.9) lub hierarchicznego (rys. 1.10).

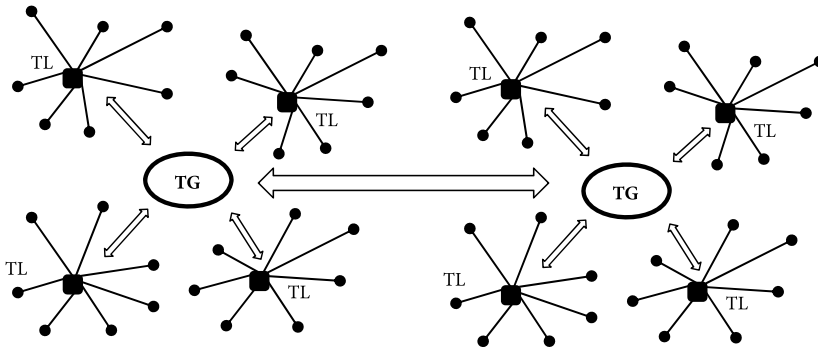
W systemie wieloterminalowym równorzędnym każdy terminal ma przydzielony swój rejon działania i obsługuje zlokalizowane tam punkty odbioru (nadania) produktu. Ponadto terminale te mogą tworzyć strukturę sieciową, w której ramach nadawana przesyłka, zanim trafi do odbiorcy, może przejść przez więcej niż dwa terminale przeładunkowe (początkowy i końcowy).

W wypadku systemu wieloterminalowego hierarchicznego tworzone są przynajmniej dwa poziomy pomiędzy samymi terminalami. Zastosowanie tego systemu sprawdza się w sytuacji, gdy np. na duże odległości przewożone są małe przesyłki, ale na niewielką skalę. Wyróżnić tu można terminale główne i terminale odpowiednio niższego rzędu. Zazwyczaj terminalowi głównemu przyporządkowany jest pewien zbiór terminali niższego rzędu (np. lokalnych). Taka struktura systemu transportowego jest niewątpliwie najdroższa z punktu widzenia infrastruktury oraz kosztów dokonywania operacji logistycznych, lecz może się z nią wiązać najwyższy stopień wykorzystania środków transportowych. Ponadto taki system transportowy ma największe predyspozycje do pokrycia geograficznie bardzo dużego obszaru dystrybucji.



Rys. 1.9. System wieloterminalowy równorzędny (TR – terminal regionalny)

Źródło: opracowanie własne na podstawie (Tarkowski i in., 1995, s. 176).



Rys. 1.10. System wieloterminalowy hierarchiczny (TG – terminal główny)

Źródło: opracowanie własne na podstawie (Tarkowski i in., 1995, s. 176).

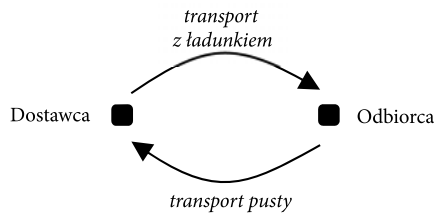
W ramach prezentowanych powyżej struktur systemów transportowych konstruowane są, a następnie rozwiązywane konkretne zadania transportowe realizujące procesy przewozowe. Konstrukcja tych zadań wynika nie tylko z bezpośredniej struktury kanału dystrybucji w ramach łańcucha dostaw czy dopasowanego do niego systemu transportowego, ale bardzo istotne są także dodatkowe warunki stawiane systemom transportowym, do których należą między innymi:

- konieczność dostawy, odbioru bądź dostawy i odbioru produktów w ogniwach łańcucha dostaw;
- ograniczenia czasowe dotyczące zarówno pracy samych środków transportowych, jak i pracy dostawców i odbiorców;
- ograniczenia dotyczące możliwości przewozowych poszczególnych środków transportowych, a także stopnia ich wykorzystania;
- ograniczenia co do pewności danych niezbędnych do budowy konkretnych zadań przewozowych (losowość parametrów);
- charakter i liczba celów, jakie są stawiane przed systemami transportowymi.

Ogólnie najczęściej spotykane modele dotyczące organizacji procesów przewozowych, na podstawie których budowane są konkretne zadania przewozowe, można podzielić na (Stajniak i in., 2008, s. 89–91):

- modele wahadłowe;
- modele promieniste;
- modele obwodowe;
- modele sztafetowe.

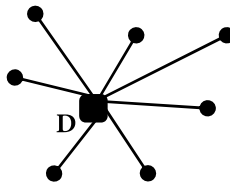
W modelu wahadłowym (rys. 1.11) środki transportowe dokonują regularnych kursów pomiędzy punktem dostawy a punktem odbioru, przy czym w przeciwną stronę pojazd wraca pusty. Jako przykład posłużyć może całopojazdowy transport materiałów budowlanych pomiędzy składem budowlanym a miejscem budowy.



Rys. 1.11. Model wahadłowy

Źródło: opracowanie własne na podstawie (Stajniak i in., 2008, s. 89).

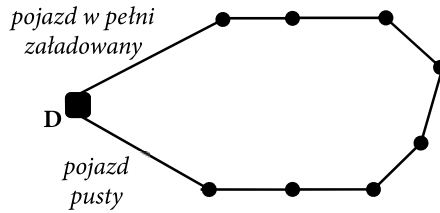
Model promienisty jest naturalnym rozwinięciem poprzedniego modelu, jednak dostarczenie z jednego miejsca nadania następuje do wielu miejsc odbioru, przy czym jeden pojazd wysyłany jest tylko do jednego miejsca odbioru (rys. 1.12). Taki model ma zastosowanie w realizacji systemów transportowych, w których np. jeden magazyn centralny (centrum dystrybucji) zaopatruje w produkty wiele magazynów regionalnych (np. centrów lokalnych).



Rys. 1.12. Model promienisty

Źródło: opracowanie własne na podstawie (Stajniak i in., 2008, s. 90).

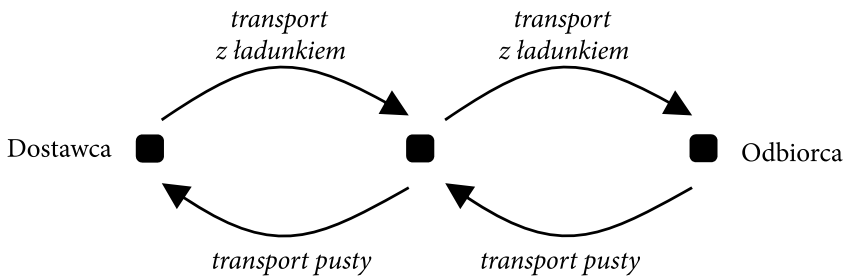
Z kolei ideą modelu obwodowego (rys. 1.13) jest dokonywanie procesów przewozowych polegających na jednoczesnej obsłudze wielu punktów odbioru przez jeden pojazd. Pojazd wyrusza od dostawcy i w każdym z punktów odbioru zostawia część swojego ładunku. Taki model jest charakterystyczny dla systemów dystrybucji produktów masowych i drobnych (np. dystrybucja produktów spożywczych).



Rys. 1.13. Model obwodowy

Źródło: opracowanie własne na podstawie (Stajniak i in., 2008, s. 91).

Ostatni z modeli to model sztafetowy (rys. 1.14), który polega na wykorzystaniu tzw. procesu cross-docking, czyli przeładunku produktu pomiędzy pojazdami (najczęściej z mniejszego na większy). Model ten w szczególności jest spotykany w wieloszczelowych (hierarchicznych) systemach transportowych.



Rys. 1.14. Model sztafetowy

Źródło: opracowanie własne na podstawie (Stajniak i in., 2008, s. 91).

Przykładami pojedynczych zadań przewozowych budowanych na podstawie zaprezentowanych powyżej modeli są problemy transportowe typu:

- jeden do jednego, w którym jest jeden dostawca i jeden odbiorca produktu (np. zadanie wyznaczenia najkrótszej drogi lub maksymalnego przepływu transportu);
- jeden do wielu lub wielu do jednego, w którym jest jeden dostawca oraz pewna liczba odbiorców produktu (np. zadania układania tras jednego i wielu pojazdów);
- wielu do wielu, w którym jest wielu dostawców i wielu odbiorców tego samego produktu (np. klasyczne zadanie transportowe – układania tras wielu pojazdów z wieloma bazami).

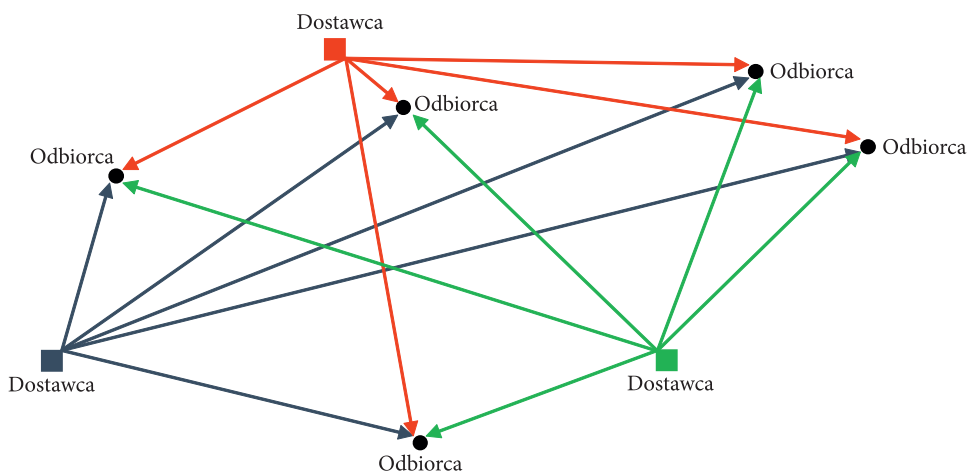
W kolejnych rozdziałach niniejszej pracy skupimy się na dwóch ostatnich typach, a w szczególności na zadaniach stanowiących najszerszą klasę: układania tras jednego i wielu pojazdów.

Rozdział 2

Klasyczny problem transportowy

Jednym z najbardziej znanych transportowych problemów optymalizacji jest klasyczny problem transportowy. Z jednej strony jest on przykładem codziennego podejmowania decyzji o charakterze operacyjnym w logistyce dystrybucji pojedynczych przedsiębiorstw (np. spedycyjnych), jak również wielu podmiotów składających się na łańcuchach dostaw. Z drugiej strony klasyczny problem transportowy stanowi przykład zastosowania zagadnień programowania liniowego.

Ogólnie klasyczne zagadnienie transportowe określić można jako problem najtańszego przewozu jednorodnego dobra pomiędzy pewną liczbą punktów nadania (dostawcami) a określoną liczbą punktów odbioru (odbiorcami) (rys. 2.1).



Rys. 2.1. Graficzna ilustracja klasycznego problemu transportowego
Źródło: opracowanie własne.

2.1. Sformułowanie problemu decyzyjnego

Aby sformułować klasyczny problem transportowy, należy przyjąć następujące oznaczenia:

- x_{ij} – wielkość dostawy od dostawcy i do odbiorcy j , czyli ilość transportowanego dobra na trasie (i, j) , gdzie $i = 1, 2, \dots, m$ oraz $j = 1, 2, \dots, n$;
- a_i – zasób dobra u dostawcy i , czyli podaż dostawcy i , gdzie $i = 1, 2, \dots, m$;
- b_j – zapotrzebowanie na dobro u odbiorcy j , czyli popyt odbiorcy j , gdzie $j = 1, 2, \dots, n$;
- c_{ij} – wysokość kosztu dostawy jednostki dobra od dostawcy i do odbiorcy j , czyli koszt jednostkowy transportu na trasie (i, j) , gdzie $i = 1, 2, \dots, m$ oraz $j = 1, 2, \dots, n$;
- r_{ij} – dopuszczalna wielkość transportu od dostawcy i do odbiorcy j , czyli przepustowość na trasie (i, j) , gdzie $i = 1, 2, \dots, m$ oraz $j = 1, 2, \dots, n$.

Zmiennymi decyzyjnymi klasycznego problemu transportowego są wielkości x_{ij} , których liczba równa jest iloczynowi liczby dostawców oraz liczby odbiorców: $m \times n$. Z racji tego, że wielkość przewozu nie może być ujemna, wszystkie zmienne decyzyjne muszą spełniać warunek nieujemności: $x_{ij} \geq 0$.

Założenia klasycznego problemu transportowego są następujące:

- 1) zgodnie z przyjętą na wstępie definicją dobro transportowane pomiędzy dostawcami a odbiorcami ma charakter jednorodny;
- 2) każdy z odbiorców musi otrzymać żądaną ilość produktu, co oznacza, że całkowita podaż wszystkich dostawców jest nie mniejsza niż łączny popyt zaopatrywanych przez nich odbiorców:

$$\sum_{i=1}^m a_i \geq \sum_{j=1}^n b_j \quad (2.1)$$

Wreszcie ostatnim podstawowym założeniem klasycznego zagadnienia transportowego jest:

- 3) przyjęcie nieograniczonej przepustowości wszystkich tras $r_{ij} = \infty$ pomiędzy i -tym dostawcą a j -tym odbiorcą: $0 \leq x_{ij} \leq r_{ij}$.

Rozpatrując szczególny przypadek zadania transportowego, w którym łączna podaż wszystkich dostawców jest równa łącznemu zapotrzebowaniu wszystkich odbiorców¹, można sformułować model matematyczny programowania liniowego w następujący sposób (2.2)–(2.5):

$$\min. F(\mathbf{x}) = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (2.2)$$

¹ Przyjęcie takiego założenia czyni zadanie transportowe zadaniem zamkniętym (zbilansowanym).

$$\sum_{j=1}^n x_{ij} = a_i \quad (2.3)$$

$$\sum_{i=1}^m x_{ij} = b_j \quad (2.4)$$

$$x_{ij} \geq 0 \quad (2.5)$$

dla każdego $i = 1, 2, \dots, m$ oraz $j = 1, 2, \dots, n$.

Funkcja określona wzorem (2.2) reprezentuje łączny koszt transportu przewożonego dobra pomiędzy wszystkimi dostawcami a wszystkimi odbiorcami.

Ograniczenia wyrażone równaniami (2.3) i (2.4) przedstawiają bilanse dla dostawców oraz dla odbiorców. Ilość produktu dostarczona od i -tego dostawcy nie może być większa (w wypadku zadania zamkniętego jest równa) od jego podaży a_i . Jednocześnie ilość dobra dostarczona do j -tego odbiorcy musi być równa jego zapotrzebowaniu b_j .

Poszukiwany jest taki program przewozowy, w którym koszt całkowity transportu będzie najmniejszy.

Sformułowane powyżej klasyczne zadanie transportowe w postaci modelu programowania liniowego ma kilka istotnych właściwości. Pierwsza z nich dotyczy rozwiązania optymalnego. Otóż jeżeli wielkości popytu i podaży są nieujemne, to zadanie transportowe ma zawsze rozwiązanie optymalne.

Druga z właściwości dotyczy współliniowości równań (2.3)–(2.4). Dowolnie wybrane z tej grupy równanie jest zawsze liniową kombinacją pozostałych $m + n - 1$ równań. Właściwość ta jest o tyle istotna, że określa liczbę niezerowych zmiennych w rozwiązaniu optymalnym. W optymalnym programie przewozowym liczb tras, na których nastąpi przewóz (wykorzystanych połączeń pomiędzy dostawcami a odbiorcami), będzie nie większa niż $m + n - 1$.

2.2. Warianty klasycznego problemu transportowego

W literaturze przedmiotu słusznie postawione zostało pytanie o sens rozpatrywania klasycznego problemu transportowego oraz o jego zastosowania praktyczne (Ignasiak (red.), 1996, s. 60). Niewątpliwie jest to szczególnie przypadkowy interesujący zastosowania zagadnienia programowania liniowego, w którym należy sformułować postaci zmiennych decyzyjnych, funkcje kryterium ich oceny oraz warunki ograniczające.

Ponadto w klasycznej postaci zastosowania praktyczne problemu transportowego nie występują tak często, jak mogłoby się wydawać. Wynika to między innymi

z tego, że praktyczne sytuacje decyzyjne mają zazwyczaj bardziej skomplikowany charakter (np. dodatkowe warunki ograniczające). Poniżej zaprezentowane zostaną wybrane modyfikacje klasycznego zadania transportowego.

2.2.1. Zadanie transportowe otwarte

Jednym z podstawowych rodzajów problemu transportowego jest zadanie otwarte (niezbilansowane). Różnica w stosunku do modelu przedstawionego wzorami (2.2)–(2.5) jest w tym wypadku taka, że ograniczenia dotyczące bilansów dla dostawców (2.3) przyjmują postać następujących nierówności:

$$\sum_{j=1}^n x_{ij} \leq a_i \quad (2.6)$$

Jednakże każde zadanie transportowe o charakterze otwartym można przekształcić w zadanie transportowe zamknięte – przez powiększenie listy odbiorców o dodatkowy punkt obioru $n + 1$. Zapotrzebowanie nowego punktu odbioru równe jest różnicy pomiędzy całkowitą podażą dostawców a całkowitym popytem odbiorców zadania wyjściowego (2.7):

$$b_{n+1} = \sum_{i=1}^m a_i - \sum_{j=1}^n b_j \quad (2.7)$$

Można przyjąć założenie, że ten dodatkowy punkt odbioru stanowią magazyny znajdujące się przy dostawcach, a tym samym koszty jednostkowe transportu od dostawcy do magazynu są równe 0. Z kolei wielkość transportu od i -tego dostawcy do nowego odbiorcy $n + 1$ jest wielkością dobra pozostawioną w magazynie i -tego dostawcy.

2.2.2. Zadanie transportowe z kryterium czasu

Przedstawione klasyczne zadanie transportowe charakteryzuje się funkcją celu, która reprezentuje całkowity koszt transportu dobra pomiędzy wszystkimi dostawcami a odbiorcami. Jednakże wyróżnić można zadanie transportowe z funkcją celu reprezentującą łączny czas przewozu dobra pomiędzy dostawcami a odbiorcami. W takim wypadku funkcja wyrażona wzorem (2.2) może mieć postać:

$$\min. T(\mathbf{x}) = \sum_{i=1}^m \sum_{j=1}^n t_{ij} x_{ij} \quad (2.8)$$

gdzie t_{ij} oznacza czas przewozu jednostki dobra pomiędzy i -tym dostawcą a j -tym odbiorcą. Pozostałe ograniczenia (2.3)–(2.5) pozostają bez zmian. Użyteczność problemów transportowych z tak sformułowanym kryterium czasu może być

szczególnie widoczna w wypadku problemów transportowych, w których występują całopojazdowe przewozy dużych jednostek ładunkowych (np. kontenerów), a jednocześnie przy ograniczonej liczbie środków transportowych decydentowi zależy na tym, aby łączny czas przewozu realizowany przez wszystkie środki transportu był jak najkrótszy.

Drugim rodzajem zadania transportowego z kryterium czasu jest sytuacja, w której celem jest jak najszybsze zakończenie całego planu dostaw. Oczywiście musi być to uwarunkowane odpowiednią liczbą środków transportowych, które mogą być wykorzystane w tym samym czasie. W takim wypadku postać funkcji celu będzie następująca (2.9):

$$\min. F(\mathbf{x}) = \max_{(i,j):x_{ij}>0} \{t_{ij}\} \quad (2.9)$$

Tutaj t_{ij} oznacza czas przejazdu (realizacji całego transportu) pomiędzy i -tym dostawcą a j -tym odbiorcą. Powyższa funkcja celu przedstawia sytuację, w której poszukuje się takiego rozwiązania (planu dostaw), gdzie najdłużej trwający przewóz zakończy się jak najszybciej z punktu widzenia wszystkich możliwych dopuszczalnych planów przewozowych. Pozostałe ograniczenia (2.3)–(2.5) nie ulegają zmianie, jednakże tak sformułowana funkcja celu powoduje, że takie zadanie transportowe nie jest zadaniem programowania liniowego.

2.2.3. Zadanie transportowe z blokadą tras

Dotychczasowe zadania transportowe przedstawiały przyjęte na początku założenie, które dotyczyło nieograniczonej przepustowości wszystkich tras. Jednakże często występują sytuacje, w których decydent ma do czynienia z częściową lub całkowitą blokadą trasy lub tras.

Całkowita blokada trasy (i, j) oznacza sytuację, w której w rozwiązaniu optymalnym nie jest pożądane wystąpienie przewozu pomiędzy i -tym dostawcą a j -tym odbiorcą. W takim wypadku modyfikacja wyjściowego zadania transportowego polega na zastąpieniu kosztu jednostkowego c_{ij} dowolnie dużą liczbą M . Nie oznacza to wcale, że rozwiązanie zadania zawsze będzie uwzględniało ten warunek (Ignasiak (red.), 1996, s. 66). Może się okazać, że układ warunków ograniczających (2.3)–(2.4) w zakresie podaży a_i oraz popytu b_j będzie tak skonstruowany, że rozwiązanie zawierać będzie wszystkie nieujemne przewozy na trasach należących np. do wiersza i lub kolumny j , w których występuje blokowana trasa.

W wypadku częściowej blokady trasy (i, j) zakłada się sytuację, w której przewóz pomiędzy i -tym dostawcą a j -tym odbiorcą nie może być większy od przyjętego ograniczenia r_{ij} . Modyfikacja wyjściowego zadania transportowego może polegać na podzieleniu kolumny (j -tego odbiorcy) na dwie części (osobne kolumny) lub wiersza (i -tego dostawcy) na dwie części (osobne wiersze). W pierwszej kolumnie

(lub wierszu) popyt (lub podaż) jest równy postawionemu ograniczeniu częściowemu r_{ij} , natomiast w drugiej kolumnie (lub wierszu) jest równy różnicy $b_j - r_{ij}$ (lub $a_j - r_{ij}$).

2.2.4. Wielokryteriove zadanie transportowe

Sytuacja, gdy poszukiwane jest rozwiązanie optymalne problemu decyzyjnego, w którym rozważanych jest więcej niż jedna funkcja kryterium, nosi nazwę zadania programowania wielokryterialnego.

Dotychczas omawiane warianty klasycznego problemu transportowego przedstawiały modele, w których optymalizowana była w danym momencie tylko jedna funkcja celu. Był nią albo całkowity koszt transportu pomiędzy dostawcami a odbiorcami, albo czas (traktowany w dwojaki sposób: jako całkowity czas wszystkich przewozów lub najdłuższy przewóz ze wszystkich koniecznych do wykonania). Jednakże obydwie wielkości (koszt i czas) mogą być rozpatrywane jednocześnie. Należy tutaj zauważyć, że znalezienie takiego programu przewozowego, który optymalizowałby obydwa kryteria naraz, a w szczególności gdy nie są one wprost od siebie zależne (są konfliktowe)², należy do rzadkości. Dlatego też najczęściej spotykanymi sytuacjami są pewne rozwiązania o charakterze kompromisowym, stanowiące swego rodzaju pośrednie plany dostaw przewozowych pomiędzy rozwiązaniami optymalnymi zorientowanymi tylko na jedno optymalizowane kryterium.

Jak pokazują A. Całczyński i in. (2000), przy rozpatrywaniu dwóch kryteriów: czasu i kosztu, najczęściej obserwuje się trzy następujące przypadki:

- 1) Celem dominującym jest znalezienie rozwiązania optymalnego, które minimalizuje łączny koszt transportu całego programu przewozowego. Jeśli zadanie transportowe ma więcej niż jedno równorzędne rozwiązanie optymalne zorientowane na minimalizację kosztu (występują alternatywne rozwiązania optymalne zadania kosztowego), należy wybrać to, dla którego czas realizacji programu przewozowego jest najkrótszy.
- 2) Celem dominującym jest znalezienie rozwiązania optymalnego, które minimalizuje łączny czas przewozu całego programu przewozowego. Jeśli zadanie transportowe ma więcej niż jedno równorzędne rozwiązanie optymalne zorientowane na minimalizację czasu (występują alternatywne rozwiązania optymalne zadania z kryterium czasu), należy wybrać to, dla którego koszt realizacji programu przewozowego jest najmniejszy.
- 3) Decydent ma do dyspozycji pewien dopuszczalny przedział czasowy stanowiący różnicę pomiędzy przyjętym granicznym czasem przewozu a minimalnym

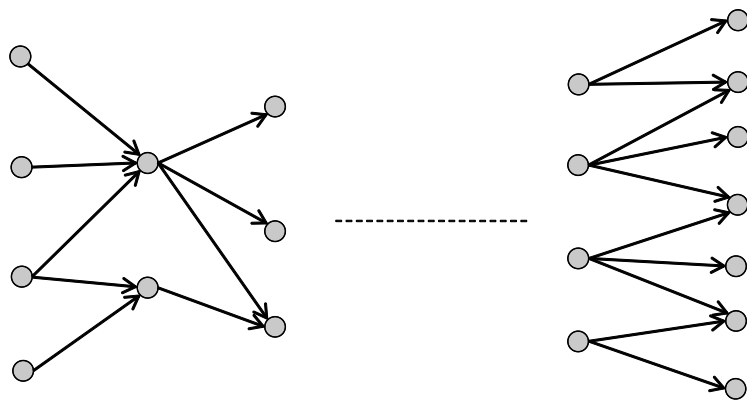
² To znaczy sytuacje, które występują bardzo często np. w aglomeracjach miejskich, gdzie trasa korzystniejsza z punktu widzenia pokonanego dystansu nie musi oznaczać trasy krótszej z punktu widzenia czasu.

możliwym czasem przewozu. W takiej sytuacji należy znaleźć rozwiązanie optymalne zadania w granicach przyjętego przedziału czasowego (zorientowanego na kryterium czasu), a następnie wyznaczyć rozwiązanie optymalne z zastosowaniem kryterium kosztowego mieszczącego się w przyjętych granicach.

Przedstawiony powyżej trzeci przypadek daje decydentowi większe możliwości poszukiwania rozwiązania optymalnego z punktu widzenia kryterium kosztu, ponieważ występuje tu więcej alternatywnych decyzji dopuszczalnych niż w sytuacji przedstawionej w punkcie drugim.

2.2.5. Wieloszczeblowe zadanie transportowe

Opisane powyżej problemy transportowe prezentowały łańcuch dostaw w jego najprostszej formie: bezpośredniej relacji dostawca–odbiorca. Jednakże relacje te stanowią nierzadko fragmenty o wiele bardziej skomplikowanych struktur w logistyce dystrybucji. W rzeczywistości konieczność stosowania nowoczesnych strategii zarządzania logistycznego, tj. ECR (*Efficient Consumer Response* – efektywnej obsługi klienta), wymusza na decydentach formułowanie także bardziej złożonych modeli problemów przewozowych (Baraniecka, 2004)³. Przykładem takiego problemu transportowego jest wieloszczeblowe (wieloetapowe) zadanie transportowe.



Rys. 2.2. Graficzna ilustracja wieloszczeblowego problemu transportowego

Źródło: opracowanie własne.

Najprostszym przykładem wieloszczeblowego zadania transportowego z kolei jest model trójszczeblowy (dwuetapowy), np. producenci–centra dystrybucji–sklepy detaliczne. Jednakże stopień złożoności łańcucha dostaw może być znacznie

³ Efektywna obsługa klienta jest strategią wymagającą ścisłej współpracy pomiędzy wieloma ogniwami łańcucha dostaw.

większy (dowolny), a powstanie przez dodanie kolejnych różnego rodzaju ogniów, jak np. hurtowników, regionalnych przedstawicieli sprzedaży itp.

Wieloszczeblowe zadanie transportowe może zostać zaklasyfikowane na dwa sposoby. Pierwszy z nich dokonuje podziału na zadanie, w którym:

- 1) ilość produktu przewożonego pomiędzy początkowymi dostawcami a finalnymi odbiorcami nie ulega zmianie przy przechodzeniu przez punkty pośrednie;
- 2) ilość produktu przewożonego pomiędzy początkowymi dostawcami a finalnymi odbiorcami ulega określonym zmianom przy przechodzeniu przez punkty pośrednie.

Druga ze wspomnianych klasyfikacji opiera się na podziale wieloszczeblowych zadań transportowych ze względu na możliwości dotyczące przepustowości punktów pośrednich:

- 1) przepustowość punktów pośrednich jest ograniczona wielkością r_k ($k = 1, 2, \dots, p$) – zadanie z ograniczeniami;
- 2) przepustowość punktów pośrednich jest nieograniczona (zadanie bez ograniczeń).

Ograniczenia przepustowości poszczególnych punktów pośrednich mogą być ustanowione np. w wyniku limitowanych możliwości magazynowych lub przeładunkowych centrów dystrybucji bądź centrów logistycznych. W tego typu zadaniach należy przyjąć założenie, że łączna przepustowość wszystkich punktów pośrednich danego szczebla powinna być większa od łącznego popytu wszystkich odbiorców finalnych (2.10).

$$\sum_{k=1}^p r_k > \sum_{j=1}^n b_j \quad (2.10)$$

Jeśli wystąpi równość pomiędzy łączną przepustowością punktów pośrednich danego szczebla a łącznym popytem odbiorców finalnych, wtedy wieloszczeblowe zadanie transportowe może zostać przekształcone w zbiór niezależnych od siebie klasycznych zbilansowanych zadań transportowych, które byłyby formułowane dla każdego etapu oddzielnie.

Wieloszczeblowe zadanie transportowe może zostać sprowadzone do postaci klasycznego zadania transportowego niezależnie od tego, czy łączna podaż wszystkich dostawców jest równa łącznemu popytowi, czy też jest większa od łącznego popytu wszystkich odbiorców finalnych. Ponadto sposób przekształcenia zadania wieloetapowego w klasyczny problem transportowy jest taki sam w wypadku, gdy przepustowość punktów pośrednich jest ograniczona lub gdy nie ma nałożonych limitów. Jeżeli przepustowość punktów pośrednich jest nieograniczona, wtedy można przyjąć założenie, że ich górny limit jest równy nieistotnej z punktu widzenia zadania wielkości (np. jest to bardzo duża liczba lub równa co najmniej łącznej wielkości przewozów, jakie mają zostać wykonane).

Przekształcenie wieloszczęblowego zadania transportowego w klasyczne zadanie transportowe polega na takim zbudowaniu zadania transportowego, gdzie liczba dostawców oraz liczba odbiorców powiększone zostaną – odpowiednio – o liczbę punktów pośrednich. Oznacza to, że punkty pośrednie poszczególnych szczebli przyjmują charakter zarówno dostawców, jak i odbiorców. Jednocześnie koszty jednostkowe transportu pomiędzy samymi punktami pośrednimi są równe 0. Z kolei koszty jednostkowe przewozu pomiędzy różnymi punktami pośrednimi, lecz należącymi do tego samego szczebla (etapu), równe są ustalonej bardzo dużej liczbie M (gdzie $M \rightarrow \infty$). Co więcej, w związku z tym, że nie ma możliwości bezpośredniego przewozu produktu pomiędzy dostawcami a odbiorcami, a także punktami pośrednimi niesąsiadującymi ze sobą szczebli, trasy tych połączeń muszą być blokowane przez wprowadzenie w miejsce kosztu jednostkowego odpowiednio dużej liczby M . Powyższe ograniczenie dotyczy także połączeń od punktów pośrednich szczebla wyższego do punktów pośrednich szczebla niższego (połączenia zwrotne).

Przykład

W zamieszczonych poniżej tabelach 2.1 i 2.2 przedstawiono zadanie wyjściowe dla trójszczęblowego zadania transportowego. Szczebel pierwszy stanowi m dostawców zgłaszających podaż o wielkości: a_1, a_2, \dots, a_m . Szczebel drugi wieloetapowego zadania transportowego to p punktów pośrednich o przepustowościach wynoszących odpowiednio: r_1, r_2, \dots, r_p . Wreszcie ostatni szczebel reprezentowany jest przez liczbę n odbiorców końcowych w wartościach popytu wynoszących: b_1, b_2, \dots, b_n . Koszt jednostkowy transportu pomiędzy i -tym dostawcą a j -tym punktem pośrednim wynosi $c_{ij}^{(1)}$ (gdzie $i = 1, 2, \dots, m$ oraz $j = 1, 2, \dots, p$). Z kolei koszt jednostkowy transportu pomiędzy i -tym punktem pośrednim a j -tym odbiorcą finalnym wynosi $c_{ij}^{(2)}$ (gdzie $i = 1, 2, \dots, p$ oraz $j = 1, 2, \dots, n$).

Tabela 2.1. Pierwszy i drugi szczebel wieloetapowego zadania transportowego (pierwszy etap)

	Pośrednik 1	Pośrednik 2	Pośrednik p	Podaż
Dostawca 1	$c_{11}^{(1)}$	$c_{12}^{(1)}$	$c_{1p}^{(1)}$	a_1
Dostawca 2	$c_{21}^{(1)}$	$c_{22}^{(1)}$	$c_{2p}^{(1)}$	a_2
.....
Dostawca m	$c_{m1}^{(1)}$	$c_{m2}^{(1)}$	$c_{mp}^{(1)}$	a_m
Przepustowość	r_1	r_2	r_p	

Źródło: opracowanie własne.

Tabela 2.2. Drugi i trzeci szczebel wieloetapowego zadania transportowego (drugi etap)

	Odbiorca 1	Odbiorca 2	Odbiorca n	Przepustowość
Pośrednik 1	$c_{11}^{(2)}$	$c_{12}^{(2)}$	$c_{1n}^{(2)}$	r_1
Pośrednik 2	$c_{21}^{(2)}$	$c_{22}^{(2)}$	$c_{2n}^{(2)}$	r_2
.....
Pośrednik p	$c_{p1}^{(2)}$	$c_{p2}^{(2)}$	$c_{pn}^{(2)}$	r_p
Popyt	b_1	b_2	b_n	

Źródło: opracowanie własne.

Ponieważ łączny popyt wszystkich n odbiorców końcowych wynosi $B = \sum_{j=1}^n b_j$, można przyjąć⁴, że w wypadku zadania wieloszczeblowego bez ograniczeń przepustowości wszystkich p punktów pośrednich drugiego szczebla będą wynosić $r_i = B$.

Klasykne zadanie transportowe skonstruowane na podstawie zadania wyjściowego zostało przedstawione w tabeli 2.3.

Tabela 2.3. Klasykne zadanie transportowe dla trójszczeblowego problemu transportowego

	Pośrednik 1	Pośrednik 2	Pośrednik p	Odbiorca 1	Odbiorca 2	Odbiorca n	Podaż
Dostawca 1	$c_{11}^{(1)}$	$c_{12}^{(1)}$	$c_{1p}^{(1)}$	M	M	M	a_1
Dostawca 2	$c_{21}^{(1)}$	$c_{22}^{(1)}$	$c_{2p}^{(1)}$	M	M	M	a_2
.....
Dostawca m	$c_{m1}^{(1)}$	$c_{m2}^{(1)}$	$c_{mp}^{(1)}$	M	M	M	a_m
Pośrednik 1	0	M	M	$c_{11}^{(2)}$	$c_{12}^{(2)}$	$c_{1n}^{(2)}$	r_1
Pośrednik 2	M	0	M	$c_{21}^{(2)}$	$c_{22}^{(2)}$	$c_{2n}^{(2)}$	r_2
.....
Pośrednik p	M	M	0	$c_{p1}^{(2)}$	$c_{p2}^{(2)}$	$c_{pn}^{(2)}$	r_p
Popyt	r_1	r_2	r_p	b_1	b_2	b_n	

Źródło: opracowanie własne.

4 Można też przyjąć, że $r_i = M$, gdzie $i = 1, 2, \dots, p$.

Dokonując uogólnienia powyższego przykładu trójszczebłowego (dwuetapowego) zadania transportowego, należy przyjąć następujące oznaczenia:

- m – liczba dostawców;
- n – liczba odbiorców finalnych;
- s – liczba szczebli pośrednich;
- t – liczba etapów, gdzie $t = s + 1$;
- $p(k)$ – liczba pośredników szczebla k , gdzie $k = 2, 3, \dots, s$;
- D_i – oznaczenie dostawcy (pierwszy szczebel), gdzie $i = 1, 2, \dots, m$;
- O_j – oznaczenie odbiorcy finalnego (ostatni szczebel), gdzie $j = 1, 2, \dots, n$;
- $R_{i(k)}$ – oznaczenie pośrednika szczebla k , gdzie $i = 1, 2, \dots, p(k)$ oraz $k = 2, 3, \dots, s$;
- a_i – podaż dostawcy i , gdzie $i = 1, 2, \dots, m$;
- b_j – popyt odbiorcy finalnego j , gdzie $j = 1, 2, \dots, n$;
- $r_{i(k)}$ – przepustowość pośrednika szczebla k , gdzie $i = 1, 2, \dots, p(k)$ oraz $k = 2, 3, \dots, s$;
- $c_{ij}^{(l)}$ – koszt jednostkowy transportu pomiędzy i -tym dostawcą (lub i -tym punktem pośrednim) a j -tym odbiorcą (lub j -tym punktem pośrednim) następnego w kolejności szczebla) na l -tym etapie, gdzie $l = 1, 2, \dots, t$.

Konstrukcja klasycznego zadania transportowego dla dowolnej liczby punktów pośrednich oraz dowolnej liczby szczebli (etapów) została pokazana w tabeli 2.4.

Tabela 2.4. Ogólna postać klasycznego zadania transportowego dla wieloszczebłowego problemu transportowego

	$R_{1(2)}$	\dots	$R_{p(2)}$	\dots	$R_{1(s)}$	\dots	$R_{p(s)}$	O_1	\dots	O_m	Podaż
D_1	$c_{11}^{(1)}$	\dots	$c_{1p(2)}^{(1)}$	\dots	$c_{11}^{(t-1)}$	\dots	$c_{1p(s)}^{(t-1)}$	$c_{11}^{(t)}$	\dots	$c_{1n}^{(t)}$	a_1
\vdots	\vdots		\vdots		\vdots		\vdots	\vdots		\vdots	\vdots
D_m	$c_{m1}^{(1)}$	\dots	$c_{mp(2)}^{(1)}$	\dots	$c_{m1}^{(t-1)}$	\dots	$c_{mp(s)}^{(t-1)}$	$c_{m1}^{(t)}$	\dots	$c_{mn}^{(t)}$	a_m
$R_{1(2)}$	0	\dots	M	\dots	M	\dots	M	M	\dots	M	$r_{1(2)}$
\vdots	\vdots		\vdots		\vdots		\vdots	\vdots		\vdots	\vdots
$R_{p(2)}$	M	\dots	0	\dots	M	\dots	M	M	\dots	M	$r_{p(2)}$
\vdots	\vdots		\vdots		\vdots		\vdots	\vdots		\vdots	\vdots
$R_{1(s)}$	M	\dots	M	\dots	0	\dots	M	$c_{11}^{(t)}$	\dots	$c_{1n}^{(t)}$	$r_{1(s)}$
\vdots	\vdots		\vdots		\vdots		\vdots	\vdots		\vdots	\vdots
$R_{p(s)}$	M	\dots	M	\dots	M	\dots	0	$c_{p1}^{(t)}$	\dots	$c_{pn}^{(t)}$	$r_{p(s)}$
Popyt	$r_{1(2)}$	\dots	$r_{p(2)}$	\dots	$r_{1(s)}$	\dots	$r_{p(s)}$	b_1	\dots	b_n	

Źródło: opracowanie własne.

2.2.6. Zadanie transportowo-produkcyjne

W poprzednich podrozdziałach przedstawiono różne warianty klasycznego problemu transportowego, które dotyczyły tylko przewozów produktu. Jednakże bardzo częstymi przypadkami zadań transportowych (w szczególności z logistyki dystrybucji) są zadania transportowo-produkcyjne lub transportowo-magazynowe. Wtedy istotny jest nie tylko koszt przewozu, ale również koszt produkcji lub koszt magazynowania. Nierzadko oszczędności poczynione na organizacji procesów transportowych mogą prowadzić do wzrostu kosztów związanych np. z utrzymaniem zapasów. Skrócenie czasu realizacji dostaw wpływa w sposób bezpośredni na średni poziom utrzymania zapasów i brak korekt w tym zakresie skutkować może wzrostem kosztów z tytułu zarządzania zapasami.

Najprostszy przykład zadania transportowo-produkcyjnego można przedstawić w następujący sposób⁵. Z jednej strony znana jest liczba m producentów (a jednocześnie dostawców) pewnego jednorodnego dobra, których możliwości produkcyjne wynoszą odpowiednio a_i . Z drugiej strony, tak jak w wypadku klasycznego zadania transportowego, znana jest liczba n odbiorców tego dobra, którzy zgłaszają popyt w wielkości b_j . Oprócz jednostkowych kosztów transportu c_{ij} pomiędzy i -tym producentem a j -tym odbiorcą znany jest także jednostkowy koszt produkcji p_i i -tego producenta. Zadaniem decydenta jest wyznaczenie optymalnego planu zarówno przewozu, jak i produkcji w taki sposób, aby całkowity koszt będący sumą kosztów transportu i produkcji był jak najniższy (2.11). W tym ujęciu plan produkcji należy rozumieć jako wielkość produkcji i -tego producenta (lub stopień wykorzystania jego mocy produkcyjnych).

$$\min. F(\mathbf{x}) = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + \sum_{i=1}^m p_i \sum_{j=1}^n x_{ij} \quad (2.11)$$

Do tak sformułowanego problemu decyzyjnego znajduje zastosowanie klasyczny problem transportowy. Dostawcami są poszczególni producenci, których moce produkcyjne należy traktować jako podażę dobra. Ponieważ łączna moc produkcyjna wszystkich producentów jest większa od całkowitego zapotrzebowania wszystkich odbiorców, należy zastosować przekształcenie jak w wypadku niezbilansowanego problemu transportowego⁶. Jednakże tutaj fikcyjny odbiorca, którego popyt jest wyrażony równaniem (2.7), reprezentuje niewykorzystane moce produkcyjne wszystkich producentów. Tym samym w tabeli przewozów zmienna x_{m+1} nie przedstawia wielkości transportu na trasie od i -tego producenta ani jego

5 Inne wersje zadania transportowo-produkcyjnego, np. z uwzględnieniem nieliniowej funkcji kryterium, znaleźć można m.in. w (Całczyński i in., 2000).

6 W przeciwnym razie koszty produkcji nie są rozważane, ponieważ cała moc produkcyjna i tak musi zostać wykorzystana.

wielkości produkcji. Wartość tej zmiennej decyzyjnej reprezentuje niewykorzystaną moc produkcyjną i -tego producenta.

Modyfikacji podlega także macierz kosztów jednostkowych. Koszt jednostkowy połączenia pomiędzy i -tym producentem a j -tym odbiorcą jest równy sumie kosztów przewozu c_{ij} oraz kosztu produkcji p_i u i -tego producenta. W wypadku odbiorcy fikcyjnego $n + 1$ koszty jednostkowe wynoszą 0. Tak przekształcone wyjściowe zadanie transportowe ma postać klasycznego zadania transportowego (tab. 2.5).

Tabela 2.5. Postać zadania transportowo-produkcyjnego

	Odbiorca 1	Odbiorca 2	Odbiorca n	Odbiorca n	Podaż
Producent 1	$p_1 + c_{11}$	$p_1 + c_{12}$	$p_1 + c_{1n}$	0	a_1
Producent 2	$p_2 + c_{21}$	$p_2 + c_{22}$	$p_2 + c_{2n}$	0	a_2
.....
Producent m	$p_m + c_{m1}$	$p_m + c_{m2}$	$p_m + c_{mn}$	0	a_m
Popyt	b_1	b_2	b_n	b_{n+1}	

Źródło: opracowanie własne.

2.2.7. Zagadnienie przydziału

Zagadnienie przydziału jest problemem decyzyjnym nieodnoszącym się w sposób bezpośredni do problemów przewozu produktu pomiędzy dostawcami a odbiorcami. Jednakże może być ono traktowane jako szczególny przypadek zadania transportowego.

Problem decyzyjny optymalnego przydziału można przedstawić w następujący sposób. Dysponując pewną liczbą m środków (narzędzi), należy zrealizować n celów (zadań). Każdorazowe zaangażowanie i -tego środka do realizacji j -tego celu wiąże się z poniesieniem kosztu w wielkości c_{ij} (lub osiągnięciem odpowiedniej korzyści). Należy tu także zaznaczyć, że przyporządkowanie środków do realizacji założonych celów musi być dokonane w sposób wzajemnie jednoznaczny. Oznacza to, że jeden cel może być zrealizowany z wykorzystaniem tylko jednego środka, jak również jeden środek może zostać wykorzystany jeden raz do realizacji danego celu. Należy zauważyć, że liczby środków i celów mogą być sobie równe (wtedy każdy cel jest zrealizowany oraz zostaną wykorzystane wszystkie środki). Model takiej postaci zadania przydziału został przedstawiony poniżej (2.12)–(2.15).

$$\min. (\max.) F(\mathbf{x}) = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \tag{2.12}$$

$$\sum_{j=1}^n x_{ij} = 1 \tag{2.13}$$

$$\sum_{i=1}^n x_{ij} = 1 \quad (2.14)$$

$$x_{ij} = \begin{cases} 0 \\ 1 \end{cases} \quad (2.15)$$

gdzie $i = 1, 2, \dots, n$ oraz $j = 1, 2, \dots, n$.

Liczba środków może być także większa od liczby celów lub też odwrotnie – może być od niej mniejsza. W takiej sytuacji ograniczenia (2.13) i (2.14) przyjmują postać odpowiednich nierówności.

Należy zauważyć, że tak sformułowane zadanie przydziału może być potraktowane jako szczególny przypadek zadania transportowego, w którym podaże dostawców (środki) są równe 1 oraz popyty odbiorców (cele) są równe 1 (tab. 2.6). Zmienne decyzyjne odpowiadające w zadaniu transportowym wielkościom przewozu na poszczególnych trasach mogą wynosić 0 lub 1. Natomiast w wypadku zadania przydziału z kryterium maksymalizacji należy dokonać odpowiedniego przekształcenia wyjściowej funkcji celu $f(x)$ w funkcję $f'(x)$ w taki sposób, aby można było poszukiwać jej wartości minimalnej⁷.

Tabela 2.6. Przykład dopuszczalnego rozwiązania zadania przydziału o n środkach i n celach

	Cel 1	Cel 2	Cel $n - 1$	Cel n	Podaż
Środek 1		1			1
Środek 2				1	1
.....
Środek $m - 1$			1		1
Środek m	1				1
Popyt	1	1	1	1	n

Źródło: opracowanie własne.

2.2.8. Minimalizacja pustych przebiegów

Interesującym zagadnieniem optymalizacyjnym, do którego rozwiązania zastosowanie znajduje klasyczny problem transportowy, jest zadanie minimalizacji pustych przebiegów. W zarządzaniu łańcuchem dostaw (lub szerzej – siecią dostaw) jest to bardzo ważny problem decyzyjny, mający ugruntowaną pozycję w procesie projektowania przepływów fizycznych produktów pomiędzy dostawcami a odbiorcami.

⁷ Przykładem takiego przekształcenia może być: $f'(x) = M - f(x)$, gdzie $M \rightarrow \infty$.

W zagadnieniu minimalizacji pustych przebiegów przyjmuje się założenie, że dostawca jest jednocześnie odbiorcą, stąd też istnieje sieć wzajemnych połączeń pomiędzy n punktami, do których produkty są dostarczane, jak również są z nich wywożone. Znane są zarówno wielkości wywozu, jak i wielkości przywozu a_{ij} pomiędzy punktami i a j . Wielkość ta może być traktowana jako liczba środków transportu lub jednostek ładunkowych (np. kontenerów) niezbędnych do obsługi dostawy. Tym samym można dla każdego dostawcy (oraz jednocześnie odbiorcy) obliczyć wielkość w_i wywożonego produktu z i -tego punktu:

$$w_i = \sum_{j=1}^n a_{ij} \quad (2.16)$$

oraz wielkość p_j przywożonego produktu do j -tego punktu z pozostałych punktów:

$$p_j = \sum_{i=1}^n a_{ij} \quad (2.17)$$

gdzie $i = 1, 2, \dots, n$ oraz $j = 1, 2, \dots, n$ oraz $i \neq j$.

Jeżeli wystąpi sytuacja, w której ilość wywożonego z punktu i produktu do pozostałych punktów jest większa od ilości przywożonego do tego punktu produktu ($w_i > p_i$), oznacza to, że należy do tego punktu dostarczyć puste środki transportu (jednostki ładunkowe), aby można było zrealizować nadwyżkę wywozów nad przywozami. Środki transportu powinny być dostarczone przez punkty, w których przywóz jest większy od wywozu ($w_i < p_i$). Tym samym punkty te stałyby się dostawcami pustych środków transportu, natomiast punkty dysponujące nadwyżką wywozów nad przywozami byłyby odbiorcami pustych pojazdów. W wypadku punktów, dla których zachodzi zerowy bilans wywozów i przewozów ($w_i = p_i$), nie są one ani dostawcami, ani odbiorcami pustych środków transportu, czyli są pomijane.

Rozwiązanie zadania z minimalizacją pustych przebiegów sprowadza się do konstrukcji klasycznego problemu transportowego, w którym przewożonym dobrem są puste środki transportu. Dostawcy pustych środków transportu, czyli punkty, gdzie $w_i < p_i$, dysponują ich podażą w wielkości $p_i - w_i$, natomiast odbiorcy, czyli punkty, gdzie $w_i > p_i$, zgłaszają na nie popyt w wielkości: $w_i - p_i$. Macierz kosztów jednostkowych stanowi macierz fizycznych lub czasowych odległości pomiędzy rozpatrywanymi punktami.

Rozwiązanie optymalne tak skonstruowanego klasycznego zadania transportowego pozwala ustalić, od którego punktu i do którego punktu j oraz ile należy wysłać pustych środków transportu, aby zminimalizować ich łączne puste przebiegi liczone np. liczbą przebytych kilometrów.

2.3. Algorytm transportowy

Klasyczne zagadnienie transportowe jest problemem programowania liniowego, w związku z czym może być rozwiązane dobrze znaną metodą simpleks. Jednakże ze względu na swoją szczególną postać jest ono rozwiązywane za pomocą odrębnego algorytmu transportowego, w którym – począwszy od określonego dopuszczalnego rozwiązania początkowego – poszukiwane jest rozwiązanie coraz lepsze, aż do momentu osiągnięcia rozwiązania optymalnego.

Pierwszą czynnością algorytmu transportowego jest wygenerowanie początkowego dopuszczalnego planu przewozowego. Początkowy program przewozowy może mieć dowolną postać, lecz w literaturze przedmiotu znaleźć można kilka najpopularniejszych metod jego tworzenia, do których należą między innymi:

- metoda kąta północno-zachodniego;
- metoda minimalnego kosztu macierzy kosztów jednostkowych;
- metoda VAM-Szwarcza.

Uzyskany dopuszczalny program przewozowy będzie zawierał nie więcej niż $m + n - 1$ niezerowych tras, na których wystąpi transport pomiędzy danym dostawcą a danym odbiorcą. Jest on pierwszym punktem przestrzeni rozwiązań o wartości funkcji celu będącej całkowitym kosztem transportu pomiędzy dostawcami a odbiorcami. Celem algorytmu transportowego jest poszukiwanie coraz lepszych punktów przestrzeni rozwiązań przez modyfikację rozwiązania reprezentowanego przez aktualnie rozpatrywany punkt.

Algorytm transportowy można podzielić na dwa etapy. W etapie pierwszym sprawdza się, czy aktualnie rozpatrywany program przewozowy jest optymalny. Drugi etap to przekształcenie ocenianego programu przewozowego w inny, lepszy (tańszy) w wypadku, gdy dokonana wcześniej (w etapie pierwszym) ocena optymalności okaże się negatywna.

Etap oceny optymalności programu przewozowego polega na obliczeniu wskaźników optymalności Δ_{ij} według formuły (2.18):

$$\Delta_{ij} = c_{ij} - (u_i + v_j) \quad (2.18)$$

Powyższe wskaźniki obliczane są dla tras, dla których w aktualnie rozpatrywanym programie przewozowym wielkość transportu wynosi 0. Jeżeli dla ocenianego planu dostaw wszystkie Δ_{ij} są nieujemne, wtedy reprezentuje on rozwiązanie optymalne. W przeciwnym razie należy przejść do etapu drugiego algorytmu transportowego, czyli przekształcenia ocenianego programu przewozowego w kolejny, tańszy.

Przedstawiony algorytm transportowy wymaga sformułowania kilku istotnych spostrzeżeń. Pierwsze z nich dotyczy wskaźników optymalności. W programie przewozowym reprezentującym rozwiązanie optymalne (wszystkie $\Delta_{ij} \geq 0$) mogą

wystąpić wskaźniki optymalności równe 0. Liczba $\Delta_{ij} = 0$ determinuje wielkość zbioru alternatywnych rozwiązań optymalnych.

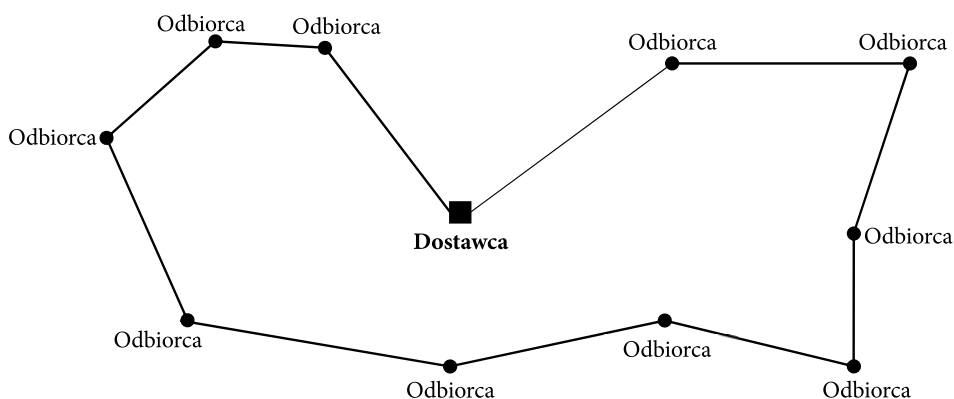
Jak już zauważono, optymalny plan dostaw wykorzystuje z wszystkich możliwych $m \times n$ połączeń tylko część z nich – nie więcej niż $m + n - 1$. Jeżeli program przewozowy składa się z mniejszej liczby niepustych tras niż $m + n - 1$, wtedy nosi on nazwę zdegenerowanego planu dostaw. Różnica między $m + n - 1$ a faktyczną liczbą niepustych tras jest stopniem degeneracji. Taki program przewozowy dalej można traktować jako plan dostaw składający się z $m + n - 1$ tras, lecz trzeba pamiętać, które z nich (gdzie wielkość przewozu wynosi 0) należą do rozpatrywanego (ocenianego) planu dostaw.

Rozdział 3

Układanie trasy jednego pojazdu

Jeżeli klasyczne zagadnienie transportowe jest jednym z najbardziej znanych problemów optymalizacji w transporcie, to niewątpliwie problem komiwojażera uznać należy za jedno z najstarszych tego typu zagadnień. Przy czym należy on do najczęściej pojawiających się problemów operacyjnych logistyki dystrybucji.

Zagadnienie komiwojażera znane jest w literaturze przedmiotu jako problem podróżującego sprzedawcy (*traveling salesman problem* lub *traveling salesperson problem*). Samo sformułowanie tego decyzyjnego problemu logistycznego jest bardzo proste i można je przedstawić następująco (rys. 3.1): celem dostawcy jest dostarczenie za pomocą jednego pojazdu żądanej ilości produktu do zadanej liczby odbiorców w taki sposób, aby koszt realizacji dostawy poniesiony przez dostawcę (lub podmiot organizujący przewóz) był jak najmniejszy.



Rys. 3.1. Graficzna ilustracja problemu komiwojażera
Źródło: opracowanie własne.

Po raz pierwszy tak sformułowany problem komiwojażera przedstawiony został przez W.R. Hamiltona w 1859 roku, natomiast jego pierwsze ujęcie jako problemu matematycznego zaprezentowane zostało w 1934 roku przez H. Whitneyego (Całczyński, 1992).

3.1. Sformułowanie problemu decyzyjnego

Problem komiwojażera jest zagadnieniem nie tylko logistycznym, ale może nawet przede wszystkim problemem matematycznym, a w szczególności kombinatorycznym. Stąd też zamieszczony powyżej słowny opis problemu decyzyjnego może być niewystarczający. W literaturze przedmiotu można się zresztą spotkać z różnymi sposobami jego formułowania.

Jednym z najczęstszych sposobów prezentacji zagadnienia komiwojażera jest wykorzystanie języka programowania matematycznego. Pojazd, wyruszając z pewnego punktu D (bazy dostawcy), ma za zadanie odwiedzić n innych punktów: O_1, O_2, \dots, O_n , a następnie powrócić do punktu początkowego¹. Celem jest ustalenie takiego porządku odwiedzanych odbiorców, aby całkowita długość (koszt) trasy pojazdu była jak najmniejsza. Kolejność odwiedzania poszczególnych odbiorców O_j jest dowolna, jednakże każdy z nich może być odwiedzony przez pojazd tylko jeden raz.

Budowę modelu programowania matematycznego dla przedstawionego problemu komiwojażera należy rozpocząć od zdefiniowania zmiennych decyzyjnych. Zmienna decyzyjna oznaczona jako x_{ij} reprezentuje występowanie w trasie pojazdu połączenia pomiędzy dostawcą a odbiorcą lub pomiędzy odbiorcami. Ze względu na to, że mogą wystąpić tylko dwa przypadki, ma ona postać binarną:

$$x_{ij} = \begin{cases} 0 \\ 1 \end{cases} \quad (3.1)$$

Zmienna x_{ij} przyjmuje wartość 1, jeżeli w aktualnie rozpatrywanej trasie pojazdu występuje połączenie pomiędzy punktem i a punktem j , natomiast 0 przyjmuje w sytuacji, gdy takiego połączenia brak.

Model programowania matematycznego dla przedstawionego problemu komiwojażera należy do klasy modeli liniowych i całkowitoliczbowych. Przedstawić go można w następujący sposób (3.2)–(3.5) (Miller i in., 1960):

$$\min. F(\mathbf{x}) = \sum_{i=0}^n \sum_{j=0}^n c_{ij} x_{ij} \quad (3.2)$$

$$\sum_{i=0}^n x_{ij} = 1 \quad \text{dla } j = 0, 1, \dots, n \quad (3.3)$$

$$\sum_{j=0}^n x_{ij} = 1 \quad \text{dla } i = 0, 1, \dots, n \quad (3.4)$$

1 W dalszej części przez punkt o indeksie i lub j równym 0 definiowane będzie pojęcie dostawcy (lub bazy, z której wyrusza pojazd na swoją trasę), natomiast i oraz j większe od 0 oznaczają indeksy punktów reprezentujących poszczególnych odbiorców.

$$z_i - z_j + (n+1)x_{ij} \leq (n+1) - 1 \text{ dla } i, j = 1, \dots, n \text{ oraz } i \neq j \quad (3.5)$$

Funkcja celu (3.2) przedstawia długość (lub zdefiniowany odpowiednio koszt) pokonanej przez pojazd trasy. Współczynnik c_{ij} oznacza odległość (lub koszt) przejazdu pomiędzy punktem i a punktem j . Dla $i = j$ przyjęć należy założenie, że odległość (koszt) c_{ij} jest równa dowolnie dużej liczbie M .

Szczególnych wyjaśnień wymagają ograniczenia w prezentowanym modelu matematycznym. Ponieważ każdy punkt $j = 1, 2, \dots, n$ musi zostać odwiedzony przez pojazd tylko jeden raz, ograniczenia występujące w omawianym modelu przedstawione są w postaci równości (3.3) i (3.4). Oznaczają one, odpowiednio, że pojazd może tylko jeden raz wjechać do danego punktu i raz go opuścić.

Specjalnego wyjaśnienia wymaga warunek określony zespołem nierówności (3.5). Zapewnia on, że do trasy pojazdu muszą należeć wszystkie odwiedzane punkty. Liczba tych nierówności jest znacząca i wynosi $(n+1)(n+1) - n$. Zawierają one łącznie n zmiennych decyzyjnych z_i oraz z_j , których optymalny dobór ma zapewnić odwiedzenie przez pojazd wszystkich punktów². Innymi słowy, warunki (3.5) zapewniają, że z elementów zbioru $\{i, j / x_{ij} = 1\}$ można ułożyć następujący ciąg: $(0, 1), (1, 2), (2, 3), \dots, (n-1, n), (n, 0)$ (Solich, 1974).

Przykład

Niech k oznacza liczbę odwiedzonych punktów przez komiwojażera w niepełnej trasie, przy czym $k < n$. Jednocześnie niech p oznacza odwiedzone punkty w tej trasie. Tym samym trasę tę zapisać można w następujący sposób: $(p_0, p_1, p_2, \dots, p_k, p_0)$. Dla każdego połączenia pomiędzy punktami p_i a p_j tej niepełnej trasy przedstawione zostały poniżej ograniczenia (3.5), które po zsumowaniu stronami prowadzą do sprzeczności.

$$\begin{aligned} z_{p_0} - z_{p_1} + (n+1) &\leq n \\ z_{p_1} - z_{p_2} + (n+1) &\leq n \\ \dots & \\ z_{p_k} - z_{p_0} + (n+1) &\leq n \end{aligned} \quad (3.6)$$

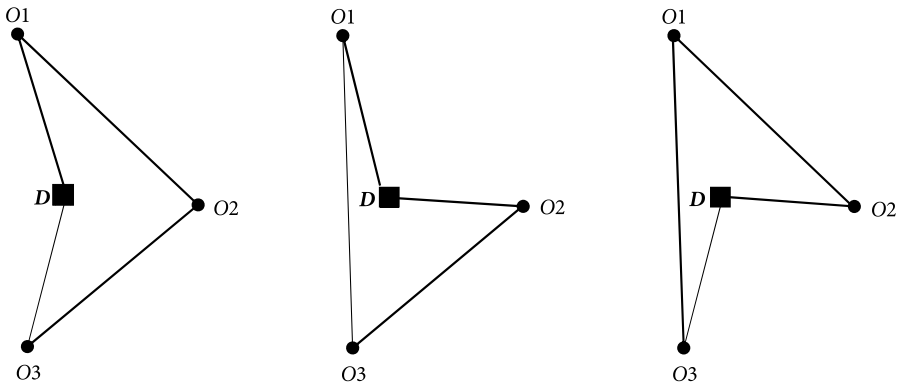
Brak w modelu przedstawionych powyżej ograniczeń (3.5) czyni z zagadnienia komiwojażera zadanie przydziału (3.2)–(3.4), którego jedno z rozwiązań zostało zilustrowane poniżej (rys. 3.4). W omawianym modelu zmienne z_i oraz z_j są dowolnymi liczbami rzeczywistymi, choć jak stwierdzają J.J. Finkelsztein i A.A. Korbut (1974), bez zmniejszenia ogólności można założyć, że mogłyby one przyjmować wartości całkowite nieujemne.

² $z_i, z_j \in R$.

W przedstawionym wzorami (3.2)–(3.5) modelu jednego komiwojażera liczba zmiennych decyzyjnych x_{ij} wynosi $(n + 1)^2$. Natomiast liczba poszczególnych ograniczeń jest równa: $n + 1$ (3.2), $n + 1$ (3.4), $n^2 - n$ (3.5). Czyli dla przedstawionej ilustracji graficznej problemu komiwojażera, gdzie liczba odbiorców wynosi $n = 10$ (rys. 3.1), liczba zmiennych decyzyjnych jest równa 121, natomiast liczba wszystkich ograniczeń wynosi 112.

Przykład

Na rysunku 3.2 przedstawiony został problem komiwojażera, w którym dostawca ma za zadanie obsłużyć trzech odbiorców. Pokazano wszystkie możliwe układy połączeń pomiędzy odbiorcami. Każdy z układów tras prezentuje dwie możliwe trasy przy założeniu, że można pokonać je w obydwu kierunkach. Oznacza to, że komiwojażer (pojazd) dla rozpatrywanego przykładu może pokonać sześć różnych tras.



Rys. 3.2. Graficzna ilustracja możliwych tras problemu komiwojażera dla $n = 3$
Źródło: opracowanie własne.

Zgodnie z modelem wyrażonym wzorami (3.2)–(3.5) model programowania matematycznego dla powyższego przykładu jest następujący:

$$\begin{aligned} \min. F(\mathbf{x}) = & Mx_{00} + c_{01}x_{01} + c_{02}x_{02} + c_{03}x_{03} + \\ & c_{10}x_{10} + Mx_{11} + c_{12}x_{12} + c_{13}x_{13} + \\ & c_{20}x_{20} + c_{21}x_{21} + Mx_{22} + c_{23}x_{23} + \\ & c_{30}x_{30} + c_{31}x_{31} + c_{32}x_{32} + Mx_{33} \end{aligned} \tag{3.7}$$

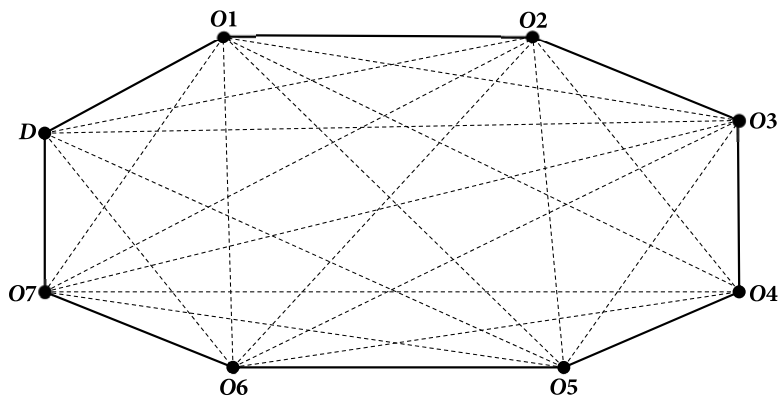
$$\begin{aligned} x_{00} + x_{10} + x_{20} + x_{30} &= 1 \\ x_{01} + x_{11} + x_{21} + x_{31} &= 1 \\ x_{02} + x_{12} + x_{22} + x_{32} &= 1 \\ x_{03} + x_{13} + x_{23} + x_{33} &= 1 \end{aligned} \tag{3.8}$$

$$\begin{aligned}
 x_{00} + x_{01} + x_{02} + x_{03} &= 1 \\
 x_{10} + x_{11} + x_{12} + x_{13} &= 1 \\
 x_{20} + x_{21} + x_{22} + x_{23} &= 1 \\
 x_{30} + x_{31} + x_{32} + x_{33} &= 1
 \end{aligned} \tag{3.9}$$

$$\begin{aligned}
 z_1 - z_2 + 4x_{12} &\leq 3 \\
 z_1 - z_3 + 4x_{13} &\leq 3 \\
 z_2 - z_3 + 4x_{23} &\leq 3 \\
 z_2 - z_1 + 4x_{21} &\leq 3 \\
 z_3 - z_1 + 4x_{31} &\leq 3 \\
 z_3 - z_2 + 4x_{32} &\leq 3
 \end{aligned} \tag{3.10}$$

$$\begin{aligned}
 x_{00} = 0 \text{ lub } 1; x_{01} = 0 \text{ lub } 1; x_{02} = 0 \text{ lub } 1; x_{03} = 0 \text{ lub } 1; \\
 x_{10} = 0 \text{ lub } 1; x_{11} = 0 \text{ lub } 1; x_{12} = 0 \text{ lub } 1; x_{13} = 0 \text{ lub } 1; \\
 x_{20} = 0 \text{ lub } 1; x_{21} = 0 \text{ lub } 1; x_{22} = 0 \text{ lub } 1; x_{23} = 0 \text{ lub } 1; \\
 x_{30} = 0 \text{ lub } 1; x_{31} = 0 \text{ lub } 1; x_{32} = 0 \text{ lub } 1; x_{33} = 0 \text{ lub } 1; \\
 z_1 \geq 0; z_2 \geq 0; z_3 \geq 0;
 \end{aligned} \tag{3.11}$$

Inny opis zagadnienia jednego komiwojażera można sformułować, opierając się na teorii grafów. Dany jest nieskierowany, spójny i pełny graf $G(V, E, c)$, w którym zbiór V stanowi $n + 1$ węzłów odpowiadających odbiorcom oraz dostawcy (rys. 3.3).



Rys. 3.3. Graf pełny dla problemu komiwojażera wraz z przykładową dopuszczalną trasą (cyklem Hamiltona)

Źródło: opracowanie własne.

Zbiór krawędzi E grafu reprezentuje możliwe połączenia pomiędzy punktami i oraz j . Każdej krawędzi grafu łączącej dwa dowolne węzły przyporządkowana jest pewna nieujemna liczba c_{ij} (waga krawędzi) odpowiadająca długości (kosztowi) połączenia pomiędzy punktami i oraz j . Jeden z węzłów grafu reprezentuje

punkt początkowy D , z którego komiwojazer rozpoczyna swoją trasę oraz do niego powraca. Za trasę komiwojazera uważać należy taki zbiór krawędzi, dla którego:

- pierwsza krawędź rozpoczyna się w węźle początkowym 0;
- ostatnia krawędź kończy się w węźle początkowym 0,
- każdy węzeł grafu stanowi połączenie dwóch i tylko dwóch krawędzi grafu.

Dla tak sformułowanego problemu komiwojazera przedstawić można także model programowania całkowitoliczbowego (3.12)–(3.15) (Dantzig i in., 1954):

$$\min. F(\mathbf{x}) = \sum_{i=0}^n \sum_{j=0}^n c_{ij} x_{ij} \quad (3.12)$$

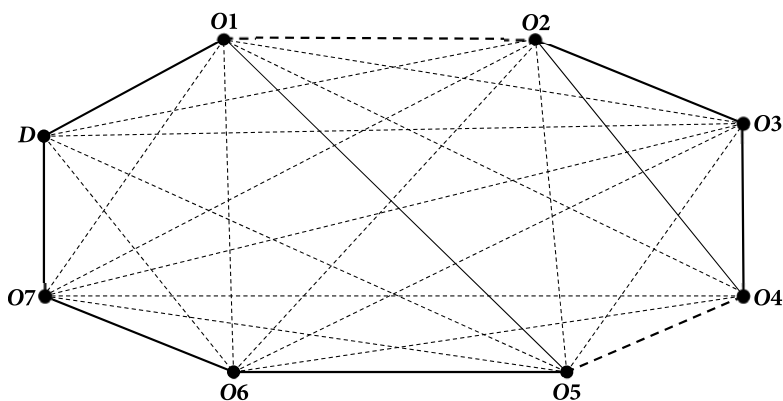
$$\sum_{i=0}^n x_{ij} = 1 \quad \text{dla } j = 0, 1, \dots, n \quad (3.13)$$

$$\sum_{j=0}^n x_{ij} = 1 \quad \text{dla } i = 0, 1, \dots, n \quad (3.14)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1 \quad \text{dla } S \subset V \text{ oraz } 2 \leq |S| \leq n - 1 \quad (3.15)$$

Różnica modelu (3.12)–(3.15) w porównaniu do modelu przedstawionego równaniami (3.2)–(3.5) polega na innym sformułowaniu ograniczeń zapewniających odwiedzenie przez komiwojazera wszystkich punktów (3.15). Tak samo jak ograniczenia (3.5), zabraniają one formowania się niepełnych tras komiwojazera, czyli takich, które nie odwiedzają wszystkich zadanych punktów, a tylko pewien ich podzbiór S . W wypadku gdy pojazd odwiedzi mniej niż n punktów, wtedy w takiej trasie występuje $|S|$ wierzchołków grafu oraz tyle samo krawędzi. Prowadzi to do sytuacji, w której ograniczenie (3.15) nie jest spełnione. Lewa strona tego ograniczenia wynosi $|S|$, natomiast prawa strona tego samego ograniczenia jest równa $|S| - 1$. W związku z tym, że niepełnej trasy komiwojazera nie można otrzymać dla jednego punktu, a także dla n punktów, należy zauważyć, że ograniczenie (3.15) jest istotne dla $|S| \geq 2$ oraz dla $|S| \leq n - 1$.

Przedstawiony powyżej zbiór krawędzi grafu tworzy ciąg połączeń pomiędzy wszystkimi węzłami należącymi do grafu i w teorii grafów nosi on nazwę cyklu Hamiltona. Tym samym można stwierdzić, że trasa komiwojazera jest dopuszczalna, jeżeli stanowi ona cykl Hamiltona (rys. 3.3). Na rysunku 3.3 przedstawiono wszystkie możliwe krawędzie grafu składającego się z ośmiu węzłów (linie ciągłe i przerywane). Natomiast na rysunku 3.4 linią ciągłą zaznaczono niedopuszczalną trasę komiwojazera. Pojazd, rozpoczynając trasę z punktu D , odwiedza kolejno punkty: $O1, O5, O6, O7$ (lub w odwrotnej kolejności: $O7, O6, O5, O1$), a następnie powraca do punktu D . Węzły: $O2, O3, O4$ zostały pominięte, co oznacza, że punkty, które one reprezentują, musiałyby być odwiedzone po ponownym opuszczeniu przez pojazd punktu D . Taka sytuacja jest niedopuszczalna, ponieważ pojazd może tylko jeden raz opuścić punkt startowy oraz jeden raz do niego wrócić. Przedstawione na rysunku 3.4 dwa zbiory krawędzi w teorii grafów noszą nazwę cykli skróconych.



Rys. 3.4. Graf pełny dla problemu komiwojażera wraz z przykładową niedopuszczalną trasą (cyklem skróconym)

Źródło: opracowanie własne.

Rozwiązaniem optymalnym w tak sformułowanym w teorii grafów zagadnieniu jednego komiwojażera będzie cykl Hamiltona, dla którego łączna suma liczb c_{ij} przyporządkowanych do krawędzi wchodzących w skład cyklu będzie najmniejsza.

Zagadnienie komiwojażera jest nie tylko problemem, przed którym stoi wiele przedsiębiorstw prowadzących działalność logistyczną, ale znajduje się ono także w centrum zainteresowania matematyków zajmujących się problemami optymalizacji kombinatorycznej. Sformułowanie problemu jest bardzo proste, lecz znalezienie najlepszego rozwiązania, czy nawet bliskiego optymalnemu, nie jest już tak oczywiste. Liczba możliwych tras (cykli Hamiltona) jest równa liczbie wszystkich permutacji odwiedzanych punktów. Jeżeli stopniowo będzie zwiększana liczba punktów, które ma odwiedzić komiwojażer, liczba rozwiązań dopuszczalnych będzie rosła bardzo szybko. Dla zadania, w którym pojazd ma odwiedzić n punktów oraz macierz odległości C jest symetryczna ($c_{ij} = c_{ji}$), liczba wszystkich możliwych tras wyniesie $n!/2$. W tabeli 3.1 pokazano, w jaki sposób wzrasta liczba możliwych permutacji, gdzie liczba odwiedzanych przez pojazd punktów należy do przedziału $[1, 15]$. Tylko dla $n = 10$ liczba tras, jakie może pokonać komiwojażer, wynosi aż 3 628 800.

Dany problem decyzyjny lub optymalizacyjny możemy rozwiązywać na wiele różnych sposobów, czyli możemy stosować do jego rozwiązania różne algorytmy. Złożoność obliczeniowa tych algorytmów może okazać się w niektórych wypadkach bardzo duża. W teorii złożoności obliczeniowej dokonano podziału problemów ze względu na czas, w jakim mogą zostać rozwiązane przez dany algorytm (Błażewicz, 1986; Papamitriou, 1997; 2002):

- *NP (Nondeterministic Polynomial)* – klasa problemów, które mogą zostać rozwiązane w czasie wielomianowym algorytmem niedeterministycznym;
- *P (Polynomial)* – klasa problemów, które mogą zostać rozwiązane w czasie wielomianowym algorytmem deterministycznym.

Tabela 3.1. Liczba możliwych rozwiązań dla zadania komiwojażera ($c_{ij} = c_{ji}$)

n	Liczba tras
1	1
2	1
3	3
4	12
5	60
6	360
7	2 570
8	20 160
9	181 400
10	1 814 400
11	19 958 400
12	89 500 800
13	3 113 510 400
14	43 589 145 600
15	653 837 184 000

Źródło: opracowanie własne.

Pomiędzy klasą problemów P a klasą problemów NP zachodzi relacja $P \subseteq NP$, natomiast do dziś problemem nierozstrzygniętym jednoznacznie pozostaje, czy zachodzi relacja $P \neq NP$.

Odrębnie badaną klasę złożoności obliczeniowej stanowią problemy NP -zupełne. Dany problem nazwiemy NP -zupełnym, jeżeli należy on do klasy NP oraz każdy inny problem z klasy NP da się sprowadzić do niego w czasie wielomianowym. Własnością problemów tej klasy jest to, że w wypadku znalezienia algorytmu deterministycznego rozwiązującego w czasie wielomianowym jeden z problemów należących do tej klasy znajdujemy także taki algorytm dla pozostałych problemów w tej klasie.

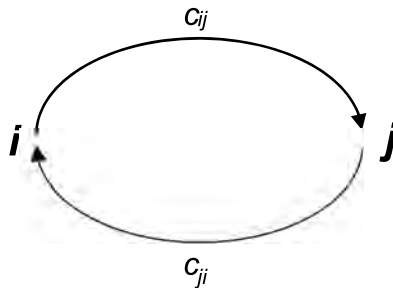
Omawiane w tej części rozdziału zagadnienie jednego komiwojażera należy do klasy zadań NP -zupełnych (Lenstra i Rinnooy Kan, 1981).

3.2. Warianty problemu komiwojażera

Podobnie jak przy formułowaniu klasycznego zadania transportowego, tak i omawiając problem komiwojażera, należy scharakteryzować jego najważniejsze odmiany. Warianty problemu komiwojażera mają swoją genezę zarówno w sposobie definiowania poszczególnych parametrów samego zadania, jak i wynikają z potrzeb lub ograniczeń formułowanych bezpośrednio przez decydentów. Oznacza to, że różne warianty problemu komiwojażera są efektem i rozważań teoretycznych, i doświadczeń zebranych z praktyki gospodarczej.

3.2.1. Symetryczny i niesymetryczny problem komiwojażera

Istotą odmienności symetrycznego i niesymetrycznego problemu komiwojażera jest oczywiście relacja wartości współczynnika kosztu c_{ij} pomiędzy punktami i a j do współczynnika kosztu c_{ji} pomiędzy punktami j a i (rys. 3.5).



Rys. 3.5. Połączenie pomiędzy punktami i a j
Źródło: opracowanie własne.

W symetrycznym problemie komiwojażera zachodzi równość: $c_{ij} = c_{ji}$, która z kolei nie musi być prawdziwa w problemie niesymetrycznym ($c_{ij} \neq c_{ji}$). Zmiana ta powoduje, że liczba wszystkich możliwych tras wzrasta dwukrotnie w stosunku do wartości pokazanych w tabeli 3.1. Jest to często występująca sytuacja nie tylko o charakterze teoretycznym, lecz również praktycznym, gdyż nierzadko pojawia się np. w logistyce dystrybucji w aglomeracjach miejskich, w których ze względu na dużą liczbę dróg jednokierunkowych połączenie z punktu A do punktu B nie jest tożsame z trasą z punktu B do punktu A .

W teorii grafów asymetryczny problem komiwojażera należy sformułować w następujący sposób: dany jest graf skierowany (digraf) $D(W, A, d)$, w którym W stanowi zbiór wierzchołków z wyodrębnionym wierzchołkiem początkowym (odpowiadającym dostawcy) $W = \{0, 1, 2, \dots, n\}$, natomiast A jest zbiorem

krawędzi skierowanych (w szczególności zbiorem uporządkowanych par wierzchołków): $A \subseteq W \times W$. Symbolem d_{ij} oznaczony jest koszt połączenia wierzchołków i oraz j , w którym początek stanowi węzeł i , natomiast koniec wierzchołek j .

Na podstawie tak zdefiniowanego grafu skierowanego D można zbudować graf nieskierowany $G(V, E, c)$. W grafie tym zbiór wierzchołków V jest dwukrotnie liczniejszy:

$$V = W \cup \{n + 1, n + 2, \dots, 2n\} \quad (3.16)$$

Z kolei zbiór krawędzi jest następujący – E jest sumą dwóch zbiorów krawędzi:

$$E = \{(i, n + i) \mid i = 1, 2, \dots, n\} \text{ oraz } \{n + i, j\} \mid (i, j) \in A\} \quad (3.17)$$

Koszty połączenia c_{ij} (wagi grafu) wynoszą odpowiednio:

$$C_{i, n+1} = -M \text{ dla } i = 1, 2, \dots, n \quad (3.18)$$

$$C_{n+1, j} = d_{ij} \text{ dla } (i, j) \in A \quad (3.19)$$

Wielkość M oznacza odpowiednio dużą liczbę (np. $M \rightarrow \infty$), choć może być to także wartość równa np. $\sum_{(i, j) \in A} d_{ij}$.

Należy tutaj zauważyć, że dla wcześniej sformułowanego skierowanego grafu D dla każdej trasy komiwojażera będącej cyklem Hamiltona o danym koszcie D_D istnieje cykl Hamiltona określony w grafie G o długości równej C_G wyrażonej wzorem (3.20):

$$C_G = D_G - nM \text{ dla } (i, j) \in A \quad (3.20)$$

Ponadto wszystkie krawędzie grafu G o koszcie równym $-M$ zawierają się w optymalnym cyklu Hamiltona.

3.2.2. Problem komiwojażera w grafie niepełnym

Problem komiwojażera sformułowany został na podstawie definicji grafu pełnego $G(V, E, c)$, czyli takiego, w którym występują bezpośrednie połączenia pomiędzy wszystkimi jego wierzchołkami. Jednakże bardzo często takie bezpośrednie trasy nie są możliwe, a tym samym graf może nie zawierać pewnej liczby krawędzi. Wynikać to może z tego, że np. nie istnieje droga (np. linia kolejowa) łącząca dwa punkty A i B , która nie przechodzi pośrednio przez inny punkt C . Taki graf nosi nazwę grafu niepełnego.

W sytuacji przedstawionej powyżej graf niepełny można przekształcić do postaci grafu pełnego przez dodanie brakujących krawędzi oraz określenie ich kosztu

jako wartości M , gdzie M jest odpowiednio dużą liczbą ($M > \sum_{(i,j) \in E} c_{ij}$). Jeżeli istnieje optymalny cykl Hamiltona dla przekształconego grafu, w skład którego nie wchodzi dodane krawędzie z kosztem równym M , wtedy taki cykl Hamiltona jest jednocześnie optymalnym cyklem Hamiltona dla niepełnego grafu wyjściowego. Natomiast jeżeli w przekształconym grafie optymalny cykl Hamiltona zawiera przynajmniej jedną dodaną krawędź z kosztem równym M , oznacza to, że niepełny graf wyjściowy nie zawiera ani jednej trasy komiwojażera będącej cyklem Hamiltona.

Inne podejście do sformułowania problemu komiwojażera w grafie niepełnym to sytuacja często występująca w tzw. problemie komiwojażera w układzie planarnym (graficzny problem komiwojażera). Przykładem takiego grafu może być sieć kolejowa na zadanym obszarze, gdzie węzłami grafu są węzły kolejowe (miejsca krzyżowania się kolejowych szlaków komunikacyjnych). Graf jest kompletny, co oznacza, że możliwe jest przemieszczenie się z dowolnego węzła kolejowego do każdego innego, choć nie zawsze musi to nastąpić z pominięciem pozostałych węzłów.

Wtedy celem jest znalezienie takiej trasy komiwojażera, co do której zachodzi warunek, że każdy węzeł musi zostać przez niego odwiedzony, a przy tym koszt całej trasy musi być jak najmniejszy. W grafie niepełnym może nie istnieć cykl Hamiltona. Co więcej, nawet jeżeli istnieje bezpośrednie połączenie pomiędzy punktami A i B , to i tak w rzeczywistości może być ono mniej korzystne niż połączenie pośrednie przez punkt C : $A \rightarrow C \rightarrow B$. Tym samym należy odstąpić od ograniczenia mówiącego o tym, że każdy węzeł (punkt w trasie komiwojażera) może być odwiedzony tylko jeden raz, aczkolwiek musi być odwiedzony co najmniej jeden raz. To samo dotyczy krawędzi grafów, które w takiej sytuacji mogą być wykorzystane wielokrotnie.

Niepełny graf wyjściowy dla problemu komiwojażera w układzie planarnym może zostać przekształcony w graf pełny przez dodanie brakujących krawędzi. Koszt dodanej krawędzi pomiędzy wierzchołkami i oraz j reprezentuje koszt najkrótszej możliwej drogi pomiędzy nimi, która będzie przechodzić przez przynajmniej jeden inny wierzchołek grafu wyjściowego. W tak przekształconym grafie pełnym istnieje optymalny cykl Hamiltona, który zawierać może dodane krawędzie. Jeżeli istnieje optymalny cykl Hamiltona dla przekształconego grafu, w skład którego nie wchodzi dodane krawędzie, wtedy taki cykl Hamiltona jest jednocześnie optymalnym cyklem Hamiltona dla niepełnego grafu wyjściowego. W przeciwnej sytuacji dodane krawędzie grafu wchodzące w skład optymalnego cyklu Hamiltona należy przekształcić do postaci najkrótszej drogi łączącej wierzchołki, których ta krawędź dotyczy. Po przekształceniach uzyskana zostanie trasa komiwojażera dla grafu wyjściowego będąca odpowiednikiem optymalnego cyklu Hamiltona dla grafu pełnego.

3.2.3. Problem komiwojażera z minimalizacją czasu

Dotychczas formułowana funkcja celu zakładała minimalizację całkowitej długości trasy komiwojażera, która była sumą wszystkich odcinków pomiędzy punktami, jakie wchodziły w skład rozpatrywanej trasy będącej cyklem Hamiltona. Funkcja celu może także reprezentować koszt przejazdu, jak również całkowity czas przejazdu. Wymaga to oczywiście uprzedniej znajomości macierzy, odpowiednio, kosztów lub czasów przejazdu pomiędzy wszystkimi punktami, w tym punktem startowym.

Interesującym wariantem klasycznego problemu komiwojażera jest zagadnienie, w którym celem nie jest znalezienie trasy minimalizującej całkowity czas jej pokonania³. Rozpatruje się trasy będące cyklem Hamiltona z punktu widzenia czasu przejazdu pomiędzy dwoma punktami i a punktem j . Poszukiwana jest taka trasa, w której najdłuższe połączenie punktów i oraz j będzie jak najmniejsze.

Formalną definicję powyższej modyfikacji klasycznego problemu komiwojażera przedstawić należy w następujący sposób: dany jest pełny i nieskierowany graf $G(V, E, c)$, gdzie c_e jest wagą krawędzi $e \in E$. Celem jest znalezienie takiego cyklu Hamiltona H^* grafu G , dla którego zachodzi:

$$\wedge_{H \in G} \max. \{c_e : e \in H^*\} \leq \max. \{c_e : e \in H\} \quad (3.21)$$

Rozpatrując natomiast postać modelu komiwojażera wyrażonego zależnościami (3.12)–(3.15), funkcja celu może przyjąć następującą postać:

$$\min. F(\mathbf{x}) = \max_{(i,j):x_{ij}=1} \{c_{ij}\} \quad (3.22)$$

Pewną odmianą omawianej tu modyfikacji klasycznego problemu komiwojażera jest zagadnienie, w którym poszukiwanie rozwiązania optymalnego oparte jest na przeciwnym kierunku optymalizacji funkcji celu (3.21) i (3.22):

$$\max. F(\mathbf{x}) = \min_{(i,j):x_{ij}=1} \{c_{ij}\} \quad (3.23)$$

Jednakże każde zagadnienie MSTSP (*Maximum Scatter Traveling Salesman Problem*) może zostać przekształcone w zadanie z funkcją celu (3.22) w wyniku przekształcenia macierzy wag c_e :

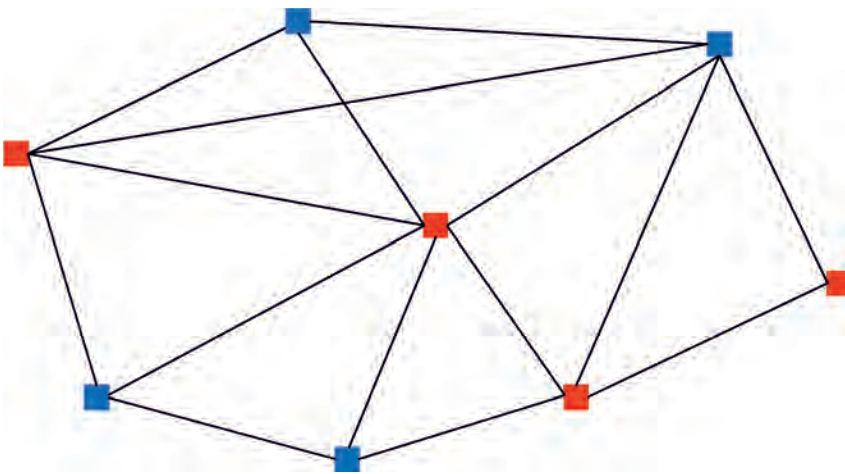
$$\tilde{c}_e = \max_{e \in E} c_e - c_e \quad (3.24)$$

³ W literaturze angielskiej zagadnienie to powszechnie znane jest pod nazwą *Bottleneck Travelling Salesman Problem*.

3.2.4. Problem chińskiego listonosza

W klasycznym problemie komiwojażera podstawowym założeniem jest odwiedzenie przez pojazd każdego punktu dokładnie jeden raz. W problemie chińskiego listonosza – *Chinese Postman Problem* – założenie to dotyczy krawędzi grafu. Przede wszystkim należy zaznaczyć, że graf musi być spójny. Ponadto każda jego krawędź (połączenie pomiędzy punktami i oraz j) musi znaleźć się w rozpatrywanej trasie co najmniej jeden raz. Celem jest znalezienie najkrótszej trasy uwzględniającej wszystkie krawędzie grafu.

Po raz pierwszy zagadnienie chińskiego listonosza przedstawione zostało na początku lat sześćdziesiątych XX wieku. Sformułowanie problemu wymaga wprowadzenia kilku niezbędnych definicji. Pierwszą z nich jest pojęcie wierzchołka stopnia parzystego i nieparzystego. Stopień ten wyznaczony jest przez liczbę krawędzi skojarzonych z danym wierzchołkiem grafu. Jeżeli liczba ta jest parzysta, wtedy wierzchołek jest stopnia parzystego, w przeciwnym razie jest on stopnia nieparzystego (rys. 3.6).



Rys. 3.6. Wierzchołki stopnia parzystego (czerwony kolor) i nieparzystego (niebieski kolor)

Źródło: opracowanie własne.

Jednym z najbardziej znanych problemów w teorii grafów, a jednocześnie rodzajów grafów jest tzw. graf Eulera. Graf Eulera to taki graf, w którym istnieje cykl zamknięty przechodzący przez wszystkie krawędzie grafu dokładnie jeden raz (cykl Eulera). Innymi słowy, można stwierdzić, że jeżeli istnieje trasa listonosza przechodząca przez wszystkie krawędzie grafu jeden i tylko jeden raz, to jest to trasa optymalna. W grafie nieskierowanym istnienie cyklu Eulera warunkowane jest tym, że wszystkie wierzchołki grafu muszą być stopnia parzystego. Przykład grafu, w którym nie istnieje cykl Eulera, został przedstawiony na rysunku 3.6.

Z kolei w grafie skierowanym warunek wystarczalności istnienia cyklu Eulera przedstawiony został przez L.R. Forda i D.R. Fulkersona (1962). Mówi on, że w takim wypadku liczba krawędzi grafu wchodzących do wierzchołka (skierowanych ku danemu wierzchołkowi) musi być równa liczbie krawędzi wychodzących z tego wierzchołka.

Wreszcie dla grafów mieszanych⁴ warunek wystarczający do istnienia cyklu Eulera jest trochę bardziej skomplikowany. Otóż mówi on, że jest on spełniony, jeżeli każdy wierzchołek grafu G jest skojarzony z parzystą liczbą krawędzi skierowanych i nieskierowanych, a także gdy dla każdego podzbioru wierzchołków S należącego do grafu $G(S \subset V)$ bezwzględna różnica pomiędzy:

- liczbą krawędzi wychodzących z wierzchołków należących do S i wchodzących do wierzchołków nienależących do $S(V \setminus S)$
- a liczbą krawędzi wychodzących z wierzchołków nienależących do $S(V \setminus S)$ i wchodzących do wierzchołków należących do S

jest większa lub równa liczbie nieskierowanych krawędzi grafu łączących wierzchołki należące do zbioru S z wierzchołkami do niego nienależącymi ($V \setminus S$).

Jeżeli graf nie jest grafem Eulera, zadanie problemu listonosza polega na przejściu przez wszystkie krawędzie grafu w taki sposób, że niektóre z nich pokonane zostaną więcej niż jeden raz. Tym samym problem ten sprowadzić można do zagadnienia wyznaczenia powtórnych przejść przez te krawędzie, przy czym całkowity koszt powtórnych przejść przez te krawędzie będzie jak najmniejszy.

Problem listonosza może być sformułowany z wykorzystaniem programowania liniowego. Aby tego dokonać, należy uprzednio przyjąć następujące założenia:

- 1) dla grafu nieskierowanego, który nie jest grafem Eulera, należy w pierwszej kolejności zidentyfikować wszystkie wierzchołki stopnia nieparzystego (ich liczba zawsze będzie parzysta), a następnie należy określić koszt przypisany do krawędzi grafu łączącej każdą parę takich wierzchołków \tilde{c}_{ij} będący kosztem najkrótszej drogi w grafie pomiędzy wierzchołkami i oraz j ;
- 2) przez zmienną decyzyjną x_{ij} oznaczyć należy liczbę powtórzonych przejść przez krawędź pomiędzy wierzchołkami i a j (liczba kopii krawędzi dodanych do grafu G , aby ten stał się grafem Eulera), gdzie $i < j$ oraz $x_{ij} \in \mathbb{C}$;
- 3) należy zdefiniować zbiór $\delta(S)$ wszystkich krawędzi (i, j) należących do grafu G , takich że jeden koniec krawędzi stanowi wierzchołek stopnia nieparzystego, a drugi stopnia parzystego.

Model programowania liniowego dla problemu listonosza w grafie nieskierowanym przedstawić można w następujący sposób (Edmonds i Johnson, 1973):

$$\min. \tilde{C}(\mathbf{x}) = \sum_{(i,j) \in E} \tilde{c}_{ij} x_{ij} \quad (3.25)$$

4 Graf mieszany to taki, który zawiera zarówno krawędzie nieskierowane, jak i krawędzie określające kierunek przemieszczania się pomiędzy wierzchołkami.

$$\sum_{(i,j) \in \delta(S)} x_{ij} \geq 1 \quad (3.26)$$

$$x_{ij} \geq 0 \quad \text{dla } (i, j) \in E \quad (3.27)$$

Nieco inaczej prezentuje się sformułowanie problemu listonosza dla grafu skierowanego (Beltrami i Bodin, 1974). Można w nim odnaleźć pewne analogie do jednej z prezentowanych wcześniej modyfikacji klasycznego zadania transportowego – problemu minimalizacji pustych przebiegów. Tak jak w problemie listonosza dla grafu nieskierowanego, tak i dla grafu skierowanego należy uprzednio przedstawić następujące założenia:

- 1) zmienna decyzyjna x_{ij} oznacza liczbę powtórzonych przejść przez krawędź pomiędzy wierzchołkami i a j (liczba kopii krawędzi dodanych do grafu G , aby ten stał się grafem Eulera);
- 2) należy wyznaczyć zbiór wierzchołków A , dla których liczba krawędzi mających w nim swój koniec jest większa od liczby krawędzi mających w nim swój początek, a następnie określić tę różnicę (a_i);
- 3) należy wyznaczyć zbiór wierzchołków B , dla których liczba krawędzi mających w nim swój początek jest większa od liczby krawędzi mających w nim swój koniec, a następnie określić tę różnicę (b_j);
- 4) należy określić koszt połączenia \tilde{c}_{ij} pomiędzy każdą parą wierzchołków, z których jeden należy do zbioru A , a drugi do zbioru B , jako koszt najkrótszej drogi w grafie pomiędzy wierzchołkami.

Model programowania liniowego dla problemu listonosza w grafie skierowanym przedstawić można z wykorzystaniem klasycznego problemu transportowego, w którym wielkość a_i traktowana jest jako podaż, natomiast b_j jako popyt:

$$\min. \tilde{C}(\mathbf{x}) = \sum_{i \in A} \sum_{j \in B} \tilde{c}_{ij} x_{ij} \quad (3.28)$$

$$\sum_{i \in A} x_{ij} = a_i \quad \text{dla każdego } j \in B \quad (3.29)$$

$$\sum_{j \in B} x_{ij} = b_j \quad \text{dla każdego } i \in A \quad (3.30)$$

$$x_{ij} \geq 0 \quad \text{dla } i \in A \text{ oraz } j \in B \quad (3.31)$$

Dopełniając prezentacji modeli problemu listonosza dla wszystkich rodzajów grafów, należy także przedstawić model dla grafu mieszanego G , który zawiera zarówno zbiór krawędzi nieskierowanych E , jak i zbiór krawędzi skierowanych F . Przyjmując następujące założenia:

- 1) zmienna decyzyjna x_{ij} oznacza liczbę powtórzonych przejść przez krawędź skierowaną pomiędzy wierzchołkami i a j (liczba kopii krawędzi skierowanych dodanych do grafu G , aby ten stał się grafem Eulera);

- 2) zmienna decyzyjna y_{ij} oznacza liczbę powtórzonych przejść przez krawędź nieskierowaną pomiędzy wierzchołkami i a j (liczba kopii krawędzi nieskierowanych dodanych do grafu G , aby ten stał się grafem Eulera);
- 3) dla każdego podzbioru wierzchołków S należącego do grafu $G(S \subset V)$ należy określić:
- zbiór krawędzi skierowanych $F^+(S)$, których początek jest w wierzchołkach należących do zbioru S , natomiast koniec stanowi wierzchołek nienależący do tego zbioru ($V \setminus S$);
 - zbiór krawędzi skierowanych $F^-(S)$, których początek jest w wierzchołkach nienależących do zbioru S , natomiast koniec stanowi wierzchołek należący do tego zbioru;
 - zbiór krawędzi nieskierowanych $\delta(S)$, których jednym z końców jest wierzchołek nienależący do zbioru S , natomiast drugi koniec stanowi wierzchołek należący do tego zbioru;
 - różnicę $u(S)$ pomiędzy sumą liczby krawędzi skierowanych mających początek w S i koniec w $V \setminus S$ oraz liczby krawędzi skierowanych w przeciwnym kierunku (początek w $V \setminus S$ i koniec w S) a liczbą krawędzi nieskierowanych mających jeden koniec w S i drugi koniec w $V \setminus S$:

$$u(S) = |F^+(S)| - |F^-(S)| - |\delta(S)| \quad (3.32)$$

- 4) p_k jest stałą przyjmującą wartości binarne, gdzie 1 oznacza, że wierzchołek k jest stopnia nieparzystego, 0 – że wierzchołek k jest stopnia parzystego, natomiast z_k jest zmienną przyjmującą wartości całkowitoliczbowe.

Model programowania liniowego dla problemu listonosza w grafie mieszanym może przyjąć następującą postać (Norbert i Picard, 1996):

$$\min. C(\mathbf{x}) = \sum_{(i,j) \in F} c_{ij} x_{ij} + \sum_{(i,j) \in E} c_{ij} y_{ij} \quad (3.33)$$

$$\sum_{(i,j) \in F} x_{ij} + \sum_{(i,j) \in E} y_{ij} = 2z_k + p_k \quad \text{dla } k \in S \quad (3.34)$$

$$\sum_{(i,j) \in F^-(S)} x_{ij} - \sum_{(i,j) \in F^+(S)} x_{ij} + \sum_{(i,j) \in \delta(S)} y_{ij} \geq u(S) \quad \text{dla } S \subset V \quad (3.35)$$

$$x_{ij} \geq 0 \quad \text{dla } (i,j) \in E \cup F \quad (3.36)$$

$$y_{ij} \geq 0 \quad \text{dla } (i,j) \in E \cup F \quad (3.37)$$

$$z_k \geq 0 \quad \text{dla } k \in V \quad (3.38)$$

3.2.5. Problem m -komiwojażerów

Osobnym przypadkiem problemu komiwojażera jest zagadnienie, w którym odwiedzanie zadanych punktów (klientów) może być realizowane nie przez jeden, lecz przez np. kilka pojazdów. W literaturze przedmiotu taki problem decyzyjny określany jest już często mianem zagadnienia wielu komiwojażerów lub układania tras pojazdów, o którym traktuje kolejna część niniejszej książki. Jednakże w pewnej sytuacji uznać go można za modyfikację problemu jednego komiwojażera.

W omawianej modyfikacji problemu komiwojażera z punktu D wyrusza nie jeden, a m komiwojażerów, którzy mają za zadanie odwiedzić n innych punktów: O_1, O_2, \dots, O_n , a następnie powrócić do punktu początkowego. Celem jest ustalenie takiego porządku odwiedzanych odbiorców, aby całkowita długość (koszt) trasy pojazdu wszystkich komiwojażerów razem była jak najmniejsza.

Problem m -komiwojażerów może zostać przekształcony do problemu jednego komiwojażera przez modyfikację macierzy odległości pomiędzy odwiedzanymi punktami. Zmiana polega na dodaniu do macierzy odległości $m - 1$ kopii pierwszego wiersza (reprezentującego wektor odległości pomiędzy punktem D a odwiedzanymi odbiorcami O_j) oraz kopii pierwszej kolumny (reprezentującej wektor odległości pomiędzy odwiedzanymi odbiorcami O_j a dostawcą D) (rys. 3.7).

	$m - 1$	$n + 1$
$m - 1$	$c_{kl} = \infty$ $k = 1, \dots, m - 1$ $l = 1, \dots, m - 1$	$c_{kl} = d_{0, l - M}$ $k = 1, \dots, m - 1$ $l = m, \dots, m + m$
$n + 1$	$c_{kl} = d_{k - m, 0}$ $k = m, \dots, m + n$ $l = 1, \dots, m - 1$	$c_{kk} = \infty$ $c_{kl} = d_{k - m, l - m}$ $k = m, \dots, m + n$ $l = m, \dots, m + n$

Rys. 3.7. Postać nowej macierzy odległości C' dla zadania m -komiwojażerów
Źródło: (Całczyński, 1992).

Nowa macierz C' składająca się z elementów c_{kl} powstała w ten sposób, że kopie pierwszego wiersza wyjściowej macierzy odległości C dodane zostały od góry, natomiast kopie pierwszej kolumny dodane zostały z lewej strony. Nowa macierz C' jest macierzą kwadratową o rozmiarach $(m + n) \times (m + n)$.

Zgodnie z tym sformułować można problem komiwojażera w następujący sposób (3.39)–(3.42):

$$\min. F(\mathbf{x}) = \sum_{k=1}^{m+n} \sum_{l=1}^{m+n} c_{kl} x_{kl} \quad (3.39)$$

$$\sum_{k=1}^{m+n} x_{kl} = 1 \quad \text{dla } l = 1, 2, \dots, m+n \quad (3.40)$$

$$\sum_{l=1}^{m+n} x_{kl} = 1 \quad \text{dla } k = 1, 2, \dots, m+n \quad (3.41)$$

$$z_k - z_l + \frac{(n+1)}{m} x_{kl} \leq \frac{(n+1)}{m} + m - 1 \quad \text{dla } k, l = m, \dots, m+n+1 \text{ oraz } k \neq l \quad (3.42)$$

$$x_{kl} = \begin{cases} 0 \\ 1 \end{cases} \quad \text{dla } k, l = 0, 1, \dots, n \quad (3.43)$$

Dla $m = 1$ otrzymany zostanie model jednego komiwojażera, który sformułowany został wzorami (3.1)–(3.5).

Rozdział 4

Układanie tras wielu pojazdów

Przedstawione dotychczas modele problemu jednego komiwojażera oraz jego modyfikacje ukazują podstawowe kwestie, jakie trzeba brać pod uwagę przy układaniu tras dla pojazdów, i eksponują ich kombinatoryczny charakter. Praktyczne znaczenie tych modeli może się wydawać ograniczone, ponieważ w rzeczywistości operatorzy logistyczni muszą niejednokrotnie uwzględniać także wiele innych dodatkowych warunków, które wynikają ze skomplikowanej specyfiki działalności firm. Warunki te dotyczyć mogą chociażby ilości transportowanego dobra czy czasu pracy kierowców. W związku z tym, na początku tylko ogólnie formułując zagadnienie układania tras pojazdów, należy jednocześnie wprowadzić kilka pojęć, które będą niezbędne dla zrozumienia dalszej części książki.

Przedsiębiorstwo lub operator logistyczny działający na jego rzecz może zarówno dystrybuować do sieci odbiorców (klientów) pewne dobro, jak i odbierać je od sieci dostawców. Pierwszy przypadek przedstawia sytuację charakterystyczną dla logistyki dystrybucji, w której rolę przedsiębiorstwa może odgrywać zakład produkcyjny lub centrum dystrybucyjne dostarczające do swoich klientów produkt finalny. W literaturze przedmiotu spotkać można różne określenia tego typu zadań, jak np. zadania dostaw lub zadania rozwózkowe. Drugi przypadek przewiduje, że przedsiębiorstwo odbiera dane dobro od swoich klientów i wtedy mówimy o zadaniach odbioru lub zwózki towarów. Przykładem takiej relacji może być sytuacja, w której przedsiębiorstwo jest zakładem przetwórczym odbierającym od dostawców surowiec niezbędny do produkcji gotowych do spożycia wyrobów. Są to także zadania bardzo często charakterystyczne dla tzw. logistyki zwrotnej, w której w centrum zainteresowania są przepływy fizyczne produktów po dokonaniu ich sprzedaży i dostawie do finalnego odbiorcy (konsumenta).

Celem rozróżnienia obu tych pojęć jest zwrócenie uwagi na to, że decydent może mieć do czynienia z transportem polegającym zarówno na dostawie, jak i na odbiorze dóbr. Zadania dostawy i zadania odbioru nie zawsze występują rozdzielnie. W działalności przedsiębiorstwa może być wpisany dwukierunkowy przepływ

dóbr pomiędzy nim a klientami. Przedsiębiorstwo, dostarczając do klienta dane dobro, może jednocześnie odbierać od niego inne. Stąd też niektóre zagadnienia z zakresu układania tras pojazdów noszą również nazwę problemów rozwózkowo-zwózkowych¹.

Liczba klientów, z którymi związane jest przedsiębiorstwo, może być bardzo duża. Jeżeli nie jest istotne, w jakim kierunku odbywa się transport dóbr, klientów przedsiębiorstwa można utożsamiać w sposób ogólny z punktami obsługi odwiedzanymi przez pojazd. Pojęcie „punkt obsługi” jest szczególnie adekwatne w sytuacji dostawy i jednoczesnego odbioru dóbr lub wtedy, gdy klient przedsiębiorstwa identyfikowany jest z geograficznie (lub przestrzennie) zlokalizowanym miejscem nadania i odbioru dobra.

W podstawowych sformułowaniach problemu jednego komiwojażera oraz jego modyfikacji często zakłada się, że komiwojażer wyrusza z punktu startowego oznaczonego numerem 0. Jeśli chodzi o zadania układania tras pojazdów, tak w literaturze przedmiotu, jak i w praktyce gospodarczej również często spotyka się w tym kontekście pojęcie bazy, które może mieć szerokie znaczenie. Określa się tak miejsce (punkt), z którego są dystrybuowane dobra lub do którego są one zwożone. Tak rozumiane pojęcie bazy nie musi zawsze oznaczać bazy transportowej. Przedsiębiorstwa nierzadko korzystają z usług transportu obcego (outsourcing usług transportowych lub spedycyjnych), którego bazy transportowe nie muszą być wcale zlokalizowane w bezpośrednim sąsiedztwie siedziby zleceniodawcy. Poza tym niekoniecznie do danej firmy będzie przypisany tylko jeden geograficznie zlokalizowany punkt, z którego następuje wysyłanie dóbr lub do którego są one zwożone. Szczególnie gdy ma się do czynienia z dużym obszarem geograficznym, przedsiębiorstwo może posiadać oddziały lub filie działające w równorzędnym zakresie. Ich funkcjonowanie może mieć na celu usprawnienie organizacji transportu pomiędzy przedsiębiorstwem a klientami zarówno pod względem czasochłonności, jak i ponoszonych kosztów.

Należy także rozróżnić pojęcia trasy i kursu. Pod terminem „trasa” kryje się marszruta pojazdu, który wyjeżdża z bazy, a następnie odwiedza pewną liczbę punktów obsługi jeden raz, przy czym w bazie pojazd może pojawić się wielokrotnie. Natomiast „kurs” oznacza drogę pojazdu wiodącą przez pewną liczbę punktów obsługi, z których każdy także odwiedzany jest wyłącznie jeden raz. Pojazd tylko jeden raz wyrusza z bazy i do niej powraca. Oznacza to, że na trasie może odbywać się więcej niż jeden kurs.

Przedsiębiorstwo, mając do dyspozycji odpowiednie zaplecze w postaci środków transportu (własnych lub obcych), ma za zadanie zorganizować system obsługi transportowej wszystkich powiązanych z nim punktów obsługi w taki sposób, aby dostarczyć (i/lub odebrać) żadaną ilość dobra, a jednocześnie ponieść jak

1 W literaturze obcej powszechnie występującym określeniem obejmującym problemy układania tras pojazdów jest *Vehicle Routing Problem*.

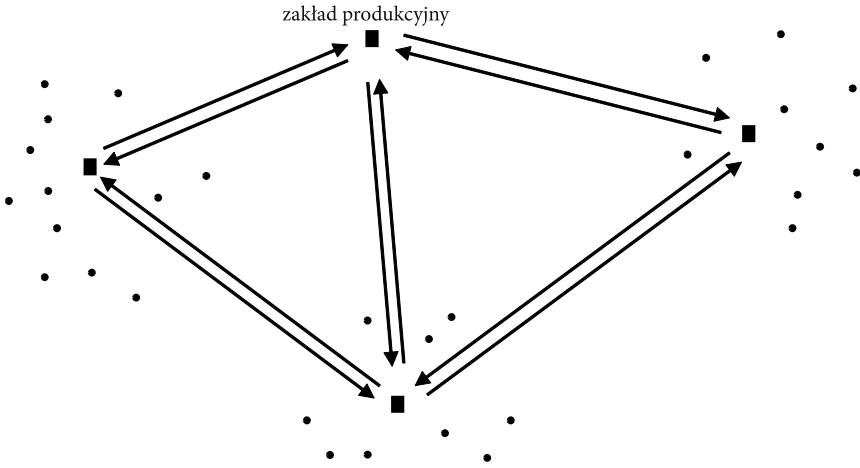
najniższe koszty związane z jego funkcjonowaniem. Tak przedstawiony problem układania tras dla pojazdów stanowi odrębny zbiór zagadnień transportowych badań operacyjnych, choć mogą się nam nasunąć pewne analogie do powszechnie znanych i prostszych w rozwiązywaniu klasycznych zadań transportowych. Jednak nie można ich ze sobą utożsamiać z dwóch powodów.

Po pierwsze, modele klasycznych problemów transportowych nie odzwierciedlają pewnych istotnych cech charakterystycznych tylko dla zagadnień wyznaczania tras pojazdów. Należą do nich między innymi: czas pracy kierowców, terminy jednorazowych partii dostawy, ograniczenia wynikające z posiadanego taboru pojazdów. Po drugie, istotną różnicą między klasycznym zadaniem transportowym a zadaniem ustalania tras dla pojazdów jest sposób dostarczania towarów od nadawcy do odbiorcy. W zadaniach klasycznych mamy do czynienia z transportem, gdzie nadawca, wysyłając towar do odbiorcy, przeznacza dla niego całkowitą ładowność pojazdu lub jego wielokrotność (wykorzystując kilka pojazdów jednocześnie). W zadaniach układania tras dla pojazdów ładowność pojazdu przeznaczona może być dla więcej niż jednego punktu obsługi. Nie oznacza to wszakże, że klasyczne zagadnienie transportowe jest tu zupełnie bezużyteczne. Może ono stanowić bardzo istotne narzędzie wykorzystywane przy rozwiązywaniu złożonych problemów, jakimi są problemy układania tras pojazdów.

4.1. Sformułowanie problemu decyzyjnego

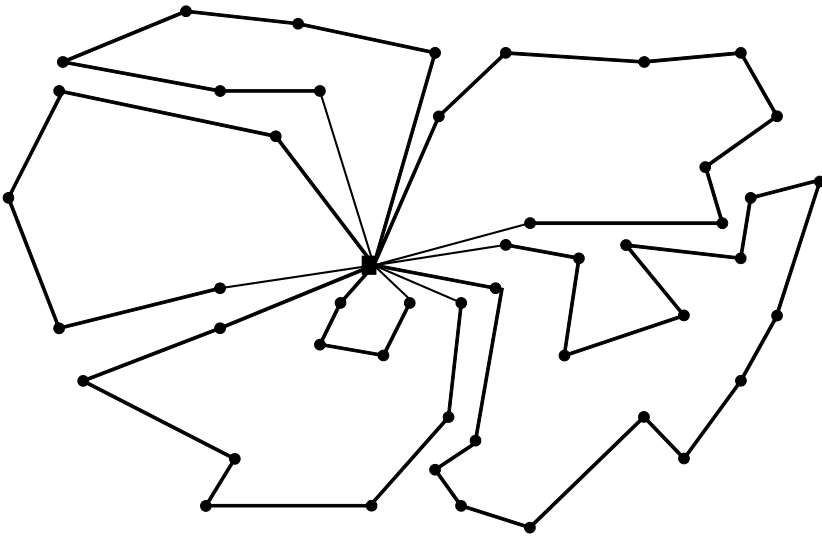
W podstawowej postaci problemu układania tras pojazdów przyjmuje się założenie, że wszystkie punkty obsługi znajdujące się w zasięgu działania bazy są jej jednoznacznie przyporządkowane. Interpretować to należy w ten sposób, że dostawy (i/lub odbiory) produktów realizowane są tylko i wyłącznie przez jednego dostawcę (odbiorcę). Jest to sytuacja charakterystyczna dla logistyki dystrybucji. Budowa centrów dystrybucyjnych wiąże między sobą oraz z głównym dostawcą sieć klientów. Pozwala to usprawnić przepływ zarówno fizyczny, jak i informacyjny w ramach łańcucha dostaw. Pomiędzy samymi centrami dystrybucyjnymi lub między nimi a zakładami produkcyjnymi odbywają się tzw. transporty ładunków skonsolidowanych, które pozwalają wykorzystać inne, tańsze formy transportu (np. transport kolejowy).

Zagadnienie układania tras pojazdów w swojej podstawowej postaci znane jest w literaturze przedmiotu jako problem wielu komiwojażerów. Podobnie jak w problemie jednego komiwojażera, samo sformułowanie problemu decyzyjnego jest trywialne i może być przedstawione w następujący sposób (rys. 4.2).



Rys. 4.1. Jednoznaczne przyporządkowanie punktów obsługi do bazy

Źródło: opracowanie własne.



Rys. 4.2. Graficzna ilustracja problemu wielu komiwojażerów

Źródło: opracowanie własne.

Celem bazy jest obsłużenie zlokalizowanych wokół niej punktów obsługi za pomocą pozostających w jej dyspozycji środków transportowych (pojazdów). W klasycznej postaci problemu wielu komiwojażerów każdy punkt może zostać odwiedzony tylko jeden raz, a także tylko przez jeden pojazd. Pojazd, wyruszając z bazy, ma za zadanie odwiedzić r spośród n wszystkich punktów obsługi: O_1, O_2, \dots, O_r , a następnie powrócić do bazy. Celem jest znalezienie takiego układu tras obsługi punktów przez pojazdy, jakimi dysponuje baza, aby odpowiadając na dwa pytania:

- 1) który punkt obsługi ma być odwiedzany przez dany pojazd określonego typu?
- 2) w jakiej kolejności dany pojazd ma obsłużyć punkty obsługi, które zostały mu jednoznacznie przydzielone?

znaleźć rozwiązanie optymalne reprezentujące taki zbiór tras, dla którego całkowity koszt ponoszony przez wszystkie pojazdy byłby jak najniższy.

W przeciwieństwie do problemu jednego komiwojażera tutaj występują dwa problemy optymalizacyjne:

- 1) problem optymalnego przydziału punktów obsługi do pojazdów, którymi dysponuje baza;
- 2) problem ustalenia optymalnej kolejności odwiedzania poszczególnych punktów obsługi przyporządkowanych do danego pojazdu.

Aby przejść do sformułowania problemu wielu komiwojażerów w języku programowania matematycznego, należy przyjąć następujące oznaczenia:

- n – liczba obsługiwanych punktów znajdujących się w rejonie obsługi bazy;
- K – liczba pojazdów, którymi dysponuje baza;
- c_{ij} – koszt transportu pomiędzy punktami i oraz j , a także pomiędzy i -tym punktem obsługi a bazą;
- x_{ij}^k – zmienna decyzyjna przyjmująca wartość 0, jeżeli pojazd k ma w swojej trasie połączenie pomiędzy punktami i oraz j , lub 1, gdy takiego połączenia nie ma, przy czym $i \neq j$;
- z_i, z_j – zmienne rzeczywiste, przy czym $i \neq j$.

Model dla problemu wielu komiwojażerów w swojej podstawowej postaci może zostać sformułowany w następujący sposób (4.1)–(4.6):

$$\min. F(\mathbf{x}) = \sum_{i=0}^n \sum_{j=0}^n \sum_{k=1}^K c_{ij} x_{ij}^k \quad (4.1)$$

$$\sum_{i=0}^n \sum_{k=1}^K x_{ij}^k = 1 \quad \text{dla } j = 0, 1, \dots, n \quad (4.2)$$

$$\sum_{i=0}^n x_{ip}^k - \sum_{j=0}^n x_{pj}^k = 0 \quad \text{dla } p = 0, 1, \dots, n \text{ oraz } k = 1, \dots, K \quad (4.3)$$

$$\sum_{j=1}^n x_{0j}^k \leq 1 \quad \text{dla } k = 1, \dots, K \quad (4.4)$$

$$z_i - z_j + (n+1) \sum_{k=1}^K x_{ij}^k \leq (n+1) - 1 \quad \text{dla } i, j = 1, \dots, n \text{ oraz } i \neq j \quad (4.5)$$

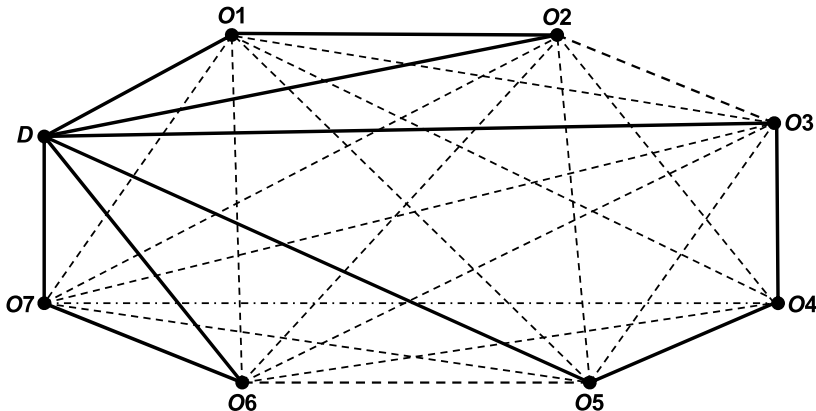
$$x_{ij}^k = \begin{cases} 0 \\ 1 \end{cases} \quad \text{dla } i, j = 0, 1, \dots, n \text{ oraz } k = 1, \dots, K \quad (4.6)$$

Liczba zmiennych decyzyjnych w prezentowanym modelu jest równa iloczynowi: $(n+1)(n+1)k$.

Ograniczenia wyrażone wzorem (4.2) przedstawiają założenie, że każdy punkt obsługi jest jednoznacznie przyporządkowany do trasy jednego pojazdu. Ograniczenia określone zespołem równań (4.3) zapewniają, że każdy punkt obsługi zostanie odwiedzony (pojazd raz do niego wjedzie i raz z niego wyjedzie). Założenie to dotyczy także bazy. Natomiast zespół nierówności (4.4) ma zagwarantować, że pojazd rozpoczyna i kończy kurs w bazie. Wreszcie ostatnia grupa ograniczeń (4.5) zabezpiecza przed powstaniem takich tras poszczególnych pojazdów, które są cyklami skróconymi.

Podobnie jak w wypadku problemu jednego komiwojażera, także zagadnienie układania tras wielu pojazdów można sformułować, opierając się na teorii grafów. Dany jest nieskierowany, spójny i pełny graf $G(V, E, c)$, w którym zbiór V stanowi $n + 1$ węzłów odpowiadających odbiorcom oraz dostawcy (rys. 4.3). Zbiór krawędzi E grafu reprezentuje możliwe połączenia pomiędzy punktami i oraz j . Każdej krawędzi grafu łączącej dwa dowolne węzły przyporządkowana jest nieujemna liczba c_{ij} (waga krawędzi) odpowiadająca długości (kosztowi) połączenia pomiędzy punktami i oraz j . Jeden z węzłów grafu reprezentuje punkt początkowy D , z którego dany pojazd rozpoczyna swoją trasę oraz do niego powraca. Za rozwiązanie dopuszczalne uważa się taki zbiór krawędzi, dla którego:

- pierwsza krawędź trasy danego pojazdu rozpoczyna się w węźle początkowym 0;
- ostatnia krawędź trasy danego pojazdu kończy się w węźle początkowym 0;
- każdy węzeł grafu stanowi połączenie dwóch i tylko dwóch krawędzi grafu;
- węzeł początkowy grafu stanowi połączenie krawędzi w liczbie równej dwukrotności liczby użytych pojazdów.



Rys. 4.3. Graf pełny dla problemu wielu komiwojażerów wraz z przykładowymi dopuszczalnymi trasami (cyklami Hamiltona)

Źródło: opracowanie własne.

4.2. Warianty problemu wielu komiwojazerów

W rozdziale poświęconym problemowi komiwojazera wymienione zostały jego odmiany wyróżnione między innymi ze względu na: symetryczność i niesymetryczność macierzy kosztów przejazdu pomiędzy punktami i oraz j , charakter grafu (pełny lub niepełny) czy charakter funkcji celu (minimalizacja czasu przejazdu). W odniesieniu do zagadnienia układania tras wielu pojazdów jednocześnie można rozpatrywać takie same jego odmiany, a ich sposób formułowania będzie analogiczny jak w wypadku klasycznego problemu komiwojazera.

Postać matematyczną problemu wielu komiwojazerów sformułowaną wzorami (4.1)–(4.6) należy uznać za bazowy model zagadnienia ustalania tras dla pojazdów. Jednakże istnienie wielu innych dodatkowych ograniczeń, z jakimi można się spotkać w praktyce gospodarczej, powoduje, że liczba wariantów zagadnienia wielu komiwojazerów jest olbrzymia. W dalszej części niniejszego rozdziału przedstawione zostaną najważniejsze z nich:

- problem wielu komiwojazerów z ograniczeniami zasobowymi;
- problem wielu komiwojazerów z oknami czasowymi;
- problem wielu komiwojazerów z dostawą i/lub odbiorem dóbr;
- problem wielu komiwojazerów z obsługą rozdzieloną;
- problem wielu komiwojazerów z uzupełnieniami ładunku;
- problem wielu komiwojazerów z obsługą okresową;
- stochastyczny problem wielu komiwojazerów;
- dynamiczny problem wielu komiwojazerów;
- wielokryterialny problem wielu komiwojazerów;
- problem wielu komiwojazerów z wieloma bazami.

4.2.1. Problem wielu komiwojazerów z ograniczeniami zasobowymi

Jednym z podstawowych rodzajów problemu wielu komiwojazerów jest zagadnienie z ograniczeniami zasobowymi CVRP (*Capacitated Vehicle Routing Problem*). Należy tu jednocześnie podkreślić, że wariant ten jest bardzo często uznawany za podstawowe sformułowanie problemu decyzyjnego układania tras pojazdów. Z tego też powodu właśnie tej odmianie problemu komiwojazerów należy poświęcić więcej uwagi.

Formułując model wielu komiwojazerów z ograniczeniami zasobowymi, trzeba przyjąć następujące założenia:

- w rejonie działania bazy (punkt 0) znajduje się określona liczba n obsługiwanych punktów i , do których dostarczane jest (lub jest od nich odbierane) określone jednorodne dobro;

- z każdym punktem i związana jest pewna wielkość q_i przedstawiająca popyt punktu na dobro (w wypadku zadania dostawy) lub podaż punktu i (w wypadku zadania odbioru);
- baza (punkt 0) jest w stanie zaspokoić popyt q_i każdego punktu obsługi i lub odwrotnie – zapewnia możliwość odbioru dobra w każdej ilości q_i od każdego punktu obsługi i ;
- baza dysponuje pewną liczbą K pojazdów, z których każdy ma określoną ładowność Q_k (lub pojemność V_k);
- każdy punkt obsługi i może być odwiedzony przez pojazd k tylko jeden raz podczas swojego kursu;
- każdy punkt obsługi i może być odwiedzony tylko przez określony pojazd k (jednoznaczne określenie rejonu obsługi punktów przez określony pojazd);
- znany jest koszt c_{ij} transportu dobra pomiędzy bazą a poszczególnymi punktami obsługi i oraz pomiędzy samymi punktami i oraz j ;
- wszystkie punkty muszą być odwiedzone przez pojazdy, tzn. zbiór tras pojazdów w ramach przyporządkowanych im punktów jest rozwiązaniem dopuszczalnym, jeżeli składa się ono z tras będących cyklami hamiltonowskimi (rys. 4.3).

Postać modelu wielu komiwojazerów z ograniczeniami zasobowymi jest rozszerzeniem modelu przedstawionego za pomocą równań (4.1)–(4.6).

$$\min. F(\mathbf{x}) = \sum_{i=0}^n \sum_{j=0}^n \sum_{k=1}^K c_{ij} x_{ij}^k \quad (4.7)$$

$$\sum_{i=0}^n \sum_{k=1}^K x_{ij}^k = 1 \quad \text{dla } j = 0, 1, \dots, n \quad (4.8)$$

$$\sum_{i=0}^n x_{ip}^k - \sum_{j=0}^n x_{pj}^k = 0 \quad \text{dla } p = 0, 1, \dots, n \text{ oraz } k = 1, \dots, K \quad (4.9)$$

$$\sum_{j=1}^n x_{0j}^k \leq 1 \quad \text{dla } k = 1, \dots, K \quad (4.10)$$

$$\sum_{i=1}^n q_i \sum_{j=0}^n x_{ij}^k \leq Q_k \quad \text{dla } k = 1, \dots, K; q_0 = 0 \quad (4.11)$$

$$\sum_{i=1}^n v_i \sum_{j=0}^n x_{ij}^k \leq V_k \quad \text{dla } k = 1, \dots, K; v_0 = 0 \quad (4.12)$$

$$z_i - z_j + (n+1) \sum_{k=1}^K x_{ij}^k \leq (n+1) - 1 \quad \text{dla } i, j = 1, \dots, n \text{ oraz } i \neq j \quad (4.13)$$

$$x_{ij}^k = \begin{cases} 0 \\ 1 \end{cases} \quad \text{dla } i, j = 0, 1, \dots, n \text{ oraz } k = 1, \dots, K \quad (4.14)$$

W powyższym modelu programowania liniowego ograniczenia wyrażone wzorami (4.8)–(4.10) oraz (4.13) mają taką samą interpretację jak ograniczenia

określone nierównościami (4.2)–(4.5). Ograniczenia (4.11) i (4.12) reprezentują ograniczenia zasobowe w postaci możliwości załadunkowych pojazdów wyrażonych zarówno w postaci dopuszczalnej masy ładunku, jak i dopuszczalnej pojemności ładunku.

Wśród ograniczeń zasobowych spotkać można także te, które nie dotyczą bezpośrednio możliwości załadunkowych posiadanego taboru pojazdów. Ograniczenia te są nakładane na kierowców w postaci dopuszczalnego czasu pracy. Zakładając, że na czas pracy kierowcy składają się:

- czasy przejazdu t_{ij}^k pojazdu k z punktu i do punktu j ;
- czasy obsługi t_i^k pojazdu k w punkcie i ;

ograniczenie dotyczące maksymalnego czasu pracy kierowcy T_k , a tym samym danego pojazdu sformułować można w następujący sposób:

$$\sum_{i=1}^n t_i^k \sum_{j=0}^n x_{ij}^k + \sum_{i=0}^n \sum_{j=0}^n t_{ij}^k x_{ij}^k \leq T_k \quad k = 1, \dots, K \quad (4.15)$$

Bardzo często czas t_{ij}^k można określić jako iloraz odległości pomiędzy punktem i oraz j oraz średniej prędkości pojazdu.

Inne postaci problemu wielu komiwojazerów, które można spotkać, oparte są na przedstawionej wcześniej definicji grafu pełnego $G(V, E, c)$. Przyjmując następujące założenia:

- V – zbiór wierzchołków $n + 1$ grafu, z których jeden oznaczony jako 0 reprezentuje bazę, a pozostałe wierzchołki $i \in V \setminus \{0\}$ reprezentują n obsługiwanych punktów;
- E – zbiór wszystkich krawędzi grafu łączących dwa dowolne wierzchołki i oraz j ;
- e – krawędź łącząca dwa wierzchołki i oraz j : $e \in E = \{(i, j) : i, j \in V, i < j\}$;
- c_e – koszt przypisany do krawędzi e grafu (koszt przejazdu pomiędzy punktem i a punktem j);
- Q – ładowność pojazdów;
- K – liczba takich samych pojazdów o ładowności Q ;
- x_e – zmienna decyzyjna oznaczająca, ile razy dana krawędź e jest włączana do rozwiązania (stanowi fragment trasy pojazdu);
- $r(S)$ – minimalna liczba pojazdów potrzebna do obsłużenia podzbioru S obsługiwanych punktów;
- $\delta(S)$ – zbiór krawędzi e , których jeden koniec należy do zbioru S , natomiast drugi koniec do niego nie należy: $\delta(S) = \{(i, j) : i \in S, j \notin S \cup i \notin S, j \in S\}$;
- $\delta(0)$ – zbiór krawędzi e , których jeden koniec stanowi baza, natomiast drugi koniec stanowi punkt obsługi;

zaproponowany model symetrycznego problemu układania tras pojazdów przedstawia się następująco (Laporte i in., 1985):

$$\min. F(\mathbf{x}) = \sum_{e \in E} c_e x_e \quad (4.16)$$

$$\sum_{e \in \delta(i)} x_e = 2 \quad \text{dla } i \in V \setminus \{0\} \quad (4.17)$$

$$\sum_{e \in \delta(0)} x_e = 2K \quad (4.18)$$

$$\sum_{e \in \delta(S)} x_e \geq 2r(S) \quad \text{dla } S \subseteq V \setminus \{0\}, S \neq \emptyset \quad (4.19)$$

$$x_e \in [0, 1] \quad \text{dla } e \notin \delta(0) \quad (4.20)$$

$$x_e \in [0, 1, 2] \quad \text{dla } e \in \delta(0) \quad (4.21)$$

Ograniczenia w modelu wyrażone równaniami (4.17) przedstawiają stopień wierzchołków w grafie, który jest równy 2, co oznacza, że z każdym wierzchołkiem grafu skojarzone muszą być tylko dwie krawędzie (dojazd pojazdu do punktu i jego opuszczenie przez pojazd). Ograniczenia te dotyczą wszystkich wierzchołków z wyjątkiem oznaczonego numerem 0, reprezentującego bazę.

Stopień wierzchołka oznaczonego numerem 0 jest równy dwukrotnej liczbie pojazdów, jakimi dysponuje baza, i jest reprezentowany przez ograniczenie wyrażone równościami (4.18). Innymi słowy, zapewnia ono, że utworzony zostanie zbiór tras dla K pojazdów.

Ograniczenia wyrażone nierównościami (4.19) gwarantują zarówno spójność rozwiązania, jak i obostrzenia dotyczące ładowności pojazdów. Służą do tego określenie odpowiedniej liczby krawędzi wchodzących w skład dowolnego podzbioru wierzchołków. Na uwagę zasługuje tutaj wielkość określona jako $r(S)$. Oznacza ona minimalną liczbę pojazdów niezbędną do obsłużenia wszystkich punktów należących do podzbioru S . Wartość ta może zostać ustalona przez rozwiązanie odpowiedniego problemu plecakowego, w którym zbiorem przedmiotów jest rozpatrywany podzbiór klientów z określonymi przez nich popytami q_i , natomiast zbiorem pojemników (plecaków) są pojazdy o ładowności Q_k . Należy zauważyć, że problem plecakowy, podobnie jak problem jednego czy wielu komiwojażerów, jest zagadnieniem o charakterze *NP-trudnym*. Stąd też wielkość $r(S)$ może zostać oszacowana jako dolna granica rozwiązania problemu plecakowego równa w szczególności:

$$r(S) = \left\lceil \frac{\sum_{i \in S} q_i}{Q} \right\rceil \quad (4.22)$$

Ostatnie dwie grupy ograniczeń przedstawionego powyżej modelu (4.20) i (4.21) zapewniają odpowiednio, że:

- każda krawędź grafu łącząca dwa dowolne wierzchołki i oraz j jest włączona do rozwiązania końcowego co najwyżej jeden raz (trasa pomiędzy punktami i oraz j pokonywana jest tylko przez jeden pojazd jeden raz lub wcale);
- każda krawędź grafu skojarzona z wierzchołkiem reprezentującym bazę jest włączona do rozwiązania końcowego co najwyżej dwa razy (w wypadku, gdy do trasy pojazdu wchodzi tylko jeden odwiedzany punkt, krawędź łącząca wierzchołek reprezentujący ten punkt oraz wierzchołek oznaczony numerem 0 oznacza połączenie pokonywane przez pojazd dwukrotnie).

Kolejnym często spotykanym sformułowaniem problemu wielu komiwojazerów jest jego wersja oparta na zagadnieniu pokrycia zbioru. Można je określić jako zagadnienie podziału pewnego zbioru Z na rodzinę podzbiorów $\{Z_1, Z_2, \dots, Z_n\}$, która w sumie daje zbiór Z . Celem jest znalezienie takiej rodziny podzbiorów, której liczebność będzie jak najmniejsza. Jak można zauważyć, w zagadnieniu wielu komiwojazerów takimi podzbiarami będą trasy poszczególnych pojazdów pokrywające poszczególne odwiedzane punkty. Zbiór tras dopuszczalnych będzie stanowił rodzinę podzbiorów podlegającą ocenie.

Przyjmując następujące założenia:

- T – zbiór wszystkich możliwych dopuszczalnych tras: $T = \{T_1, T_2, \dots, T_s\}$, gdzie s oznacza liczebność zbioru T : $s = |T|$;
- c_j – koszt trasy T_j ;
- a_{ij} – zmienna przyjmująca wartość 1 wtedy i tylko wtedy, gdy trasa T_j uwzględnia (pokrywa) punkt i , oraz 0, gdy go nie uwzględnia;
- Q – ładowność pojazdów;
- K – liczba takich samych pojazdów o ładowności Q ;
- x_j – zmienna decyzyjna przyjmująca wartość 1 wtedy i tylko wtedy, gdy trasa T_j stanowi część rozwiązania (zbioru tras), oraz 0, gdy jej nie stanowi;

zaproponowany model problemu wielu komiwojazerów oparty na problemie pokrycia zbioru jest następujący (Balinski i Quandt, 1964):

$$\min. F(\mathbf{x}) = \sum_{j=1}^s c_j x_j \quad (4.23)$$

$$\sum_{j=1}^s a_{ij} x_j = 1 \quad \text{dla } i \in V \setminus \{0\} \quad (4.24)$$

$$\sum_{j=1}^s x_j = K \quad (4.25)$$

$$x_j \in [0, 1] \quad \text{dla } j = 1, 2, \dots, s \quad (4.26)$$

Ograniczenia wyrażone równaniami (4.24) zapewniają, że każdy punkt obsługi i jest zawarty dokładnie w jednej trasie. Natomiast zespół równań (4.25) warunkuje wykorzystanie w rozwiązaniu końcowym K pojazdów.

4.2.2. Problem wielu komiwojażerów z oknami czasowymi

Jednym z najważniejszych, a zarazem najczęściej rozpatrywanych wariantów problemu wielu komiwojażerów jest zagadnienie układania tras pojazdów z oknami czasowymi (*Vehicle Routing Problem With Time Windows* – VRPTW). Istotą tego problemu jest to, że pojazd, który ma za zadanie obsłużyć zadany punkt i , musi to zadanie wykonać w ściśle określonym przedziale czasowym $[a_i, b_i]$. Dopuszcza się przybycie pojazdu do punktu i przed czasem, czyli zanim zostanie otwarte okno czasowe obsługi punktu i . W takim wypadku pojazd musi poczekać na moment, w którym rozpocznie się możliwość obsługi punktu i . Natomiast niedopuszczalne jest przybycie pojazdu do rozpatrywanego punktu i po tym czasie.

Problem układania tras pojazdów z oknami czasowymi można uznać za uogólnienie zagadnienia układania tras pojazdów z ograniczeniami zasobowymi. Jeżeli przyjęte zostaną następujące założenia: $a_i = 0$ oraz $b_i = \infty$, wtedy zagadnienie układania tras pojazdów staje się szczególnym przypadkiem problemu wielu komiwojażerów z oknami czasowymi.

Jedną z formalnych postaci problemu wielu komiwojażerów z oknami czasowymi jest model podobny częściowo do przedstawionego wzorami (4.7)–(4.14) (Desrochers i in., 1988; 1992). Punktem wyjścia jest tutaj graf skierowany $G(V, A, c)$, na podstawie którego przyjęto następujące założenia:

- V – zbiór wierzchołków $n + 2$ grafu, w którym dwa wierzchołki oznaczone zostały odpowiednio jako 0 oraz $n + 1$ i reprezentują bazę, a pozostałe wierzchołki reprezentują n obsługiwanych punktów;
- A – zbiór wszystkich krawędzi skierowanych grafu łączących dwa dowolne wierzchołki i oraz j ;
- c_{ij} – koszt przypisany do każdej krawędzi grafu (koszt przejazdu pomiędzy punktem i a punktem j);
- q_i – zapotrzebowanie punktu i ;
- Q – ładowność pojazdów;
- K – liczba takich samych pojazdów o ładowności Q ;
- $\delta^+(i)$ – zbiór tych wierzchołków grafu, które stanowią koniec krawędzi (i, j) mającej swój początek w wierzchołku i : $\delta^+(i) = \{j : (i, j) \in A\}$;
- $\delta^-(j)$ – zbiór tych wierzchołków grafu, które stanowią początek krawędzi (i, j) mającej swój koniec w wierzchołku j : $\delta^-(j) = \{i : (i, j) \in A\}$;
- s_i – czas obsługi punktu i , przy czym $s_0 = 0$ oraz $s_{n+1} = 0$;
- t_{ij} – czas przypisany do każdej krawędzi grafu (czas przejazdu pomiędzy punktem i a punktem j);
- a_i – początek przedziału czasowego obsługi punktu i ;
- b_i – koniec przedziału czasowego obsługi punktu i ;

- x_{ij}^k – zmienna decyzyjna przyjmująca wartość 1 wtedy i tylko wtedy, gdy do trasy pojazdu $k \in K$ przyporządkowany został odcinek pomiędzy punktami i oraz j , natomiast 0, gdy nie został on przyporządkowany;
- w_i^k – zmienna decyzyjna oznaczająca moment, w którym następuje rozpoczęcie obsługi zadanego punktu i przez pojazd k .

Model dla problemu wielu komiwojazerów z oknami czasowymi sformułowany został w następujący sposób (4.27)–(4.36):

$$\min. F(\mathbf{x}) = \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ij}^k \quad (4.27)$$

$$\sum_{k \in K} \sum_{j \in \delta^+(i)} x_{ij}^k = 1 \quad \text{dla } i = 0, 1, \dots, n \quad (4.28)$$

$$\sum_{j \in \delta^+(0)} x_{0j}^k = 1 \quad \text{dla } k = 1, \dots, K \quad (4.29)$$

$$\sum_{i \in \delta^-(j)} x_{ij}^k - \sum_{i \in \delta^+(j)} x_{ij}^k = 0 \quad \text{dla } j = 0, 1, \dots, n \text{ oraz } k = 1, \dots, K \quad (4.30)$$

$$\sum_{j \in \delta^-(n+1)} x_{i,n+1}^k = 1 \quad \text{dla } k = 1, \dots, K \quad (4.31)$$

$$x_{ij}^k (w_i^k + s_i + t_{ij} - w_j^k) \leq 0 \quad \text{dla } i, j = 0, 1, \dots, n+1 \text{ oraz } k = 1, \dots, K \quad (4.32)$$

$$w_i^k \geq a_i \quad \text{dla } i = 1, \dots, n+1 \text{ oraz } k = 1, \dots, K \quad (4.33)$$

$$w_i^k \leq b_i \quad \text{dla } i = 1, \dots, n+1 \text{ oraz } k = 1, \dots, K \quad (4.34)$$

$$\sum_{i \in V \setminus \{0, n+1\}} q_i \sum_{j \in \delta^+(i)} x_{ij}^k \leq Q \quad \text{dla } k = 1, \dots, K \quad (4.35)$$

$$x_{ij}^k = \begin{cases} 0 \\ 1 \end{cases} \quad \text{dla } i, j = 0, 1, \dots, n+1 \text{ oraz } k = 1, \dots, K \quad (4.36)$$

Celem reprezentowanym przez funkcję (4.27) jest minimalizacja łącznego dystansu pokonanego przez wszystkie pojazdy. Ograniczenie (4.28) warunkuje odwiedzenie każdego punktu obsługi. Ograniczenia (4.29)–(4.31) zapewniają spełnienie podstawowych założeń problemu wielu komiwojazerów przez dany pojazd, czyli odpowiednio: raz opuszcza bazę (4.29), raz wjeżdża i raz wyjeżdża z zadanego punktu (4.30) oraz raz wraca do bazy (4.31). Ograniczenia wyrażone nierównościami (4.32) zapewniają ciągłość czasu obsługi zadanych punktów przez każdy pojazd. Natomiast nierówności (4.33) i (4.34) gwarantują obsługę punktów przez pojazd w ramach zadanych okien czasowych. Wreszcie zbiór ograniczeń (4.35) stanowi o nieprzekraczaniu ładowności pojazdu.

Na szczególną uwagę zasługują ograniczenia wyrażone nierównościami (4.32). Ich postać czyni z modelu (4.27)–(4.36) zadanie programowania nieliniowego. Jednakże mogą one zostać sprowadzone do postaci liniowej w następujący sposób:

$$w_j^k \geq w_i^k + s_i + t_{ij} - M_{ij}(1 - x_{ij}^k) \quad \text{dla } i, j = 0, 1, \dots, n + 1 \text{ oraz } k = 1, \dots, K \quad (4.37)$$

gdzie M_{ij} jest stałą równą:

$$M_{ij} = \max. \{0, w_i^k + s_i + t_{ij} - a_j\} \quad (4.38)$$

Ponadto zauważono, że ograniczenia dotyczące zmiennej decyzyjnej w_i^k mogą zostać zaostrzone odpowiednio (Desrochers i Laporte, 1991):

$$w_i^k \geq a_i + \sum_{j \in \delta^-(i)} \max. \{0, a_j - a_i + s_j + t_{ji}\} x_{ji}^k \quad \text{dla } i, j = 0, 1, \dots, n + 1 \text{ oraz } k = 1, \dots, K \quad (4.39)$$

$$w_i^k \leq b_i + \sum_{j \in \delta^+(i)} \max. \{0, b_i - b_j + s_j + t_{ji}\} x_{ij}^k \quad \text{dla } i, j = 0, 1, \dots, n + 1 \text{ oraz } k = 1, \dots, K \quad (4.40)$$

Na koniec warto zwrócić jeszcze uwagę na granice przedziałów czasowych wyznaczających dopuszczalny okres obsługi konkretnych punktów. Zazwyczaj nie są one w jakiś szczególnie sposób sprecyzowane, jeżeli chodzi o pracę bazy, skąd wyrusza i dokąd wraca każdy pojazd: a_0 , b_0 , a_{n+1} oraz b_{n+1} . Niemniej jednak można je wyznaczyć w następujący sposób:

$$a_0 = \min_{i \in V \setminus \{0, n+1\}} \{a_i - t_{0i}\} \quad (4.41)$$

$$b_0 = \max_{i \in V \setminus \{0, n+1\}} \{b_i - t_{0i}\} \quad (4.42)$$

$$a_{n+1} = \min_{i \in V \setminus \{0, n+1\}} \{a_i + s_i = t_{i, n+1}\} \quad (4.43)$$

$$b_{n+1} = \max_{i \in V \setminus \{0, n+1\}} \{b_i + s_i = t_{i, n+1}\} \quad (4.44)$$

Istnieją dwie odmiany zagadnienia układania tras pojazdów z oknami czasowymi. Pierwsza z nich w sposób mniej restrykcyjny traktuje określony przedział czasu przeznaczony na obsługę punktu. W tej sytuacji aprobuje się późniejsze przybycie pojazdu. Pozwala to uznać takie rozwiązanie za dopuszczalne, jednak za cenę nałożenia pewnej kary w postaci dodatkowego kosztu dodanego do wartości funkcji celu. Zwykle wynika to z przyjętych przedziałów czasowych dla poszczególnych punktów. Jak zauważono, samo znalezienie rozwiązania dopuszczalnego problemu

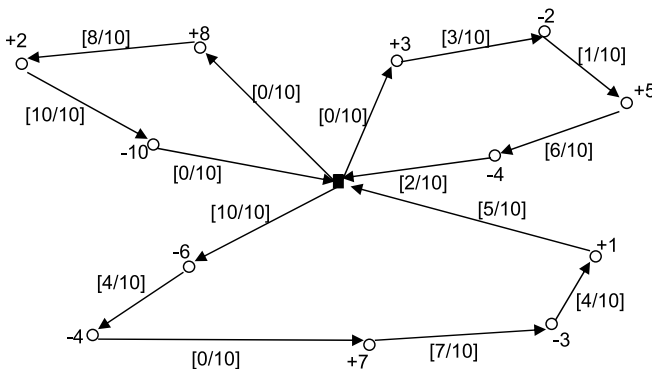
wielu komiwojazerów z oknami czasowymi może być bardzo trudne i może mieć charakter zadania *NP*-zupełnego (Savelsbergh, 1985). Druga odmiana nakazuje ściśle przestrzeganie czasu, w którym może nastąpić obsługa punktu.

4.2.3. Problem wielu komiwojazerów z dostawą i/lub odbiorem dóbr

Dotychczas analizowane warianty zagadnienia układania tras pojazdów dotyczyły sytuacji, w której dostawca dostarcza produkt (lub produkty) do sieci odbiorców bądź dokonuje tylko jego odbioru. W związku z coraz większą popularnością ekologii, a także rosnącym znaczeniem logistyki zwrotnej coraz istotniejsze wśród problemów wielu komiwojazerów stają się te, w których produkt może być zarówno dostarczany do klienta, jak i od niego odbierany.

Jako że punkty obsługiwane przez pojazd mogą być: tylko dostawcami, tylko odbiorcami bądź zarówno dostawcami, jak i odbiorcami, wyróżnić można kilka typów problemów wielu komiwojazerów z dostawą i/lub odbiorem dóbr (Berbeglia i in., 2007).

Pierwszym z nich jest zagadnienie układania tras pojazdów typu „wielu-do-wielu”, w którym dostawcami i odbiorcami produktu (lub wielu produktów) oprócz bazy są obsługiwane przez pojazd punkty. Część z nich stanowi zbiór tylko dostawców, część zaś – tylko odbiorców. Co więcej, każdy z odbiorców może otrzymać produkt z różnych miejsc źródłowych, przy czym jednocześnie każdy z dostawców może wysłać produkt do różnych miejsc przeznaczenia. Na rysunku 4.4 przedstawiono postać graficzną omawianego problemu, gdzie w nawiasach kwadratowych podano dwie liczby $[u / Q_{\max}]$: u – aktualne wykorzystanie ładowności pojazdu oraz Q_{\max} – maksymalna ładowność pojazdu (10 jednostek), natomiast przy każdym punkcie obsługi podane zostały odpowiednio liczba jednostek podejmowanych przez pojazd (ze znakiem „+”) oraz liczba jednostek pozostawianych przez pojazd (ze znakiem „-”).



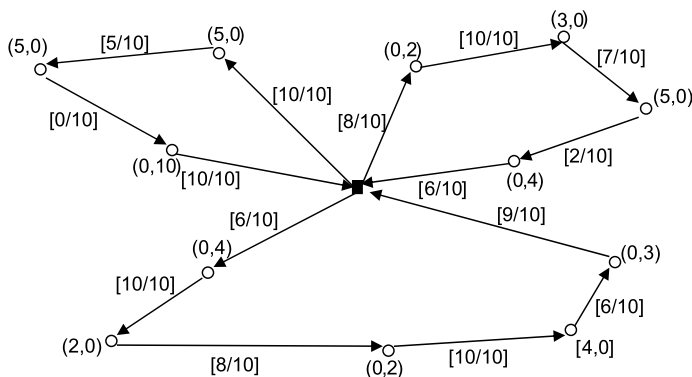
Rys. 4.4. Problem wielu komiwojazerów z dostawą lub odbiorem dóbr typu „wielu-do-wielu”
Źródło: opracowanie własne.

Ów typ zagadnienia wielu komiwojazerów z dostawą i/lub odbiorem dóbr znajduje zastosowanie w sytuacjach, gdy na przykład istnieje konieczność relokacji posiadanych zasobów pomiędzy różnymi oddziałami tego samego przedsiębiorstwa.

Drugim rodzajem tego właśnie zagadnienia jest problem typu „jeden-do-wielu-do-jeden”. Występują tu przynajmniej dwa rodzaje przewożonych przez pojazdy produktów. Jeden z nich jest dostarczany do klientów tylko z bazy („jeden-do-wielu”), drugi zaś jest od nich odbierany i dostarczany tylko do bazy („wielu-do-jeden”). W omawianym zagadnieniu wyróżnić można kilka wariantów, które wynikają z następujących przyczyn:

- każdy z obsługiwanych punktów zgłasza tylko popyt na dostarczany jeden produkt lub tylko podaż na odbierany drugi produkt;
- każdy z obsługiwanych punktów zgłasza zarówno popyt na dostarczany jeden produkt, jak i podaż na odbierany drugi produkt;
- dla wykorzystanych pojazdów nie ma znaczenia kolejność dostaw i odbiorów (dostawa i odbiór przez pojazd może występować naprzemiennie);
- odbiory produktu przez pojazdy mogą zostać rozpoczęte tylko w sytuacji, gdy zakończone zostaną na trasie pojazdu wszystkie dostawy.

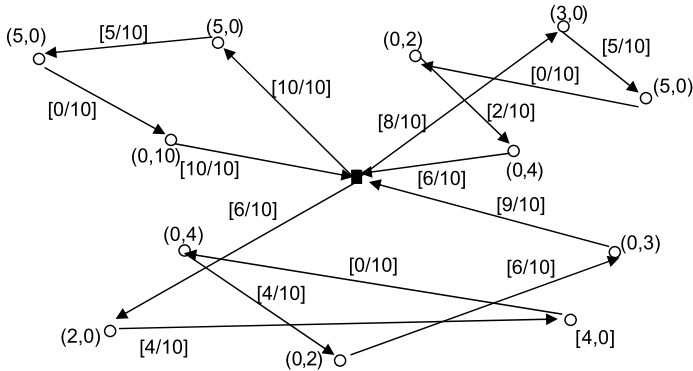
Na problem wielu komiwojazerów z odbiorem i/lub dostawą dóbr typu „jeden-do-wielu-do-jeden” składają się tym samym cztery warianty pokazane na poniższych ilustracjach. Pierwsza z nich (rys. 4.5a) przedstawia sytuację, w której każdy z punktów obsługi zgłasza tylko popyt na produkt lub podaż, a jednocześnie kolejność ich odwiedzania przez dany pojazd może być dowolna. Przy każdym punkcie obsługi w nawiasach podane są dwie liczby: pierwsza z nich oznacza popyt (pozostawiana liczba jednostek ładunkowych), natomiast druga przedstawia podaż (pobierana liczba jednostek ładunkowych). Wartości w nawiasach kwadratowych reprezentują stopień wykorzystania ładowności pojazdów.



Rys. 4.5a. Problem typu „jeden-do-wielu-do-jeden” – dostawa lub odbiór z dowolną kolejnością obsługi

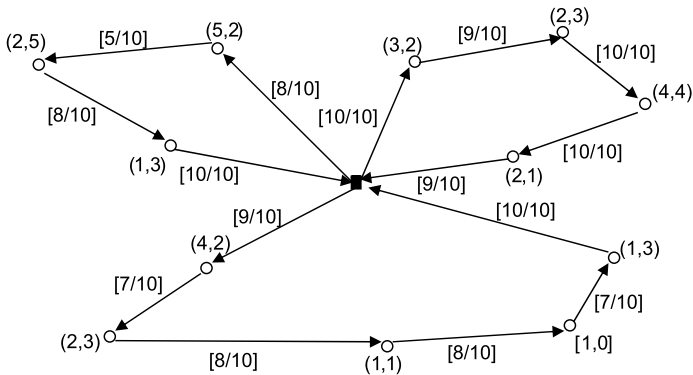
Źródło: opracowanie własne.

Druga sytuacja, przedstawiona na rysunku 4.5b, jest analogiczna do poprzedniej, lecz odbiory produktu mogą się rozpocząć dopiero po zrealizowanych dostawach.



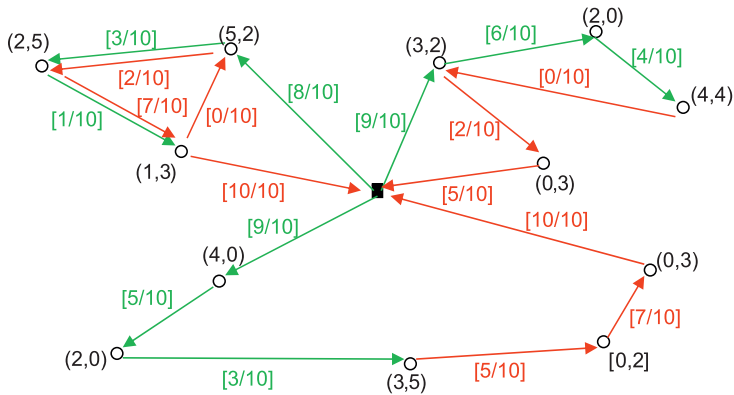
Rys. 4.5b. Problem typu „jeden-do-wielu-do-jeden” – dostawa lub odbiór w kolejności: najpierw dostawy, potem odbiory
Źródło: opracowanie własne.

Kolejny wariant (rys. 4.5c) to jednoczesna dostawa i odbiór jednostek produktów w każdym punkcie obsługi wraz z dowolną kolejnością ich odwiedzania przez zadany pojazd.



Rys. 4.5c. Problem typu „jeden-do-wielu-do-jeden” – dostawa oraz odbiór z dowolną kolejnością obsługi
Źródło: opracowanie własne.

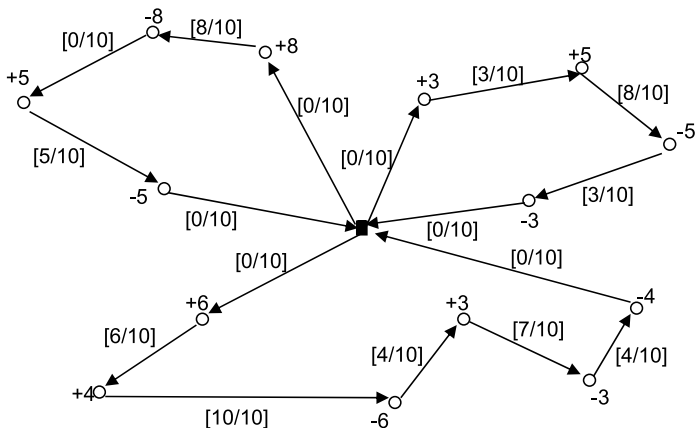
Ostatni przypadek (rys. 4.5d) to jednoczesna dostawa i odbiór jednostek produktów w każdym punkcie obsługi wraz z realizacją przez pojazdy najpierw dostaw, a dopiero w następnej kolejności odbiorów. W tej sytuacji niektóre punkty oraz odcinki tras mogą być odwiedzane dwukrotnie przez pojazd.



Rys. 4.5d. Problem typu „jeden-do-wielu-do-jeden” – dostawa oraz odbiór w kolejności: najpierw dostawy, potem odbiory

Źródło: opracowanie własne.

Wreszcie ostatnim rodzajem problemu wielu komiwojażerów z dostawą i/lub odbiorem dóbr jest zagadnienie typu „jeden-do-jeden”. Podobnie jak w zagadnieniu „wielu-do-wielu” punkty obsługi mogą być zarówno dostawcami, jak i odbiorcami produktu. Jednakże produkt ma tylko jednego dostawcę i jednego skojarzonego z nim odbiorcę. Tym samym w trasie każdego pojazdu musi pojawić się relacja dostawca–odbiorca o zapotrzebowaniu odbiorcy równym podaży dostawcy (rys. 4.6). Oczywiście jest, że w tworzonej kolejności obsługi punktów przez pojazd odbiorca nie może pojawić się przed swoim dostawcą. Może się natomiast pojawić po odwiedzeniu przez pojazd innych dostawców i odbiorców. Nie pozwala to na podzielenie problemu na mniejsze podproblemy jednego komiwojażera, co byłoby możliwe, gdyby odbiorca miał być w trasie bezpośrednio po



Rys. 4.6. Problem wielu komiwojażerów z dostawą lub odbiorem dóbr typu „jeden-do-jeden”

Źródło: opracowanie własne.

swoim dostawcy (w trasach pojazdu byłyby odcinki tworzone przez bezpośrednie połączenia par punktów dostawca–odbiorca).

Przykładem formalnej postaci zagadnienia wielu komiwojazerów z dostawą i odbiorem dóbr jest model przedstawiony w pracy G. Desaulniers'a i in. (2002) – typu „jeden-do-jeden”. Uwzględni on dodatkowo ograniczenia czasowe dla poszczególnych punktów obsługi. Tym samym jest też reprezentantem problemu wielu komiwojazerów z oknami czasowymi.

Do budowy modelu matematycznego założono istnienie dwóch tak samo licznych zbiorów punktów obsługi:

- 1) zbiór dostawców $P = \{1, 2, \dots, n\}$;
- 2) zbiór odbiorców $D = \{n + 1, n + 2, \dots, 2n\}$;

stanowiących łącznie zbiór wszystkich obsługiwanych punktów N .

Przyjęto następujące oznaczenia:

- A – zbiór wszystkich możliwych połączeń pomiędzy punktami obsługi oraz bazą;
- d_i – wielkość transportu z punktu obsługi i do punktu obsługi $n + i$;
- l_i – pobierana ilość d_i produktu u odbiorcy i ;
- l_{n+i} – pozostawiana ilość d_i produktu u odbiorcy i ($l_{n+i} = -d_i$);
- Q – ładowność pojazdów;
- K – liczba pojazdów o ładowności Q ;
- P_k – zbiór dostawców obsługiwanych przez pojazd k ;
- D_k – zbiór odbiorców obsługiwanych przez pojazd k ;
- N_k – zbiór dostawców i odbiorców obsługiwanych przez pojazd k ;
- $o(k)$ – punkt startowy, z którego wyrusza pojazd k ;
- $d(k)$ – punkt docelowy, w którym pojazd k kończy trasę;
- A_k – zbiór wszystkich możliwych połączeń pomiędzy punktami obsługi oraz bazą przypisanymi do pojazdu k ;
- c_{ij} – koszt przejazdu pomiędzy punktem i a punktem j ;
- t_{ij} – czas przejazdu pomiędzy punktem i a punktem j ;
- s_i – czas obsługi punktu i ;
- a_i – początek przedziału czasowego obsługi punktu i ;
- b_i – koniec przedziału czasowego obsługi punktu i ;
- x_{ij}^k – zmienna decyzyjna przyjmująca wartość 1 wtedy i tylko wtedy, gdy do trasy pojazdu k przyporządkowany został odcinek pomiędzy punktami i oraz j , natomiast 0, gdy nie został on przyporządkowany;
- T_i^k – zmienna decyzyjna oznaczająca moment, w którym następuje rozpoczęcie obsługi zadanego punktu i przez pojazd k ;
- L_i^k – ładowność pojazdu k po obsłudze zadanego punktu i .

Postać modelu wielu komiwojazerów dla dostawy i odbioru dóbr typu „jeden-do-jeden” jest następująca (4.45)–(4.61):

$$\min. F(\mathbf{x}) = \sum_{k \in K} \sum_{(i,j) \in A_k} c_{ij} x_{ij}^k \quad (4.45)$$

$$\sum_{k \in K} \sum_{j \in N_k \cup \{d(k)\}} x_{ij}^k = 1 \quad \text{dla } i \in P_k \quad (4.46)$$

$$\sum_{j \in N_k} x_{ij}^k - \sum_{j \in N_k} x_{j,n+i}^k = 0 \quad \text{dla } i \in P_k \text{ oraz } k \in K \quad (4.47)$$

$$\sum_{j \in P_k \cup \{d(k)\}} x_{ij}^k = 1 \quad \text{dla } i \in P_k \text{ oraz } k \in K \quad (4.48)$$

$$\sum_{j \in N_k \cup \{o(k)\}} x_{ij}^k - \sum_{j \in N_k \cup \{d(k)\}} x_{ij}^k = 0 \quad \text{dla } i \in N_k \text{ oraz } k \in K \quad (4.49)$$

$$\sum_{i \in D_k \cup \{o(k)\}} x_{i,d(k)}^k = 1 \quad \text{dla } k \in K \quad (4.50)$$

$$x_{ij}^k (T_i^k + s_i + t_{ij} - T_j^k) \leq 0 \quad \text{dla } (i,j) \in A_k \text{ oraz } k \in K \quad (4.51)$$

$$T_i^k \geq a_i \quad \text{dla } i \in V_k \text{ oraz } k \in K \quad (4.52)$$

$$T_i^k \leq b_i \quad \text{dla } i \in V_k \text{ oraz } k \in K \quad (4.53)$$

$$T_i^k + t_{i,n+1}^k \leq T_{n+i}^k \quad \text{dla } i \in P_k \text{ oraz } k \in K \quad (4.54)$$

$$x_{ij}^k (L_i^k + l_j - L_j^k) = 0 \quad \text{dla } (i,j) \in A_k \text{ oraz } k \in K \quad (4.55)$$

$$L_i^k \geq l_i \quad \text{dla } i \in P_k \text{ oraz } k \in K \quad (4.56)$$

$$L_i^k \leq Q \quad \text{dla } i \in P_k \text{ oraz } k \in K \quad (4.57)$$

$$L_i^k \geq 0 \quad \text{dla } i \in D_k \text{ oraz } k \in K \quad (4.58)$$

$$L_i^k \leq Q - l_i \quad \text{dla } i \in D_k \text{ oraz } k \in K \quad (4.59)$$

$$L_{o(k)}^k = 0 \quad \text{dla } k \in K \quad (4.60)$$

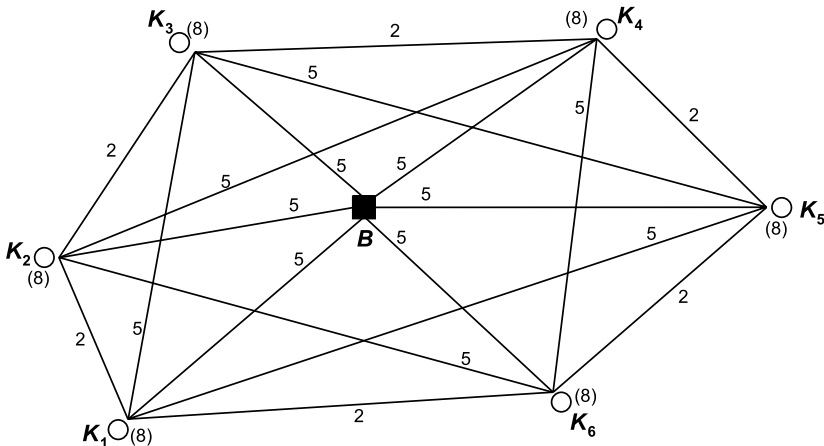
$$x_{ij}^k = \begin{cases} 0 \\ 1 \end{cases} \quad \text{dla } (i,j) \in A_k \text{ oraz } k \in K \quad (4.61)$$

Funkcja celu (4.45) minimalizuje całkowity koszt transportu wszystkich wykorzystanych pojazdów. Ograniczenia (4.46) i (4.47) zapewniają, że każda relacja dostawca–odbiorca jest obsługiwana tylko jeden raz i tylko przez jeden pojazd. Ograniczenia (4.48)–(4.50) dotyczą podstawowych założeń problemu wielu

komiwojazerów (każdy punkt odwiedzany jest tylko jeden raz, pojazd raz opuszcza bazę i raz do niej wraca). Ograniczenia (4.51)–(4.53) dotyczą okien czasowych obsługi poszczególnych punktów, których interpretacja jest taka sama jak omówionych w poprzednim podrozdziale ograniczeń (4.32)–(4.34). Również tutaj w trasie każdego pojazdu moment przyjazdu do dostawcy musi pojawić się przed momentem dostarczenia produktu odbiorcy, co jest zagwarantowane ograniczeniem (4.54). Ograniczenie (4.55) zapewnia zgodność bilansu produktów podjętych u dostawców i zostawionych u odbiorców przez pojazd. Nierówności (4.56)–(4.59) stanowią przedziały ograniczające dla wykorzystanej ładowności pojazdu po odwiedzeniu, odpowiednio, dostawców i odbiorców. Wreszcie pojazd, wyruszając w trasę, opuszcza bazę pusty, co przedstawia ograniczenie (4.60).

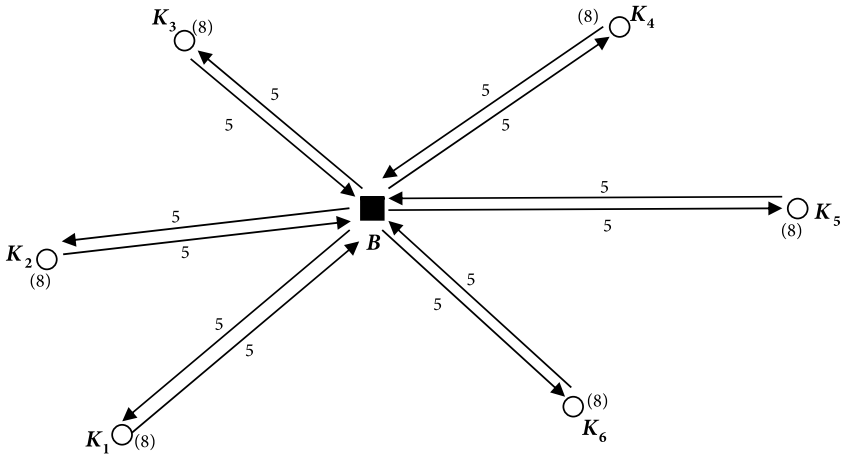
4.2.4. Problem wielu komiwojazerów z obsługą rozdzieloną

W podstawowym zagadnieniu układania tras pojazdów każdy punkt obsługi odwiedzany jest tylko jeden raz. Oznacza to wzajemnie jednoznaczne przyporządkowanie pojazdu do obsługiwanego punktu. W problemie wielu komiwojazerów z obsługą rozdzieloną każdy z obsługiwanym punktem może być odwiedzony przez więcej niż jeden pojazd. Założenie to pozwala rozdzielić popyt (lub podaż) punktu obsługi na wiele pojazdów będących w dyspozycji bazy (Dror i in., 1994; Chen i in., 2007). Może mieć ono szczególne uzasadnienie, gdy np. wielkość dostarczanego produktu do jednego klienta w znacznym stopniu, lecz nie całkiem wypełnia ładowność pojazdu. Omawiana sytuacja została przedstawiona na zamieszczonych poniżej ilustracjach. I tak na rysunku 4.7a widzimy przykładową sieć połączeń pomiędzy sześcioma klientami oraz bazą.



Rys. 4.7a. Wyjściowy problem wielu komiwojazerów
Źródło: opracowanie własne na podstawie (Chen i in., 2007).

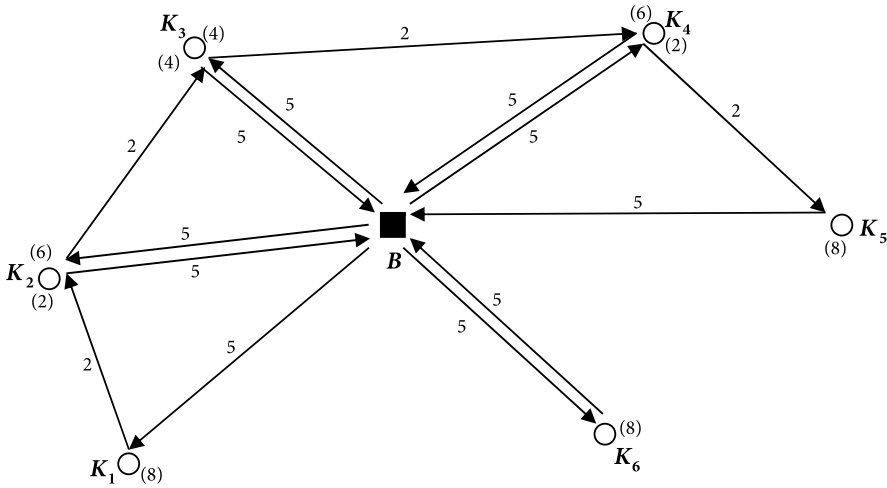
Na rysunku 4.7a przy krawędziach grafu nieskierowanego podano koszt połączenia (transportu), natomiast obok węzłów w nawiasach określona została liczba jednostek produktu, jaką należy dostarczyć do poszczególnych klientów. Zakładając, że ładowność każdego pojazdu będącego w dyspozycji bazy wynosi 10 jednostek, łatwo zauważyć, że w klasycznym problemie wielu komiwojażerów z ograniczeniami zasobowymi (dotyczącymi ładowności pojazdów) rozwiązaniem optymalnym będzie układ sześciu tras (rys. 4.7b). Koszt każdej trasy wyniesie 10, natomiast optymalna wartość funkcji celu (optymalny całkowity koszt wszystkich użytych środków transportu) wynosić będzie 60.



Rys. 4.7b. Rozwiązanie optymalne klasycznej postaci problemu wielu komiwojażerów z ograniczeniami zasobowymi

Źródło: opracowanie własne na podstawie (Chen i in., 2007).

W zaprezentowanym rozwiązaniu optymalnym dla klasycznego problemu wielu komiwojażerów z ograniczeniami zasobowymi nie ma możliwości obsługi przez pojazd więcej niż jednego klienta. Dołączenie do istniejącej trasy każdego następnego spowoduje przekroczenie dopuszczalnej ładowności pojazdu. Jak można zauważyć, ładowność każdego z nich wykorzystana zostanie w 80%. Maksymalizacja stopnia wykorzystania ładowności pojazdu jest bardzo ważnym celem wielu dostawców, a w szczególności operatorów logistycznych, którzy ich obsługują. Jednym ze sposobów jego osiągnięcia jest zrezygnowanie z warunku obsługi punktu tylko przez jeden pojazd, a tym samym dopuszczenie do obsługi rozdzielonej. Na rysunku 4.7c zaprezentowano układ tras dla przedstawionego problemu wielu komiwojażerów z obsługą rozdzieloną. Liczba pojazdów wykorzystanych do obsługi zadanych punktów jest o jeden mniejsza niż w wypadku rozwiązania klasycznego problemu wielu komiwojażerów. Ponadto ładowność czterech z pięciu pojazdów wykorzystana została w 100%, natomiast ładowność ostatniego także wynosi 80%. Oszczędność uzyskano też w wyniku zmniejszenia się łącznego kosztu wszystkich tras, która tutaj wynosi 58.



Rys. 4.7c. Układ tras dla problemu wielu komiwojazerów z obsługą rozdzieloną
Źródło: opracowanie własne na podstawie (Chen i in., 2007).

Dla problemu wielu komiwojazerów z obsługą rozdzieloną przedstawiona została formalna postać zagadnienia (Archetti i in., 2006). Przyjęto następujące oznaczenia:

- n – liczba punktów obsługi (baza oznaczona jest numerem 0);
- d_i – liczba jednostek produktu, jaką należy dostarczyć do punktu i (odebrać z punktu i);
- c_{ij} – koszt przejazdu pomiędzy punktem i a punktem j ;
- Q – ładowność pojazdów;
- K – liczba pojazdów o ładowności Q ;
- x_{ij}^k – zmienna decyzyjna przyjmująca wartość 1 wtedy i tylko wtedy, gdy trasy pojazdu k przyporządkowany został odcinek pomiędzy punktami i oraz j , natomiast 0 wtedy, gdy nie został on przyporządkowany;
- y_i^k – zmienna decyzyjna oznaczająca wielkość popytu (podaży) punktu obsługi i dostarczaną (odbieraną) przez pojazd k .

A postać modelu jest następująca:

$$\min. F(\mathbf{x}) = \sum_{i=0}^n \sum_{j=0}^n \sum_{k=1}^K c_{ij} x_{ij}^k \quad (4.62)$$

$$\sum_{i=0}^n \sum_{k=1}^K x_{ij}^k \geq 1 \quad \text{dla } j = 0, 1, \dots, n \quad (4.63)$$

$$\sum_{i=0}^n x_{ip}^k - \sum_{j=0}^n x_{pj}^k = 0 \quad \text{dla } p = 0, 1, \dots, n \text{ oraz } k = 1, \dots, K \quad (4.64)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij}^k \leq |S| - 1 \quad \text{dla } k = 1, \dots, K \text{ oraz } S \subset V \setminus \{0\} \quad (4.65)$$

$$y_i^k \leq d_i \sum_{j=0}^n x_{ij}^k \quad \text{dla } i = 1, \dots, n \text{ oraz } k = 1, \dots, K \quad (4.66)$$

$$\sum_{k=1}^K y_i^k = d_i \quad \text{dla } i = 1, \dots, n \quad (4.67)$$

$$\sum_{i=1}^n y_i^k \leq Q \quad \text{dla } k = 1, \dots, K \quad (4.68)$$

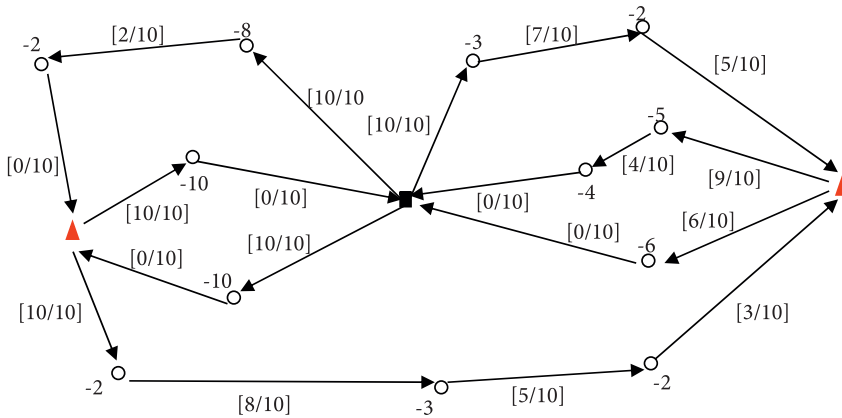
$$x_{ij}^k = \begin{cases} 0 \\ 1 \end{cases} \quad \text{dla } i, j = 0, 1, \dots, n \text{ oraz } k = 1, \dots, K \quad (4.69)$$

$$y_i^k \geq 0 \quad \text{dla } i = 1, \dots, n \text{ oraz } k = 1, \dots, K \quad (4.70)$$

Funkcja celu (4.62) reprezentuje całkowity koszt transportu wszystkich wykorzystanych pojazdów. Ograniczenia (4.63)–(4.65) to podstawowe warunki dotyczące klasycznego problemu wielu komiwojażerów, z tą różnicą, że zespół nierówności (4.63) zezwala na odwiedzenie danego punktu obsługi przez więcej niż jeden pojazd. Kolejna grupa ograniczeń (4.66)–(4.68) zapewnia prawidłowy rozdział dostarczanego (odbieranego) produktu pomiędzy poszczególne pojazdy. Oznacza to, że klient i jest obsługiwany przez pojazd k wtedy i tylko wtedy, gdy znajduje się on na trasie tego pojazdu (4.66), a także popyt (podaż) i -tego klienta musi zostać zrealizowany (4.67) bez przekroczenia ładowności k -tego pojazdu (4.68).

4.2.5. Problem wielu komiwojażerów z uzupełnieniami ładunku

Szczególnie interesującą odmianą zagadnienia układania tras pojazdów jest problem wielu komiwojażerów, w którym dopuszcza się uzupełnienia ładunku przez pojazd w trakcie realizowanej przez niego trasy. Sytuacje takie często się zdarzają między innymi w transporcie paliw płynnych (tj. gaz skroplony, olej opałowy czy paliwa napędowe). W klasycznej postaci problemu wielu komiwojażerów przyjmuje się podstawowe założenie, że baza dysponuje flotą pojazdów, która ma za zadanie w danym dniu obsłużyć zadaną liczbę klientów. W momencie, w którym pojazd rozwiezie cały ładunek, wraca do bazy, aby zakończyć swoją pracę lub ewentualnie uzupełnić ładunek, a następnie kontynuować dalszą część trasy. W wariacie układania tras pojazdów z możliwością uzupełnienia ładunku pojazdów zakłada się istnienie na obsługiwanym obszarze pewnej liczby punktów, w których każdy pojazd może dokonać uzupełnienia ładunku, a następnie dalej kontynuować swoją trasę. Innymi słowy, nie ma konieczności powrotu pojazdu do bazy. Omawiana odmiana problemu wielu komiwojażerów pokazana została na rysunku 4.8, gdzie przy krawędziach grafu podano aktualną wykorzystaną ładowność pojazdu, koło oznacza klienta, natomiast trójkąt – punkt uzupełnień ładunku.



Rys. 4.8. Problem wielu komiwojazerów z uzupełnieniami ładunku
Źródło: opracowanie własne.

Aby przedstawić problem układania tras pojazdów z uzupełnieniami ładunku w postaci formalnej, najpierw trzeba przyjąć kilka założeń. Te podstawowe są adekwatne do klasycznej postaci problemu wielu komiwojazerów, czyli każdy z pojazdów odwiedza klienta tylko jeden raz, każdy klient zgłasza nieujemny popyt na produkt, początek tras każdego pojazdu znajduje się w bazie. Ponadto zbiór punktów uzupełnienia ładunku dysponuje (podobnie jak baza) nieograniczoną podażą rozwożonego produktu. Jednakże mogą istnieć ograniczenia co do liczby wizyt pojazdów w takim punkcie w ciągu dnia (czasami ograniczenia dotyczące samego czasu pracy pojazdu zastępują ten limit). Podstawowa różnica między punktem uzupełnienia ładunku a bazą polega na tym, że baza stanowi początek i koniec całej trasy pojazdu w np. danym dniu.

Przedstawiona propozycja modelu (Bard i in., 1998) dla zagadnienia wielu komiwojazerów z uzupełnieniami ładunku opiera się na grafie skierowanym $G(V, A)$, w którym oprócz wierzchołków reprezentujących poszczególne punkty obsługi znajdują się wierzchołki reprezentujące jedną wizytę danego pojazdu w celu uzupełnienia ładunku. Oznacza to, że jeżeli dany punkt uzupełnienia ładunku może być odwiedzony co najwyżej 5 razy, wtedy liczba wierzchołków grafu reprezentujących ten punkt wynosi 5. Dotyczy to również bazy, ponieważ przyjmuje się założenie, że każdy pojazd może swój ładunek uzupełnić także tam w celu kontynuowania swojej trasy (może to wynikać z odpowiedniej bliskości bazy w stosunku do aktualnej pozycji pojazdu w realizowanej trasie). Stąd też liczbę wierzchołków grafu G (z wyjątkiem wierzchołka bazy jako początkowego i zarazem końcowego) określić można następująco:

$$\tilde{n} = n + \sum_{s=0}^S n_s \quad (4.71)$$

Tym samym całkowita liczba wszystkich wierzchołków (uwzględniająca także wierzchołek bazy jako początkowy i końcowy punkt trasy każdego pojazdu) rozszerzonego grafu $G(V, A)$ wynosi $\tilde{n} + 1$.

Dla tak sformułowanych założeń zagadnienia układania tras pojazdów z uzupełnieniami ładunku należy przyjąć następujące oznaczenia:

- n – liczba punktów obsługi – klientów (baza oznaczona jest numerem 0);
- S – liczba punktów, w których można dokonać uzupełnienia ładunku;
- I – zbiór wszystkich klientów, natomiast $I_0 = I \cup \{0\}$;
- F – zbiór wszystkich punktów, w których można uzupełnić ładunek;
- F_s – zbiór wierzchołków dodanych do grafu G odpowiadających potencjalnym wizytom pojazdów w celu uzupełnienia ładunku, gdzie $s \in F \cup \{0\}$; natomiast F_0 oznacza wierzchołek dodany do grafu, reprezentujący wizytę pojazdu w bazie tylko w celu uzupełnienia ładunku;
- c_{ij} – koszt przejazdu pomiędzy punktem reprezentowanym przez wierzchołek i grafu G a punktem reprezentowanym przez wierzchołek j grafu G (koszty przejazdu nie tylko pomiędzy klientami, ale także pomiędzy bazą oraz punktami uzupełnień ładunków);
- t_{ij} – czas przejazdu pomiędzy punktem reprezentowanym przez wierzchołek i grafu G a punktem reprezentowanym przez wierzchołek j grafu G ;
- Q – ładowność pojazdów;
- K – liczba pojazdów o ładowności Q ;
- n_s – limit wizyt w danym punkcie uzupełnień ładunku, gdzie $n_s = |F_s|$;
- \tilde{Q} – maksymalna ilość produktu, jaką dysponuje pojazd, poniżej której można dokonać uzupełnienia ładunku;
- T – maksymalny czas pracy pojazdu;
- q_i – zapotrzebowanie na produkt i -tego klienta;
- t_i – czas obsługi pojazdu w wierzchołku reprezentującym i -tego klienta (czas rozładunku produktu) lub reprezentującym i -ty punkt uzupełnienia ładunku (czas załadunku produktu);
- x_{ij} – zmienna decyzyjna przyjmująca wartość 1 wtedy i tylko wtedy, gdy wierzchołek i poprzedzony jest przez wierzchołek j , natomiast 0, gdy nie jest przezeń poprzedzony;
- τ_j – zmienna decyzyjna oznaczająca moment rozpoczęcia obsługi w wierzchołku j (rozładunku lub załadunku);
- y_j – zmienna decyzyjna oznaczająca wielkość ładunku pozostającą w dyspozycji pojazdu w sytuacji, gdy następnym odwiedzanym przez pojazd punktem jest klient lub punkt uzupełnienia ładunku reprezentowany przez wierzchołek j .

Matematyczny model układania tras pojazdów z uzupełnieniami ładunków przez pojazdy ma postać modelu programowania mieszanego i jest ona następująca (4.72)–(4.89):

$$\min. F(\mathbf{x}) = \sum_{i=0}^{\tilde{n}} \sum_{j=0}^{\tilde{n}} c_{ij} x_{ij} \quad (4.72)$$

$$\sum_{j=0}^{\tilde{n}} x_{ij} = 1 \quad \text{dla } i = 1, 2, \dots, n \quad (4.73)$$

$$\sum_{j=1}^n x_{ij} \leq 1 \quad \text{dla } i \in F_0 \cup F_1 \cup \dots \cup F_s \quad (4.74)$$

$$\sum_{i=0}^{\tilde{n}} x_{ji} - \sum_{i=0}^{\tilde{n}} x_{ij} = 0 \quad \text{dla } j = 0, 1, \dots, \tilde{n} \text{ oraz } i \neq j \quad (4.75)$$

$$\sum_{j=1}^n x_{0j} \leq K \quad (4.76)$$

$$\tau_j \geq \tau_i + t_{ij}x_{ij} - T_{ij}(1-x_{ij}) \quad \text{dla } i \in I \cup F_0 \cup \dots \cup F_s; \\ j \in I \cup F_0 \cup \dots \cup F_s, i \neq j \quad (4.77)$$

$$\tau_0 \geq \max_{l \in I} (2t_{0l} + t_l) \quad (4.78)$$

$$\tau_0 \leq T \quad (4.79)$$

$$\tau_j \geq t_{0j} \quad \text{dla } j = 1, 2, \dots, n \quad (4.80)$$

$$\tau_j \leq T - (t_j + t_{j0}) \quad \text{dla } j = 1, 2, \dots, n \quad (4.81)$$

$$\tau_j \geq \min_{l \in I} (t_{0l} + t_l + t_{lj}) \quad \text{dla } j \in F_0 \cup F_1 \cup \dots \cup F_s \quad (4.82)$$

$$\tau_j \leq \max_{l \in I} (T - t_j - t_{jl} - t_l - t_{l0}) \quad \text{dla } j \in F_0 \cup F_1 \cup \dots \cup F_s \quad (4.83)$$

$$y_j \leq y_i - \bar{q}_i x_{ij} + \bar{Q}_i (1 - x_{ij}) \quad \text{dla } i \in I \cup F_0 \cup \dots \cup F_s; \\ j \in I \cup F_0 \cup \dots \cup F_s, i \neq j \quad (4.84)$$

$$y_j \geq q_j \quad \text{dla } j = 1, 2, \dots, n \quad (4.85)$$

$$y_j \leq Q \quad \text{dla } j = 1, 2, \dots, n \quad (4.86)$$

$$y_j \geq 0 \quad \text{dla } j \in F_0 \cup F_1 \cup \dots \cup F_s \quad (4.87)$$

$$y_j \leq \tilde{Q} \quad \text{dla } j \in F_0 \cup F_1 \cup \dots \cup F_s \quad (4.88)$$

$$x_{ij} = \begin{cases} 0 \\ 1 \end{cases} \quad \text{dla } i, j \in I \cup F_0 \cup \dots \cup F_s \quad (4.89)$$

Funkcja celu w powyższym modelu (4.72) reprezentuje całkowity koszt transportu wszystkich występujących w grafie G połączeń. Zespół ograniczeń wyrażony równościami (4.73) zapewnia, że każdy wierzchołek grafu G ma tylko jeden wierzchołek po sobie następujący w trasie pojazdu, którym może być zarówno klient,

baza, jak i dowolny punkt uzupełnienia ładunku. Jeżeli następującym wierzchołkiem jest baza, należy zauważyć, że indeks j przyjmie odpowiednią wartość w zależności od tego, czy baza będzie oznaczała koniec trasy pojazdu ($j = 0$), czy punkt uzupełnienia ładunku pojazdu ($j \in F_0$). Ograniczenia wyrażone zespołem nierówności (4.74) przedstawiają założenie, że każdy wierzchołek grafu G reprezentujący punkt uzupełniania ładunku pojazdu może mieć co najwyżej jeden następujący po sobie inny wierzchołek. Zbiór równań (4.75) jest charakterystyczny dla każdego problemu układania tras pojazdów i gwarantuje ciągłość trasy każdego pojazdu, a w szczególności oznacza, że liczba wyjazdów z każdego wierzchołka musi być równa liczbie wjazdów do niego. Ograniczenie (4.76) oznacza, że liczba wjazdów pojazdów do bazy (jako punktu końcowego trasy, a nie punktu uzupełnienia ładunku) musi być równa co najwyżej liczbie dostępnych pojazdów K (np. możliwych do wykorzystania w danym dniu).

Nierówności (4.77) zapewniają poprawność momentu obsługi przez pojazd punktu reprezentowanego przez wierzchołek j . Należy zauważyć pojawiającą się w tym miejscu wielkość T_{ij} , która może przyjąć następujące wartości (4.90) w zależności od tego, czy wierzchołek grafu reprezentuje klienta, czy też punkt uzupełnienia ładunku:

$$T_{ij} = \begin{cases} T - t_{0j} - (t_i + t_{i0}) & \text{dla } i \in I \\ T - t_{0j} - \min. (t_i + t_{il} + t_l + t_{l0}) & \text{dla } i, j \in F_0 \cup \dots \cup F_S \end{cases} \quad (4.90)$$

Ograniczenia wyrażone zespołem nierówności (4.78)–(4.81) gwarantują przede wszystkim, że każda trasa nie przekracza założonego czasu pracy dla pojazdów będących w dyspozycji bazy. Ograniczenia (4.78) i (4.79) stanowią odpowiednio dolną i górną granicę dla momentu oznaczającego powrót wszystkich pojazdów do bazy (zmienna decyzyjna τ_0). Dolna granica oznacza maksymalny czas niezbędny do dotarcia do wszystkich klientów, ich obsługi oraz powrotu do bazy. Natomiast górną granicę stanowi maksymalny czas pracy pojazdów. Z kolei ograniczenia (4.80) i (4.81) także prezentują dolną i górną granicę, lecz dla momentu w czasie oznaczającego rozpoczęcie obsługi klienta reprezentowanego przez wierzchołek j (zmienna decyzyjna τ_j). Dolna granica jest wyznaczona przez czas, jaki jest niezbędny na dotarcie pojazdu z bazy do klienta oznaczonego wierzchołkiem j . Natomiast górną granicę stanowi różnica czasu pomiędzy maksymalnym czasem pracy pojazdu a sumą czasów: obsługi j -tego klienta i czasu dotarcia pojazdu od niego do bazy.

Dolne i górne granice dla zmiennej decyzyjnej τ_j reprezentującej moment odwiezienia punktu uzupełnienia ładunku określone zostały w ograniczeniach (4.82) i (4.83). Oznacza to, że moment rozpoczęcia obsługi j -tego klienta nie może nastąpić, zanim nie upłynie minimalna ilość czasu, jaka jest niezbędna na podróż pojazdu z bazy do klienta, jego obsługę oraz podróż do miejsca uzupełnienia ładunku. Jednocześnie górna granica jest determinowana przez najdłuższy czas potrzebny na podróż pojazdu od punktu uzupełnienia ładunku do klienta oraz powrót do bazy.

Ponadto w połączeniu z omówionymi wcześniej ograniczeniami ciągłości (4.76) oraz uwzględniając tę okoliczność, że zmienne decyzyjne τ_j mają charakter narastający, przedmiotowe warunki zapewniają, że powstałe trasy będą cyklami Hamiltona (żaden klient nie zostanie odwiedzony więcej niż jeden raz, a jednocześnie nie będzie stanowił części trasy będącej podcyklem). Tak sformułowane ograniczenia dotyczące czasu pracy pojazdu eliminują konieczność wprowadzania dodatkowych zmiennych decyzyjnych oraz nierówności, które zostały sformułowane dla podstawowej postaci problemu wielu komiwojażerów (4.5).

Kolejne ograniczenia (4.84)–(4.88) odnoszą się do wykorzystanej ładowności pojazdów. Ograniczenia wyrażone zespołem nierówności (4.84) dotyczą obciążenia pojazdu przed odwiedzeniem klienta reprezentowanego przez wierzchołek j grafu. W ograniczeniu tym pojawiają się dwie nowe wielkości: \bar{q}_i oraz \bar{Q}_i . Pierwsza z nich przyjmuje następujące wartości (4.91):

$$\bar{q}_i = \begin{cases} q_i & \text{dla } i \in I \\ -Q & \text{dla } i \in F_0 \cup \dots \cup F_S \end{cases} \quad (4.91)$$

Druga natomiast przyjmuje wartości także w zależności od tego, czy wierzchołek i grafu reprezentuje odwiedzanego przez dany pojazd klienta, czy też punkt uzupełnienia ładunku (4.92):

$$\bar{Q}_i = \begin{cases} Q - q_i & \text{dla } i \in I \\ Q & \text{dla } i \in F_0 \cup \dots \cup F_S \end{cases} \quad (4.92)$$

Ograniczenia (4.85) i (4.86) przedstawiają odpowiednio dolne i górne granice dla zmiennej decyzyjnej y_j oznaczającej w tym wypadku obciążenie pojazdu przed odwiedzeniem wierzchołka j grafu reprezentującego j -tego klienta. Natomiast ostatnie dwa ograniczenia – (4.87) oraz (4.88) – dotyczą także górnych i dolnych granic tej samej zmiennej decyzyjnej, lecz w sytuacji, gdy wierzchołek j grafu reprezentuje punkt, w którym pojazdy uzupełniają ładunek.

4.2.6. Problem wielu komiwojażerów z obsługą okresową

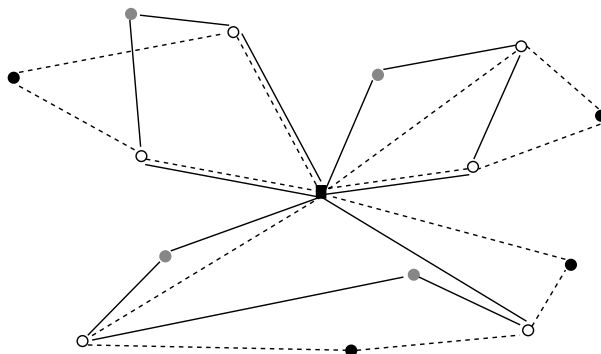
Bardzo istotnym rozszerzeniem klasycznej postaci problemu wielu komiwojażerów, znajdującym często swoje odzwierciedlenie w praktyce gospodarczej, jest zagadnienie układania tras pojazdów z obsługą okresową. Istotą tej modyfikacji jest proces układania tras pojazdów na pewien z góry założony okres. Najczęściej stanowi on krotność dni (np. jeden tydzień). Nie oznacza to, że trzeba układać tyle osobnych zadań transportowych, ile jest dni w rozpatrywanym okresie, ponieważ celem optymalizacji jest minimalizacja kosztów transportu w całym rozpatrywanym okresie.

Podstawowym założeniem zagadnienia układania tras pojazdów z obsługą okresową jest informacja o częstotliwości odwiedzania obsługiwanego przez bazę punktu w przyjętym okresie. Przykładowo, jeżeli rozpatrywanym okresem planowania jest tydzień (pięć dni roboczych), wtedy wśród wszystkich punktów obsługi część będzie obsługiwana jednokrotnie (w jednym z pięciu dni), część dwukrotnie (w dwóch z pięciu dni, np. wtorek i czwartek), a inna część z częstotliwością codzienną (poniedziałek, wtorek, środa, czwartek, piątek). Tym samym oprócz ustalania tras pojazdów na dany dzień powstaje dodatkowy problem decyzyjny: w który dzień lub które dni z zadanego przedziału czasowego dany punkt obsługi ma być odwiedzony przez pojazd przy jednoczesnej znajomości częstotliwości jego obsługi.

Reasumując, można wyróżnić kilka podstawowych cech charakterystycznych dla zagadnień układania tras pojazdów z obsługą okresową (Francis i Smilowitz, 2006), takich jak:

- okres planistyczny (np. miesiąc) jako zbiór kolejno następujących po sobie okresów jednostkowych (np. dni);
- częstotliwość obsługi poszczególnych punktów (klientów) w zadanym okresie planistycznym, co oznacza, że klient w tym okresie może zostać przez pojazd odwiedzony więcej niż jeden raz;
- zbiór alternatywnych harmonogramów obsługi poszczególnych punktów (klientów) w rozpatrywanym okresie planistycznym jako zbiór dopuszczalnych kombinacji okresów jednostkowych.

Na rysunku 4.9 przedstawiono graficzną postać zagadnienia układania tras pojazdów z obsługą okresową, gdzie punkty oznaczone kolorem szarym odwiedzane są dwukrotnie w zadanym okresie planistycznym, punkty oznaczone kolorem czarnym – trzykrotnie, natomiast pozostałe punkty odwiedzane są codziennie. Tym samym trasy oznaczone linią ciągłą pokonywane są przez wykorzystane pojazdy w dwóch z pięciu dni, natomiast w pozostałych dniach pojazdy pokonują trasy oznaczone linią przerywaną.



Rys. 4.9. Problem wielu komiwojażerów z obsługą okresową
Źródło: opracowanie własne.

Przyjmując, że T oznacza okres planistyczny jako zbiór okresów jednostkowych (np. dni), harmonogramem obsługi poszczególnych punktów nazwać należy zbiór konkretnych okresów jednostkowych (np. dni), w których ta obsługa następuje. Innymi słowy, przyporządkowanie obsługiwanego punktu do konkretnego harmonogramu oznacza, że pojazd odwiedzi zadany punkt we wszystkie dni przedmiotowego harmonogramu. Oznaczając przez S zbiór wszystkich harmonogramów, pojedynczy harmonogram obsługi przedstawić można jako wektor \mathbf{a} , którego elementy $a_{s(t)}$ przyjmują następujące wartości: 1 wtedy i tylko wtedy, gdy okres jednostkowy $t \in T$ (np. dzień) należy do danego harmonogramu $s \in S$, natomiast 0, gdy do niego nie należy. Oznaczając przez f_i częstotliwość odwiedzin i -tego punktu przez pojazd w planowanym okresie T , należy wyznaczyć podzbiór potencjalnych harmonogramów S_i spełniających warunek:

$$S_i = \{s \in S : \sum_{t \in T} a_{s(t)} = f_i\} \tag{4.93}$$

Przykład

W tabeli 4.1 zaprezentowany został przykładowy zbiór wszystkich możliwych harmonogramów s przy założeniu, że $T = 5$, natomiast częstotliwość odwiedzania punktu obsługi wynosi 2.

Tabela 4.1. Zbiór harmonogramów dla $T = 5$ i częstotliwości 2

T	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}
t_1	1	1	1	1						
t_2	1				1	1	1			
t_3		1			1			1	1	
t_4			1			1		1		1
t_5				1			1		1	1

Źródło: opracowanie własne.

Należy tutaj zauważyć, że jeśli $|S_i| = 0$ dla któregośkolwiek i -tego punktu obsługi, wtedy problem wielu komiwojazerów z obsługą okresową nie posiada rozwiązań dopuszczalnych. Z kolei jeżeli $|S_i| = 1$ dla każdego i -tego punktu obsługi, oznacza to, że każdy punkt obsługi zawiera tylko jeden możliwy harmonogram wizyt pojazdów. Tym samym problem układania tras pojazdów z obsługą okresową sprowadza się do niezależnych (klasycznych) problemów układania tras pojazdów osobno dla każdego okresu jednostkowego $t \in T$.

Formalna postać zagadnienia układania tras pojazdów z obsługą okresową została sformułowana za pomocą grafu skierowanego $G(V, A)$ reprezentującego zbiór N obsługiwanego punktów oraz bazę $\{0\}$, a także zbiór krawędzi A oznaczających

sieć połączeń pomiędzy punktami obsługi, w tym także bazą. Zanim przedstawimy model programowania matematycznego, należy przyjąć następujące oznaczenia:

- T – okres planistyczny składający się z okresów jednostkowych;
 K – zbiór pojazdów;
 S – zbiór dopuszczalnych harmonogramów odwiedzania punktów obsługi;
 W_i – całkowity popyt (podaż) punktu obsługi w całym okresie planowania T ;
 f_i – częstotliwość (wymagana liczba wizyt) i -tego punktu obsługi w całym okresie planowania T ;
 Q – dopuszczalna ładowność pojazdów;
 L – dopuszczalna długość trasy pojazdu;
 c_{ij} – koszt przejazdu pomiędzy punktem reprezentowanym przez wierzchołek i grafu G a punktem reprezentowanym przez wierzchołek j grafu G (koszty przejazdu pomiędzy klientami, w tym także bazą);
 x_{ij}^{kt} – zmienna decyzyjna przyjmująca wartość 1 wtedy i tylko wtedy, gdy do trasy pojazdu $k \in K$ przyporządkowany został odcinek pomiędzy punktami i oraz j w okresie jednostkowym $t \in T$, natomiast 0, jeżeli nie został on przyporządkowany;
 y_{ik}^s – zmienna decyzyjna przyjmująca wartość 1 wtedy i tylko wtedy, gdy pojazd $k \in K$ obsługuje i -ty wierzchołek reprezentujący punkt obsługi zgodnie z harmonogramem $s \in S$, natomiast 0, gdy go nie obsługuje.

Model programowania matematycznego dla zagadnienia układania tras pojazdów z obsługą okresową został zaproponowany przez N. Christofidesa i J.-E. Beasleya (1984) i ma on następującą postać:

$$\min. F(\mathbf{x}) = \sum_{t \in T} \sum_{(i,j) \in A} \sum_{k \in K} c_{ij} x_{ij}^{kt} \quad (4.94)$$

$$\sum_{s \in S_s} \sum_{k \in K} y_{ik}^s = 1 \quad \text{dla } i \in V \setminus \{0\} \quad (4.95)$$

$$v_i^t - v_j^t - \sum_{k \in K} x_{ij}^{kt} \geq 0 \quad \text{dla } t \in T, i, j \in V \setminus \{0\} \text{ oraz } i \neq j \quad (4.96)$$

$$\sum_{j \in V \setminus \{0\}} x_{ij}^{kt} - \sum_{j \in V \setminus \{0\}} x_{ji}^{kt} = 0 \quad \text{dla } i \in V, k \in K, t \in T \quad (4.97)$$

$$\sum_{k \in K} \sum_{i \in V} x_{ij}^{kt} = v_j^t \quad \text{dla } j \in V \setminus \{0\}, t \in T \quad (4.98)$$

$$\sum_{k \in K} \sum_{i \in V} x_{i0}^{kt} \leq |K| \quad \text{dla } t \in T \quad (4.99)$$

$$\sum_{i,j \in H} x_{ij}^{kt} \leq |H| - 1 \quad \text{dla każdego } H \subseteq V \setminus \{0\}, k \in K, t \in T \quad (4.100)$$

$$\sum_{i \in V \setminus \{0\}} \sum_{j \in V \setminus \{0\}} x_{ij}^{kt} \leq 1 \quad \text{dla } k \in K, t \in T \quad (4.101)$$

$$\sum_{i \in V \setminus \{0\}} \omega_i \sum_{j \in V} x_{ij}^{kt} \leq Q \quad \text{dla } k \in K, t \in T \quad (4.102)$$

$$\sum_{(i,j) \in A} c_{ij} x_{ij}^{kt} \leq L \quad \text{dla } k \in K, t \in T \quad (4.103)$$

$$x_{ij}^{kt} = \begin{cases} 0 \\ 1 \end{cases} \quad \text{dla } (i, j) \in A, k \in K, t \in T \quad (4.104)$$

Funkcja celu (4.94) przedstawia minimalizację całkowitych kosztów wszystkich tras pojazdów, jakie zostały zrealizowane w całym okresie planowania T . Zespół ograniczeń wyrażony równaniami (4.95) zapewnia dopuszczalność wybieranych harmonogramów wizyt pojazdów w poszczególnych punktach obsługi. Ograniczenia reprezentowane przez zbiór nierówności (4.96) warunkują występowanie połączeń tylko pomiędzy wierzchołkami, które reprezentują punkty obsługi przyporządkowane do obsługi przez pojazdy w danym okresie jednostkowym $t \in T$. Występujące w powyższych ograniczeniach są wielkościami zagregowanymi, które są równe (4.105):

$$v_i^t = \sum_{s \in S_i} a_{st} \sum_{k \in K} y_{ik}^s \quad \text{dla każdego } t \in T \text{ oraz } i \in V \setminus \{0\} \quad (4.105)$$

Ograniczenia w postaci nierówności (4.97) przedstawiają podstawowe założenie układania tras pojazdów, czyli ciągłość tworzonych tras. Z kolei dwie grupy ograniczeń (4.98) i (4.99) zapewniają, że punkty obsługi reprezentowane przez wierzchołki grafu należą do tras występujących w okresach jednostkowych $t \in T$ odpowiadających przypisanym im harmonogramom. Nierówności (4.100) eliminują trasy, które nie stanowią cykli Hamiltona. Zespół ograniczeń w postaci nierówności (4.101) pozwala wykorzystać każdy pojazd tylko jeden raz w ciągu okresu jednostkowego $t \in T$. Wreszcie dwie grupy ograniczeń (4.102) i (4.103) stanowią o fizycznych możliwościach pojazdów, zapewniając odpowiednio, że ani ładowność, ani czas pracy pojazdu (wyrażony tutaj maksymalną długością trasy) nie zostaną przekroczone. W wypadku ograniczeń (4.102) przyjęto założenie, że całkowity popyt (podaż) W_i i -tego punktu obsługi w całym okresie planistycznym T zostaje podzielony na f_i równych części ω_i :

$$\omega_i = \frac{W_i}{f_i} \quad \text{dla } i \in V \setminus \{0\} \quad (4.106)$$

Przedstawiony powyżej za pomocą funkcji celu (4.94) oraz zespołu ograniczeń (4.95)–(4.104) model matematyczny reprezentuje:

- problem alokacji punktów obsługi reprezentowanych przez wierzchołki grafu G do odpowiedniego harmonogramu w taki sposób, że każdy punkt obsługi jest odwiedzany przez pojazd z żadaną częstotliwością w planowanym okresie T , a jednocześnie
- problem konstrukcji odpowiednich tras w jednostkowym okresie $t \in T$,

i ma na celu minimalizację kosztu odwiedzenia wszystkich punktów obsługi przez wszystkie wykorzystane do tego celu pojazdy będące w dyspozycji bazy.

Z przedstawionej definicji problemu układania tras pojazdów z obsługą okresową, a także z matematycznego modelu tego zagadnienia wynika, że w przeciwieństwie do klasycznego problemu wielu komiwojazerów występują tu nie dwa, lecz trzy odrębne problemy optymalizacyjne, które powinny być rozpatrywane równolegle. Oprócz charakterystycznych dla klasycznego zadania układania tras pojazdów: problemu przydziału punktów obsługi do pojazdów oraz kolejności ich odwiedzin, trzecim problemem jest wybór dopuszczalnego harmonogramu obsługi punktów w okresie planistycznym T (wskazanie konkretnych okresów jednostkowych $t \in T$, w których ta obsługa ma nastąpić).

Na koniec rozważań dotyczących układania tras pojazdów z obsługą okresową warto jeszcze zwrócić szczególną uwagę na pewną jego odmianę, którą można określić jako problem wielu komiwojazerów z obsługą okresową i wyborem częstotliwości serwisu (*Period Vehicle Routing Problem with Service Choice*) (Francis i in., 2006). W tej modyfikacji zakłada się, że każdy punkt obsługi nie charakteryzuje się dokładną częstotliwością odwiedzin w okresie planistycznym T , lecz częstotliwość ta jest minimalną liczbą wizyt, jakie muszą odbyć przypisane pojazdy. Tym samym liczba dopuszczalnych harmonogramów dla każdego punktu obsługi znacznie wzrasta. Ponadto postać funkcji celu może uwzględniać także pewne korzyści wynikające z możliwości zwiększenia częstotliwości obsługi. Korzyści te mogą uwzględniać np. niższy poziom zapasów utrzymywanych w poszczególnych punktach obsługi, a wynikający z częstszych dostaw. Czasami także klienci są skłonni zapłacić dodatkowo za częstszą obsługę, co również może stanowić korzyść dla optymalizowanego systemu transportowego (np. częstszy wywóz śmieci z wybranych lokalizacji). W takiej sytuacji przedmiotowe dodatkowe korzyści wynikające z modyfikacji częstotliwości obsługi poszczególnych punktów pozytywnie wpływają na wartość funkcji celu (4.94), pomniejszając całkowity koszt wszystkich zrealizowanych tras.

Oprócz wskazanej powyżej modyfikacji zagadnienia wielu komiwojazerów z obsługą okresową w literaturze przedmiotu znaleźć można także inne warianty, do których należą między innymi problem uwzględniający równomierność obciążenia tras pomiędzy pojazdami czy problem wielu komiwojazerów z obsługą okresową (Gulczyński i in., 2011), uwzględniającą wiele okresów planistycznych.

4.2.7. Stochastyczny problem wielu komiwojazerów

Deterministyczny charakter zagadnienia wielu komiwojazerów bardzo często nie odpowiada rzeczywistym problemom optymalizacji układania tras pojazdów występującym w praktyce. Wartości jednego lub kilku jego parametrów mogą nie być znane *a priori*, a tym samym mogą przyjmować wartości losowe. Wśród parametrów, które najczęściej przyjmować mogą postać zmiennej losowej, wymienia się (Gendreau i in., 1996):

- obecność punktu obsługi w zbiorze ustalonych tras pokonywanych przez poszczególne pojazdy;
- popyt lub podaż punktu obsługi;
- czas pokonania odcinka trasy przez pojazd pomiędzy przydzielonymi do niego punktami obsługi;
- czas obsługi przydzielonego do pojazdu punktu (np. czas załadunku lub wyładunku towaru).

Obecność zmiennej lub zmiennych losowych w problemie układania tras pojazdów oznacza, że w trakcie pokonywania przez pojazdy zaplanowanych wcześniej tras może dojść do złamania niektórych warunków. Jako przykład posłużyć może sytuacja, w której np. nie znając wcześniej popytu lub podaży punktu obsługi, pojazd zmuszony będzie wcześniej zakończyć realizację swojej trasy, ponieważ jego ładowność może zostać przekroczona. Oznacza to, że w wypadku losowości wybranych parametrów problemu układania tras pojazdów także niektóre ograniczenia uwzględniające te parametry realizowane będą z pewnym prawdopodobieństwem. Tym samym w stochastycznych problemach układania tras pojazdów, w których postać zmiennej losowej przyjmują popyty lub podaże obsługiwanych przez bazę punktów lub czasy przejazdów pomiędzy nimi, proponowane są modele, gdzie prawdopodobieństwo przekroczenia ograniczenia o założoną wartość nie jest większe niż ustalony wcześniej próg. Dzieje się tak na przykład wtedy, gdy pojazd, pokonując zaplanowaną trasę w czasie dłuższym niż założony, obciążony zostaje pewną karą, która jest proporcjonalna do wartości przekroczenia założonego ograniczenia dotyczącego maksymalnego czasu trwania tej trasy.

Dla tej konkretnej sytuacji zaproponowany został model programowania matematycznego o następującej postaci (Laporte i in., 1992; Laporte i Louveaux, 1997):

$$\min. F(\mathbf{x}) = \sum_{i=0}^n \sum_{j=0}^n c_{ij} x_{ij} \quad (4.107)$$

$$\sum_{j=1}^n x_{0j} = 2m \quad (4.108)$$

$$\sum_{i=1}^n x_{ik} + \sum_{j=1}^n x_{jk} = 2 \quad \text{dla } k = 1, \dots, n \quad (4.109)$$

$$\text{trasy niedopuszczalne} \quad (4.110)$$

$$x_{ij} \in [0, 1] \text{ dla } i, j = 1, \dots, n \quad (4.111)$$

$$x_{0j} \in [0, 1, 2] \text{ dla } j = 1, \dots, n \quad (4.112)$$

W swojej ogólnej postaci model określony równaniami (4.107)–(4.112) przedstawia podstawowy problem wielu komiwojazerów. Jednakże na uwagę zasługuje tu zbiór ograniczeń (4.110) określonych przez autorów jako trasy niedopuszczalne. W tej grupie znajdują się standardowe ograniczenia, które mają nie dopuścić do powstawania tras niebędących cyklami Hamiltona. Autorzy modelu proponują także umieszczenie tu ograniczeń blokujących powstawanie tras o czasie trwania dłuższym niż założony z prawdopodobieństwem większym niż ustalony próg, które przy przyjęciu oznaczeń:

- B – wielkość przekroczenia dopuszczalnego czasu trwania trasy pojazdu;
- $t_{k,k+1}$ – czas przejazdu pojazdu pomiędzy dwoma sąsiednimi punktami obsługi występującymi w rozpatrywanej trasie;
- τ_k – czas obsługi pojazdu w k -tym punkcie obsługi występującym w rozpatrywanej trasie;
- α – przyjęty maksymalny próg dotyczący dopuszczalnego prawdopodobieństwa przekroczenia założonego opóźnienia w realizacji trasy;

wyrażone zostało następującą formułą:

$$P\left\{\sum_{k=0}^u (t_{k,k+1} + \tau_k) > B\right\} > \alpha \quad (4.113)$$

Ograniczenie eliminujące taką trasę składającą się z punktów: $\{0, 1, 2, \dots, k, k+1, \dots, u, 0\}$ ma postać:

$$\sum_{k=0}^u x_{k,k+1} \leq u \quad (4.114)$$

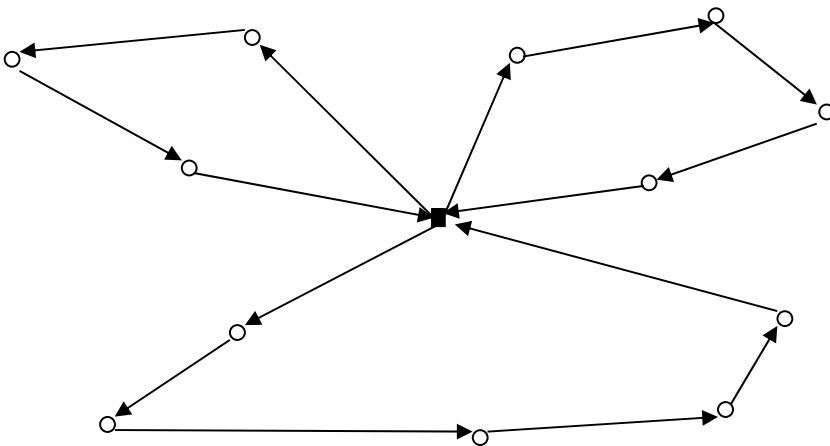
Takie podejście pozwala na przekształcenie stochastycznego problemu układania tras pojazdów w wersję deterministyczną. Innym wariantem stosowanym w wypadku stochastycznych problemów wielu komiwojazerów jest uwzględnienie w funkcji celu oczekiwanego kosztu związanego z przekroczeniem nałożonych ograniczeń. W tej sytuacji rozwiązanie uzyskuje się w dwóch etapach. W pierwszym etapie generowane jest rozwiązanie, które nie uwzględnia faktycznej realizacji zmiennej losowej. Natomiast w drugim etapie dokonywana jest korekta wcześniej uzyskanego rozwiązania, która uwzględnia wiedzę o wartościach, jakie przyjęły zmienne losowe (*Stochastic Program with Recourse*). Rozwiązanie generowane w pierwszym etapie, oprócz kosztu tras bez znajomości realizacji zmiennych losowych, minimalizuje także oczekiwany koszt czynności podjętych w drugim etapie. Choć drugie podejście

do rozwiązywania stochastycznych zadań układania tras pojazdów jest trudniejsze w zastosowaniu, to jednak uważa się, że postać funkcji celu jest adekwatna do charakteru problemu (4.115) (Laporte i Louveaux, 1998):

$$\min. F(\mathbf{x}) = \sum_{i < j} c_{ij} x_{ij} + Q(\mathbf{x}) \quad (4.115)$$

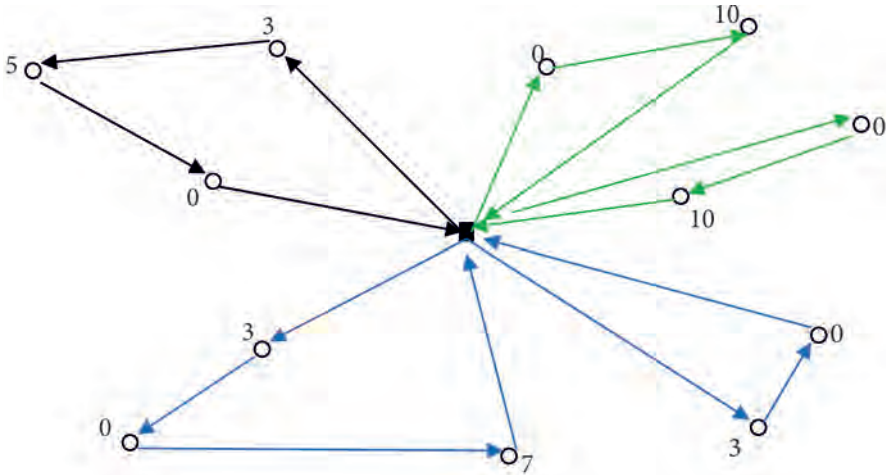
gdzie $Q(x)$ oznacza oczekiwany koszt czynności podjętych w wyniku posiadanej wiedzy o realizacji zmiennych losowych.

Ze względu na wymienione na początku podrozdziału parametry, które zazwyczaj przyjmują postać zmiennych losowych, w literaturze przedmiotu najczęściej rozpatrywany jest stochastyczny problem układania tras pojazdów z losowymi popytami (bądź podażami w wypadku zagadnień zwózki towarów do bazy). Postać zmiennej losowej przyjmować może także obecność punktu obsługi w zbiorze obsługiwanych przez bazę punktów. D. Bertsimas (1992) proponuje dwie możliwe strategie przy rozwiązywaniu stochastycznego problemu wielu komiwojazerów, w którym i -ty punkt obsługi występuje w zbiorze obsługiwanych przez bazę punktów z zadaniem prawdopodobieństwem p_i . Tym samym prawdopodobieństwo jego absencji wynosi $1 - p_i$. Absencja punktu obsługi w zbiorze może być także wynikiem tego, że dany punkt nie wymaga obsługi, ponieważ np. popyt zgłoszony przez niego do bazy równy jest 0. W pierwszym etapie można skonstruować według znanych algorytmów trasy *a priori*, czyli trasy pojazdów uwzględniające wszystkie możliwe do odwiedzenia punkty obsługi. Na rysunku 4.10a przedstawiono przykładowy układ tras *a priori* trzech pojazdów – każdy o ładowności 10 jednostek.



Rys. 4.10a. Układ tras pojazdów *a priori*
Źródło: opracowanie własne.

Drugi etap uzależniony jest od momentu, kiedy decydent uzyskuje informację o ewentualnej absencji punktu obsługi. Jeżeli informacja o realizacji zmiennej losowej jest znana w chwili przybycia pojazdu do obsługiwanego punktu, proponuje się realizację strategii polegającej na odwiedzaniu przez pojazd wszystkich punktów ustalonych w pierwszym etapie, lecz obsłudze tylko tych, które tego wymagają (rys. 4.10b).



Rys. 4.10b. Układ tras pojazdów według strategii pierwszej

Źródło: opracowanie własne.

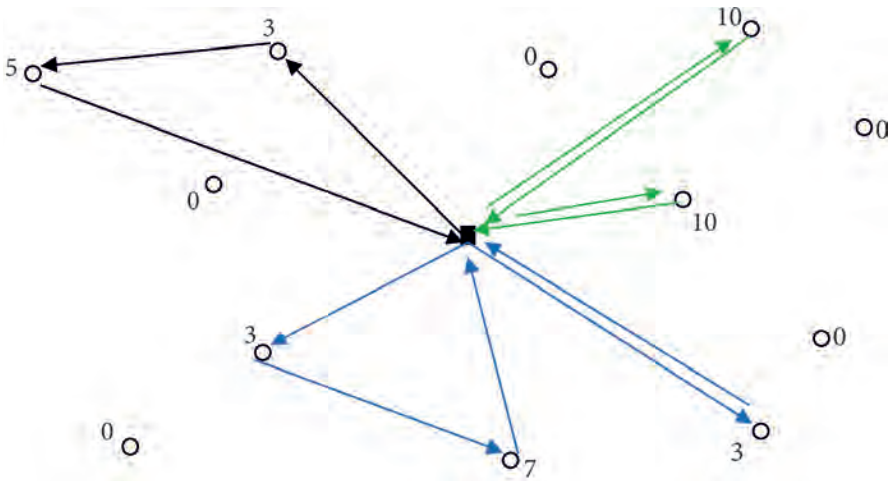
Oznacza to, że ze względu na wielkość popytu zgłaszaną przez obsługiwany punkt konieczny będzie powrót pojazdu do bazy, a następnie wznowienie obsługi dalszej części trasy. Taka sytuacja dla jednego pojazdu może powtórzyć się wielokrotnie.

W drugiej strategii przyjmuje się założenie, że ewentualna absencja punktów obsługi jest znana, zanim pojazd rozpocznie swoją trasę. To mogłoby oznaczać, że w takim wypadku trasy powinny zostać ustalone na nowo na podstawie aktualnych danych. Jednakże w praktyce czasami jest to niemożliwe ze względu na brak czasu lub z powodów organizacyjnych (chęć zachowania regularności tras). Tym samym w tej strategii zakłada się, że punkty są po prostu pomijane (rys. 4.10c).

Postać funkcji celu określonej wzorem (4.115) jest wprost uzależniona od rodzaju stochastycznego układania tras dostaw, jak również od czynności podjętych w drugim etapie, kiedy znane są realizacje zmiennych losowych. Jakkolwiek pierwsza jej składowa jest wprost całkowitym dystansem trasy *a priori*, tak funkcja $Q(x)$ reprezentować może oczekiwaną dodatkową długość trasy, jaką muszą pokonać pojazdy w razie niepowodzenia polegającego na konieczności powrotu do bazy w wyniku poznania realizacji zmiennej losowej dotyczącej popytu (Christiansen i Lysgaard, 2007). Ponieważ powrót do bazy skutkuje pokonaniem dodatkowej odległości między i -tym punktem obsługi a bazą, stąd oczekiwany dodatkowy

dystans będzie iloczynem dodatkowego dystansu do pokonania między i -tym punktem a bazą oraz prawdopodobieństwa wystąpienia niepowodzenia (konieczności powrotu pojazdu do bazy) (4.116)²:

$$Q(i) = 2d_{0i} \text{ FAIL}(i) \tag{4.116}$$



Rys. 4.10c. Układ tras pojazdów według strategii drugiej
Źródło: opracowanie własne.

Realizując powyższą koncepcję, można przedstawić model programowania matematycznego oparty na przedstawionym wcześniej modelu podstawowego zagadnienia układania tras pojazdów z ograniczeniami zasobowymi (4.23)–(4.26) (Gauvin i in., 2014):

$$\min. \sum_{p \in P} \hat{c}_p \lambda_p \tag{4.117}$$

$$\sum_{p \in P} \alpha_{ip} \lambda_p = 1 \quad \text{dla } i \in V \setminus \{0\} \tag{4.118}$$

$$\sum_{p \in P} \lambda_p \geq K \tag{4.119}$$

$$\lambda_p \in [0, 1] \quad \text{dla } p = P \tag{4.120}$$

gdzie:

- V – zbiór wszystkich punktów obsługi oraz baza;
- P – zbiór wszystkich możliwych dopuszczalnych tras: $P = \{P_1, P_2, \dots, P_s\}$,
gdzie s oznacza liczebność zbioru P : $s = |P|$;

2 Autorzy prawdopodobieństwo to nazywają *fail*.

- \hat{c}_p – całkowity oczekiwany koszt trasy p ;
- α_{ip} – parametr przyjmujący wartość 1 wtedy i tylko wtedy, gdy trasa $p \in P$ uwzględnia i -ty punkt, oraz 0, gdy go nie uwzględnia;
- λ_p – zmienna decyzyjna przyjmująca wartość 1 wtedy i tylko wtedy, gdy $p \in P$ zostanie wybrana;
- Q – ładowność pojazdów;
- K – dolna granica całkowitej liczby pojazdów niezbędnych do odwiedzenia wszystkich punktów obsługi.

Ograniczenia (4.118) zapewniają, że każdy punkt obsługi umieszczony jest tylko w jednej trasie. Natomiast ograniczenie wyrażone nierównością (4.119) gwarantuje, że uzyskane rozwiązanie wykorzysta co najmniej K pojazdów. Liczbę pojazdów niezbędnych do obsługi wszystkich punktów otrzymać można ze wzoru:

$$K = \left\lceil \frac{\sum_{p \in P} E[\xi_i]}{Q} \right\rceil \quad (4.121)$$

gdzie ξ_i jest zmienną losową popytu i -tego punktu obsługi o wartości oczekiwanej $E[\xi_i]$ oraz wariancji $\text{Var}[\xi_i]$.

W modelu tym szczególnej uwagi wymaga sposób policzenia oczekiwanego całkowitego kosztu trasy pojazdów. Dla każdej trasy pojazdu $\{i_0, i_1, i_2, \dots, i_{p-1}, i_p\}$ aktualna skumulowana wielkość popytu w danym h -tym punkcie trasy i_h jest zmienną losową $\Psi(E[i_h], \text{Var}[i_h])$, której wartość oczekiwana $E[i_h]$ oraz wariancja $\text{Var}[i_h]$ są równe odpowiednio:

$$E[i_h] = \sum_{k=1}^h E[\xi_{i_k}] \quad (4.122)$$

$$\text{Var}[i_h] = \sum_{k=1}^h \text{Var}[\xi_{i_k}] \quad (4.123)$$

Oczekiwany koszt EFC (*Efficient Fail Cost*) związany z niepowodzeniem obsługi całej trasy przez pojazd, ponieważ w h -tym punkcie realizacja zmiennej losowej zmusza do powrotu do bazy, a tym samym poniesienia dodatkowego kosztu w związku z wydłużeniem się trasy, wynosi:

$$\text{EFC}[i_h] = 2c_{0i_h} \sum_{u=1}^{\infty} (P\{\Psi(E[i_{h-1}], \text{Var}[i_{h-1}]) \leq uQ\} - P\{\Psi(E[i_h], \text{Var}[i_h]) \leq uQ\}) \quad (4.124)$$

W powyższym wzorze określone jest prawdopodobieństwo wystąpienia kolejnego (u -tego) niepowodzenia w obsłudze całej trasy przez pojazd za jednym razem. Zgodnie z definicją podaną przez Ch.H. Christiansena i J. Lysgaarda (2007, s. 776) dla danej wartości całkowitej dodatniej u oraz częściowo pokonanej trasy

przez pojazd $\{i_0, i_1, i_2, \dots, i_h, i_{h-1}\}$ konieczność u -tego powrotu do bazy następuje wtedy, gdy aktualna skumulowana wartość popytu w h -tym punkcie obsługi przekracza dopuszczalną wielkość uQ , natomiast nie przekracza jej w poprzednim na trasie punkcie obsługi ($h - 1$). Oznacza to, że pojazd może wielokrotnie wracać do bazy, zanim obsłuży wszystkie przyporządkowane do niego punkty obsługi. Tym samym całkowity oczekiwany koszt trasy pojazdu jest sumą deterministycznego kosztu trasy oraz wartości oczekiwanej EFC:

$$\hat{c} = \sum_{h=1}^{p-1} (c_{i_h i_{h+1}} + \text{EFC}_{i_{h+1}}(E[i_{h+1}], \text{Var}[i_{h+1}])) \quad (4.125)$$

Obok stochastycznego problemu układania tras pojazdów z losowymi popytami obsługiwanymi punktów drugim równie często podejmowanym zagadnieniem jest to, które zakłada losowość czasu przejazdu pojazdu pomiędzy punktami. W problemach wielu komiwojażerów, w których określony jest czas podróży pomiędzy i -tym a j -tym punktem obsługi, bardzo często występują bowiem również ograniczenia dotyczące dopuszczalnego przedziału czasowego obsługi poszczególnych punktów (okna czasowe). W sytuacji, w której dopuszcza się obsługę pojazdu przed założonym przedziałem czasowym (wcześniej) lub po jego upływie (później), zazwyczaj na przewoźnika nakładana jest kara. Tym samym losowość czasu przejazdu pojazdu pomiędzy dowolnymi obsługiwanymi punktami wpływa w sposób bezpośredni nie tylko na koszt pokonywanej trasy, ale także na dodatkowe koszty związane z obsługą klienta poza wyznaczonym przedziałem czasowym, a niekiedy także na dodatkowe koszty związane z dłuższą niż zakładana pracą kierowcy pojazdu (np. przekroczenie dziennego czasu pracy, co może generować koszty wynikające z konieczności zapłaty za nadgodziny).

Podobnie jak w wypadku losowości popytu, tak i w sytuacji losowości czasu przejazdu pojazdów budowa modelu programowania matematycznego może być oparta na deterministycznym problemie wielu komiwojażerów z oknami czasowymi, natomiast jego istota sprowadza się do konstrukcji odpowiedniej funkcji celu. Zaproponowana przez D. Tasa i in. (2014) postać funkcji celu w tym zakresie jest następująca:

$$F(\mathbf{X}) = \sum_{k \in K} \left[\rho \frac{1}{c_1} \left(c_d \sum_{j \in N} D_{jk}(\mathbf{x}) + c_e \sum_{j \in N} E_{jk}(\mathbf{x}) \right) + (1 - \rho) \frac{1}{c_2} \left(c_t \sum_{i \in N} \sum_{j \in N} d_{ij} x_{ijk} + c_f \sum_{j \in \{N \setminus 0\}} x_{0jk} + c_0 O_k(\mathbf{x}) \right) \right] \quad (4.126)$$

Powyższa funkcja celu reprezentuje całkowity ważony koszt transportu oraz obsługi w punktach odwiedzanych przez pojazdy. Oznaczenia przyjęte w tej formule są następujące:

- x_{ijk} – zmienna decyzyjna przyjmująca wartość 1 wtedy i tylko wtedy, gdy trasa pojazdu k uwzględnia odcinek pomiędzy i -tym a j -tym punktem obsługi, oraz 0, gdy go nie uwzględnia;
- \mathbf{x} – wektor reprezentujący trasę k -tego pojazdu;
- d_{jk} – odległość pomiędzy i -tym a j -tym punktem obsługi;
- $D_{jk}(\mathbf{x})$ – wartość oczekiwana opóźnienia w j -tym punkcie obsługi, który jest obsługiwany przez k -ty pojazd;
- $E_{jk}(\mathbf{x})$ – wartość oczekiwana czasu, mówiąca o tym, o ile wcześniej w j -tym punkcie obsługi pojawił się k -ty pojazd;
- $O_k(\mathbf{x})$ – wartość oczekiwana czasu, o jaki przekroczony został dopuszczalny limit czasu pracy k -tego pojazdu;
- c_d – koszt kary nakładanej na pojazd za każdą jednostkę czasu, jaka upłynęła po ustalonym przedziale czasowym, gdy pojazd przybył za późno do obsługiwanego punktu;
- c_e – koszt kary nakładanej na pojazd za każdą jednostkę czasu, która mówi o tym, o ile wcześniej w punkcie obsługi pojawił się pojazd, niż zakładany przedział czasowy obsługi tego punktu;
- c_t – koszt jednostki odległości przejazdu pojazdu pomiędzy punktami obsługi wchodzącymi w skład jego trasy;
- c_f – koszt obsługi pojazdu;
- c_o – koszt jednostkowy nadgodzin;
- C_1, C_2 – parametry skalujące dwa główne komponenty funkcji celu (koszt transportu i koszt obsługi)³.

4.2.8. Dynamiczny problem wielu komiwojażerów

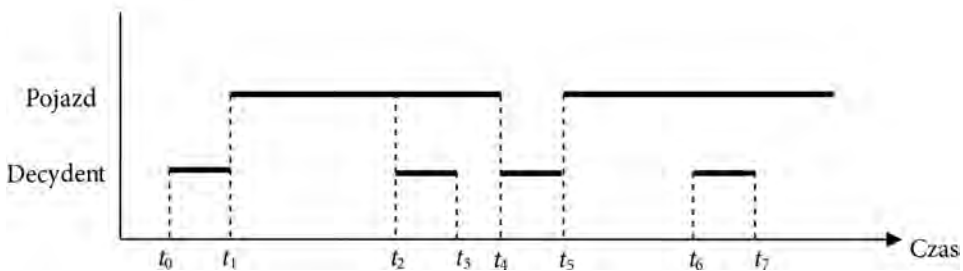
Dotychczas omawiane problemy miały charakter statyczny (niezmienny), czyli wszystkie dane niezbędne do sformułowania i rozwiązania zadania były znane decydentowi. Dynamiczny rozwój technologii informacyjnych (w tym informatycznych) oraz komunikacyjnych nakazuje nieco inaczej spojrzeć na zagadnienia układania tras pojazdów. Wśród technologii, które nie tylko są pomocne, ale wręcz determinują rozwój zagadnień optymalizacji procesów transportowych, wyróżnić można chociażby: systemy GPS, systemy informacji geograficznej czy systemy bezprzewodowego przesyłania danych GPRS. Wykorzystanie i rozbudowywanie tych systemów, a także możliwość przetwarzania informacji w czasie rzeczywistym stanowią pierwszy powód, dla którego dynamiczne zagadnienia układania tras pojazdów zyskały istotne miejsce w rodzinie problemów wielu komiwojażerów. Drugi powód coraz szerszego zainteresowania tymi problemami ma charakter typowo

3 Wartości parametrów C_1 i C_2 nie są ustalane arbitralnie, lecz obliczane według odpowiednich formuł, które znaleźć można w pracy (Tas i in., 2013, s. 216).

logistyczny. Ciągła potrzeba coraz szybszego i częstszego podejmowania decyzji o charakterze operacyjnym wymusza na decydencie dokonywanie ciągłych korekt w zbiorze posiadanych przez niego informacji o optymalizowanym problemie.

Jeśli chodzi o dynamiczny problem układania tras pojazdów, należy zwrócić uwagę na jego podstawowe właściwości świadczące o dynamicznym charakterze optymalizowanego problemu. Pierwsza z nich dotyczy tego, że dane wejściowe do optymalizowanego problemu decyzyjnego są zależne od czasu. W szczególności mogą być one ciągle aktualizowane w czasie realizacji ustalonego wcześniej rozwiązania. Druga z właściwości mówi, że nowy układ tras pojazdów, który mamy znaleźć w wyniku aktualizacji danych, musi zostać wygenerowany jednocześnie w czasie aktualizacji tych danych (na bieżąco). Oznacza to, że w przeciwieństwie do stochastycznych problemów układania tras pojazdów nie mamy tu do czynienia z konstrukcją rozwiązania *a priori*, a jedynie z pewnymi strategiami dokonywania korekty istniejących rozwiązań. Ponadto dynamiczny charakter problemów układania tras pojazdów nie dotyczy sytuacji, gdy dane wejściowe są różne, ale zależne od czasu, co oznacza, że wcześniej znany już jest sposób, w jaki te informacje zmieniają się w czasie. Taka właściwość oznaczałaby, że wcześniej można uzyskać kilka scenariuszy rozwiązań i w zależności od momentu wybrać właściwy bez potrzeby przeprowadzenia ponownego procesu optymalizacji.

Wykres zamieszczony na rysunku 4.11 ilustruje na osi czasu proces podejmowania decyzji oraz komunikowania się decydenta z pojazdem w czasie, kiedy realizowany jest już plan jego trasy.



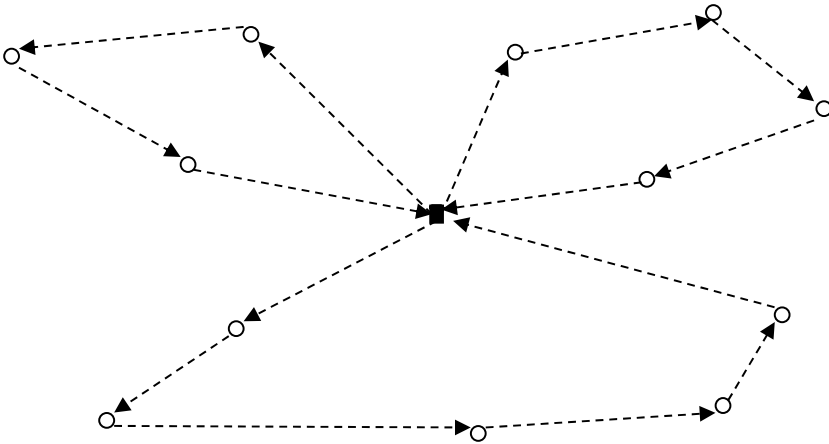
Rys. 4.11. Proces podejmowania decyzji w czasie pracy pojazdu
Źródło: opracowanie własne na podstawie (Gendreau i in., 2013).

W momencie t_0 pojazd zgłasza swoją gotowość do rozpoczęcia pracy. W przedziale czasu (t_0, t_1) decydent podejmuje decyzję dotyczącą przebiegu trasy pojazdu, którą pojazd rozpoczyna w momencie t_1 . W momencie t_2 pojazd zawiadamia decydenta o dotarciu do pierwszego klienta i rozpoczęciu jego obsługi, która trwać będzie w okresie (t_2, t_4) . Równolegle decydent w okresie (t_2, t_3) dokonuje aktualizacji postępu realizacji trasy przez pojazd. W momencie t_5 następuje podjęcie i przekazanie pojazdowi decyzji o wyruszeniu na kolejny odcinek trasy do drugiego

klienta. W czasie pokonywania przez pojazd drogi do drugiego klienta, w momencie t_6 do decydenta napływa informacja o konieczności odwiedzenia i obsługi niezaplanowanego wcześniej nowego klienta. W okresie (t_6, t_7) następuje modyfikacja wcześniej zaplanowanej trasy pojazdu. Po dotarciu przez pojazd do klienta drugiego oraz po dokonaniu jego obsługi kierowcy znów zostanie przekazana przez decydenta informacja o kolejnym punkcie docelowym.

Najczęściej spotykaną sytuacją świadczącą o dynamizmie problemu układania tras pojazdów jest wystąpienie żądania obsłużenia punktu, którego obsługa nie była wcześniej zaplanowana. Szczególnym tego przypadkiem może być wystąpienie dodatniej wielkości popytu (podaży). Do innych parametrów dynamicznego problemu wielu komiwojażerów można zaliczyć (Pillac i in., 2013):

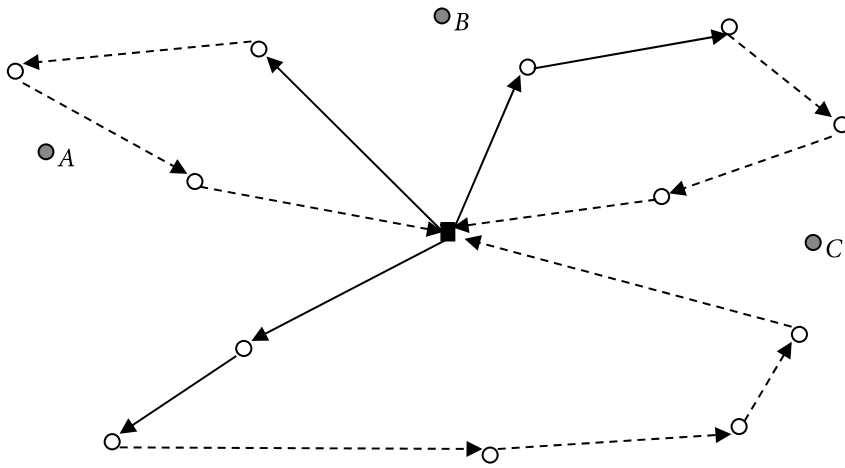
- czas przejazdu pomiędzy obsługiwanymi punktami;
- czas obsługi punktu przez pojazd;
- dostępność pojazdu warunkowana np. możliwością wystąpienia awarii podczas pokonywania zadanej trasy.



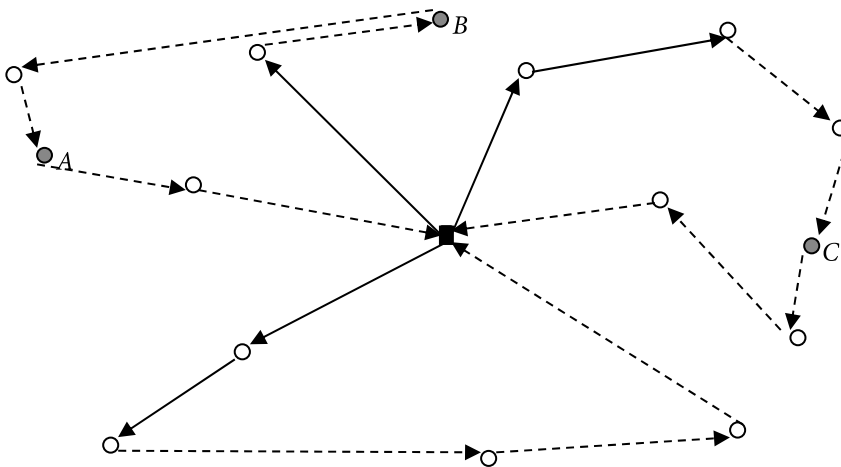
Rys. 4.12a. Wyjściowy (planowany) układ tras pojazdów

Źródło: opracowanie własne.

Na rysunkach 4.12a–4.12c pokazano przypadki i konsekwencje dla pojazdu wystąpienia konieczności obsługi dodatkowych punktów w czasie, gdy pojazd ten pokonuje już swoją trasę. Na pierwszym z nich (rys. 4.12a) przedstawiony został układ tras trzech pojazdów w momencie, gdy pojazdy te opuszczają bazę. Kolejny (rys. 4.12b) ilustruje sytuację, w której do decydenta napływa informacja o konieczności odwiedzenia trzech nowych klientów (A, B i C). Jednocześnie każdy z pojazdów ma już za sobą część pokonanej trasy. Jeden z nich pokonał odcinek do klienta pierwszego lub jest w trakcie jego pokonywania. Z kolei pozostałe dwa pojazdy mają za sobą drugi odcinek do zaplanowanego na początku drugiego klienta lub są w trakcie jego pokonywania.



Rys. 4.12b. Trasy w czasie realizacji
Źródło: opracowanie własne.



Rys. 4.12c. Trasy pojazdów po dokonanej modyfikacji
Źródło: opracowanie własne.

Konieczność uwzględnienia nowych zgłoszeń do obsługi przez pojazdy sprawiła, że trasy pojazdów uległy modyfikacji, co zostało pokazane na rysunku 4.12c. Klient A w najmniejszym stopniu zmodyfikował trasę pojazdu. Można powiedzieć, że znalazł się w bezpośrednim sąsiedztwie planowanej trasy pojazdu (w pobliżu punktów 2 i 3), co tylko nieznacznie zwiększyło długość trasy pojazdu. Klient B z kolei znajduje się w bezpośrednim sąsiedztwie klienta 1 z trasy drugiego pojazdu. Zapewne zostałyby to uwzględnione przy wcześniejszym planowaniu jego trasy. Jednakże informacja o konieczności jego obsługi pojawiła się w momencie, kiedy

pojazd ten zmierzał (lub już dotarł) do klienta 2. Tym samym znacznie się oddalił od klienta B. Biorąc to pod uwagę, jak również aktualną pozycję pierwszego pojazdu (klient 1), decydent uznał, że zostanie on przyporządkowany do pojazdu pierwszego. Wreszcie najwięcej zmian wymusiła konieczność odwiedzenia nowego klienta C. Tutaj po dokonaniu nowych obliczeń modyfikacji uległa nie tylko trasa drugiego pojazdu, lecz także zmieniona została trasa pojazdu trzeciego w wyniku zmiany przyporządkowania jednego z klientów do trasy innego pojazdu.

Pomimo że wiele problemów układania tras pojazdów może mieć charakter dynamiczny, to należy także zwrócić szczególną uwagę na stopień tego dynamizmu. W literaturze przedmiotu proponowane są różne miary opisujące tę właściwość. Mogą one dotyczyć zarówno częstotliwości, z jaką pojawiają się nowi klienci, jak i stopnia szybkości obsługi tych klientów wyrażonej jako różnica czasu pomiędzy zgłoszeniem do obsługi a jej rozpoczęciem.

Jednym z mierników stopnia dynamizmu jest zaproponowany współczynnik δ wyrażający się następującym wzorem (Lund i in., 1996):

$$\delta = \frac{n_d}{N} \quad (4.127)$$

gdzie:

- n_d – liczba nowych klientów zgłaszających się do obsługi w czasie realizacji zaplanowanych tras pojazdów;
- N – całkowita liczba wszystkich punktów obsługi obejmująca zarówno klientów nowych, jak i klientów zaplanowanych wcześniej.

Ze względu na równie istotny charakter dynamizmu odzwierciedlający się w czasie pojawiania się nowych klientów zaproponowany został tzw. efektywny stopień dynamizmu, który wyrażono poniższą formułą (Larsen i in., 2004):

$$\delta^e = \frac{1}{N} \sum_{i \in N_d} \frac{t_i}{T} \quad (4.128)$$

gdzie:

- N_d – zbiór nowych klientów zgłaszających się do obsługi w czasie realizacji zaplanowanych tras pojazdów;
- N – całkowita liczba wszystkich punktów obsługi obejmująca zarówno klientów nowych, jak i klientów zaplanowanych wcześniej;
- t_i – czas pojawienia się nowego klienta w systemie;
- T – długość założonego przez decydenta horyzontu czasowego.

Wreszcie biorąc pod uwagę szybkość obsługi pojawiającego się nowego klienta, zaproponowano rozszerzoną wersję współczynnika (4.128). Wersja ta adresowana jest do problemów układania tras pojazdów z oknami czasowymi i wprowadza pojęcie czasu reakcji na pojawienie się nowego zgłoszenia. Reakcja ta jest różnicą pomiędzy czasem zgłoszenia do systemu a najpóźniejszym terminem obsługi

nowego klienta. Im ten czas reakcji jest dłuższy, tym więcej możliwości ma decydent w modyfikacji aktualnych tras pojazdów (Larsen i in., 2004):

$$\delta_{TW}^e = \frac{1}{N} \sum_{i \in N_d} \left(1 - \frac{l_i - t_i}{T} \right) \quad (4.129)$$

gdzie:

- N_d – zbiór nowych klientów zgłaszających się do obsługi w czasie realizacji zaplanowanych tras pojazdów;
- N – całkowita liczba wszystkich punktów obsługi obejmująca zarówno klientów nowych, jak i klientów zaplanowanych wcześniej;
- l_i – najpóźniejszy termin obsługi i -tego punktu;
- t_i – czas pojawienia się nowego klienta w systemie;
- T – długość założonego przez decydenta horyzontu czasowego.

Na podstawie drugiego z tych wskaźników dynamiczne zagadnienia układania tras pojazdów podzielono na trzy kategorie:

- 1) problemy o małej dynamice, dla których wartość efektywnego współczynnika dynamizmu jest mniejsza od 0,3;
- 2) problemy o średniej dynamice, dla których wartość efektywnego współczynnika dynamizmu jest większa lub równa 0,3 i mniejsza od 0,8;
- 3) problemy o dużej dynamice, dla których wartość efektywnego współczynnika dynamizmu jest większa od 0,3.

Wśród dynamicznych problemów układania tras pojazdów najwięcej uwagi poświęcono zagadnieniom, w których dynamizm polega przede wszystkim na pojawianiu się w systemie nowych zgłoszeń do obsługi (nowych klientów). Interesującym tego przykładem jest tzw. rzeczywisty (praktyczny) problem układania tras pojazdów, w którym przyjęto następujące założenia (Branchini i in., 2009):

- każdy klient odwiedzany jest tylko przez jeden pojazd oraz każdy pojazd zaczyna i kończy swoją trasę w bazie;
- pojazdy mają określoną ładowność, która nie może zostać przekroczona;
- klienci dzielą się na priorytetowych (klienci znani przed rozpoczęciem tras przez pojazdy, a także nowi, którzy zgłosili się do systemu w ustalonych godzinach: 8.00–14.00 i muszą zostać obsłużeni tego samego dnia) oraz nie-priorytetowych (klienci, którzy zgłosili się do systemu po godzinie 14.00 i nie muszą, lecz mogą zostać obsłużeni tego samego dnia);
- każdy klient ma określony przedział czasowy obsługi;
- dopuszcza się maksymalny czas opóźnienia w obsłudze i -tego klienta, lecz nie dłuższy niż 2 godziny;
- wszystkie pojazdy muszą wrócić do bazy przed godziną 17.00;
- każdy pojazd ma dopuszczalną przerwę w pracy wynoszącą 1 godzinę, przy czym może być ona wykorzystana tylko w ustalonych godzinach: 11.00–14.00.

Celem decydenta jest maksymalizacja zysku, który wyrażony został następującą formułą:

$$Z = \sum_{i \in S} r_i - \sum_{i \in S} \max. (0, p_i(b_i - l_i)) - c \sum_{i \in R} d_j - f|R| \quad (4.130)$$

gdzie:

- r_i – dochód związany z obsługą i -tego klienta;
- b_i – moment rozpoczęcia obsługi i -tego klienta;
- l_i – najpóźniejszy moment zakończenia obsługi i -tego klienta;
- p_i – kara za późniejszą niż założona obsługę i -tego klienta;
- d_j – całkowita długość j -tej trasy;
- f – ustalony stały dzienny koszt obsługi pojazdu;
- c – jednostkowy koszt podróży pojazdu;
- R – zbiór tras pojazdów.

4.2.9. Wielokryterialny problem wielu komiwojażerów

W literaturze przedmiotu wielokryterialny charakter podejmowania decyzji nie jest dominującą cechą w opisywanym zbiorze zagadnień układania tras pojazdów. Na początku należy jednak przedstawić ogólną definicję problemu wielokryterialnego. Zdefiniować go można w następujący sposób:

$$WPD = \begin{cases} \min. F(\mathbf{x}) = \{f_1(x), f_2(x), \dots, f_n(x)\} \\ \mathbf{x} \in X \end{cases} \quad (4.131)$$

gdzie liczba celów $f_i(x)$ wynosi co najmniej 2, $F(x)$ jest wektorem celów, natomiast X jest przestrzenią rozwiązań dopuszczalnych. Przyjmując następujące oznaczenia: Y jest przestrzenią rozwiązań dopuszczalnych w przestrzeni kryteriów, gdzie $y = (y_1, y_2, \dots, y_n)$ oraz $y_i = f_i(x)$ jest rozwiązaniem przestrzeni kryteriów, można stwierdzić, że rozwiązanie $z = (z_1, z_2, \dots, z_n)$ jest zdominowane przez rozwiązanie $y = (y_1, y_2, \dots, y_n)$ wtedy i tylko wtedy, gdy dla każdego $i \in \{1, 2, \dots, n\}$ $y_i \leq z_i$ oraz istnieje przynajmniej jedna funkcja celu, dla której zachodzi: $y_i < z_i$. Rozwiązanie y jest Pareto optymalne, jeżeli nie można znaleźć innego rozwiązania, które dominuje y .

W literaturze przedmiotu najczęściej można spotkać następujące trzy podejścia do rozwiązywania problemów wielokryterialnych:

- *a priori*;
- *a posteriori*;
- interaktywne.

Podejścia *a priori* są charakterystyczne dla przypadków, gdy decydent, który chce optymalizować kilka celów jednocześnie, w pierwszej kolejności formułuje

co do nich swoje preferencje przez określenie hierarchii kryteriów. Do tej grupy zaliczyć można problemy układania tras pojazdów, w których wykorzystywane są matematyczne metody transformacji kilku funkcji celów do postaci jednej funkcji. Przykładem takiej transformacji jest liniowa ważona metoda addytywna, w której wyniku wyjściowa postać jednej funkcji staje się sumą ważonych postaci pierwotnych funkcji celu. Niewątpliwą zaletą tego podejścia jest optymalizacja jednej funkcji celu, wobec której można zastosować np. metody optymalizacji przeznaczone do rozwiązywania jednokryteriowych problemów wielu komiwojażerów. Podejście to ma jednak także wady. Po pierwsze, decydent nie ma gwarancji znalezienia wszystkich rozwiązań optymalnych w sensie Pareto. Po drugie zaś, zachodzi tu konieczność uprzedniego określenia przez decydenta ważności poszczególnych celów. Ustalanie tych wag przez decydenta nie zawsze musi się odbywać w sposób obiektywny. Ponadto nawet niewielka zmiana w tym zakresie może mieć znaczący wpływ na uzyskane rozwiązanie.

Druga strategia rozwiązywania problemów wielokryterialnych ma charakter podejścia *a posteriori*. W tej sytuacji wyznaczany jest zbiór rozwiązań niezdominowanych, spośród których decydent wybiera rozwiązanie końcowe.

Wreszcie trzecie z wymienionych podejść ma charakter interaktywny, co oznacza, że decyzje podejmowane są przez decydenta w trakcie procesu poszukiwania rozwiązania optymalnego.

Podstawowy podział wielokryterialnych zagadnień układania tras pojazdów ma swoje źródło przede wszystkim w genezie formułowania tego typu problemów transportowych. Wyróżnić można tutaj:

- rozszerzenia jednokryterialnego problemu układania tras pojazdów na problemy wielokryterialne;
- uogólnienia jednokryterialnego problemu układania tras pojazdów na problemy wielokryterialne;
- problemy wielokryterialne, których sformułowanie zostało zaczerpnięte z szeroko rozumianej praktyki gospodarczej.

W pierwszym typie wielokryterialnych problemów wielu komiwojażerów rozszerzenie jednokryterialnej wersji zagadnienia polega na tym, że dotychczas definiowana funkcja celu (całkowity koszt transportu ponoszony przez pojazdy wyrażony jako np. łączny dystans lub całkowity czas pracy pojazdów) uzupełniana jest o inne postaci funkcji reprezentujących dodatkowe cele określane przez decydenta.

Następny rodzaj wielokryterialnego problemu wielu komiwojażerów – generalizacja jednokryterialnego zagadnienia układania tras pojazdów do problemu wielokryterialnego – uwzględnia dwa sposoby jego rozwiązywania. Pierwszy z nich polega na dodaniu nowych celów w miejsce jednego lub kilku ograniczeń występujących w problemie. Innymi słowy, ograniczenie problemu układania tras pojazdów staje się celem optymalizacji. Drugi z kolei sposób stanowi odwrotność pierwszego. Otóż w swoim ogólnym sformułowaniu problem układania tras

pojazdów może mieć charakter wielokryterialny (a przynajmniej może być tak postrzegany). Jednakże do modelu problemu wielu komiwojażerów dodawane są dodatkowe ograniczenia (lub parametry) wyrażające wielkości, które w naturalny sposób mogłyby stanowić odrębny cel optymalizacji.

Ostatnia z wymienionych kategorii problemów optymalizacji wielokryterialnej w odniesieniu do zagadnień układania tras pojazdów dotyczy przypadków, w których postaci funkcji celu zdefiniowane zostały zazwyczaj na podstawie konkretnych sytuacji decyzyjnych obserwowanych w praktyce gospodarczej.

Interesującym przykładem wielokryterialnego problemu układania tras pojazdów, w którym dokonane zostało rozszerzenie problemu jednokryterialnego, jest zaproponowany przez T.-R. Lee i J.-H. Uenga (1999) model dwukryterialny. Obok dotychczasowej funkcji celu minimalizującej całkowity dystans pokonywany przez pojazdy autorzy przedstawili w nim kryterium, którego celem jest równoważenie obciążenia tras poszczególnych pojazdów. Ponadto w matematycznej postaci modelu programowania całkowitoliczbowego przedstawiono jedną funkcję celu. Stanowi to wyraz podejścia autorów do sposobu poszukiwania rozwiązania optymalnego. Wykorzystano tu technikę skalaryzacji, gdzie funkcja celu w modelu jest sumą ważoną dwóch pierwotnych funkcji celu: całkowitej długości wszystkich tras oraz równowagi tras pomiędzy pojazdami. Postać ogólna omawianej funkcji celu, w której wypadku poszukujemy jej najmniejszej wartości, została przedstawiona poniżej:

$$Z = \gamma_1 A + \gamma_2 B \quad (4.132)$$

Pierwszy człon funkcji (4.132) to całkowity czas wszystkich tras wykorzystanych pojazdów, który wyrażony jest następującą formułą:

$$A = \left(\sum_{k=1}^K \sum_{i=0}^N \sum_{j=0}^N t_{ij} x_{ij}^k \right) T_a \quad (4.133)$$

gdzie:

- t_{ij} – najkrótszy dystans pomiędzy i -tym a j -tym klientem;
- T_a – średni jednostkowy (np. na kilometr) czas podróży pojazdu;
- x_{ij}^k – zmienna decyzyjna przyjmująca wartość 1 wtedy i tylko wtedy, gdy trasa pojazdu k uwzględnia odcinek pomiędzy i -tym a j -tym klientem, oraz 0, gdy go nie uwzględnia.

Na szczególną uwagę zasługuje tutaj drugi człon omawianej funkcji celu (4.132), w którym autorzy starają się wyrazić istniejącą różnicę w czasie pracy poszczególnych pojazdów. Na czas pracy danego pojazdu składają się zarówno czas podróży pojazdu od i -tego do j -tego klienta, jak i czas obsługi przez pojazd u klienta (np. czas rozładunku towaru). Postać tej części funkcji celu została zaproponowana przez autorów w następujący sposób:

$$\begin{aligned}
 B = & \left(T_a \sum_{i=0}^N \sum_{j=0}^N t_{ij} x_{ij}^1 + T_b \sum_{i=0}^N \sum_{j=0}^N q_i x_{ij}^1 \right) + \\
 & + \left(T_a \sum_{i=0}^N \sum_{j=0}^N t_{ij} x_{ij}^2 + T_b \sum_{i=0}^N \sum_{j=0}^N q_i x_{ij}^2 \right) + \\
 & \dots \\
 & + \left(T_a \sum_{i=0}^N \sum_{j=0}^N t_{ij} x_{ij}^{K-1} + T_b \sum_{i=0}^N \sum_{j=0}^N q_i x_{ij}^{K-1} \right) - \\
 & - (K-1) \left(T_a \sum_{i=0}^N \sum_{j=0}^N t_{ij} x_{ij}^K + T_b \sum_{i=0}^N \sum_{j=0}^N q_i x_{ij}^K \right) -
 \end{aligned} \tag{4.134}$$

gdzie:

- t_{ij} – najkrótszy dystans pomiędzy i -tym a j -tym klientem;
- T_a – średni jednostkowy (np. na kilometr) czas podróży pojazdu;
- T_b – średni czas rozładunku jednostki towaru u klienta;
- q_i – popyt i -tego klienta;
- x_{ij}^k – zmienna decyzyjna przyjmująca wartość 1 wtedy i tylko wtedy, gdy trasa pojazdu k uwzględnia odcinek pomiędzy i -tym a j -tym klientem, oraz 0, gdy go nie uwzględnia.

Funkcja (4.134) przedstawia równowagę czasu pracy poszczególnych pojazdów wykorzystanych do obsługi zadanego zbioru klientów. Została wyrażona jako suma różnic czasu pracy k -tego pojazdu i pojazdu, dla którego ten czas jest najkrótszy.

Przykład

Czas pracy pojazdu $k_1 = 360$ minut, $k_2 = 480$ minut, $k_3 = 400$ minut oraz $k_4 = 420$ minut. Wartość drugiego członu funkcji celu (4.132), zgodnie z formułą określoną wzorem (4.134), będzie równa, a tym samym bilans wyniesie 220 minut:

$$B = (360 - 360) + (480 - 360) + (400 - 360) + (420 - 360) = (480 + 400 + 420) - 3 \cdot 360 = 220$$

Tak sformułowana postać drugiej części funkcji celu wymaga dodatkowo znalezienia najkrótszego czasu pracy pojazdu, co przez autorów zostało odzwierciedlone w modelu programowania matematycznego w postaci zaproponowanego odpowiedniego ograniczenia przez spełnienie nierówności dla każdego $k = 1, 2, \dots, K - 1$:

$$\begin{aligned}
 T_a \sum_{i=0}^N \sum_{j=0}^N t_{ij} x_{ij}^k + T_b \sum_{i=0}^N \sum_{j=0}^N q_i x_{ij}^k \geq T_a \sum_{i=0}^N \sum_{j=0}^N t_{ij} x_{ij}^K + \\
 T_b \sum_{i=0}^N \sum_{j=0}^N q_i x_{ij}^K
 \end{aligned} \tag{4.135}$$

Współczynniki (wagi) γ_1 oraz γ_2 ustalane są przez decydenta w sposób arbitralny. Autorzy przyjęli, że wagi obydwu członów zaproponowanej przez nich funkcji celu (4.132) mają jednakową wartość.

Drugim przykładem wielokryterialnego podejścia do problemu układania tras pojazdów, które jednocześnie stanowi rozszerzenie jednokryterialnego zagadnienia wielu komiwojażerów, jest model opracowany przez R. Ribeiro i H.R. Lourenço (2001). Jest to model zaproponowany dla problemu układania tras pojazdów z obsługą okresową, w którym przedstawiona została minimalizacja trzech funkcji celu. Tak jak we wcześniej prezentowanej pracy, pierwsze dwa kryteria mają podobny charakter.

Pierwsza z funkcji celu minimalizuje całkowity koszt całego programu wszystkich tras K realizowanych w zadanym przedziale czasowym P .

$$F_1 = \sum_{t=1}^P \sum_{i=0}^N \sum_{j=0}^N c_{ij} \sum_{k=1}^K x_{ijk}^t \quad (4.136)$$

Druga funkcja celu wyraża równowagę w zaangażowaniu pojazdów, lecz rozumianą tutaj jako równomierność wielkości transportowanego dobra w poszczególnych okresach ustalonego przedziału czasowego P . W tym celu posłużono się odchyleniem standardowym pracy każdego pojazdu, co oznacza, że funkcja celu F_2 nie jest liniowa:

$$F_2 = \sqrt{\frac{\sum_{k=1}^K w_k^2}{m} - \left(\frac{\sum_{k=1}^K w_k}{m}\right)^2} \quad (4.137)$$

Omówienia wymaga tutaj wielkość w_k , która wyraża całkowitą pracę k -tego pojazdu. Jest ona sformułowana w następujący sposób:

$$w_k = \sum_{t=1}^P \sum_{i=1}^N q_i^t y_{ik}^t \quad (4.138)$$

gdzie jest zmienną decyzyjną oznaczającą wizytę k -tego pojazdu u i -tego klienta w t -tym okresie (np. dniu) przedziału czasowego P .

Trzecia z zaprezentowanych przez autorów funkcji ma wyraźnie charakter marketingowy i wynika z tego, że korzystne dla przedsiębiorstwa może być przyporządkowanie danego pojazdu (a raczej konkretnego kierowcy) zawsze do tego samego klienta. Takie podejście z jednej strony pozwala na budowanie oraz utrzymywanie dobrych relacji pomiędzy przedsiębiorstwem a klientami. Z drugiej strony może prowadzić do szybkiego naruszenia ograniczeń zasobowych, np. ładowności pojazdów. Tym samym postanowiono zastosować tę strategię przynajmniej do zbioru najlepszych klientów. Dla każdego klienta ustalono parę dni (g, h) , dla których występuje dodatni popyt (lub podaż w wypadku zwózki towaru). Niech T_i oznacza zbiór takich dni dla każdego i -tego klienta, przy czym $g \neq h$. Postać trzeciej z zaproponowanych funkcji celu jest następująca:

$$F_3 = \sum_{i=0}^N \sum_{k=1}^K \sum_{\substack{(g,h) \in T_i \\ g < h}} \left[\left(\sum_{t=1}^P q_i \right) \left| y_{ik}^g - y_{ik}^h \right| \right] \quad (4.139)$$

Z funkcji celu (4.139) wynika, że celem jest minimalizacja różnicy $|y_{ik}^g - y_{ik}^h|$. Szczególne znaczenie ma tutaj popyt (lub podaż) i -tego klienta q_i . Jest on pewnego rodzaju wagą określającą „istotność” danego klienta obsługiwanego przez bazę.

Wszystkie trzy przedstawione powyżej funkcje celu F_1 , F_2 oraz F_3 zostały zagregowane do postaci jednej funkcji. Ma ona postać sumy ważonej, w której wagi stanowią wektor preferencji użytkownika (decydenta). Należy podkreślić, że pierwszy człon tej funkcji stanowi koszt, drugi – odchylenie standardowe czasu pracy pojazdów, natomiast trzeci wyrażony jest ilością dostarczanego (lub odbieranego) towaru. Ze względu na to, że poszczególne człony nie są wyrażone w tych samych jednostkach, konieczna jest modyfikacja drugiej i trzeciej funkcji celu (F_2 i F_3), tak aby można było wyrazić ich wartości w jednostkach pierwszej funkcji celu F_1 . Stąd też przyjęto jako c_B koszt jednostkowy odchylenia standardowego, który może być interpretowany jako swoisty koszt przedsiębiorstwa wynikający z nierówności tras pojazdów. Natomiast wielkość c_C oznacza jednostkowy koszt przedsiębiorstwa wynikający z tego, że ten sam klient nie jest obsługiwany przez jednego kierowcę (koszt kary), i wtedy postać zagregowanej funkcji celu jest następująca:

$$F_c = \alpha_1 F_1 + \alpha_2 c_B F_2 + \alpha_3 c_C F_3 \quad (4.140)$$

przy czym $\alpha_1 + \alpha_2 + \alpha_3 = 1$. Autorzy nie określają niestety sposobu definiowania wag, co sugeruje dowolność decydenta w tym zakresie. Ponadto, jak sami podkreślają, dodanie dodatkowych sztucznych kosztów ma wpływ na wartość funkcji celu, a tym samym na uzyskane wyniki.

Kolejnym przykładem rozszerzenia jednokryterialnego problemu wielu komiwojazerów na zagadnienie wielokryterialne jest model złożonego problemu układania tras pojazdów z jednoczesną dostawą dóbr do klientów i odbiorem dóbr od klientów, a także wraz z określonymi oknami czasowymi. Jego autorzy zaproponowali jednoczesną optymalizację pięciu funkcji celu (Wang i in., 2015).

Pierwsze dwie funkcje określane są mianem funkcji kosztu transportu. Jedna funkcja ma zapewnić minimalizację liczby użytych pojazdów, a tym samym minimalizację całkowitego kosztu zakupu lub wynajmu pojazdów niezbędnych do obsługi punktów:

$$F_1 = |R| = M \rightarrow \min. \quad (4.141)$$

gdzie:

$$R - \text{zbiór } M \text{ tras, } R = \{r_1, r_2, \dots, r_M\}.$$

Druga z funkcji kosztowych reprezentuje podstawową funkcję celu każdego zagadnienia układania tras pojazdów, czyli minimalizację całkowitej długości wszystkich tras wykorzystanych pojazdów:

$$F_2 = \sum_{j=1}^M \text{Dist}_j \rightarrow \min. \quad (4.142)$$

gdzie:

$Dist_j$ – wielkość określająca długość trasy j -tego pojazdu; jest sumą odcinków tras, jakie pokonuje pojazd pomiędzy przyporządkowanymi do niego klientami, w tym także odcinków z i do bazy.

Kolejna funkcja celu omawianego modelu ma postać:

$$F_3 = \max. \{T_j\} \rightarrow \min. \quad (4.143)$$

gdzie:

T_j – wielkość reprezentująca całkowity czas pracy j -tego pojazdu (dla $j = 1, 2, \dots, M$); jest sumą czasów: podróży pojazdu, obsługi klienta przez pojazd, a także oczekiwania na obsługę.

Autorzy wskazują, że cel ten ma dwojaki wpływ na zróżnicowany transport. Z jednej strony pozwala zachować równowagę obciążenia poszczególnych tras pojazdów. Z drugiej zaś minimalizowanie całkowitego czasu pracy pojazdu, a tym samym jego kierowcy, daje możliwość zwolnienia dodatkowych zasobów czasowych, które mogłyby zostać wykorzystane w inny sposób, np. na szkolenia i rozwój zawodowy, co niewątpliwie jest wartością dodaną dla przedsiębiorstwa.

Czwartym kryterium podejmowania decyzji proponowanym przez autorów tego modelu jest minimalizacja łącznego czasu oczekiwania pojazdów na obsługę klientów należących do ich tras:

$$F_4 = \sum_{j=1}^M W_j \rightarrow \min. \quad (4.144)$$

gdzie:

W_j – wielkość reprezentująca łączny czas oczekiwania na obsługę j -tego pojazdu (dla $j = 1, 2, \dots, M$).

Cel ten wyraża efektywność pracy, minimalizując straty czasowe, jakie pojazd może ponieść na niepotrzebne oczekiwanie.

Wreszcie ostatnia zaproponowana funkcja celu odnosi się do czasu opóźnienia w obsłudze klientów:

$$F_5 = \sum_{j=1}^M DT_j \rightarrow \min. \quad (4.145)$$

gdzie:

DT_j – wielkość reprezentująca sumę opóźnień w rozpoczęciu obsługi klientów j -tego pojazdu (dla $j = 1, 2, \dots, M$).

Cel ten jest wyrazem jednej z podstawowych zasad odpowiedniego poziomu obsługi logistycznej – „dokładnie na czas”.

Pięć przedstawionych powyżej kryteriów podejmowania decyzji w zagadnieniu układania tras pojazdów rozpatruje się jako minimalizację wektora funkcji F oraz poddaje optymalizacji w sensie Pareto.

Dotychczas omówione zostały przykłady wielokryterialnych problemów wielu komiwojazerów, w których kilka funkcji celu zostało sprowadzonych do postaci jednej funkcji. Drugim typem problemów wielokryterialnych są zagadnienia, które powstały przez generalizację (uogólnienie) problemów jednokryterialnych. Przykładem takiego podejścia jest sytuacja, w której coś, co w sposób naturalny stanowić by mogło cel decydenta, zostało przedstawione w formie ograniczenia. D. Feillet i in. (2005) opisali grupę trzech problemów wielokryterialnych, a ściślej mówiąc – dwukryterialnych. Dotyczą one co prawda problemu jednego komiwojazera, lecz w prosty sposób mogą zostać rozszerzone na klasę zagadnień wielu pojazdów. Oprócz kosztu (dystansu), jaki ponosi pojazd, pokonując swoją trasę, zakłada się, że wizyta u każdego klienta łączy się z osiągnięciem pewnego zysku. Oznacza to, że decydent ma do czynienia z dwoma konfliktowymi celami, jakimi są: całkowity koszt obsługi klientów przez pojazd oraz całkowity zysk wynikający z tej obsługi. Autorzy przedstawiają trzy podejścia do tak ujętych celów w problemie jednego komiwojazera:

- 1) obydwie cele stanowią składowe w jednej funkcji kryterium, co oznacza, że celem decydenta jest znalezienie rozwiązania minimalizującego różnicę kosztu obsługi trasy przez pojazd i zysku wynikającego z obsługi klientów:

$$\min. \left(\sum_{(i,j) \in V} c_{ij} x_{ij} - \sum_{i \in V} p_i y_i \right) \quad (4.146)$$

- 2) koszt obsługi punktów przez pojazd przedstawiony jest jako ograniczenie, w którym koszt ten nie może przekroczyć zadanej wartości maksymalnej c_{\max} , natomiast celem decydenta jest znalezienie trasy maksymalizującej zysk wynikający z odwiedzania klientów:

$$\max. \sum_{i \in V} p_i y_i \quad (4.147)$$

przy spełnieniu warunku wyrażonego nierównością

$$\sum_{(i,j) \in V} c_{ij} x_{ij} \leq c_{\max}. \quad (4.148)$$

- 3) zysk wynikający z obsługi punktów przez pojazd przedstawiony jest w postaci warunku, w którym nie może on być niższy od zadanej wartości minimalnej p_{\min} , natomiast celem decydenta jest minimalizacja łącznego kosztu podróży pojazdu:

$$\min. \sum_{(i,j) \in V} c_{ij} x_{ij} \quad (4.149)$$

przy spełnieniu warunku wyrażonego nierównością

$$\sum_{i \in V} p_i y_i \geq p_{\min}. \quad (4.150)$$

Innym przykładem uogólnienia do jednokryterialnego problemu o podobnym charakterze jest propozycja, której autorami są J. Riera-Ledesman i J.J. Salazar-Gonzales (2005), odniesiona także do problemu jednego pojazdu, nazwanego zadaniem podróżującego kupca (*Traveling Purchaser Problem*). W klasycznym problemie układania tras pojazdów konieczne jest odwiedzenie wszystkich punktów (klientów) obsługiwanych przez bazę. W propozycji wymienionych autorów zakłada się, że kupiec ma za zadanie dostarczyć niezbędnych ilości d_k z listy produktów: $K = \{p_1, p_2, \dots, p_m\}$. Produkty te mogą być kupione w podzbiorze klientów $M_K \subset M$, co oznacza, że nie wszyscy klienci dysponują danym produktem. Ponadto każdy klient dysponujący danym produktem ma go w ilości nie większej niż q_{ki} oraz sprzedaje go po cenie b_{ki} . Dopuszczalna trasa w przedstawionym problemie podróżującego kupca może nie zawierać wszystkich dostępnych punktów, co stanowi tutaj charakterystyczną cechę tego problemu. Może się okazać, że pewien podzbiór klientów (w tym wypadku dostawców–sprzedawców produktu) zapewni żądaną wielkość zakupu kupca. Sformułowane dwa cele optymalizacyjne są następujące:

1) minimalizacja kosztu zakupu wszystkich produktów:

$$F_1 = \sum_{p_k \in K} \min. \sum_{i \in M_k} b_{ki} z_{ki} \quad (4.151)$$

gdzie z_{ki} jest zmienną decyzyjną oznaczającą ilość zakupionego produktu p_k u i -tego klienta oraz $\sum_{i \in M_k} z_{ki} = d_k$ i $z_{ki} \leq q_{ki}$;

2) minimalizacja kosztu podróży wyrażona jako:

$$F_2 = \sum_{(i,j) \in M_k} \min. \sum_{i \in M_k} c_{ij} \quad (4.152)$$

W literaturze przedmiotu dominuje podejście, w którym obie funkcje przedstawione są w postaci jednej funkcji celu przez dodanie do siebie obydwu rodzajów kosztu. Jednakże jak zaznaczają autorzy, ze względu na to, że oba cele mogą być w rzeczywistości trudno porównywalne (np. koszt podróży mierzony jest jednostkami odległości lub czasu) oraz ze względu na wzajemne relacje obydwu celów (niższa cena – dłuższy czas podróży), zachodzi konieczność zastosowania optymalizacji wielokryterialnej w sensie Pareto.

Zgola odwrotnym podejściem do uogólnienia jednokryterialnego zagadnienia układania tras pojazdów w wielokryterialne będzie w tym wypadku sytuacja, w której dodawane są nowe kryteria oceny tras przez usuwanie z modelu odpowiednich ograniczeń lub parametrów. Takie przykłady uogólnień są szczególnie dobrze widoczne w problemach wielu komiwojażerów z oknami czasowymi. Reprezentantem takiego podejścia może być tutaj propozycja B. Barana i M. Schaerera (2003), w której oprócz dobrze znanych funkcji celu dotyczących minimalizacji całkowitego czasu podróży i minimalizacji liczby użytych pojazdów dodano dodatkowo jako cel (nie ograniczenie) całkowity czas dostaw (obsługi punktów) przez pojazdy. Z kolei

M.J. Geiger (2001) proponuje minimalizację czterech funkcji celu. Pierwsze dwie są takie same jak u Barana i Schaerera. Kolejne funkcje celu powstały z ograniczeń dotyczących okien czasowych. Jedna z nich odnosi się do minimalizacji spóźnień pojazdów w obsłudze klientów. Druga zaś stanowi minimalizację liczby okien czasowych, które nie zostały dotrzymane przez wykorzystane do obsługi tras pojazdy.

Nieco odmienną, lecz niewątpliwie godną uwagi prezentacją wielokryterialnego uogólnienia problemu wielu komiwojażerów znajdujemy w pracy S.-Ch. Honga i Y.-B. Parka (1999). Opisują oni dwukryterialny model układania tras pojazdów z wykorzystaniem programowania celowego. Obok minimalizacji całkowitego czasu podróży pojazdów autorzy uwzględnili także minimalizację całkowitego czasu oczekiwania klientów na obsługę przez pojazdy. Prezentują oni wektor celów składający się z trzech elementów: A , B i C .

Gdy przyjmiemy dodatkowo następujące oznaczenia:

- M – dowolnie duża liczba;
- NV – liczba wykorzystanych do obsługi punktów pojazdów;
- N – liczba lokalizacji punktów obsługi oraz bazy (wierzchołek nr 1);
- a_i – czas przybycia pojazdu do i -tego punktu obsługi;

pierwsza składowa wektora A reprezentuje minimalizację łącznego odchylenia ujemnego lub dodatniego (lub obydwu jednocześnie) dotyczącego ograniczeń:

$$A = \sum_{j=2}^{N+NV} (s_{2j}^- + s_{2j}^+) + \sum_{i=2}^N (s_{3i}^- + s_{3i}^+) + (s_4^- + s_4^+) + \sum_{i=1}^N \sum_{j=2}^{N+NV} s_{5ij}^- + \sum_{i=1}^N s_{6i}^+ + \sum_{i=N+1}^{N+NV} s_{7i}^+ + \sum_{i=1}^N \sum_{j=2}^{N+NV} s_{8ij}^- + \sum_{i=N+1}^{N+NV} s_{9i}^+ \quad (4.153)$$

gdzie:

s_{2j}^- i s_{2j}^+ – są odchyleniami od pierwszej grupy ograniczeń zapewniających, że każdy klient odwiedzany jest przez pojazd tylko jeden raz:

$$\sum_{i=1}^N x_{ij} + s_{2j}^- - s_{2j}^+ = 1$$

dla $j = 2, \dots, N$

s_{3i}^- i s_{3i}^+ – są odchyleniami od drugiej grupy ograniczeń zapewniających, że każdy klient odwiedzany jest przez pojazd tylko jeden raz:

$$\sum_{j=2}^{N+NV} x_{ij} + s_{3i}^- - s_{3i}^+ = 1$$

dla $i = 2, \dots, N$

s_4^- i s_4^+ – są odchyleniami od ograniczeń zapewniających, że użytych zostanie NV pojazdów:

$$\sum_{j=2}^N x_{1j} + s_4^- - s_4^+ = NV$$

s_{5ij}^- i s_{5ij}^+ – są odchyleniami od ograniczeń, w których obliczany jest czas przybycia pojazdu do klienta:

$$a_j - a_i - s_{11i}^- - B_1 x_{ij} + s_{5ij}^- - s_{5ij}^+ = t_{ij} + s_i - M$$

dla $i = 1, \dots, N; j = 2, \dots, N + NV$ oraz $i \neq j$

s_{6i}^- i s_{6i}^+ – są odchyleniami od ograniczeń, w których definiowany jest najpóźniejszy termin obsługi klienta przez pojazd, przy czym požądane jest, aby $s_{6i}^+ = 0$:

$$a_i + s_{6i}^- - s_{6i}^+ = L_i$$

dla $i = 1, \dots, N$

s_{7i}^- i s_{7i}^+ – są odchyleniami od ograniczeń, w których zapewnia się powrót pojazdu do bazy obsługującej klientów przed ustalonym czasem (innymi słowy, jest okno czasowe wyznaczone dla pojazdów):

$$a_i + s_{7i}^- - s_{7i}^+ = R$$

dla $i = N + 1, \dots, N + V$

s_{8ij}^- i s_{8ij}^+ – są odchyleniami od ograniczeń, w których obliczane jest aktualne wykorzystanie ładowności pojazdu opuszczającego j -tego klienta:

$$q_j - q_i - M x_{ij} + s_{8ij}^- - s_{8ij}^+ = c_j - B_2$$

dla $i = 1, \dots, N; j = 2, \dots, N + NV$ oraz $i \neq j$

s_{9i}^- i s_{9i}^+ – są odchyleniami od ograniczeń zapewniających, że ładowność pojazdu nie zostanie przekroczona:

$$q_i + s_{9i}^- - s_{9i}^+ = Q$$

dla $i = N + 1, \dots, N + V$

Pozostałe dwa elementy – B oraz C – wektora celów są zdefiniowane przez autorów w następujący sposób:

$$B = s_{10}^+ \tag{4.154}$$

$$C = s_{11i}^+ \text{ dla } i = 1, \dots, N \tag{4.155}$$

gdzie:

s_{10}^- i s_{10}^+ – są odchyleniami dotyczącymi ograniczenia zapewniającego, że czas pracy pojazdu jest ograniczony przez ustaloną wartość (lub jest to docelowa wartość):

$$\sum_{i=1}^N \sum_{j=2}^{N+NV} t_{ij} x_{ij} + s_{10}^- - s_{10}^+ = TT$$

s_{11i}^- i s_{11i}^+ – są odchyleniami od ograniczeń, w których obliczany jest odpowiednio czas oczekiwania pojazdu u i -tego klienta (pojazd przybył za wcześniej) oraz czas oczekiwania i -tego klienta na pojazd (pojazd przybył za późno):

$$a_i + s_{11i}^- - s_{11i}^+ = E_i$$

dla $i = 1, \dots, N$

4.2.10. Wielobazowy problem wielu komiwojazerów

Problem układania tras dostaw pojazdów z wieloma bazami podejmowany jest w literaturze przedmiotu z jednej strony jako odmiana problemu wielu komiwojazerów, z drugiej zaś traktowany jest także jako odrębne zagadnienie, podobnie jak to obserwowaliśmy w wypadku problemu jednego komiwojazera. Niewątpliwie zagadnienie układania tras pojazdów z jedną bazą jest szczególnym przypadkiem problemów z wieloma bazami. Jednakże należy uznać, że w piśmiennictwie z tego zakresu dominuje podejście, w którym wielobazowy problem wielu komiwojazerów stanowi odmianę szerokiej klasy zadań układania tras pojazdów.

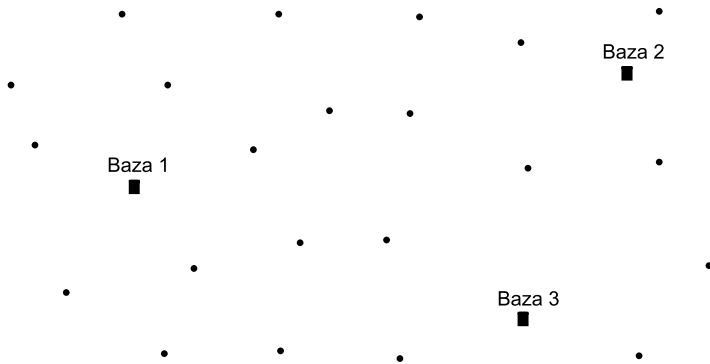
Problem ustalania tras pojazdów z wieloma bazami (rys. 4.13) można zdefiniować w następujący sposób (Hjorring, 1995; Li i Simchi-Levi, 1990): przedsiębiorstwo dysponuje pewną liczbą baz, którymi mogą być ośrodki produkcyjne (zakłady produkcyjne), ośrodki dystrybucyjne lub składowania (magazyny). Ośrodki te odpowiedzialne są za kompleksową obsługę wszystkich klientów (punktów odbioru lub nadania towaru). Zarówno bazy, jak i obsługiwane punkty zlokalizowane są w obrębie pewnego określonego obszaru geograficznego (np. miasto, region). Przedsiębiorstwo dysponuje również określonym taborem transportowym przystosowanym do obsługi punktów, i tu możemy mieć do czynienia z dwoma wariantami:

- 1) liczba oraz rodzaj użytych pojazdów, jakimi dysponuje każda baza, są narzucone z góry i nie można ich przekroczyć/zmienić;
- 2) liczba środków transportu, która zostanie użyta do obsługi punktów, jest niewiadomą i dopiero trzeba ją będzie określić.

Należy znaleźć taki zbiór tras dla pojazdów, gdzie dany pojazd wyrusza z bazy, odwiedza pewien podzbiór wszystkich przyporządkowanych mu punktów obsługi,

przydzielonych do tej bazy, by następnie do niej powrócić. Celem jest uzyskanie jak najniższego kosztu związanego z obsługą transportową swoich klientów.

Tak jak zagadnienie układania tras pojazdów z jedną bazą, tak i problem wielu komiwojazerów z wieloma bazami posiada swoją definicję opartą na teorii grafów (Renaud i in., 1996). Dany jest nieskierowany i spójny graf $G(V, E, c)$, w którym zbiór V składa się z dwóch części: V_c oraz V_d . Podzbiór V_c odpowiada zbiorowi n klientów: $V_c = \{v_1, \dots, v_n\}$, natomiast podzbiór V_d odpowiada zbiorowi p baz: $V_d = \{v_{n+1}, \dots, v_{n+p}\}$. Zbiór krawędzi E grafu reprezentuje możliwe połączenia pomiędzy punktami i oraz j , a także pomiędzy bazami a poszczególnymi punktami obsługi.



Rys. 4.13. Zadanie wyjściowe problemu wielu komiwojazerów z wieloma bazami
Źródło: opracowanie własne.

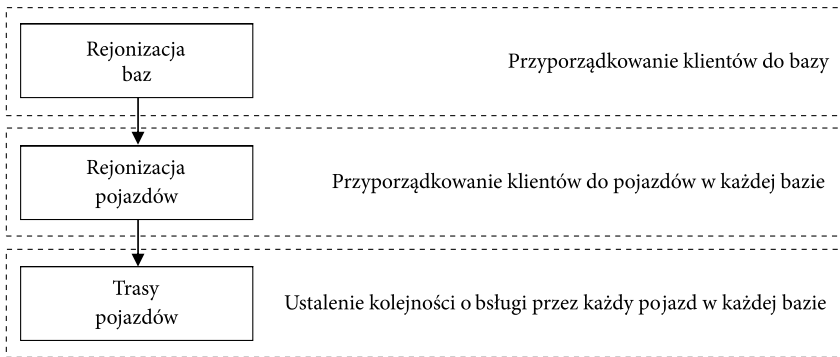
Każdej krawędzi grafu łączącej dwa dowolne węzły przyporządkowana jest nieujemna liczba c_{ij} (waga krawędzi) odpowiadająca długości (kosztowi) połączenia pomiędzy punktami i oraz j oraz pomiędzy bazami a poszczególnymi punktami obsługi. M węzłów grafu reprezentuje punkty początkowe, z których pojazdy rozpoczynają swoją trasę oraz do nich powracają. Za rozwiązanie dopuszczalne uważać należy taki zbiór krawędzi, dla którego:

- pierwsza krawędź trasy danego pojazdu rozpoczyna się w węźle reprezentującym bazę;
- ostatnia krawędź trasy danego pojazdu kończy się w węźle reprezentującym tę samą bazę, z której rozpoczynał on swoją trasę;
- każdy węzeł grafu reprezentujący punkt obsługi stanowi połączenie dwóch i tylko dwóch krawędzi grafu;
- węzeł grafu reprezentujący bazę stanowi połączenie krawędzi w liczbie równej dwukrotności liczby użytych pojazdów tej bazy.

Z punktu widzenia liczby wierzchołków grafu wśród zagadnień układania tras pojazdów problem ustalania tras dostaw z wieloma bazami jest problemem o najszerszym charakterze. Świadczy o tym także liczba problemów optymalizacyjnych, jakie należy rozwiązać:

- które punkty obsługi mają być przyporządkowane do danej bazy, czyli jaki jest rejon działania każdej z baz;
- w ramach zasięgu działania każdej z baz, które punkty mają być przyporządkowane do danego pojazdu z floty, jaką dysponuje baza, czyli jaki jest rejon działania każdego z pojazdów;
- w jakiej kolejności mają być obsługiwane punkty, które są przyporządkowane do konkretnego pojazdu, czyli jaki jest optymalny kształt każdej trasy.

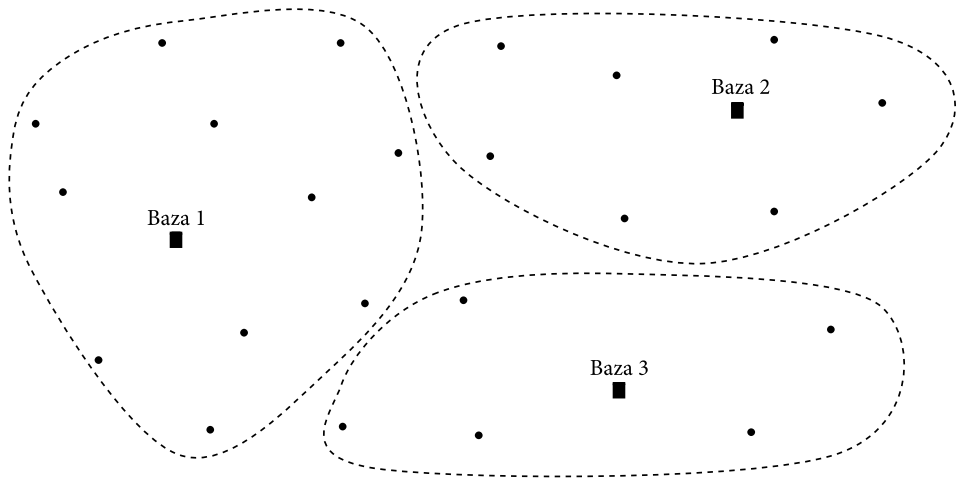
Trzy powyższe problemy optymalizacyjne mają charakter hierarchiczny (rys. 4.14), przy czym hierarchia nie oznacza tutaj ważności danego problemu, a raczej kolejność podejmowania decyzji stojących przed decydem.



Rys. 4.14. Hierarchia podejmowania decyzji w problemie wielu komiwojazerów
Źródło: (Ho i in., 2008).

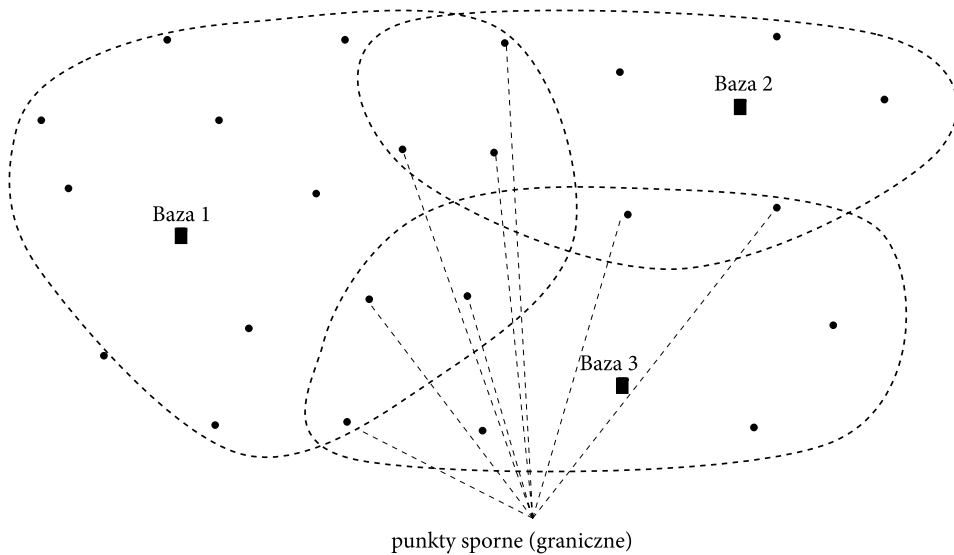
Rozwiązanie pierwszego z wymienionych problemów optymalizacyjnych (określenie rejonów działania każdej z baz) wpływa w bezpośredni sposób na określenie rejonu obsługi przez dany pojazd przyporządkowany do konkretnej bazy. W problemie rejonizacji poszczególnych baz należy zwrócić szczególną uwagę na jeden istotny aspekt. Zazwyczaj dąży się do tego, aby rejony obsługi każdej z baz przedsiębiorstwa ustalone były na dłuższy okres, czyli żeby nie były każdorazowo zmieniane. Bardzo ważne jest również to, aby każdy punkt obsługi był jednoznacznie przyporządkowany do danej bazy. Rejony baz powinny być izolowane (rys. 4.15a). Na ten problem zwracają uwagę B.L. Golden i in. (1997), wyodrębniając punkty obsługi, które nazywają spornymi lub granicznymi (rys. 4.15b).

Wyodrębnienie punktów spornych następuje na podstawie kryterium odległości danego punktu między bazą leżącą najbliżej oraz drugą w kolejności najbliższą bazą. Punkty, które nie są sporne, są bezpośrednio przyporządkowane do najbliższej bazy. Natomiast przydział pozostałych punktów obsługi może się odbywać pośrednio przez odrębne zastosowanie specjalnie w tym celu skonstruowanych metod optymalizacyjnych. W ten sposób tworzone są izolowane rejony działań dla baz, dla których oddzielnie rozwiązywane jest zagadnienie ustalania tras pojazdów z jedną bazą.



Rys. 4.15a. Izolowane rejony działania poszczególnych baz

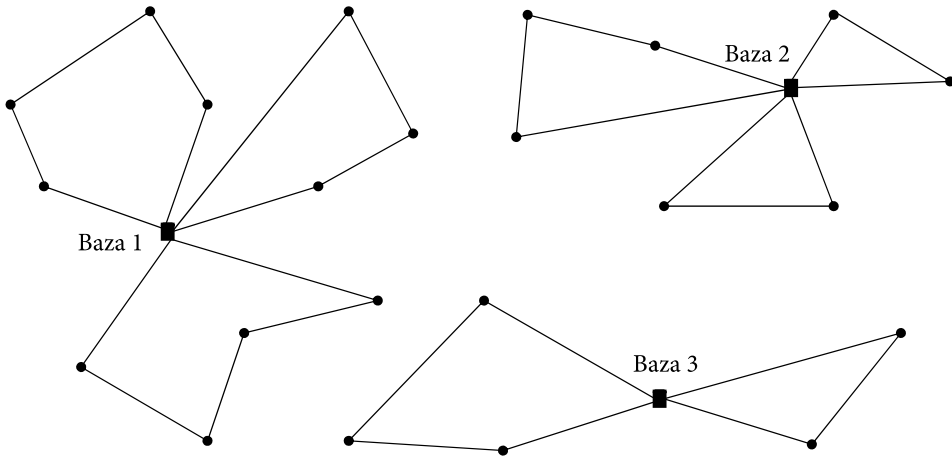
Źródło: opracowanie własne.



Rys. 4.15b. Rejony działania poszczególnych baz ze spornymi punktami obsługi

Źródło: opracowanie własne.

Wskazanie rejonu obsługi klientów przez pojazd z danej bazy warunkuje uzyskanie konkretnej trasy przez ten pojazd (rys. 4.16). Należy przy tym pamiętać, że omawiane problemy optymalizacyjne są dla rozwiązującego zadanie układania tras dla pojazdów z wieloma bazami tak samo istotne.



Rys. 4.16. Przykładowe rozwiązanie problemu wielu komiwojazerów z wieloma bazami
Źródło: opracowanie własne.

Dla problemu wielu komiwojazerów z wieloma bazami zaproponowanych zostało kilka sformułowań modelu w języku programowania matematycznego. Gdy przyjmujemy następujące oznaczenia:

- M – liczba baz;
- N – liczba obsługiwanych punktów znajdujących się w rejonie obsługi wszystkich baz;
- K – całkowita liczba pojazdów będąca w dyspozycji wszystkich baz;
- q_i – popyt (podaż) i -tego punktu obsługi;
- Q_k – ładowność k -tego pojazdu;
- T_k – całkowity koszt pracy (czas) k -tego pojazdu;
- c_{ij} – koszt (lub czas) transportu pomiędzy punktami i oraz j , a także pomiędzy i -tym punktem obsługi a m -tą bazą;
- x_{ij}^k – zmienna decyzyjna przyjmująca wartość 1, jeżeli pojazd k ma w swojej trasie połączenie pomiędzy punktami i oraz j , lub 0, gdy takiego połączenia nie ma, przy czym $i \neq j$;
- y_i, y_j – zmienne przyjmujące wartości liczb rzeczywistych, przy czym $i \neq j$;

model programowania matematycznego będący rozwinięciem modelu dla zagadnienia układania tras pojazdów z jedną bazą będzie się przedstawiał następująco (Montoya-Torres i in., 2015):

$$\min. F(\mathbf{x}) = \sum_{i=1}^{N+M} \sum_{j=1}^{N+M} \sum_{k=1}^K c_{ij} x_{ij}^k \tag{4.156}$$

$$\sum_{i=1}^{N+M} \sum_{k=1}^K x_{ij}^k = 1 \text{ dla } j = 1, \dots, N \tag{4.157}$$

$$\sum_{j=1}^{N+M} \sum_{k=1}^K x_{ij}^k = 1 \quad \text{dla } i = 1, \dots, N \quad (4.158)$$

$$\sum_{i=1}^{N+M} x_{ih}^k - \sum_{j=1}^{N+M} x_{hj}^k = 0 \quad \text{dla } h = 1, \dots, N+M \text{ oraz } k = 1, \dots, K \quad (4.159)$$

$$\sum_{i=N+1}^{N+M} \sum_{j=1}^N x_{ij}^k \leq 1 \quad \text{dla } k = 1, \dots, K \quad (4.160)$$

$$\sum_{j=N+1}^{N+M} \sum_{i=1}^N x_{ij}^k \leq 1 \quad \text{dla } k = 1, \dots, K \quad (4.161)$$

$$\sum_{i=1}^{N+M} \sum_{j=1}^{N+M} x_{ij}^k \leq Q_k \quad \text{dla } k = 1, \dots, K \quad (4.162)$$

$$\sum_{i=0}^n v_i \sum_{j=0}^n c_{ij} x_{ij}^k \leq T_k \quad \text{dla } k = 1, \dots, K; v_0 = 0 \quad (4.163)$$

$$y_i - y_j + (N+M)x_{ij}^k \leq N+M-1 \quad \text{dla } i, j = 1, \dots, N \text{ i } i \neq j \text{ oraz } k = 1, \dots, K \quad (4.164)$$

$$x_{ij}^k = \begin{cases} 0 \\ 1 \end{cases} \quad \text{dla } i, j = 0, 1, \dots, n \text{ oraz } k = 1, \dots, K \quad (4.165)$$

W powyższym modelu programowania liniowego funkcja celu wyrażona wzorem (4.156) przedstawia minimalizację całkowitego kosztu (lub czasu) związanego z obsługą wszystkich punktów przez wszystkie bazy. Ograniczenia wyrażone wzorami (4.157)–(4.159) zapewniają, że każdy punkt obsługi odwiedzany jest tylko jeden raz (pojazd do klienta przyjedzie i go opuści) i tylko przez jeden pojazd. Zespół nierówności (4.160) i (4.161) ma zapewnić, że każdy pojazd rozpoczyna i kończy kurs w bazie. Natomiast ograniczenia wyrażone nierównościami (4.162) oraz (4.163) warunkują zasoby, jakimi dysponują bazy, odnośnie ładowności pojazdów oraz ewentualne ograniczenia nakładane na trasy poszczególnych pojazdów (np. ograniczenia dotyczące maksymalnej długości tras). Wreszcie ostatnia grupa ograniczeń (4.164) zabezpiecza przed powstaniem takich tras poszczególnych pojazdów, które są cyklami skróconymi.

Uzupełnieniem tego modelu może być dodatkowe założenie dotyczące maksymalnej wydajności każdej z baz obsługujących zadany zbiór klientów. Gdy wprowadzimy oznaczenia:

z_{ij} – zmienna decyzyjna przyjmująca wartość 1, jeżeli punkt obsługi j jest przyporządkowany do i -tej bazy, lub 0, gdy nie jest do niej przyporządkowany;

V_i – maksymalna wydajność (podaż lub popyt) i -tej bazy;

model (4.156)–(4.165) może zostać uzupełniony o ograniczenie (Surekha i Sumathi, 2011):

$$\sum_{i=1}^N q_i z_{ij} \leq V_i \quad \text{dla } i = N+1, \dots, N+M \quad (4.166)$$

Inne postaci sformułowania problemu wielu komiwojazerów z wieloma bazami można znaleźć w pracy C. Contarda i R. Martinello (2014). Przedstawione tam zostały dwa różne warianty tego zagadnienia oparte na teorii grafów, nazywane odpowiednio modelem przepływu pojazdów oraz modelem podziału zbioru (*Vehicle-flow Formulation* i *Set-partitioning Formulation*). Przyjmując następujące założenia:

- E – zbiór wszystkich krawędzi grafu $G(V, E, t)$;
- V_d – zbiór baz obsługujących klientów $V_d \subset V$;
- V_c – zbiór klientów $V_c \subset V$;
- t_{ij} – nieujemna waga krawędzi e grafu G oznaczająca czas przejazdu pojazdu pomiędzy klientami lub pomiędzy klientami a bazą;
- y_{ij} – zmienna przyjmująca wartość 1, jeżeli j -ty punkt obsługi należy do trasy pojazdu wyruszającego z i -tej bazy;
- x_e – zmienna przyjmująca wartość 1, jeżeli krawędź e grafu G należy do trasy pojazdu obsługującego co najmniej dwóch klientów;
- m_i – liczba pojazdów będących w dyspozycji i -tej bazy;
- $r(S)$ – minimalna liczba pojazdów potrzebna do obsłużenia zbioru klientów S , gdzie $S \subset V$, wynikająca z ograniczenia dotyczącego ich ładowności (iloraz łącznego zapotrzebowania zbioru klientów S oraz ładowności pojazdu);
- $\rho(S)$ – minimalna liczba pojazdów potrzebna do obsłużenia zbioru klientów S , gdzie $S \subset V$, wynikająca z ograniczenia dotyczącego czasu pracy pojazdu;
- $\delta(U)$ – zredukowany zbiór wierzchołków grafu G , gdzie $U \subset V$, który jest równy podzbiorowi krawędzi z dokładnie jednym końcem trasy pojazdu;
- $x(F)$ – suma zmiennych decyzyjnych x_e dotyczących krawędzi e grafu G należących do podzbioru F , gdzie $F \subseteq E$;
- $y(F)$ – suma zmiennych decyzyjnych y_{ij} należących do podzbioru F , gdzie $F \subseteq \delta(V_d)$;

pierwszy z modeli ma następującą postać (Contardo i Martinelli, 2014):

$$\min. \sum_{e \in E} t_e x_e + 2 \sum_{i \in V_d, j \in V_c} t_{ij} y_{ij} \quad (4.167)$$

$$x(\delta\{j\}) + 2y((\delta\{j\}) \cap \delta(V_d)) = 2 \quad j \in V_c \quad (4.168)$$

$$x(\delta\{i\}) + 2y(\delta\{i\}) \leq 2m_i \quad j \in V_d \quad (4.169)$$

$$x(\delta\{S\}) + 2y((\delta\{S\}) \cap \delta(V_d)) \geq 2 \max. \{r(S), \rho(S)\} \quad S \subseteq V_c \quad (4.170)$$

$$x(\delta\{S\}) \geq 2[x(\delta(\{h\}) \cap \delta\{V'_d\}) + x(\delta(\{j\}) \cap \delta\{V_d \setminus V'_d\})] \quad S \subseteq V_c; j \in S; V'_d \subseteq V_d \quad (4.171)$$

$$x_e \in \{0, 1\} \quad e \in \delta(V_d) \quad (4.172)$$

$$y_e \in \{0, 1\} \quad e \in E \quad (4.173)$$

W powyższym modelu funkcja celu minimalizuje całkowity czas podróży pojazdów obsługujących klientów będących w zasięgu wszystkich baz. Ograniczenia (4.168) dotyczą stopnia wierzchołka grafu G , co oznacza, że każdy punkt obsługi może zostać odwiedzony tylko jeden raz. Ograniczenia (4.169) odnoszą się do rozmiarów floty pojazdów pozostającej w dyspozycji każdej z baz. Ograniczenia (4.170) reprezentują warunki wynikające z ładowności pojazdów oraz dopuszczalnej długości trasy pojazdów. Natomiast ograniczenia (4.171) zapewniają, że trasa każdego pojazdu rozpoczyna i kończy się w tej samej bazie. Wreszcie warunki brzegowe (4.172) i (4.173) określają binarność zmiennych decyzyjnych modelu.

Do sformułowania drugiego modelu dla problemu wielu komiwojażerów z wieloma bazami – modelu podziału zbiorów – przyjęto następujące założenia:

- y_{ij} – zmienna przyjmująca wartość 1, jeżeli j -ty punkt obsługi należy jako jedyny do trasy pojazdu wyruszającego z i -tej bazy (czas trasy pojazdu wynosi $2t_{ij}$);
- Ω – zbiór wszystkich tras pojazdów, w których odwiedzane są co najmniej dwa punkty obsługi, a także nie są przekroczone ograniczenia dotyczące ładowności oraz czasu pracy pojazdów;
- Ω_i – podzbiór tych tras pojazdów, gdzie $\Omega_i \subset \Omega$, których początek i koniec stanowi i -ta baza;
- θ_l – zmienna decyzyjna przyjmująca wartość 1, jeśli trasa l jest rozpatrywana w rozwiązaniu, oraz 0, gdy nie jest rozpatrywana (koszt tej trasy to t_l);
- a_j^l – liczba oznaczająca, ile razy j -ty punkt obsługi znajduje się w trasie l (dla każdego $j \in V_c$);
- $y(D : C)$ – suma zmiennych decyzyjnych y_{ij} , gdzie $i \in D$ oraz $j \in C$ dla każdego podzbioru $D \subseteq V_d$ oraz $C \subseteq V_c$.

Model podziału zbiorów dla zagadnienia układania tras pojazdów z wieloma bazami sformułowany jest w następujący sposób (Contardo i Martinelli, 2014):

$$\min. \sum_{l \in \Omega} t_l \theta_l + 2 \sum_{i \in V_d, j \in V_c} t_{ij} y_{ij} \quad (4.174)$$

$$\sum_{l \in \Omega} a_j^l \theta_l + y(V_d : \{j\}) = 1 \quad j \in V_d \quad (4.175)$$

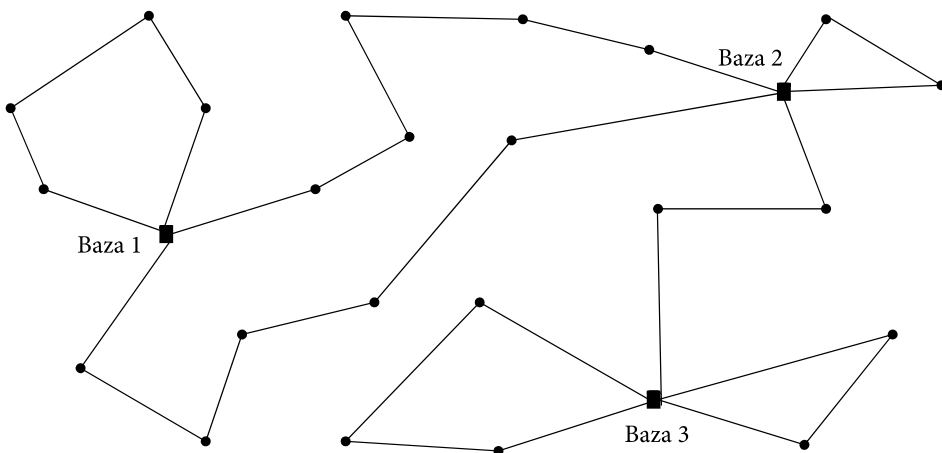
$$\sum_{l \in \Omega_i} \theta_l + y(\{j\} : V_c) \leq m_i \quad i \in V_d \quad (4.176)$$

$$y_{ij} \in \{0, 1\} \quad j \in \delta(V_d) \quad (4.177)$$

$$\theta_l \in \{0, 1\} \quad l \in \Omega \quad (4.178)$$

Podobnie jak w poprzednim modelu (4.167)–(4.173), funkcja celu reprezentuje minimalizację całkowitego czasu podróży pojazdów obsługujących klientów będących w zasięgu wszystkich baz. Zespół ograniczeń (4.175) zapewnia, że każdy punkt obsługi odwiedzany jest tylko jeden raz. Tak jak zespół ograniczeń (4.169), tak i ograniczenia (4.176) dotyczą rozmiarów floty pojazdów pozostającej w dyspozycji każdej z baz. Warunki brzegowe (4.177) i (4.178) określają binarność zmiennych decyzyjnych modelu. Należy tutaj zauważyć, że w modelu nie ma ograniczeń dotyczących ładowności oraz dopuszczalnych długości tras pojazdów, ponieważ warunki te są zachowane w definicji (założeniu) podzbioru tras Ω . To samo dotyczy warunku rozpoczęcia i kończenia trasy przez pojazd w tej samej bazie.

Przedstawione powyżej modele układania tras pojazdów z wieloma bazami odnosiły się do sytuacji, w której pojazd wyruszał z bazy w celu obsługi zadanego zbioru klientów w trasie, a następnie powracał do tej samej bazy. Jednakże można spotkać się też z sytuacją, w której warunek powrotu do tej samej bazy nie musi być spełniony. Oznacza to, że pojazd może zakończyć trasę w dowolnej bazie (rys. 4.17).



Rys. 4.17. Przykładowe rozwiązanie problemu wielu komiwojazerów z wieloma bazami, w którym pojazd nie musi wracać do bazy, z której wyruszył

Źródło: opracowanie własne.

Tak sformułowany problem wielu komiwojazerów z wieloma bazami jest podobny do przedstawionego wcześniej zagadnienia układania tras pojazdów z uzupełnieniami ładunku. Jednakże w tamtym wariantcie pojazdy wyruszające z bazy mogły odwiedzić punkty uzupełnienia ładunku w dowolnym momencie, a także powracały zawsze do tej samej bazy. W tym wypadku różne bazy mogą stanowić punkty początkowe i końcowe dla tras pojazdów. Należy także zauważyć, że ten problem może stanowić szczególny przypadek problemu wielu komiwojazerów z uzupełnieniami ładunku – np. gdy pojazd (pusty) odwiedza bazę w celu

uzupełnienia ładunku i ponownego wyruszenia w trasę. Jednakże można taki przypadek traktować również jako rozpoczęcie nowej trasy.

Problem wielu komiwojażerów z wieloma bazami, z których jedna nie musi być początkiem i końcem tej samej trasy, został też sformułowany w języku programowania matematycznego (Crevier i in., 2007). W opisanym przez autorów modelu trasa h pojazdu reprezentowana jest przez zbiór punktów obsługi, które ona zawiera. Stąd też zdefiniowane zostały dwie wielkości: e_{ih} oraz f_{hl} . Pierwsza może przyjmować wartości binarne: 1, jeżeli i -ty punkt obsługi należy do trasy h , oraz 0, gdy do niego nie należy. Natomiast drugi z parametrów może przyjmować trzy wartości: 0, 1 lub 2. Parametr f_{hl} będzie miał wartość 2, jeżeli trasa h rozpoczyna się i kończy w tej samej bazie. Wartość 1 oznaczać będzie przypadek, w którym trasa h rozpoczyna się w jednej, a kończy w innej bazie, natomiast 0, gdy np. baza stanowi punkt pośredni trasy w celu uzupełnienia ładunku.

Podstawą sformułowania tego modelu także może być graf $G(V, A, c)$, przyjęto zatem następujące założenia:

- A – zbiór wszystkich krawędzi grafu $G(V, E)$;
- V_d – zbiór r baz obsługujących klientów $V_d \subset V$;
- V_c – zbiór n klientów $V_c \subset V$;
- q_i – popyt (podaż) i -tego punktu obsługi;
- d_i – czas potrzebny na obsługę i -tego punktu obsługi;
- c_{ij} – koszt transportu pomiędzy punktami i oraz j , a także pomiędzy i -tym punktem obsługi a j -tą bazą;
- m – liczba pojazdów;
- Q – ładowność pojazdu;
- D – całkowity czas pracy pojazdu;
- T – zbiór wszystkich h tras spełniających ograniczenia związane z ładownością pojazdu oraz rozpoczynających się i kończących w tej samej lub innej bazie;
- x_h^k – zmienna decyzyjna przyjmująca wartość 1, jeżeli trasa h została przyporządkowana do pojazdu k , oraz 0, gdy nie została przyporządkowana;
- y_l^k – zmienna przyjmująca wartość 1, jeżeli zbiór tras przyporządkowanych do pojazdu k rozpoczyna się w bazie l , oraz 0, gdy się w niej nie rozpoczyna (pojazd uzupełnia ładunek i kontynuuje trasę);
- z_l^k – zmienna całkowitoliczbowa oznaczająca, ile razy k -ty pojazd przybywa i opuszcza bazę l ;
- π_h – czas trwania podróży dla trasy h ;
- μ_h – czas pracy pojazdu na trasie h uwzględniający dodatkowo czas obsługi punktów należących do tej trasy;
- I – zbiór tras, w którym trasa rozpoczyna się i kończy w innych bazach, przy czym $I \subseteq T$;

$\Delta(S)$ – zbiór tras, w którym trasa rozpoczyna się i kończy w bazach należących do zbioru S , przy czym $S \subseteq V_d$ oraz $\Delta(S) \subseteq T$;

$\Psi(S)$ – zbiór tras, w którym jedna baza (początkowa lub końcowa) należy do zbioru S , a druga do niego nie należy, przy czym $S \subseteq V_d$ oraz $\Psi(S) \subseteq T$.

Dla tak przedstawionych założeń postać modelu problemu wielu komiwojazerów z wieloma bazami, w którym trasy mogą się rozpoczynać i kończyć w różnych bazach, jest następująca:

$$\min. F(\mathbf{x}) = \sum_{k=1}^m \sum_{h=1}^{|T|} \pi_h x_h^k \quad (4.179)$$

$$\sum_{k=1}^m \sum_{h=1}^{|T|} e_{ih} x_h^k = 1 \quad \text{dla } i = 1, \dots, n \quad (4.180)$$

$$\sum_{l=1}^r y_l^k \leq 1 \quad \text{dla } k = 1, \dots, m \quad (4.181)$$

$$\sum_{h \in I} f_{hl} x_h^k - 2z_l^k = 0 \quad \text{dla } k = 1, \dots, m \text{ oraz } l = 1, \dots, r \quad (4.182)$$

$$\sum_{h=1}^{|T|} \mu_h x_h^k \leq D \quad \text{dla } k = 1, \dots, m \quad (4.183)$$

$$\sum_{h \in \Delta(S)} x_h^k \leq |\Delta(S)| \left(\sum_{h \in \Psi(S)} x_h^k + \sum_{l \in S} y_l^k \right) \quad \text{dla } k = 1, \dots, m \text{ oraz } \forall S \subseteq V_d \quad (4.184)$$

$$x_h^k \in \{0, 1\} \quad \forall h \text{ oraz } \forall k \quad (4.185)$$

$$y_l^k \in \{0, 1\} \quad \forall k \text{ oraz } \forall l \quad (4.186)$$

$$z_l^k \in C \quad \forall k \text{ oraz } \forall l \quad (4.187)$$

Zadaniem funkcji celu (4.179) jest minimalizacja całkowitego czasu jazdy wszystkich użytych pojazdów dostępnych we wszystkich bazach. Ograniczenie (4.180) stanowi gwarancję, że każdy punkt obsługi zostanie odwiedzony tylko jeden raz. Nierówności (4.181) zapewniają, że każdy pojazd zostanie użyty najwyżej jeden raz. Warunek wyrażony zespołem nierówności (4.182) zapewnia, że jeżeli pojazd przybędzie do bazy w celu uzupełnienia ładunku, to ją następnie opuści. Warunki wyrażone nierównościami (4.183) stanowią założenie, że czas pracy każdego pojazdu nie może przekroczyć ustalonej wielkości D . Natomiast nierówności (4.184) decydują o eliminacji z dopuszczalnego rozwiązania tras zawierających skrócone cykle.

Na koniec rozważań dotyczących zagadnienia układania tras pojazdów z wieloma bazami warto jeszcze wspomnieć o polskim wkładzie w rozwój wiedzy na ten temat, a w szczególności o modelu zaprezentowanym przez A. Całczyńskiego. Co prawda charakteryzuje on tylko zasięg działania każdej z baz, jednakże

uwzględnia dodatkowo sytuację, w której równie ważna jest optymalizacja kosztów ponoszonych z tytułu produkcji (jeżeli bazami są zakłady produkcyjne) bądź magazynowania (jeżeli bazami są magazyny). Takie ujęcie jest uzasadnione, ponieważ gdy przedsiębiorstwo generuje wysokie jednostkowe koszty produkcji (lub magazynowania), spadek kosztów związanych z racjonalizacją systemu transportowego może czasami prowadzić do niczym nieuzasadnionego wzrostu całkowitych kosztów produkcji (magazynowania). Ponadto uwzględniona została w tym modelu wieloasortymentowość oferty⁴.

Do sformułowania owego modelu przyjęte zostały następujące założenia:

- m – liczba baz, którymi dysponuje przedsiębiorstwo;
- n – liczba punktów obsługi (odbioru towaru) zlokalizowanych w zasięgu działania wszystkich baz;
- r – liczebność asortymentu towarów produkowanych lub magazynowanych we wszystkich bazach;
- s – liczba środków transportu będących w dyspozycji przedsiębiorstwa;
- a_i – zdolność produkcyjna bądź magazynowa i -tej bazy, dla $i = 1, \dots, m$;
- u_{ik} – współczynnik, który wyraża wielkość zaangażowania zdolności produkcyjnej bądź magazynowej i -tej bazy w celu wytworzenia lub magazynowania jednostki dobra z k -tego asortymentu, dla $i = 1, \dots, m$ oraz $k = 1, \dots, r$;
- b_{jk} – wielkość zapotrzebowania j -tego punktu obsługi (odbioru) na asortyment o numerze k , dla $j = 1, \dots, n$ oraz $k = 1, \dots, r$;
- x_{ij}^k – ilość dobra k -tego asortymentu transportowanego od i -tej bazy do j -tego punktu odbioru, dla $i = 1, \dots, m, j = 1, \dots, n$ oraz $k = 1, \dots, r$; dla $j = 0$ x_{ij}^k oznacza wielkość niewykorzystanych zdolności produkcyjnych bądź magazynowych dobra k -tego asortymentu w i -tej bazie;
- f_{ik} – jednostkowy koszt produkcji bądź magazynowania dobra k -tego asortymentu w i -tej bazie, dla $i = 1, \dots, m$ oraz $k = 1, \dots, r$;
- c_{ij}^l – koszt przejazdu samochodu l -tego typu od i -tej bazy do j -tego punktu obsługi, dla $l = 1, \dots, s, i = 1, \dots, m$ oraz $j = 0, 1, \dots, n$;
- y_{ij}^l – liczba kursów samochodów l -tego typu przewożących towar pomiędzy i -tą bazą a j -tym punktem odbioru, dla $l = 1, \dots, s, i = 1, \dots, m$ oraz $j = 0, 1, \dots, n$;
- q^l – maksymalna ładowność pojazdu l -tego typu;
- y^l – liczba kursów, jakie należy wykonać samochodami l -tego typu.

Zaprezentowany model dla rejonizacji działania poszczególnych baz ma następującą postać (Całczyński, 1979; 1992):

$$\min. Z = \sum_{l=1}^s \sum_{i=1}^m \sum_{j=1}^n c_{ij}^l + \sum_{i=1}^m \sum_{k=1}^r f_{ik} \sum_{j=1}^n x_{ij}^k \quad (4.188)$$

4 Dla układania konkretnych tras poszczególnych pojazdów istnieje osobny model.

$$\sum_{k=1}^r \sum_{j=0}^n u_{jk} x_{ij}^k = a_i \quad \text{dla } i = 1, \dots, m \quad (4.189)$$

$$\sum_{k=1}^r x_{ij}^k = b_{jk} \quad \text{dla } j = 0, 1, \dots, n; \text{ oraz } k = 1, \dots, r \quad (4.190)$$

$$\sum_{k=1}^r x_{ij}^k \leq \sum_{l=1}^s q^l y_{ij}^l \quad \text{dla } i = 1, \dots, m; j = 0, 1, \dots, n \quad (4.191)$$

$$b_{ij}^k \geq 0 \quad \text{dla } i = 1, \dots, m; j = 0, 1, \dots, n; k = 1, \dots, r \quad (4.192)$$

$$\sum_{i=1}^m \sum_{j=1}^n y_{ij}^l \leq y^l \quad \text{dla } l = 1, \dots, s \quad (4.193)$$

$$y_{ij}^l \geq 0 \quad \text{oraz } y_{ij}^l \in C \quad (4.194)$$

W powyższym modelu funkcja celu (4.188) przedstawia sumę kosztów transportu oraz produkcji (bądź magazynowania). Ograniczenia (4.189) i (4.190) określają odpowiednio podaż baz oraz popyt punktów obsługi na konkretny asortyment dobra. Pozostałe ograniczenia wyznaczają możliwości transportowe, jakimi dysponują bazy (np. ładowność pojazdu).

Rozwiązanie zadania uzyskane na podstawie przedstawionego powyżej modelu da odpowiedź na pytanie, które punkty obsługi będą przyporządkowane do konkretnej bazy. Za sprawą rozwiązania zadania opartego na przedstawionym modelu rejonizacji baz przedsiębiorstwa możemy określić dla każdej z nich również inne parametry, do których należą: liczba rozwożonych asortymentów, podaż każdego asortymentu w bazie, łączny popyt na dany asortyment punktów obsługi przyporządkowanych do bazy, popyt obsługiwanego punktu na dany asortyment, typy pojazdów niezbędnych dla danej bazy oraz ładowność każdego z nich.

Rozdział 5

Metody dokładne układania tras pojazdów

Wśród metod dokładnych zaproponowanych dla rozwiązania problemu jednego oraz wielu komiwojazerów znakomitą większością stanowią algorytmy, których konstrukcja opiera się na metodzie podziału i ograniczeń. Metody podziału i ograniczeń stanowią powszechnie stosowane sposoby rozwiązywania zadań programowania całkowitoliczbowego, do których należą także zagadnienia układania tras pojazdów.

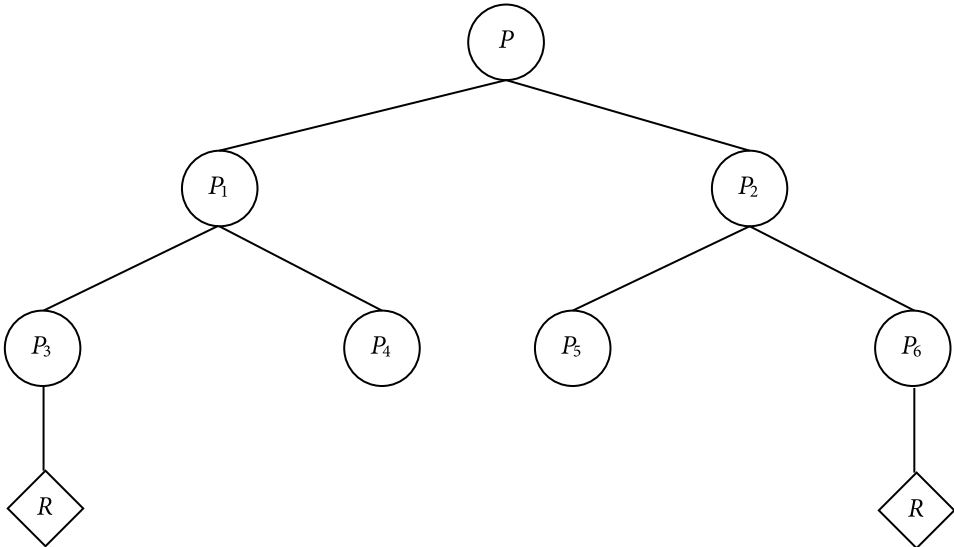
5.1. Podstawowe założenia metody podziału i ograniczeń

Ogólna idea metody podziału i ograniczeń opiera się na założeniu, że dany problem optymalizacyjny można przedstawić w postaci skończonego zbioru (listy) podproblemów. Konstrukcja podproblemów bazuje na podziale przestrzeni rozwiązań problemu wyjściowego. Podproblemy optymalizacyjne mogą zawierać zbiór rozwiązań dopuszczalnych, niedopuszczalnych, ale także mogą w ogóle nie posiadać rozwiązania i wtedy są usuwane z listy. Rozpatrywane w toku postępowania algorytmu są tylko problemy optymalizacyjne, których rozwiązanie jest dopuszczalne lub niedopuszczalne.

Istotą metody podziału i ograniczeń jest sposób tworzenia listy problemów optymalizacyjnych oraz sposób – a raczej kolejność – rozpatrywania (rozwiązywania) problemów optymalizacyjnych, które nie są usuwane z listy. W tym celu dla problemów optymalizacyjnych z minimalizacją funkcji celu obliczane są tzw. dolne granice (dolne oszacowania) każdego z podproblemów. Oznaczają one oszacowania z dołu wartości funkcji celu, czyli rozwiązanie dopuszczalne danego podproblemu będzie miało wartość funkcji celu nie mniejszą niż ustalona dolna granica.

Podział problemu wyjściowego na kolejne podproblemy następuje do momentu, kiedy zostanie uzyskane rozwiązanie dopuszczalne problemu głównego, którego wartość funkcji celu (w wypadku minimalizacji) jest nie większa niż najniższa dolna granica niepodzielonych dotychczas dalej podproblemów optymalizacyjnych.

Graficzną prezentacją metody podziału i ograniczeń jest drzewo, w którym każdy wierzchołek stanowi problem optymalizacyjny (rys. 5.1). Wierzchołek początkowy drzewa (korzeń drzewa) reprezentuje główny problem optymalizacyjny, który ma zostać rozwiązany przez decydenta. Pozostałe węzły drzewa dzielą się na dwa rodzaje. Pierwszy to węzły pośrednie przedstawiające podproblemy optymalizacyjne (zawierające rozwiązania niedopuszczalne) poddawane dalszym podziałom. Natomiast drugi rodzaj to węzły nazywane liśćmi drzewa, przedstawiające podproblemy optymalizacyjne, których rozwiązania mają charakter dopuszczalny i te nie podlegają dalszym podziałom.



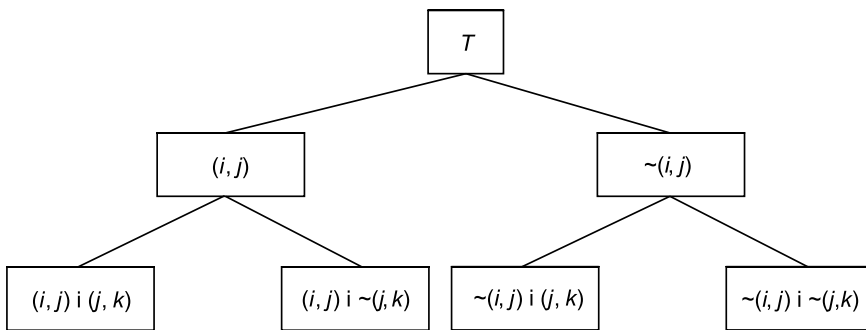
Rys. 5.1. Drzewo metody podziału i ograniczeń

Źródło: opracowanie własne.

Przykładem realizacji strategii poszukiwania rozwiązania optymalnego metodą podziału i ograniczeń dla problemu zarówno jednego, jak i wielu komiwojazerów może być algorytm, w którym podział na poszczególne podproblemy odbywa się w myśl zasady: połączenie (i, j) należy do trasy pojazdu lub połączenie (i, j) nie należy do trasy pojazdu (rys. 5.2). Gałąź drzewa tworzona będzie do momentu, aż będzie reprezentować trasę dopuszczalną (w wypadku problemu wielu komiwojazerów – trasy dopuszczalne), czyli wszystkie punkty trasy zostaną uwzględnione. Nie wszystkie gałęzie drzewa muszą być rozwijane

(zakończone liśćmi) aż do uzyskania rozwiązania dopuszczalnego, ponieważ w trakcie ich budowy może się okazać, że będą one reprezentować rozwiązania gorsze od dotychczas znalezionych. Ta właśnie cecha stanowi zaletę metod podziału i ograniczeń, pozwala bowiem na eliminację z dalszych rozważań grup (gałęzi) potencjalnych rozwiązań, które nie rokują optymistycznie. Ale też należy pamiętać, że w najgorszym razie rozwinięte do liści mogą zostać wszystkie możliwe gałęzie drzewa, co będzie oznaczało wygenerowanie wszystkich możliwych rozwiązań. W takiej sytuacji efektywność metody podziału i ograniczeń będzie taka sama jak metody polegającej na sukcesywnym przeglądaniu i ocenie wszelkich możliwych kombinacji tras.

Ocena jakości potencjalnych rozwiązań w trakcie konstrukcji drzewa metody podziału i ograniczeń odbywa się przez obliczanie dla każdego węzła drzewa dolnej granicy, czyli minimalnej długości, jaką osiągnie trasa należąca do grupy tras reprezentowanej przez daną gałąź drzewa.



Rys. 5.2. Drzewo metody podziału i ograniczeń dla problemu układania tras pojazdów
Źródło: opracowanie własne.

Jednym z najprostszych sposobów wyznaczania dolnego oszacowania węzła drzewa dla problemu układania tras pojazdów jest skorzystanie z właściwości dotyczącej podstawowych założeń problemu. Mówi ona, że każdy pojazd może odwiedzić dany punkt obsługi (węzeł trasy) jeden i tylko jeden raz oraz wszystkie punkty (węzły) muszą być odwiedzone. Oznacza to, że pojazd może do tego punktu wjechać jeden raz i jeden raz go opuścić. W wypadku reprezentacji problemu układania tras pojazdów w postaci grafu dla konkretnej trasy każdy jego węzeł będzie połączeniem dwóch sąsiednich krawędzi.

Suma długości dwóch dowolnych krawędzi wychodzących z tego samego węzła (sąsiednich) jest zawsze większa lub równa sumie dwóch sąsiednich krawędzi danego węzła o najmniejszych długościach. Tym samym można stwierdzić, że biorąc pod uwagę wszystkie węzły grafu, długość każdej trasy C będzie nie mniejsza niż połowa całkowitej sumy par sąsiednich krawędzi o najmniejszych długościach dla wszystkich węzłów:

$$C = \frac{1}{2} \sum_{i=1}^N (c_{ij}^a + c_{ik}^b) \quad (5.1)$$

gdzie:

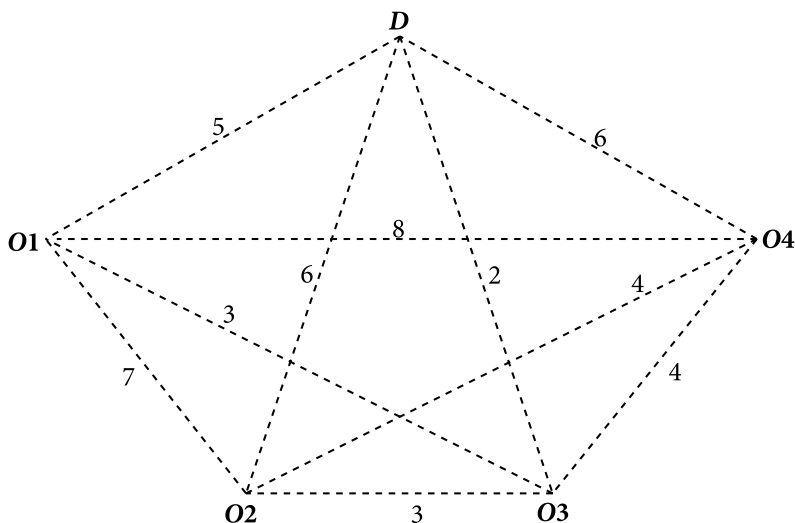
N – liczba wszystkich obsługiwanych punktów (węzłów);

c_{ij}^a, c_{ik}^b – najmniejsze długości dwóch sąsiednich krawędzi i -tego węzła.

Mając tak sformułowaną regułę podziału węzłów drzewa (zbiór tras zawiera lub nie połączenie pomiędzy i -tym a j -tym punktem obsługi) oraz dla każdego z nich sposób oszacowania dolnej granicy długości trasy, można wyznaczyć rozwiązanie optymalne problemu układania tras pojazdów.

Przykład

Na rysunku 5.3 przedstawiono nieskierowany graf dla problemu jednego komiwojażera z czterema punktami obsługi.

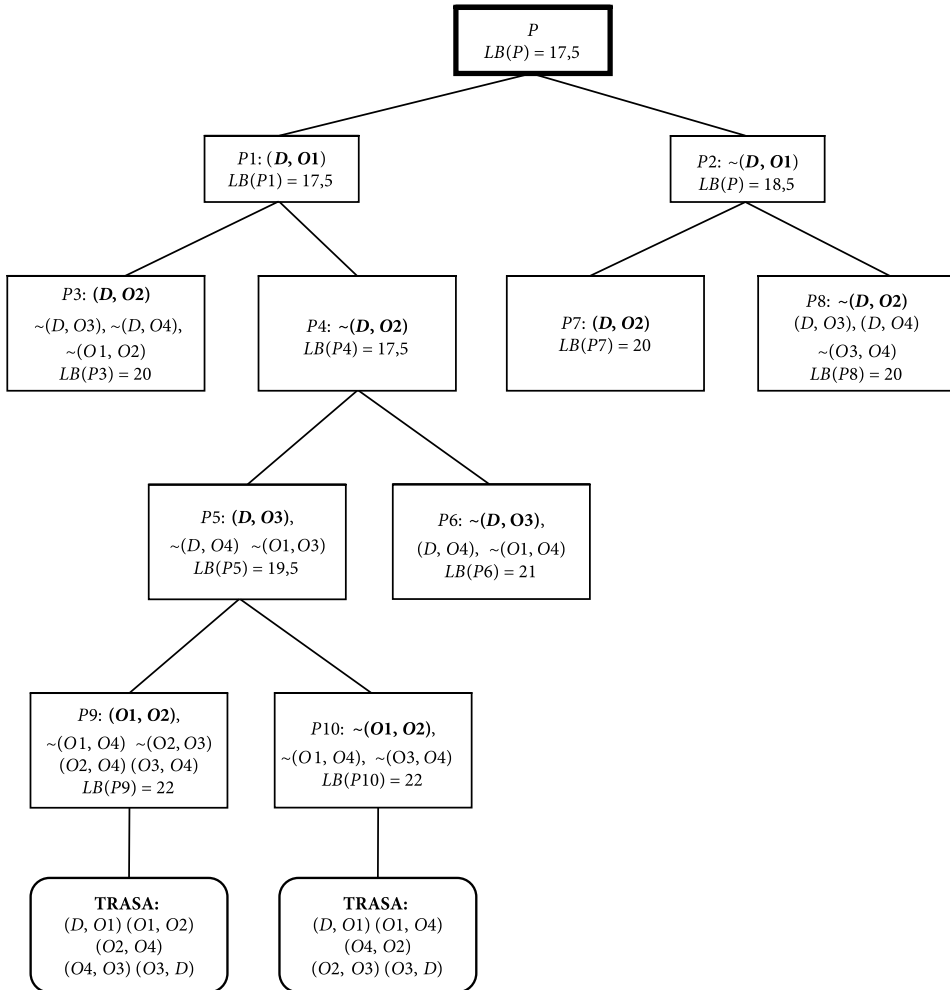


Rys. 5.3. Graf nieskierowany problemu komiwojażera dla czterech odbiorców

Źródło: opracowanie własne.

Dla problemu wyjściowego, jakim w tym wypadku jest zbiór wszystkich możliwych połączeń pomiędzy punktami obsługi oraz bazą (D), w pierwszej kolejności dla każdego wierzchołka grafu należy wyznaczyć pary krawędzi o najkrótszych długościach:

D :	$(D, O3); (D, O1)$	$2 + 5 = 7$
$O1$:	$(O1, O3); (O1, D)$	$3 + 5 = 8$
$O2$:	$(O2, O3); (O2, O4)$	$3 + 4 = 7$
$O3$:	$(O3, D); (O3, O1)$	$2 + 3 = 5$
$O4$:	$(O4, O3); (O4, O2)$	$4 + 4 = 8$



Rys. 5.4. Fragment drzewa metody podziału i ograniczeń dla czterech odbiorców
Źródło: opracowanie własne.

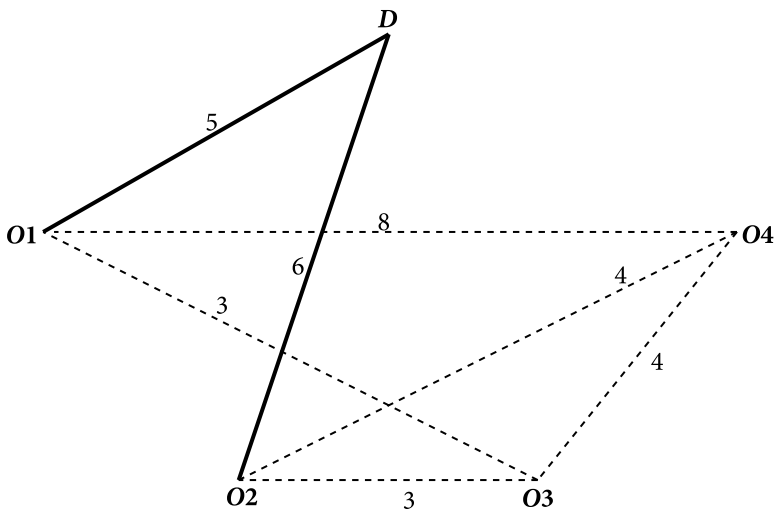
Dolna granica dla pierwszego węzła drzewa metody podziału i ograniczeń (P) zostaje obliczona w następujący sposób:

$$LB(P) = 0,5(7 + 8 + 7 + 5 + 8) = 17,5$$

Na rysunku 5.4 przedstawiono fragment drzewa zastosowanej metody podziału i ograniczeń, w której do kolejnych podziałów problemu na podproblemy wybierano krawędzie łączące wierzchołki o kolejnych numerach.

Do dalszych podziałów wybierane są podproblemy reprezentowane przez węzły drzewa o najniższych dolnych oszacowaniach. Należy zauważyć, że gdy

dokonyjemy podziału kolejnych węzłów drzewa, przez dodawanie lub usuwanie kolejnych połączeń (krawędzi grafu), może to niekiedy prowadzić do pewnych naturalnych konsekwencji, które są wynikiem podstawowych założeń dotyczących problemu układania tras pojazdów. Przykładowo węzeł $P1$ podzielony został na dwa podzbiory tras, z których jeden ($P3$) zawiera połączenie $(D, O2)$, natomiast drugi ($P4$) reprezentuje zbiór wszystkich tras niezawierający tego połączenia. Oznacza to, że w węźle $P3$ jest zbiór wszystkich tras zawierających połączenia $(D, O1)$ i $(D, O2)$. Jednocześnie trasy te nie mogą zawierać połączeń $(D, O3)$ i $(D, O4)$, ponieważ z wierzchołka D grafu mogą wychodzić tylko dwie krawędzie. Zbiór tras reprezentowany przez węzeł $P3$ drzewa nie może zawierać także połączenia $(O1, O2)$, gdyż jego wystąpienie prowadziłoby do uzyskania trasy częściowej stanowiącej cykl zamknięty, niebędący cyklem Hamiltona. Sytuacja przedstawiona powyżej zobrazowana została grafem na rysunku 5.5.



Rys. 5.5. Graf przedstawiający zbiór możliwych tras dla węzła $P3$

Źródło: opracowanie własne.

W węzłach $P9$ i $P10$ uzyskane zostały dopuszczalne trasy pojazdów, których długości wynosiły 22 każda. W związku z tym, że w węzłach: $P3$, $P6$, $P7$ oraz $P8$ dolne oszacowania wynosiły mniej niż 22, nie można uznać, że uzyskane dotychczas rozwiązania dopuszczalne problemu komiwojagera stanowią już rozwiązania optymalne. Należy więc dokonać podziału węzła $P3$ na kolejne dwa podproblemy (odpowiednio: zawierający oraz niezawierający połączenie pomiędzy odbiorcą $O1$ a odbiorcą $O3$) i obliczyć ich dolne oszacowania. Dalszemu podziałowi nie będą podlegać węzły drzewa, dla których obliczone oszacowanie wartości funkcji celu (długości całej trasy) będzie większe niż dotychczas znaleziona najlepsza trasa dopuszczalna lub będą one reprezentować trasę niedopuszczalną. Natomiast gdy nie

będziemy już mieli do dalszego podziału żadnego wężła drzewa reprezentującego częściową trasę, wtedy postępowanie można uznać za zakończone, a rozwiązaniem optymalnym będzie najlepsza dotąd znaleziona dopuszczalna trasa.

5.2. Metody dokładne a zagadnienie przydziału

Zaprezentowana powyżej metoda podziału i ograniczeń ilustruje sposób poszukiwania rozwiązania optymalnego przez rozwiązywanie pewnego uproszczonego problemu decyzyjnego i obliczanie jego dolnego oszacowania. Innymi słowy, przyjmuje się założenie, że jeżeli znalezione zostanie rozwiązanie optymalne pewnego uproszczonego problemu decyzyjnego, a jednocześnie będzie ono spełniało wszystkie postawione warunki problemu wyjściowego, to rozwiązanie to będzie także optymalne dla problemu wyjściowego.

W kontekście sformułowania problemu układania tras pojazdów w języku programowania liniowego można zauważyć, że uproszczeniem (relaksacją) problemu zarówno jednego, jak i wielu komiwojażerów jest usunięcie z modelu zbioru ograniczeń zapewniających, że uzyskane trasy pojazdów są cyklami Hamiltona. Rezultatem tego zabiegu jest otrzymanie problemu przydziału, gdzie odbiorcy (w tym także baza) są punktami, które należy przyporządkować do tego samego zbioru odbiorców. Przyporządkowanie j -tego punktu obsługi do punktu i może być interpretowane jako umieszczenie w trasie pojazdu połączenia pomiędzy punktami i oraz j . Wartość funkcji celu tak sformułowanego zagadnienia przydziału stanowi dolną granicę problemu układania tras pojazdów. Jeżeli znalezione rozwiązanie optymalne zagadnienia przydziału spełniałoby jednocześnie warunki cyklu Hamiltona, wtedy to rozwiązanie należałoby uznać także za rozwiązanie optymalne dla problemu układania tras pojazdów.

W literaturze przedmiotu zaproponowano już wiele algorytmów podziału i ograniczeń opartych na przedstawionym powyżej uproszczeniu. Do najbardziej znanych zaliczyć należy algorytm J.D.C. Little'a i in. (1963). Zanim określone zostaną zasady dotyczące podziału wężłów drzewa na kolejne wężły oraz przystąpi się do obliczania dolnego oszacowania, należy wprowadzić pojęcie standaryzacji macierzy odległości C pomiędzy punktami obsługi (w tym także bazą). Macierz standaryzowana C' jest macierzą, której elementy obliczane są wedle formuły:

$$c'_{ij} = c_{ij} - a_i - b_j \quad (5.2)$$

gdzie:

$$a_i = \min_j \{c_{ij}\} \quad \text{dla } i = 0, 1, \dots, N \quad (5.3)$$

$$b_j = \min_i \{c_{ij} - a_i\} \quad \text{dla } j = 0, 1, \dots, N \quad (5.4)$$

Dolnym oszacowaniem problemu wyjściowego P , które reprezentuje wszystkie możliwe rozwiązania, jest granica $\omega(P)$, której wartość jest równa:

$$\omega(P) = \sum_{i=0}^N a_i + \sum_{j=0}^N b_j \quad (5.5)$$

Omawiając regułę podziału węzłów drzewa, należy zauważyć, że macierz C' zawiera co najmniej N elementów równych 0. Dla każdego połączenia odpowiadającego elementowi zerowemu macierzy C' należy wyznaczyć wartość p_{kl} równą:

$$p_{kl} = \min_{j \neq l} \{c'_{kj}\} + \min_{i \neq k} \{c'_{il}\} \quad (5.6)$$

Połączenie (r, s) , dla którego wartość p_{kl} jest najwyższa, stanowi odcinek, wedle jakiego dokonywany jest podział problemu na dwa kolejne podproblemy $P1$ oraz $P2$ – odpowiednio – zawierające połączenie (r, s) i niezawierające takiego połączenia.

Oszacowanie dolne podproblemu $\omega(P1)$ pierwszego z podproblemów musi zostać ustalone po uprzednim dokonaniu standaryzacji macierzy C_1 , powstałej w wyniku wykreślenia z macierzy C' r -tego wiersza i s -tej kolumny, a także po zablokowaniu połączenia (s, r) . Jeżeli przez ω_1 oznaczymy sumę współczynników standaryzacyjnych macierzy C_1 wyznaczanych zgodnie z formułami (5.3)–(5.4), to dolna granica dla podproblemu $P1$ będzie wynosić:

$$\omega(P1) = \omega(P) + \omega_1 \quad (5.7)$$

Natomiast dolne oszacowanie $\omega(P2)$ dla drugiego z podproblemów obliczane jest według wzoru:

$$\omega(P2) = \omega(P) + p_{rs} \quad (5.8)$$

Zgodnie z ogólną zasadą dotyczącą metod podziału i ograniczeń do dalszych podziałów wybierany jest węzeł drzewa reprezentujący ten podzbiór rozwiązań $P(i)$, dla którego dolne oszacowanie $\omega(Pi)$ jest najniższe.

Innym przykładem zastosowania metody podziału i ograniczeń, w której wykorzystano relaksację z zagadnieniem przydziału, jest algorytm zaproponowany przez G. Carpaneto i P. Totha (1980). Wykorzystuje on zmodyfikowane zagadnienie przydziału, w którym liczba środków i liczba celów są sobie równe, a także są równe $N + 1$, czyli liczbie N punktów, które ma odwiedzić pojazd (włączając w to bazę). W zadaniu przydziału ustala się, że wszystkie zmienne decyzyjne x_{ii} są równe 0. Jeżeli uzyskane rozwiązanie zadania przydziału zawiera takie przyporządkowania punktu i do punktu j , że można zbudować jedną trasę pojazdu przechodzącą przez wszystkie punkty oraz bazę, to uzyskane rozwiązanie jest także rozwiązaniem

dopuszczalnym dla problemu komiwojażera. W przeciwnym razie rozwiązanie zadania przydziału przedstawia niedopuszczalne rozwiązanie z punktu widzenia problemu komiwojażera. W takim rozwiązaniu występują przynajmniej dwa cykle skrócone, co stanowi podstawę do zastosowania określonej reguły podziału wężła drzewa na kolejne wężły.

Prezentując algorytm podziału i ograniczeń z wykorzystaniem relaksacji zadaniem przydziału, należy przyjąć następujące oznaczenia:

- F^* – długość najlepszej dotychczas znalezionej trasy komiwojażera;
- F_k – wartość funkcji celu zmodyfikowanego zadania przydziału zbudowanego w k -tym wężle drzewa;
- $\omega(k)$ – dolna granica w k -tym wężle drzewa;
- A_k – zbiór włączonych do trasy połączeń pomiędzy punktami (krawędzi grafu) w zbiorze tras reprezentowanych przez k -ty wężel drzewa;
- B_k – zbiór wyłączonych z trasy połączeń pomiędzy punktami (krawędzi grafu) w zbiorze tras reprezentowanych przez k -ty wężel drzewa.

Zbiory A_k i B_k służą do generowania w danym wężle drzewa przeszukiwania nowych wężłów reprezentujących podproblemy powstałe z problemu wyjściowego. W wypadku zbioru krawędzi A_k zmienne decyzyjne zadania przydziału x_{ij} reprezentujące włączane krawędzie przyjmują wartość 1, natomiast w wypadku zbioru B_k zmienne x_{ij} reprezentujące wyłączone krawędzie przyjmują wartość 0.

Schemat algorytmu podziału i ograniczeń zaproponowanego przez Carpaneto i Totha, używając pseudokodu, można przedstawić w następujący sposób (rys. 5.6):

```

krok 1: inicjalizacja
repeat
    krok 2: wybór wężła drzewa
    krok 3: podział wężła na podproblemy
    repeat
        repeat
            krok 4: obliczenie dolnego oszacowania
        until  $\omega(k_{l1}) < F^*$ 
        krok 5: rozwiązanie  $l$ -tego podproblemu
    until  $F_{k[l]} < F^*$ 
    krok 6: sprawdzenie dopuszczalności rozwiązania
until lista zadań do podziału jest pusta
  
```

Rys. 5.6. Schemat algorytmu podziału i ograniczeń Carpaneto i Totha
Źródło: opracowanie własne.

W kroku pierwszym (inicjalizacji) algorytmu przedstawionego na rysunku 5.6 należy dokonać rozwiązania zmodyfikowanego zadania przydziału, w którym modyfikacja może polegać na przyjęciu założenia, że odległość pomiędzy i -tym

punktem obsługi a tym samym punktem wynosi ∞ , co powinno prowadzić do rozwiązania, że zmienne decyzyjne $x_{ii} = 0$. Jeżeli z uzyskanego rozwiązania można zbudować trasę komiwojażera, to postępowanie się kończy, a uzyskane rozwiązanie zadania przydziału jest rozwiązaniem optymalnym dla problemu komiwojażera. W przeciwnym razie rozwiązanie zadania przydziału będzie przedstawiać przy najmniej dwa podcykle (podtrasy), czyli niedopuszczalne rozwiązania zadania komiwojażera, co oznacza, że problem wyjściowy trzeba umieścić na liście zadań przeznaczonych do podziału. Zadanie to będzie stanowiło pierwszy wierzchołek drzewa (tzw. korzeń) metody podziału i ograniczeń.

Kolejne kroki algorytmu przeprowadza się do momentu, aż lista zadań przeznaczonych do dalszego podziału będzie pusta. Krok drugi stanowi etap wyboru zadania do dalszego podziału. Jeżeli lista ta zawiera więcej niż jedno zadanie, wybór wierzchołka drzewa w celu utworzenia nowych wierzchołków odbywa się zgodnie z zasadą najmniejszej wartości dolnego oszacowania $\omega(k)$ przyporządkowanego do k -tego wierzchołka drzewa.

Krok trzeci algorytmu przedstawia schemat podziału zadania wyjściowego reprezentowanego przez k -ty wierzchołek drzewa na kolejne wierzchołki (wierzchołki potomne): $k(1), k(2), \dots, k(r)$. Wierzchołki te reprezentować będą podproblemy określone zbiorami $A_{k(r)}$ oraz $B_{k(r)}$:

$$A_{k(r)} = \begin{cases} A_k & \text{dla } l=1 \\ A_k \cup \{(i_u, j_u) : u = 1, \dots, l-1\} & \text{dla } l=2, \dots, s \end{cases} \quad (5.9)$$

$$B_{k(r)} = \begin{cases} B_k & \text{dla } l=1 \\ B_k \cup \{(i_l, j_l)\} & \text{dla } l=1, \dots, s \end{cases} \quad (5.10)$$

Krok czwarty to określenie dolnego oszacowania dla k -tego węzła drzewa. Granica ta obliczana jest przez redukcję odpowiedniego wiersza i kolumny macierzy kosztu zadania przydziału. Jeżeli $\omega(k) > F^*$, to należy przejść do następnego wierzchołka drzewa ze zbioru wygenerowanych podproblemów, a następnie powtórzyć procedurę obliczenia jego dolnej granicy.

Jeżeli obliczona w kroku czwartym dolna granica jest większa od F^* , w kroku piątym trzeba rozwiązać podproblem związany z danym wierzchołkiem drzewa. Znalezienie rozwiązania następuje przez rozwiązanie odpowiedniego zadania przydziału (ograniczonego zbiorami A_k oraz B_k). Jeżeli wartość funkcji uzyskanego rozwiązania jest większa od F^* , należy przejść ponownie do kroku czwartego. W przeciwnym razie należy sprawdzić, czy uzyskane rozwiązanie jest dopuszczalne z punktu widzenia problemu komiwojażera, co dokonuje się w kroku szóstym algorytmu. Jeżeli rozwiązanie dla rozpatrywanego aktualnie k -tego węzła drzewa nie zawiera podcykli, to należy je przyjąć jako dotychczas najlepsze znalezione, a tym samym $F^* = F_k$. W przeciwnym razie rozpatrywany

k -ty wierzchołek drzewa reprezentować będzie zadanie, które zostanie umieszczone na liście zadań przeznaczonych do dalszego podziału. Należy do tej listy powrócić, aby dokonać wyboru kolejnego zadania, z którego stworzone zostaną nowe podproblemy.

Zagadnienie przydziału stanowi także relaksację dla zastosowanej przez D.L. Millera i J.F. Pekny'ego (1991) metody podziału i ograniczeń. Tam jednak rozważano dualne zadanie przydziału (5.11)–(5.12):

$$\min F(DZP) = \sum_{i=0}^N u_i + \sum_{j=0}^N v_j \quad (5.11)$$

$$c_{ij} - u_i - v_j \geq 0 \quad \text{dla } i, j = 0, 1, \dots, N \text{ oraz } i \neq j \quad (5.12)$$

Oznaczając przez $F^*(ZP)$ optymalną wartość funkcji celu zadania przydziału oraz przez $F^*(DZP)$ optymalną wartość funkcji celu dualnego zadania przydziału, należy zauważyć, że $F^*(ZP) = F^*(DZP)$. Ponadto jeżeli w rozwiązaniu zadania przydziału znajduje się połączenie pomiędzy punktami i oraz j , to dolna granica dla takiego zadania przydziału wynosi $F^*(ZP) + (c_{ij} - u_i - v_j)$.

Zaproponowany przez tych autorów algorytm dla problemu komiwojażera, wykorzystujący relaksację za pomocą dualnego zadania przydziału, został pokazany na rysunku 5.7.

krok 1: inicjalizacja
repeat
 krok 2: rozwiązanie zadania TSP'
 krok 3: sprawdzenie warunku stopu
until warunek stopu jest spełniony

Rys. 5.7. Schemat algorytmu podziału i ograniczeń D.L. Millera i J.F. Pekny'ego
Źródło: opracowanie własne.

W pierwszym kroku algorytmu (inicjalizacji) autorzy wprowadzają pewien parametr λ , a następnie dokonują modyfikacji wyjściowego problemu komiwojażera do problemu, w którym eliminowane są wszystkie połączenia o koszcie przekraczającym ustaloną wartość λ . Tym samym ze zmodyfikowanym problemem komiwojażera związane są odpowiednio problemy zadania przydziału oraz dualnego zadania przydziału, gdzie:

$$c'_{ij} \begin{cases} c_{ij} & \text{dla } c_{ij} \leq \lambda \\ \infty & \text{dla } c_{ij} \geq \lambda \end{cases} \quad (5.13)$$

Przyjmując oznaczenia:

- F^* – długość trasy rozwiązania optymalnego problemu komiwojażera;
- $F^*(\text{mod.})$ – długość trasy rozwiązania optymalnego zmodyfikowanego problemu komiwojażera;
- $F^*(\text{ZP})$ – optymalna wartość funkcji celu zadania przydziału;
- $u_i^{*(\text{mod.})}, v_j^{*(\text{mod.})}$ – rozwiązanie optymalne dualnego zadania przydziału powiązanego ze zmodyfikowanym problemem komiwojażera;
- $v_{\max}^{*(\text{mod.})}$ – wartość maksymalna ze wszystkich elementów $v_j^{*(\text{mod.})}$

autorzy udowadniają, że rozwiązanie optymalne dla zmodyfikowanego wzorem (5.13) zadania komiwojażera jest także rozwiązaniem optymalnym dla wyjściowego problemu komiwojażera, jeżeli spełnione są dwie nierówności: (5.14) i (5.15):

$$F^*(\text{mod.}) - F^*(\text{ZP}) \leq \lambda + 1 - u_i^{*(\text{mod.})} - v_{\max}^{*(\text{mod.})} \quad \text{dla } i = 0, 1, \dots, N \quad (5.14)$$

$$\lambda + 1 - u_i^{*(\text{mod.})} - v_{\max}^{*(\text{mod.})} \geq 0 \quad \text{dla } i = 0, 1, \dots, N \quad (5.15)$$

W drugim kroku algorytmu rozwiązywany jest zmodyfikowany problem komiwojażera, w którym budowane są związane z nim problemy, odpowiednio: zadania przydziału i dualnego zadania przydziału. W tym celu wykorzystywana jest metoda podziału i ograniczeń z zastosowaniem relaksacji skonstruowanym wcześniej zagadnieniem przydziału. W kroku trzecim algorytmu należy sprawdzić warunek końca algorytmu. Jeżeli nierówności (5.14) i (5.15) są spełnione, oznacza to, że uzyskane rozwiązanie optymalne zmodyfikowanego problemu komiwojażera jest także optymalnym rozwiązaniem wyjściowego problemu komiwojażera. Jeżeli te warunki nie są spełnione, należy podwoić wartość parametru λ , a następnie przejść do kroku drugiego algorytmu.

Bardzo istotny jest tutaj parametr λ . Według autorów zaproponowanego algorytmu powinien być on odpowiednio dobrany. Sugerują oni, że jeżeli np. wartość λ jest równa największemu kosztowi połączenia pomiędzy punktami i oraz j w rozwiązaniu uzyskanym metodą heurystyczną, to nie ma zazwyczaj potrzeby wykonywania drugiej iteracji algorytmu.

W kontekście zagadnienia układania tras wielu pojazdów także zaproponowano zastosowanie metody podziału i ograniczeń wykorzystującej relaksację zagadnieniem przydziału (Laporte i in., 1986) – przez naturalne rozszerzenie problemu jednego komiwojażera do zagadnienia m -komiwojażerów. Przy ustalonej górnej granicy liczby pojazdów K problem m -komiwojażerów może zostać przekształcony w problem jednego komiwojażera. Można tego dokonać przez budowę nowego grafu $G'(V', E', c')$, w którym będzie $K - 1$ kopii wierzchołka grafu reprezentującego bazę. Tym samym liczebność zbioru $V' = \{1, \dots, n'\}$ wszystkich węzłów w grafie jest równa $n' = n + K - 1$. Natomiast zbiór krawędzi grafu jest zdefiniowany jako:

$A' = A \cup \{(i, j) : i, j \in V', i \neq j, i \neq j, i \text{ lub } j \in V \setminus V\}$. Macierz odległości C' związana z długościami krawędzi zbioru A' jest równa:

$$c'_{ij} = \begin{cases} c_{ij} & (i, j \in V) \\ c_{i1} & (i \in V \setminus \{1\}, j \in V \setminus V) \\ c_{1i} & (i \in V \setminus V, j \in V \setminus \{1\}) \\ \gamma & (i, j \in (V \setminus V) \cup \{1\}) \end{cases} \quad (5.16)$$

gdzie γ przyjmuje wartość nieskończoności, jeżeli rozpatrywany jest najczęściej występujący wariant problemu układania tras pojazdów, w którym poszukiwany jest zbiór tras minimalizujący ich łączną długość. Gdy funkcją celu jest natomiast minimalizacja najdłuższej trasy pojazdu, γ przyjmuje wartość 0, a jeżeli celem jest poszukiwanie rozwiązania angażującego jak najmniejszą liczbę pojazdów, to $\gamma = -\infty$.

Tym samym problem układania tras pojazdów może zostać sformułowany w postaci modelu:

$$\min. F(\mathbf{x}) = \sum_{i \neq j} c'_{ij} x_{ij} \quad (5.17)$$

$$\sum_{j=1}^{n'} x_{ij} = 1 \quad \text{dla } i = 1, \dots, n' \quad (5.18)$$

$$\sum_{i=1}^{n'} x_{ij} = 1 \quad \text{dla } j = 1, \dots, n' \quad (5.19)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - \nu(S) \quad \text{dla } S \subset V' \setminus \{1\} \text{ oraz } |S| \geq 2 \quad (5.20)$$

$$x_{ij} \in \{0, 1\} \quad \text{dla } i, j = 1, \dots, n' \text{ oraz } i \neq j \quad (5.21)$$

W powyższym modelu równania (5.17)–(5.19) oraz (5.21) prezentują zmodyfikowane zagadnienie przydziału, w którym modyfikacja polega na zastąpieniu elementów na głównej przekątnej macierzy kosztów C' liczbami uniemożliwiającymi przydział i -tego punktu do punktu i . Ograniczenia wyrażone zespołem nierówności (5.20) eliminują powstawanie podcykli tras. Wyjaśnienia wymaga tutaj wielkość $\nu(S)$, która oznacza dolną granicę liczby pojazdów niezbędnych do odwiedzenia wszystkich zadanych punktów obsługi w rozwiązaniu optymalnym. Wartość, jaką ona przyjmuje, uzależniona jest od typu zagadnienia układania tras pojazdów. Dla problemu wielu komiwojażerów z ograniczeniami zasobowymi (określoną taką samą ładownością wszystkich pojazdów Q) może ona przyjąć wartość:

$$\nu(S) = \left\lfloor \frac{\sum_{i \in S} q_i}{Q} \right\rfloor \quad \text{dla } i, j = 1, \dots, n' \text{ oraz } i \neq j \quad (5.22)$$

gdzie q_i oznacza popyt zgłaszany przez klienta w i -tym punkcie obsługi. Oznacza to, że powyższe ograniczenie pełni podwójną funkcję. Nie tylko zapewnia brak podcykli niebędących cyklami Hamiltona, ale również pozwala spełnić warunek nieprzekraczania dopuszczalnej ładowności wykorzystywanych pojazdów.

Algorytm podziału i ograniczeń dla problemu układania tras pojazdów jest bardzo podobny do algorytmu opracowanego dla problemu komiwojażera (zob. rys. 5.6). W zbliżony sposób są w nim formułowane i rozwiązywane kolejne podproblemy będące odpowiednio zmodyfikowanymi zagadnieniami przydziału. Cechą różnicującą algorytm podziału i ograniczeń dla zagadnienia wielu komiwojażerów od algorytmu dla jednego komiwojażera jest sposób określania niedopuszczalnych tras pojazdów. W zadaniu wielu komiwojażerów, oprócz niedopuszczalności tras wynikającej z możliwości wystąpienia podcykli, niedopuszczalność trasy definiuje także przekroczenie założonej ładowności pojazdu. Takie trasy składają się z ciągu punktów obsługi (i_1, i_2, \dots, i_r) , gdzie $i_1, i_r \in \{1, n+1, \dots, n+K-1\}$, natomiast $i_2, i_{r-1} \in V \setminus \{1\}$ oraz zachodzi nierówność $\sum_{i=2}^{r-1} q_i > Q$.

Interpretacja uzyskanego rozwiązania jako rozwiązania dla zadania układania tras pojazdów z ograniczeniami zasobowymi musi się odbywać z zastosowaniem następujących reguł. Jeżeli dla dowolnej krawędzi (i, j) uzyskanego rozwiązania punkt i należy do zbioru $V \setminus \{1\}$ oraz punkt j należy do zbioru $V' \setminus V$, to należy dokonać zamiany połączenia (i, j) na połączenie $(i, 1)$. Jeżeli występuje odwrotna sytuacja, czyli dla połączenia (i, j) uzyskanego rozwiązania punkt i należy do zbioru $V' \setminus V$, natomiast j należy do zbioru $V \setminus \{1\}$, to należy dokonać zamiany połączenia (i, j) na połączenie $(1, i)$. Wreszcie jeżeli krawędzi (i, j) uzyskanego rozwiązania obydwa punkty i oraz j należą do zbioru $V' \setminus V$, konieczne staje się usunięcie takiego połączenia ze zbioru tras.

5.3. Metody dokładne a minimalne drzewo rozpinające

Przedstawione dotychczas algorytmy podziału i ograniczeń oparte były na idei relaksacji problemu wyjściowego zagadnieniem przydziału. Jednakże dla symetrycznego problemu układania tras jednego pojazdu zaproponowano także relaksację bazującą na zagadnieniu minimalnego drzewa rozpinającego (Held i Karp, 1971). Zanim do niej przejdziemy, należy przedstawić sformułowanie symetrycznego problemu komiwojażera. Rozpatrując graf $G(V, E, c)$, w którym zbiór V stanowi zbiór węzłów grafu odpowiadających bazie (węzeł nr 1) oraz obsługiwanym punktom, zbiór E reprezentuje możliwe połączenia pomiędzy punktami i oraz j o koszcie c_{ij} , postać modelu jest następująca:

$$\min. F(\mathbf{x}) = \sum_{i < j} c_{ij} x_{ij} \quad (5.23)$$

$$\sum_{i < k} x_{ik} + \sum_{j > k} x_{kj} = 2 \quad \text{dla } k = 1, \dots, N \quad (5.24)$$

$$\sum_{i, j \in S} x_{ij} \leq |S| - 1 \quad \text{dla } S \subset V \text{ oraz } 3 \leq |S| \leq N - 3 \quad (5.25)$$

$$x_{ij} \in \{0, 1\} \quad \text{dla } i, j = 1, \dots, N \text{ oraz } i < j \quad (5.26)$$

W powyższym sformułowaniu symetrycznego problemu komiwojażera każdy wierzchołek grafu jest stopnia drugiego, co zapewnia zbiór ograniczeń (5.24). Oznacza to, że z każdym wierzchołkiem grafu związane są tylko dwie krawędzie (pojazd jeden raz wjeżdża do obsługiwanego punktu oraz jeden raz z niego wyjeżdża). Ograniczenia (5.25) zapewniają eliminowanie podcykli tras. Natomiast zmienne decyzyjne zdefiniowane są jako zmienne binarne dla $i < j$ (5.26).

Ograniczenia przedstawione zespołem nierówności (5.25) mogą zostać zastąpione ich inną postacią (5.27):

$$\sum_{\substack{i \in S, j \in \bar{S} \\ \text{lub } j \in S, i \in \bar{S}}} x_{ij} \geq 2 \quad \text{dla } S \subset V \text{ oraz } 3 \leq |S| \leq N - 3 \quad (5.27)$$

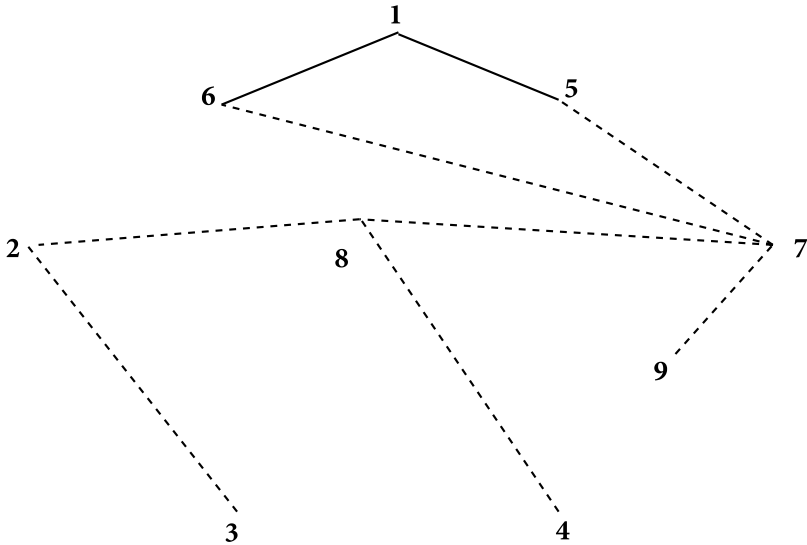
Powyższą nierówność można wykazać w następujący sposób: gdy oznaczymy przez $D(S)$ sumę stopni wierzchołków k grafu należących do zbioru S , będzie ona równa:

$$D(S) = \sum_{k \in S} \left(\sum_{i < k} x_{ik} + \sum_{j > k} x_{kj} \right) \quad (5.28)$$

Każda krawędź grafu (i, j) , dla której wierzchołki i oraz j należą do zbioru S , powoduje, że stopień wierzchołka k jest równy 2, a jeżeli jeden z wierzchołków i lub j nie należy do zbioru S , to stopień wierzchołka k jest równy 1. Stąd też $D(S)$ można przedstawić jako:

$$D(S) = 2 \sum_{i, j \in S} x_{ij} + \sum_{\substack{i \in S, j \in \bar{S} \\ \text{lub } j \in S, i \in \bar{S}}} x_{ij} \quad (5.29)$$

Zastosowanie problemu minimalnego drzewa rozpinającego jako relaksacji symetrycznego problemu komiwojażera polega na przyjęciu następującego założenia: w każdym dopuszczalnym rozwiązaniu symetrycznego problemu komiwojażera wierzchołek 1 reprezentujący bazę musi być stopnia drugiego, natomiast wszystkie pozostałe wierzchołki grafu muszą być ze sobą połączone. Stąd też odpowiednim dolnym ograniczeniem dla rozwiązania optymalnego symetrycznego zadania komiwojażera będzie długość minimalnego drzewa rozpinającego zbudowanego na zbiorze wierzchołków grafu $V \setminus \{1\}$ powiększona o dwie krawędzie wierzchołka nr 1.



Rys. 5.8. Przykład drzewa rozpinającego z wyodrębnionym wierzchołkiem nr 1
Źródło: opracowanie własne.

Zapis formalny tego problemu jest następujący:

$$\min. F(\mathbf{x}) = \sum_{i < j} c_{ij} x_{ij} \tag{5.30}$$

$$\sum_{i < j} x_{ij} = N \tag{5.31}$$

$$\sum_{j=2}^N x_{1j} = 2 \tag{5.32}$$

$$\sum_{\substack{i \in S, j \in \bar{S} \setminus \{1\} \\ \text{lub } j \in S, i \in \bar{S} \setminus \{1\}}} x_{ij} \geq 1 \quad \text{dla } S \subset V \setminus \{1\} \text{ oraz } 1 \leq |S| \leq N - 1 \tag{5.33}$$

$$x_{ij} \in \{0, 1\} \tag{5.34}$$

Powyższy model matematyczny jest uproszczeniem modelu matematycznego sformułowanego dla symetrycznego zadania komiwojażera (5.23)–(5.27). Ograniczenia wyrażone nierównością (5.31) są połową sumy wyrażonej nierównościami (5.24). Ograniczenie (5.32) jest szczególnym przypadkiem ograniczenia (5.24) dla $k = 1$. Natomiast ograniczenia (5.33) są słabszą postacią ograniczeń (5.27).

Zaproponowany przez Helda i Karpa algorytm podziału i ograniczeń wykorzystujący relaksację opartą na zagadnieniu minimalnego drzewa rozpinającego został przedstawiony na rysunku 5.9.

```

krok 1: inicjalizacja
repeat
  krok 2: wybór węzła drzewa
  krok 3: podział węzła na podproblemy
  repeat
    repeat
      krok 4: obliczenie dolnego oszacowania podproblemu
    until  $\omega(k_{[l]}) < F^*$ 
    krok 5: sprawdzenie dopuszczalności rozwiązania
  until  $F_{k[l]} < F^*$ 
until lista zadań do podziału jest pusta

```

Rys. 5.9. Schemat algorytmu podziału i ograniczeń M. Helda i R.M. Karpa
Źródło: opracowanie własne.

W powyższym algorytmie podziału i ograniczeń Helda i Karpa przyjęte zostały następujące oznaczenia:

- F^* – długość najlepszej dotychczas znalezionej trasy komiwojażera;
- $\omega(k)$ – dolna granica oszacowana w k -tym węźle drzewa;
- A_k – zbiór włączonych do trasy połączeń pomiędzy punktami (krawędzi grafu) w zbiorze tras reprezentowanych przez k -ty węzeł drzewa;
- B_k – zbiór wyłączonych z trasy połączeń pomiędzy punktami (krawędzi grafu) w zbiorze tras reprezentowanych przez k -ty węzeł drzewa.

W pierwszym kroku algorytmu, którym jest inicjalizacja, należy otrzymać pierwsze rozwiązanie zadania komiwojażera przez zastosowanie dowolnej metody heurystycznej. Celem tego zabiegu jest uzyskanie dopuszczalnego rozwiązania początkowego, które stanie się na początku dotychczas najlepszym znalezionym rozwiązaniem z odpowiadającą mu wartością funkcji celu F^* . Rozwiązaniu temu odpowiadać będzie pierwszy wierzchołek drzewa przeszukiwania przestrzeni rozwiązań metodą podziału i ograniczeń. Dla tego wierzchołka należy także policzyć dolną granicę. Sposób jej wyznaczenia dla tego wierzchołka drzewa, jak również dla wszystkich pozostałych, jest taki sam i opiera się na funkcji Lagrange'a. Jest to jeden z najistotniejszych elementów algorytmu charakteryzujących podejście zaproponowane przez wspomnianych autorów. Funkcja celu w modelu określonym wzorami (5.31)–(5.34) jest zmodyfikowana przez wstawienie do niej ograniczeń (5.32), dzięki czemu otrzymuje się w ten sposób funkcję Lagrange'a:

$$L(\lambda) = \min_x \left\{ \sum_{i < j} c_{ij} x_{ij} + \sum_{k \in V} \lambda_k \left(\sum_{i < j} x_{ik} + \sum_{i < j} x_{kj} - 2 \right) \right\} \quad (5.35)$$

W funkcji wyrażonej wzorem (5.35) x jest rozwiązaniem dopuszczalnym problemu minimalnego drzewa rozpinającego dla wszystkich punktów odwiedzanych

przez pojazd z wyjątkiem wierzchołka nr 1 reprezentującego bazę. Wierzchołek ten dołączany jest do drzewa przez dodanie dwóch najkrótszych krawędzi łączących go z dowolnymi pozostałymi wierzchołkami drzewa.

Ponadto należy zauważyć, że jeżeli dla każdego i -tego punktu odwiedzanego przez pojazd dodana zostanie do współczynnika funkcji celu (5.31) pewna wielkość λ_i dla wszystkich połączeń (krawędzi grafu), długość całkowita trasy wzrośnie o $\sum_{i \in V} 2\lambda_i$. Wynika to z tego, że w trasie z każdym punktem odwiedzanym przez pojazd związane są dwa połączenia (krawędzie grafu). Innymi słowy, gdy rozpatrujemy dwa odwiedzone przez pojazd punkty, to samo połączenie występuje dwukrotnie. Stąd też dla dowolnego połączenia pomiędzy punktami i oraz j (krawędzi grafu $\langle i, j \rangle$) odpowiadający mu współczynnik funkcji celu c'_{ij} jest równy:

$$c'_{ij} = c_{ij} + \lambda_i + \lambda_j \quad (5.36)$$

W efekcie funkcję $L(\lambda)$ można przedstawić w następującej postaci:

$$L(\lambda) = \min_x \left\{ \sum_{k \in V} \sum_{i < j} (c_{ij} + \lambda_i + \lambda_j) x_{ij} \right\} - \sum_{i \in V} 2\lambda_i \quad (5.37)$$

Dolna granica $\omega(k)$, która jest szacowana dla k -tego węzła drzewa algorytmu metody podziału i ograniczeń, jest równa:

$$\omega(k) = \max_{\lambda} \{L(\lambda)\} \quad (5.38)$$

Jeżeli obliczona dolna granica dla zadania odpowiadającego pierwszemu wierzchołkowi drzewa metody podziału i ograniczeń jest większa lub równa F^* , oznacza to, że rozwiązanie początkowe uzyskane zastosowaną metodą heurystyczną jest optymalne. W przeciwnym razie wierzchołek ten wraz z problemem, który reprezentuje, umieszczony zostaje na liście zadań przeznaczonych do podziału.

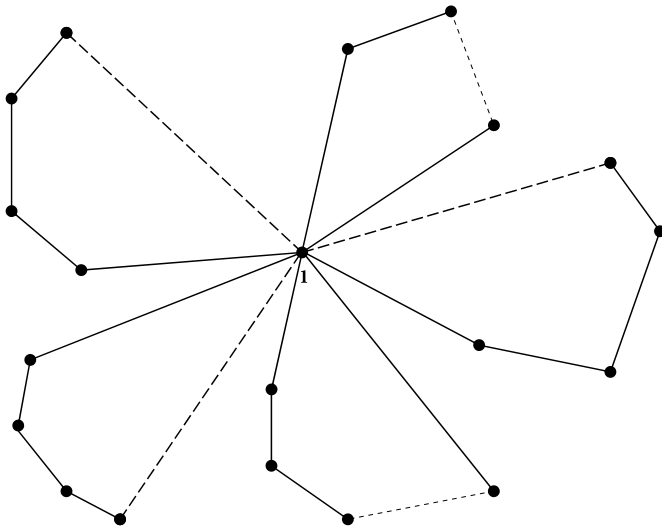
Sam algorytm podziału i ograniczeń w swej ogólnej postaci jest bardzo podobny do wcześniej prezentowanego algorytmu, w którym wykorzystywana była relaksacja zagadnieniem przydziału. W kroku drugim algorytmu następuje wybór węzła drzewa reprezentującego zadanie, które zostanie poddane podziałowi na podproblemy. Pod warunkiem, że lista zadań przeznaczonych do dalszego podziału nie jest pusta, do podziału wybrane zostaje to, dla którego dolna granica ma wartość najmniejszą. Krok trzeci to podział wybranego zadania reprezentowanego przez wierzchołek k drzewa na kolejne podproblemy, które będą odzwierciedlone w wierzchołkach drzewa: $k(1)$, $k(2)$, ..., $k(s)$. Podproblemy te są tworzone przez odpowiednią konstrukcję zbiorów krawędzi A_k oraz B_k . W kroku czwartym oblicza się dolną granicę $\omega(k)$ dla każdego ze skonstruowanych podproblemów. Jeżeli obliczona dolna granica odpowiadająca k -temu wierzchołkowi drzewa jest mniejsza od F^* , to należy sprawdzić, czy rozwiązanie podproblemu stanowi rozwiązanie dopuszczalne. W sytuacji, gdy uzyskane rozwiązanie stanowi cykl Hamiltona,

należy przyjąć dolną granicę rozpatrywanego podproblemu jako nową wartość F^* . W przeciwnym razie rozpatrywany podproblem należy umieścić na liście zadań przeznaczonych do kolejnych podziałów. Iteracje prezentowanego algorytmu podziału i ograniczeń wykonywane są do momentu, gdy lista zadań nie zawiera ani jednego podproblemu, który mógłby zostać podzielony na kolejne.

Podobny schemat algorytmu podziału i ograniczeń dla zagadnienia układania tras wielu pojazdów, wykorzystujący zarazem relaksację minimalnego drzewa rozpinającego, został przedstawiony przez N. Christofidesa, A. Mignozziego i P. Totha (1979). Jest to pewnego rodzaju rozszerzenie podejścia wykorzystującego relaksację minimalnego drzewa rozpinającego dla problemu jednego komiwojażera. Relaksacja zaproponowana do rozwiązania problemu wielu komiwojażerów nosi nazwę drzewa z centralnym wierzchołkiem m -tego stopnia. Mając ustaloną liczbę K pojazdów, graf $G(V, E)$, w którym wierzchołek nr 1 reprezentuje bazę, może zostać podzielony na cztery rodzaje (podzbiory) krawędzi (Christofides i in., 1981):

- 1) E_0 , podzbiór krawędzi grafu, które nie należą do rozwiązania;
- 2) E_1 , podzbiór krawędzi grafu, które tworzą drzewo z centralnym wierzchołkiem m -tego stopnia;
- 3) E_2 , podzbiór krawędzi grafu o liczebności y , których jednym z wierzchołków jest wierzchołek nr 1, przy czym $0 \leq y \leq K$;
- 4) E_3 , podzbiór krawędzi grafu o liczebności $m - y$, których żadnym z wierzchołków nie jest wierzchołek nr 1.

Przykładem podzbioru krawędzi grafu E_1 może być minimalne drzewo rozpinające, w którym wierzchołek nr 1 jest stopnia m , przy czym $m = 2K - y$ (rys. 5.10).



Rys. 5.10. Ilustracja podziału zbioru krawędzi grafu
Źródło: opracowanie własne.

Na rysunku 5.10 linią ciągłą zaznaczone zostały wszystkie krawędzie grafu stanowiące drzewo z wierzchołkiem centralnym siódmego stopnia. Linia przerywana oznacza krawędzie, dla których wierzchołek nr 1 stanowi ich jeden z dwóch wierzchołków. Liczba tych krawędzi wynosi w tym wypadku $y = 3$. Wreszcie krawędzie oznaczone linią kropkowaną nie mają wspólnego punktu z wierzchołkiem nr 1 grafu. Liczebność podzbioru tych krawędzi jest równa $K - y = 5 - 2 = 2$:

Dla tak podzielonego zbioru krawędzi grafu G , na którym zdefiniowany jest problem wielu komiwojazerów, N. Christofides, A. Mignozzi i P. Toth (1979) przedstawiają następujący model programowania matematycznego:

$$\min. F(\xi) = \sum_{l \in E} c_l (\xi_l^1 + \xi_l^2 + \xi_l^3) \quad (5.39)$$

$$\sum_{l \in (S, \bar{S})} \xi_l^1 \geq 1 \quad \text{dla } S \subset V \text{ oraz } 1 \leq |S| \quad (5.40)$$

$$\sum_{l \in E} \xi_l^1 = 2K - y \quad (5.41)$$

$$\sum_{l \in E} \xi_l^1 = N - 1 \quad (5.42)$$

$$\sum_{l \in E^1} \xi_l^2 = y \quad (5.43)$$

$$\sum_{l \in E \setminus E^1} \xi_l^3 = K - y \quad (5.44)$$

$$\sum_{l \in E^i} (\xi_l^1 + \xi_l^2 + \xi_l^3) = 2 \quad \text{dla } i = 1, 2, \dots, N \quad (5.45)$$

$$0 \leq y \leq K \quad \text{dla } y \in C \quad (5.46)$$

$$\xi_l^1 \in \{0, 1\} \quad \text{dla } l \in E \quad (5.47)$$

$$\xi_l^2 \in \{0, 1\} \quad \text{dla } l \in E \quad (5.48)$$

$$\xi_l^3 \in \{0, 1\} \quad \text{dla } l \in E \quad (5.49)$$

gdzie:

- l – dowolna krawędź grafu;
- c_l – koszt krawędzi l grafu;
- ξ_l^t – dla $t = 1, 2, 3$ oraz $l \in E$ oznaczają zmienne binarne, przyjmujące wartość 1 wtedy i tylko wtedy, gdy krawędź l grafu należy do danego podzbioru E_i ;
- E^i – zbiór wszystkich krawędzi grafu, dla których jeden z końców stanowi i -ty wierzchołek.

(S, \bar{S}) – zbiór wszystkich krawędzi grafu, dla których jeden z końców należy do zbioru S , natomiast drugi do zbioru \bar{S} .

W powyższym modelu (5.39)–(5.43) ograniczenia wyrażone nierównościami i równaniami (5.40)–(5.42) definiują zagadnienie drzewa z centralnym wierzchołkiem m -tego stopnia. Ograniczenia (5.43) zapewniają, że musi być dodatkowo dokładnie y krawędzi grafu, których jednym z końców jest wierzchołek nr 1 reprezentujący bazę. Z kolei ograniczenia wyrażone równościami (5.44) określają dodatkową liczbę $K - y$ krawędzi, dla których wierzchołek nr 1 nie stanowi żadnego z ich końców. Ostatnia grupa ograniczeń (5.45) konstryuuje stopień każdego z wierzchołków grafu (z wyjątkiem wierzchołka nr 1), który musi być równy 2.

Najważniejszym elementem w każdym algorytmie realizującym strategię poszukiwania rozwiązania optymalnego metodą podziału i ograniczeń jest sposób obliczenia dolnej granicy dla każdego nowo tworzonego podproblemu. W algorytmie N. Christofidesa, A. Mignozziego i P. Totha, podobnie jak w algorytmie M. Helda i R.M. Karpa dla problemu jednego komiwojażera, zastosowana została w tym celu funkcja Lagrange’a. Otóż ograniczenia wyrażone równościami (5.45) zostały opuszczone, a jednocześnie została zmodyfikowana funkcja celu:

$$L(\lambda) = \sum_{l \in E} c_l (\xi_l^1 + \xi_l^2 + \xi_l^3) + \sum_{l \in E} (\lambda_{\alpha(l)} + \lambda_{\beta(l)}) (\xi_l^1 + \xi_l^2 + \xi_l^3) - 2 \sum_{i=2}^N \lambda_i \quad (5.50)$$

gdzie $\lambda_i = 0$ oraz $\alpha(l)$ i $\beta(l)$ są wierzchołkami końcowymi krawędzi l . Tym samym dla ustalonej wartości y model przedstawiony wzorami (5.39)–(5.43) może zostać zapisany jako taki sam model przy ograniczeniach (5.39)–(5.44) z funkcją celu:

$$L(\lambda) = \sum_{t=1}^3 \sum_{l \in E} (c_l + \lambda_{\alpha(l)} + \lambda_{\beta(l)}) \xi_l^t - 2 \sum_{i=2}^N \lambda_i \quad (5.51)$$

Można zauważyć, że wartość funkcji celu (5.51) nie będzie mniejsza, jeżeli jej pierwszą część zastąpimy sumą trzech osobnych funkcji celu:

$$z^1(\lambda, y) + z^2(\lambda, y) + z^3(\lambda, y) - 2 \sum_{i=2}^N \lambda_i \quad (5.52)$$

Trzy funkcje: $z^1(\lambda, y)$, $z^2(\lambda, y)$, $z^3(\lambda, y)$ wraz z odpowiadającymi im ograniczeniami z modelu (5.39)–(5.49) tworzą trzy odrębne problemy, które mogą zostać rozwiązane w czasie wielomianowym. Pierwszy z nich ma postać:

$$\min. z^1(\lambda, y) = \sum_{l \in E} (c_l + \lambda_{\alpha(l)} + \lambda_{\beta(l)}) \xi_l^1 \quad (5.53)$$

$$\sum_{l \in (S, \bar{S})} \xi_l^1 \geq 1 \quad \text{dla } S \subset V \text{ oraz } 1 \leq |S| \quad (5.54)$$

$$\sum_{l \in E} \xi_l^1 = 2K - y \quad (5.55)$$

$$\sum_{l \in E} \xi_l^1 = N - 1 \quad (5.56)$$

$$\xi_l^1 \in \{0, 1\} \quad \text{dla } l \in E \quad (5.57)$$

Drugi sformułowany jest w następujący sposób:

$$\min. z^2(\lambda, y) = \sum_{l \in E} (c_l + \lambda_{\alpha(l)} + \lambda_{\beta(l)}) \xi_l^2 \quad (5.58)$$

$$\sum_{l \in E^1} \xi_l^2 = y \quad (5.59)$$

$$\xi_l^2 \in \{0, 1\} \quad \text{dla } l \in E \quad (5.60)$$

Natomiast postać trzeciego jest taka:

$$\min. z^3(\lambda, y) = \sum_{l \in E} (c_l + \lambda_{\alpha(l)} + \lambda_{\beta(l)}) \xi_l^3 \quad (5.61)$$

$$\sum_{l \in E \setminus E^1} \xi_l^3 = K - y \quad (5.62)$$

$$\xi_l^3 \in \{0, 1\} \quad \text{dla } l \in E \quad (5.63)$$

Ostatecznie wartość dolnego oszacowania ustalanego dla każdego nowo tworzonego podproblemu w algorytmie opartym na metodzie podziału i ograniczeń jest obliczana zgodnie z następującą formułą:

$$\omega(k) = \max_{m_1 \leq y \leq m} \max_{\lambda} \left\{ z^1(\lambda, y) + z^2(\lambda, y) + z^3(\lambda, y) - 2 \sum_{i=2}^N \lambda_i \right\} \quad (5.64)$$

Wartość m_1 z kolei jest to dolna granica liczby tras, które składają się tylko i wyłącznie z bazy (wierzchołek nr 1 grafu) oraz tylko jednego obsługiwanego punktu.

Rozdział 6

Heurystyki układania tras pojazdów

W ciągu ostatnich przeszło pięćdziesięciu lat zaproponowano wiele metod heurystycznych, znanych również pod nazwą metod przybliżonych. Pomimo przekonania, że algorytmy te przeszukują wciąż niewielki obszar przestrzeni rozwiązań, są one powszechnie używane, ponieważ cechują się wysoką elastycznością, a ich implementacja komputerowa nie nastręcza trudności, w przeciwieństwie do algorytmów dokładnych. Elastyczność tych metod polega na ich łatwym zmodyfikowaniu w celu uwzględnienia różnorodnych ograniczeń pojawiających się w rzeczywistych problemach.

Rozpatrując metody heurystyczne, jakimi są algorytmy poszukiwania rozwiązania optymalnego dla zadań układania tras dla pojazdów, można wyróżnić wśród nich kilka podstawowych grup:

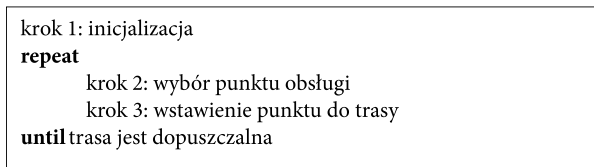
- metody konstrukcyjne;
- metody dekompozycyjne;
- metody wzrostu (poprawy rozwiązania).

Metody konstrukcyjne i metody wzrostu są charakterystyczne dla problemów układania tras jednego i wielu pojazdów. Metody dekompozycyjne zaś odnoszą się do zagadnień wielu komiwojażerów, w których subproblemy podlegające optymalizacji, a dotyczące rejonizacji działań pojazdów oraz sekwencji obsługi poszczególnych punktów (klientów), traktowane są osobno.

Należy również zwrócić uwagę na metody heurystyczne mieszane, wśród których zacierają się granice między wyżej wymienionymi grupami. W szczególności ma to zastosowanie do metod łączących procedury konstrukcyjne lub dekompozycyjne z metodami wzrostu. W niniejszym rozdziale omówionych zostanie kilka metod reprezentatywnych dla każdej z dwóch pierwszych wymienionych klas. Trzecia grupa metod, metody poprawy rozwiązań, stanowią klasę metod realizujących bardzo różne strategie przeszukiwania przestrzeni rozwiązań. Strategie te noszą nazwę metaheurystyk, czyli zbioru reguł służących do budowy konkretnych algorytmów optymalizacyjnych, dlatego zostaną opisane w osobnym rozdziale niniejszej publikacji.

6.1. Heurystyki konstrukcyjne

Ogólna idea algorytmów należących do metod konstrukcyjnych polega na stopniowej budowie rozwiązania dopuszczalnego, przy jednoczesnym zwracaniu uwagi na wartość funkcji celu. Ogólny schemat algorytmu podejścia konstrukcyjnego dla zagadnienia jednego komiwojażera został przedstawiony na rysunku 6.1.



Rys. 6.1. Schemat metody konstrukcyjnej dla problemu komiwojażera
Źródło: opracowanie własne.

W kroku pierwszym (inicjalizacji) algorytmu dokonywany jest wybór punktu, z którego rozpoczynać się będzie trasa pojazdu. Może to być baza, lecz również dowolny inny punkt obsługi. W wypadku, gdy baza nie jest punktem początkowym, trzeba pamiętać, że musi ona zostać uwzględniona w trasie pojazdu jako jeden z punktów, które pojazd musi odwiedzić.

W kroku drugim wybieramy kolejny punkt obsługi, który zostanie umieszczony w stopniowo budowanej trasie pojazdu. Natomiast krok trzeci polega na wstawieniu do częściowo zbudowanej trasy punktu obsługi wybranego w kroku drugim. Kroki drugi i trzeci są sukcesywnie powtarzane do momentu, w którym powstanie dopuszczalna trasa pojazdu, zawierająca wszystkie punkty obsługi, jakie ma odwiedzić pojazd. To, co odróżnia algorytmy proponowane dla problemu komiwojażera, to procedury zawarte w krokach drugim oraz trzecim przedstawionego powyżej schematu metody konstrukcyjnej.

Procedura generowania trasy dla jednego komiwojażera może w naturalny sposób zostać przeniesiona na problem wielu komiwojażerów (rys. 6.2). Jediną tutaj różnicą jest konieczność zwracania uwagi na sytuację, w której podczas sukcesywnego budowania tras mogą zostać złamane dodatkowe warunki dotyczące np. maksymalnej ładowności lub maksymalnego czasu pracy pojazdu.

Najbardziej znanym przykładem algorytmu należącego do metod konstrukcyjnych dla problemu układania tras jednego lub wielu pojazdów jest algorytm drogi do najbliższego sąsiada (Rosenkrantz i in., 1977). Jego idea polega na stopniowym konstruowaniu trasy przez dołączanie najbliższego kolejnego punktu obsługi w stosunku do punktu, w którym aktualnie znajduje się pojazd. Wybrany punkt zostaje dołączony na końcu częściowo zbudowanej trasy pojazdu. Schemat algorytmu dla problemu komiwojażera zaprezentowany jest na rysunku 6.3.

```

krok 1: inicjalizacja
repeat
    repeat
        krok 2: inicjalizacja trasy
        krok 3: wybór punktu obsługi
        krok 4: wstawienie punktu do trasy
    until trasa jest dopuszczalna
until rozwiązanie jest dopuszczalne

```

Rys. 6.2. Schemat metody konstrukcyjnej dla problemu wielu komiwojazerów
Źródło: opracowanie własne.

```

repeat
    krok 1: inicjalizacja
    repeat
        krok 2: wybór najbliższego punktu obsługi
        krok 3: wstawienie punktu na koniec trasy T
    until trasa jest dopuszczalna
until wszystkie trasy rozpatrzone

```

Rys. 6.3. Schemat algorytmu drogi do najbliższego sąsiada
Źródło: opracowanie własne.

Możliwą modyfikacją algorytmu drogi do najbliższego sąsiada jest jego wersja „dwustronna”. Modyfikacja odbywa się w kroku trzecim, gdzie wybrany punkt można wstawić do częściowo budowanej trasy zarówno na jej końcu, jak i na jej początku. Oznacza to, że trasa pojazdu może być rozbudowywana w obydwu kierunkach, startując od wybranego punktu początkowego. Ponadto algorytm daje lepsze efekty, jeżeli zostanie powtórzony tyle razy, ile jest wszystkich punktów, które pojazd musi odwiedzić. Za każdym powtórzeniem algorytmu w kroku pierwszym należy wybrać inny punkt startowy dla pojazdu.

Innym dobrze znanym algorytmem heurystycznym należącym do grupy metod konstrukcyjnych jest algorytm wstawiania węzłów (sukcesywnego dołączania węzłów), którego schemat zamieszczono na rysunku 6.4.

```

krok 1: częściowa trasa początkowa
repeat
    krok 2: wybór punktu najbliższego trasie
    krok 3: wstawienie punktu w najlepsze miejsce trasy
until trasa jest dopuszczalna

```

Rys. 6.4. Schemat algorytmu sukcesywnego dołączania węzłów
Źródło: opracowanie własne.

W kroku pierwszym należy zbudować częściową trasę początkową. Trasą tą może być ustalony arbitralnie punkt początkowy (np. baza) lub też ustalone arbitralnie połączenie dwóch punktów (np. najkrótszy odcinek między dwoma punktami obsługi).

Istotny w tym algorytmie jest krok drugi, w którym należy wskazać kolejny punkt obsługi P^* , jaki zostanie dołączony do istniejącej częściowej trasy pojazdu T_k w k -tej iteracji algorytmu. Punktem tym może być ten, który spełnia warunek:

$$c(T_k, P^*) = \min. \{c(T_k, P_k) : P_k \notin T_k\} \quad (6.1)$$

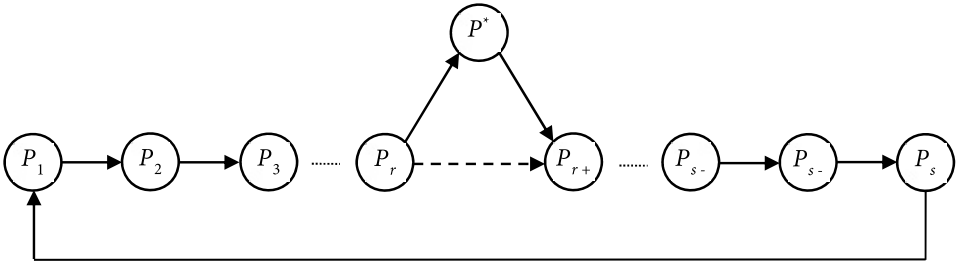
co oznacza, że wybrany punkt P^* musi leżeć najbliżej trasy spośród wszystkich pozostałych punktów niewłączonych jeszcze do trasy T_k . Odległość punktu P_k od trasy T_k należy uprzednio wyznaczyć zgodnie z formułą:

$$c(T_k, P_k) = \min. \{c(P_j, P_k) : P_k \notin T_k \text{ oraz } P_j \in T_k\} \quad (6.2)$$

gdzie:

$c(P_j, P_k)$ – odległość pomiędzy punktem P_j należącym do trasy T_k a punktem P_k , który do niej nie należy, a jest aktualnie rozpatrywany.

W kroku trzecim prezentowanego algorytmu wstawiania węzłów do trasy wybrany punkt P^* należy wstawić w odpowiednie miejsce w trasie. Spowoduje to przyrost długości częściowo zbudowanej trasy T_k o wartość $\Delta c(T_k)$, co pokazuje rysunek 6.5.



Rys. 6.5. Wstawienie punktu P^* do częściowo zbudowanej trasy

Źródło: opracowanie własne.

W tym wypadku będzie nim miejsce między taką parą punktów P_r o raz P_{r+1} w istniejącej już częściowo zbudowanej trasie T_k , dla których przyrost kosztu trasy $\Delta c(T_k)$ będzie jak najmniejszy. Przyrost ten wyznaczany jest zgodnie z formułą:

$$\Delta(T_k) = \min. \{c(P_r, P^*) + c(P^*, P_{r+1}) - c(P_r, P_{r+1})\} \quad (6.3)$$

Przedstawiony algorytm sukcesywnego dołączania kolejnych punktów do trasy pojazdu zakłada w kroku drugim wybór punktu, który znajduje się najbliżej

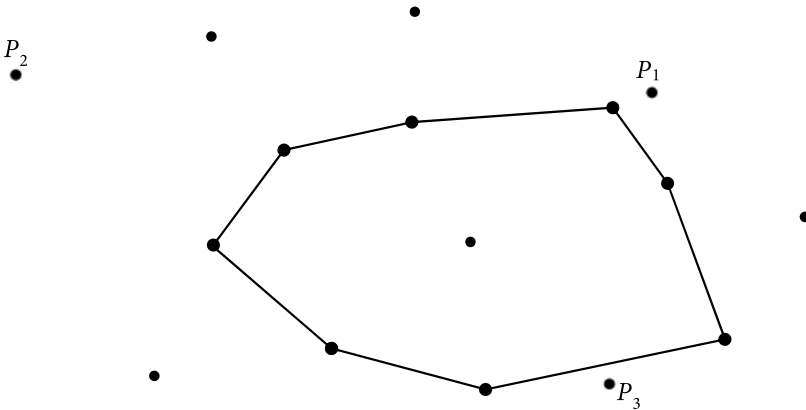
częściowo już zbudowanej trasy. Jednakże istnieją pewne odmiany tego algorytmu polegające na modyfikacji postępowania opisanego w kroku drugim. Jedną z nich jest wybór punktu, który zlokalizowany jest w największej odległości od częściowo zbudowanej trasy:

$$c(T_k, P^*) = \max. \{c(T_k, P_k) : P_k \notin T_k\} \quad (6.4)$$

Inna modyfikacja zakłada wybór takiego punktu obsługi P_k^* , że jego wstawienie w odpowiednie miejsce częściowo zbudowanej trasy pojazdu prowadzić będzie do wzrostu jej długości o jak najmniejszą wartość.

$$\Delta(T_k, P^*) = \min. \left\{ \min. \left[c(P_r, P_k^*) + c(P_k^*, P_{r+1}) - c(P_k^*, P_{r+1}) \right] \right\} \quad (6.5)$$

Na rysunku 6.6 przedstawione zostały omówione powyżej sposoby wyboru kolejnego punktu P^* . Punkt P_1 znajduje się najbliżej trasy T_k , natomiast P_2 jest od niej najbardziej oddalony. Z kolei wstawienie do trasy T_k punktu P_3 spowoduje przyrost jej długości o najmniejszą wartość w porównaniu do sytuacji, w której wstawione zostałyby pozostałe punkty.



Rys. 6.6. Ilustracja podziału zbioru krawędzi grafu
Źródło: opracowanie własne.

Dla problemu wielu komiwojazerów zastosowanie algorytmu sukcesywnego dołączania węzłów będzie możliwe przy jednoczesnym pilnowaniu ograniczeń dotyczących np. ładowności pojazdu. W wypadku przyporządkowania do pojazdu określonego zbioru punktów obsługi, których łączne zapotrzebowanie będzie wyczerpywało możliwości ładunkowe pojazdu, konieczne okaże się utworzenie nowej trasy początkowej składającej się z jednego punktu, jakim będzie tutaj baza.

Innym godnym uwagi algorytmem konstrukcyjnym dla problemu układania tras pojazdów, który znajduje zastosowanie zwłaszcza wtedy, gdy macierz

odległości pomiędzy punktami obsługi nie jest symetryczna, jest algorytm łączenia podcykli wykorzystujący zagadnienie przydziału (rys. 6.7) (Karp, 1979).

W kroku pierwszym algorytmu należy rozwiązać zadanie przydziału, w którym elementy macierzy kosztów stanowią koszty połączeń pomiędzy punktami obsługi dla wyjściowego zadania komiwojażera. W efekcie otrzymamy takie przyporządkowanie punktów obsługi, że możliwe będzie skonstruowanie jednej dopuszczalnej trasy komiwojażera. Wtedy algorytm kończy postępowanie. W innej sytuacji może się okazać, że rozwiązanie zadania przydziału wskazuje na kilka mniejszych tras (podcykli) komiwojażera, co oznacza, że należy przejść do kroku drugiego algorytmu.

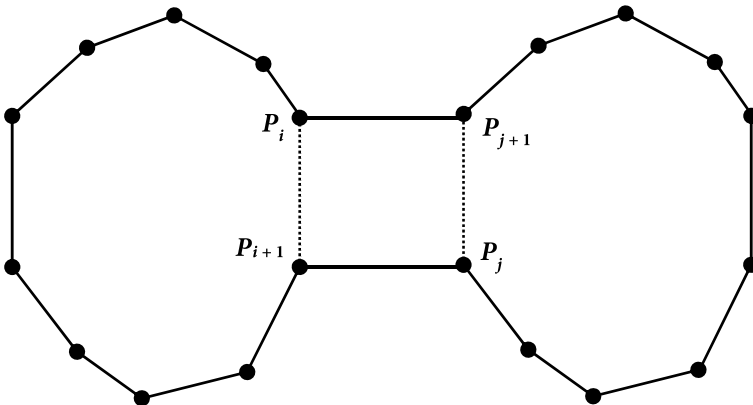
```

krok 1: rozwiązanie zadania przydziału
repeat
    krok 2: wybór podcykli
    krok 3: połączenie podcykli
until trasa jest dopuszczalna
  
```

Rys. 6.7. Schemat algorytmu łączenia podcykli

Źródło: opracowanie własne.

W kroku tym, jeżeli są więcej niż dwa podcykle, należy dokonać wyboru dwóch z nich. Do połączenia wybierane są te podcykle, których zbiory należących do nich punktów obsługi są najliczniejsze. Krok trzeci algorytmu to połączenie dwóch wybranych wcześniej podcykli w jedną trasę (rys. 6.8).



Rys. 6.8. Łączenie dwóch podcykli w jedną trasę

Źródło: opracowanie własne.

Aby dokonać połączenia dwóch wybranych w kroku drugim podcykli, należy uprzednio wskazać w każdym z nich po jednym połączeniu (P_i, P_{i+1}) oraz (P_j, P_{j+1}) , które zostaną przerwane. Chodzi o połączenia, dla których powstała po połączeniu

obydwu podcykli trasa będzie miała najmniejszą długość. Wskazanie właściwych połączeń odbywa się przez minimalizację kosztu połączenia wyrażonego wzorem:

$$c(P_i^*, P_j^*) = \min. \{c(P_i^*, P_{j+1}^*) + c(P_j^*, P_{i+1}^*) - c(P_i^*, P_{i+1}^*) - c(P_j^*, P_{j+1}^*)\} \quad (6.6)$$

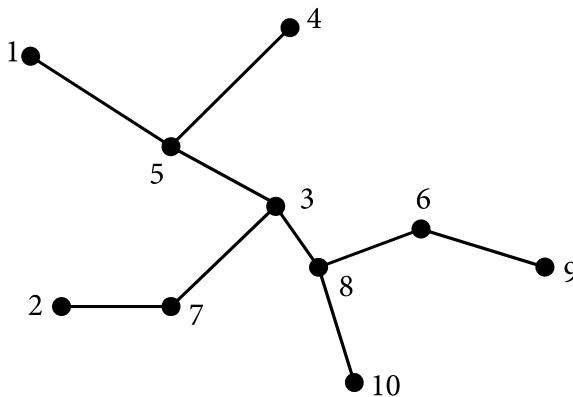
Osobną grupę metod konstrukcyjnych dla zadania układania trasy pojazdu stanowią metody, które – podobnie jak metody podziału i ograniczeń – wykorzystują relaksację minimalnego drzewa rozpinającego. Najprostszą jej wersją jest algorytm przedstawiony na rysunku 6.9.

krok 1: budowa minimalnego drzewa rozpinającego
 krok 2: podwojenie krawędzi drzewa
 krok 3: wybór węzła startowego
repeat
 krok 4: dołączanie węzłów drzewa do trasy
until trasa jest dopuszczalna

Rys. 6.9. Algorytm budowy trasy pojazdu z wykorzystaniem minimalnego drzewa rozpinającego

Źródło: opracowanie własne.

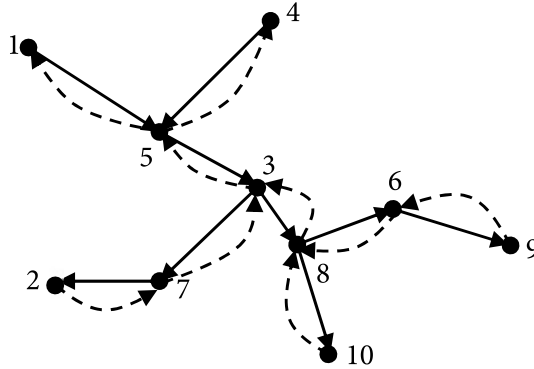
W kroku pierwszym algorytmu należy dla zadanego zbioru punktów obsługi, które mają zostać odwiedzone przez pojazd, skonstruować minimalne drzewo rozpinające. Do uzyskania odpowiedniego rozwiązania można wykorzystać np. algorytm Prima (Sysło i in., 1995) lub Kruskala (Kruskal, 1956). Przykład takiego drzewa rozpinającego dla zadania układania tras pojazdu z dziesięcioma punktami obsługi zawiera rysunek 6.10a.



Rys. 6.10a. Minimalne drzewo rozpinające dla dziesięciu punktów obsługi

Źródło: opracowanie własne.

W kroku drugim należy każdą nieskierowaną krawędź uzyskanego drzewa rozpinającego zastąpić dwiema skierowanymi przeciwnie krawędziami, uzyskując podwójne minimalne drzewo rozpinające, co zostało pokazane na rysunku 6.10b.

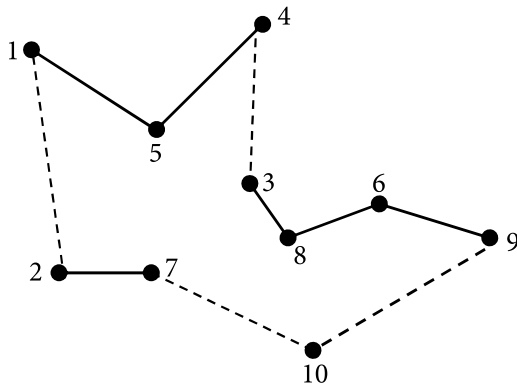


Rys. 6.10b. Podwójne minimalne drzewo rozpinające dla dziesięciu punktów obsługi
Źródło: opracowanie własne.

Krok trzeci algorytmu polega na określeniu punktu startowego dla trasy pojazdu, którym może być dowolny węzeł stopnia drugiego minimalnego drzewa rozpinającego przedstawionego na rysunku 6.10b (dowolny liść drzewa). Startując od wybranego węzła drzewa, w kroku czwartym algorytmu należy przejść przez wszystkie gałęzie drzewa i powrócić do węzła startowego. Po drodze należy dołączać do trasy pojazdu kolejne punkty (węzły drzewa) z wyjątkiem tych, które zostały już umieszczone w trasie. Dla podwójnego drzewa rozpinającego z rysunku 6.10b trasa przechodząca przez wszystkie gałęzie drzewa, rozpoczynając od węzła nr 1, może mieć postać: [1, 5, 4, 5, 3, 8, 6, 9, 6, 8, 10, 8, 3, 7, 2, 7, 3, 5, 1]. Po usunięciu węzłów, które się powtarzają, finalna trasa pojazdu jest następująca: [1, 5, 4, 3, 8, 6, 9, 10, 7, 2, 1]. Rysunek 6.10c przedstawia uzyskaną trasę pojazdu, gdzie zaznaczono połączenia pomiędzy węzłami należące do minimalnego drzewa rozpinającego (linia ciągła), jak również nowo utworzone w wyniku usunięcia węzłów, które się powtarzały (linia przerywana).

Należy zauważyć, że przedstawiona na rysunku 6.10c trasa nie jest jedynym możliwym rozwiązaniem, jakie da się uzyskać w wyniku wcześniej skonstruowanego minimalnego drzewa rozpinającego. Ostateczny jej kształt zależy będzie od kolejności odwiedzanych gałęzi drzewa. Ponadto jeżeli spełniony jest warunek trójkąta dotyczący odległości pomiędzy odwiedzanymi przez pojazd punktami, to ostateczna długość uzyskanej finalnej trasy pojazdu nie będzie większa niż dwukrotność długości minimalnego drzewa rozpinającego.

Drugim algorytmem opartym na konstrukcji minimalnego drzewa rozpinającego jest algorytm zaproponowany przez N. Christofidesa (1976). Jest on nieco bardziej złożony od poprzedniego algorytmu, lecz stanowi jego znaczące usprawnienie. Schemat działania tego algorytmu można przedstawić tak jak to pokazano na rysunku 6.11.



Rys. 6.10c. Finalna trasa pojazdu algorytmu minimalnego drzewa rozpinającego

Źródło: opracowanie własne.

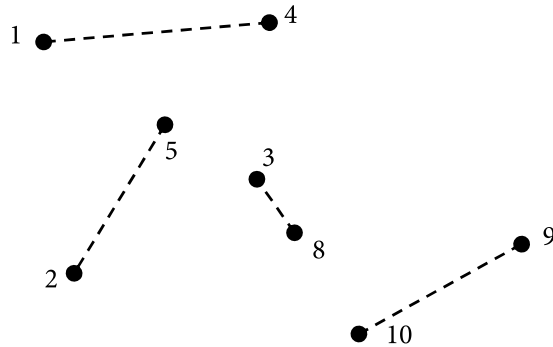
krok 1: budowa minimalnego drzewa rozpinającego
 krok 2: budowa grafu Eulera
 krok 3: konstrukcja ścieżki Eulera
 krok 4: wybór węzła startowego
repeat
 krok 5: dołączanie węzłów drzewa do trasy
until trasa jest dopuszczalna

Rys. 6.11. Algorytm N. Christofidesa

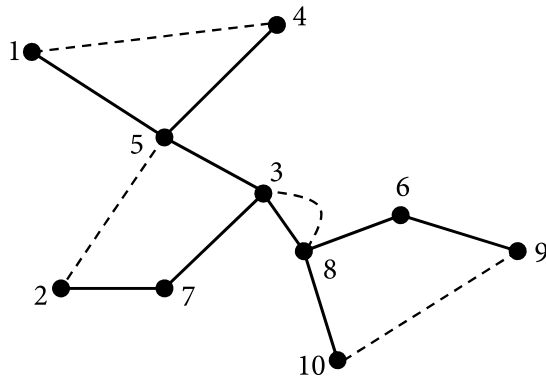
Źródło: opracowanie własne.

W kroku pierwszym należy ponownie rozpocząć od budowy minimalnego drzewa rozpinającego dla wyjściowego problemu układania trasy pojazdu (rys. 6.10a). Istotna różnica w stosunku do poprzedniego algorytmu dotyczy działań, jakie należy wykonać w krokach drugim i trzecim. Otrzymane w kroku pierwszym minimalne drzewo rozpinające nie jest grafem Eulera, czyli grafem posiadającym cykl Eulera. Cyklem Eulera w grafie jest ścieżka, która rozpoczyna się w ustalonym węźle początkowym, przechodzi przez wszystkie krawędzie grafu dokładnie jeden raz, a następnie wraca do węzła początkowego (Ross i Wright, 1996). Aby utworzyć graf Eulera, należy w pierwszej kolejności rozwiązać problem skojarzenia wierzchołków drzewa o minimalnym koszcie, przy czym dotyczy to tylko i wyłącznie wierzchołków drzewa rozpinającego, które mają nieparzysty stopień (rys. 6.12a). W powyższym przykładzie wierzchołkami, które posiadają nieparzysty stopień, są wszystkie z wyjątkiem wierzchołków nr 6 i nr 7, tym samym zostają one pominięte.

Następnie otrzymane nowe połączenia węzłów grafu dołącza się do minimalnego drzewa rozpinającego, otrzymując w ten sposób graf Eulera (rys. 6.12b).



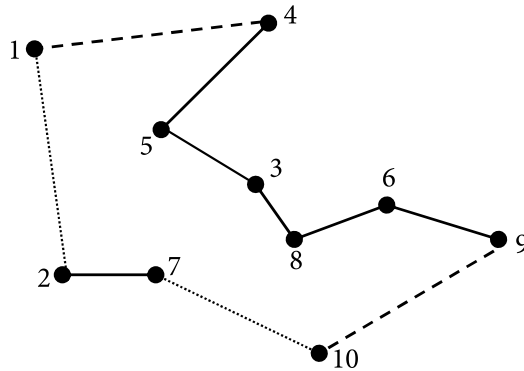
Rys. 6.12a. Rozwiązanie problemu skojarzenia wierzchołków o minimalnym koszcie
Źródło: opracowanie własne.



Rys. 6.12b. Graf Eulera
Źródło: opracowanie własne.

W kroku czwartym należy skonstruować ścieżkę Eulera wychodzącą z ustalonego wężła początkowego i tam się kończącą. Jeżeli arbitralnie wybranym punktem początkowym trasy będzie węzeł nr 1, to ścieżka Eulera ma postać: [1, 4, 5, 3, 8, 6, 9, 10, 8, 3, 7, 2, 5, 1]. Tym samym, powtarzając procedurę kroku ostatniego dla wcześniej prezentowanego algorytmu dla minimalnego drzewa rozpinającego, ostateczna trasa pojazdu po usunięciu powtarzających się punktów będzie miała postać cyklu Hamiltona: [1, 4, 5, 3, 8, 6, 9, 10, 7, 2, 1]. Na rysunku 6.12c przedstawiona została trasa finalna: zaznaczono połączenia punktów będących krawędziami minimalnego drzewa rozpinającego (linia ciągła), nowe krawędzie grafu uzyskane w wyniku rozwiązania problemu skojarzenia wierzchołków o minimalnym koszcie oraz dodatkowo utworzone połączenia wskutek usunięcia węzłów (linia przerywana), które się powtarzały (linia kropkowana).

Jak wskazuje autor przedstawionego algorytmu, całkowita długość finalnej trasy pojazdu jest mniejsza lub równa $3/2$ długości skonstruowanego na początku minimalnego drzewa rozpinającego.



Rys. 6.12c. Finalna trasa pojazdu algorytmu N. Christofidesa
Źródło: opracowanie własne.

Należy zauważyć, że opisane powyżej dwa algorytmy oparte na konstrukcji minimalnego drzewa rozpinającego, jak również pozostałe wcześniej prezentowane algorytmy mogą posłużyć także do konstrukcji dopuszczalnego rozwiązania dla zagadnienia układania tras wielu pojazdów, jeśli będzie się jednocześnie przestrzegać np. ograniczeń zasobowych dotyczących ładowności lub maksymalnej długości trasy (czasu pracy pojazdu). Algorytm oszczędnościowego łączenia tras G. Clarka i J.W. Wrighta (1964) (rys. 6.13) skonstruowany został z myślą o zagadnieniu wielu komiwojazerów z ograniczeniami zasobowymi. Jednakże przy założeniu, że zbiór wszystkich klientów może zostać obsłużony przez jeden pojazd, algorytm ten może zostać z powodzeniem wykorzystany również w celu uzyskania suboptymalnej trasy dla problemu jednego komiwojazera. Ogólna idea algorytmu opiera się na początkowym utworzeniu zbioru tras składających się tylko z jednego punktu obsługi, którego liczebność jest równa liczbie wszystkich punktów, jakie mają zostać obsłużone przez bazę (rys. 6.13).



Rys. 6.13. Stopniowe formowanie się tras w algorytmie oszczędnościowego łączenia tras
Źródło: opracowanie własne.

W następnych krokach algorytmu trasy te są stopniowo ze sobą łączone, aż do momentu uzyskania dopuszczalnego rozwiązania końcowego. W algorytmie tym dwa problemy optymalizacyjne stawiane w zagadnieniu wielu komiwojazerów (rejon obsługi oraz kolejność odwiedzania punktów obsługi przez pojazd)

realizowane są jednocześnie. Nazwa algorytmu wywodzi się od obliczanych podczas wykonywania procedury pewnych wielkości, nazywanych oszczędnościami. Przebieg działania algorytmu można przedstawić w następujący sposób (rys. 6.14):

```

krok 1: inicjalizacja
krok 2: obliczenie oszczędności
krok 3: utworzenie listy połączeń tras
repeat
  repeat
    krok 4: sprawdzenie możliwości połączenia tras
  until połączenie jest możliwe
  krok 5: łączenie tras i aktualizacja listy tras i połączeń
until lista połączeń jest pusta
  
```

Rys. 6.14. Algorytm G. Clarka i J.W. Wrighta

Źródło: opracowanie własne.

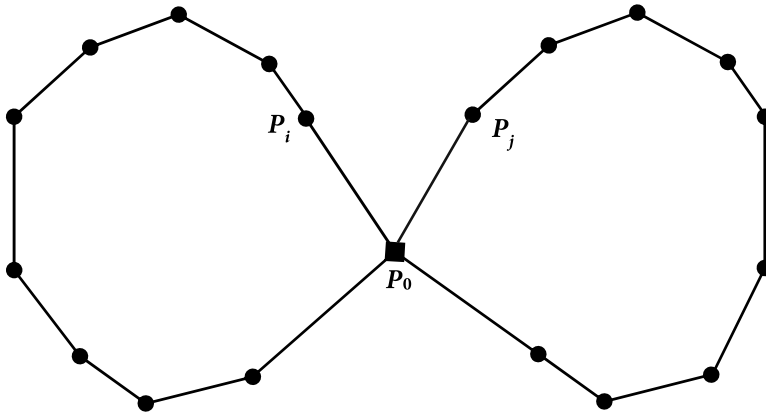
Krok pierwszy algorytmu to budowa wyjściowego zbioru tras, w których pojazd wyrusza z bazy, odwiedza i -ty punkt obsługi i do bazy powraca: $P_0 \rightarrow P_i \rightarrow P_0$. Dla każdej pary punktów obsługi (P_i, P_j) należy obliczyć wartość $s(P_i, P_j)$, która będzie wyrażać oszczędność wynikającą z tego tytułu, że dwa rozpatrywane punkty obsługi znajdują się w jednej trasie, a nie w odrębnych trasach. Wielkość $s(P_i, P_j)$ wyznaczana jest według poniższego wzoru:

$$s(P_i, P_j) = c(P_0, P_i) + c(P_j, P_0) - c(P_i, P_j) \quad (6.7)$$

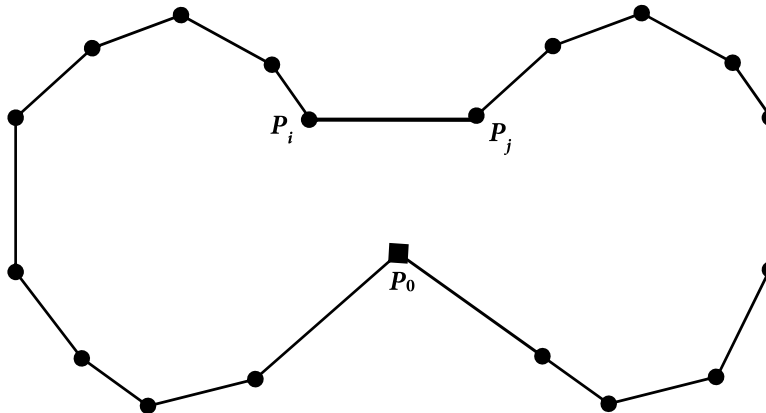
Wyznaczona na podstawie macierzy kosztów C macierz oszczędności S zawierać może zarówno elementy dodatnie, zerowe, jak i ujemne. Te ostatnie nie będą w niej występować, jeżeli elementy macierzy kosztów odpowiadają odległościom pomiędzy punktami obsługi spełniającymi warunek trójkąta. W celu utworzenia w trzecim kroku listy ewentualnych połączeń (P_i, P_j) należy rozpatrywać tylko oszczędności dodatnie, które zostaną posortowane malejąco. Tym samym rozpatrywane będą w pierwszej kolejności te połączenia tras, które mogą przynieść największe korzyści.

W kroku czwartym algorytmu sprawdza się możliwości połączenia dwóch tras w jedną. Służą do tego dwa kryteria. Pierwsze zapewnia, że nie zniszczy się tego, co dotychczas zostało zbudowane w trakcie działania algorytmu. Oznacza to, że można łączyć ze sobą tylko dwie trasy, w których wskazywane przez oszczędność $s(P_i, P_j)$ punkty obsługi P_i oraz P_j są krańcowymi punktami w swoich trasach, czyli umiejscowione są w nich na pierwszej lub ostatniej pozycji (rys. 6.15a).

Drugie z kryteriów sprawdzających możliwość połączenia dwóch tras dotyczy możliwości ładunkowych pojazdu. Jeżeli pierwsze kryterium połączenia dwóch tras jest spełnione, a także ładowność pojazdu pozwala na obsłużenie wszystkich klientów, trasy są ze sobą łączone (rys. 6.15b).



Rys. 6.15a. Dwie osobne (częściowo już zbudowane) trasy, z krańcowymi punktami P_i i P_j
Źródło: opracowanie własne.



Rys. 6.15b. Nowa trasa zawierająca połączenie w (P_i, P_j)
Źródło: opracowanie własne.

Krok piąty algorytmu to aktualizacja listy utworzonych tras, która dokonuje się po połączeniu, oraz wykreślenie z listy sprawdzonej już oszczędności. Kolejne iteracje algorytmu wykonywane są do momentu sprawdzenia wszystkich potencjalnych korzyści (oszczędności) wynikających z umieszczenia pary punktów obsługi (P_i, P_j) w jednej trasie. Jednakże działanie algorytmu może zostać przerwane wcześniej, jeżeli pomimo spełnienia pierwszego kryterium połączenie tras i tak nie będzie możliwe ze względu na przekraczanie dopuszczalnej ładowności lub długości pracy pojazdu.

Przedstawiony algorytm oszczędnościowego łączenia tras ma dwie wersje: sekwencyjną i równoległą. W wersji równoległej poddawane połączeniu mogą być

różne pary tras, tak jak to zostało przedstawione powyżej. W wersji sekwencyjnej rozpatrywana jest wciąż ta sama trasa, która jest łączona z kolejnymi do momentu, aż nie zostanie spełnione drugie kryterium sprawdzenia możliwości połączenia dwóch tras. Dla częściowo zbudowanej trasy pojazdu $(P_0, P_i, \dots, P_j, P_0)$ należy na posortowanej liście oszczędności znaleźć największą możliwą oszczędność $s(P_r, P_i)$ lub $s(P_j, P_s)$, która pozwoli na wskazanie odpowiednio punktów P_r lub P_s należących do swoich tras (i będących w nich krańcowymi punktami). Po dokonaniu połączenia trasy $(P_0, P_i, \dots, P_j, P_0)$ np. z jedną z tras: (P_0, P_r, \dots, P_0) lub (P_0, \dots, P_s, P_0) , powstaje nowa trasa $(P_0, \dots, P_r, P_i, \dots, P_j, P_0)$ lub $(P_0, P_i, \dots, P_j, P_s, \dots, P_0)$, która ponownie podlega dalszym możliwościom połączeń.

Na koniec rozważań dotyczących algorytmu oszczędnościowego łączenia tras warto dodać, że był on później rozwijany i modyfikowany przez wielu autorów. Jedną z modyfikacji to propozycja parametrycznej funkcji oszczędności H. Paesensa (1988) określonej wzorem:

$$s(P_i, P_j) = c(P_0, P_i) + c(P_j, P_0) - \gamma c(P_i, P_j) + \mu lc(P_0, P_i) + lc(P_j, P_0) \quad (6.8)$$

w której dodany człon w stosunku do funkcji (6.7) zachęca do zestawiania ze sobą punktów obsługi, których odległości od bazy znacznie się różnią. Wielkości γ i μ są parametrami tej funkcji, które autor przyjął na poziomie: $0 < \gamma \leq 3$ i $0 \leq \mu \leq 1$.

Idea oszczędności w algorytmie oszczędnościowego łączenia tras G. Clarka i J.W. Wrighta stała się podstawą innych metod rozwiązywania zagadnień układania tras pojazdów. Przykładem takiej metody jest algorytm zaproponowany przez R.H. Mole'a i S.R. Jamesona (1976), oparty na strategii wstawiania sekwencyjnego kolejnych punktów obsługi do budowanej trasy pojazdu.

repeat
 krok 1: inicjalizacja trasy pojazdu
repeat
 krok 2: wskazanie miejsca w trasie
 krok 3: wskazanie punktu obsługi
 krok 4: wstawienie punktu obsługi do trasy
until trasa jest dopuszczalna
 krok 5: optymalizacja trasy
until wszystkie punkty obsługi są umieszczone w trasach

Rys. 6.16. Algorytm R.H. Mole'a i S.R. Jamesona

Źródło: opracowanie własne.

W kroku pierwszym algorytmu należy zainicjować trasę k -tego pojazdu przez arbitralny wybór pierwszego punktu obsługi, który zostanie przez pojazd odwiedzony: $P_0 \rightarrow P_i \rightarrow P_0$. Następnie w kroku drugim dla wszystkich punktów obsługi P_k , jakie nie zostały jeszcze przyporządkowane do żadnej trasy, należy obliczyć wielkość $\alpha(P_r, P_j)$, a jej najmniejsza wartość wskaże parę punktów (P_i^*, P_j^*) znajdujących się

już w częściowo zbudowanej trasie, pomiędzy które wybrany punkt obsługi zostanie wstawiony. Wielkość $\alpha(P_i, P_j)$ obliczana jest zgodnie ze wzorem:

$$\alpha(P_i, P_j) = c(P_i, P_k) - c(P_k, P_j) - \lambda c(P_i, P_j) \quad (6.9)$$

Krok trzeci algorytmu to wybór punktu obsługi P_k^* , który zostanie wstawiony do częściowo zbudowanej tras, na miejscu wskazanym w kroku drugim. Wybrany zostanie punkt P_k^* , dla którego wartość wyrażenia (6.10) jest największa.

$$\beta(P_i^*, P_j^*) = \mu c(P_0, P_k) - \alpha(P_i, P_j) \quad (6.10)$$

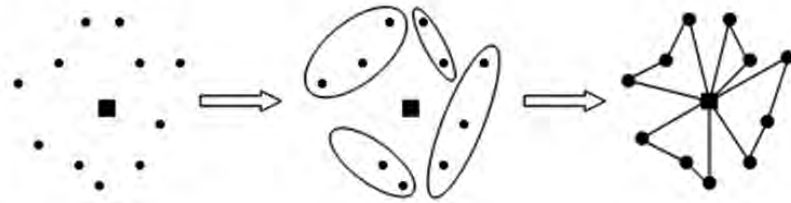
Wykorzystane są tutaj dwa parametry: λ oraz μ . Gdy $\lambda > 0$ oraz $\mu = \infty$, punkt P_k^* , który zostanie wstawiony do trasy, odpowiadać będzie najdalej położonemu punktowi od bazy. Gdy $\lambda = 0$ oraz $\mu = 0$, punkt P_k^* odnosić się będzie do najmniejszej sumy odległości pomiędzy dwoma sąsiednimi punktami. Wskazany punkt obsługi P_k^* zostanie wstawiony w kroku czwartym do trasy pomiędzy parę sąsiednich punktów P_i^* oraz P_j^* pod warunkiem, że trasa nie spowoduje przekroczenia dodatkowych ograniczeń związanych z pojazdem (np. ładowności). Jeżeli dodatkowe ograniczenia nie zostaną spełnione oraz żaden z pozostałych nieobsłużonych jeszcze punktów nie będzie mógł zostać dołączony do aktualnie rozpatrywanej trasy, wtedy uważana jest ona za zamkniętą, a następnie inicjowana jest nowa trasa. W ostatnim, piątym kroku algorytmu należy spróbować dokonać poprawy kolejności obsługi punktów przez pojazd w zbudowanej trasie w celu sprawdzenia, czy możliwe jest uzyskanie krótszej trasy niż ta, która została skonstruowana w krokach drugim, trzecim i czwartym.

Przedstawiona powyżej procedura jest powtarzana do momentu, aż wszystkie punkty obsługi zostaną jednoznacznie przyporządkowane do swoich pojazdów. Algorytm Mole'a i Jamesona jako metoda układania tras pojazdów zaliczany jest przez niektórych badaczy do grupy metod dekompozycyjnych. Wynika to z tego, że pomimo stopniowej budowy tras pojazdów są one jeszcze poddawane w ostatnim kroku odrębnej postoptimalizacji.

6.2. Heurystyki dekompozycyjne

Odrębną klasę algorytmów heurystycznych dla zagadnień układania tras wielu pojazdów stanowią metody opierające się na strategii rozdzielającej problem przyporządkowania punktów obsługi do pojazdów oraz problem ustalania kolejności odwiedzania tych punktów przez dany pojazd (rys. 6.17). Strategia ta może być realizowana w dwóch kierunkach: najpierw rejon obsługi, potem kolejność, lub

odwrotnie – najpierw kolejność, potem rejon. W tym drugim wariantcie celem jest budowa jednej niedopuszczalnej trasy dla wszystkich obsługiwanych przez bazę punktów, by następnie podzielić ją na mniejsze podtrasy obsługiwane przez poszczególne pojazdy. Najczęściej rozpatrywanym podejściem jest ustalenie najpierw rejonów, które mają być obsłużone przez pojazdy, jakimi dysponuje baza, a potem określenie kolejności, w jakiej punkty mają być obsłużone przez przydzielone im pojazdy.



Rys. 6.17. Dwuetapowa konstrukcja rozwiązania w problemie wielu komiwojażerów
Źródło: opracowanie własne.

Do tej klasy metod należy algorytm zaproponowany przez B.E. Gilletta i L.R. Millera (1974), nazywany algorytmem *sweep* ('zagarniać, zamiatać'). Przeznaczony jest on do zadań wielu komiwojażerów, w których można przedstawić punkty obsługi P_i w przestrzeni euklidesowej za pomocą współrzędnych (x_i, y_i) . Przyjmując założenie, że baza pojazdów znajduje się w środku koła obejmującego wszystkie punkty obsługi, można poprowadzić promień R^* przechodzący przez arbitralnie wybrany początkowy punkt obsługi P_1 . Każdy następny promień R_i poprowadzony przez pozostałe punkty obsługi P_i będzie nachylony do promienia R^* pod pewnym kątem. W związku z tym każdy punkt obsługi P_i będzie się charakteryzował dwiema wielkościami: kątem nachylenia α_i przechodzącego przez niego promienia R_i (odległością kątową) oraz odległością $c(P_i, P_0)$ punktu P_i od bazy. Dla arbitralnie wybranego początkowego punktu obsługi P_1 odległość kątowa $\alpha_1 = 0$. Postać algorytmu *sweep* Gilletta i Millera została przedstawiona w postaci pseudokodu na rysunku 6.18.

W kroku pierwszym algorytmu należy wskazać arbitralnie początkowy punkt obsługi P_1 i poprowadzić prostą przechodzącą przez bazę oraz przez ten punkt. Następnie dla wszystkich punktów P_i należy obliczyć odległość kątową α_i od wskazanego punktu początkowego i posortować je w porządku rosnącym. Po zainicjalizowaniu nowej trasy pojazdu w kroku drugim przyporządkowanie poszczególnych punktów obsługi P_i odbywa się w kroku trzecim algorytmu.

Począwszy od najmniejszego α_i , należy stopniowo przypisywać punkty obsługi P_i do pojazdu, aż do momentu, kiedy trzeba będzie użyć nowego środka transportu. Odbywa się to przez obracanie promienia R^* przechodzącego przez początkowy punkt P_1 w ustalonym wcześniej jednym z dwóch kierunków: zgodnym z ruchem wskazówek zegara lub przeciwnym do ruchu wskazówek zegara (rys. 6.19).

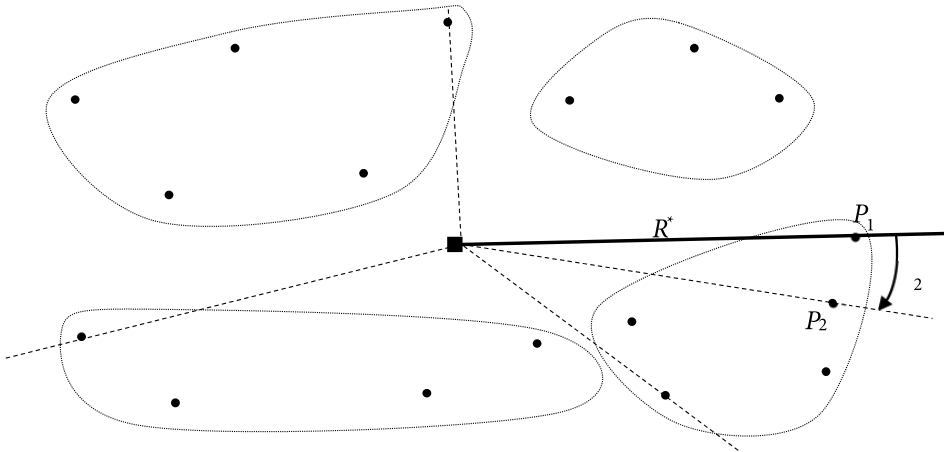
```

krok 1: inicjalizacja algorytmu
repeat
  krok 2: inicjalizacja trasy
  repeat
    krok 3: przyporządkowanie punktu do pojazdu
  until trasa jest dopuszczalna
until wszystkie punkty obsługi są umieszczone w trasach

repeat
  krok 4: rozwiązanie problemu komiwojażera
until wszystkie rejony mają ustaloną trasę

```

Rys. 6.18. Algorytm *sweep*
Źródło: opracowanie własne.

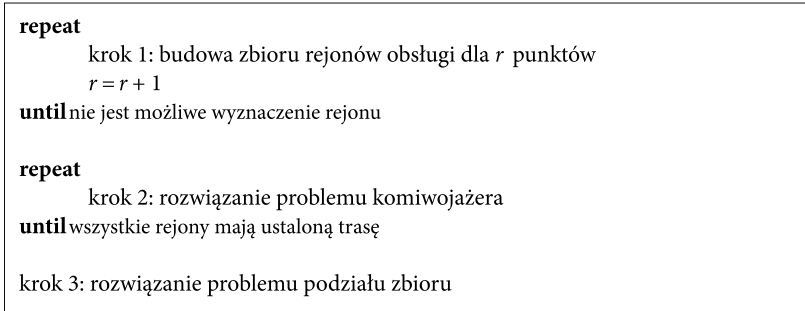


Rys. 6.19. Rejony obsługi w algorytmie *sweep*
Źródło: opracowanie własne.

W wyniku przyporządkowania wszystkich punktów obsługi do poszczególnych pojazdów otrzymujemy zbiór rejonów obsługi, dla których należy odrębnie w ostatnim kroku algorytmu rozwiązać problem jednego komiwojażera.

Do pewnego stopnia ideę procedury *sweep* wykorzystuje przedstawiony przez D.M. Ryana i B.A. Fostera (Ryan i in., 1993) algorytm *petal* ('płatek').

Rejony obsługi dla pojazdów są tworzone w ten sposób, że wszystkie punkty obsługi należące do danego rejonu stanowią obszar będący pewnym wycinkiem koła o promieniu, którego początek znajduje się w punkcie wyznaczonym przez bazę (tak jak w metodzie *sweep*). W przeciwieństwie do metody *sweep* celem algorytmu jest wyznaczenie wszystkich dopuszczalnych rejonów obsługi dla pojazdów, by wybrać z nich zbiór tych, dla których łączna długość zbudowanych dla nich tras pojazdów będzie jak najmniejsza (rys. 6.20).

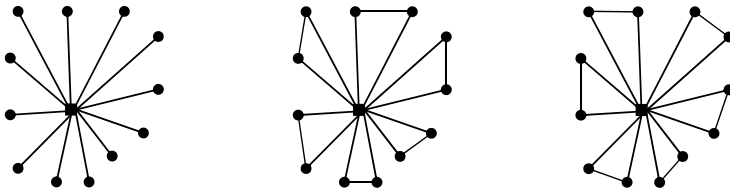


Rys. 6.20. Algorytm *petal*

Źródło: opracowanie własne.

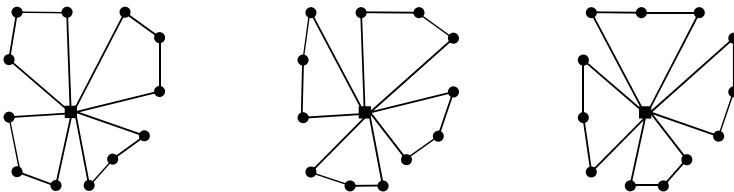
Krok pierwszy algorytmu to budowa wszystkich dopuszczalnych rejonów obsługi dla pojazdów. Oznacza to, że rejon składać się może zarówno z jednego punktu obsługi, jak i kilku takich punktów. Jednakże każdy rejon musi zawierać te punkty obsługi, dla których łączny popyt (podaż) nie przekracza ładowności pojazdu oraz punkty te i tylko te znajdują się obok siebie (na jednym wycinku koła). W praktyce należy stopniowo wyznaczać wszystkie rejonu obsługi składające się kolejno z jednego punktu obsługi, a następnie dla dwóch, trzech itd. Krok ten jest powtarzany do momentu, gdy nie jest już możliwe wyznaczenie rejonu obsługi dla r punktów, ponieważ za każdym razem dopuszczalna ładowność pojazdu zostanie przekroczona.

Dla utworzonych w kroku pierwszym rejonów obsługi należy zbudować i rozwiązać problem komiwojażera, otrzymując w ten sposób trasę o ustalonym koszcie (długości) c_i (rys. 6.21a i 6.21b).



Rys. 6.21a. Rejonu obsługi i odpowiadające im trasy składające się, odpowiednio, z jednego i dwóch punktów obsługi

Źródło: opracowanie własne.



Rys. 6.21b. Rejonu obsługi i odpowiadające im trasy składające się z trzech punktów obsługi

Źródło: opracowanie własne.

Krok trzeci to budowa i rozwiązanie problemu podziału zbioru, w którego wyniku uzyskiwany jest podzbiór tras (składających się z jednego lub wielu punktów obsługi) stanowiący rozwiązanie końcowe (6.11)–(6.13):

$$\min. F(\mathbf{x}) = \sum_{l \in L} c_l x_l \quad (6.11)$$

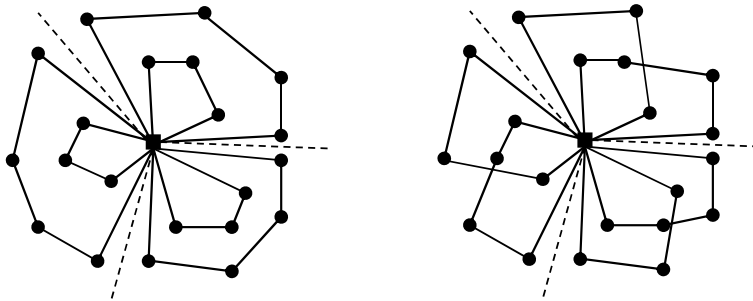
$$\sum_{l \in L} a_{il} x_l = 1 \quad \text{dla } i = 1, \dots, N \quad (6.12)$$

$$x_l \in \{0, 1\} \quad \text{dla } l \in L \quad (6.13)$$

gdzie:

- L – zbiór wszystkich utworzonych dopuszczalnych tras pojazdów;
- a_{il} – parametr przyjmujący wartość 1, jeżeli i -ty punkt obsługi należy do trasy l , natomiast 0, gdy do niej nie należy;
- c_l – koszt (długość) l -tej trasy pojazdu.

Pewną modyfikacją przedstawionego powyżej algorytmu jest algorytm *2-petal* zaproponowany przez J. Renauda, F.F. Boctora i G. Laportea (Renaud i in., 1996), w którym dla danego wycinka koła (sektora) tworzone są dwie trasy, a nie tylko jedna (rys. 6.22). W tym wypadku łączne zapotrzebowanie wszystkich punktów obsługi nie może przekroczyć dwukrotnej dopuszczalnej ładowności pojazdu. Dwie trasy tworzone są w oparciu o punkty początkowe, które są w danym sektorze najbardziej od siebie oddalone, natomiast pozostałe punkty sektora są do nich dołączane z wykorzystaniem metody najniższego kosztu wstawiania punktu do trasy.



Rys. 6.22. Rejony obsługi i odpowiadające im trasy w algorytmie *2-petal* (po lewej: jedna trasa zawiera się w drugiej; po prawej: trasy się przecinają)

Źródło: opracowanie własne.

Kolejnym algorytmem realizującym strategię „najpierw rejon, potem kolejność” jest algorytm wykorzystujący uogólnione zagadnienie przydziału, a przedstawiony przez M. Fishera i R. Jaikumara (1981) (rys. 6.23).

Pierwszy krok algorytmu stanowi wybór arbitralnie ustalonych przez decydenta punktów źródłowych P_k , do których zostaną przyporządkowane pojazdy (rys. 6.24).

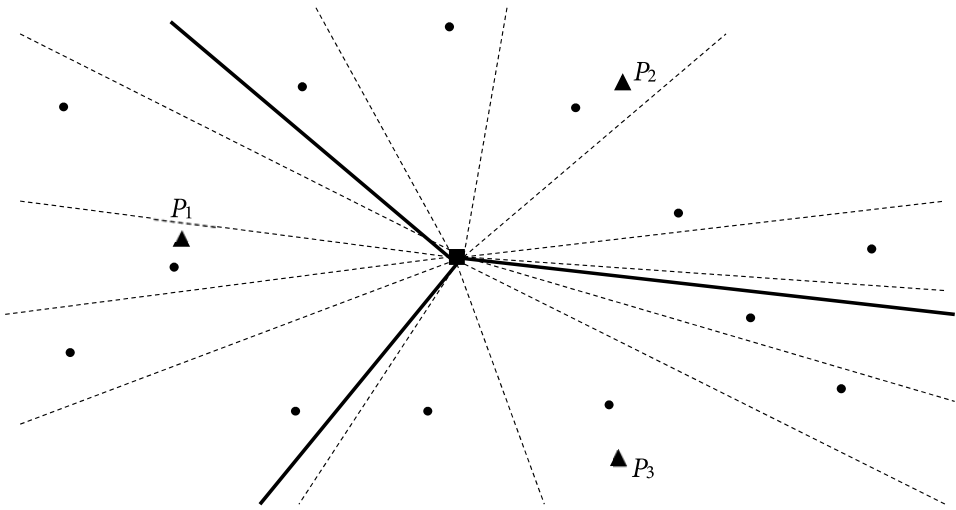
krok 1: wybór punktów źródłowych
 krok 2: rozwiązanie uogólnionego zadania przydziału

repeat
 krok 3: rozwiązanie problemu komiwojażera

until wszystkie rejony mają ustaloną trasę

Rys. 6.23. Algorytm M. Fishera i R. Jaikumara

Źródło: opracowanie własne.



Rys. 6.24. Rejony obsługi w algorytmie M. Fishera i R. Jaikumara

Źródło: opracowanie własne na podstawie (Fisher i Jaikumar, 1981).

Stąd też ich liczba może być mniejsza lub równa liczbie pojazdów, jaka jest w dyspozycji bazy. Problem, przed jakim stoi decydent, to odpowiedni wybór punktów, które będą stanowić załazek rejonów obsługi dla pojazdów. Jak stwierdzają przywoływani autorzy, mogą to być punkty, dla których wielkość dostawy (lub odbioru produktu) stanowi więcej niż połowę ładowności pojazdu, ponieważ i tak niemożliwe byłoby umieszczenie tych punktów w jednej trasie. Inny powód wyboru właściwych punktów źródłowych może być związany z rozlokowaniem wszystkich punktów obsługi wokół bazy. Bardzo często rozlokowanie takie ma charakter promienisty, gdzie punkty obsługi zlokalizowane są wzdłuż głównych ciągów komunikacyjnych. W takim wypadku naturalnym wyborem punktu źródłowego będzie wskazanie punktu najbardziej oddalonego od bazy wzdłuż rozpatrywanego ciągu komunikacyjnego. Punkty źródłowe mogą zostać także wyznaczone automatycznie zgodnie z opisaną przez autorów regułą podziału obszaru, na którym znajdują się wszyscy klienci, na mniejsze obszary, adekwatne do zgłaszanego przez nich

zapotrzebowania. Wtedy punktami źródłowymi nie są rzeczywiste punkty obsługi, lecz leżące w tej samej odległości od bazy fikcyjne punkty, wobec których formowane są w dalszej kolejności rejonu obsługi.

Krok drugi algorytmu Fishera i Jaikumara wymaga konstrukcji oraz rozwiązania uogólnionego liniowego zadania przydziału, którego postać określona jest modelem programowania matematycznego:

$$\min. F(\mathbf{y}) = \sum_{k=1}^K \sum_{i=1}^N c_{ik} y_{ik} \quad (6.13)$$

$$\sum_{i=1}^N q_i y_{ik} \leq Q_k \quad \text{dla } k = 1, \dots, K \quad (6.14)$$

$$\sum_{i=1}^N y_{ik} = 1 \quad \text{dla } i = 1, \dots, N \quad (6.15)$$

$$y_{ik} \in \{0, 1\} \quad \text{dla } k = 1, \dots, K \text{ oraz } i = 1, \dots, N \quad (6.16)$$

gdzie:

- N – liczba wszystkich obsługiwanych punktów (węzłów);
- K – liczba wszystkich pojazdów;
- Q_k – ładowność k -tego pojazdu;
- q_i – zapotrzebowanie i -tego punktu obsługi;
- c_{ik} – koszt wstawienia i -tego punktu obsługi do trasy k -tego pojazdu;
- y_{ik} – zmienna decyzyjna przyjmująca wartość 1, gdy i -ty punkt obsługi przyporządkowany zostanie do trasy k -tego pojazdu, lub 0, gdy nie zostanie przyporządkowany.

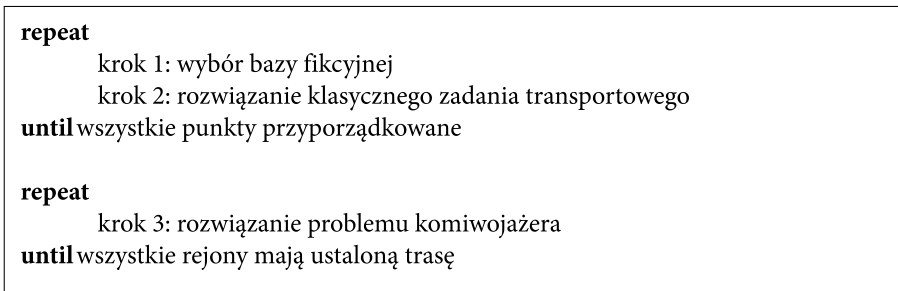
Każdy punkt obsługi musi być przyporządkowany do pojazdu w sposób jednoznaczny, o czym zapewniają ograniczenia wyrażone równościami (6.15), natomiast zbiory punktów obsługi w poszczególnych rejonach warunkowane są nierównościami przedstawionymi w ograniczeniach (6.14). Celem prezentowanego uogólnionego zadania przydziału jest minimalizacja funkcji wyrażającej całkowity koszt przyporządkowania punktów do pojazdów będących w dyspozycji bazy. Na szczególną uwagę zasługują tutaj współczynniki funkcji celu, które wyrażają koszt wstawienia i -tego punktu obsługi do rejonu obsługiwanego przez k -ty pojazd. Wyznaczane są one zgodnie z formułą:

$$c_{ik} = \min. [c(P_i, P_0) + c(P_i, P_k) + c(P_k, P_0), c(P_0, P_k) + c(P_k, P_i) + c(P_i, P_0)] - [c(P_0, P_k) + c(P_k, P_0)] \quad (6.17)$$

Tak samo jak w wypadku algorytmu *sweep*, w ostatnim kroku algorytmu należy dla każdego rejonu oddzielnie rozwiązać problem komiwojażera, otrzymując zbiór finalnych tras pojazdów.

W polskiej literaturze z tego zakresu na uwagę zasługuje dwuetapowy algorytm przedstawiony przez A. Całczyńskiego noszący nazwę BF-WOT (baz fikcyjnych

i wymiany odcinków tras) (Całczyński, 1992; Jadczak, 2001). W pierwszym etapie, w którym konstruowane są rejony obsługi dla pojazdów, jest on trochę podobny do algorytmu Fishera i Jaikumara. Także w tym wypadku na początku wybierane są punkty źródłowe, dla których budowane jest klasyczne zadanie transportowe.



Rys. 6.25. Algorytm BF-WOT

Źródło: opracowanie własne.

W przeciwieństwie do wcześniej prezentowanego algorytmu rejony zasięgu działania pojazdów dla poszczególnych punktów obsługi budowane są stopniowo. W kroku pierwszym należy spośród nieprzyporządkowanych do żadnego pojazdu punktów obsługi wybrać punkt, który jest zlokalizowany w najdalszej odległości od bazy. Do tego punktu zostanie wysłany pojazd z żadaną ilością produktu, skąd rozpocznie obsługę kolejnych przyporządkowanych do niego klientów. Wybrany w ten sposób punkt startowy pojazdu nazywany jest bazą fikcyjną. Baza fikcyjna oraz baza rzeczywista stanowią podstawę budowy klasycznego zadania transportowego z dwoma dostawcami. Podaż bazy rzeczywistej jest równa sumie popytów wszystkich nieodwiedzonych punktów obsługi (z wyjątkiem punktu obsługi, jaki wybrany został jako baza fikcyjna), od której należy odjąć niewykorzystaną jeszcze ładowność pojazdu. Natomiast podaż bazy fikcyjnej jest równa różnicy ładowności wysłanego do niej pojazdu i popytu punktu obsługi stanowiącego bazę fikcyjną. Odbiorcami w tak konstruowanym klasycznym zadaniu transportowym są wszystkie nieodwiedzone punkty obsługi, których popyty są równe ich rzeczywistemu zapotrzebowaniu na produkt. Wreszcie macierz kosztów zadania transportowego stanowią odległości pomiędzy punktami obsługi i bazami (rzeczywistą i fikcyjną).

Przykład

W tabeli 6.1 przedstawiono odległości pomiędzy bazą a wszystkimi punktami obsługi ze wskazaniem tego, który zostanie bazą fikcyjną (punkt P_j jako najbardziej oddalony od bazy).

Tabela 6.1. Odległości od bazy oraz popyty punktów obsługi

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	Razem
Baza	5	2	6	3	7	2	9	1	8	7	
Popyt	20	15	20	20	15	15	20	20	15	20	180

Źródło: opracowanie własne.

Do tego punktu wysłany zostanie pojazd o ładowności 80 jednostek ładunkowych. Tym samym w klasycznym zadaniu transportowym, które należy zbudować, drugim dostawcą (bazą fikcyjną) będzie punkt P_7 , o podaży wynoszącej 60 jednostek ładunkowych. W tabeli 6.2 zaprezentowano przykład budowy zadania transportowego w pierwszym etapie metody BF-WOT.

Tabela 6.2. Przykład liczbowy budowy klasycznego zadania transportowego w metodzie BF-WOT

	P_1	P_2	P_3	P_4	P_5	P_6	P_8	P_9	P_{10}	Podaż
Baza	5	2	6	3	7	2	1	8	7	100
P_7 (BF)	4	6	7	4	8	2	6	5	3	60
Popyt	20	15	20	20	15	15	20	15	20	160

Źródło: opracowanie własne.

W wyniku rozwiązania przedstawionego powyżej klasycznego zadania transportowego uzyskany zostanie rejon obsługi dla pojazdu uprzednio wysłanego do punktu P_7 (tab. 6.3).

Tabela 6.3. Rozwiązanie klasycznego zadania transportowego w metodzie BF-WOT (rejon obsługi pojazdu)

	P_1	P_2	P_3	P_4	P_5	P_6	P_8	P_9	P_{10}	Podaż
Baza	0	15	20	20	15	10	20	0	0	100
P_7 (BF)	20	0	0	0	0	5	0	15	20	60
Popyt	20	15	20	20	15	15	20	15	20	160

Źródło: opracowanie własne.

Przyjmując podstawowe założenia dotyczące sformułowania problemu układania tras pojazdów (jednoznaczne przyporządkowanie punktów obsługi do pojazdów), do wyznaczenia rejonu obsługi pojazdu w prezentowanym przykładzie należy wybrać punkty: P_7 , P_1 , P_9 oraz P_{10} . Łączny popyt wszystkich klientów dla wskazanego rejonu wyniesie 75 jednostek ładunkowych, co oznacza, że ładowność pojazdu nie zostanie w pełni wykorzystana (pozostaje wolne miejsce na 5 jednostek ładunkowych). Jeżeli dokonana zostanie relaksacja tego ograniczenia (dopuszczona zostanie obsługa rozdzielona), do rejonu obsługi pojazdu należy dołączyć także punkt P_6 .

W wyniku konstrukcji rejonu obsługi dla pierwszego pojazdu nieprzyporządkowane zostaną jeszcze punkty: P_2 , P_3 , P_4 , P_5 , P_8 oraz punkt P_6 . Popyt punktu P_6 będzie równy 15 (jeżeli w rejonie obsługi pojazdu nie zostanie on uwzględniony) lub 10 (jeżeli w rejonie obsługi pojazdu zostanie on uwzględniony). Powyższą procedurę przedstawioną w tabelach 6.1–6.3 należy powtórzyć w celu wyznaczenia kolejnego rejonu obsługi. Ostatni rejon obsługi wyznaczony zostanie bez konieczności budowy i rozwiązania klasycznego zadania transportowego i zawierać będzie wszystkie punkty obsługi, których łączny popyt nie przekroczy założonej ładowności pojazdu.

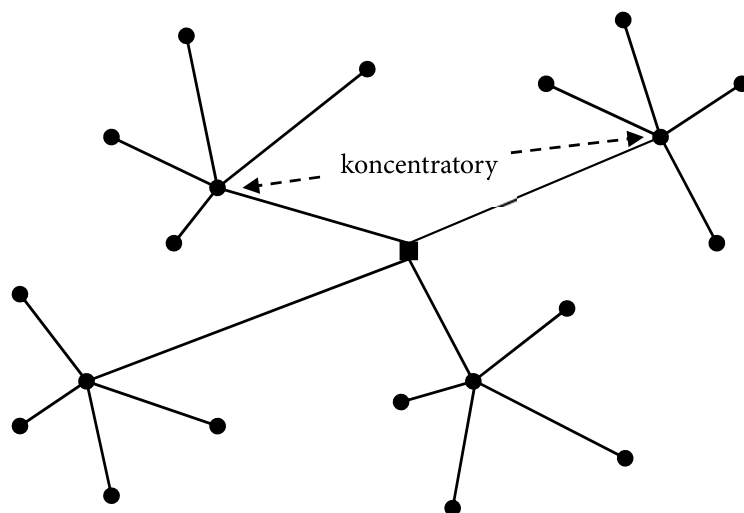
Drugi etap algorytmu A. Całczyńskiego (krok trzeci) to konstrukcja i rozwiązanie zadań komiwojażera dla każdego pojazdu na podstawie wyznaczonego dla niego w etapie pierwszym rejonu obsługi. Algorytm baz fikcyjnych A. Całczyńskiego sukcesywnie konstruuje poszczególne rejony obsługi, inicjując je punktami obsługi, które są najbardziej oddalone od bazy spośród jeszcze nieprzyporządkowanych do żadnego pojazdu. Nieco inne podejście do konstrukcji rejonów pojazdów mamy w algorytmie zaproponowanym przez J. Bramela i D. Simchi-Levia (1995) (rys. 6.26).

krok 1: wybór potencjalnych koncentratorów
 krok 2: sformułowanie i rozwiązanie zadania lokalizacji
repeat
 krok 3: umieszczenie punktu w trasie
until wszystkie rejony mają ustaloną trasę

Rys. 6.26. Algorytm „najpierw trasa, potem rejon”

Źródło: opracowanie własne.

Budowa rejonów poszczególnych pojazdów opiera się na wybranych punktach obsługi nazywanych koncentratorami, do których przyporządkowywane („podłączane”) są pozostałe punkty (rys. 6.27). Proces ten można porównać do konstrukcji sieci telekomunikacyjnej składającej się z głównych węzłów (koncentratory) o określonej przepustowości oraz podłączonych do nich punktów końcowych (terminali). Do danego węzła głównego może być podłączona taka liczba terminali, aby dopuszczalna przepustowość koncentratora nie została przekroczona.



Rys. 6.27. Rejony obsługi wyznaczone w oparciu o koncentratory
Źródło: opracowanie własne.

W kroku pierwszym należy wybrać potencjalnych kandydatów na koncentratory. Metoda pozwala na korzystanie z wiedzy i doświadczenia decydenta, który wskaże punkty obsługi, jakie zostaną umieszczone w jednym rejonie obsługi (a tym samym jeden z nich zostanie koncentratorem, a pozostałe zostaną do niego przyporządkowane). Jednakże wspomniani autorzy proponują, aby zbiór potencjalnych koncentratorów obejmował zbiór wszystkich punktów obsługi, czyli każdy punkt obsługi ma jednakowe szanse, że zostanie wskazany jako inicjator rejonu obsługi pojazdu.

Istotą prezentowanego algorytmu jest krok drugi, w którym należy skonstruować zagadnienie lokalizacji koncentratorów z ograniczonymi przepustowościami. W tym celu należy przyjąć następujące oznaczenia:

- x_{ik} – zmienna decyzyjna przyjmująca wartość 1, jeżeli terminal i (i -ty punkt obsługi) jest przyporządkowany do k -tego koncentratora;
- y_k – zmienna decyzyjna przyjmująca wartość 1, jeżeli koncentrator ulokowany został w k -tym miejscu (k -ty punkt obsługi został koncentratorem);
- c_{ik} – koszt przyporządkowania i -tego terminala (i -tego punktu obsługi) do k -tego koncentratora;
- v_k – koszt umieszczenia koncentratora w k -tym punkcie (uznania k -tego punktu obsługi za koncentrator);
- Q_k – przepustowość k -tego koncentratora (ładowność pojazdu użytego do obsługi k -tego rejonu);
- w_i – zapotrzebowanie na przepustowość i -tego terminala (zapotrzebowanie lub podaż i -tego punktu obsługi).

Postać modelu lokalizacji koncentratorów z ograniczonymi przepustowościami, na którego podstawie należy wskazać punkty obsługi będące inicjatorami rejonów dla poszczególnych pojazdów, jest następująca:

$$\min. (\mathbf{x}, \mathbf{y}) = \sum_{i=1}^N \sum_{k=1}^K c_{ik} x_{ik} + \sum_{k=1}^K v_k y_k \quad (6.18)$$

$$\sum_{k=1}^K x_{ik} = 1 \quad \text{dla } i = 1, \dots, N \quad (6.19)$$

$$x_{ik} \leq y_k \quad \text{dla } i = 1, \dots, N \text{ oraz } k = 1, \dots, K \quad (6.20)$$

$$x_{ik} \in \{0, 1\} \quad \text{dla } i = 1, \dots, N \text{ oraz } k = 1, \dots, K \quad (6.21)$$

$$y_k \in \{0, 1\} \quad \text{dla } k = 1, \dots, K \quad (6.22)$$

Ograniczenia wyrażone wzorami (6.19)–(6.20) zapewniają, że każdy terminal (punkt obsługi) przyporządkowany jest tylko do jednego koncentratora, a jednocześnie ich łączne zapotrzebowanie na przepustowość (podaż lub popyt przyporządkowanych punktów obsługi) nie przekracza dopuszczalnej przepustowości koncentratora (dopuszczalnej ładowności wyznaczonego do obsługi rejonu pojazdu). Celem sformułowanego problemu jest znalezienie optymalnej lokalizacji koncentratorów (wskazanie, które punkty obsługi zostaną koncentratorami) oraz przyporządkowanie pozostałych punktów obsługi do koncentratorów w taki sposób, aby łączny koszt umieszczenia w k -tych punktach koncentratorów oraz przyporządkowania do nich pozostałych punktów obsługi był jak najniższy (6.18).

Wyjaśnienia wymagają tutaj sposoby obliczenia kosztów umieszczenia koncentratora w k -tym punkcie oraz przyporządkowania do niego i -tego punktu obsługi. Autorzy proponują, aby koszt v_k obliczony został jako koszt wysłania pojazdu do tego punktu: $v_k = 2d_{0k}$. Natomiast jeśli chodzi o obliczanie kosztu przyporządkowania i -tego punktu obsługi do k -tego koncentratora c_{ik} , proponuje się tutaj dwa sposoby. Pierwszy z nich jest kosztem wstawienia i -tego punktu obsługi pomiędzy bazę a k -ty koncentrator: $c_{ik} = d_{0i} + d_{ik} - d_{0k}$. Drugi ze sposobów nazywany jest kosztem bezpośrednim i obliczany jest jako: $c_{ik} = 2d_{ik}$.

Uzyskane w kroku drugim rozwiązanie stanowi wyznaczenie rejonów obsługi dla poszczególnych pojazdów, które w kroku trzecim należy przekształcić w konkretne trasy pojazdów przez minimalizację kosztu wstawienia punktu obsługi do trasy pojazdu.

Ostatnim algorytmem z grupy metod „najpierw rejon, potem trasa”, o którym należy wspomnieć, jest dwufazowa procedura przedstawiona przez N. Christofidesa, A. Mingozziego i P. Totha (Christofides i in., 1979). Składa się ona z fazy sekwencyjnej konstrukcji tras (rys. 6.28) oraz fazy równoległej konstrukcji tras pojazdów (rys. 6.29).

```

repeat
  krok 1: inicjalizacja trasy pojazdu
  repeat
    krok 2: umieszczenie punktu w trasie
    krok 3: poprawa kolejności obsługi punktów w trasie
  until ładowność pojazdu nie jest przekroczona
until wszystkie punkty przyporządkowane

```

Rys. 6.28. Faza sekwencyjnej konstrukcji tras w algorytmie N. Christofidesa, A. Mingozziiego i P. Totha

Źródło: opracowanie własne.

W kroku pierwszym algorytmu należy wybrać dowolny punkt obsługi r , który zainicjalizuje trasę pierwszego pojazdu. Następnie dla każdego punktu obsługi, który nie jest jeszcze przyporządkowany do trasy pojazdu, należy policzyć wielkość δ_i według wzoru:

$$\delta_i = d_{0i} + \lambda d_{ir} \quad (6.23)$$

gdzie $\lambda \geq 1$ jest parametrem ustalonym przez decydenta. W kroku drugim w aktualnie rozpatrywanej trasie pojazdu umieszczony zostanie punkt obsługi i^* , dla którego $\delta_{i^*} = \min_{i \in S_k} \{\delta_i\}$, gdzie S_k jest zbiorem wszystkich punktów obsługi, które mogłyby być wstawione do trasy pojazdu bez naruszenia warunku jego dopuszczalnej ładowności. Dołączony do trasy punkt i^* powoduje jej powiększenie i należy spróbować poprawić jej całkowitą długość przez zmianę kolejności odwiedzania przez pojazd poszczególnych punktów. Procedura opisana w krokach drugim i trzecim jest powtarzana do momentu, w którym żaden punkt obsługi nie może zostać już dołączony do trasy pojazdu, ponieważ ładowność pojazdu została wykorzystana. W takiej sytuacji należy zainicjować nową trasę kolejnego pojazdu, a następnie powtórzyć kroki pierwszy, drugi i trzeci.

```

krok 1: inicjalizacja tras pojazdów
repeat
  repeat
    krok 2: przyporządkowanie punktu obsługi do trasy
  until wszystkie punkty przyporządkowane
  repeat
    krok 3: wstawienie punktu obsługi do trasy
    krok 4: poprawa kolejności obsługi punktów w trasie
  until wszystkie punkty umieszczone w trasie
until wszystkie trasy pojazdów zbudowane

```

Rys. 6.29. Faza równoległej konstrukcji tras w algorytmie N. Christofidesa, A. Mingozziiego i P. Totha

Źródło: opracowanie własne.

W fazie równoległej omawianego algorytmu inicjowane są jednocześnie wszystkie k trasy pojazdów, gdzie k jest liczbą uzyskanych tras z fazy sekwencyjnej. Inicjalizacja tych tras następuje w wyniku wyboru punktów początkowych r_1, \dots, r_k . Następnie w kroku drugim należy dokonać skojarzenia i -tego punktu obsługi z daną l -tą trasą przez policzenie wielkości ε_i^l według wzoru:

$$\varepsilon_i^l = d_{0i} + \mu d_{in} - d_{0r_l} \quad (6.24)$$

gdzie $\mu \geq 1$ jest parametrem ustalonym przez decydenta. Z l -tą trasą zostanie skojarzony punkt i , dla którego $\varepsilon_i^{l*} = \min_l \{\varepsilon_i^l\}$. Krok drugi należy powtarzać do momentu, aż wszystkie punkty obsługi zostaną skojarzone ze swoimi trasami.

W kroku trzecim następuje stopniowa konstrukcja poszczególnych tras pojazdów. Dla każdego punktu i skojarzonego z trasą l należy policzyć wielkość τ_i zgodnie z formułą:

$$\tau_i = \varepsilon_i^{l*} - \varepsilon_i^l \quad (6.25)$$

gdzie $\varepsilon_i^{l*} = \min_l \{\varepsilon_i^l\}$. Do trasy l wstawiony zostanie punkt i^* , dla którego $\tau_{i^*} = \max_{i \in S_l} \{\tau_i\}$, gdzie S_l jest zbiorem wszystkich punktów obsługi, które nie zostały jeszcze umieszczone w trasie l , a są z nią skojarzone. Podobnie jak w fazie sekwencyjnej, stopniowo rozbudowywana trasa jest za każdym razem poprawiana w kroku czwartym. Faza równoległej konstrukcji tras jest powtarzana do momentu, kiedy wszystkie trasy pojazdów zostaną skonstruowane.

Odwrotne podejście do dotychczas zaprezentowanych algorytmów dekompozycyjnych stanowić może metoda realizująca strategię „najpierw trasa, potem rejon” (rys. 6.30).

krok 1: rozwiązanie problemu komiwojażera

repeat

krok 2: przyporządkowanie kolejnych punktów trasy do pojazdu

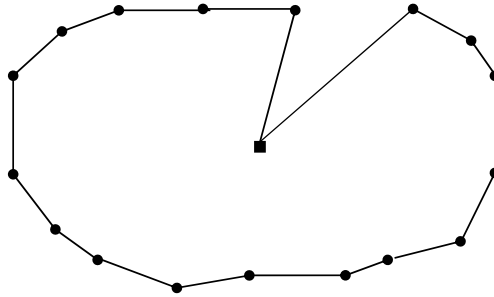
until wszystkie punkty przyporządkowane

Rys. 6.30. Algorytm „najpierw trasa, potem rejon”

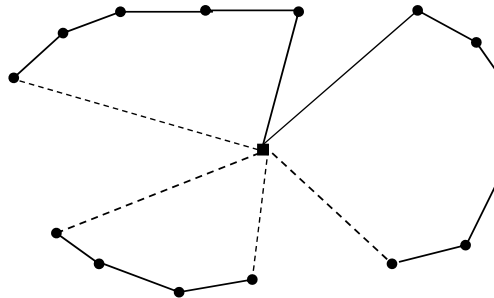
Źródło: opracowanie własne.

W pierwszym kroku algorytmu należy zbudować zadanie komiwojażera składające się z wszystkich punktów obsługi będących w zasięgu działania bazy. W wyniku rozwiązania zadania komiwojażera powstanie jedna supertrasa, która z punktu widzenia założonej ładowności pojazdu jest niedopuszczalna. W związku z tym w kolejnym kroku algorytmu należy sukcesywnie budować rejonny obsługi dla poszczególnych pojazdów, uwzględniając ich dopuszczalne ładowności, jednakże zachowując kolejność obsługi poszczególnych punktów wyznaczoną rozwiązaniem

uzyskany w kroku pierwszym (rys. 6.31a i 6.31b). Pojazd, wyruszając z bazy, odwiedza kolejne punkty obsługi według kolejności zgodnej z supertrasą wyznaczoną wcześniej. W sytuacji, gdy ładowność pojazdu zostanie wyczerpana, powraca on do bazy, skąd do dalszej obsługi punktów wyznaczany jest kolejny pojazd, począwszy od kolejnego nieodwiedzzonego w trasie punktu obsługi. Procedura jest powtarzana do momentu, aż wszystkie punkty obsługi supertrasy zostaną przyporządkowane do poszczególnych pojazdów.



Rys. 6.31a. Supertrasa
Źródło: opracowanie własne.



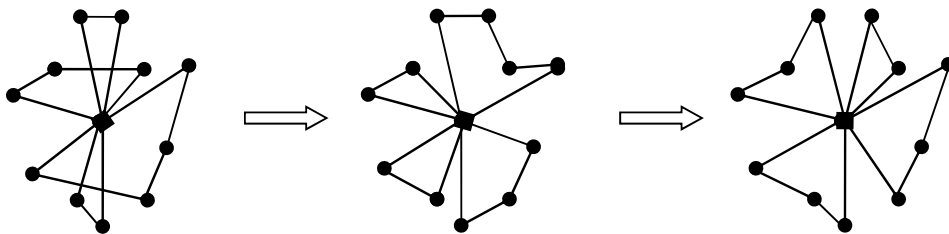
Rys. 6.31b. Przykład podziału supertrasy na poszczególne rejony (trasy pojazdów)
Źródło: opracowanie własne.

Poszukiwanie rozwiązania optymalnego zagadnienia układania tras pojazdów z zastosowaniem strategii „najpierw trasa, potem rejon” ma kilka zalet (Beasley, 1983). Po pierwsze, takie podejście zapewnia, że punkty obsługi znajdujące się blisko siebie będą blisko w poszczególnych trasach pojazdów. Po drugie, generowanie tras poszczególnych pojazdów przy zachowaniu wcześniej ustalonego uporządkowania punktów obsługi jest szybką i nieskomplikowaną procedurą. Po trzecie, pierwszy etap metody (kolejność obsługi) może być przeprowadzony z wykorzystaniem różnych algorytmów dla problemu jednego komiwojażera. Ponadto drugi etap (rejon obsługi) może także polegać na wygenerowaniu jednego dużego zbioru dopuszczalnych tras pojazdów (oczywiście przy zachowaniu wcześniej ustalonej kolejności obsługi), a następnie wyborze spośród nich podzbioru najlepszego.

Rozdział 7

Metaheurystyki układania tras pojazdów

Osobną szeroką klasę algorytmów heurystycznych dla zagadnień układania tras jednego i wielu komiwojażerów stanowią metody wzrostu, nazywane także metodami poprawy (*improving heuristics*). Ogólna idea tej grupy metod opiera się na poszukiwaniu rozwiązania coraz lepszego przez sukcesywne przekształcanie aktualnie rozpatrywanego rozwiązania w kolejne (rys. 7.1).



Rys. 7.1. Poszukiwanie najlepszego rozwiązania na podstawie strategii wzrostu (poprawy rozwiązań)

Źródło: opracowanie własne.

Sposoby poprawy istniejących rozwiązań mogą być bardzo różne. Mogą dotyczyć tylko jednej trasy (dla problemu komiwojażera), jak również wielu tras jednocześnie (w problemie wielu komiwojażerów). Wprowadzane w ich efekcie zmiany mogą być niewielkie bądź też uzyskane rozwiązania w niczym nie będą przypominały wyjściowych.

Jako że przestrzeń rozwiązań zagadnień układania tras jednego i wielu pojazdów jest bardzo duża, istnieje także konieczność tworzenia pewnych reguł jej eksploracji. Reguły te mają różnorodny charakter, począwszy od bardzo prostych, a skończywszy na nieco bardziej skomplikowanych, nierzadko wzorujących się na regułach funkcjonujących w środowisku naturalnym. Stąd też osobne miejsca w piśmiennictwie z zakresu optymalizacji tras pojazdów zyskały sobie metody, które realizują przyjęte reguły eksploracji przestrzeni rozwiązań,

nazywane metaheurystykami. Uszczegółowienie określonych działań w ramach reguł, którymi kieruje się metaheurystyka, sprawia, że staje się ona konkretnym algorytmem heurystycznym przeznaczonym do rozwiązania postawionego problemu decyzyjnego.

Wśród metaheurystyk wykorzystywanych do zagadnień układania tras pojazdów wyróżnić można dwie podstawowe grupy:

- 1) jednoagentowe metody przeszukiwania lokalnego;
- 2) wieloagentowe metody przeszukiwania równoległego.

Pierwsza z nich zawiera algorytmy, które w procesie poszukiwania przestrzeni rozwiązań optymalizowanego problemu w danym momencie dokonują przekształcenia i oceny tylko jednego rozwiązania. Z kolei metody wieloagentowe rozpatrują w danej chwili wiele rozwiązań jednocześnie. W celu znalezienia lepszych tras pojazdów przekształcane są w inne nie pojedyncze trasy, a całe ich zbiory jednocześnie.

7.1. Jednoagentowe przeszukiwanie lokalne

Pierwszą grupę metaheurystyk stanowiących zbiór reguł przeszukiwania przestrzeni rozwiązań dla problemów układania tras pojazdów są jednoagentowe metody przeszukiwania lokalnego. Należą do nich algorytmy, które dokonują przekształcenia jednego rozwiązania w inne. Wśród nich na szczególne wyróżnienie zasługują trzy typy metaheurystyk:

- 1) metody lokalnej optymalizacji;
- 2) metody przeszukiwania tabu;
- 3) metody symulowanego wyżarzania.

Ogólny schemat metod przeszukiwania przestrzeni rozwiązań opartych na klasie algorytmów realizujących strategię przeszukiwania lokalnego został przedstawiony za pomocą zamieszczonego poniżej pseudokodu (rys. 7.2):

```

krok 1: inicjalizacja
repeat
    krok 2: konstrukcja sąsiedztwa
    krok 3: ocena sąsiedztwa
repeat warunek końca
  
```

Rys. 7.2. Algorytm wzrostu

Źródło: opracowanie własne.

W pierwszym kroku algorytmów realizujących strategię poszukiwania coraz lepszego rozwiązania należy skonstruować rozwiązanie początkowe. Może to być zarówno rozwiązanie wygenerowane w sposób losowy, jak i rozwiązanie utworzone z wykorzystaniem jednej z wcześniej prezentowanych metod (np. algorytmu konstrukcyjnego lub dekompozycyjnego). Utworzone rozwiązanie jest rozwiązaniem aktualnie rozpatrywanym, a jednocześnie dotychczas najlepszym znalezionym. Istotnym elementem algorytmów przeszukiwania lokalnego jest sposób realizacji kroku drugiego. Według założonego schematu dokonywane jest w nim przekształcenie aktualnie rozpatrywanego rozwiązania w inne sąsiednie, które w kroku trzecim poddawane jest ocenie, czyli porównaniu z dotychczas najlepszym znalezionym rozwiązaniem. W danej iteracji algorytmu może być generowane jedno rozwiązanie sąsiednie albo pewien ich zbiór. Jeżeli tworzonych jest więcej rozwiązań niż jedno rozwiązanie sąsiednie, spośród nich wybierane jest najlepsze i porównywane z aktualnie rozpatrywanym. W wypadku znalezienia rozwiązania lepszego należy je uznać za dotychczas najlepsze znalezione rozwiązanie i rozpocząć od nowa cały proces poszukiwania lepszego rozwiązania. W przeciwnym razie trzeba skonstruować kolejne rozwiązanie (lub zbiór rozwiązań sąsiednich) z aktualnie rozpatrywanego. Proces konstrukcji kolejnych rozwiązań powstałych z aktualnie rozpatrywanego może być powtarzany do momentu, aż nie jest możliwe skonstruowanie kolejnego rozwiązania według założonego wcześniej schematu. Innym warunkiem końca działania algorytmu jest także wykonanie określonej liczby iteracji lub brak uzyskanej poprawy w określonym czasie działania algorytmu. Taki sposób ustalenia końca działania algorytmu może być wynikiem bardzo licznego zbioru rozwiązań sąsiednich, wynikającego między innymi z przyjętego sposobu ich tworzenia.

Znalezione przez algorytm przeszukiwania lokalnego dotychczas najlepsze rozwiązanie, przy jednoczesnym wyczerpaniu wszystkich możliwości konstrukcji nowych rozwiązań (lub spełnieniu innego warunku końca działania algorytmu), jest rezultatem końcowym algorytmu wzrostu.

Sposób konstrukcji kolejnych rozwiązań jest jednym z najważniejszych elementów algorytmów przeszukiwania lokalnego, ponieważ w bezpośredni sposób wpływa on zarówno na jakość generowanych rozwiązań końcowych, jak i na czas działania algorytmu.

7.1.1. Algorytmy lokalnej optymalizacji

Algorytmy lokalnej optymalizacji należą do grupy metaheurystyk przeszukiwania lokalnego, a realizują ich strategię przeszukiwania przestrzeni rozwiązań w najbardziej uproszczony sposób. W ich wypadku zazwyczaj nie jest generowane liczne sąsiedztwo rozwiązań w danym momencie, lecz za każdym razem tworzone i rozpatrywane jest jedno rozwiązanie sąsiednie (rys. 7.3).

```

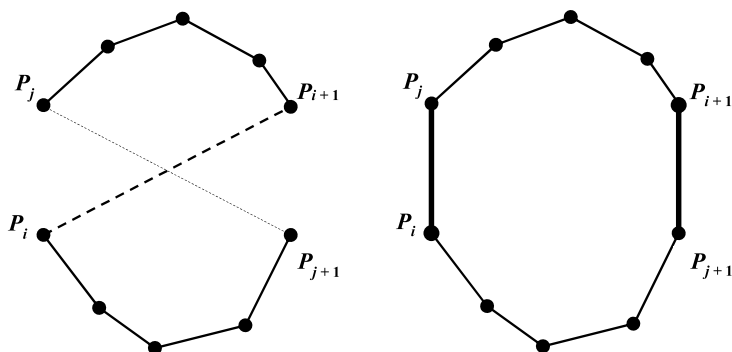
krok 1: inicjalizacja
repeat
  repeat
    krok 2: konstrukcja nowego rozwiązania
    krok 3: ocena rozwiązania
  until brak poprawy rozwiązania
repeat wyczerpane możliwości konstrukcji nowych rozwiązań

```

Rys. 7.3. Algorytm lokalnej optymalizacji

Źródło: opracowanie własne.

Pierwszym, a zarazem najprostszym algorytmem poprawiającym aktualnie rozpatrywane rozwiązanie problemu jednego komiwojażera jest procedura 2-opt (Lin, 1965), która polega na usunięciu dwóch połączeń pomiędzy punktami obsługi, a następnie zastąpieniu ich innymi (rys. 7.4). Geneza tej metody wynika z obserwacji, że w problemach euklidesowych (zdefiniowanych w przestrzeni dwuwymiarowej za pomocą współrzędnych punktów) długość cyklu Hamiltona w grafie zawierającego przecinające się krawędzie jest większa niż długość takiego cyklu bez takich krawędzi.



Rys. 7.4. Wymiana dwóch odcinków tras w algorytmie 2-opt

Źródło: opracowanie własne.

Na rysunku 7.4 przedstawiono dwa cykle Hamiltona, przy czym cykl z prawej strony powstał w wyniku usunięcia połączeń pomiędzy dwoma parami sąsiednich punktów P_i i P_{i+1} oraz P_j i P_{j+1} . Połączenia te następnie zostały zastąpione nowymi odcinkami tras: P_i i P_j oraz P_{j+1} i P_{i+1} . Zadaniem algorytmu 2-opt jest sprawdzenie wszystkich możliwych połączeń par pomiędzy oddzielnymi punktami obsługi w trasie pojazdu, a następnie zastąpienie ich nowymi, które przyniosą korzyści z punktu widzenia długości całej trasy (rys. 7.5). Należy zauważyć, że liczba możliwych sposobów na usunięcie par połączeń i zastąpienie ich nowymi w trasie pojazdu wynosi N^2 .

```

krok 1: inicjalizacja
repeat
    krok 2: ustal trasę  $T$ 
    krok 3: wybierz punkt  $P_i$  trasy  $T$ 
    repeat
        krok 4: wybierz punkt  $P_j$  trasy  $T$ 
        krok 5: konstrukcja i ocena trasy  $T'$ 
    until warunek końca 1
repeat warunek końca 2

```

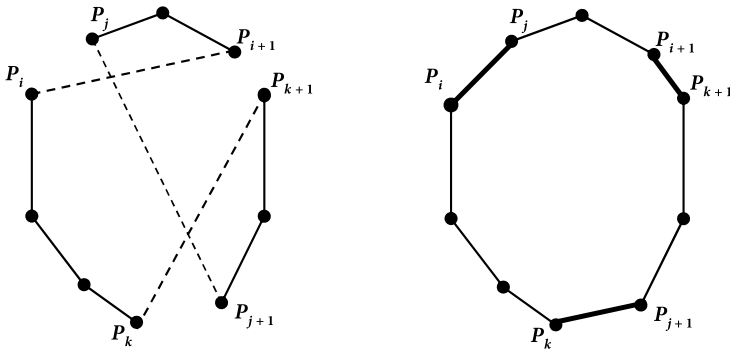
Rys. 7.5. Algorytm 2-opt
Źródło: opracowanie własne.

W kroku pierwszym algorytmu należy wygenerować trasę początkową w sposób losowy lub np. z wykorzystaniem metody konstrukcyjnej. Trasa ta stanowić będzie aktualnie rozpatrywaną trasę pojazdu T . Dla pierwszego punktu P_i trasy T należy wybrać punkt początkowy P_j drugiego połączenia (krok czwarty). Wybrane punkty początkowe dwóch połączeń w trasie T determinują odcinki trasy (P_i, P_{i+1}) oraz (P_j, P_{j+1}) , które zostaną zastąpione nowymi odcinkami (P_i, P_j) oraz (P_{j+1}, P_{i+1}) , tworząc w kroku piątym trasę T' . Trasa ta zostaje poddana ocenie przez policzenie różnicy ΔT pomiędzy długościami tras T i T' (7.1).

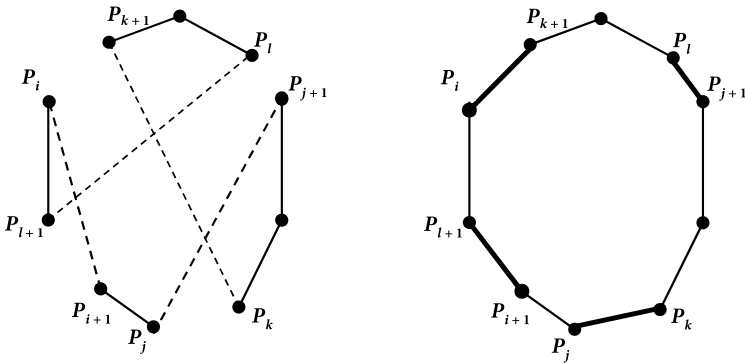
$$\Delta T = d(P_i, P_{i+1}) + d(P_j, P_{j+1}) - d(P_i, P_j) - d(P_{i+1}, P_{i+1}) \quad (7.1)$$

Dodatnia wartość różnicy ΔT wskazuje na poprawę uzyskanego rozwiązania w postaci trasy T' . Stanowi to podstawę do zastąpienia aktualnie rozpatrywanej trasy T trasą T' . W przeciwnym razie należy ponownie przejść do kroku czwartego i wybrać kolejny punkt początkowy P_j drugiego odcinka trasy i zbudować nową trasę T' . Kroki czwarty i piąty algorytmu przedstawionego na rysunku 7.5 powtarzane są do momentu, kiedy spełniony jest pierwszy warunek końca. Jest on spełniony w dwóch przypadkach. Pierwszy dotyczy sytuacji, gdy utworzona nowa trasa T' jest lepsza od aktualnie rozpatrywanej – wtedy, przechodząc ponownie do kroku drugiego algorytmu, należy zastąpić aktualnie rozpatrywaną trasę T trasą T' . Natomiast w razie braku poprawy rozwiązania ponowne przejście do kroku drugiego warunkowane jest wyczerpaniem wszystkich możliwości budowy nowych tras T' przy ustalonym pierwszym odcinku (P_i, P_{i+1}) trasy T .

Brak poprawy rozwiązania, a także rozpatrzenie wszystkich możliwości konstrukcji tras sąsiednich T' dla każdej pary pierwszego odcinka aktualnie rozpatrywanej trasy (P_i, P_{i+1}) stanowi podstawę do zakończenia działania algorytmu 2-opt. Algorytm – ze względu na swoją prostotę i szybkość działania – może być uruchamiany wielokrotnie, rozpoczynając działanie za każdym razem od trasy początkowej wygenerowanej z wykorzystaniem innej metody heurystycznej.



Rys. 7.6a. Wymiana trzech odcinków tras w algorytmie k -opt dla $k = 3$
Źródło: opracowanie własne.



Rys. 7.6b. Wymiana trzech odcinków tras w algorytmie k -opt dla $k = 4$
Źródło: opracowanie własne.

Rozwinięciem algorytmu 2-opt pozwalającym na sprawdzenie większej liczby tras przestrzeni rozwiązań jest algorytm k -opt. Jego schemat działania jest w ogólnym zarysie zbieżny z procedurą przedstawioną dla metody 2-opt (rys. 7.5). Różnica polega na zastąpieniu k połączeń pomiędzy odrębnymi parami punktów w trasie trzema nowymi odcinkami tras, co może zostać wykonane na N^k sposobów (rys. 7.6a i 7.6b). Rozwiązanie uzyskane w ten sposób nazywane jest rozwiązaniem k -optymalnym. Oznacza to, że nie jest możliwe uzyskanie lepszej trasy w wyniku zastąpienia jej inną, a powstałą także na drodze wymiany k jej odcinków. Ponadto każda trasa k -optymalna jest także k' -optymalna dla każdego $1 \leq k' \leq k$. Prowadzi to do stwierdzenia, że każda trasa składająca się z N punktów obsługi jest optymalna wtedy i tylko wtedy, jeżeli jest N -optymalna. Wniosek, jaki można z tego wyprowadzić, jest następujący: im większa wartość k , tym większe jest prawdopodobieństwo, że uzyskane rozwiązanie jest optymalne. Jednocześnie należy pamiętać, że liczba operacji wymiany k połączeń

w rozpatrywanych trasach rośnie bardzo szybko, przez co dla dużych wartości k algorytm ten staje się nieefektywny. Najczęściej przyjmuje się wartość parametru k wynoszącą 2 lub 3, choć w literaturze przedmiotu przyjmowane były także wartości 4 i 5 (Christofides i Eilon, 1972).

Wadą algorytmów k -opt jest konieczność ustalenia na początku wartości parametru k , który będzie określał liczbę wymienianych odcinków trasy w każdej jego iteracji. Wartość parametru k należy traktować tutaj jako pewnego rodzaju kompromis pomiędzy czasem poszukiwania najlepszego rozwiązania a jego jakością. Stąd też interesującą propozycją pomijającą tę niedogodność jest algorytm Lina-Kernighana (Lin i Kernighan, 1973), który okazał się na tyle efektywny, że doczekał się wielu modyfikacji (Mak i Morton, 1993; Helsgaun, 2000; 2009).

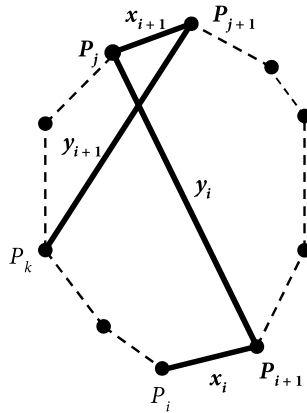
```

krok 1: inicjalizacja – budowa trasy  $T$ 
repeat
    repeat
        krok 2: wybrać odcinki  $x_i$  oraz  $y_i$ 
    until warunek końca
    krok 3: wymiana wybranych odcinków trasy
until brak poprawy rozwiązania
  
```

Rys. 7.7. Algorytm Lina-Kernighana
Źródło: opracowanie własne.

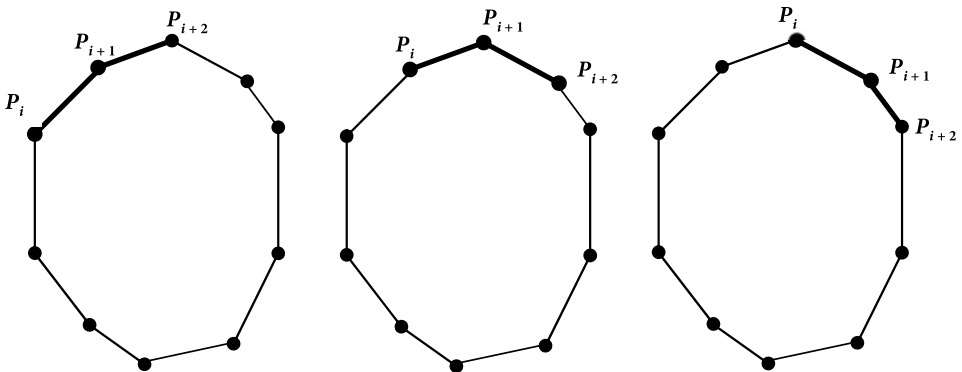
A algorytmie Lina-Kernighana wartość parametru k wyznaczającego liczbę odcinków w trasie, jaką należy wymienić, aby otrzymać nową trasę, ustalana jest w każdej iteracji oddzielnie. Począwszy od $k = 2$, rozpatrywane są kolejne wartości tego parametru i sprawdzane są pewne kryteria w celu ustalenia, jaką liczbę odcinków trasy w danej iteracji należy wymienić. Ogólna postać tego algorytmu została przedstawiona na rysunku 7.7. Sercem algorytmu jest krok drugi, w którym następuje budowa dwóch zbiorów odcinków trasy: odcinków wymienianych x_i (należących do aktualnie rozpatrywanej trasy T) oraz odcinków y_i , które je zastąpią (nienależących do aktualnie rozpatrywanej trasy T). Za każdym razem wybierana jest para odcinków x_i oraz y_i , w wyniku wymiany których otrzymana zostanie wartość g_i oznaczająca korzyść (zysk z zastąpienia odcinka x_i odcinkiem y_i). Sumując korzyści g_i z wymian kolejnych par odcinków (x_i, y_i) dla $i = 1, \dots, k$, proces ich poszukiwania jest kontynuowany do momentu,

aż $\sum_{i=1}^{k+1} g_i \leq 0$. Aby w wyniku wymiany k odcinków x_i na k odcinków y_i można było otrzymać nową dopuszczalną trasę pojazdu, obydwa zbiory krawędzi muszą spełniać kilka warunków. Pierwszy z nich mówi, że odcinki trasy x_i oraz y_i , a także y_i oraz x_{i+1} muszą mieć wspólny punkt obsługi (rys. 7.8). Tym samym pary odcinków: (x_i, y_i) , (x_{i+1}, y_{i+1}) , \dots , (x_{i+r}, y_{i+r}) stanowią ciąg kolejnych dołączanych odcinków podlegających wymianie.



Rys. 7.8. Pierwszy warunek konstrukcji dopuszczalnej trasy w algorytmie Lina-Kernighana
Źródło: opracowanie własne.

Warunkiem koniecznym (aczkolwiek nie wystarczającym), aby nowa trasa była dopuszczalna (aby była cyklem zamkniętym), jest wymóg, by odcinek y_{i+r} miał wspólny punkt obsługi z odcinkiem y_i . Dlatego też następnym warunkiem, jaki musi być spełniony, jest odpowiedni wybór kolejnego odcinka trasy x_r . Musi on być tak wybrany, aby po połączeniu jego końca z początkiem odcinka x_1 (przez wskazanie odpowiadającego mu odcinka y_i) utworzony został cykl zamknięty. Trzecim kryterium wyboru par odcinków podlegających wymianie w algorytmie Lina-Kernighana jest spełnienie warunku, aby całkowita korzyść G , liczona jako suma korzyści g_i z wymiany wszystkich wskazanych dotychczas par odcinków, była dodatnia. Oznacza, to że nie wszystkie wartości g_i muszą być dodatnie, lecz ich suma tak. Wreszcie ostatnim kryterium wyboru par odcinków (x_r, y_i) jest warunek rozłączności zbiorów wszystkich wybranych odcinków x_i oraz y_i .



Rys. 7.9. Zmiana położenia trzech punktów w trasie w algorytmie Or-opt dla $s = 3$
Źródło: opracowanie własne.

Innym przykładem algorytmu lokalnej optymalizacji jest algorytm Or-opt (Or, 1976) polegający na sekwencyjnej wymianie położenia s kolejnych punktów obsługi w trasie i sukcesywnym umieszczaniu ich w innych miejscach (rys. 7.9). Po wyczerpaniu wszystkich możliwości wartość s jest zmniejszana o 1, a procedura zmiany położenia tym razem dwóch kolejnych punktów obsługi w rozpatrywanej trasie pojazdu jest powtarzana w analogiczny sposób. Za każdym razem, gdy tylko uda się znaleźć lepsze rozwiązanie, rozwiązanie aktualnie rozpatrywane jest natychmiast zastępowane przez nowe, a cały proces powtarzany jest od początku (rys. 7.10).

```

krok 1: inicjalizacja  $s = 3, t = 1$ 
repeat
    repeat
        repeat
            krok 2: usuń z trasy ciąg  $s$  punktów
            krok 3: wstaw ciąg  $s$  punktów w inne miejsce
            krok 4: ocena nowej trasy
        until wszystkie miejsca rozpatrzone
        krok 5:  $t = t + 1$ 
    until  $t = N + 1$ 
    krok 6:  $t = 1, s = s - 1$ 
until  $s = 0$ 

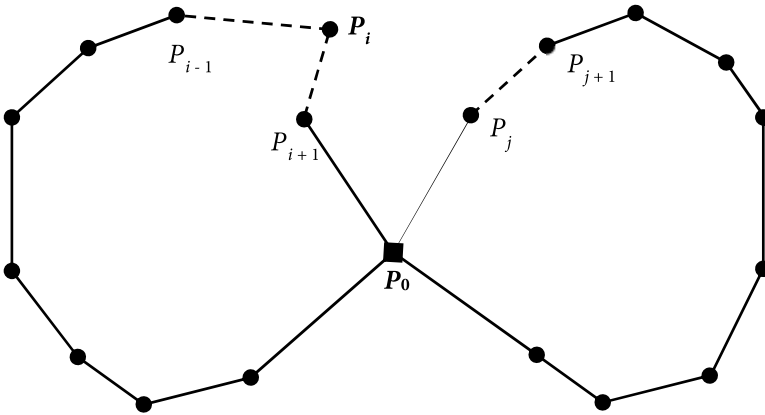
```

Rys. 7.10. Algorytm Or-opt
Źródło: opracowanie własne.

Jakkolwiek przedstawione wyżej algorytmy lokalnej optymalizacji mogą być stosowane także do poszukiwania rozwiązania optymalnego dla zagadnienia układania tras wielu pojazdów, rozpatrują one za każdym razem wyłącznie jedną wybraną trasę pojazdu. Jednakże w problemie wielu komiwojazerów sformułowane są dwa cele optymalizacyjne: optymalna kolejność odwiedzania poszczególnych punktów obsługi przez pojazdy oraz optymalne ich przyporządkowanie do każdego pojazdu. Stąd też zaproponowano algorytmy lokalnego przeszukiwania przestrzeni rozwiązań, które starają się sprostać tym oczekiwaniom, dokonując modyfikacji za każdym razem nie jednej, tylko kilku wybranych tras pojazdów jednocześnie.

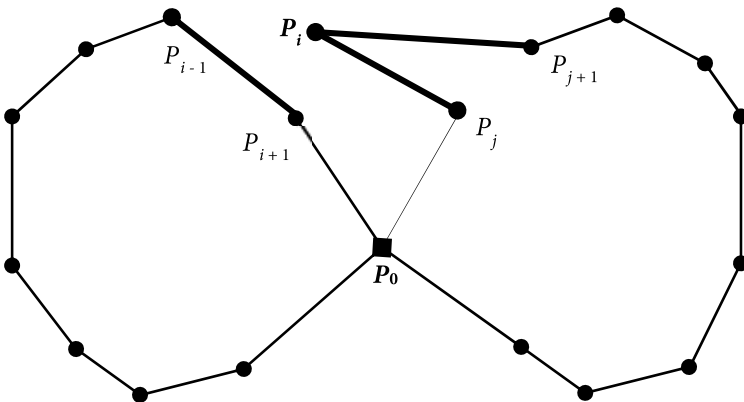
Przykładem algorytmów lokalnej optymalizacji dokonujących jednocześnie modyfikacji dwóch tras pojazdów są sposoby generowania rozwiązań sąsiednich przedstawione przez G.A.P. Kindervatera i M.W.P. Savelsbergha (1997), a polegające na wymianie istniejących krawędzi w rozpatrywanym rozwiązaniu na inne, przy jednoczesnym zachowaniu dopuszczalności rozwiązania (spełnieniu warunków ograniczających dotyczących np. ładowności pojazdu lub czasu ich pracy). Pierwszą z propozycji odnośnie sposobu tworzenia nowego rozwiązania jest relokacja punktu

obsługi z jednej trasy do drugiej (rys. 7.11a i 7.11b). W aktualnie rozpatrywanym rozwiązaniu trzy krawędzie: $(i - 1, i)$, $(i, i + 1)$ oraz $(j, j + 1)$ są zmieniane w nowym sąsiednim rozwiązaniu na inne trzy krawędzie: (j, i) , $(i, j + 1)$ oraz $(i - 1, i + 1)$. Należy zauważyć, że w tym wypadku relokowany punkt obsługi umieszczany jest w drugiej trasie pomiędzy dwoma innymi, ale sąsiednimi punktami.



Rys. 7.11a. Układ dwóch tras przed relokacją punktu obsługi

Źródło: opracowanie własne.

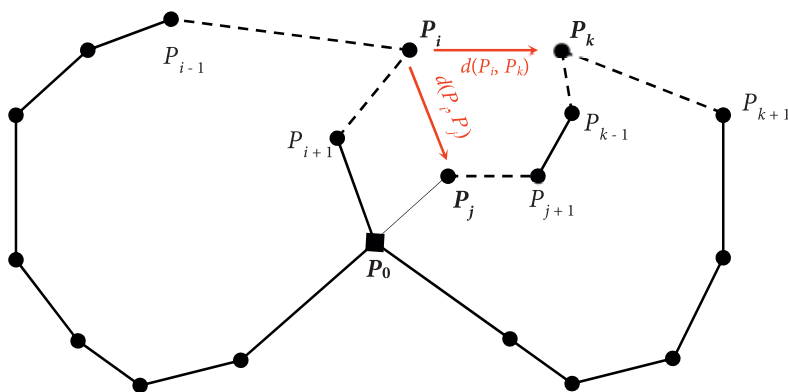


Rys. 7.11b. Układ dwóch tras po relokacji punktu obsługi

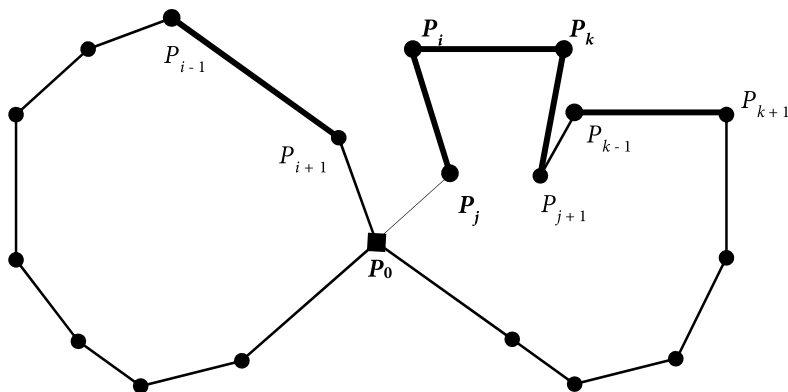
Źródło: opracowanie własne.

Szczególną odmianą relokacji punktu obsługi z trasy jednego pojazdu do innego jest umieszczenie go w nowej trasie, lecz nie pomiędzy dwoma sąsiednimi punktami nowej trasy (Gendreau i in., 1992). Wyznacznikiem tego, pomiędzy którymi punktami nowej trasy powinien znaleźć się przenoszony punkt obsługi, jest ich odległość od rozpatrywanego punktu obsługi. Innymi słowy, rozpatrywany punkt obsługi umieszczany jest w nowej trasie bezpośrednio pomiędzy dwoma

najbliższymi mu punktami, jednocześnie należącymi do tej trasy. Oznacza to, że wybrane w ten sposób punkty nie muszą wcale być w tej trasie punktami sąsiednimi. Ponadto tworząc nową trasę po relokacji, dokonuje się w niej znacznie więcej zmian niż w wypadku poprzednio omówionego sposobu (rys. 7.12a i 7.12b). W pierwszej trasie tak samo kasowane są połączenia (P_{i-1}, P_i) , (P_j, P_{j+1}) , natomiast w drugiej usuwane są: (P_j, P_{j+1}) oraz (P_{k-1}, P_k) i (P_k, P_{k+1}) . W ich miejsce pojawia się także pięć nowych połączeń, jedno (P_{i-1}, P_i) w pierwszej trasie oraz cztery: (P_j, P_i) , (P_j, P_k) , (P_k, P_{j+1}) i (P_{k-1}, P_{k+1}) w trasie drugiego pojazdu.



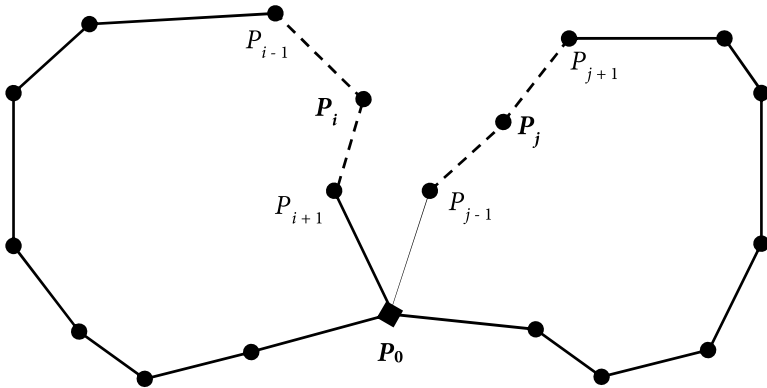
Rys. 7.12a. Układ dwóch tras przed relokacją punktu obsługi
Źródło: opracowanie własne.



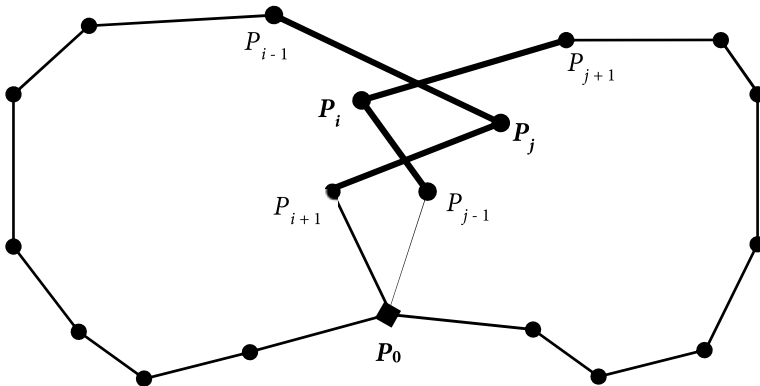
Rys. 7.12b. Układ dwóch tras po relokacji punktu obsługi pomiędzy najbliższymi punktami (nie sąsiednimi)
Źródło: opracowanie własne.

Kolejnym sposobem utworzenia nowego rozwiązania sąsiedniego dla problemu układania tras wielu pojazdów w algorytmie przeszukiwania lokalnego jest

dokonanie relokacji nie jednego, a dwóch punktów obsługi pomiędzy trasami. Relokacja ta ma charakter wzajemnej wymiany dwóch wybranych punktów pomiędzy wskazanymi trasami (po jednym z każdej z nich). W takim wypadku dwie krawędzie (połączenia punktów) w każdej trasie są zastępowane innymi dwoma krawędziami, co zostało pokazane na rysunkach 7.13a i 7.13b. Para krawędzi (P_{i-1}, P_i) , (P_i, P_{i+1}) w pierwszej trasie została zastąpiona parą (P_{i-1}, P_j) , (P_j, P_{i+1}) i analogicznie para połączeń (P_{j-1}, P_j) , (P_j, P_{j+1}) zostaje zastąpiona parą krawędzi (P_{j-1}, P_i) , (P_i, P_{j+1}) w drugiej trasie.



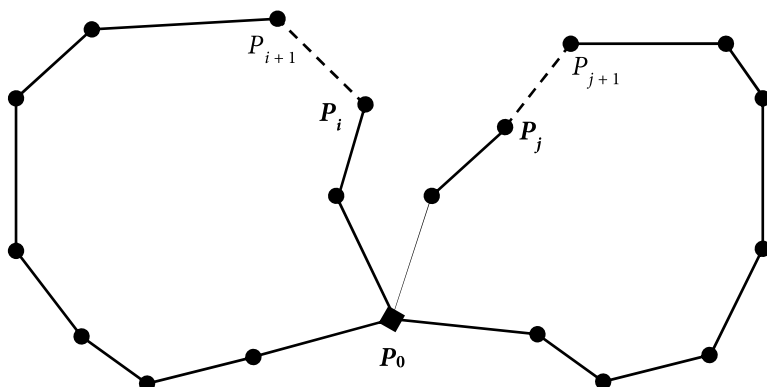
Rys. 7.13a. Układ dwóch tras przed wymianą punktów obsługi
Źródło: opracowanie własne.



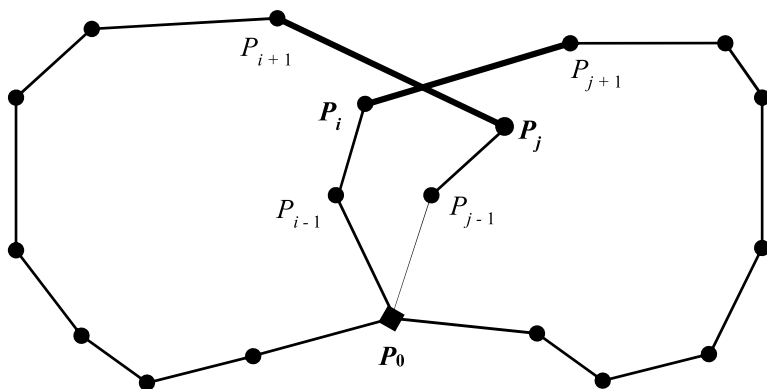
Rys. 7.13b. Układ dwóch tras po wymianie punktów obsługi
Źródło: opracowanie własne.

Wreszcie ostatnią metodą pozwalającą uzyskać rozwiązanie sąsiednie dla problemu wielu komiwojazerów z zastosowaniem algorytmów przeszukiwania lokalnego jest metoda krzyżowania. W metodzie tej część trasy jednego pojazdu jest łączona z fragmentem trasy drugiego pojazdu. Analogicznie pozostałe

fragmenty tras pojazdów tworzą drugą osobną trasę (rys. 7.14a i 7.14b). Para połączeń pomiędzy rozpatrywanymi punktami obsługi i ich następnikami: (P_i, P_{i+1}) oraz (P_j, P_{j+1}) jest zastępowana nową parą (P_i, P_{j+1}) oraz (P_j, P_{i+1}) . Pozostałe krawędzie w trasach pozostają bez zmian, z tą różnicą, że połączenia w wyjściowych trasach następujące po punktach – odpowiednio P_i oraz P_j – zmieniają swoją przynależność i zostają przypisane do innego pojazdu.



Rys. 7.14a. Układ dwóch tras przed krzyżowaniem
Źródło: opracowanie własne.

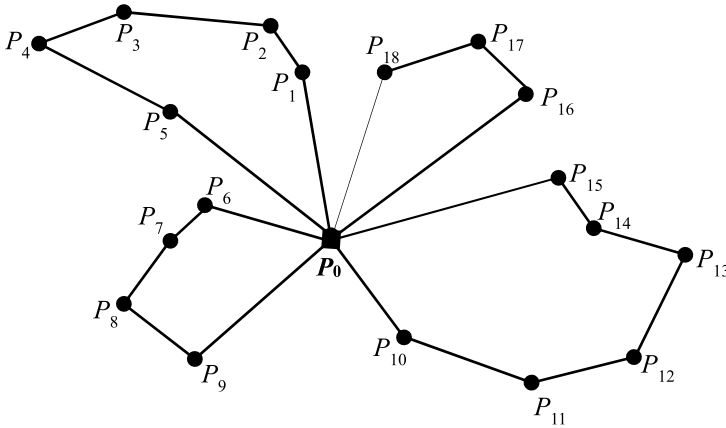


Rys. 7.14b. Układ dwóch tras po krzyżowaniu
Źródło: opracowanie własne.

Nieco bardziej skomplikowany sposób generowania nowych rozwiązań problemów układania tras wielu pojazdów, które mogą stanowić zbiór rozwiązań sąsiednich poddawany ocenie w algorytmach przeszukiwania lokalnego, nosi nazwę transferów cyklicznych i został zaproponowany przez P.M. Thompsona i H.N. Psaraftisa (1993). Podstawową ideą tej metody jest dokonywanie przemieszczeń (transferów) niewielkiej liczby punktów obsługi pomiędzy trasami w pewien

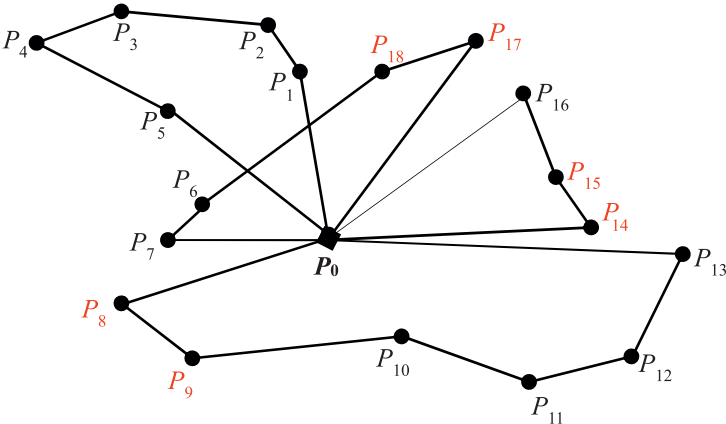
cykliczny sposób. W przeciwieństwie do wcześniej pokazanych sposobów budowy nowych dwóch tras metoda transferów cyklicznych pozwala dokonać zmiany dwóch lub większej liczby tras jednocześnie. Liczba modyfikowanych tras stanowi parametr tej metody.

Przyjmując, że $T = \{T_1, T_2, \dots, T_m\}$ stanowi zbiór m -tras rozwiązań dopuszczalnych problemu układania tras pojazdów, pojęcie transferu cyklicznego można zdefiniować w następujący sposób: niech ρ będzie permutacją cykliczną pewnego podzbioru T' zbioru T (gdzie $T' \subset T$). Oznacza to, że np. dla podzbioru $T' = \{T_4, T_2, T_6\}$ permutacja cykliczna $\rho(T_4, T_2, T_6)$ dokonuje następującego odwzorowania: $\rho(T_4) = T_2, \rho(T_2) = T_6, \rho(T_6) = T_4$. Tym samym jednoczesne przeniesienie punktów obsługi z trasy T_i do trasy $T_{\rho(T)}$ dla każdego i stanowi transfer cykliczny.



Rys. 7.15a. Układ tras przed transferem cyklicznym

Źródło: opracowanie własne.



Rys. 7.15b. Układ tras po przeprowadzeniu transferu cyklicznego dla $b = 3$ i $k = 2$

Źródło: opracowanie własne.

Szczególnymi przypadkami transferów cyklicznych są b -cykliczne k -transfery, gdzie wartości b oraz k stanowią swego rodzaju parametry tej metody. Na przykład 3-cykliczny 2-transfer oznacza cykliczne przemieszczanie pomiędzy trzema trasami dwóch punktów obsługi. Dla czterech przykładowych tras (rys. 7.15a): $T_1 = \{P_1, P_2, P_3, P_4, P_5\}$, $T_2 = \{P_6, P_7, P_8, P_9\}$, $T_3 = \{P_{10}, P_{11}, P_{12}, P_{13}, P_{14}, P_{15}, P_{16}\}$ i $T_4 = \{P_{17}, P_{18}, P_{19}\}$, pokazany został na rysunku 7.15b układ nowych tras jako efekt dokonania transferu cyklicznego $\rho(T_2, T_3, T_4)$ z parametrami $b = 3$ i $k = 2$. Przemieszczeniu pomiędzy trzema trasami ($b = 3$) podlegały zawsze dwa punkty obsługi ($k = 2$). Z trasy T_2 do trasy T_3 przemieszczone zostały punkty P_8 i P_9 , z trasy T_3 do trasy T_4 punkty P_{14} i P_{15} , natomiast z trasy T_4 do T_2 para punktów P_{17} i P_{18} .

Wykonanie procedury transferu cyklicznego powoduje zmianę wartości funkcji celu rozwiązywanego zadania układania tras pojazdów. Tym samym zmiana ta nazywana jest kosztem transferu cyklicznego i jest równa:

$$c(\rho) = \sum_{i=1}^p [f(T_i^\rho) - f(T_i)] \tag{7.2}$$

gdzie:

- $c(\rho)$ – koszt transferu cyklicznego;
- p – liczba tras rozpatrywanego rozwiązania zadania układania tras pojazdów;
- T_i – trasa i -tego pojazdu przed dokonaniem transferu cyklicznego ρ ;
- T_i^ρ – trasa i -tego pojazdu po dokonaniu transferu cyklicznego ρ ;
- $f(T_i)$ – długość (koszt) trasy i -tego pojazdu przed dokonaniem transferu cyklicznego ρ ;
- $f(T_i^\rho)$ – długość (koszt) trasy i -tego pojazdu po dokonaniu transferu cyklicznego ρ .

Na koniec należy dodać, że jeżeli dla całego dostępnego sąsiedztwa aktualnie rozpatrywanego rozwiązania (zbioru tras) T^* zadania układania tras pojazdów wszystkie wartości kosztów transferu cyklicznego są nieujemne, oznacza to, że rozwiązanie to reprezentowane przez zbiór tras T^* jest CT-optymalne (*Cyclic Transfer Optimal*).

7.1.2. Algorytmy symulowanego wyżarzania

Drugą grupę jednoagentowych metaheurystyk przeszukiwania przestrzeni rozwiązań stanowią algorytmy symulowanego wyżarzania (*simulated annealing*). W swoim ogólnym zarysie realizują one przedstawioną wcześniej strategię reprezentowaną przez algorytmy wzrostu.

Podstawowa idea algorytmów symulowanego wyżarzania oparta jest na prawach termodynamiki, które mówią, że przy danej wartości temperatury wzrost energii cząsteczki o wartość δE może zostać określony z prawdopodobieństwem obliczonym zgodnie z następującą formułą (7.3):

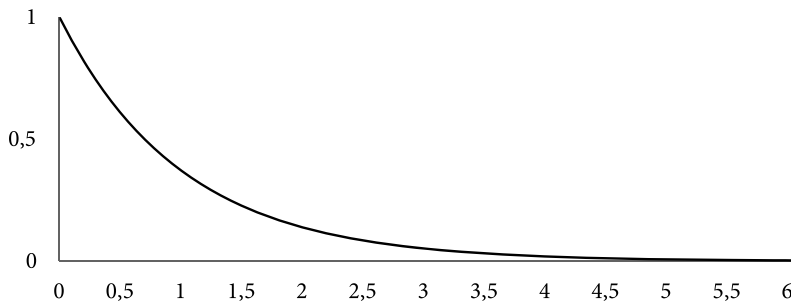
$$p(\delta E) = e^{\frac{-\delta E}{kt}} \quad (7.3)$$

gdzie:

k – stała Boltzmana;

t – aktualna wartość temperatury.

Przy założeniu, że wartość stałej Boltzmana k jest równa 1 oraz wartość temperatury także jest stała i wynosi 1, to zależność (7.3) przyjmuje postać $e^{-\delta E}$, a jej postać graficzna przedstawiona została na wykresie zamieszczonym na rysunku 7.16. Oznacza to, że wraz ze wzrostem zmiany poziomu energii cząsteczki maleje prawdopodobieństwo wzrostu energii cząsteczki.



Rys. 7.16. Wykres zależności prawdopodobieństwa δE od różnicy wartości energii cząsteczki
Źródło: opracowanie własne.

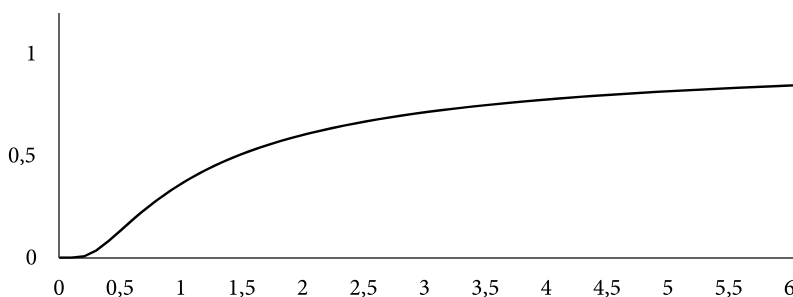
Natomiast sytuacja wygląda odwrotnie w wypadku zależności prawdopodobieństwa wzrostu energii cząsteczki od zadanej temperatury. Tutaj prawdopodobieństwo to wzrasta wraz ze wzrostem wartości temperatury, co zostało pokazane na wykresie zamieszczonym na rysunku 7.17. Podobnie jak poprzednio, przyjmując wartość stałej Boltzmana $k = 1$, a także zmianę wartości poziomu energii

$\delta E = 1$, zależność określona wzorem (7.3) ma postać: $e^{\frac{-1}{t}}$.

Zależności przedstawione powyżej stanowią fundament, który został wykorzystany przez S. Kirkpatricka, C.D. Gelatta i M.P. Vecchiego (1983) do rozwiązywania problemów optymalizacyjnych. Aby zastosować owe prawa termodynamiki do rozwiązywania problemów optymalizacyjnych, konieczne było dokonanie zmian w wymienionych wyżej zależnościach (7.4):

$$p = e^{\frac{-\delta}{T}} \quad (7.4)$$

Przed wszystkim wartość energii cząsteczki zastąpiona została wartością funkcji celu ocenianego rozwiązania. Tym samym zmiana wartości energii cząsteczki odpowiada zmianie wartości funkcji celu w wyniku zmiany rozwiązania zadania optymalizacyjnego. Ponadto usunięto ze wzoru (7.3) stałą Boltzmana, pozostawiono natomiast wielkość T , która dalej nazywana jest temperaturą.



Rys. 7.17. Wykres zależności prawdopodobieństwa ΔE od wartości temperatury
Źródło: opracowanie własne.

Algorytm symulowanego wyżarzania można przyrównać do pieca hutniczego, który stopniowo jest wygaszany w wyniku obniżania jego temperatury. Przy wysokich temperaturach cząsteczki charakteryzują się bardzo dużymi zmianami energii, które stopniowo maleją wraz ze spadkiem temperatury, aż do momentu osiągnięcia stanu równowagi, gdy zmiany energii cząsteczek są niewielkie, a wręcz niezauważalne. W algorytmie symulowanego wyżarzania przeznaczonym do rozwiązywania zadań optymalizacyjnych w pierwszej jego fazie (przy wysokiej temperaturze) przeszukiwane są duże obszary przestrzeni rozwiązań, charakteryzujące się dużymi zmianami wartości funkcji celu rozpatrywanych rozwiązań. Następnie w wyniku obniżania wartości temperatury eksplorowane obszary przestrzeni rozwiązań są coraz mniejsze, by na końcu zawęzić się do jednego. Takie podejście daje większe szanse na uniknięcie zagrożenia, że utknie się w optimum lokalnym, a tym samym zwiększa szanse na znalezienie optimum globalnego rozwiązywanego zadania optymalizacyjnego.

Dla problemów zarówno jednego, jak i wielu komiwojażerów algorytm symulowanego wyżarzania w swojej ogólnej postaci jest bardzo podobny do zwykłego algorytmu lokalnej optymalizacji (rys. 7.18) ze względu na konieczność konstrukcji oraz oceny rozwiązań sąsiednich. Tak jak w algorytmach przeszukiwania lokalnego, także i tu sposób generowania rozwiązań sąsiednich ma fundamentalne znaczenie i dla jakości uzyskiwanych rozwiązań, i dla czasu działania algorytmu. Każde wygenerowane nowe rozwiązanie sąsiednie poddawane jest ocenie (sprawdzenie dopuszczalności rozwiązania oraz określenie wartości funkcji celu). Istotą algorytmu symulowanego wyżarzania stanowi jednak to, że w przeciwieństwie do zwykłego algorytmu lokalnej optymalizacji sąsiednie gorsze rozwiązanie może zostać zaakceptowane, a tym samym może stać się rozwiązaniem, z którego będą generowane nowe rozwiązania sąsiednie.

Cecha ta jest wyrażona w kroku czwartym prezentowanego algorytmu: jeżeli znalezione rozwiązanie sąsiednie jest lepsze od poprzedniego, jest ono akceptowane bez żadnych dodatkowych warunków. Natomiast gdy uzyska się rozwiązanie o gorszej wartości funkcji celu, może zostać ono zaakceptowane, ale z pewnym prawdopodobieństwem p , które oblicza się zgodnie z formułą (7.4). W wypadku

zadań optymalizacji, w których funkcja celu jest minimalizowana (zadania układania tras pojazdów), wartość δ liczona jest jako różnica $f(x') - f(x)$, gdzie x' oznacza rozwiązanie sąsiednie utworzone w wyniku przekształcenia rozwiązania x . Ponadto należy pamiętać, że w praktyce stosowania algorytmów symulowanego wyżarzania, pomimo przejścia do rozwiązania gorszego, rozwiązania dotychczas najlepsze znalezione nie są zapominane. Ze względu na to, że prawdopodobieństwo akceptacji rozwiązania gorszego jest uzależnione od wartości temperatury t , to im ona będzie wyższa, tym większe będą szanse na akceptację gorszego rozwiązania sąsiedniego. Z kolei im różnica pomiędzy gorszym rozwiązaniem sąsiednim a aktualnie rozpatrywanym będzie większa, tym mniejsze będą szanse na akceptację takiego rozwiązania przez algorytm.

```

krok 1: inicjalizacja
repeat
  repeat
    krok 2: konstrukcja i ocena nowego rozwiązania
    krok 3: akceptacja rozwiązania
  until limit prób
  krok 4: obniżenie temperatury
until warunek końca

```

Rys. 7.18. Algorytm symulowanego wyżarzania

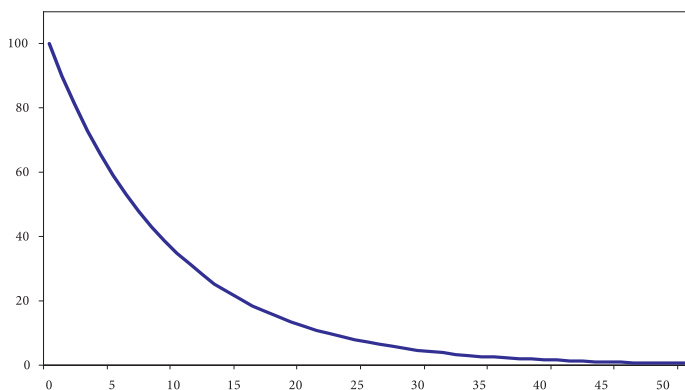
Źródło: opracowanie własne.

Następne bardzo istotne elementy algorytmu to określenie: temperatury początkowej, sposobu jej obniżania oraz temperatury końcowej. Temperatura początkowa i funkcja jej spadku stanowią parametry warunkujące zarówno czas działania algorytmu symulowanego wyżarzania, jak i jego efektywność. Nie można mówić o optymalnej wartości początkowej temperatury, ponieważ jej wartość jest wprost uzależniona od rodzaju oraz wielkości rozwiązywanego problemu układania tras pojazdów. Natomiast wartości temperatury początkowej i końcowej, a także sposób jej obniżania wpływają wprost na liczbę iteracji wykonywanych przez algorytm. Zbyt mała ich liczba skutkować będzie zbyt szybką zbieżnością algorytmu do optimum lokalnego. Zbyt duża zaś liczba analizowanych wartości temperatur w trakcie działania algorytmu może powodować zbyt długi (nieakceptowalny) czas jego działania. Osiągnięcie przez algorytm temperatury końcowej stanowi warunek końca jego działania.

Osobny punkt rozważań przy konstrukcji algorytmów symulowanego wyżarzania stanowi sposób obniżania temperatury początkowej, aż do osiągnięcia jej wartości końcowej. Najczęstszym spotykanym sposobem „chłodzenia” jest korzystanie z zależności:

$$T_i = \alpha T_{i-1} \quad (7.5)$$

gdzie α jest współczynnikiem przyjmującym wartości bliskie jedności, najczęściej z przedziału $(0,8; 0,99)$. Na wykresie zamieszczonym na rysunku 7.19 zilustrowano przebieg tej funkcji, gdzie oś pionowa oznacza wartość temperatury T_i , a oś pozioma reprezentuje kolejne wartości indeksu i .

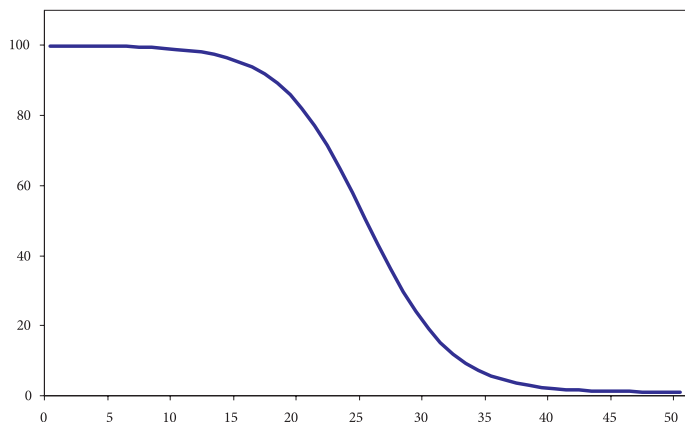


Rys. 7.19. Wykres spadku temperatury dla zależności określonej wzorem (7.5)

Źródło: opracowanie własne.

Nie są to wszakże jedyne schematy chłodzenia, jakie można spotkać w literaturze przedmiotu. Poniżej przedstawione zostały wzory dotyczące dwóch innych sposobów oraz odpowiadające im wykresy (Trojanowski, 2005, s. 53–54):

$$T_i = \frac{T_0 - N}{1 + e^{0,3 \left(i - \frac{N}{2} \right)}} \quad (7.6)$$

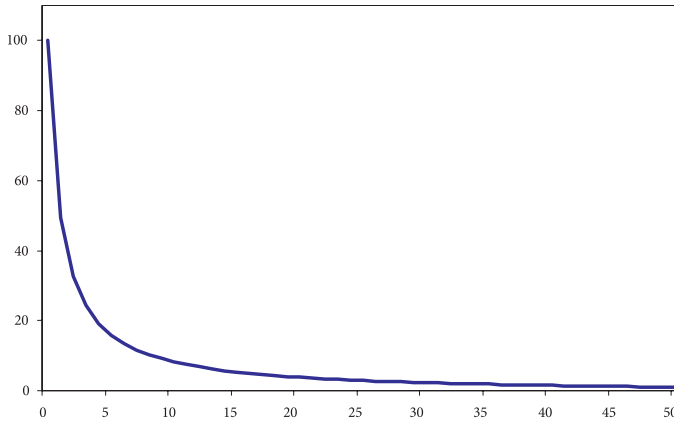


Rys. 7.20. Wykres spadku temperatury dla zależności określonej wzorem (7.6)

Źródło: opracowanie własne.

$$T_i = \frac{A}{i+1} + T_0 - A \quad (7.7)$$

$$A = \frac{(T_0 - T_N)(N+1)}{N} \quad (7.8)$$



Rys. 7.21. Wykres spadku temperatury dla zależności określonej wzorami (7.7) i (7.8)
Źródło: opracowanie własne.

W pierwszym ze schematów chłodzenia (rys. 7.20) przestrzeń rozwiązań jest przeszukiwana dosyć obszernie (częsta akceptacja rozwiązań gorszych), co ma na celu znalezienie najbardziej obiecującego rejonu, aby go w końcowej fazie przeszukać. Z kolei w schemacie pokazanym na rysunku 7.21 algorytm dosyć szybko skupia się na wybranym rejonie przestrzeni rozwiązań zadania, aby dokonać jak najdokładniejszego jego przeszukania.

Ostatnim parametrem, na jaki należy jeszcze zwrócić uwagę, jest limit prób, który stanowi warunek końca procesu generowania sąsiednich rozwiązań w każdej iteracji algorytmu. Jest to liczba wygenerowanych rozwiązań sąsiednich przy danej temperaturze. Liczba ta powinna zostać także w odpowiedni sposób dobrana, a jednym z kryteriów określenia jej wartości może być liczebność zbioru potencjalnego sąsiedztwa dla aktualnie rozpatrywanego rozwiązania. Warto zauważyć, że zarówno limit prób, wartość temperatury początkowej i końcowej, jak i schemat chłodzenia stanowią jednocześnie o czasie działania algorytmu. Iloczyn liczby prób przy danej temperaturze i liczby określonych w trakcie działania algorytmu poziomów temperatur stanowi liczbę wygenerowanych i ocenionych rozwiązań problemu optymalizacyjnego.

Dla zagadnień układania tras jednego i wielu komiwojażerów zaproponowano wiele algorytmów, które należą do rodziny metaheurystyk symulowanego

wyżarzania. Różnice pomiędzy nimi najczęściej dotyczyły kilku aspektów. Jednym z nich jest sposób generowania rozwiązań sąsiednich, gdzie do ich tworzenia mogą być wprost implementowane metody lokalnej optymalizacji omówione w poprzednim podrozdziale. Drugim z elementów wyróżniających dany algorytm od innych jest dobór i sposób ustalania wartości odpowiednich parametrów charakterystycznych dla algorytmów symulowanego wyżarzania (temperatury początkowej i końcowej, schematu chłodzenia, warunków stopu algorytmu). Wreszcie algorytmy symulowanego wyżarzania stanowią też nierzadko element większej całości będącej pewnego rodzaju hybrydą różnych metod, a służący do optymalizacji lub przynajmniej poprawy np. fragmentu uzyskanego wcześniej rozwiązania.

Jedną z pierwszych implementacji algorytmu symulowanego wyżarzania dla problemu układania tras pojazdów był algorytm przedstawiony przez I.H. Osmana (1993). W algorytmie tym rozwiązanie początkowe generowane jest jedną z konstrukcyjnych metod heurystycznych, by w dalszym etapie generować kolejne rozwiązania sąsiednie i je akceptować lub odrzucać zgodnie z przedstawionymi wcześniej zasadami. Na szczególną uwagę zasługuje tutaj schemat chłodzenia, który ma postać (7.9)–(7.10):

$$T_k = \frac{T_k}{(1 + \beta_k T_k)} \quad (7.9)$$

$$\beta_k = \frac{T_s - T_f}{(\alpha + \gamma \sqrt{k}) T_s T_f} \quad (7.10)$$

gdzie:

- T_s – temperatura początkowa;
- T_f – temperatura końcowa;
- T_k – temperatura aktualna.

W kroku inicjującym algorytm generowane jest tzw. próbne sąsiedztwo rozwiązania początkowego, które jest oceniane z punktu widzenia minimalnej Δ_{\min} i maksymalnej Δ_{\max} poprawy wartości funkcji celu. Wartości Δ_{\min} oraz Δ_{\max} są przypisane odpowiednio do T_s oraz T_f . Ponadto w sytuacji, gdy algorytm nie znajduje lepszego sąsiedztwa w stosunku do aktualnie rozpatrywanego rozwiązania, realizowany jest okresowy wzrost temperatury (zresetowanie temperatury) do poziomu T_r , gdzie $T_r = \max\{T_r / 2; T_b\}$, gdzie T_b oznacza temperaturę, przy której znaleziono dotychczas najlepsze rozwiązanie. Kryterium stopu algorytmu stanowi ustalona liczba kolejnych wykonanych zresetowań temperatury, przy których nie osiągnięto poprawy dotychczas najlepszego znalezionego rozwiązania.

Przykładem realizacji metaheurystyki symulowanego wyżarzania dla problemu jednego komiwojażera jest także algorytm zaproponowany przez Z. Wanga, X. Genga i Z. Shao (2009). Algorytm symulowanego wyżarzania wykorzystywany jest w nim dwukrotnie. Najpierw w celu wygenerowania zbioru dobrych rozwiązań problemu, aby w drugim etapie, na podstawie tych rozwiązań oraz pewnych

dotychczasowych informacji o przestrzeni rozwiązań, wygenerować już najlepsze możliwe rozwiązania. Postać algorytmu użytego w obydwu etapach jest do siebie bardzo zbliżona i została zilustrowana na rysunku 7.22. W porównaniu do schematu zamieszczonego na rysunku 7.18 można tutaj zauważyć brak pętli algorytmu, w której przy określonej temperaturze generowane są i oceniane nowe rozwiązania sąsiednie (brak limitu prób). Oznacza to, że przy każdej nowej iteracji algorytmu, a tym samym dla każdego ocenianego nowego rozwiązania sąsiedniego generowana jest nowa wartość temperatury.

<p>krok 1: inicjalizacja repeat krok 2: konstrukcja i ocena nowego rozwiązania krok 3: akceptacja rozwiązania lub odrzucenie rozwiązania krok 4: obniżenie temperatury until temperatura końcowa</p>

Rys. 7.22. Algorytm symulowanego wyżarzania Z. Wanga, X. Genga i Z. Shao
Źródło: opracowanie własne.

W kroku pierwszym algorytmu dokonywana jest jego inicjalizacja polegająca na ustawieniu wartości temperatur początkowej i końcowej, a także innych istotnych parametrów, które zostaną omówione w kolejnych krokach. Przyjęto temperaturę początkową równą 200, a końcową równą 0,1. Ze względu na przyjęty przez autorów schemat chłodzenia (krok 4 algorytmu), który jest zgodny ze wzorem (7.5), oraz rozmiary rozwiązywanych zagadnień komiwojażera współczynnik chłodzenia określony został na poziomie od 0,99998 do 0,999998. Tak bliska jedności wartość przyjętego współczynnika chłodzenia pozwala autorom na wykonanie dostatecznie dużej liczby iteracji algorytmu. Krok drugi to konstrukcja nowego rozwiązania sąsiedniego, w którym na podstawie losowo wygenerowanych dwóch punktów istniejącej trasy oraz informacji o odległościach pomiędzy nimi tworzona jest i poddawana ocenie nowa trasa. W kroku tym następuje także zapamiętanie najlepszej dotychczas znalezionej trasy, co jest konieczne, gdyż w kolejnym kroku algorytmu rozwiązanie to mogłoby zostać przez algorytm zgubione.

Istotą algorytmu jest krok czwarty, w którym podejmowana jest decyzja o akceptacji wygenerowanego rozwiązania sąsiedniego lub o jego odrzuceniu. Rozwiązanie sąsiednie jest bezwarunkowo akceptowane, jeżeli jest ono lepsze od poprzedniego. Natomiast w razie uzyskania gorszej trasy sąsiedniej wykorzystywane jest prawdopodobieństwo akceptacji rozwiązania obliczane zgodnie z formułą (7.4). Ponadto autorzy wprowadzają jeszcze dodatkowe dwa parametry algorytmu: g_{score} i s_{score} . Pierwszy z nich, g_{score} , jest swego rodzaju licznikiem mierzącym, ile razy z rzędu uzyskane rozwiązanie sąsiednie było gorsze od poprzedniego. Jeżeli ustalona na etapie inicjalizacji wartość tego parametru zostaje przekroczona, dopiero wtedy liczone jest prawdopodobieństwo akceptacji p oraz z tym prawdopodobieństwem

rozpatrywane rozwiązanie sąsiednie jest akceptowane lub odrzucane (generowana jest liczba losowa r z przedziału $[0, 1]$). Pomimo że np. wylosowana liczba $r \geq p$, nie przesądza to o tym, że trasa sąsiednia zostanie odrzucona, ponieważ wprowadzony został drugi z parametrów s_{score} , który także stanowi licznik, ale liczby odrzuceń z rzędu gorszego rozwiązania sąsiedniego. Jeżeli ustalona wcześniej wartość s_{score} zostanie przekroczona, wtedy bez względu na wynik przeprowadzonego losowania liczby r gorsza trasa sąsiednia jest akceptowana. Wartości g_{score} i s_{score} są ustalane w kroku pierwszym algorytmu i przez autorów przyjęte zostały na poziomie odpowiednio: 8 oraz $3n$ (gdzie n oznacza liczbę punktów obsługi w problemie komiwojażera).

Bliższa klasycznej postaci algorytmu symulowanego wyżarzania (rys. 7.18) jest procedura autorstwa H. Harmananiego, D. Azara, N. Helal i W. Keirouza (2011) dla zagadnień układania tras wielu pojazdów. Oprócz standardowego schematu obniżenia temperatury (7.5) wykonuje ona także określoną liczbę iteracji (prób) generowania rozwiązań sąsiednich przy zadanej wartości temperatury, choć liczba ta w trakcie działania algorytmu podlega zmianom. Wraz z każdym obniżeniem temperatury (zastosowany współczynnik chłodzenia $\alpha = 0,99$) zwiększa się także limit prób M przy zadanej temperaturze zgodnie z formułą $M_i = \beta M_{i-1}$ (gdzie $\beta = 1,05$). Ponadto algorytm zlicza sukcesywnie liczebność wygenerowanych rozwiązań sąsiednich (prób) w trakcie całkowitego czasu swojego działania. Podyktowane jest to tym, że autorzy oprócz określenia wartości końcowej temperatury (równiej 0,001 przy temperaturze początkowej równej 5000) postanowili dodać jednocześnie jeszcze jeden warunek zakończenia działania algorytmu. Jest nim maksymalny czas działania całego algorytmu liczony jako maksymalna liczba wszystkich rozwiązań sprawdzonych przez algorytm.

Sposób generowania sąsiednich zbiorów tras przez algorytm opiera się tutaj na założeniu, że nowe trasy spełniają wszystkie ograniczenia (rozwiązania są zawsze dopuszczalne). Natomiast w celu utworzenia rozwiązania sąsiedniego autorzy stosują dwie procedury. Pierwsza polega na wyznaczeniu pięciu najkrótszych połączeń pomiędzy punktami obsługi w zbiorze wszystkich tras, usunięciu ich z istniejących tras, a następnie ich wstawieniu do pięciu losowo wybranych istniejących tras (pod warunkiem, że ograniczenia dotyczące pracy pojazdu zostaną spełnione). Wstawienie tych punktów do wybranych tras następuje z zachowaniem jak najmniejszego kosztu rozbudowy trasy. Druga z procedur generowania rozwiązań sąsiednich, nazwana procedurą zamiany największej średniej, polega na obliczeniu wartości średniej długości połączeń skojarzonych z każdym punktem obsługi według wzoru:

$$d_i = \frac{d_{i-1,i} + d_{i,i+1}}{2} \quad (7.11)$$

Następnie wskazywanych jest pięć punktów obsługi ze zbioru wszystkich tras, dla których obliczone wartości średnie (7.11) są najmniejsze. W dalszym etapie, podobnie jak

w wypadku pierwszej procedury, punkty te wstawiane są do losowo wybranych tras także w taki sposób, aby koszt przyrostu długości trasy był jak najniższy.

Kolejnym algorytmem symulowanego wyżarzania, którego konstrukcja przeznaczona została do rozwiązania problemu wielu komiwojażerów, jest algorytm przedstawiony przez S. Afifiego, D.-C. Danga i A. Moukrima (2013) (rys. 7.23).

```

krok 1: inicjalizacja
repeat
  repeat
    krok 2: dywersyfikacja 1
    krok 3: konstrukcja i ocena nowego rozwiązania
    krok 4: obniżenie temperatury
  until maksymalna liczba iteracji
  krok 5: zwiększenie temperatury
  krok 6: dywersyfikacja 2
until liczba wywołań algorytmu

```

Rys. 7.23. Algorytm symulowanego wyżarzania S. Afifiego, D.-C. Danga i A. Moukrima
Źródło: opracowanie własne.

W kroku pierwszym algorytmu konstruowane jest rozwiązanie początkowe z wykorzystaniem heurystycznej metody konstrukcyjnej. Dokonuje ona sekwencyjnej budowy tras pojazdów, startując albo od pustego zbioru tras, albo od częściowo wygenerowanego w inny sposób rozwiązania. Wstawianie nowych nieodwiedzonych punktów obsługi do częściowo zbudowanej trasy odbywa się z zastosowaniem zasady najniższego kosztu rozbudowy trasy (np. najniższego przyrostu długości trasy pojazdu).

Ponadto omawiany algorytm wyróżnia się spośród innych kilkoma istotnymi cechami. Po pierwsze, właściwa postać algorytmu zawiera się w krokach 2, 3 oraz 4. Po drugie, w przeciwieństwie do klasycznej postaci algorytmu symulowanego wyżarzania określona jest tutaj pewna maksymalna liczba iteracji, jaka może zostać wykonana przez algorytm. Tym samym to właśnie ta maksymalna liczba iteracji, a nie temperatura końcowa jest warunkiem zakończenia działania algorytmu. Po trzecie, algorytm określony w krokach 2–4 zawiera się w zewnętrznej pętli, co oznacza, że jest on uruchamiany wielokrotnie po ponownym zwiększeniu temperatury początkowej. Wreszcie przedstawiona przez autorów propozycja algorytmu symulowanego wyżarzania zawiera pewien bardzo istotny element określony w krokach 2 i 6, którym jest procedura dywersyfikacji. Jej celem jest umożliwienie algorytmowi dokonania zróżnicowania generowanych stopniowo rozwiązań sąsiednich, co w efekcie ma doprowadzić do unikania zbyt szybkiej zbieżności algorytmu do rozwiązania, które mogłoby się okazać optimum lokalnym. Procedura ta charakteryzuje się dwoma parametrami: d_{\min} i d_{\max} , które oznaczają granice przedziału, z którego losowana jest pewna liczba całkowita. Wylosowana wartość oznacza liczbę losowo wskazanych punktów obsługi, które zostaną usunięte z aktualnie rozpatrywanego rozwiązania, a następnie za pomocą wskazanego w kroku pierwszym algorytmu

heurystycznego ponownie umieszczone w trasach pojazdów, pamiętając przy tym, aby trasy te dalej stanowiły rozwiązanie dopuszczalne.

Procedura dywersyfikacji uruchamiana jest w każdej iteracji właściwego algorytmu symulowanego wyżarzania (krok 2), gdzie $d_{\min.} = 1$ (autorzy nie podają wartości $d_{\max.}$), a także w kroku 6 po ponownym zwiększeniu temperatury w kroku 5. Wartości parametrów procedury dywersyfikacji w kroku 6 wynoszą odpowiednio: $d_{\min.} = N / 2$ oraz $d_{\min.} = N$ (gdzie N stanowi liczbę punktów obsługi). W celu wygenerowania rozwiązań sąsiednich autorzy proponują wykorzystanie omawianych w pierwszym podrozdziale heurystyk wzrostu 2-opt i Or-opt, natomiast przyjęty schemat chłodzenia ma postać wyrażoną wzorem (7.5).

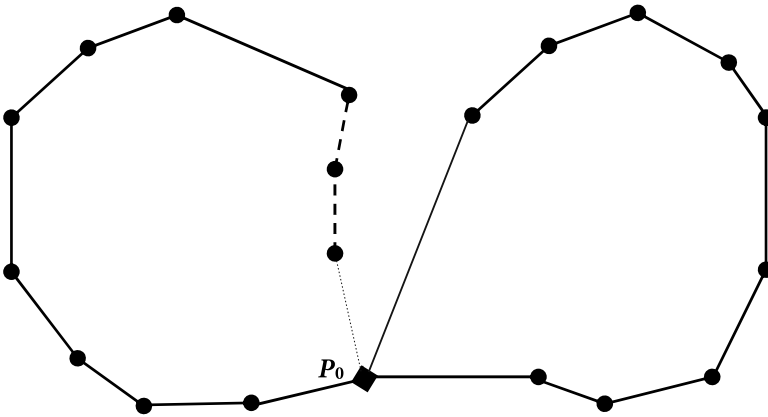
Algorytm symulowanego wyżarzania w swojej klasycznej ogólnej postaci (przedstawionej pseudokodem na rys. 7.18) został zaproponowany także przez A. van Breedama (1995). Podobnie jak we wcześniej prezentowanych algorytmach zastosowano tutaj klasyczny schemat chłodzenia (7.5) przy wartości współczynnika $\alpha = 0,9$. Limit prób generowania rozwiązań sąsiednich przy zadanej wartości temperatury wynosił 1000. Autor nie podaje początkowej wartości temperatury, przy jakiej rozpoczyna się działanie algorytmu, lecz stwierdza jedynie, że jest ona funkcją, której argumentami są współczynnik akceptowalności rozwiązania oraz liczba wyrażająca brak poprawy rozwiązania. Współczynnik akceptowalności rozwiązania definiowany jest jako iloraz liczby zaakceptowanych rozwiązań dopuszczalnych do liczby wszystkich rozwiązań sąsiednich wygenerowanych przez algorytm. Warto tutaj także wspomnieć sugestię autora dotyczącą sposobu ustalania wartości temperatury początkowej. Według niego jej wartość powinna być swego rodzaju kompromisem pomiędzy dostępnością wszystkich rozwiązań dopuszczalnych a czasem działania algorytmu. Duża wartość temperatury początkowej pozwala na większą różnorodność rozpatrywanych rozwiązań sąsiednich (większe prawdopodobieństwo przejścia do rozwiązania gorszego) przy jednoczesnym wydłużeniu czasu działania algorytmu. I odwrotnie, zbyt mała wartość temperatury może powodować zbyt szybką zbieżność algorytmu do optimum lokalnego.

Algorytm kończy działanie przy osiągnięciu ustalonej w kroku inicjalizacji temperatury końcowej. Ponadto w proponowanym algorytmie symulowanego wyżarzania przedstawiono dwa dodatkowe kryteria stopu dla algorytmu. Pierwszym jest osiągnięcie przez algorytm maksymalnej liczby iteracji (ustalonej przez autora na 10 000 000). Natomiast drugim dodatkowym warunkiem końca jest osiągnięcie przez współczynnik akceptowalności wartości mniejszej od zadanego w kroku inicjalizacji progu wynoszącego 0,01.

Na koniec rozważań na temat prezentowanego algorytmu symulowanego wyżarzania należy zwrócić uwagę na proponowany przez A. van Breedama sposób generowania rozwiązań sąsiednich. Przede wszystkim może on bowiem powodować generowanie tras (a tym samym rozwiązań) niedopuszczalnych. Jednakże jeżeli wygenerowane zostanie rozwiązanie niedopuszczalne, nie jest ono brane pod uwagę przez algorytm, ale sam fakt jego utworzenia jest rejestrowany (ma wpływ na wartość współczynnika akceptowalności). Rozwiązanie sąsiednie jest generowane

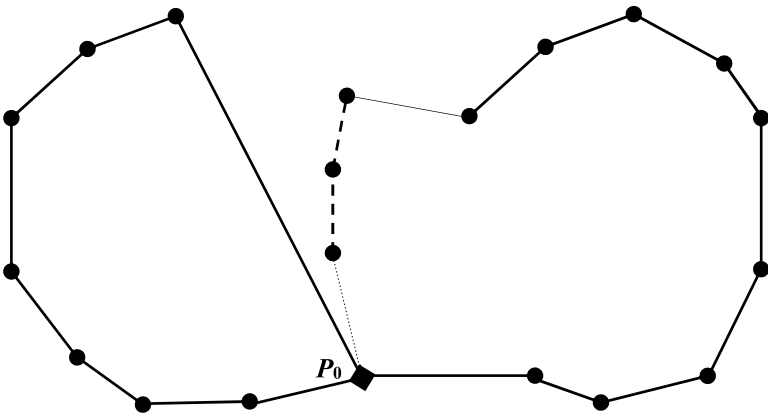
do momentu, aż wszystkie składające się na nie trasy będą dopuszczalne. Ponadto autor proponuje dwie metody generowania nowych rozwiązań, które można uznać za pewne rozszerzenie prezentowanych wcześniej metod relokacji lub wymiany punktu obsługi pomiędzy dwiema trasami pojazdu.

Pierwsza z proponowanych metod nosi nazwę relokacji łańcucha (punktów obsługi) i polega na wymianie pomiędzy dwiema trasami wybranego (np. losowo) podciągu punktów obsługi (rys. 7.24a i 7.24b). Szczególnym przypadkiem tej metody jest wymiana podciągu trasy składającego się z jednego punktu obsługi. Jest to wariant pokazany na rysunkach 7.11a i 7.11b. Natomiast druga z metod to wymiana łańcuchów, która generuje rozwiązanie sąsiednie przez wskazanie dwóch podciągów w każdej z dwóch wybranych tras pojazdów, a następnie wymianę tych podciągów pomiędzy wybranymi trasami (rys. 7.25a i 7.25b).



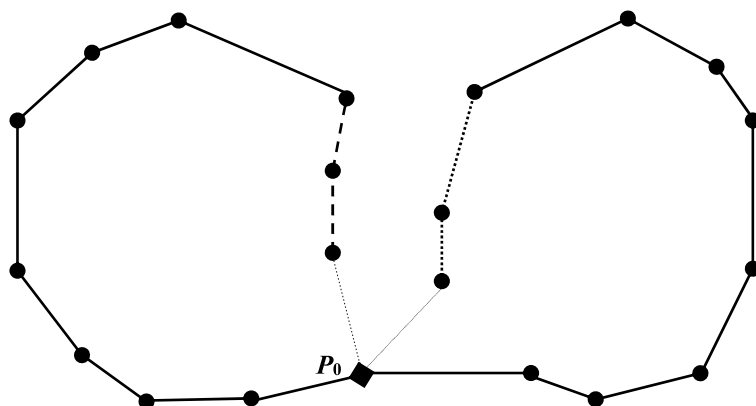
Rys. 7.24a. Układ dwóch tras przed relokacją łańcucha (punktów obsługi)

Źródło: opracowanie własne.

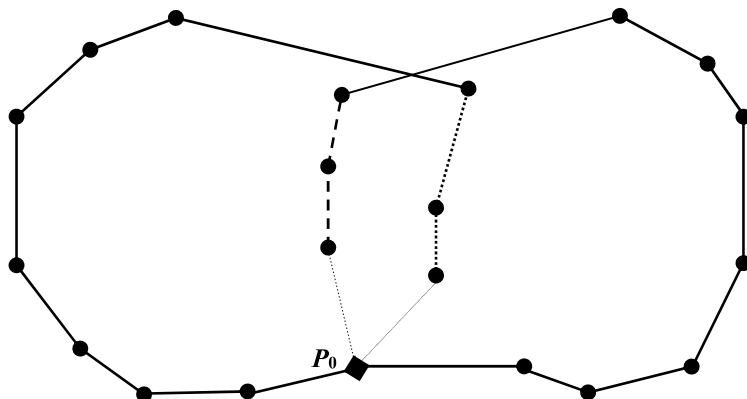


Rys. 7.24b. Układ dwóch tras po relokacji łańcucha (punktów obsługi)

Źródło: opracowanie własne.



Rys. 7.25a. Układ dwóch tras przed wymianą łańcucha
Źródło: opracowanie własne.



Rys. 7.25b. Układ dwóch tras po wymianie łańcucha
Źródło: opracowanie własne.

7.1.3. Algorytmy przeszukiwania tabu

Do grupy jednoagentowych metaheurystyk przeszukiwania przestrzeni rozwiązań problemu optymalizacyjnego zaliczyć należy zaproponowane przez F. Glover (1989) algorytmy przeszukiwania tabu. Podobnie jak w wypadku algorytmów symulowanego wyżarzania ich idea opiera się na iteracyjnym poszukiwaniu coraz lepszego rozwiązania problemu optymalizacyjnego, dopuszczając przy tym możliwość rozpatrywania rozwiązania gorszego od znajdującego w danej iteracji. Jednakże w przeciwieństwie do algorytmów symulowanego wyżarzania elementem, który dopuszcza do dalszej eksploracji przestrzeni rozwiązań rozwiązanie gorsze od poprzedniego, jest nie zewnętrzny parametr algorytmu (temperatura), ale pewna reguła akceptacji lub odrzucenia rozwiązania sąsiedniego. Innymi słowy,

algorytmy przeszukiwania tabu skupiają się w swoim działaniu przede wszystkim na sposobie (metodzie) generowania rozwiązań sąsiednich.

Akceptacja lub odrzucenie rozwiązań sąsiednich wygenerowanych na podstawie danego aktualnie rozpatrywanego rozwiązania odbywa się na podstawie pewnego zbioru reguł, stanowiących ograniczenia w wyborze najlepszego z sąsiadów, a nazywanych „tabu”. Oznacza to, że oprócz konieczności określenia najlepszej wartości funkcji celu ze zbioru rozwiązań sąsiednich niezbędne jest także sprawdzenie spełnienia przez te rozwiązania pewnych założonych wcześniej reguł dotyczących ich wyboru do dalszego generowania nowych rozwiązań. Podstawowym celem stosowania reguł tabu jest, tak jak w wypadku algorytmów symulowanego wyżarzania, zapobieganie zapętłaniu się algorytmu i jego zbyt szybkiej zbieżności do optimum lokalnego.

krok 1: inicjalizacja

repeat

krok 2: konstrukcja i ocena nowych rozwiązań sąsiednich

krok 3: akceptacja lub odrzucenie rozwiązania

krok 4: aktualizacja struktury tabu

until warunek końca

Rys. 7.26. Algorytm przeszukiwania tabu

Źródło: opracowanie własne.

Algorytm przeszukiwania tabu przedstawiony na rysunku 7.26 wymaga w pierwszej kolejności zdefiniowania reguł tabu oraz sposobu ich aktualizacji. W pierwszym kroku algorytmu konieczne jest określenie, kiedy algorytm będzie „zabraniał” przejść do wybranego najlepszego rozwiązania sąsiedniego z aktualnie rozpatrywanego. Najczęściej stosowaną strukturą tabu jest lista, w której mogą być zapamiętywane ostatnio wygenerowane i wybrane przez algorytm rozwiązania sąsiednie lub też sposoby utworzenia tych rozwiązań. Dla problemu układania tras pojazdów taka lista może mieć następującą postać:

Rozwiązanie A: [1 5 7 9 2 4 3 6 8]	zamiana {3} i {4}
Rozwiązanie B: [1 5 7 3 2 4 9 6 8]	zamiana {3} i {9}
Rozwiązanie C: [1 6 7 3 2 4 9 5 8]	zamiana {5} i {6}
Rozwiązanie D: [2 6 7 3 1 4 9 5 8]	zamiana {1} i {2}
Rozwiązanie E: [2 6 8 3 1 4 9 5 7]	zamiana {7} i {8}

Przyjmując, że rozwiązaniem wyjściowym była trasa [1 5 7 9 2 3 4 6 8], w wyniku jej modyfikacji w kolejnych iteracjach algorytmu, polegających na

sukcesywnej zamianie miejscami dwóch punktów obsługi, uzyskano pięć nowych tras. Na szczycie listy znajduje się rozwiązanie A, ponieważ uzyskane zostało najwcześniej, natomiast na dole tej listy jest najnowsze rozwiązanie E. Wygenerowanie nowego rozwiązania sąsiedniego F z aktualnie rozpatrywanego rozwiązania E powodować będzie aktualizację tej listy. Modyfikacja tej struktury odbywać się będzie przez usunięcie z pierwszego miejsca listy najdłuższego występującego w niej rozwiązania A i wprowadzenie na jej koniec nowo utworzonego rozwiązania F:

- Rozwiązanie B: [1 5 7 3 2 4 9 6 8] zamiana {3} i {9}
- Rozwiązanie C: [1 6 7 3 2 4 9 5 8] zamiana {5} i {6}
- Rozwiązanie D: [2 6 7 3 1 4 9 5 8] zamiana {1} i {2}
- Rozwiązanie E: [2 6 8 3 1 4 9 5 7] zamiana {7} i {8}
- Rozwiązanie F: [2 9 8 3 1 4 6 5 7] zamiana {6} i {9}

Ponieważ modyfikacja aktualnie rozpatrywanego rozwiązania polega na zamianie miejscami dwóch punktów obsługi, to listy przedstawione powyżej mogą być także reprezentowane w postaci tabeli dwuwymiarowej o liczbie wierszy i kolumn równej liczbie wszystkich punktów obsługi (tab. 7.1a i 7.1b):

Tabela 7.1a. Struktura tabelaryczna listy tabu dla rozwiązań A–E

	1	2	3	4	5	6	7	8	9
1		4							
2									
3				1					2
4									
5						3			
6									
7								5	
8									
9									

Źródło: opracowanie własne.

Tabela 7.1b. Struktura tabelaryczna listy tabu dla rozwiązań B–F

	1	2	3	4	5	6	7	8	9
1		3							
2									
3									1
4									
5						2			
6									5
7								4	
8									
9									

Źródło: opracowanie własne.

Parametrem struktury tabu algorytmu jest długość listy lub w wypadku reprezentacji tabelarycznej maksymalna wartość w tabeli. W obydwu wariantach oznacza to czas wyrażony liczbą iteracji, jaki jest przeznaczony na oznaczenie sposobu uzyskania rozwiązania jako tabu. Oznacza to, że do czasu, kiedy sposób utworzenia rozwiązania znajduje się w strukturze pamięci tabu, nie może on zostać wybrany do dalszego utworzenia (wybrania) kolejnego rozwiązania sąsiedniego w następnej iteracji algorytmu. Zakładając, że w trakcie działania algorytmu wygenerowane zostało kolejne najlepsze rozwiązanie sąsiednie G o postaci:

Rozwiązanie G: [2 9 8 3 1 4 5 6 7] zamiana {5} i {6}

nie zostanie ono uwzględnione przez algorytm, nawet gdyby długość trasy była tutaj korzystniejsza od długości trasy reprezentowanej przez rozwiązanie F, ponieważ przejście polegające na zamianie miejscami punktów obsługi {5} i {6} znajduje się w strukturze pamięci tabu i będzie tam jeszcze przez okres 3 iteracji. W związku z tym należy spośród pozostałych utworzonych w kroku drugim algorytmu rozwiązań sąsiednich wybrać najlepsze, którego sposób utworzenia nie jest oznaczony jako tabu (puste miejsce w tabeli tabu). Takie podejście może powodować, że do kolejnej iteracji algorytmu równie dobrze wybrane zostanie rozwiązanie reprezentujące trasę lepszą od poprzedniej, jak i reprezentujące trasę gorszą. Przyjęcie klasycznego schematu zakończenia działania algorytmu polegającego na braku

możliwości uzyskania w kolejnej iteracji rozwiązania lepszego skutkować może bardzo szybko jego zbieżnością do optimum lokalnego. Dlatego też w kroku inicjującym działanie algorytmu powinny zostać dobrane odpowiednie warunki końca działania algorytmu, tak jak w wypadku algorytmów symulowanego wyżarzania. Najczęściej stosowane kryteria zakończenia działania algorytmu to wykonanie przez niego określonej maksymalnej liczby iteracji, jak również brak uzyskania w ogóle lub znaczącej poprawy najlepszego dotychczas znalezionego rozwiązania.

Przedstawiona powyżej reguła tabu algorytmów przeszukiwania tabu stanowi pewnego rodzaju krótkookresową pamięć algorytmu, której celem jest unikanie jego zapętlenia się wokół tego samego rozwiązania. Zamiana miejscami punktów obsługi nie jest oczywiście jedyną regułą, jaka może zostać zastosowana do problemu układania tras jednego lub wielu pojazdów. Reguły tabu muszą wynikać przede wszystkim ze sposobu reprezentowania rozwiązania problemu, a także ze sposobu generowania nowych rozwiązań sąsiednich na podstawie aktualnie rozpatrywanego. Przykładem może być wykorzystanie heurystyki 3-opt do tworzenia nowego sąsiedztwa, która polega na zamianie miejscami nie dwóch jak poprzednio, ale trzech punktów obsługi. Przyjmując i tutaj, że wybór trzech punktów obsługi stanowi regułę tabu przez pewną liczbę iteracji, struktura pamięci powinna być reprezentowana przez tabelę trójwymiarową. Jednakże również ta reguła może zostać zaostrzona, co spowoduje zawężenie potencjalnych „dopuszczalnych” rozwiązań sąsiednich, przyjmując, że regułę tabu stanowić będzie wybór przynajmniej dwóch z trzech punktów znajdujących się w tabeli tabu, a wtedy tabela tabu zawierająca np. tylko dwa kolejne przejścia tabu polegające na zamianie miejscami punktów obsługi odpowiednio: {3}, {6} i {8} oraz {2}, {3} i {7} będzie miała postać (tab. 7.2):

Tabela 7.2. Struktura tabelaryczna listy tabu dla metody 3-opt

	1	2	3	4	5	6	7	8	9
1									
2			5				5		
3						4	5	4	
4									
5									
6								4	
7									
8									
9									

Źródło: opracowanie własne.

Często podejmowaną czynnością w trakcie działania algorytmu przeszukiwania tabu jest dopuszczanie łamania ustalonych reguł tabu w ściśle określonych przypadkach, noszących nazwę kryteriów aspiracji. Polega to na zezwoleniu algorytmowi na wybór rozwiązania sąsiedniego z zastosowaniem znajdującego się w tabeli tabu przejścia pod warunkiem spełnienia postawionego kryterium aspiracji. Praktycznie zawsze przyjmowanym kryterium aspiracji jest wygenerowanie z uwzględnieniem przejścia tabu rozwiązania sąsiedniego, które staje się najlepszym dotychczas znalezionym rozwiązaniem.

Jak wspomniano, długość listy tabu lub też czas wyrażony jako liczba iteracji algorytmu stanowi parametr algorytmu przeszukiwania tabu, który ma bardzo istotne znaczenie. Wpływa bowiem w sposób bezpośredni na jakość generowanych rozwiązań. W literaturze przedmiotu ustalany jest on w różnoraki sposób. Przede wszystkim jego wartość powinna zależeć od rozmiarów rozwiązywanego zadania układania tras pojazdów, a także od przyjętej metody konstrukcji rozwiązania sąsiedniego. Ponadto czas ten może być:

- stały przez cały okres działania algorytmu;
- zmienny w trakcie działania algorytmu, gdzie zmiana jego wartości podyktowana jest liczbą wykonanych już iteracji przez algorytm lub jakością najlepszego dotychczas znalezionego rozwiązania;
- losowy, gdzie wartość czasu jest losowo generowana z pewnego arbitralnie ustalonego przedziału o rozkładzie równomiernym.

Obok reguł tabu stanowiących krótkookresową strukturę pamięci algorytmu przeszukiwania tabu należy wymienić także reguły, które mogą stanowić średniookresową i długookresową strukturę pamięci, aczkolwiek nie są regułami tabu. Role, jakie odgrywają one w procesie przeszukiwania przestrzeni rozwiązań, są diametralnie różne. Pierwsze z nich mają na celu dokonanie intensyfikacji procedur przeszukiwania fragmentów przestrzeni rozwiązań, które uznać można za najbardziej obiecujące z punktu widzenia istnienia optimum globalnego. Do przykładowych reguł kształtujących procedury średniookresowej pamięci algorytmów przeszukiwania tabu zaliczyć można (Basu, 2012):

- wykorzystywanie przez określony czas (liczbę iteracji) do tworzenia rozwiązań sąsiednich często pojawiających się połączeń pomiędzy punktami obsługi, które należą do tras o niskiej długości;
- ponowne uruchamianie procesu przeszukiwania przestrzeni rozwiązań, począwszy od trasy, która jest najlepsza lub jest jedną z najlepszych znalezionych w ciągu ostatnich k przeprowadzonych iteracji algorytmu;
- w wypadku gdy rozwiązania sąsiadujące z aktualnie rozpatrywanym wybierane są w sposób losowy (np. z powodu bardzo dużej liczebności sąsiedztwa), przypisywanie większego prawdopodobieństwa wyboru tym sąsiadom, którzy zawierają połączenia należące do krótkich tras znalezionych w ostatnich k iteracjach;

- zmiana okresu, przez jaki utrzymywana jest reguła przejścia tabu w zależności od osiągnięcia optimum lokalnego.

Drugie w kolejności z wymienionych reguł to reguły długookresowe, których podstawowym celem jest dywersyfikacja procesu przeszukiwania przestrzeni rozwiązań. Zadaniem tych reguł jest stworzenie możliwości eksploracji nowych fragmentów przestrzeni rozwiązań, aby mieć większą pewność, że uzyskane dotychczas najlepsze rozwiązanie nie stanowi tylko optimum lokalnego.

Wśród reguł długookresowych algorytmów przeszukiwania tabu wymienić można między innymi:

- dodawanie pewnego rodzaju kar do pojawiających się w nowych trasach połączeń pomiędzy punktami obsługi, a tym samym zniechęcanie algorytmu do wyboru w szczególności tych połączeń, które pojawiają się najczęściej w ostatnim czasie;
- zmiana sposobu generowania nowych sąsiednich tras tworzonych na podstawie aktualnie rozpatrywanych;
- zmiana w trakcie działania kryteriów stopu algorytmu;
- wielokrotne uruchamianie algorytmu przeszukiwania tabu (stosowanie strategii wielostartu), począwszy od różnych rozwiązań początkowych;
- wprowadzenie dodatkowych parametrów do algorytmu, jak np. miary wpływu na długość trasy nowych rozwiązań sąsiednich i dokonywanie zmian w sposobie ich konstrukcji, jeżeli wpływ ten jest niesatysfakcjonujący.

Biorąc pod uwagę postać algorytmu przeszukiwania tabu pokazaną na rysunku 7.26 oraz najważniejsze charakteryzujące go elementy, można pokusić się o krótkie przedstawienie konkretnego, a zarazem prostego algorytmu przeszukiwania tabu dla problemu jednego komiwojażera. W pierwszym kroku należy ustalić rozwiązanie początkowe, od którego algorytm rozpocznie przeszukiwanie przestrzeni rozwiązań. Pierwsza trasa może zostać wygenerowana jedną z konstrukcyjnych metod heurystycznych (np. sukcesywnie wstawianie punktów obsługi). Następnie należy określić sposób generowania rozwiązań sąsiednich, czyli sposób przejścia od aktualnie rozpatrywanej trasy do następnej. W tym celu można posłużyć się sposobem przekształcenia wykorzystywanym przez algorytm 2-opt. Tym samym może mieć zastosowanie prezentowana wcześniej reguła tabu (tab. 7.1a i 7.1b) mówiąca, że przez k liczbę iteracji (parametr algorytmu) nie będzie możliwe przejście do trasy sąsiedniej, której utworzenie wymagałoby dokonania wymiany miejscami dwóch punktów obsługi znajdujących się w strukturze (tabeli) pamięci tabu.

Inny sposób budowy reguł tabu dla procedury 2-opt polegać może na utworzeniu dwóch odrębnych list tabu T_{out} oraz T_{in} zawierających – odpowiednio – połączenia usuwane z trasy i nowe połączenia do niej wstawiane (Glover, 1989; Fiechter, 1994). Obydwie listy mogą (ale nie muszą) przetrzymywać te połączenia jako tabu przez ten sam ustalony okres. Długości obu list mogą być różne. W takiej sytuacji przejście od aktualnie rozpatrywanego rozwiązania do następnego może

być zabronione, jeżeli w swojej konstrukcji uwzględnia ono usuwane stare i dodawane nowe połączenia znajdujące się w listach tabu – odpowiednio – T_{out} oraz T_{in} .

Dla aktualnie rozpatrywanej trasy pojazdu należy w następnej kolejności (krok drugi algorytmu) utworzyć zbiór tras sąsiednich będących modyfikacją trasy wyjściowej zgodnie z procedurą 2-opt i spośród tych, które nie zawierają przejścia tabu, wybrać trasę najlepszą. W kroku trzecim algorytmu, jeżeli wskazana trasa sąsiednia jest najlepszą dotychczas znaną, należy ją zapamiętać. Gdy wśród tras sąsiednich najlepsza z nich zawiera przejście tabu, można tę regułę złamać pod warunkiem, że nowa trasa będzie najlepszą dotychczas znaną. Zaakceptowana w kroku trzecim algorytmu trasa sąsiednia staje się trasą aktualnie rozpatrywaną. Ostatni, czwarty krok algorytmu przeszukiwania tabu to wykonanie aktualizacji struktury pamięci (tabeli) tabu przez dodanie do listy nowego przejścia tabu na okres k iteracji, a także zmniejszenie o jednostkę czasu uznawania jako tabu poprzednio dokonanych przejść do tras sąsiednich. Wymienione procedury należy powtórzyć tyle razy, ile wynosi ustalona na początku maksymalna liczba iteracji algorytmu.

Nieco bardziej złożoną postać algorytmu przeszukiwania tabu zaproponował C.N. Fiechter (1994), zalecaną zwłaszcza do rozwiązywania problemów jednego komiwojażera o znacznych rozmiarach¹. Algorytm ten nazwany został przez autora algorytmem wysokiego poziomu, którego ogólny schemat przedstawiono na rysunku 7.27.

```

krok 1: inicjalizacja
repeat
    krok 2: intensyfikacja
    repeat
        krok 3: dywersyfikacja
        krok 4: aktualizacja struktury tabu
    until maksymalna liczba iteracji 2
until maksymalna liczba iteracji 1
  
```

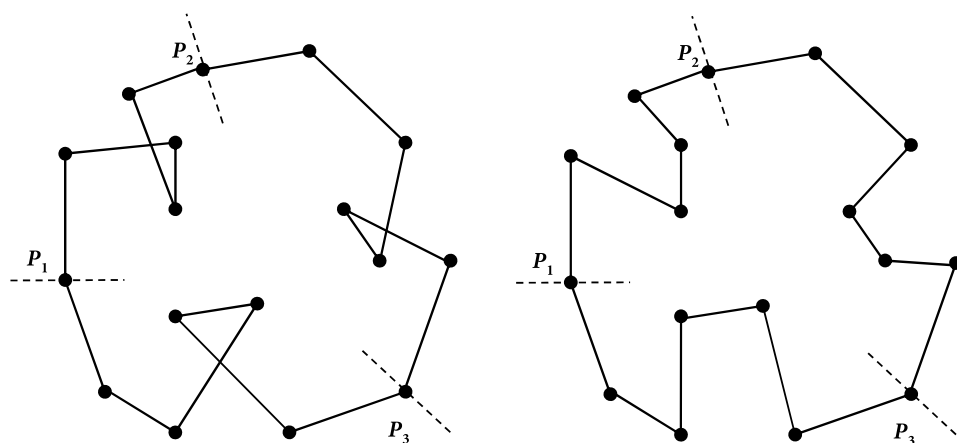
Rys. 7.27. Algorytm przeszukiwania tabu wyższego poziomu C.-N. Fiechtera

Źródło: opracowanie własne.

Algorytm ten odróżnia od podstawowej wersji algorytmu przeszukiwania tabu przede wszystkim wprowadzenie do ogólnej jego postaci dwóch bardzo istotnych elementów: realizowanej w kroku drugim procedury intensyfikacji oraz w kroku trzecim procedury dywersyfikacji. Stanowią one realizację przedstawionych wcześniej reguł – odpowiednio – średniookresowych (intensyfikacja) i długoterminowych (dywersyfikacja). To właśnie te procedury, zdaniem autora, stanowią o wyższym poziomie algorytmu przeszukiwania tabu w porównaniu do najprostszej

1 Rozwiązywane problemy komiwojażera liczyły od 500 do 100 000 punktów obsługi.

jego postaci, nazywanej algorytmem niższego poziomu. Pozwalają one na bardziej dokładną, a zarazem szerszą eksplorację przestrzeni rozwiązań w szczególności dla problemów układania tras pojazdu o dużych rozmiarach.



Rys. 7.28. Procedura intensyfikacji w algorytmie przeszukiwania tabu C.-N. Fiechtera
Źródło: opracowanie własne na podstawie (Fiechter, 1994).

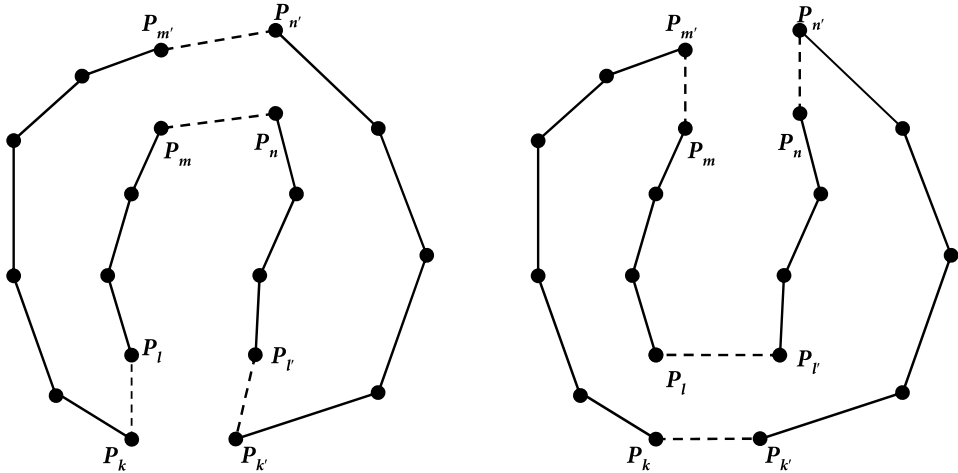
Pierwsza z procedur realizowanych w kroku drugim algorytmu polega na dekompozycji (pocięciu) aktualnie rozpatrywanej trasy na pewną liczbę jej podzbiórów (podtras) i zakłada ich niezależną od siebie poprawę z wykorzystaniem prostych metod lokalnej optymalizacji (rys. 7.28).

Każda z tras dzielona jest na mniejsze trasy, zawierające ustaloną (poniżej 100) liczbę punktów obsługi². Trasa jest dzielona w wybranych punktach obsługi, które stanowią granicę pomiędzy kolejnymi podtrasami. W dalszej kolejności wykorzystana zostaje metoda 2-opt przy zachowaniu tych samych granicznych punktów obsługi. W efekcie po ponownym złożeniu zmodyfikowanych fragmentów tras uzyskiwana jest nowa trasa pojazdu, której długość (lub koszt) jest nie gorsza od trasy rozpatrywanej w tej procedurze trasy wyjściowej.

Istota prezentowanego algorytmu przeszukiwania tabu zawiera się w krokach trzecim i czwartym algorytmu. Tam też określone zostały reguły tabu dla kolejno generowanych rozwiązań sąsiednich. Proces przejścia od jednego rozwiązania do drugiego odbywa się na podstawie tzw. superprzejścia, które polega na znacznej modyfikacji aktualnie rozpatrywanej trasy w porównaniu do np. procedury 2-opt. Zastosowanie przedstawionej poniżej modyfikacji trasy pojazdu wynika z poczynionej obserwacji, że wiele dobrych tras zawiera często wiele takich samych

² Liczba ta jest stała i niezależna od rozmiarów rozwiązywanego zadania komiwojażera, co oznacza, że dla problemów komiwojażera o większych rozmiarach rośnie liczba podzbiórów trasy.

podciągów punktów obsługi (fragmentów tras), lecz ułożonych w tych trasach w odwrotnej kolejności. Stąd też proponowany sposób superprzejścia pozwala na zachowanie tych sekcji tras przy jednoczesnym odwróceniu kolejności obsługi punktów przez pojazd w ramach każdej z nich (rys. 7.29).



Rys. 7.29. Procedura superprzejścia
Źródło: opracowanie własne na podstawie (Fiechter, 1994).

Idea przedstawionego superprzejścia polega na znalezieniu dwóch najbliższych, ale nie sąsiadujących ze sobą par połączeń pomiędzy punktami obsługi (P_k, P_l) i $(P_{k'}, P_{r'})$ oraz (P_m, P_n) i (P_m', P_n') , a następnie usunięciu ich i zastąpieniu nowymi. Najbliższa sobie para połączeń w trasie pojazdu (P_k, P_l) i $(P_{k'}, P_{r'})$ wyznaczona przez wskazany punkt obsługi P_k jest to ta para, dla której wartość Δf_{P_k} określona zależnością (7.12) jest najmniejsza.

$$\Delta f_{P_k} = d(P_k, P_{k'}) + d(P_l, P_{r'}) - [d(P_k, P_l) + d(P_{r'}, P_{k'})] \quad (7.12)$$

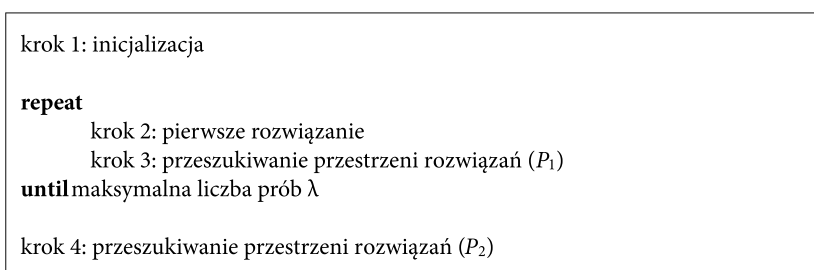
Superprzejście jest wykonywane przez usunięcie połączeń (P_k, P_l) i $(P_{k'}, P_{r'})$, a następnie zastąpienie ich nowymi odcinkami trasy: $(P_k, P_{k'})$ i $(P_{r'}, P_l)$. Po znalezieniu najbliższych par połączeń dla każdego punktu obsługi w trasie należy wybrać te punkty, dla których zachodzi (7.13):

$$\Delta f_{P_k} + \Delta f_{P_m} \rightarrow \min. \quad (7.13)$$

i wykonać stosowne superprzejście (rys. 7.29). Co więcej, autor rozważa przy wyborze odpowiedniego superprzejścia tylko takie, gdzie najkrótszy z fragmentów trasy (P_r, \dots, P_m) , (P_n, \dots, P_l) , $(P_{k'}, \dots, P_n')$ i (P_m', \dots, P_k) zawiera nie mniej niż r punktów obsługi, gdzie r jest parametrem algorytmu.

Właściwa algorytmowi przeszukiwania tabu reguła tabu została ustalona właśnie dla przedstawionego powyżej superprzejścia. Polega ona na utworzeniu stosownej listy, w której przechowywane są wszystkie punkty obsługi zaangażowane w proces superprzejścia: $P_k, P_k', P_r, P_r', P_m, P_m', P_n, P_n'$. Natomiast rozważane w danym momencie superprzejście będzie uznane za tabu (a tym samym dane rozwiązanie sąsiednie zostanie odrzucone), jeżeli będzie zawierało w swojej konstrukcji co najmniej ustaloną liczbę p punktów obsługi znajdujących się na liście tabu (gdzie p jest także parametrem algorytmu)³. Długość listy tabu jest uzależniona od rozmiarów rozwiązywanego problemu (wartość ta zmienia się i może liczyć od 15 do 800 pozycji) i za każdym razem usuwanych jest z niej 8 najdłużej tam występujących punktów obsługi, a na ich miejsce umieszczane są nowe, tworzące superprzejście.

Najbardziej znanym reprezentantem algorytmów przeszukiwania tabu dla problemu układania tras wielu pojazdów jest algorytm *Taburoute* zaproponowany przez M. Gendreaua, A. Hertza i G. Laporte'a (1994) (rys. 7.30).



Rys. 7.30. Algorytm *Taburoute*
Źródło: opracowanie własne.

W ogólnym zarysie składa się on z trzech etapów. Pierwszy to inicjalizacja, w której ustalane są wartości początkowe poszczególnych parametrów algorytmu. Drugi etap to konstrukcja kilku próbnych rozwiązań i próba ich poprawy procedurą przeszukiwania przestrzeni rozwiązań z pierwszym zestawem parametrów P_1 . Wreszcie trzeci, ostatni etap to implementacja tej samej procedury poszukiwania lepszego rozwiązania, ale z innym zestawem parametrów P_2 .

Przed wszystkim należy zauważyć, że omawiany algorytm przeszukiwania tabu dopuszcza możliwość uwzględniania i oceny rozwiązań niedopuszczalnych. W sformułowanym problemie układania tras pojazdów przedstawione zostały dwa typy ograniczeń. Jedno dotyczące maksymalnej ładowności pojazdów, drugie zaś – maksymalnej długości pojedynczej trasy pojazdu. Złamanie któregośkolwiek z ograniczeń prowadzi do uzyskania zbioru tras niedopuszczalnych. Autorzy proponują następującą formułę oceny rozpatrywanych w trakcie działania algorytmu rozwiązań (7.14):

3 Przyjęte przez autorów wartości p i r wynoszą odpowiednio 5 oraz 25.

$$F_2(S) = F_1(S) + \alpha K_Q + \beta K_L \quad (7.14)$$

gdzie:

- $F_1(S)$ – koszt ocenianego rozwiązania dopuszczalnego;
- $F_2(S)$ – koszt ocenianego rozwiązania niedopuszczalnego;
- K_Q – kara za przekroczenie warunku ładowności pojazdu;
- K_L – kara za przekroczenie warunku długości trasy pojazdu;
- α, β – parametry przyjmujące wartości dodatnie.

W sytuacji, kiedy wygenerowany przez algorytm zbiór tras jest dopuszczalny, wartość takiego rozwiązania wprost przyjmuje wartość funkcji celu (suma długości wszystkich tras). W przeciwnym razie łączna długość wszystkich tras rozwiązania niedopuszczalnego powiększana jest o ważoną współczynnikiem karę, która jest wprost proporcjonalna do wartości przekroczenia dla zadanego ograniczenia. Jeżeli przekroczone jest więcej niż jedno ograniczenie dotyczące ładowności lub maksymalnej długości trasy, wtedy nakładane kary są kumulowane.

Serce algorytmu *Taburoute* stanowi uruchamiana tak w kroku trzecim, jak i w kroku czwartym algorytmu procedura przeszukiwania przestrzeni rozwiązań, która stosuje reguły tabu (rys. 7.31).

krok 1: inicjalizacja

repeat

krok 2: wybór q punktów obsługi z W

repeat

krok 3: konstrukcja rozwiązania sąsiedniego i jego ocena

krok 4: akceptacja lub odrzucenie rozwiązania sąsiedniego

until wszystkie wybrane q punkty obsługi zostaną rozpatrzone

krok 5: wybór najlepszego rozwiązania sąsiedniego

krok 6: przejście do rozwiązania sąsiedniego

krok 7: aktualizacja struktury tabu

krok 8: dostosowanie kar

until warunek końca

Rys. 7.31. Procedura przeszukiwania tabu algorytmu *Taburoute*

Źródło: opracowanie własne.

Procedura ta posiada następujący zbiór parametrów wejściowych:

- W – niepusty zbiór punktów obsługi, które mogą zostać usunięte z trasy aktualnie rozpatrywanego rozwiązania;
- q – liczba punktów obsługi należących do zbioru W , które mogą zostać wstawione do innej trasy;

- $\theta_{\min.}, \theta_{\max.}$ – granice okresu tabu;
 g – współczynnik skalowania sztucznej wartości funkcji celu;
 h – częstotliwość, z jaką są modyfikowane wartości współczynników α i β ;
 $n_{\max.}$ – maksymalna liczba iteracji, przy której wywoływana jest procedura bez żadnej poprawy dotychczas najlepszego znalezionej rozwiązania.

Dla każdego z wybranych ze zbioru W w pierwszym kroku procedury punktów obsługi generowane jest możliwe sąsiedztwo aktualnie rozpatrywanego rozwiązania (krok drugi). Odbywa się to przez usunięcie punktu z aktualnej trasy pojazdu i umieszczenie go w nowej istniejącej trasie (w najlepszym możliwym miejscu) lub w nowej pustej trasie (jeżeli liczba użytych dotychczas pojazdów nie przekracza maksymalnej liczby pojazdów, jakimi dysponuje baza). Przemieszczenie punktu obsługi do innej trasy jest możliwe, jeżeli nie jest ono identyfikowane jako tabu. Oznacza to, że zaproponowana przez autorów struktura pamięci tabu stanowi listę, której elementy określają, czy punkt obsługi P_i był umieszczany w ostatnim czasie w trasie k -tego pojazdu. Po dokonaniu przemieszczenia punktu do innej trasy należy ocenić nowe rozwiązanie i dokonać jego akceptacji lub je odrzucić. Pomimo istniejącej reguły tabu rozwiązanie, które powstanie przez dokonanie operacji oznaczonej jako tabu, może zostać zaakceptowane. Dzieje się tak wtedy, gdy zostanie spełnione kryterium aspiracji mówiące o tym, że możliwe jest zaakceptowanie takiego rozwiązania, jeśli jest ono lepsze od najlepszego dotychczas znalezionej.

Śród zaakceptowanych rozwiązań sąsiednich w kroku piątym należy wskazać rozwiązanie sąsiednie najlepsze dopuszczalne lub niedopuszczalne. Wskazane najlepsze rozwiązanie sąsiednie wcale nie musi być nowym aktualnie rozpatrywanym rozwiązaniem sąsiednim. W kroku szóstym należy sprawdzić, czy poprzednie rozwiązanie było dopuszczalne i czy nie było poprawiane przez lokalną poprawę oddzielnie każdej z tras. Jeżeli spełnione są łącznie trzy warunki:

- 1) najlepsze znalezione rozwiązanie sąsiednie jest gorsze od aktualnie rozpatrywanego;
- 2) wszystkie trasy pojazdów aktualnie rozpatrywanego rozwiązania problemu wielu komiwojazerów są dopuszczalne;
- 3) w poprzedniej iteracji algorytmu nie dokonywano poprawy poszczególnych tras pojazdów;

to należy dokonać lokalnej poprawy każdej z tras oddzielnie, a uzyskane w ten sposób rozwiązanie przyjąć jako nowe w kolejnej iteracji procedury przeszukiwania tabu. W takiej sytuacji nie ma mowy o przejściu z aktualnie rozpatrywanego rozwiązania do rozwiązania sąsiedniego, lecz następuje modyfikacja (poprawa) istniejącego.

W kroku siódmym procedury przeszukiwania tabu dokonywana jest aktualizacja listy tabu pod warunkiem, że w danej iteracji procedury nastąpiło przejście do rozwiązania sąsiedniego, a nie lokalna poprawa aktualnie rozpatrywanego. Umieszczenie na liście tabu przejścia określającego, który punkt obsługi został

umieszczony w której trasie pojazdu, następuje na okres θ iteracji, gdzie wartość θ jest losowana z przedziału $[\theta_{\min.}, \theta_{\max.}]$. Wreszcie ostatnią czynnością, jaką należy wykonać, jest krok ósmy, w którym następuje dostosowanie wartości ustanowionych kar dla rozwiązań niedopuszczalnych. Następuje to nie w każdej iteracji procedury, lecz co liczbę h iteracji. Jeżeli w poprzednich h iteracjach wszystkie wybierane rozwiązania były dopuszczalne z punktu widzenia ładowności pojazdu, to wartość współczynnika α jest zmniejszana do poziomu $\alpha / 2$. Natomiast jeżeli wszystkie rozwiązania były niedopuszczalne, to wartość tego współczynnika jest podwajana i wynosi 2α .

Analogiczny schemat postępowania odnosi się do współczynnika β skalującego wartość kary w razie przekroczenia ograniczenia dotyczącego długości trasy pojazdu. W wypadku, gdy w ostatnich h iteracjach procedury przeszukiwania tabu część rozwiązań była dopuszczalna, a część niedopuszczalna, to obydwaj współczynniki należy pozostawić bez zmian. Warunkiem końca działania prezentowanej procedury przeszukiwania tabu jest wykonanie ustalonej liczby jej iteracji bez poprawy znalezionej rozwiązania.

Pewną modyfikacją przedstawionego powyżej algorytmu *Taburoute* jest propozycja algorytmu GTS (*Granular Tabu Search*) zaprezentowana przez P. Totha i D. Vigo (2003). Algorytm GTS zawiera większość przedstawionych cech algorytmu *Taburoute*. Pierwsza różnica pojawia się w kroku inicjalizacji algorytmu. W celu wygenerowania rozwiązania początkowego wykorzystywana jest konstrukcyjna metoda heurystyczna oszczędnościowego łączenia tras G. Clarka i J.W. Wrighta. Jeżeli utworzona liczba tras jest większa od liczby posiadanych pojazdów, wtedy trasy (nadprogramowe) najmniej wykorzystanych pojazdów są likwidowane, a punkty obsługi są wstawiane do pozostałych tras. Druga różnica pomiędzy algorytmem GTS a *Taburoute* dotyczy sposobu konstruowania rozwiązań sąsiednich. Trasy sąsiednie konstruowane są na podstawie trasy aktualnej przez zastosowanie omawianych wcześniej metod wymiany k połączeń pomiędzy punktami obsługi, polegającej na usunięciu z tras k odcinków pomiędzy sąsiednimi punktami obsługi i zastąpieniu ich innym zbiorem k odcinków. Ze względu na bardzo szybki wzrost liczby potencjalnych rozwiązań sąsiednich wraz ze wzrostem k proponowane są tutaj metody wymiany k odcinków, gdzie $k \leq 4$.

Istotnym elementem w zasadniczy sposób odróżniającym algorytm GTS od innych są kryteria wyboru połączeń pomiędzy punktami obsługi, które są wybierane do konstrukcji sąsiednich zbiorów tras. Połączenia te tworzą pewien graf $G'(V, A')$, będący podzbiorem grafu wyjściowego $G(V, A)$, gdzie zbiór połączeń pomiędzy punktami obsługi (krawędzi grafu) A' stanowi zbiór odcinków tras (krawędzi), które:

- są początkowe lub końcowe w trasach (jednym z końców jest baza);
- są ważne z punktu widzenia jakości uzyskanego wcześniej rozwiązania (należą do krótkich tras);
- charakteryzują się długością nieprzekraczającą pewnego progu.

Definiując w trzecim punkcie krótkie odcinki pomiędzy punktami obsługi, autorzy określają sposób obliczenia progu (7.15), stanowiącego o uznaniu takiego połączenia za wartę rozpatrzenia przy konstrukcji rozwiązania sąsiedniego.

$$\vartheta = \beta \frac{z'}{(N + K)} \quad (7.15)$$

gdzie:

- N – liczba punktów obsługi;
- K – liczba pojazdów będących w dyspozycji bazy (maksymalna liczba tras);
- z' – wartość funkcji celu rozwiązania uzyskanego z zastosowaniem algorytmu *savings* G. Clarka i J.W. Wrighta;
- β – parametr, gdzie $\beta \geq 0$.

Idea zastosowanego kryterium wyboru połączeń za pomocą progu ϑ ma swoje źródło w obserwacji, że długie połączenia pomiędzy punktami obsługi mają raczej niewielkie prawdopodobieństwo uzyskania dobrych jakościowo rozwiązań. Te charakteryzują się znacznie częściej zbiorem krótkich odcinków tras. Obserwację tę poczyniono dla wybranego zbioru zadań testowych, dla których znane są najlepsze znalezione rozwiązania (dowolnym algorytmem), porównując obliczoną wartość progu ϑ z długościami odpowiednio minimalnymi, maksymalnymi i średnimi połączeń pomiędzy punktami obsługi. Zauważono, że znaczna większość wszystkich możliwych połączeń pomiędzy punktami obsługi dla danego zadania testowego cechuje się długością większą od obliczonego dla niego progu ϑ . Stąd też autorzy doszli do wniosku, że w znacznym stopniu można usprawnić działanie algorytmu przez konstruowanie rozwiązań sąsiednich zawierających te trasy, w których skład wchodzi połączenia o długości nieprzekraczającej obliczonej wcześniej wartości progu ϑ .

Parametrem jest tutaj wartość współczynnika β , stanowiąca o rozpiętości zbioru rozwiązań sąsiednich. Wyższa wartość β powoduje możliwość utworzenia większej liczby rozwiązań sąsiednich. Współczynnik β zazwyczaj zawiera się w przedziale $[0,5; 5]$, natomiast najlepsze rozwiązania uzyskano dla wartości β z przedziału $[1; 2,5]$.

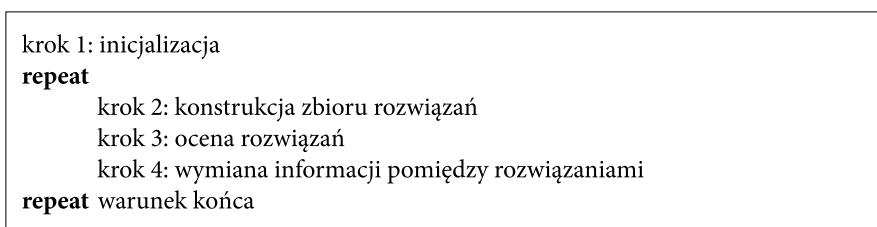
7.2. Metaheurystyki wieloagentowe

Obok metod należących do heurystyk jednoagentowych osobną klasę stanowią metaheurystyki wieloagentowe. W przeciwieństwie do tych pierwszych w procesie przeszukiwania przestrzeni rozwiązań zadań układania tras jednego lub wielu pojazdów dokonują one przekształcenia w danym momencie nie jednego, a wielu rozwiązań

jednocześnie. Do tej klasy algorytmów zaliczyć można te, które konstruowane są na podstawie dwóch następujących typów metaheurystyk:

- algorytmów ewolucyjnych;
- algorytmów mrówkowych.

Ogólny schemat działania metod heurystycznych przeszukiwania przestrzeni rozwiązań należących dla klasy heurystyk wieloagentowych przedstawiono na rysunku 7.32.



Rys. 7.32. Algorytm wieloagentowy

Źródło: opracowanie własne.

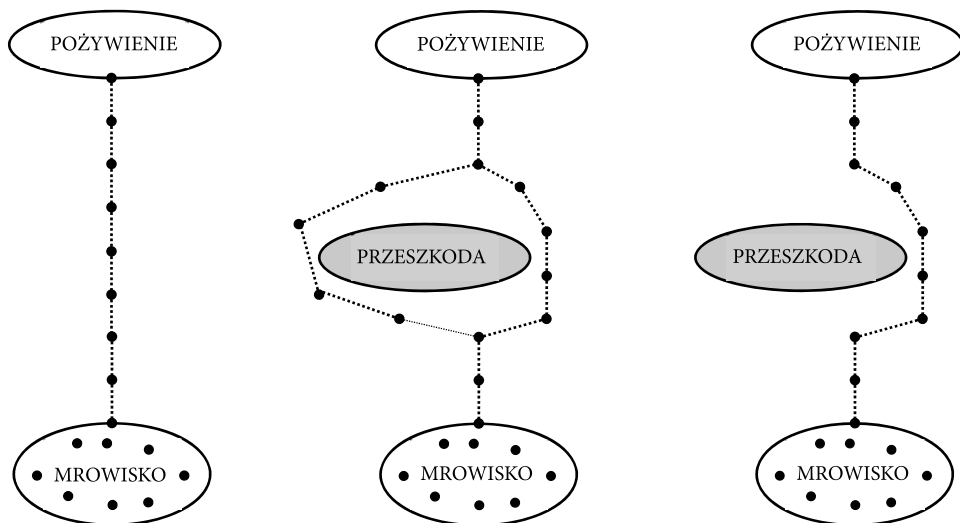
Krok pierwszy algorytmu wieloagentowego to konstrukcja początkowego zbioru rozwiązań. Dla zadania układania tras pojazdów oznacza to konieczność skonstruowania zbioru pojedynczych tras (dla problemu komiwojażera) lub zbioru kompletnych rozwiązań (zbiorów tras dla problemów wielu komiwojażerów). Tak jak w wypadku prezentowanych wcześniej algorytmów jednoagentowych, procedura ta może odbyć się z wykorzystaniem zarówno mechanizmów losowych, jak i znanych algorytmów konstrukcyjnych. W dalszej kolejności jednoczesnej ocenie poddawane jest nie jedno, a wszystkie skonstruowane rozwiązania. Czasami konieczna jest także jedna ocena wszystkich rozwiązań razem (całego aktualnie rozpatrywanego zbioru), w celu określenia, czy algorytm uzyskuje coraz lepsze rozwiązania, czy też np. osiągnął zbieżność do optimum lokalnego.

Konstrukcja nowych rozwiązań odbywać się może zarówno przez modyfikacje indywidualnych rozwiązań z ocenianego zbioru, jak i (co wyróżnia wieloagentowe systemy przeszukiwania przestrzeni rozwiązań) przez wymianę informacji zawartych w więcej niż jednym rozwiązaniu. Nowe rozwiązanie może powstać np. w wyniku zestawienia ze sobą części tras z jednego rozwiązania z częścią tras innego rozwiązania. Warunkiem końca algorytmu jest zazwyczaj wykonanie określonej liczby iteracji.

Tak jak w algorytmach należących do systemów jednoagentowych jednym z najważniejszych elementów jest sposób konstrukcji kolejnych rozwiązań sąsiednich, tak w algorytmach należących do systemów wieloagentowych ciężar ten spoczywa na sposobie wymiany informacji pomiędzy poszczególnymi rozwiązaniami.

7.2.1. Algorytmy mrówkowe

W porównaniu do omawianych w następnym podrozdziale algorytmów ewolucyjnych algorytmy mrówkowe stanowią stosunkowo prosty przykład realizacji zasad, którymi kierują się algorytmy wieloagentowe. Algorytmy mrówkowe po raz pierwszy zostały zaprezentowane na początku lat dziewięćdziesiątych XX wieku w odniesieniu do problemów optymalizacyjnych o charakterze kombinatorycznym, a w szczególności do problemu jednego komiwojażera i szeregowania zleceń produkcyjnych (Coloni i in., 1991; 1994). Ich idea opiera się na obserwacji zachowania pewnych zbiorowości organizmów żywych różnych gatunków. Poszczególne osobniki mogą w swoim zachowaniu nie wykazywać szczególnie interesujących cech lub właściwości, w przeciwieństwie do tych właściwości lub cech, które mogą zostać zidentyfikowane, gdy będzie się obserwować całą zbiorowość osobników.



Rys. 7.33. Zachowanie mrówek podążających do miejsca pożywienia
Źródło: opracowanie własne na podstawie (Dorigo i in., 1996).

Taką obserwowaną zbiorowością organizmów żywych były mrówki. W wyniku przeprowadzonych eksperymentów zauważono, że mrówki, działając indywidualnie, mają znacznie ograniczone zdolności (np. w poszukiwaniu pożywienia), natomiast gdy działają w grupie, ich możliwości znacznie wzrastają. Przeprowadzony eksperyment polegał na obserwacji zachowania mrówek wyruszających z mrowiska do miejsca, w którym znajduje się pożywienie, a następnie powracających z nim do swojego gniazda. Gdy droga prowadząca do miejsca pożywienia była pusta, widać było tylko jedną ścieżkę mrówek prowadzącą do tego miejsca. Z kolei gdy na ścieżce tej umieszczono przeszkodę, początkowo mrówki utworzyły dwie

ścieżki omijające z obu stron przeszkodę, aby w późniejszym czasie znów powrócić do jednej ścieżki, po której się poruszały do docelowego miejsca pożywienia (rys. 7.33).

Takie zachowanie mrówek jest wynikiem przekazywania sobie informacji o drodze, jaką pokonują ich poprzedniczki. Odbywa się to przez pozostawianie na przebytej ścieżce przez każdą z mrówek pewnej ilości substancji chemicznej nazywanej feromonem. Każda następna mrówka, poszukując pożywienia, kieruje się ilością feromonu znajdującą się na możliwych do przebycia przez nią ścieżkach, wybierając z największym prawdopodobieństwem tę ścieżkę, na której natężenie feromonu jest największe. Mrówka, podążając wybraną przez siebie ścieżką, także zostawia pewną ilość feromonu. Tym samym można stwierdzić, że im więcej mrówek przejdzie daną ścieżką, tym większa ilość feromonu zostanie na niej zostawiona, a to z kolei prowadzić będzie do tego, że dana ścieżka okaże się chętniej wybierana przez kolejne mrówki. Ponadto krótsze ścieżki prowadzące do miejsca pożywienia powodują, że więcej mrówek może nimi przejść w jednostce czasu, a tym samym zostawić na nich więcej feromonu. Oznacza to, że krótsze ścieżki będą bardziej preferowane przez kolejne mrówki. Należy jednocześnie zauważyć, że cały proces nie jest zdeterminowany – mrówki pomimo dużego natężenia feromonu na ścieżce nie muszą wcale jej wybrać i mogą udać się do celu inną drogą.

Przedstawione powyżej obserwacje posłużyły do konstrukcji podstawowego sztucznego systemu mrówkowego, który dał początek szerokiej klasie rozwijanych później algorytmów mrówkowych. Ogólny kształt algorytmu mrówkowego w postaci pseudokodu przedstawiony został na rysunku 7.34.

```

krok 1: inicjalizacja
repeat
    krok 2: wybór miejsca startu
    krok 3: odparowanie feromonu
    krok 4: budowa rozwiązania
    krok 5: aktualizacja feromonu
until określona liczba iteracji
  
```

Rys. 7.34. Algorytm mrówkowy (budowa przez mrówki całych tras)

Źródło: opracowanie własne.

Ze względu na dość naturalną analogię poszukiwania najkrótszej drogi do pożywienia przez mrówki do problemu jednego komiwojażera przedstawiony algorytm mrówkowy zastosowany został w pierwszej kolejności do jego rozwiązania i na przykładzie tego problemu optymalizacyjnego szczegółowo objaśniony.

Trasa (ścieżka) każdej mrówki odpowiada jednej zbudowanej trasie pojazdu. W kroku pierwszym algorytmu, pomimo że jeszcze żadna mrówka nie zbudowała żadnej trasy, na każde połączenie pomiędzy dowolną parą punktów obsługi P_i i P_j

należy nanieść pewną taką samą ilość feromonu τ_0 oraz utworzyć populację (zbiorowość) m mrówek. Liczebność utworzonej populacji mrówek jest istotnym parametrem algorytmu. Kroki drugi, trzeci, czwarty oraz piąty algorytmu mrówkowego stanowią jego główną iterację. W pierwszej kolejności należy ustalić, z którego punktu obsługi mrówka rozpocznie budowę swojej trasy. Następnie odparowywana jest pewna ilość feromonu (krok trzeci), co w rzeczywistości odpowiada procesowi ulatniania się tej substancji chemicznej. Natomiast w optymalizacji problemu jednego komiwojażera oznacza to sytuację, w której nieodwiedzane przez pojazd połączenia są z biegiem czasu coraz mniej atrakcyjne dla kolejnych mrówek (w szczególności długie odcinki tras pomiędzy wybranymi punktami obsługi).

Krok czwarty jest jednym z najistotniejszych etapów algorytmu, a następuje w nim proces sekwencyjnego wyboru kolejnych punktów obsługi przez mrówkę (pojazd). W kroku tym mrówka, wybierając kolejny punkt obsługi, kieruje się ilością feromonu znajdującą się na wszystkich możliwych połączeniach pomiędzy miejscem, w którym się znajduje, a kolejnym jeszcze nieodwiedzonym punktem obsługi. W tym kroku także musi być odpowiednio aktualizowana i wykorzystywana struktura pamięci każdej mrówki, w której przechowywane są informacje o już częściowo zbudowanej trasie. Wreszcie po skonstruowaniu całej ścieżki (trasy) przez mrówkę w kolejnym, piątym kroku algorytmu należy dokonać aktualizacji poziomu feromonu, ale tylko na połączeniach pomiędzy punktami obsługi, które należą do zbudowanej przez nią trasy. Przedstawiona powyżej procedura powtarzana jest dla każdej mrówki oddzielnie, rozpoczynając od losowo wygenerowanego punktu obsługi. Liczba mrówek wykorzystanych do budowy tras stanowi liczbę iteracji wykonywanych przez algorytm.

Nieco inna postać algorytmu mrówkowego zakłada, że w każdej jego iteracji wykonywane jest przejście każdej mrówki od jednego punktu obsługi do innego (rys. 7.35). W pierwszym inicjującym kroku algorytmu w każdym punkcie obsługi umieszczana jest pewna (niekoniecznie taka sama) liczba mrówek, które w głównej pętli algorytmu stopniowo powiększają swoje trasy o kolejne punkty obsługi. Mrówki te, wybierając kolejne punkty (w pętli wewnętrznej algorytmu – krok 3), także kierują się ilością feromonu znajdującą się na poszczególnych połączeniach

```

krok 1: inicjalizacja
repeat
    krok 2: odparowanie feromonu
    repeat
        krok 3: dołączenie punktu obsługi do trasy mrówki
    until wszystkie mrówki w nowym punkcie obsługi
    krok 4: aktualizacja feromonu
until wszystkie trasy mrówek zbudowane

```

Rys. 7.35. Algorytm mrówkowy (przejścia mrówek do kolejnych punktów)
Źródło: opracowanie własne.

między punktami obsługi. Podobnie jak we wcześniejszej postaci algorytmu liczba mrówek jest istotnym parametrem algorytmu (stanowi o liczbie powtórzeń kroku trzeciego), natomiast liczba iteracji algorytmu równa jest liczbie punktów obsługi, które należy dołączyć do trasy każdej mrówki.

Taka konstrukcja algorytmu mrówkowego powoduje, że poziom feromonu aktualizowany jest na wszystkich połączeniach, a nie tylko na pewnym ich podzbiorze po zbudowaniu trasy przez jedną mrówkę.

Wybór kolejnego punktu obsługi przez mrówkę w procesie konstrukcji jej ścieżki (trasy) ma charakter zrandomizowany. Prawdopodobieństwo wyboru przez k -tą mrówkę połączenia pomiędzy punktami obsługi P_i oraz P_j jest uzależnione od ilości znajdującego się na nim feromonu oraz od jego długości (7.16):

$$p_{ij}^k = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} \{[\tau_{il}(t)]^\alpha [\eta_{il}]^\beta\}} \quad (7.16)$$

gdzie:

- τ_{ij} – poziom feromonu na połączeniu pomiędzy i -tym a j -tym punktem obsługi;
- η_{ij} – widziana przez mrówkę ilość feromonu na połączeniu pomiędzy i -tym a j -tym punktem obsługi, przy czym η_{ij} jest odwrotnie proporcjonalne do długości połączenia pomiędzy nimi ($\eta_{ij} = 1 / d_{ij}$);
- α, β – parametry pozwalające na pewną kontrolę ważności relacji pomiędzy faktyczną ilością feromonu znajdującą się na połączeniu pomiędzy wskazanymi punktami obsługi a ilością widzianą przez k -tą mrówkę.

Szczególną grupę algorytmów mrówkowych stanowią algorytmy kolonii mrówkowych, w których *novum* stanowi sposób wyboru przez mrówkę kolejnego punktu obsługi w celu jego wstawienia do trasy (Dorigo i Gambardella, 1997). Wybór j -tego punktu obsługi przez mrówkę, która znajduje się w i -tym punkcie, następuje zgodnie z przedstawioną poniżej regułą (7.17):

$$S = \begin{cases} \arg \max_{j \in N_i^k} [\tau_{ij}(t) \eta_{ij}^\beta], & \text{dla } q \leq q_0 \\ p_{ij}^k, & \text{dla } q \geq q_0 \end{cases} \quad (7.17)$$

gdzie:

- q_0 – parametr algorytmu, przy czym $q_0 \in (0, 1)$;
- q – losowa liczba z przedziału $(0, 1)$;
- τ_{ij} – poziom feromonu na połączeniu pomiędzy i -tym a j -tym punktem obsługi;
- η_{ij} – widziana przez mrówkę ilość feromonu na połączeniu pomiędzy i -tym a j -tym punktem obsługi, przy czym η_{ij} jest odwrotnie proporcjonalne do długości połączenia pomiędzy nimi ($\eta_{ij} = 1 / d_{ij}$);

- β – parametr określający ważność pomiędzy śladem feromonu a odwrotnością odległości ($\eta_{ij} = 1 / d_{ij}$);
- N_i^k – zbiór punktów obsługi, które jeszcze nie zostały odwiedzone przez k -tą mrówkę znajdującą się w i -tym punkcie;
- p_{ij}^k – prawdopodobieństwo wyboru przez k -tą mrówkę do trasy połączenia pomiędzy punktami obsługi P_i oraz P_j obliczone zgodnie ze wzorem (7.16).

Przedstawiona powyżej reguła, nazywana pseudolosową regułą proporcjonalności, pozwala na poruszanie się algorytmu pomiędzy procesami eksploatacji a eksploracji przestrzeni rozwiązań. W pierwszym wypadku (eksploatacja) dokonuje się wybór (preferowanie) tych połączeń pomiędzy punktami obsługi, które są najkrótsze. Natomiast w drugim obliczane jest prawdopodobieństwo wyboru każdego nieodwiedzonego jeszcze punktu obsługi zgodnie z podstawową regułą opisaną wzorem (7.16).

Choć w przedstawionych postaciach algorytmów mrówkowych odparowanie i aktualizacja poziomu feromonu przedstawione zostały jako osobne kroki algorytmu, to bardzo często są one traktowane jako jeden proces opisany następującym wzorem (7.18):

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}(t) \quad (7.18)$$

gdzie:

- ρ – część odparowanego feromonu;
- $\Delta\tau_{ij}$ – przyrost poziomu feromonu na połączeniu pomiędzy i -tym a j -tym punktem obsługi.

Pierwszy składnik sumy (7.18) reprezentuje krok algorytmu, w którym następuje odparowanie feromonu w danej iteracji algorytmu, podczas gdy drugi składnik nakłada pewną ilość dodatkowego feromonu na połączenia odwiedzane przez mrówki.

Ze względu na różne sposoby ustalania wielkości $\Delta\tau_{ij}$ wyróżniono trzy najczęściej występujące modele: model ze stałym poziomem, model z poziomem średnim (model natężenia feromonu) i model z poziomem cyklicznym. Pierwszy z nich zakłada, że k -ta mrówka po przejściu pomiędzy i -tym a j -tym punktem obsługi zostawia na nim pewną stałą ilość Q_1 feromonu:

$$\Delta\tau_{ij}^k(t, t+1) = \begin{cases} Q_1, & \text{jeżeli } (P_i, P_j) \in T \\ 0, & \text{jeżeli } (P_i, P_j) \notin T \end{cases} \quad (7.19)$$

Model z poziomem średnim natomiast zakłada, że ilość pozostawionego przez mrówkę feromonu Q_2 jest odwrotnie proporcjonalna do długości połączenia pomiędzy odwiedzanymi przez nią punktami obsługi P_i oraz P_j (7.20).

$$\Delta\tau_{ij}^k(t, t+1) = \begin{cases} \frac{Q_2}{d_{ij}}, & \text{jeżeli } (P_i, P_j) \in T \\ 0, & \text{jeżeli } (P_i, P_j) \notin T \end{cases} \quad (7.20)$$

Powyższe dwa modele są możliwe do zastosowania w wypadku algorytmu mrówkowego, w którym wszystkie mrówki jednocześnie przechodzą z jednego punktu obsługi do innego i na bieżąco pozostawiają swój ślad feromonowy, a opisanego pseudokodem zamieszczonym na rysunku 7.35. Natomiast ostatni z modeli – model z poziomem cyklicznym – jest charakterystyczny dla algorytmów kolonii mrówkowych i zakłada, że aktualizacja poziomu feromonu jest możliwa dopiero po zbudowaniu trasy przez mrówkę, a przyrost poziomu feromonu na połączeniu pomiędzy dwoma wskazanymi punktami obsługi dany jest wzorem (7.21):

$$\Delta\tau_{ij}^k(t, t+1) = \begin{cases} \frac{Q_3}{L^k}, & \text{jeżeli } (P_i, P_j) \in T \\ 0, & \text{jeżeli } (P_i, P_j) \notin T \end{cases} \quad (7.21)$$

gdzie:

L^k – długość całej trasy zbudowanej przez k -tą mrówkę.

Model ten w pewien sposób odpowiada modelowi ze stałym poziomem. W przeciwieństwie do niego poziom feromonu nakładany jest cyklicznie na wszystkie wykorzystane połączenia na końcu (po zbudowaniu trasy), a nie oddzielnie w każdym przejściu z punktu P_i do punktu P_j .

Ponadto w algorytmach kolonii mrówek dodatkowo wyróżnione zostały dwie reguły aktualizacji poziomu feromonu: reguła aktualizacji globalnej i reguła aktualizacji lokalnej (Boryczka, 2006, s. 37–39). Obie reguły są stosowane przez algorytm jednocześnie. Pierwsza z nich wykorzystuje wzór dla modelu poziomu cyklicznego (7.21) i ma zastosowanie w momencie, gdy wszystkie mrówki zbudują swoje trasy. Polega na umieszczeniu feromonu tylko na tych połączeniach pomiędzy punktami obsługi, które należą do trasy mrówki o najmniejszej całkowitej długości. Reguła ta może dotyczyć zarówno najlepszej dotychczas znalezionej trasy (od momentu rozpoczęcia działania algorytmu), jak i najlepszej znalezionej trasy w danej iteracji algorytmu.

Reguła lokalnej aktualizacji poziomu feromonu na poszczególnych połączeniach punktów należących do trasy mrówki zakłada niezwłoczne pozostawienie feromonu w ilości $\Delta\tau'_{ij}$ danej wzorem (7.22):

$$\Delta\tau'_{ij}(t) = (1 - \xi)\Delta\tau_{ij}(t) + \xi\tau_0 \quad (7.22)$$

gdzie:

$\Delta\tau_{ij}$ – przyrost poziomu feromonu na połączeniu pomiędzy i -tym a j -tym punktem obsługi obliczony zgodnie z formułą modelu z poziomem cyklicznym;

τ_0 – pewna stała ilość feromonu;

ξ – parametr algorytmu określający wagę należącą do przedziału $(0, 1)$.

Celem reguły lokalnej aktualizacji poziomu feromonu jest pewnego rodzaju zniechęcenie kolejnych mrówek do skorzystania z połączeń już odwiedzanych przez inne mrówki. Wielkość τ_0 wyznaczana jest w następujący sposób (7.23):

$$\Delta\tau_0 = \frac{1}{NL_{NN}} \quad (7.23)$$

gdzie:

N – liczba wszystkich punktów obsługi;

L_{NN} – długość trasy wyznaczonej przez zastosowanie heurystycznego algorytmu zachłannego – drogi do najbliższego sąsiada.

Przedstawione powyżej podstawowe algorytmy mrówkowe oraz algorytmy kolonii mrówek poddawane były wielu ulepszeniom w celu poprawy ich efektywności. Jedną z proponowanych modyfikacji jest stosowanie reguły elitaryzmu (Bullnheimer i in., 1999a). Jej ideą jest zwrócenie szczególnej uwagi na najlepszą znaną w danej iteracji algorytmu ścieżkę (trasę) mrówki. Trasa ta traktowana jest tak, jakby pewna liczba mrówek (elitarnych mrówek) ją wybrała. Ze względu na niemałe prawdopodobieństwo, że niektóre połączenia pomiędzy punktami obsługi tej trasy należeć mogą do rozwiązania optymalnego, celem tej reguły jest ich uwzględnienie w procesie przeszukiwania przestrzeni rozwiązań w kolejnych iteracjach algorytmu. Stąd też aktualizacja poziomu feromonu na połączeniach pomiędzy wybranymi punktami obsługi przyjmuje formułę (7.24):

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) + \Delta\tau_{ij}^*(t) \quad (7.24)$$

przy czym:

$$\Delta\tau_{ij}^*(t) = \begin{cases} \sigma \frac{Q}{L^*}, & \text{jeżeli } (P_i, P_j) \in T \\ 0, & \text{jeżeli } (P_i, P_j) \notin T \end{cases} \quad (7.25)$$

gdzie:

$\Delta\tau_{ij}^*$ – przyrost poziomu feromonu na połączeniu pomiędzy i -tym a j -tym punktem obsługi spowodowany przez elitarną mrówkę;

σ – liczba elitarnych mrówek;

L^* – długość najlepszej znalezionej trasy mrówki.

Liczba elitarnych mrówek jest kolejnym parametrem algorytmu, przy czym jego wartość ustalana jest w drodze przeprowadzanych eksperymentów obliczeniowych.

W literaturze przedmiotu można spotkać także inne podejścia korzystające z zasady elitaryzmu. Takim przykładem jest wykorzystanie systemu budowy rankingu mrówek, które znalazły najlepsze trasy. Jest to pewna modyfikacja elitaryzmu, która polega na uwzględnieniu nie jednej, najlepszej znalezionej trasy, a pewnej liczby ω najlepszych tras (liczba ω najlepszych tras stanowi następny parametr algorytmu kolonii mrówek). Po tym jak wszystkie mrówki wygenerują kompletne rozwiązania, są one sortowane w kolejności malejącej według całkowitej długości trasy. Natomiast wkład danej mrówki w aktualizację (przyrost) poziomu feromonu uzależniony jest od pozycji μ , jaką w rankingu zajmuje trasa przez nią pokonana. Im pokonana trasa jest krótsza (wyższa pozycja μ w rankingu), tym większy ślad (feromon) zostawia dana mrówka. Stąd też przyrost poziomu feromonu spowodowany jest tylko przez mrówkę należącą do zbioru mrówek elitarnych (ω najlepszych) i jest równy (7.26):

$$\Delta\tau_{ij}^{\mu}(t) = \begin{cases} (\sigma - \mu) \frac{Q}{L^{\mu}}, & \text{jeżeli } (P_i, P_j) \in T^{\mu} \\ 0, & \text{jeżeli } (P_i, P_j) \notin T^{\mu} \end{cases} \quad (7.26)$$

gdzie:

- $\Delta\tau_{ij}^{\mu}$ – przyrost poziomu feromonu na połączeniu pomiędzy i -tym a j -tym punktem obsługi spowodowany przez elitarną mrówkę zajmującą miejsce μ w zbudowanym rankingu;
- μ – indeks w zbudowanym rankingu;
- L^{μ} – długość trasy znalezionej przez mrówkę zajmującą miejsce μ w zbudowanym rankingu.

Innym przykładem modyfikacji klasycznej postaci algorytmów kolonii mrówek jest zastosowanie reguły Min-Max przedstawionej przez T. Stützlego i H. Hoosa (2000). Także w tej regule tylko jedna mrówka zostawia ślad feromonowy (reprezentująca najlepsze dotychczas znalezione rozwiązanie lub najlepsze rozwiązanie w danej iteracji), lecz ilość pozostawionego feromonu na danym połączeniu pomiędzy punktami obsługi P_i a P_j musi zawierać się w przedziale $(\tau_{\min}, \tau_{\max})$. Jeżeli przedział jest przekroczony w dół lub w górę, to przyjmowane są ilości feromonu na połączeniach odpowiednio: τ_{\min} i τ_{\max} . Wspomniani autorzy zwrócili też uwagę na sposób wyboru mrówki, która zostawiać będzie ślad feromonu. W początkowej fazie algorytmu powinna to być mrówka reprezentująca najlepszą trasę znaną w danej iteracji algorytmu, by stopniowo w kolejnych iteracjach algorytmu wybierać mrówkę, która przedstawia najlepszą dotychczas znaną trasę. Pozwala to w początkowej fazie algorytmu mrówkowego dokonywać szerszej eksploracji przestrzeni rozwiązań, by w końcowej jego fazie skupić się na lokalnym przeszukiwaniu wybranego jej fragmentu. Proponuje się także umieszczenie na wszystkich połączeniach

między punktami obsługi początkowej ilości feromonu równej τ_{\max} , aby dać równe szanse wszystkim połączeniom na to, że zostaną wybrane przez kolejne mrówki (zwiększenie możliwości eksploracyjnych algorytmu przez początkowe niewielkie różnice w atrakcyjności poszczególnych połączeń dla mrówek).

Algorytmy mrówkowe dla problemów jednego oraz wielu komiwojażerów są do siebie bardzo zbliżone. Wynika to z tego, że podstawowy algorytm mrówkowy zbudowany został właśnie dla zadania układania tras jednego pojazdu oraz może zostać w bardzo prosty sposób rozszerzony na zagadnienie optymalizacji tras wielu pojazdów. Takim pierwszym rozszerzeniem podstawowego algorytmu mrówkowego był algorytm zaproponowany przez B. Bullnheimera, R.F. Hartla i Ch. Strauss (1997) dla zadania układania tras wielu pojazdów z ograniczeniami zasobowymi w postaci maksymalnej ładowności pojazdu oraz maksymalnej dopuszczalnej długości jednej trasy. Tak jak w algorytmie mrówkowym dla problemu jednego komiwojażera, mrówka konstruuje trasę, dołączając poszczególne punkty obsługi do momentu, aż wszystkie zostaną przez nią odwiedzone. Za każdym razem, kiedy dołączenie kolejnego punktu obsługi powoduje naruszenie warunku ładowności pojazdu lub maksymalnej długości trasy, do budowanego przez mrówkę rozwiązania dołączane jest na końcu połączenie z bazą. Następnie rozpoczyna się nowa trasa pojazdu przez umieszczenie w tworzonym rozwiązaniu połączenia bazy z kolejnym punktem obsługi wybranym przez mrówkę. W celu określenia wielkości feromonu pozostawionego przez mrówkę na swojej ścieżce (trasie) wykorzystany został tutaj model poziomego cyklicznego, gdzie ilość pozostawionego feromonu jest odwrotnie proporcjonalna do sumy długości tras wszystkich użytych pojazdów: $1 / L^k$.

Zastosowano także strategię elitaryzmu polegającą na położeniu dodatkowej ilości $1 / L^k$ feromonu na połączeniach, które należą do układu tras stanowiącego dotychczas najlepsze znalezione rozwiązanie. Dodatkowy feromon składany jest tylko przez mrówki należące do σ -najlepszych w danej iteracji. Algorytm ten przedstawiono na rysunku 7.36. W fazie inicjującej zakłada się umieszczenie liczby N mrówek po jednej w każdym punkcie obsługi, skąd rozpoczynać będą

krok 1: inicjalizacja

repeat

repeat

 krok 2: konstrukcja tras pojazdów pojedynczej mrówki

 krok 3: lokalna optymalizacja tras pojazdów

until wszystkie mrówki

 krok 4: aktualizacja feromonu

until określona liczba iteracji

Rys. 7.36. Algorytm mrówkowy B. Bullnheimera, R.F. Hartla i Ch. Strauss
Źródło: opracowanie własne.

budowę swoich tras. W każdej iteracji generowane jest rozwiązanie dla każdej mrówki (kroki 2 i 3) oraz następuje aktualizacja poziomu feromonu na połączeniach pomiędzy punktami obsługi (krok 4). Algorytm wykonuje zadaną w kroku inicjalizacji liczbę iteracji. W kroku trzecim omawianego algorytm mrówkowego wykonywana jest procedura poprawy rozwiązania uzyskanego przez każdą mrówkę. Polega ona na zastosowaniu heurystycznej metody lokalnej optymalizacji 2-opt w odniesieniu do każdej trasy pojazdu oddzielnie.

Przedstawiony powyżej algorytm mrówkowy dla problemu wielu komiwojażerów podlegał z czasem stopniowym modyfikacjom mającym na celu poprawę jego efektywności. Jedną z takich modyfikacji motywowana jest tym, że w zagadnieniach układania tras wielu pojazdów istotną jest nie tylko wzajemna lokalizacja punktów obsługi w trasie pojazdu. Równie ważną jest bowiem także lokalizacja bazy w prezentowanym rozwiązaniu, a w szczególności połączenia baza a pierwszy obsługiwany przez nią punkt danej trasy pojazdu. W związku z tym zaproponowano wykorzystanie tzw. miary oszczędności, której idea została zaczerpnięta z heurystycznego algorytmu konstrukcyjnego oszczędnościowego łączenia tras G. Clarka i J.W. Wrighta. Wysoka wartość oszczędności s_{ij} może świadczyć o korzystnym dla procesu przeszukiwania przestrzeni rozwiązań umieszczeniu w jednej trasie j -tego punktu obsługi po i -tym punkcie obsługi. Idea ta wykorzystana została w algorytmie mrówkowym przez odpowiednią modyfikację preferencji dotyczącej wyboru przez mrówkę kolejnego połączenia do swojej trasy, a wyrażona we wzorze na jego prawdopodobieństwo (7.27) (Doerner i in., 2002):

$$P_{ij}^k = \frac{[\tau_{ij}(t)]^\alpha [s_{ij}]^\beta}{\sum_{l \in N_i^k} \{[\tau_{il}(t)]^\alpha [s_{il}]^\beta\}} \quad (7.27)$$

gdzie:

s_{ij} – wartość oszczędności z wykonania połączenia pomiędzy i -tym a j -tym punktem obsługi, gdzie $s_{ij} = d_{i0} + d_{0j} - d_{ij}$.

Nieco inna postać idei oszczędności przedstawiona jest w formie składnika parametru „widzialności” η_{ij} przez mrówkę ilości feromonu na poszczególnych połączeniach, która została nazwana parametryczną funkcją oszczędności (Bullnheimer i in., 1999b). Widzialność η_{ij} wyrażona została formułą:

$$\eta_{ij} = d_{i0} + d_{0j} - g d_{ij} + f |d_{i0} - d_{0j}| \quad (7.28)$$

gdzie:

f, g – parametry funkcji oszczędności stanowiące także parametry algorytmu⁴.

4 Autorzy podają znalezione najlepsze wartości dla f i g równe 2.

Obok oszczędności wynikających z umieszczenia w jednej trasie dwóch punktów obsługi drugą kwestią, na jaką zwrócono uwagę, uwzględniając specyfikę zagadnień układania tras wielu pojazdów, jest stopień wykorzystania ich ładowności. Otóż lepsze wykorzystanie ładowności przez pojazdy prowadzi do mniejszej liczby tras, którymi cechują się najlepsze rozwiązania. Oznaczając przez Q_i dotychczas wykorzystaną ładowność pojazdu, który znajduje się w i -tym punkcie obsługi, a przez q_j zapotrzebowanie (podaż) kolejnego możliwego do odwiedzenia j -tego punktu obsługi, stopień wykorzystania ładowności pojazdu κ_{ij} przedstawia się następująco:

$$\kappa_{ij} = \frac{Q_i + q_j}{Q} \quad (7.29)$$

Wysoka wartość stopnia wykorzystania ładowności pojazdu κ_{ij} prowadzić może do tego, że połączenie pomiędzy punktami P_i a P_j będzie chętniej wybierane. Ponadto w połączeniu z omawianą wcześniej wielkością oszczędności s_{ij} postać wzoru na prawdopodobieństwo wyboru przez mrówkę kolejnego połączenia do stopniowo budowanego rozwiązania jest następująca:

$$P_{ij}^k = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta [s_{ij}]^\gamma [\kappa_{ij}]^\lambda}{\sum_{l \in N_i^k} \{[\tau_{il}(t)]^\alpha [\eta_{il}]^\beta [s_{il}]^\gamma [\kappa_{il}]^\lambda\}} \quad (7.30)$$

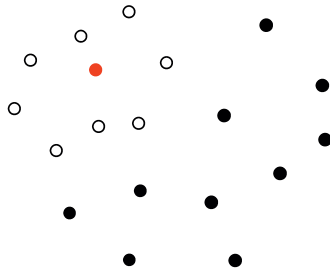
Modyfikacje wzoru na prawdopodobieństwo wyboru przez mrówkę kolejnego połączenia do trasy nie są jedynymi proponowanymi usprawnieniami algorytmów mrówkowych dla zagadnień układania tras jednego lub wielu pojazdów. Dotyczyć mogą także innych elementów algorytmu, jak np. procesu aktualizacji poziomu feromonu, a w szczególności jego odparowania. Taka propozycja dla zadania układania tras pojazdów z ograniczeniami zasobowymi przedstawiona została w pracy M.Y. Khoshbakhta i M. Sedighpoura (2011). Zazwyczaj współczynnik odparowania ρ jest wartością stałą, ustalaną raz w kroku inicjalizacyjnym algorytmu i dobieraną w drodze prób przy wielokrotnym jego uruchamianiu. Natomiast wspomniani autorzy proponują określenie współczynnika ρ_t jako funkcji trygonometrycznej, której argumentami są liczba punktów obsługi oraz numer kolejnej iteracji algorytmu, co oznacza, że wartość współczynnika ρ_t jest zmienna w czasie działania algorytmu.

$$\rho_t = 1 - \rho \cos \frac{\pi t}{3N} \quad (7.31)$$

gdzie:

- t – numer kolejnej iteracji algorytmu;
- N – liczba wszystkich punktów obsługi.

Podejście to ma na celu sterowanie procesem przeszukiwania przestrzeni rozwiązań. Jest ono motywowane spostrzeżeniami autorów, którzy stwierdzili, że w pierwszej fazie działania algorytmu ilości feromonu pozostające na poszczególnych połączeniach pomiędzy punktami obsługi powinny odparowywać znacznie wolniej, a tym samym proces poszukiwania nowych rozwiązań powinien być bardziej zrandomizowany. Z kolei wraz ze zwiększaniem się liczby wykonanych przez algorytm iteracji proces odparowania feromonu powinien przyspieszać.



Rys. 7.37. Lista kandydatów w algorytmie mrówkowym

Źródło: opracowanie własne.

Kolejnym przykładem modyfikacji klasycznej postaci algorytmu mrówkowego jest wprowadzenie tzw. listy kandydatów, która reprezentuje zbiór możliwych punktów obsługi, jakie w następnej kolejności mogą zostać dołączone przez mrówkę do trasy pojazdu. Parametrem algorytmu korzystającego z listy kandydatów jest liczebność tego zbioru (Mazzeo i Loiseau, 2004). Ponadto istotny jest także sposób tworzenia takiego zbioru kandydatów dla mrówki znajdującej się aktualnie w i -tym punkcie obsługi. Najczęściej zbiór ten stanowi pewna liczba r punktów obsługi (białe kropki na rys. 7.37), które znajdują się najbliżej punktu, gdzie aktualnie przebywa mrówka. Poza tym do tego zbioru nie są oczywiście wliczane punkty obsługi już przez pojazd odwiedzone. Takie podejście wydaje się szczególnie uzasadnione dla problemów układania tras pojazdów o dużych rozmiarach, gdzie wskazane jest ograniczenie wysiłków skierowanych w stronę budowy tras składających się z ponadprzeciętnie długich połączeń. Kwestią otwartą jednak pozostaje określenie progu, dla którego dane połączenie pomiędzy punktami obsługi uznane zostanie za zbyt odległe.

Interesującą ideę zaimplementowaną w algorytmie mrówkowym dla problemu wielu komiwojazerów przedstawiono w pracy J.E. Bella i P. McMullena (2004), w której wykorzystano naturalne zjawisko podziału zadań wśród poszczególnych osobników w kolonii mrówek. Zauważono, że w środowisku naturalnym poszczególne zespoły mrówek mają przydzielone konkretne zadania. Część z nich realizuje funkcję poszukiwania jedzenia, część buduje mrowisko, a jeszcze inne pełnią funkcję obronną. W związku z tym autorzy postanowili podzielić kolonię mrówek na pewną liczbę osobnych kolonii mrówek, w ramach których mrówki składają

unikalny feromon. Oznacza to, że mrówki, dokonując wyboru połączeń do swojej budowanej trasy, kierują się tylko feromonem, który potrafią same zidentyfikować. Liczba kolonii jest adekwatna do liczby pojazdów mających za zadanie odwiedzić wszystkie punkty obsługi, a do danego pojazdu przyporządkowana jest konkretna kolonia mrówek. Pozwala to na skoncentrowanie wysiłków algorytmu na modyfikacji już znalezionych dobrych tras pojazdów bez dokonywania zbyt dużych zmian pomiędzy nimi.

Nowy komponent do zaproponowanego przez siebie algorytmu mrówkowego wprowadzony został przez B. Yu, Z.-Z. Yanga i B. Yao (2009). Jest to operacja mutacji, charakterystyczna zwłaszcza dla omawianych w kolejnym podrozdziale algorytmów ewolucyjnych. Przedstawiony przez wspomnianych autorów algorytm mrówkowy ma postać (rys. 7.38):

```

krok 1: inicjalizacja
repeat
    krok 2: konstrukcja tras pojazdów
    krok 3: mutacja
    krok 4: lokalna optymalizacja
    krok 5: aktualizacja feromonu
until określona liczba iteracji
  
```

Rys. 7.38. Algorytm mrówkowy B. Yu, Z.-Z. Yanga i B. Yao

Źródło: opracowanie własne.

Przedstawiona operacja mutacji polega na losowym wskazaniu w każdej iteracji algorytmu dwóch różnych tras, a w następnej kolejności po jednym z punktów obsługi w każdej z nich. Wskazane punkty obsługi są zamieniane miejscami w swoich trasach. Tak poddane modyfikacji poszczególne trasy są dodatkowo poprawiane z użyciem metody heurystycznej 2-opt. Operacja mutacji dokonywana jest z określonym prawdopodobieństwem p_m , którego wartość jest na bieżąco dostosowywana w trakcie działania algorytmu (jest funkcją liczby wykonanych dotychczas iteracji). Jest ono mniejsze w początkowej fazie działania algorytmu (uzyskanie szybszej zbieżności do dobrych rozwiązań), potem zaś stopniowo się zwiększa (możliwość unikania lokalnych optimum). W omawianym algorytmie mrówkowym zastosowano także inny sposób obliczania wielkości nowego feromonu pozostawionego przez mrówkę. Jest on wyrażony wzorem:

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{KL} \cdot \frac{D^k - d_{ij}}{m^k D^k}, & \text{jeżeli } (P_i, P_j) \in T_k \\ 0, & \text{jeżeli } (P_i, P_j) \notin T_k \end{cases} \quad (7.32)$$

gdzie:

L – całkowita długość wszystkich tras;

D^k – długość k -tej trasy w rozwiązaniu;

m^k – liczba punktów obsługi k -tej trasy w rozwiązaniu ($m^k > 0$).

Ponadto zastosowano omawiane już wcześniej dolną i górną granicę poziomu feromonu, jaki może być na połączeniu pomiędzy punktami P_i a P_j (7.33)–(7.34), co ma także minimalizować ryzyko zbyt szybkiej zbieżności algorytmu do minimum lokalnego.

$$\tau_{\min.} = \frac{Q}{\sum_i 2d_{oi}} \quad (7.33)$$

$$\tau_{\max.} = \frac{Q}{\sum_i d_{oi}} \quad (7.34)$$

Podobny algorytm został zaproponowany przez X. Zhanga i J. Wanga (2012). Różnica dotyczy przede wszystkim sposobu aktualizacji poziomu feromonu. Nowa ilość feromonu znajdująca się na połączeniach obliczana jest w sposób charakterystyczny dla klasycznych algorytmów mrówkowych. Natomiast początkowa ilość feromonu τ_0 ustalona została na podstawie rozwiązania uzyskanego za pomocą heurystyki konstrukcyjnej drogi do najbliższego sąsiada: $\tau_0 = (mL_{NN})^{-1}$. Także inaczej określone są tutaj granice minimalnego i maksymalnego poziomu feromonu na połączeniach i wynoszą one odpowiednio:

$$\tau_{\min.} = \frac{\tau_{\max.}}{5} \quad (7.35)$$

$$\tau_{\max.} = \frac{1}{2(1-\rho)} \frac{1}{L(T^*)} \quad (7.36)$$

7.2.2. Algorytmy ewolucyjne

Algorytmy ewolucyjne dokonują przeszukiwania przestrzeni rozwiązań, symulując występujące w naturze mechanizmy doboru naturalnego oraz dziedziczności. Działanie algorytmów ewolucyjnych polega na symulacji procesów istotnych dla ewolucji. W przyrodzie jednostki z pewnej grupy żywych osobników rywalizują ze sobą o ogólnie dostępne zasoby. Rywalizują również o względy partnerów. Jednostki, które osiągną najlepsze rezultaty w obu wypadkach, mają najwięcej szans nie tylko na to, aby przetrwać, ale również spodziewać się mogą największej liczby potomków. Oznacza to, że cechy najlepszych jednostek będą się rozprzestrzeniały o wiele szybciej niż

w wypadku słabszych jednostek. Brak umiejętności przystosowania oznacza brak potomstwa, a tym samym brak możliwości przekazania swoich cech.

Do podstawowych pojęć używanych w terminologii algorytmów ewolucyjnych zaliczane są m.in.: „chromosom”, „gen”, „populacja”, „pokolenie”, „funkcja przystosowania”, „reprodukcja”, „sukcesja” (Cytowski, 1996; Goldberg, 2003). Jednym z najważniejszych elementów algorytmów ewolucyjnych jest kodowanie rzeczywistych rozwiązań w postaci chromosomu. Innymi słowy, chromosom odpowiadający w środowisku naturalnym osobnikowi to ciąg kodowy, w którego postaci przedstawiony jest każdy element przestrzeni rozwiązań problemu optymalizacyjnego. Nie zawsze możliwe jest przedstawienie każdego punktu przestrzeni rozwiązań w postaci jednego chromosomu. Gdy rozwiązanie reprezentowane jest przez więcej niż jeden chromosom, wtedy mówimy o genotypie. Każdy najmniejszy niepodzielny element chromosomu stanowi gen, który przyjmuje wartości w zależności od ustalonego sposobu kodowania rozwiązań rzeczywistych. Zbiór ocenianych w danym momencie rozwiązań stanowi populację.

Funkcja przystosowania jest to funkcja, która pozwala na ocenę chromosomów w populacji. Czasami używane jest tutaj zamiennie pojęcie funkcji dopasowania lub funkcji oceny. Dzięki niej konsekwentnie wskazywać można te osobniki, które reprezentują nowe, coraz lepsze punkty przestrzeni rozwiązań. Funkcja przystosowania, która swą nazwę również bierze bezpośrednio z nauk przyrodniczych, odpowiada za realizację podstawowej zasady ewolucji: „przetrwają tylko najsilniejsi”. W zależności od rozpatrywanego problemu funkcja przystosowania może przyjmować różne postaci. W zagadnieniach optymalizacyjnych, w tym zadaniach ustalania tras dla pojazdów, funkcja oceny budowana jest bezpośrednio zazwyczaj na podstawie funkcji celu, czasami uwzględniając pewne dodatkowe elementy służące dyskryminacji lub faworyzowaniu pewnych rozwiązań.

```

krok 1: inicjalizacja
repeat
    krok 2: reprodukcja
    krok 3: operatory genetyczne
        krok 3a: krzyżowanie
        krok 3b: mutacja
    krok 4: sukcesja
repeat określona liczba pokoleń
  
```

Rys. 7.39. Algorytm ewolucyjny
Źródło: opracowanie własne.

Algorytm ewolucyjny jest procesem iteracyjnym (rys. 7.39), w którym wykonanie wszystkich kroków wchodzących w jego skład nosi nazwę pokolenia. Głównymi etapami każdej iteracji algorytmu (pokolenia) są: reprodukcja, operatory genetyczne oraz sukcesja, które powodują przekształcenie bieżącej populacji (aktualnie

rozpatrywanego zbioru rozwiązań) w populację nowego pokolenia (zbiór nowych rozwiązań problemu optymalizacyjnego). W drugim kroku algorytmu ewolucyjnego dokonywany jest proces reprodukcji, gdzie spośród chromosomów (rozwiązań) wybierane są na podstawie funkcji przystosowania zazwyczaj te najlepsze, które tworzą tzw. populację rodzicielską. W kolejnym, trzecim kroku chromosomy populacji rodzicielskiej poddawane są działaniu operatorów genetycznych, do których najczęściej należą krzyżowanie oraz mutacja. Krzyżowanie zachodzi, jeżeli powstanie nowy osobnik (chromosom) w wyniku zaangażowania do tego procesu przynajmniej dwóch chromosomów z populacji rodzicielskiej. Rolą krzyżowania jest dokonanie wymiany informacji zawartej w każdym wykorzystanym w tym celu chromosomie. Z kolei mutacja dotyczy modyfikacji jednego chromosomu z populacji rodzicielskiej lub z populacji utworzonej w wyniku przeprowadzenia krzyżowania, a jej głównym celem jest dywersyfikacja procesu przeszukiwania przestrzeni rozwiązań.

Wreszcie ostatnim krokiem algorytmu ewolucyjnego jest przeprowadzenie sukcesji, czyli utworzenie nowej populacji chromosomów, która ponownie zostanie poddana ocenie w kolejnej iteracji (pokoleniu). Utworzenie nowej populacji chromosomów powstałych w wyniku krzyżowania i mutacji nie musi być równoważne z ich sukcesją do nowego pokolenia. Bardzo często nowe pokolenie chromosomów tworzone jest jako zbiór składający się z chromosomów powstałych w wyniku działania operatorów genetycznych oraz z najlepszych chromosomów znalezionych w ostatniej lub pewnej liczbie ostatnich pokoleń.

Kolejną cechą charakterystyczną dla algorytmów ewolucyjnych jest wysoki stopień losowości opisanych wyżej procesów. Zarówno selekcja, krzyżowanie, jak i mutacja mogą zostać przeprowadzone z pewnymi prawdopodobieństwami stanowiącymi bardzo istotne parametry budowanego algorytmu ewolucyjnego. W wypadku selekcji oznacza to, że nie zawsze najlepszy chromosom (np. reprezentujący najlepsze znalezione rozwiązanie w populacji) musi zostać wybrany do populacji, która poddana zostanie procesom krzyżowania (populacji rodzicielskiej). Może wystąpić nawet sytuacja, w której najlepszy chromosom nie zostanie przez algorytm uwzględniony, a tym samym dalej zostanie „zapomniany”, jeżeli w odpowiedni sposób nie zbuduje się w kroku czwartym procedury sukcesji. Losowość charakteryzuje także krzyżowanie i mutację. Przy czym w wypadku tego drugiego operatora genetycznego prawdopodobieństwo jego wystąpienia często jest znacznie mniejsze, ponieważ mutacja dotyczy genu, a nie całego osobnika.

Szczególnym przypadkiem algorytmów ewolucyjnych są algorytmy genetyczne. Tym mianem określa się często algorytmy ewolucyjne realizujące najprostszy schemat: populacja wyjściowa – selekcja – populacja rodzicielska – krzyżowanie – mutacja – nowa populacja. W algorytmach ewolucyjnych stanowiących znacznie szerszą klasę omawianych metaheurystyk niektóre elementy (np. krzyżowanie) mogą zostać pominięte, inne natomiast dodane (np. procedury lokalnej optymalizacji, czyli innych algorytmów heurystycznych wykorzystywanych w celu lokalnej poprawy uzyskanego dotychczas rozwiązania) (Michalewicz, 1999, s. 83).

Pierwszą, a jednocześnie najważniejszą czynnością, od której należy rozpocząć budowanie algorytmów ewolucyjnych, jest ustalenie sposobu zakodowania w postaci chromosomów potencjalnych rozwiązań zadania. To sposób kodowania rzeczywistych rozwiązań problemu optymalizacyjnego stanowi najbardziej charakterystyczną cechę danego algorytmu ewolucyjnego przeznaczonego do rozwiązywania problemów jednego lub wielu komiwojażerów. Drugim takim elementem, ale wynikającym ze sposobu kodowania, jest odpowiednia konstrukcja operatorów genetycznych (krzyżowania i mutacji).

Każdy sposób kodowania rzeczywistego rozwiązania w postaci chromosomu powinien spełniać dwa podstawowe warunki. Pierwszy z nich mówi, że kod powinien być odporny na przekształcenia jednego chromosomu w drugi, co oznacza, że nowo powstały chromosom powinien dać się odkodować, aby można było ustalić rzeczywistą postać rozwiązania, którą on reprezentuje, a dalej obliczyć funkcję przystosowania. Z kolei drugi z warunków stanowi, że niewielkie, lecz spójne fragmenty chromosomu (pewne podciągi genów) powinny reprezentować konkretne cechy, jakimi charakteryzują się rozwiązania. Sposoby kodowania chromosomów dzielone są ze względu na dwa kryteria: sposób ułożenia genów w chromosomie oraz wartości, jakie przyjmują geny.

Jednym z najbardziej znanych sposobów kodowania (szczególnie charakterystycznych dla klasycznych algorytmów genetycznych) jest przyjęcie alfabetu binarnego. Innymi słowy, chromosom składa się z ciągu zer i jedynek. Przykładową liczbę 615 zakodować można w postaci chromosomu składającego się z dziesięciu genów: [1 0 0 1 1 0 0 1 1 1], ponieważ:

$$1 \times 2^9 + 0 \times 2^8 + 0 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 615$$

Dla zagadnień układania tras pojazdów wykorzystanie reprezentacji binarnej nie jest naturalnym i najchętniej stosowanym sposobem kodowania, choć możliwym do zastosowania. Przyporządkowując kolejne liczby naturalne punktom obsługi, można przedstawić je w postaci wektora zer i jedynek (tab. 7.3).

Tabela 7.3. Przykład sposobu kodowania binarnego dla zadania układania tras pojazdów

Punkt obsługi	Liczba naturalna	Wartość binarna
P_1	1	001
P_2	2	010
P_3	3	011
P_4	4	100
P_5	5	101

Źródło: opracowanie własne.

Dla trasy jednego komiwojażera $P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4 \rightarrow P_5$ chromosom reprezentujący to rozwiązanie miałby postać: [001 010 011 100 101] (Larrañaga i in., 1999). Każdy element chromosomu odpowiadający danemu punktowi składa się z trzech genów, więc każda kombinacja zer i jedynek reprezentuje konkretny punkt. Jednakże nie są wykorzystane trzy spośród ośmiu możliwych układów zer i jedynek: 000, 110 oraz 111, gdzie np. 110 mogłoby odpowiadać punktowi P_6 , a 111 – P_7 , pod warunkiem, że punkty istnieją. Oznacza to, że w wyniku działania algorytmu genetycznego mogłyby się pojawić chromosomy, które nie mają swoich odpowiedników w rzeczywistej przestrzeni rozwiązań zadania (nie są odporne).

Innym przykładem kodowania tras pojazdów opartego na alfabecie binarnym jest sposób przedstawiony przez D. Whitleya i in. (1991), w którym punktem wyjścia jest zdefiniowanie listy wszystkich możliwych połączeń pomiędzy punktami obsługi. Liczba genów w chromosomie odzwierciedla liczbę tych połączeń. Gen przyjmie wartość 1, jeżeli dane połączenie wchodzi w skład trasy. Dla wyżej przedstawionego przykładu z pięcioma punktami obsługi lista wszystkich połączeń jest następująca:

P_1 : $P_1 \rightarrow P_2; P_1 \rightarrow P_3; P_1 \rightarrow P_4; P_1 \rightarrow P_5$;

P_2 : $P_2 \rightarrow P_3; P_2 \rightarrow P_4; P_2 \rightarrow P_5$;

P_3 : $P_3 \rightarrow P_4; P_3 \rightarrow P_5$;

P_4 : $P_4 \rightarrow P_5$.

Ponieważ liczba możliwych krawędzi wynosi 10, chromosom składać się będzie z 10 genów. Tym samym przykładowa trasa $P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4 \rightarrow P_5$ będzie miała następującą postać chromosomu: [1 0 0 0 1 0 0 1 0 1]. Niestety inny układ zer i jedynek wyznaczać może rozwiązanie niedopuszczalne, lecz w przeciwieństwie do poprzedniego sposobu kodowania wartości genów w chromosomie nie wskazują na połączenia, które w ogóle nie istnieją.

O wiele bardziej przydatnym alfabetem wykorzystywanym do kodowania tras pojazdów w postaci chromosomów jest alfabet liczb naturalnych (Potvin, 1996). Podkreśla on zwłaszcza homogeniczność poszczególnych genów, czyli ich rozróżnialność w odniesieniu do konkretnych punktów obsługi znajdujących się w trasie pojazdu. Chromosomy zbudowane na podstawie tego alfabetu stanowić mogą permutację pewnego zbioru liczb naturalnych odzwierciedlających wprost zbiór wszystkich punktów obsługi.

Jednym z najbardziej naturalnych sposobów reprezentacji tras pojazdów w postaci chromosomów z wykorzystaniem alfabetu liczb naturalnych jest kodowanie ścieżkowe. Geny, które reprezentują punkty obsługi, przez swoje ułożenie w chromosomie określają kolejność, w jakiej mają one być odwiedzone. Na przykład

trasa: $P_1 \rightarrow P_3 \rightarrow P_2 \rightarrow P_5 \rightarrow P_6 \rightarrow P_4 \rightarrow P_7 \rightarrow P_9 \rightarrow P_8$ może być reprezentowana przez chromosom: [1 3 2 5 6 4 7 9 8]. Warto tutaj zauważyć, że w wypadku kodowania trasy pojazdu w symetrycznym problemie jednego komiwojażera trasa ta może być reprezentowana przez $2(N+1)$ liczbę różnych chromosomów, ponieważ na pozycji pierwszej może się znaleźć dowolny punkt obsługi, natomiast kolejność pozostaje bez zmian: [2 5 6 4 7 9 8 1 3].

Drugim sposobem kodowania tras pojazdów do postaci chromosom z wykorzystaniem alfabetu liczb naturalnych jest kodowanie przyległościowe. Podobnie jak w kodowaniu ścieżkowym, chromosom składa się z genów w liczbie równej liczbie obsługiwanych punktów. Różnica wynika z pozycji konkretnego genu w ciągu kodowym. Gen odzwierciedlający obsługiwany punkt obsługi P_i znajduje się na pozycji k w chromosomie, jeżeli w trasie pojazdu punkt obsługi P_i obsługiwany jest w następnej kolejności po punkcie P_k . W wypadku zadania układania tras pojazdów oznacza to wystąpienie połączenia $P_k \rightarrow P_i$. Dla przykładowej trasy pojazdu: $P_1 \rightarrow P_6 \rightarrow P_3 \rightarrow P_7 \rightarrow P_5 \rightarrow P_2 \rightarrow P_4 \rightarrow P_9 \rightarrow P_8$ odpowiadający jej ciąg kodowy chromosomu jest następujący: [6 4 7 9 2 3 5 1 8]. Szósty gen przedstawionego powyżej chromosomu, którego wartość jest równa 3, reprezentuje punkt obsługi P_3 . W przedstawionej przykładowej trasie istnieje połączenie $P_6 \rightarrow P_3$. Podobnie gen chromosomu o wartości 4, czyli punkt obsługi P_4 , zajmuje drugą pozycję w chromosomie, ponieważ w trasie znajduje się połączenie pomiędzy punktami obsługi $P_2 \rightarrow P_4$. Ta metoda kodowania tras pojazdów w postaci chromosomów kładzie nacisk przede wszystkim na występujące w trasach połączenia pomiędzy punktami obsługi, co bezpośrednio zwraca na nie uwagę przy dokonywaniu operacji tworzenia nowych lub modyfikacji istniejących chromosomów. Tutaj z kolei każdej trasie symetrycznego problemu komiwojażera odpowiadać mogą tylko dwa różne chromosomy.

Trzecim sposobem kodowania chromosomów opartym na alfabetcie liczb naturalnych jest kodowanie porządkowe (Grefenstette i in., 1985). Sposób ten wykorzystuje do budowy chromosomu (jak również potem do jego odkodowania) pewne wzorcowe uporządkowanie. Jest to ciąg bazowy będący listą obsługiwanych przez pojazd (lub pojazdy) punktów. Jeżeli przyjętym ciągiem bazowym jest lista wszystkich obsługiwanych punktów: [$P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9$], to proces budowy chromosomu dla przykładowej trasy $P_1 \rightarrow P_6 \rightarrow P_3 \rightarrow P_8 \rightarrow P_5 \rightarrow P_2 \rightarrow P_9 \rightarrow P_7 \rightarrow P_4$ przedstawić można w następujący sposób. Chromosom składać się będzie z liczby genów odpowiadającej liczbie obsługiwanych punktów.

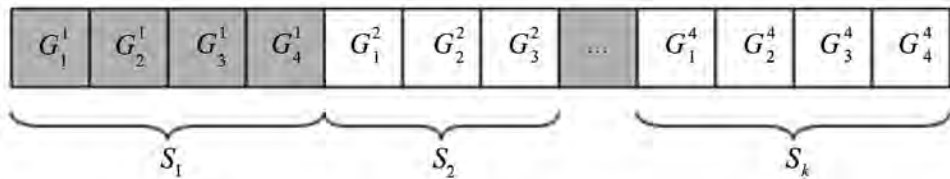
$P1:$	$[P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9]$	$[1 \times \times \times \times \times \times \times \times]$
$P6:$	$[P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9]$	$[1 5 \times \times \times \times \times \times]$
$P3:$	$[P_2, P_3, P_4, P_5, P_7, P_8, P_9]$	$[1 5 2 \times \times \times \times \times]$
$P8:$	$[P_2, P_4, P_5, P_7, P_8, P_9]$	$[1 5 2 5 \times \times \times \times \times]$
$P5:$	$[P_2, P_4, P_5, P_7, P_9]$	$[1 5 2 5 3 \times \times \times \times]$
$P2:$	$[P_2, P_4, P_7, P_9]$	$[1 5 2 5 3 1 \times \times \times]$
$P9:$	$[P_4, P_7, P_9]$	$[1 5 2 5 3 1 3 \times \times]$
$P7:$	$[P_4, P_7]$	$[1 5 2 5 3 1 3 2 \times]$
$P4:$	$[P_4]$	$[1 5 2 5 3 1 3 2 1]$

Budowa chromosomu następuje sekwencyjnie, w każdym kroku po ustaleniu danego genu usuwany jest z ciągu bazowego odpowiadający mu punkt obsługi. Pierwszym punktem w trasie jest P_1 , który w ciągu bazowym jest na pierwszym miejscu. Stąd pierwszym genem chromosomu będzie 1, natomiast punkt P_1 zostaje usunięty z ciągu bazowego jako już przyporządkowany. Kolejny punkt trasy to P_6 , który w aktualnym ciągu bazowym zajmuje piąte miejsce, więc wartość drugiego genu chromosomu wynosi 5. Po usunięciu punktu P_6 z ciągu bazowego składać się on będzie już z siedmiu elementów – na drugim miejscu widzimy punkt obsługi P_3 , czyli gen trzeci chromosomu przyjmie wartość 2. Analogiczne postępowanie dla pozostałych punktów obsługi występujących w przykładowej trasie prowadzi do uzyskania końcowej postaci chromosomu: $[1 5 2 5 3 1 3 2 1]$.

Przedstawione powyżej sposoby kodowania rzeczywistych rozwiązań mogą być wykorzystane zarówno do problemu jednego, jak i wielu komiwojazerów. W algorytmie ewolucyjnym PEWKOM (Jadczak, 2005) przeznaczonym do rozwiązania zadania układania tras pojazdów z ograniczeniami zasobowymi wykorzystana została reprezentacja ścieżkowa, która pozwala na bezpośrednie ustalenie kolejności obsługiwanych punktów przez pojazdy. Natomiast aby przyporządkować poszczególne punkty do pojazdów, chromosom dzieli się na części, których wyznaczenie następuje na podstawie możliwości przewozowych środków transportu. Każdy chromosom składa się z podciągów genów reprezentujących poszczególne trasy pojazdów. Gdy oznaczymy przez G_i^k i -ty kolejny gen chromosomu przyporządkowany do k -tego pojazdu, postać chromosomu dla zadania, w którym N punktów obsługi zostanie odwiedzone przez K pojazdów, będzie następująca (rys. 7.40).

Przydział poszczególnych genów (punktów obsługi obsługiwanych tras) do pojazdów odbywa się, począwszy od pierwszego genu chromosomu. Natomiast wybór pojazdu, który będzie odwiedzał kolejne punkty, uwarunkowany jest jego ładownością. W wypadku dysponowania pojazdami o różnej ładowności, w pierwszej kolejności wybierane są do obsługi punktów pojazdy, których ładowność

jest największa. Zgodnie z przedstawionym schematem (rys. 7.40) pierwsza część chromosomu (podciąg genów: $G_1^1, G_2^1, \dots, G_4^1$) przyporządkowana jest do pierwszego pojazdu S_1 (o największej ładowności spośród wszystkich pojazdów S_1, S_2, \dots, S_k), ponieważ jego ładowność nie pozwala na obsłużenie kolejnego punktu reprezentowanego przez gen G_1^2 . W związku z tym gen G_1^2 wejdzie w skład drugiej części chromosomu: G_1^2, G_2^2, G_3^2 , która przyporządkowana będzie do drugiego pojazdu S_2 . Analogicznie nastąpi przyporządkowanie pozostałych genów do kolejnych części chromosomu odpowiadających pozostałym użytym do obsługi wszystkich tras pojazdom.



Rys. 7.40. Kodowanie ścieżkowe w algorytmie PEWKOM
Źródło: (Jadczak, 2005).

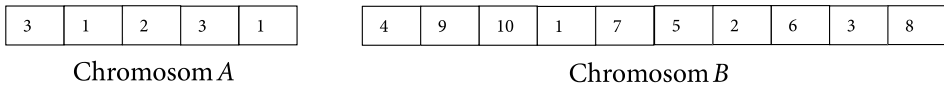
Nieco podobny sposób kodowania tras pojazdów do postaci chromosomu przedstawiony został przez P. Machado, J. Tavaresa, F.B. Pereirę oraz E. Costę (2002) i nazywany przez nich reprezentacją GVR (*Genetic Vehicle Representation*). W reprezentacji GVR każda trasa pojazdu posiada swój oddzielny chromosom, czyli genotyp osobnika stanowi zbiór chromosomów równy liczbie wykorzystanych w danym rozwiązaniu pojazdów. W sytuacji gdy powstałby nowy osobnik, gdzie jeden z chromosomów zawierałby geny reprezentujące punkty obsługi, dla których łączny popyt (podaż) przekracza dopuszczalną ładowność, trasa ta byłaby dzielona na odrębne trasy, tworząc tym samym odrębne nowe chromosomy osobnika (rys. 7.41).



Rys. 7.41. Reprezentacja GVR
Źródło: (Machado i in., 2002).

Ci sami autorzy zaproponowali także odmienny sposób reprezentacji potencjalnych rozwiązań dla zadania układania tras wielu pojazdów (Machado i in., 2002).

W przedstawionej przez nich propozycji kodowania kolejność obsługi oraz przy-
porządkowanie obsługiwanych punktów do pojazdów rozpatrywane są oddzielnie.
Ów algorytm ewolucyjny operuje na genotypie, w którym osobnik reprezentujący
rzeczywiste rozwiązanie składa się z dwóch chromosomów (rys. 7.42).



Rys. 7.42. Kodowanie za pomocą dwóch chromosomów

Źródło: (Machado i in., 2002).

Pierwszy z nich przedstawia informację o tym, ile obsługiwanych punktów
wchodzi w skład danego pojazdu. Natomiast w drugim chromosomie zakodowana
jest informacja o kolejności ich obsługi. Pierwszy gen chromosomu A oznacza, że
do trasy pierwszego pojazdu należą trzy punkty, których obsługa nastąpi w kolej-
ności wyznaczonej przez pierwsze trzy geny chromosomu B [4 9 10], należącego
do drugiej populacji. Z kolei drugi gen chromosomu A wskazuje na umieszczenie
w trasie drugiego pojazdu punktu reprezentowanego przez jeden gen chromoso-
mu B: [1], i dalej w kolejnych trasach odpowiednio: [7 5], [2 6 3] i [8]. Tym samym
uzyskano pięć tras pojazdów:

$$P_0 \rightarrow P_4 \rightarrow P_9 \rightarrow P_{10} \rightarrow P_0;$$

$$P_0 \rightarrow P_1 \rightarrow P_0;$$

$$P_0 \rightarrow P_7 \rightarrow P_5 \rightarrow P_0;$$

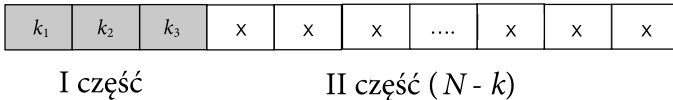
$$P_0 \rightarrow P_2 \rightarrow P_6 \rightarrow P_3 \rightarrow P_0;$$

$$P_0 \rightarrow P_8 \rightarrow P_0.$$

O ile w wypadku chromosomów obrazujących kolejność obsługi poszczegól-
nych punktów przez pojazdy nie ma wątpliwości co do liczby ich genów (zawsze
będzie ich tyle, ile jest punktów obsługi w rejonie działania bazy), to w wypad-
ku chromosomów A sytuacja wygląda inaczej. Liczba genów w chromosomie
A równa jest liczbie wszystkich obsługiwanych punktów podzielonej przez dwa.
W ten sam sposób ustalona została również maksymalna wartość każdego genu
tego chromosomu. Każda trasa może składać się z co najwyżej $N / 2$ punktów
obsługi. Przedstawiona reprezentacja zbioru tras pojazdów prowadzić może do
uzyskiwania rozwiązań niedopuszczalnych. W skład trasy danego pojazdu mogą
wchodzić punkty obsługi, których łączna podaż (lub popyt) może przekraczać ła-
downość użytego samochodu. W takiej sytuacji autorzy proponują dokonanie po-
działu danej trasy na co najmniej dwie części, aby zawsze można było uzyskiwać
rozwiązania dopuszczalne.

Kolejną propozycją kodowania tras pojazdów jest sposób przedstawiony przez takich autorów, jak S.J. Louis, X. Yin i Z.Y. Yuan (1999). Budowę chromosomu oparto na alfabecie liczb naturalnych i składa się on z tylu genów, ile jest obsługiwanych przez bazę punktów, które podzielone są na dwie części (rys. 7.43). Pierwsza część obejmuje geny w liczbie równej liczbie pojazdów, jakimi dysponuje baza. Każdy gen tej części reprezentuje pierwszy obsługiwany punkt w trasie danego pojazdu. Na przykład pierwszy gen k_1 oznacza pierwszy punkt obsługi trasy dla pierwszego pojazdu, gen drugi wyznacza pierwszy obsługiwany punkt trasy drugiego pojazdu itd. Zakładając, że baza dysponuje trzema pojazdami, liczba genów pierwszej części chromosomu wynosi 3. Pozostałe geny chromosomu przedstawiają punkty obsługi, których przyporządkowanie do istniejących tras następuje w pewien stały i powtarzalny sposób. Każdy z tych elementów jest rozpatrywany indywidualnie. Biorąc pierwszy gen z drugiej części chromosomu, należy rozpatrzyć wszystkie możliwe rozbudowy już istniejących tras i wybrać najlepszą kombinację (najlepsze miejsce wstawienia punktu do jednej z tras). Po dodaniu rozpatrywanego punktu do trasy jednego z pojazdów należy czynność tę powtórzyć dla kolejnych genów chromosomu do momentu, kiedy wszystkie punkty zostaną przyporządkowane do pojazdów.

Z tak zaproponowanego kodowania chromosomu bezpośrednio wynika jedynie początkowy kształt tras, zawierających tylko po jednym punkcie obsługi. Dalszy rozwój tras uwarunkowany jest przez deterministyczne przyporządkowanie kolejnych punktów do pojazdów, gdzie pozycja genu w chromosomie wyznacza tylko, który punkt obsługi ma być w danym momencie rozpatrywany.



Rys. 7.43. Kodowanie w algorytmie ewolucyjnym S.J. Louisa, X. Yina i Z.Y. Yuana
Źródło: (Louis i in., 1999).

Obok kodowania rzeczywistych tras pojazdów w postaci chromosomów równie ważnym elementem każdego algorytmu ewolucyjnego jest określenie operatorów genetycznych, dzięki którym konstruowane są nowe rozwiązania. Sposoby krzyżowania oraz mutacji chromosomów są ściśle związane z przyjętą przez chromosom reprezentacją rozwiązania. W wypadku kodowania opartego na alfabecie binarnym, który jest charakterystyczny w szczególności dla klasycznych algorytmów genetycznych, krzyżowanie dwóch (lub większej liczby) osobników⁵ może się odbywać przez wzajemną wymianę losowo wybranych jednego lub kilku podciągów genów pomiędzy osobnikami:

⁵ W terminologii algorytmów ewolucyjnych osobniki (w szczególności pojedyncze chromosomy) podlegające krzyżowaniu nazywane są rodzicami, natomiast nowe, powstałe w wyniku przeprowadzenia tej operacji, noszą nazwę potomków.

Chromosom A: [1 0 0 1 1 0 0 1 1 1] \Rightarrow Chromosom A': [1 0 0 1 1 0 1 1 0 0]

Chromosom B: [1 1 0 0 1 0 1 1 0 0] \Rightarrow Chromosom B': [1 1 0 0 1 0 0 1 1 1]

Z kolei mutacja dotycząca zmiany postaci jednego chromosomu może polegać na zmianie wartości losowo wybranego jednego lub kilku genów:

Chromosom A: [1 0 0 1 1 0 0 1 1 1] \Rightarrow Chromosom A': [1 1 0 1 1 1 1 1 0 0]

Łatwo zauważyć, że przy kodowaniu opartym na alfabecie liczb naturalnych, gdzie poszczególne chromosomy stanowią permutację genów (kodowanie ścieżkowe i przyległościowe), taki sposób wymiany informacji pomiędzy nimi szybko prowadzi do nowych postaci chromosomów, w których niektóre geny nie pojawiłyby się w ogóle, a inne mogłyby wystąpić wielokrotnie. Jeżeli chromosomy w przyjętym sposobie kodowania przedstawiają zawsze rozwiązania dopuszczalne, to nowo powstałe osobniki w wyniku krzyżowania z punktami cięcia reprezentowałyby rozwiązania niedopuszczalne z punktu widzenia podstawowych założeń problemu układania tras pojazdów. Tym samym konieczne byłoby przynajmniej stosowanie specjalnych procedur „naprawy” chromosomu do postaci, w której znów reprezentowałby on rozwiązanie dopuszczalne.

Biorąc pod uwagę przedstawione powyżej niedogodności związane z możliwością niespełniania przez rozwiązania podstawowych założeń problemu wielu komiwojażerów, dla chromosomów, w których różne postaci wynikają z permutacji genów, zaproponowano metody krzyżowania i mutacji prowadzące do tworzenia nowych osobników bez konieczności ich późniejszego „naprawiania” (Potvin, 1996). Jednym z powszechnie stosowanych operatorów krzyżowania dla reprezentacji ścieżkowej jest procedura krzyżowania z częściowym odwzorowaniem PMX (*Partially-Mapped Crossover*). Polega ona na utworzeniu nowego chromosomu przez skopiowanie losowo wybranego podciągu genów z jednego z rodziców oraz zachowaniu porządku i pozycji tak wielu genów, jak to możliwe, drugiego rodzica:

Rodzic 1: [4 8 5 3 1 9 2 6 7]

Potomek 1: [4 8 9 5 3 7 2 6 1]

Rodzic 2: [6 1 9 5 3 7 4 2 8]

Potomek 2: [6 7 5 3 1 9 4 2 8]

W krzyżowaniu PMX kładzie się nacisk na zachowanie pozycji jak największe liczby genów przy kopiowaniu ich z jednego lub drugiego rodzica do potomka. Liczba genów, które nie zachowują swoich pozycji w potomkach, jest równa co najwyżej liczbie genów z losowo wybranego podciągu.

Innym podobnym sposobem krzyżowania przeznaczonym do kodowania permutacyjnego jest krzyżowanie z porządkowaniem OX (*Ordinary Crossover*), w którym tak jak w PMX wybierany jest losowo podciąg genów od jednego rodzica,

a kolejność ułożenia pozostałych genów od drugiego z rodziców. W pierwszej kolejności należy losowy podciąg genów skopiować od jednego z rodziców do potomka, a następnie, poczynszy od pierwszego genu, za wybranym podciągiem skopiować pozostałe geny (unikając powtórzeń) od drugiego z rodziców:

Rodzic 1: [4 8 5 3 1 9 2 6 7]	Potomek 1: [6 7 5 3 1 9 4 2 8]
Rodzic 2: [6 1 9 5 3 7 4 2 8]	Potomek 2: [8 1 9 5 3 7 2 6 4]

Przedstawione powyżej operatory krzyżowania miały na celu zachowanie pozycji punktów obsługi w trasach (reprezentowanych przez geny w chromosomach). Ich uzupełnieniem są operatory krzyżowania, których zadaniem jest z kolei zachowanie uporządkowania punktów obsługi w trasach, a tym samym genów w odpowiadających im chromosomach. Przykładem takiego operatora krzyżowania jest krzyżowanie cykliczne CX (*Cycle Crossover*), którego działanie koncentruje się na zachowaniu uporządkowania fragmentów chromosomów od rodziców przez zachowanie pozycji każdego genu od jednego z rodziców.

Rodzic 1: [4 8 5 3 1 9 2 6 7]	Potomek 1: [4 1 9 5 3 7 2 6 8]
Rodzic 2: [6 1 9 5 3 7 4 2 8]	Potomek 2: [6 8 5 3 1 9 4 2 7]

Pewien wybrany losowo podzbiór genów z jednego rodzica jest kopiowany do potomka z zachowaniem pozycji w chromosomie tego rodzica, natomiast pozostałe geny – także z zachowaniem ich pozycji – są kopiowane od drugiego z rodziców.

Innym przykładem jest krzyżowanie oparte na porządkowaniu OBX (*Ordinary-Based Crossover*), w którym od pierwszego z rodziców wybierany jest losowy podzbiór genów i umieszczany w potomku w takim samym porządku jak u rodzica pierwszego, ale na pozycjach wskazanych w rodzicu drugim. Pozostałe geny chromosomu uzupełniane są genami z chromosomu rodzica drugiego:

Rodzic 1: [4 8 5 3 1 9 2 6 7]	Potomek 1: [6 1 5 9 3 7 4 2 8]
Rodzic 2: [6 1 9 5 3 7 4 2 8]	Potomek 2: [4 8 5 3 1 9 7 6 2]

Wymienione wcześniej przykłady operatorów krzyżowania uwzględniały w swoich konstrukcjach zachowanie albo pozycji genów w chromosomach (punktów obsługi w trasach), albo ich uporządkowanie (kolejność odwiedzania punktów obsługi). Natomiast żadne z nich nie brały pod uwagę połączeń pomiędzy punktami obsługi. Stąd też proponowane były inne operatory krzyżowania, których celem jest przede wszystkim zachowanie tak wielu połączeń pomiędzy punktami obsługi w trasach reprezentowanych przez chromosomy rodzicielskie, jak to tylko możliwe.

Na szczególną uwagę zasługuje często stosowana metoda krzyżowania uwzględniająca połączenia pomiędzy punktami obsługi, jaką jest operator wymiany krawędzi ER (*Edge Recombination*) (Whitley i in., 1989). Wykorzystuje on pojęcie tzw. mapy połączeń pomiędzy punktami obsługi występującymi w trasach reprezentowanych przez chromosomy rodzicielskie. Są to możliwe połączenia każdego punktu obsługi do innych punktów występujących w trasach obydwu chromosomów. Dla dwóch przykładowych tras przedstawionych w postaci chromosomów z kodowaniem ścieżkowym: [4 8 5 3 1 9 2 6 7] oraz [6 1 9 5 3 7 4 2 8] mapa połączeń jest następująca:

$$\begin{array}{lll}
 P_1: & P_3 P_6 P_9 & P_4: P_2 P_7 P_8 & P_7: P_3 P_4 P_6 \\
 P_2: & P_4 P_6 P_8 P_9 & P_5: P_3 P_8 P_9 & P_8: P_2 P_4 P_5 P_6 \\
 P_3: & P_2 P_5 P_7 & P_6: P_1 P_2 P_7 P_8 & P_9: P_1 P_2 P_5
 \end{array}$$

Dla punktu P_1 w mapie połączeń występują punkty: $P_3 P_6 P_9$, ponieważ w obydwu chromosomach stanowią one sąsiednie geny dla punktu P_1 . Każde umieszczenie nowego genu w chromosomie potomnym musi skutkować odpowiednią korektą wyżej przedstawionej mapy połączeń. Budowę chromosomu należy rozpocząć od losowo wskazanego genu. Po jego wybraniu z mapy usuwane są wszystkie połączenia związane z punktem obsługi reprezentowanym przez wybrany gen. Kolejne geny wybierane są spośród tych, które reprezentują punkty obsługi cechujące się najmniejszą liczbą możliwych połączeń w mapie. Jeżeli jako początkowy gen nowego chromosomu wybrany zostanie np. [5], wtedy mapa połączeń po korekcie wyjściowej będzie następująca:

$$\begin{array}{lll}
 P_1: & P_3 P_6 P_9 & P_4: P_2 P_7 P_8 & P_7: P_3 P_4 P_6 \\
 P_2: & P_4 P_6 P_8 P_9 & & P_8: P_2 P_4 P_6 \\
 P_3: & P_2 P_7 & P_6: P_1 P_2 P_7 P_8 & P_9: P_1 P_2
 \end{array}$$

Następny wybrany gen do chromosomu potomnego będzie wybrany losowo spośród genów reprezentujących punkty obsługi P_3 i P_9 (mają najmniej możliwych połączeń). Tym samym częściowo zbudowany chromosom potomny przyjmie postać: [5 3] lub [5 9]. W kolejnych krokach należy znów zmodyfikować mapę połączeń (usuwając wybrany punkt obsługi), a następnie wskazać kolejny z najmniejszą liczbą możliwych połączeń. Postępując analogicznie, jedna z możliwych postaci chromosomu potomnego to [5 3 7 4 8 2 9 6 1]. Operator krzyżowania ER przenosi od rodziców ponad 90% połączeń w reprezentowanych przez nich trasach.

Inną metodę krzyżowania uwzględniającą połączenia pomiędzy punktami obsługi w trasach przedstawia heurystyczny operator krzyżowania HX (*Heuristic Crossover*). W przeciwieństwie do poprzednich operatorów, w tym wypadku

w konstrukcji chromosomów potomnych uwzględniana jest dodatkowa wiedza o problemie układania tras pojazdów w postaci długości rozpatrywanych połączeń tras reprezentowanych przez chromosomy rodziców. Począwszy od losowo wybranego genu, rozpatrywane są kolejno możliwe połączenia punktu obsługi wybranego genu z czterema punktami znajdującymi się w trasach wyjściowych (po dwa dla każdej z nich), a następnie wybierane jest połączenie najkrótsze. Gdy wybrane połączenie prowadzi do powstania podcyklu trasy (wskazuje na punkt obsługi już w trasie występujący), należy losowo wybrać kolejne połączenie prowadzące do punktu obsługi jeszcze nieodwiedzanego przez pojazd. Operator krzyżowania HX przenosi około 60% połączeń z tras reprezentowanych przez chromosomy rodzicielskie (Grefenstette, 1987).

Obok operatorów krzyżowania przeznaczonych do kodowania rozwiązań układania tras jednego i wielu pojazdów zaproponowano także operatory mutacji, wśród których wymienić można między innymi następujące:

- przestawienie genu;
- wymiana genów;
- inwersja;
- lokalna optymalizacja.

Pierwszy z wymienionych operatorów mutacji polega na prostej zmianie miejsca położenia losowo wybranego genu chromosomu, co odpowiada zmianie umieszczenia punktu obsługi w trasie. Pewnym jego rozszerzeniem jest zmiana położenia w chromosomie pewnego losowo wybranego podciągu genów:

Chromosom A: [4 8 5 3 1 9 2 6 7] ⇒ Chromosom A': [4 8 3 1 9 5 2 6 7]

Chromosom A: [4 8 5 3 1 9 2 6 7] ⇒ Chromosom A': [4 8 9 2 6 5 3 1 7]

Drugi operator mutacji jest nieco podobny do pierwszego i polega na zamianie miejsc dwóch losowo wybranych genów lub losowo wybranych podciągów genów o tej samej długości:

Chromosom A: [4 8 5 3 1 9 2 6 7] ⇒ Chromosom A': [4 8 2 3 1 9 5 6 7]

Chromosom A: [4 8 5 3 1 9 2 6 7] ⇒ Chromosom A': [4 2 6 3 1 9 8 5 7]

Trzeci operator mutacji to inwersja, w której dokonuje się losowego wyboru podciągu przynajmniej dwóch genów, a następnie zamiany kolejności ich ułożenia:

Chromosom A: [4 8 5 3 1 9 2 6 7] ⇒ Chromosom A': [4 8 2 9 1 3 5 6 7]

Ostatni z wymienionych operatorów mutacji to lokalna optymalizacja. Podobnie jak w wypadku heurystycznego operatora krzyżowania HX wprowadza on pewną wiedzę do algorytmu ewolucyjnego o optymalizowanych trasach pojazdów. Polega na zdekodowaniu chromosomu do postaci rzeczywistej i próbie poprawy tego rozwiązania z zastosowaniem np. prostych metod lokalnej optymalizacji typu 2-opt.

Przedstawione powyżej operatory mutacji, którymi można się posłużyć w odniesieniu do chromosomów zbudowanych z wykorzystaniem alfabetu liczb naturalnych, są operacjami bardzo prostymi na osobnikach, a tym samym na rzeczywistych trasach, jakie one reprezentują. Taki też jest cel operatora mutacji, którego działanie polega na dokonaniu niewielkich zmian, tak aby można było dywersyfikować proces przeszukiwania przestrzeni rozwiązań. Jednakże w sytuacji, gdy operator mutacji dominuje lub jest jedynym stosowanym operatorem genetycznym w algorytmie ewolucyjnym, konieczna może się okazać budowa bardziej złożonych sposobów modyfikacji pojedynczych osobników. Takim przykładem jest operator mutacji MUTG będący pewnego rodzaju złożeniem operatorów mutacji polegających na inwersji i wymianie genów (Chatterjee i in., 1996). Konstrukcja tego operatora sprowadza się do umieszczenia g punktów cięcia w chromosomie, a następnie wymiany i/lub inwersji genów. W wyniku umieszczenia g punktów cięcia w chromosomie zostaje on podzielony na $g + 1$ podciągów genów, gdzie wszystkie podciągi z wyjątkiem skrajnych podlegają zmianom (przestawianiu i inwersji). Poniżej przedstawiono wybrane przykłady zastosowania operatora MUTG, w którym parametr g przyjmuje wartość 3:

Chromosom A: [4 8 | 5 3 1 | 9 2 | 6 7] \Rightarrow Chromosom B: [4 8 | 9 2 | 5 3 1 | 6 7]

Chromosom A: [4 8 | 5 3 1 | 9 2 | 6 7] \Rightarrow Chromosom C: [4 8 | 1 3 5 | 9 2 | 6 7]

Chromosom A: [4 8 | 5 3 1 | 9 2 | 6 7] \Rightarrow Chromosom D: [4 8 | 9 2 | 1 3 5 | 6 7]

W pierwszym wypadku operator MUTG odpowiada jedynie zamianie miejscami podciągów genów (Chromosom B). Chromosom C jest z kolei modyfikacją polegającą na zastosowaniu tylko inwersji na wybranym podciągu genów. Natomiast w trzecim chromosomie modyfikacja stanowi złożenie dwóch pierwszych. Zauważyć należy, że możliwych kombinacji powstałych z chromosomu wyjściowego dla $g = 3$ może być siedem. W trakcie działania algorytmu parametr g jest dopasowywany dynamicznie, przy czym prawdopodobieństwo przyjęcia przez algorytm danej wartości maleje wraz z jej wzrostem.

Przyjęcie odpowiedniego sposobu kodowania rzeczywistych tras pojazdów do postaci chromosomów, a następnie określenie odpowiednich operujących bezpośrednio na nich operatorów genetycznych to najważniejsze czynności budowy wszystkich algorytmów ewolucyjnych. Proponowane w literaturze przedmiotu algorytmy ewolucyjne przeznaczone do rozwiązywania zadań układania tras

pojazdów przyjmują w swoim ogólnym schemacie postać przedstawioną wcześniej na rysunku 7.39. Różnice w obrębie samych algorytmów dotyczą często sposobu dokonywania reprodukcji oraz sukcesji.

Reprodukcja jest procesem mającym na celu wybranie populacji rodzicielskiej spośród chromosomów stanowiących populację ocenianą. Podstawą tego procesu jest wartość funkcji przystosowania każdego chromosomu (w szczególności wartość funkcji celu rozwiązania reprezentowanego przez chromosom). Reprodukcja może się odbywać w sposób całkowicie losowy, deterministyczny lub mieszany. Przyjęte sposoby wyboru chromosomów tworzących populację rodzicielską noszą nazwę metod selekcji. Dotychczas zaproponowano wiele metod selekcji (Arabas, 2001; Goldberg, 2003; Michalewicz, 1999). Jednakże wśród nich najpopularniejsze są dwie. Pierwsza to selekcja ruletkowa, w której wycinek koła ruletki odpowiada danemu chromosomowi w wielkości odpowiedniej do jego wartości funkcji przystosowania. Proces wyboru chromosomu odbywa się przez jednokrotne zakręcenie kołem ruletki. Im lepsze przystosowanie chromosomu, tym większa szansa na umieszczenie go w populacji rodzicielskiej.

Z kolei druga metoda to selekcja turniejowa. Jej proces jest częściowo deterministyczny. Spośród populacji chromosomów wybierany jest losowo pewien ich podzbiór o określonej liczebności (liczebność podzbioru jest parametrem tej metody), a następnie na podstawie wartości funkcji przystosowania wskazywany jest najlepszy chromosom, który zostanie umieszczony w populacji rodzicielskiej. Proces ten powtarzany jest do momentu, w którym populacja rodzicielska osiągnie żądaną liczebność.

Należy tutaj zauważyć, że tak przedstawione sposoby selekcji nie gwarantują zachowania w dalszym etapie działania algorytmu chromosomu reprezentującego najlepszą trasę (lub układ tras dla problemu wielu komiwojażerów). Po pierwsze, mogą one nie zostać wybrane do populacji rodzicielskiej (nie zostaną wylosowane). Po drugie, nawet jeżeli najlepsze chromosomy znajdują się w populacji rodzicielskiej, to w wyniku przekształceń wykonanych przez operatory genetyczne mogą powstać inne, które reprezentować będą gorsze rozwiązania. Dlatego też na równorzędnym poziomie ważności z reprodukcją stoi proces sukcesji, polegający na sposobie tworzenia nowej populacji, która zostanie ponownie oceniona w kolejnej iteracji algorytmu ewolucyjnego (pokoleniu).

W wariacie charakterystycznym zwłaszcza dla algorytmów genetycznych sukcesja może polegać na przyjęciu do następnego pokolenia zbioru chromosomów powstałych w wyniku działania operatorów krzyżowania i mutacji. Jednakże sukcesja może polegać także na stosowaniu tzw. zabiegu elitarności, w którym pewien niewielki podzbiór najlepszych chromosomów populacji zostanie skopiowany bezpośrednio do populacji nowego pokolenia bez poddawania go działaniu operatorów genetycznych. Jest to swego rodzaju pamięć algorytmu nie tylko przechowująca najlepsze dotychczas znalezione trasy, ale także bezpośrednio angażująca je do generowania nowych lepszych rozwiązań. Wielkość tego podzbioru może stanowić bardzo istotny parametr działania algorytmu ewolucyjnego.

Przykładem algorytmu ewolucyjnego dla zadania układania tras wielu pojazdów, a realizującego strategię przeszukiwania przestrzeni rozwiązań przedstawioną schematem z rysunku 7.39, jest wspomniany wcześniej algorytm PEWKOM⁶. Jego działanie jest zbliżone do działania klasycznego algorytmu genetycznego. Populacja początkowa tworzona jest z losowo skonstruowanych rozwiązań. W każdym kroku pojedynczej iteracji algorytmu generowane są populacje chromosomów odpowiednio: wyjściowa (oceniana), rodzicielska (po selekcji), potomna (po krzyżowaniu i mutacji) oraz nowego pokolenia (po lokalnej optymalizacji). Do selekcji wykorzystano metodę turniejową z liczebnością zbioru losowanych chromosomów równą 5. Zarówno proces krzyżowania (OX i CX), jak i proces mutacji (zamiana miejscami genów) zachodzą z zadaniem prawdopodobieństwem, przy czym należy zwrócić uwagę, że prawdopodobieństwo mutacji jest często równe lub nawet większe od prawdopodobieństwa krzyżowania. Ostatnią czynnością przed utworzeniem osobnika, który umieszczony zostanie w populacji nowego pokolenia, jest proces lokalnej optymalizacji. Każdy z chromosomów populacji potomnej zostaje zdekodowany do postaci rozwiązania rzeczywistego, a poszczególne trasy pojazdów są indywidualnie poprawiane z zastosowaniem heurystycznego algorytmu 2-opt.

Z kolei w algorytmie zaproponowanym przez B.M. Bakera i M.A. Ayeche-wa (2003) populacja początkowa stanowi złożenie rozwiązań wygenerowanych za pomocą dwóch klasycznych metod heurystycznych: *sweep* B.E. Gilletta i L.R. Millera oraz wykorzystującej uogólnione zadanie przydziału M. Fischera i R. Jaikumara (Baker i Ayeche-wa, 2003). Następnie powtarzany jest proces iteracyjny polegający kolejno na: wyborze dwóch rodziców (selekcja turniejowa), którzy zostaną poddani procesowi krzyżowania i mutacji, ocenie utworzonego chromosomu potomnego oraz modyfikacji lub nie aktualnie rozpatrywanej populacji. Innymi słowy, w tym algorytmie nie jest tworzona odrębna nowa populacja chromosomów w każdej iteracji (pokoleniu), a jedynie jest modyfikowana aktualna przez usunięcie jednego chromosomu i zastąpienie go nowym. Cechą wyróżniającą ten algorytm jest właśnie omawiana procedura sukcesji. Nowy, utworzony w wyniku zastosowania operatorów genetycznych chromosom jest oceniany przez obliczenie dla niego dwóch wielkości: wartości funkcji przystosowania $f(ch)$ chromosomu ch (całkowita długość wszystkich tras pojazdów) oraz wartości funkcji niedopasowania $g(ch)$ chromosomu ch , która oznacza stopień przekroczenia zadanych ograniczeń dla pojazdów (np. przekroczenie ładowności pojazdu i/lub dopuszczalnej długości trasy). Następnie populacja w danej iteracji algorytmu dzielona jest na cztery podzbiory S_i (dla $i = 1, 2, 3, 4$) zgodnie z poniższą formułą (7.37):

6 PEWKOM – Program Ewolucyjny Wielu KOMiwojażerów (Jadczak, 2005).

$$S_i = \begin{cases} S_1 = \{ch_p \in P : f(ch_p) \geq f(ch_N), g(ch_p) \geq (ch_N)\} \\ S_2 = \{ch_p \in P : f(ch_p) < f(ch_N), g(ch_p) \geq (ch_N)\} \\ S_3 = \{ch_p \in P : f(ch_p) \geq f(ch_N), g(ch_p) < (ch_N)\} \\ S_4 = \{ch_p \in P : f(ch_p) < f(ch_N), g(ch_p) < (ch_N)\} \end{cases} \quad (7.37)$$

gdzie P oznacza populację chromosomów w danej iteracji, ch_p i ch_N oznaczają odpowiednio chromosomy: należący do populacji P i chromosom potomny. Stary chromosom zastępowany przez nowego potomnego osobnika wybierany jest z pierwszego niepustego zbioru w kolejności: S_1 , S_2 lub S_3 . Jeżeli wszystkie te trzy podzbiory są puste, wtedy chromosom potomny nie jest umieszczany w populacji.

A. Jaszkievicz i P. Kominek (2003) przedstawili algorytm genetyczny przeszukiwania przestrzeni rozwiązań GLS⁷, w którym algorytm ewolucyjny zawiera także – tak jak w algorytmie PEWKOM – procedurę lokalnej optymalizacji. Jednakże jest ona o wiele bardziej zaawansowana i polega na próbie znalezienia najlepszego rozwiązania sąsiedniego do aktualnie rozpatrywanego (w stosunku do wszystkich tras pojazdów jednocześnie, a nie tylko jednego pojazdu). Procedura lokalnej optymalizacji polega tu na wygenerowaniu rozwiązań sąsiednich przez przesunięcie fragmentu trasy z trasy jednego pojazdu do trasy drugiego pojazdu. Spośród znalezionych rozwiązań sąsiednich wybierane jest najlepsze. W algorytmie GLS z populacji chromosomów losowane są z zadaniem prawdopodobieństwem dwa chromosomy, które poddawane są procesowi krzyżowania, a w jego wyniku powstaje chromosom potomny. Następnie w stosunku do reprezentowanego przez niego rozwiązania rzeczywistego zastosowanie znajduje procedura lokalnej optymalizacji.

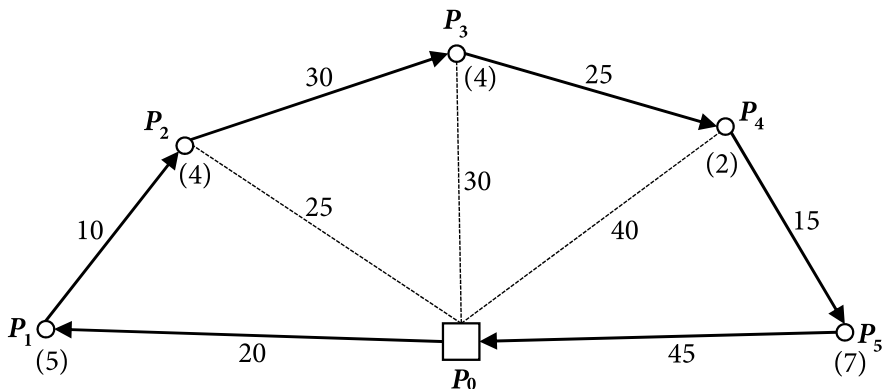
Sukcesją w algorytmie GLS polega na zastąpieniu najgorszego chromosomu lepszym, pod warunkiem że reprezentuje on rozwiązanie rzeczywiste różne od tych, które już istnieją w populacji. Od ogólnie przyjętego szkicu algorytmów ewolucyjnych ten różni się tym, że w każdej iteracji rozpatruje nie zbiór rozwiązań (populację), a jedynie jedno rozwiązanie indywidualnie. Lecz jak stwierdzają autorzy, przedstawiona idea algorytmu GLS nie stwarza żadnych przeszkód, by go zaimplementować w postaci bardziej tradycyjnej (z zastosowaniem populacji chromosomów).

Podobne rozwiązanie do wyżej przedstawionego algorytmu ewolucyjnego zaprezentował Ch. Prins (2004), w którym w każdej iteracji nie jest generowane całe pokolenie (populacja) nowych rozwiązań, a jedynie dwa nowe chromosomy. W pierwszej kolejności w wyniku selekcji turniejowej wybierane są dwa chromosomy rodzicielskie, które są poddawane krzyżowaniu. Ze względu na to, że

7 Takie połączenie algorytmów ewolucyjnych z heurystycznymi metodami lokalnej optymalizacji nosi też nazwę algorytmów memetycznych (Jaszkievicz i Kominek, 2003).

kodowanie chromosomu jest oparte na alfabecie liczb naturalnych, a także ważny jest porządek ułożenia genów, zastosowany został operator krzyżowania z porządkowaniem OX. Następnie dwa nowe chromosomy potomne poddawane są operacji mutacji z zadaniem prawdopodobieństwem p_M . Sukcesja polega na umieszczeniu nowych chromosomów w istniejącej populacji rozwiązań. Odbywa się to przez losowy wybór dwóch spośród połowy najgorszych osobników, a następnie zastąpienie ich chromosomami potomnymi (zmutowanymi lub nie). Algorytm jest powtarzany do momentu, kiedy wykonana zostanie założona w kroku inicjalizacji maksymalna liczba iteracji z rzędu, w których nie znaleziono chromosomu reprezentującego rozwiązanie lepsze od najlepszego dotychczas znalezione. Drugim jednoczesnym warunkiem końca działania algorytmu jest sprawdzenie, czy wartość funkcji celu osiągniętego rozwiązania jest optymalna. Ten drugi warunek dotyczy problemów, dla których znane jest rozwiązanie optymalne.

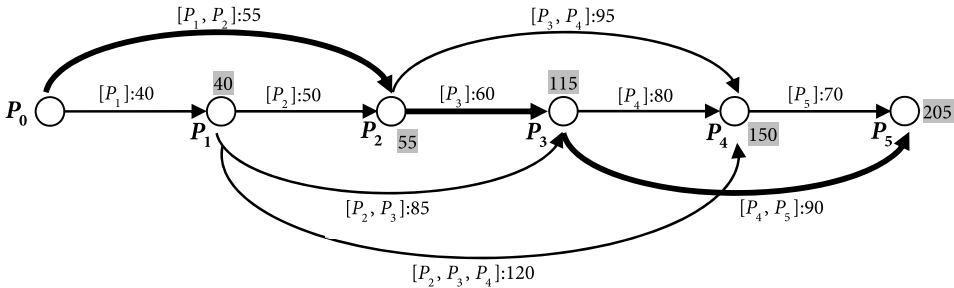
Na koniec rozważań dotyczących algorytmów ewolucyjnych elementem, na który należy zwrócić szczególną uwagę, jest sposób kodowania rozwiązania w propozycji Ch. Prinsa, a w szczególności procedura powodująca przekształcenie chromosomu do postaci rzeczywistego rozwiązania. Jak już wspomniano, chromosomy kodowane są z wykorzystaniem alfabetu liczb naturalnych, lecz nie jest to często stosowana dla problemów układania tras pojazdów naturalna metoda ścieżkowa. Każdy osobnik reprezentuje tylko porządek, w jakim należy brać pod uwagę kolejne punkty obsługi przy budowie tras poszczególnych pojazdów. Geny chromosomu reprezentują poszczególne punkty obsługi, które tworzą pewien pomocniczy graf składający się z $N + 1$ węzłów ponumerowanych od 0 do N . Ponadto krawędzie pomiędzy węzłami pomocniczego grafu (i, j) , $i < j$, reprezentują trasy zawierające tylko te kolejne punkty obsługi, które spełniają warunki ładowności pojazdu. Waga każdej takiej krawędzi stanowi koszt trasy pojazdu. Innymi słowy,



Rys. 7.44a. Potrzebne dane wynikające z postaci chromosomu [1 2 3 4 5]. W nawiasach podane są popyty punktów obsługi
Źródło: (Prins, 2004).

pomocniczy graf reprezentuje możliwe trasy pojazdów przy zachowaniu porządku punktów obsługi z reprezentującego go chromosomu. Poniżej przedstawiony jest przykład takiego grafu (rys. 7.44b) dla chromosomu: [1 2 3 4 5] przy założeniu ładowności pojazdu równej 10. Dane dotyczące tych pięciu punktów obsługi (odległości oraz zapotrzebowania), a także układ wynikający z porządku przedstawionego w chromosomie pokazano poniżej (rys. 7.44a).

Postać pomocniczego grafu składającego się ze zbioru sześciu węzłów oraz dziesięciu możliwych krawędzi jest następująca:



Rys. 7.44b. Graf pomocniczy i najkrótsza droga
 Źródło: (Prins, 2004).

Graf nie posiada krawędzi łączących węzły P_0 z P_3 oraz P_2 z P_5 , ponieważ trasy pojazdów obejmujące punkty odpowiednio: P_1, P_2, P_3 oraz P_3, P_4, P_5 przekraczałyby założoną ładowność pojazdu. Tak samo dotyczy to tras zawierających cztery i więcej punktów obsługi.

Dla tak zbudowanego grafu pomocniczego należy znaleźć najkrótszą ścieżkę prowadzącą od węzła P_0 do węzła ostatniego (w wypadku przykładowego chromosomu będzie to P_5). Krawędzie wchodzące w skład znalezionej najkrótszej ścieżki (drogi) w grafie pomocniczym wyznaczają zbiór tras pojazdów reprezentowanych przez dany chromosom. Tutaj będą to krawędzie: (P_0, P_2) , (P_2, P_3) i (P_3, P_5) , dla których suma wag wynosi 205. Tym samym chromosom [1 2 3 4 5] reprezentuje zbiór tras: $P_0 \rightarrow P_1 \rightarrow P_2 \rightarrow P_0$, $P_0 \rightarrow P_3 \rightarrow P_0$ oraz $P_0 \rightarrow P_4 \rightarrow P_5 \rightarrow P_0$. Dla porównania, gdyby bezpośrednio z chromosomu kolejno przyporządkowywać poszczególne punkty obsługi do pojazdów, uwzględniając ich ładowność, to zbiór tras byłby nieco inny: $P_0 \rightarrow P_1 \rightarrow P_2 \rightarrow P_0$, $P_0 \rightarrow P_3 \rightarrow P_4 \rightarrow P_0$ oraz $P_0 \rightarrow P_5 \rightarrow P_0$.

Przedstawione w tym rozdziale algorytmy metaheurystyczne nie wyczerpują wszystkich możliwych prób budowy efektywnych metod rozwiązywania zagadnień układania tras jednego i wielu pojazdów. Należy zwrócić uwagę, że podejmowano także próby, w których dokonywano konstrukcji algorytmów będących złożeniem (hybrydą) dwóch lub większej liczby metod należących do różnych klas metaheurystyk. Ponadto należy podkreślić szczególną rolę, jaką w algorytmach metaheurystycznych odgrywają klasyczne metody heurystyczne oraz proste

metody przeszukiwania lokalnego. Klasyczne metody heurystyczne (np. konstrukcyjne) stanowią sposób na szybkie i stosunkowo łatwe konstruowanie dobrych rozwiązań początkowych. Z kolei proste algorytmy przeszukiwania lokalnego są użytecznym narzędziem służącym do poprawy rozwiązań zarówno wygenerowanych w trakcie działania algorytmu metaheurystycznego, jak i utworzonych na jego końcu.

Podejścia metaheurystyczne omówione w tym rozdziale na pewno cechują się bardzo dużą elastycznością, która pozwala na włączanie dodatkowej wiedzy o optymalizowanym problemie, ale przede wszystkim na dowolne kształtowanie ostatecznej postaci algorytmu, dopasowanej nawet do rozwiązania konkretnego zadania. Niestety okupione jest to często bardzo dużą liczbą różnego rodzaju parametrów, których wartości trzeba ustalić na początku działania algorytmu. Z kolei określenie najlepszych kombinacji wartości tych parametrów zwykle wymaga wykonania wielu prób testowych noszących miano strojenia algorytmu. Stanowią one może dodatkowy element komplikujący zastosowanie metod metaheurystycznych. Niezależnie jednak od tych trudności stanowią one najczęściej podejmowaną drogę w poszukiwaniu coraz lepszych metod układania tras dla pojazdów.

Zakończenie

Optymalizacja procesów transportowych w obszarach dystrybucji oraz zaopatrzenia stanowi bardzo istotny element szeroko rozumianego zarządzania logistycznego, dotyczącego zarówno pojedynczego przedsiębiorstwa, jak i całego łańcucha dostaw. Procesy te nie ograniczają się jedynie do jak najtańszego lub najszybszego wykonania zadania przewozu produktu z punktu nadania *A* do punktu odbioru *B*. Także dobrze rozpoznane w literaturze z zakresu badań operacyjnych klasyczne problemy transportowe, opisujące zadania przewozu od wielu dostawców do wielu odbiorców, nie stanowią wyczerpującego zbioru zdiagnozowanych problemów obsługi transportowej łańcuchów dostaw. Ze względu na różnorodność procesów zaopatrzenia oraz dystrybucji w łańcuchach dostaw zagadnienia układania tras jednego oraz wielu pojazdów (jednego i wielu komiwojażerów) zyskały sobie szczególną, odrębną pozycję wśród problemów transportowych.

Konstruowane przez decydentów łańcuchy dostaw są bardzo zróżnicowane. Z punktu widzenia obsługi transportowej do głównych cech charakteryzujących dany łańcuch dostaw zaliczyć należy przede wszystkim:

- strukturę łańcucha, która ma czasami charakter wielopoziomowy;
- rozmiar łańcucha dostaw, w tym udział dostawców i odbiorców;
- cele stawiane w ramach formułowanych problemów transportowych;
- ograniczenia wynikające ze specyfiki nie tylko budowy łańcucha dostaw, ale i branży, w której on funkcjonuje.

Tym samym zagadnienia układania tras pojazdów stały się bardzo interesującym obszarem badawczym zarówno dla logistyków, jak i specjalistów z innych dziedzin, a w szczególności ekonomistów, matematyków, informatyków, a także praktyków. Doprowadziło to w efekcie do powstania dużej liczby podobnych, aczkolwiek z konieczności traktowanych odrębnie problemów układania tras pojazdów. Sformułowano w sumie kilkanaście różnych modeli zagadnień układania tras jednego i wielu pojazdów. Trzeba zatem uznać, że problematyka układania tras pojazdów w obsłudze transportowej łańcuchów dostaw może stanowić odrębny, zasługujący na szczególną uwagę dział logistyki operacyjnej.

Blisko siedemdziesięcioletnia historia prac badawczych nad problematyką układania tras pojazdów zaowocowała olbrzymią liczbą algorytmów rozwiązywania konkretnych zadań przewozowych, ale również propozycji stanowiących pewne ogólne podejście do poszukiwania rozwiązania optymalnego. Wśród proponowanych metod układania tras jednego i wielu pojazdów wyróżnić można dwie podstawowe grupy: metody dokładne oraz przybliżone (heurystyczne).

Metody dokładne – w przeciwieństwie do metod przybliżonych – zawsze pozwalały znaleźć rozwiązanie optymalne. Opierają się głównie na idei metod podziału i ograniczeń, sprowadzającej się do swego rodzaju upraszczania rozwiązywanego zagadnienia do innej postaci (np. minimalnego drzewa rozpinającego lub zagadnienia przydziału). Choć gwarantują znalezienie optymalnej trasy pojazdu lub optymalnego zbioru tras pojazdów, nie są wolne od wad. Należy do nich niewątpliwie wysoki stopień skomplikowania konkretnej metody, co może skutecznie utrudniać nie tylko jej zrozumienie przez praktyków, ale także powodować wysokie koszty jej implementacji i wdrożenia. Dugą wadą metod dokładnych podnoszoną w literaturze przedmiotu jest czas poszukiwania rozwiązania optymalnego, który uznawany jest za relatywnie długi w porównaniu z metodami przybliżonymi. Choć należy także zauważyć, że według wielu autorów algorytmów dokładnych zastrzeżenie to stało się nieaktualne w dobie technologii komputerowej. Jednakże czas działania algorytmów dokładnych wciąż nie jest na tyle krótki, aby był akceptowalny dla decydentów, szczególnie gdy warunki rozwiązywanego problemu charakteryzują się dużą dynamiką zmian.

Drugą, znacznie szerszą grupą algorytmów układania tras pojazdów są algorytmy heurystyczne. W przeciwieństwie do metod dokładnych nie gwarantują one znalezienia rozwiązania optymalnego i jest to zarazem ich główna wada. Jednakże ich autorzy zwykle kładli nacisk na dwa podstawowe aspekty. Pierwszy to czas generowania rozwiązania końcowego, a przy tym takiego, które byłoby akceptowalne dla decydenta. Drugi to prostota algorytmu, która pozwalałaby na minimalizację kosztu jego implementacji i wdrożenia.

Wielość propozycji heurystycznych spowodowała wyodrębnienie się dwóch podstawowych podejść do rozwiązania problemu układania tras pojazdów, jakimi są metody konstrukcyjne oraz metody wzrostu. W pierwszej strategii rozwiązanie (trasa jednego pojazdu lub trasy wielu pojazdów) konstruowane jest stopniowo przy uwzględnieniu wszystkich ograniczeń. W drugim podejściu konstruowane jest rozwiązanie początkowe (często nawet niedopuszczalne), które w trakcie działania algorytmu jest poprawiane. Początkowo większą popularność zyskała pierwsza z wymienionych grup metod, co wynikało przede wszystkim z bardzo niskiego stopnia skomplikowania algorytmu oraz bardzo szybkiego poszukiwania rozwiązania końcowego. Niestety efektywność poszukiwania rozwiązania optymalnego problemów układania tras pojazdów była daleka od oczekiwań. Rozwój technologii komputerowej, a tym samym zwiększenie mocy obliczeniowych doprowadziły do przeniesienia punktu ciężkości poszukiwań efektywnych algorytmów na

metody wzrostu. Należy jednak zauważyć, że algorytmy konstrukcyjne nie odeszły w zapomnienie – wręcz przeciwnie, zostały bowiem uznane za efektywny sposób (przede wszystkim szybki) wyznaczania dobrych rozwiązań początkowych.

Szczególne miejsce wśród metod polegających na poprawie rozwiązania początkowego zyskały sobie metaheurystyki: strategie ewolucyjne, strategie symulowanego wyżarzania czy przeszukiwanie tabu. Stanowią one w pierwszej kolejności zbiór reguł służących do konstrukcji konkretnego algorytmu. Wiele takich algorytmów, co należy podkreślić, było w stanie znaleźć rozwiązanie optymalne dla rozwiązywanego zadania testowego. Niestety tak wysoka efektywność algorytmu zwykle okupiona była nie mniej wysokim stopniem jego skomplikowania, co oznaczało wzrost kosztu jego implementacji.

Ponadto wiele z proponowanych algorytmów, w szczególności ewolucyjnych czy symulowanego wyżarzania, wyróżnia się dużą liczbą dodatkowych parametrów strojących. W algorytmach ewolucyjnych są to np. prawdopodobieństwo krzyżowania lub mutacji, natomiast w algorytmach symulowanego wyżarzania mogą to być limit prób generowania rozwiązań sąsiednich przy zadanej temperaturze czy też parametry wpływające bezpośrednio na schemat spadku temperatury. Ich właściwe ustawienia początkowe (dostrajanie) nierzadko mają bardzo istotny wpływ na jakość uzyskiwanych rozwiązań. To zaś zazwyczaj komplikuje użytkowanie takiego algorytmu, w szczególności jeżeli autor nie podaje najlepszych wartości takich parametrów lub też ich wartości są uzależnione od rozwiązywania konkretnego zadania.

Innym elementem charakterystycznym – tym razem metod lokalnej optymalizacji – jest sposób generowania sąsiedztwa. Sposoby te zazwyczaj są niezmiennie w danym algorytmie, lecz bardzo często same w sobie mogą być skomplikowane. Poza tym zwykle wraz ze wzrostem rozmiarów zadania rośnie w sposób wykładniczy potencjalna liczba rozwiązań sąsiednich. Stanowić to może pewne ograniczenie w praktycznym wykorzystaniu algorytmu, ponieważ powstaje pytanie, czy powinno się przejrzeć wszystkie rozwiązania sąsiednie, czy też tylko część z nich, a jeżeli tak – to jaką (co stanowi kolejny dodatkowy parametr algorytmu).

Pomimo przedstawionych powyżej wad algorytmów heurystycznych, opartych na metaheurystykach algorytmów ewolucyjnych, symulowanego wyżarzania, przeszukiwania tabu czy algorytmów mrówkowych, to właśnie one są dzisiaj w centrum zainteresowania badaczy problemu układania tras pojazdów, o czym świadczy rosnąca w ostatnich latach liczba publikacji z tego zakresu. Podejmowane są także próby budowy kolejnych algorytmów heurystycznych czerpiących to, co najlepsze, z każdej strategii, przez co osobną grupę stanowią mogą algorytmy hybrydowe (np. połączenie algorytmów ewolucyjnych i lokalnej optymalizacji). Konstruuując nowy algorytm mający na celu znalezienie optymalnej trasy lub optymalnego zbioru tras dla problemu wielu komiwojażerów, trzeba jednak pamiętać, że powinien on być efektywny nie tylko z punktu widzenia możliwości znalezienia rozwiązania optymalnego. Jego efektywność mierzona musi być także stopniem

skomplikowania samego algorytmu (minimalizacji kosztu implementacji i wdrożenia), czasem poszukiwania najlepszego rozwiązania oraz stopniem komplikacji jego użytkowania (minimalizacja liczby parametrów, których wartość należy ustalić na początku).

Niniejsza książka nie wyczerpuje w pełni tematyki układania tras pojazdów w łańcuchach dostaw. Dotychczas zaproponowano bardzo dużą liczbę zarówno modeli problemów jednego i wielu komiwojażerów, jak i różnych algorytmów służących ich rozwiązaniu. Wynika to z bardzo bogatej i różnorodnej praktyki gospodarczej oraz wysokiej dynamiki zmian, skutkujących pojawianiem się nowych założeń i ograniczeń, które decydenci muszą uwzględniać w procesie poszukiwania najlepszych decyzji. Ponadto chęć poszukiwania coraz lepszych rozwiązań sprawia, że powstają coraz to nowe algorytmy, w tym także takie, które czerpią to, co najlepsze, z już istniejących. Jednakże zebrana i uporządkowana w niniejszej pracy wiedza z zakresu układania tras pojazdów może stanowić przydatny punkt wyjścia do dalszych prac nad bardzo ważnym problemem optymalizacyjnym logistyki operacyjnej, jakim jest zagadnienie układania tras pojazdów.

Spis rysunków i tabel

Rysunek 1.1. Zewnętrzny łańcuch dostaw	14
Rysunek 1.2. Wewnętrzny łańcuch dostaw	15
Rysunek 1.3a. Przykłady konfiguracji bezpośredniej w łańcuchu dostaw	16
Rysunek 1.3b. Przykłady konfiguracji dwuetapowej pośredniej w łańcuchu dostaw	16
Rysunek 1.3c. Przykład konfiguracji wieloetapowej pośredniej w łańcuchu dostaw	16
Rysunek 1.3d. Przykłady konfiguracji mieszanej w łańcuchu dostaw	17
Rysunek 1.4. Sieć dostaw	17
Rysunek 1.5. Model dostaw bezpośrednich	20
Rysunek 1.6. Model scentralizowany	21
Rysunek 1.7. Model zdecentralizowany	21
Rysunek 1.8. System jednoterminalowy (TL – terminale lokalne)	29
Rysunek 1.9. System wieloterminalowy równorzędny (TR – terminal regionalny)	30
Rysunek 1.10. System wieloterminalowy hierarchiczny (TG – terminal główny)	30
Rysunek 1.11. Model wahadłowy	31
Rysunek 1.12. Model promienisty	31
Rysunek 1.13. Model obwodowy	32
Rysunek 1.14. Model sztafetowy	32
Rysunek 2.1. Graficzna ilustracja klasycznego problemu transportowego	33
Rysunek 2.2. Graficzna ilustracja wieloszczeblowego problemu transportowego	39

Rysunek 3.1. Graficzna ilustracja problemu komiwojażera	51
Rysunek 3.2. Graficzna ilustracja możliwych tras problemu komiwojażera dla $n = 3$	54
Rysunek 3.3. Graf pełny dla problemu komiwojażera wraz z przykładową dopuszczalną trasą (cyklem Hamiltona)	55
Rysunek 3.4. Graf pełny dla problemu komiwojażera wraz z przykładową niedopuszczalną trasą (cyklem skróconym)	57
Rysunek 3.5. Połączenie pomiędzy punktami i a j	59
Rysunek 3.6. Wierzchołki stopnia parzystego (czerwony kolor) i nieparzystego (niebieski kolor)	63
Rysunek 3.7. Postać nowej macierzy odległości C' dla zadania m -komiwojażerów	67
Rysunek 4.1. Jednoznaczne przyporządkowanie punktów obsługi do bazy	72
Rysunek 4.2. Graficzna ilustracja problemu wielu komiwojażerów	72
Rysunek 4.3. Graf pełny dla problemu wielu komiwojażerów wraz z przykładowymi dopuszczalnymi trasami (cyklami Hamiltona)	74
Rysunek 4.4. Problem wielu komiwojażerów z dostawą lub odbiorem dóbr typu „wielu-do-wielu”	83
Rysunek 4.5a. Problem typu „jeden-do-wielu-do-jeden” – dostawa lub odbiór z dowolną kolejnością obsługi	84
Rysunek 4.5b. Problem typu „jeden-do-wielu-do-jeden” – dostawa lub odbiór w kolejności: najpierw dostawy, potem odbiory	85
Rysunek 4.5c. Problem typu „jeden-do-wielu-do-jeden” – dostawa oraz odbiór z dowolną kolejnością obsługi	85
Rysunek 4.5d. Problem typu „jeden-do-wielu-do-jeden” – dostawa oraz odbiór w kolejności: najpierw dostawy, potem odbiory	86
Rysunek 4.6. Problem wielu komiwojażerów z dostawą lub odbiorem dóbr typu „jeden-do-jeden”	86
Rysunek 4.7a. Wyjściowy problem wielu komiwojażerów	89
Rysunek 4.7b. Rozwiązanie optymalne klasycznej postaci problemu wielu komiwojażerów z ograniczeniami zasobowymi	90
Rysunek 4.7c. Układ tras dla problemu wielu komiwojażerów z obsługą rozdzieloną	91
Rysunek 4.8. Problem wielu komiwojażerów z uzupełnieniami ładunku	93
Rysunek 4.9. Problem wielu komiwojażerów z obsługą okresową	98
Rysunek 4.10a. Układ tras pojazdów <i>a priori</i>	105

Rysunek 4.10b. Układ tras pojazdów według strategii pierwszej	106
Rysunek 4.10c. Układ tras pojazdów według strategii drugiej	107
Rysunek 4.11. Proces podejmowania decyzji w czasie pracy pojazdu	111
Rysunek 4.12a. Wyjściowy (planowany) układ tras pojazdów	112
Rysunek 4.12b. Trasy w czasie realizacji	113
Rysunek 4.12c. Trasy pojazdów po dokonanej modyfikacji	113
Rysunek 4.13. Zadanie wyjściowe problemu wielu komiwojazerów z wieloma bazami	128
Rysunek 4.14. Hierarchia podejmowania decyzji w problemie wielu komiwojazerów	129
Rysunek 4.15a. Izolowane rejony działania poszczególnych baz	130
Rysunek 4.15b. Rejony działania poszczególnych baz ze spornymi punktami obsługi	130
Rysunek 4.16. Przykładowe rozwiązanie problemu wielu komiwojazerów z wieloma bazami	131
Rysunek 4.17. Przykładowe rozwiązanie problemu wielu komiwojazerów z wieloma bazami, w którym pojazd nie musi wracać do bazy, z której wyruszył	135
Rysunek 5.1. Drzewo metody podziału i ograniczeń	142
Rysunek 5.2. Drzewo metody podziału i ograniczeń dla problemu układania tras pojazdów	143
Rysunek 5.3. Graf nieskierowany problemu komiwojazera dla czterech odbiorców	144
Rysunek 5.4. Fragment drzewa metody podziału i ograniczeń dla czterech odbiorców	145
Rysunek 5.5. Graf przedstawiający zbiór możliwych tras dla węzła P3	146
Rysunek 5.6. Schemat algorytmu podziału i ograniczeń Carpaneto i Totha	149
Rysunek 5.7. Schemat algorytmu podziału i ograniczeń D.L. Millera i J.F. Pekny'ego	151
Rysunek 5.8. Przykład drzewa rozpinającego z wyodrębnionym wierzchołkiem nr 1	156
Rysunek 5.9. Schemat algorytmu podziału i ograniczeń M. Helda i R.M. Karpa	157
Rysunek 5.10. Ilustracja podziału zbioru krawędzi grafu	159
Rysunek 6.1. Schemat metody konstrukcyjnej dla problemu komiwojazera	164

Rysunek 6.2. Schemat metody konstrukcyjnej dla problemu wielu komiwojazerów	165
Rysunek 6.3. Schemat algorytmu drogi do najbliższego sąsiada	165
Rysunek 6.4. Schemat algorytmu sukcesywnego dołączania węzłów	165
Rysunek 6.5. Wstawienie punktu P^* do częściowo zbudowanej trasy	166
Rysunek 6.6. Ilustracja podziału zbioru krawędzi grafu	167
Rysunek 6.7. Schemat algorytmu łączenia podcykli	168
Rysunek 6.8. Łączenie dwóch podcykli w jedną trasę	168
Rysunek 6.9. Algorytm budowy trasy pojazdu z wykorzystaniem minimalnego drzewa rozpinającego	169
Rysunek 6.10a. Minimalne drzewo rozpinające dla dziesięciu punktów obsługi	169
Rysunek 6.10b. Podwojone minimalne drzewo rozpinające dla dziesięciu punktów obsługi	170
Rysunek 6.10c. Finalna trasa pojazdu algorytmu minimalnego drzewa rozpinającego	171
Rysunek 6.11. Algorytm N. Christofidesa	171
Rysunek 6.12a. Rozwiązanie problemu skojarzenia wierzchołków o minimalnym koszcie	172
Rysunek 6.12b. Graf Eulera	172
Rysunek 6.12c. Finalna trasa pojazdu algorytmu N. Christofidesa	173
Rysunek 6.13. Stopniowe formowanie się tras w algorytmie oszczędnościowego łączenia tras	173
Rysunek 6.14. Algorytm G. Clarka i J.W. Wrighta	174
Rysunek 6.15a. Dwie osobne (częściowo już zbudowane) trasy, z krańcowymi punktami P_i i P_j	175
Rysunek 6.15b. Nowa trasa zawierająca połączenie w (P_i, P_j)	175
Rysunek 6.16. Algorytm R.H. Mole'a i S.R. Jamesona	176
Rysunek 6.17. Dwuetapowa konstrukcja rozwiązania w problemie wielu komiwojazerów	178
Rysunek 6.18. Algorytm <i>sweep</i>	179
Rysunek 6.19. Rejony obsługi w algorytmie <i>sweep</i>	179
Rysunek 6.20. Algorytm <i>petal</i>	180
Rysunek 6.21a. Rejony obsługi i odpowiadające im trasy składające się, odpowiednio, z jednego i dwóch punktów obsługi	180
Rysunek 6.21b. Rejony obsługi i odpowiadające im trasy składające się z trzech punktów obsługi	180

Rysunek 6.22. Rejony obsługi i odpowiadające im trasy w algorytmie 2-petal (po lewej: jedna trasa zawiera się w drugiej; po prawej: trasy się przecinają)	181
Rysunek 6.23. Algorytm M. Fishera i R. Jaikumara	182
Rysunek 6.24. Rejony obsługi w algorytmie M. Fishera i R. Jaikumara	182
Rysunek 6.25. Algorytm BF-WOT	184
Rysunek 6.26. Algorytm „najpierw trasa, potem rejon”	186
Rysunek 6.27. Rejony obsługi wyznaczone w oparciu o koncentratory	187
Rysunek 6.28. Faza sekwencyjnej konstrukcji tras w algorytmie N. Christofidesa, A. Mingozziego i P. Totha	189
Rysunek 6.29. Faza równoległej konstrukcji tras w algorytmie N. Christofidesa, A. Mingozziego i P. Totha	189
Rysunek 6.30. Algorytm „najpierw trasa, potem rejon”	190
Rysunek 6.31a. Supertrasa	191
Rysunek 6.31b. Przykład podziału supertrasa na poszczególne rejony (trasy pojazdów)	191
Rysunek 7.1. Poszukiwanie najlepszego rozwiązania na podstawie strategii wzrostu (poprawy rozwiązań)	193
Rysunek 7.2. Algorytm wzrostu	194
Rysunek 7.3. Algorytm lokalnej optymalizacji	196
Rysunek 7.4. Wymiana dwóch odcinków tras w algorytmie 2-opt	196
Rysunek 7.5. Algorytm 2-opt	197
Rysunek 7.6a. Wymiana trzech odcinków tras w algorytmie k -opt dla $k = 3$	198
Rysunek 7.6b. Wymiana trzech odcinków tras w algorytmie k -opt dla $k = 4$	198
Rysunek 7.7. Algorytm Lina-Kernighana	199
Rysunek 7.8. Pierwszy warunek konstrukcji dopuszczalnej trasy w algorytmie Lina-Kernighana	200
Rysunek 7.9. Zmiana położenia trzech punktów w trasie w algorytmie Or-opt dla $s = 3$	200
Rysunek 7.10. Algorytm Or-opt	201
Rysunek 7.11a. Układ dwóch tras przed relokacją punktu obsługi	202
Rysunek 7.11b. Układ dwóch tras po relokacji punktu obsługi	202
Rysunek 7.12a. Układ dwóch tras przed relokacją punktu obsługi	203
Rysunek 7.12b. Układ dwóch tras po relokacji punktu obsługi pomiędzy najbliższymi punktami (nie sąsiednimi)	203

Rysunek 7.13a. Układ dwóch tras przed wymianą punktów obsługi	204
Rysunek 7.13b. Układ dwóch tras po wymianie punktów obsługi	204
Rysunek 7.14a. Układ dwóch tras przed krzyżowaniem	205
Rysunek 7.14b. Układ dwóch tras po krzyżowaniu	205
Rysunek 7.15a. Układ tras przed transferem cyklicznym	206
Rysunek 7.15b. Układ tras po przeprowadzeniu transferu cyklicznego dla $b = 3$ i $k = 2$	206
Rysunek 7.16. Wykres zależności prawdopodobieństwa δE od różnicy wartości energii cząsteczki	208
Rysunek 7.17. Wykres zależności prawdopodobieństwa δE od wartości temperatury	209
Rysunek 7.18. Algorytm symulowanego wyżarzania	210
Rysunek 7.19. Wykres spadku temperatury dla zależności określonej wzorem (7.5)	211
Rysunek 7.20. Wykres spadku temperatury dla zależności określonej wzorem (7.6)	211
Rysunek 7.21. Wykres spadku temperatury dla zależności określonej wzorami (7.7) i (7.8)	212
Rysunek 7.22. Algorytm symulowanego wyżarzania Z. Wanga, X. Genga i Z. Shao	214
Rysunek 7.23. Algorytm symulowanego wyżarzania S. Afifiego, D.-C. Danga i A. Moukrima	216
Rysunek 7.24a. Układ dwóch tras przed relokacją łańcucha (punktów obsługi)	218
Rysunek 7.24b. Układ dwóch tras po relokacji łańcucha (punktów obsługi)	218
Rysunek 7.25a. Układ dwóch tras przed wymianą łańcucha	219
Rysunek 7.25b. Układ dwóch tras po wymianie łańcucha	219
Rysunek 7.26. Algorytm przeszukiwania tabu	220
Rysunek 7.27. Algorytm przeszukiwania tabu wyższego poziomu C.-N. Fiechtera	226
Rysunek 7.28. Procedura intensyfikacji w algorytmie przeszukiwania tabu C.-N. Fiechtera	227
Rysunek 7.29. Procedura superprzejścia	228
Rysunek 7.30. Algorytm <i>Taburoute</i>	229
Rysunek 7.31. Procedura przeszukiwania tabu algorytmu <i>Taburoute</i>	230
Rysunek 7.32. Algorytm wieloagentowy	234

Rysunek 7.33. Zachowanie mrówek podążających do miejsca pożywienia	235
Rysunek 7.34. Algorytm mrówkowy (budowa przez mrówki całych tras)	236
Rysunek 7.35. Algorytm mrówkowy (przejścia mrówek do kolejnych punktów)	237
Rysunek 7.36. Algorytm mrówkowy B. Bullnheimera, R.F. Hartla i Ch. Strauss	243
Rysunek 7.37. Lista kandydatów w algorytmie mrówkowym	246
Rysunek 7.38. Algorytm mrówkowy B. Yu, Z.-Z. Yanga i B. Yao	247
Rysunek 7.39. Algorytm ewolucyjny	249
Rysunek 7.40. Kodowanie ścieżkowe w algorytmie PEWKOM	255
Rysunek 7.41. Reprezentacja GVR	255
Rysunek 7.42. Kodowanie za pomocą dwóch chromosomów	256
Rysunek 7.43. Kodowanie w algorytmie ewolucyjnym S.J. Louisa, X. Yina i Z.Y. Yuana	257
Rysunek 7.44a. Potrzebne dane wynikające z postaci chromosomu [1 2 3 4 5]. W nawiasach podane są popyty punktów obsługi	266
Rysunek 7.44b. Graf pomocniczy i najkrótsza droga	267
Tabela 1.1. Porównanie form transportu	26
Tabela 2.1. Pierwszy i drugi szczebel wieloetapowego zadania transportowego (pierwszy etap)	41
Tabela 2.2. Drugi i trzeci szczebel wieloetapowego zadania transportowego (drugi etap)	42
Tabela 2.3. Klasyczne zadanie transportowe dla trójszczebłowego problemu transportowego	42
Tabela 2.4. Ogólna postać klasycznego zadania transportowego dla wieloszczebłowego problemu transportowego	43
Tabela 2.5. Postać zadania transportowo-produkcyjnego	45
Tabela 2.6. Przykład dopuszczalnego rozwiązania zadania przydziału o n środkach i n celach	46
Tabela 3.1. Liczba możliwych rozwiązań dla zadania komiwojażera ($c_{ij} = c_{ji}$)	58
Tabela 4.1. Zbiór harmonogramów dla $T = 5$ i częstotliwości 2	99
Tabela 6.1. Odległości od bazy oraz popyty punktów obsługi	185
Tabela 6.2. Przykład liczbowy budowy klasycznego zadania transportowego w metodzie BF-WOT	185

Tabela 6.3. Rozwiązanie klasycznego zadania transportowego w metodzie BF-WOT (rejon obsługi pojazdu)	185
Tabela 7.1a. Struktura tabelaryczna listy tabu dla rozwiązań A–E	221
Tabela 7.1b. Struktura tabelaryczna listy tabu dla rozwiązań B–F	222
Tabela 7.2. Struktura tabelaryczna listy tabu dla metody 3-opt	223
Tabela 7.3. Przykład sposobu kodowania binarnego dla zadania układania tras pojazdów	251

Bibliografia

- Arabas J. (2001), *Wykłady z algorytmów ewolucyjnych*, Wydawnictwa Naukowo-Techniczne, Warszawa.
- Archetti C., Hertz A., Speranza M. (2006), *A tabu search algorithm for the split delivery vehicle routing problem*, „Transport Science”, Vol. 40, s. 64–73.
- Afifi S., Dang D.-C., Moukrim A. (2013), *A simulated annealing algorithm for the vehicle routing problem with time windows and synchronization constraints*, 7th International Conference, Learning and Intelligent Optimization (LION 7), Catania, Italy, s. 259–265.
- Balinski M., Quandt R. (1964), *On an integer program for a delivery problem*, „Operations Research”, Vol. 52, s. 723–738.
- Baran B., Schaerer M. (2003), *A multiobjective ant system for vehicle routing problem with time windows*, Proceedings of the 21st IASTED International Conference Applied Informatics, s. 97–102.
- Baraniecka A. (2004), *ECR Efficient Consumer Response. Łańcuch dostaw zorientowany na klienta*, Instytut Logistyki i Magazynowania, Poznań.
- Bard J.F., Huang L., Dror M., Jaillet P. (1998), *A branch and cut algorithm for the VRP with satellite facilities*, „IIE Transactions”, Vol. 30, s. 821–834.
- Baker B.M., Ayeche M.A. (2003), *A genetic algorithm for the vehicle routing problem*, „Computers & Operations Research”, Vol. 30, s. 787–800.
- Basu S. (2012), *Tabu search implementation on traveling salesman problem and its variations: A literature survey*, „American Journal of Operations Research”, Vol. 2, s. 163–173.
- Beasley J.-E. (1983), *Route-first cluster-second methods for vehicle routing*, „OMEGA International Journal of Management Science”, Vol. 11, s. 403–408.
- Beasley J.-E., Tan C.C.R. (1984), *A heuristic algorithm for the period vehicle routing problem*, „OMEGA International Journal of Management Science”, Vol. 12, s. 497–504.
- Bell J.E., McMullen P.R. (2004), *Ant colony optimization techniques for the vehicle routing problem*, „Advanced Engineering Informatics”, Vol. 18, s. 41–48.

- Beltrami E.L., Bodin L.D. (1974), *Networks and vehicle routing for municipal waste pollution*, „Networks”, Vol. 4, s. 65–94.
- Berbeglia G., Cordeau J.-F., Gribkovskaia I., Laporte G. (2007), *Static pickup and delivery problems: A classification scheme and survey*, „Top”, Vol. 15, s. 1–31.
- Bertsimas D.J. (1992), *A vehicle routing problem with stochastic demand*, „Operation Research”, Vol. 40, s. 574–585.
- Blaik P. (2001), *Logistyka*, Polskie Wydawnictwo Ekonomiczne, Warszawa.
- Błażewicz J. (1986), *Problemy optymalizacji kombinatorycznej – złożoność obliczeniowa, algorytmy aproksymacyjne*, Polskie Wydawnictwo Ekonomiczne, Warszawa.
- Boryczka U. (2006), *Algorytmy optymalizacji mrowiskowej*, Wydawnictwo Uniwersytetu Śląskiego, Katowice.
- Bozarth C., Handfield R.B. (2007), *Wprowadzenie do zarządzania operacjami i łańcuchami dostaw*, Wydawnictwo Helion, Gliwice.
- Bramel J., Simchi-Levi D. (1995), *A location based heuristic for general routing problems*, „Operation Research”, Vol. 43, No. 4, s. 649–660.
- Branchini R.M., Armentano V.A., Loekketangen A. (2009), *Adaptive granular local search heuristic for a dynamic vehicle routing problem*, „Computers & Operations Research”, Vol. 36, s. 2955–2968.
- Bullnheimer B., Hartl R.F., Strauss C. (1997), *Applying the ANT system to the vehicle routing problem*, [w:] S. Vos, S. Martello, I.H. Osman, C. Roucairol (red.), *Meta-heuristics. Advances and trends in local search paradigms for optimization*, Kluwer Academic Publishers, Boston, s. 285–296.
- Bullnheimer B., Hartl R.F., Strauss C. (1999a), *An improved ant system algorithm for the vehicle routing problem*, „Annals of Operations Research”, Vol. 89, s. 319–328.
- Bullnheimer B., Hartl R.F., Strauss C. (1999b), *A new rank based version of the ant system – A computational study*, „Central European Journal of Operations Research”, Vol. 7, s. 25–38.
- Całczyński A. (1979), *Modele i metody ustalania tras przewozów towarowych*, Instytut Handlu Wewnętrznego i Rynku, Warszawa.
- Całczyński A. (1992), *Metody optymalizacji w obsłudze transportowej rynku*, Państwowe Wydawnictwo Ekonomiczne, Warszawa.
- Całczyński A., Kędzierska-Stróż D., Orzechowska D., Śleszyński Z. (2000), *Elementy badań operacyjnych w zarządzaniu*, Wydawnictwo Politechniki Radomskiej, Radom.
- Carpaneto G., Toth P. (1980), *Some new branching and bounding criteria for the asymmetric travelling salesman problem*, „Management Science”, Vol. 26, s. 736–743.
- Chatterjee S., Carrera C., Lynch L.A. (1996), *Genetic algorithms and traveling salesman problems*, „European Journal of Operational Research”, Vol. 93, s. 490–510.

- Chen S., Golden B., Wasil E. (2007), *The split delivery vehicle routing problem: Applications algorithms, test problems and computational results*, „Networks”, Vol. 49, s. 318–329.
- Christiansen Ch.H., Lysgaard J. (2007), *A branch and price algorithm for the capacitated vehicle routing problem with stochastic demands*, „Operations Research Letters”, Vol. 35, s. 773–781.
- Christofides N. (1976), *Worst case analysis of a new heuristic for the travelling salesman problem*, Report 388, Graduate School of Industrial Administration, Carnegie Mellon University.
- Christofides N., Beasley J.-E. (1984), *The period routing problem*, „Networks”, Vol. 14, s. 237–256.
- Christofides N., Eilon S. (1972), *Algorithms for large scale traveling salesman problems*, „Operations Research Quarterly”, Vol. 23, s. 511–518.
- Christofides N., Mingozzi A., Toth P. (1979), *The vehicle routing problem*, [w:] N. Christofides, A. Mingozzi, P. Toth, C. Sandi (red.), *Combinatorial optimization*, Wiley, Chichester, s. 431–448.
- Christofides N., Mingozzi A., Toth P. (1981), *Exact algorithms for the vehicle routing problem based on spanning tree and shortest path relaxations*, „Mathematical Programming”, Vol. 20, s. 255–282.
- Christopher M. (2000), *Logistyka i zarządzanie łańcuchem dostaw*, Polskie Centrum Doradztwa Logistycznego, Warszawa.
- Clark G., Wright J.W. (1964), *Scheduling of vehicles from a central depot to a number of delivery points*, „Operations Research”, Vol. 12, s. 568–581.
- Colorni A., Dorigo M., Maniezzo V. (1991), *Distributed optimization by ant colonies*, [w:] F.J. Varela, P. Bourguine (red.), *Proceedings of European Conference Artificial Life (ECAL '91)*, Elsevier Publishing, s. 134–142.
- Colorni A., Dorigo M., Maniezzo V., Trubian M. (1994), *Ant system for job-shop scheduling*, „JORBEL Belgian Journal of Operations Research. Statistics and Computer Science”, Vol. 34, s. 39–53.
- Contardo C., Martinelli R. (2014), *A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints*, „Discrete Optimization”, Vol. 12, s. 129–146.
- Coyle J.J., Bardi E.J., Langley Jr. C.J. (2002), *Zarządzanie logistyczne*, Polskie Wydawnictwo Ekonomiczne, Warszawa.
- Crevier B., Cordeau J.-F., Laporte G. (2007), *The multi-depot vehicle routing problem with inter-depot routes*, „European Journal of Operational Research”, Vol. 176, s. 756–773.
- Cytowski J. (1996), *Algorytmy genetyczne. Podstawy i zastosowania*, Akademicka Oficyna Wydawnicza PLJ, Warszawa.
- Dantzig G.B., Fulkerson D.R., Johnson S.M. (1954), *Solution of a large-scale traveling salesman problem*, „Operations Research”, Vol. 2, s. 393–410.

- Dantzig G.B., Ramser J.H. (1959), *Truck dispatching problem*, „Management Science”, Vol. 6, s. 80–91.
- Desaulniers G., Desrosiers J., Erdmann A., Solomon M.M., Soumis F. (2002), *VRP with pickup and delivery*, [w:] P. Toth, D. Vigo (red.), *The vehicle routing problem*, Society for Industrial and Applied Mathematics, Philadelphia.
- Desrochers M., Laporte G. (1991), *Improvements and extensions of the Miller-Tucker-Zemlin subtour elimination constraints*, „Operations Research Letters”, Vol. 10, s. 27–36.
- Desrochers M., Desrosiers J., Solomon M.M. (1992), *A new optimization algorithm for the vehicle routing problem with time windows*, „Operations Research”, Vol. 40, s. 342–354.
- Desrochers M., Lenstra J.K., Savelsbergh M.W.P., Soumis F. (1988), *Vehicle routing with time windows: Optimization and approximation*, [w:] B.L. Golden, A.A. Assad (red.), *Vehicle routing. Method and studies*, North-Holland, Amsterdam, s. 65–84.
- Doerner K., Gronalt M., Hartl R.F., Reimann M., Strauss Ch., Stummer M. (2002), *Savings ants for the vehicle routing problem*, [w:] S. Cagnosni (red.), *EvoWorkshops02*, Springer-Verlag, Berlin, s. 11–20.
- Dorigo M., Gambardella L.M. (1997), *Ant colony system: A cooperative learning approach to the traveling salesman problem*, „IEEE Transactions on Evolutionary Computation”, Vol. 1, s. 53–66.
- Dorigo M., Maniezzo V., Colorni A. (1996), *The ant system: Optimization by a colony of cooperating agents*, „IEEE Transactions on Systems, Man and Cybernetics”, Part B 26, s. 1–13.
- Dror M., Laporte G., Trudeau P. (1994), *Vehicle routing with split deliveries*, „Discrete Applied Mathematics”, Vol. 50, s. 239–254.
- Dumas Y., Desrosiers J., Soumis F. (1991), *The pickup and delivery problem with time windows*, „European Journal of Operational Research”, Vol. 54, s. 7–22.
- Duncan T. (1995), *Experiments in the use of neighborhood search techniques for vehicle routing*, Report AIAI-TR-176, University of Edinburgh, Edinburgh.
- Edmonds J., Johnson E.L. (1973), *Matching, Euler tours and the Chinese postman*, „Mathematical Programming”, Vol. 5, s. 88–124.
- Fechner I. (2007), *Zarządzanie łańcuchem dostaw*, Wyższa Szkoła Logistyki, Poznań.
- Feillet D., Dejax P., Gendreau M. (2005), *Traveling salesman problem with profits*, „Transportation Science”, Vol. 39, s. 188–205.
- Fertsch M. (red.) (2006), *Słownik terminologii logistycznej*, Instytut Logistyki i Magazynowania, Poznań.
- Fiechter C.-N. (1994), *A parallel tabu search algorithm for large traveling salesman problems*, „Discrete Applied Mathematics”, Vol. 51, s. 243–267.
- Finkelsztejn J.J., Korbut A.A. (1974), *Programowanie dyskretne*, Państwowe Wydawnictwo Naukowe, Warszawa.

- Fisher M., Jaikumar R. (1981), *A generalized assignment heuristic for the vehicle routing*, „Networks”, Vol. 11, s. 109–124.
- Ford L.R., Fulkerson D.R. (1962), *Flows in networks*, Princeton University Press, Princeton–New York.
- Francis P.M., Smilowitz K.R. (2006), *Modeling techniques for periodic vehicle routing problems*, „Transportation Research”, Part B 40, s. 872–884.
- Francis P.M., Smilowitz K.R., Tzur M. (2006), *The period vehicle routing problems*, „Transportation Science”, Vol. 40, s. 439–454.
- Gauvin Ch., Desaulniers G., Gendreau M. (2014), *A branch-cut-and-price algorithm for the vehicle routing problem with stochastic demands*, „Computers & Operations Research”, Vol. 50, s. 141–153.
- Geiger M.J. (2001), *Genetic algorithms for multiple objective vehicle routing*, Metaheuristic International Conference 2001 (MIC 2001), s. 348–353.
- Gendreau M., Hertz A., Laporte G. (1992), *New insertion and postoptimization procedures for the travelling salesman problem*, „Operations Research”, Vol. 40, s. 1086–1094.
- Gendreau M., Hertz A., Laporte G. (1994), *A tabu search heuristic for the vehicle routing problem*, „Management Science”, Vol. 40, s. 1276–1290.
- Gendreau M., Laporte G., Seguin R. (1996), *Stochastic vehicle routing*, „European Journal of Operational Research”, Vol. 88, s. 3–12.
- Gendreau M., Laporte G., Musaraganyi Ch., Taillard E.D. (1999), *A tabu search heuristic for the heterogeneous fleet vehicle routing problem*, „Computers & Operations Research”, Vol. 26, s. 1153–1173.
- Gillett B.E., Miller L.R. (1974), *A heuristic algorithm for the vehicle dispatch problem*, „Operations Research”, Vol. 22, s. 340–349.
- Glover F. (1989), *Tabu search – part I*, „ORSA Journal on Computing”, Vol. 1, s. 190–206.
- Goldberg D.E. (2003), *Algorytmy genetyczne i ich zastosowania*, Wydawnictwa Naukowo-Techniczne, Warszawa.
- Golden B.L., Magnanti T.L., Nguyen H.Q. (1997), *Implementing vehicle routing algorithms*, „Networks”, Vol. 7, s. 113–148.
- Grefenstette J. (1987), *Incorporating problem specific knowledge into genetic algorithms*, [w:] L. Davis (red.), *Genetic algorithms and simulated annealing*, Morgan Kaufmann, San Francisco, s. 42–60.
- Grefenstette J., Gopal R., Rosmaita B., Van Gucht D. (1985), *Genetic algorithms for the traveling salesman problem*, Proceedings of the First International Conference on Genetic Algorithms and Their Applications, New Jersey, s. 160–165.
- Gulczynski D., Golden B., Wasil E. (2011), *The period vehicle routing problem with service choice*, „Transportation Research”, Part E 47, s. 648–668.
- Hanghau A., Jung S. (2005), *A dynamic vehicle routing problem with dependent travel times*, „Computers & Operations Research”, Vol. 32, s. 2959–2986.

- Harmanani H., Azar D., Helal N., Keirouz W. (2011), *A simulated annealing algorithm for the capacitated vehicle routing problem*, 26th International Conference on Computers and Their Applications (CATA 2011), New Orleans, USA.
- Held M., Karp R.M. (1971), *The travelling salesman problem and minimum spanning trees: Part II*, „Mathematical Programming”, Vol. 1, s. 6–25.
- Helsgaun K. (2000), *An effective implementation of the Lin-Kernighan traveling salesman heuristic*, „European Journal of Operational Research”, Vol. 126, s. 106–130.
- Helsgaun K. (2009), *General k-opt submoves for the Lin-Kernighan TSP heuristic*, „Mathematical Programming Computation”, Vol. 1, s. 119–163.
- Hjorring C. (1995), *The vehicle routing problems and local search metaheuristics*, Ph.D., Department of Engineering Science, The University of Auckland, USA.
- Ho W., Ho G.T.S., Ji P., Lau H.C.W. (2008), *A hybrid genetic algorithm for multi-depot vehicle routing problem*, „Engineering Applications of Artificial Intelligence”, Vol. 21, s. 548–557.
- Hong S.-Ch., Park Y.-B. (1999), *A heuristic for bi-objective vehicle routing with time window constraints*, „International Journal of Production Economics”, Vol. 62, s. 249–258.
- Ignasiak E. (red.) (1996), *Badania operacyjne*, Polskie Wydawnictwo Ekonomiczne, Warszawa.
- Jacyna M. (2009), *Wybrane zagadnienia modelowania systemów transportowych*, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa.
- Jadczak R. (2001), *Zastosowanie zadań rozwózkowo-zwózkowych w transporcie mleczarskim*, [w:] D. Kopańska-Bródka (red.), *Metody i zastosowania badań operacyjnych*, Prace Naukowe Akademii Ekonomicznej w Katowicach, Katowice.
- Jadczak R. (2005), *Rozwiązywanie zagadnień układania tras pojazdów z wykorzystaniem algorytmów ewolucyjnych*, „Badania Operacyjne i Decyzje”, nr 3–4, s. 7–22.
- Jadczak R. (2006), *Zastosowanie algorytmów genetycznych w zadaniach CVRP*, [w:] J.B. Gajda, R. Jadczak, *Badania operacyjne. Przykłady zastosowań*, Wydawnictwo Uniwersytetu Łódzkiego, Łódź, s. 89–106.
- Jadczak R. (2010), *Heurystyki i metaheurystyki w problemach VRP*, Prace Naukowe Uniwersytetu Ekonomicznego we Wrocławiu Nr 108, Wrocław, s. 101–112.
- Jadczak R. (2014), *Traveling salesman problem: Approach to optimality*, „Entrepreneurship and Human Resources in Management”, Vol. 15, s. 157–169.
- Jadczak R. (2015), *Wpływ elementów algorytmu ewolucyjnego na proces poszukiwania rozwiązania optymalnego*, [w:] J.B. Gajda, R. Jadczak, *Badania operacyjne w praktyce*, Wydawnictwo Uniwersytetu Łódzkiego, Łódź, s. 123–134.
- Jaskiewicz A., Kominek P. (2003), *Genetic local search with distance preserving recombination operator for a vehicle routing problem*, „European Journal of Operational Research”, Vol. 151, s. 352–364.

- Jozefowiec N., Semet F., Talbi E.-G. (2008), *Multi-objective vehicle routing problems*, „European Journal of Operational Research”, Vol. 189, s. 293–309.
- Karp R.M. (1979), *A patching algorithm for the nonsymmetric traveling salesman problem*, „SIAM Journal on Computing”, Vol. 8, s. 561–573.
- Khoshbakht M.Y., Sedighpour M. (2011), *An optimization algorithm for the capacitated vehicle routing problem based on ant colony system*, „Australian Journal of Basic and Applied Sciences”, Vol. 5, s. 2729–2737.
- Kindervater G.A.P., Savelsbergh M.W.P. (1997), *Vehicle routing: Handling edge exchanges*, [w:] E.H.L. Aarts, J.K. Lenstra (red.), *Local search in combinatorial optimization*, Wiley, Chichester, s. 337–360.
- Kirkpatrick S., Gelatt C.D., Vecchi M.P. (1983), *Optimization by simulated annealing*, „Science”, Vol. 220, s. 671–680.
- Koźlak A. (2010), *Ekonomika transportu*, Wydawnictwo Uniwersytetu Gdańskiego, Gdańsk.
- Krawczyk S. (1996), *Metody ilościowe w logistyce (przedsiębiorstwa)*, Wydawnictwo C.H. Beck, Warszawa.
- Kruskal J.B. (1956), *On the shortest spanning subtree of a graph and the traveling salesman problem*, „Proceedings of the American Mathematical Society”, Vol. 7, s. 48–50.
- Kukuła K. (1996), *Badania operacyjne w przykładach i zadaniach*, Państwowe Wydawnictwo Naukowe, Warszawa.
- Laporte G., Louveaux F.V. (1998), *Solving stochastic routing problems with the integer L-shaped method*, [w:] T.G. Crainic, G. Laporte (red.), *Fleet management and logistics*, Centre for Research on Transportation, s. 159–167.
- Laporte G., Louveaux F.V., Mercure H. (1992), *The vehicle routing problems with stochastic travel times*, „Transportation Science”, Vol. 26, s. 161–170.
- Laporte G., Mercure H., Norbert Y. (1986), *An exact algorithm for the asymmetrical capacitated vehicle routing problem*, „Networks”, Vol. 16, s. 33–46.
- Laporte G., Norbert Y., Desrochers M. (1985), *Optimal routing under capacity and distance restrictions*, „Operations Research”, Vol. 33, s. 1050–1073.
- Laporte G., Norbert Y., Taillefer S. (1987), *A branch-and-bound algorithm for asymmetrical distance-constrained vehicle routing problem*, „Mathematical Modelling”, Vol. 9, No. 12, s. 857–868.
- Larrañaga P., Kuijpers C.M.H., Muraga R.H., Inza I., Dizdarevic S. (1999), *Genetic algorithms for the traveling salesman problem: A review of representations and operators*, „Artificial Intelligence Review”, Vol. 13, s. 129–170.
- Larsen A., Madsen O.B.G., Solomon M.M. (2004), *The a priori dynamic traveling salesman problem with time windows*, „Transportation Science”, Vol. 38, s. 459–472.
- Lee T.-R., Ueng J.-H. (1999), *A study of vehicle routing with load balancing*, „International Journal of Physical Distribution & Logistics Management”, Vol. 29, s. 646–657.

- Lenstra J.K., Rinnooy Kan A.H.G. (1975), *Some simple applications of the traveling salesman problem*, „Operations Research Quarterly”, Vol. 26, s. 717–734.
- Li Ch., Simchi-Levi D. (1990), *Worst-case analysis of heuristics for multidepot capacitated vehicle routing problems*, „ORSA Journal on Computing”, Vol. 2, No. 1, s. 64–73.
- Lin S. (1965), *Computer solutions of the traveling salesman problem*, „Bell System Computer Journal”, Vol. 44, s. 2245–2269.
- Lin S., Kernighan B.W. (1973), *An effective heuristic algorithm for traveling-salesman problem*, „Operations Research”, Vol. 21, No. 1, s. 498–516.
- Little J.D.C., Murty K.G., Sweeney D.W., Karel C. (1963), *An algorithm for the traveling salesman problem*, „Operations Research”, Vol. 11, s. 863–1025.
- Louis S.J., Yin X., Yuan Z.Y. (1999), *Multiple vehicle routing with time windows using genetic algorithms*, Proceedings of the Congress of Evolutionary Computation (CEC-99), s. 1804–1808.
- Lund K., Madsen O.B.G., Rygaard J.N. (1996), *Vehicle routing problems with varying degrees of dynamism*, Technical Report, IMM, The Department of Mathematical Modelling, Technical University of Denmark.
- Łupicka A. (2006), *Sieci logistyczne. Teorie. Modele. Badania*, Wydawnictwo Akademii Ekonomicznej w Poznaniu, Poznań.
- Machado P., Tavares J., Pereira F.B., Costa E. (2002), *Vehicle routing problem: Doing it evolutionary way*, Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2002).
- Mak K.-T., Morton A.J. (1993), *A modified Lin-Kernighan traveling-salesman heuristic*, „Operations Research Letters”, Vol. 13, s. 127–132.
- Mazzeo S., Loiseau I. (2004), *An ant colony algorithm for the capacitated vehicle routing*, „Electronic Notes in Discrete Mathematics”, Vol. 18, s. 181–186.
- Mendyk E. (2009), *Ekonomika transportu*, Wyższa Szkoła Logistyki, Poznań.
- Michalewicz Z. (1999), *Algorytmy genetyczne + struktury danych = programy ewolucyjne*, Wydawnictwa Naukowo-Techniczne, Warszawa.
- Miller C.E., Tucker A.W., Zemlin R.A. (1960), *Integer programming formulations and traveling salesman problems*, „Journal of the Association for Computing Machinery”, Vol. 7, s. 326–329.
- Miller D.L., Pekny J.F. (1991), *Exact solution of large asymmetric traveling salesman problems*, „Science”, Vol. 251, s. 754–761.
- Miszczyńska D., Miszczyński M. (1997), *Wybrane metody badań operacyjnych*, Wyższa Szkoła Ekonomiczno-Humanistyczna w Skierniewicach, Skierniewice.
- Mole R.H., Jameson S.R. (1976), *A sequential route building algorithm employing a generalized savings criterion*, „Operational Research Quarterly”, Vol. 27, s. 503–511.
- Montoya-Torres J.R., Franco J.L., Isaza S.N., Jiménez H.F., Padilla N.H. (2015), *A literature review on the vehicle routing problem with multiple depots*, „Computers & Industrial Engineering”, Vol. 79, s. 115–129.

- Mourgaya M., Nauderbeck F. (2006), *Column generation based heuristic for tactical planning in multi-period vehicle routing*, „European Journal of Operational Research”, Vol. 183, s. 1028–1041.
- Norbert Y., Picard J.-C. (1996), *An optimal algorithm for mixed Chinese postman problem*, „Networks”, Vol. 27, s. 95–108.
- Or I. (1976), *Traveling salesman-type combinatorial problems and their relation to the logistics of regional blood banking*, Northwestern University, Evanston.
- Osman I.H. (1993), *Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem*, „Annals of Operations Research”, Vol. 41, s. 421–451.
- Paessens H. (1988), *The savings for the vehicle routing problem*, „European Journal of Operational Research”, Vol. 34, s. 336–344.
- Papamitriou C.H. (2002), *Złożoność obliczeniowa*, Wydawnictwa Naukowo-Techniczne, Warszawa.
- Pillac V., Gendreau M., Gueret Ch., Medaglia A.L. (2013), *A review of dynamic vehicle routing problems*, „European Journal of Operational Research”, Vol. 225, s. 1–11.
- Potvin J.-Y. (1996), *Genetic algorithms for the traveling salesman problem*, „Annals of Operations Research”, Vol. 63, s. 339–370.
- Prins Ch. (2004), *A simple and effective evolutionary algorithm for the vehicle routing problem*, „Computers & Operations Research”, Vol. 31, s. 1985–2002.
- Psfartis H.N. (1995), *Dynamic vehicle routing: Status and prospects*, „Annals of Operations Research”, Vol. 61, s. 143–164.
- Radzikowski W. (1997), *Badania operacyjne w zarządzaniu przedsiębiorstwem*, Toruńska Szkoła Zarządzania, Toruń.
- Renaud J., Boctor F.F., Laporte G. (1996), *An improved petal heuristic for the vehicle routing problem*, „Journal of Operational Research Society”, Vol. 47, s. 329–336.
- Ribeiro R., Lourenço H.R. (2001), *A multi-objective model for a multi-period distribution management problem*, Proceedings of the First International Conference on Integrated Logistics, Singapore, s. 463–471.
- Riera-Ledesman J., Salazar-Gonzales J.J. (2005), *The biobjective traveling purchaser problem*, „European Journal of Operational Research”, Vol. 160, s. 599–613.
- Rosenkrantz D.J., Stearns R.E., Lewis P.M. (1977), *An analysis of several heuristics for the traveling salesman problem*, „SIAM Journal on Computing”, Vol. 6, s. 563–581.
- Ross K.A., Wright C.R.B. (1996), *Matematyka dyskretna*, Państwowe Wydawnictwo Naukowe, Warszawa.
- Rutkowski K. (red.) (2005), *Logistyka dystrybucji. Specyfika. Tendencje rozwojowe. Dobre praktyki*, Szkoła Główna Handlowa, Warszawa.
- Ryan D.M., Hjorring C., Glover F. (1993), *Extensions of the petal method for vehicle routing*, „Journal of Operational Research Society”, Vol. 44, s. 289–296.

- Rydzkowski W., Wojewódzka-Król K. (red.) (2009), *Transport. Problemy transportu w rozszerzonej UE*, Wydawnictwo Naukowe PWN, Warszawa.
- Savelsbergh M.W.P. (1985), *Local search in routing problems with time windows*, „Annals of Operations Research”, Vol. 4, s. 285–305.
- Savelsbergh M.W.P., Sol M. (1995), *The general pickup and delivery problem*, „Transportation Science”, Vol. 29, s. 17–29.
- Solich R. (1974), *Pewne uogólnienia zadania komiwojażera*, Prace Centrum Obliczeniowego PAN, Nr 141.
- Stajniak M., Foltyński M., Hajdul M., Krupa A. (2008), *Transport i spedycja*, Instytut Logistyki i Magazynowania, Poznań.
- Stützle T., Holger H.H. (2000), *MAX-MIN ant system*, „Future Generation Computer Systems”, Vol. 16, s. 889–914.
- Surekha P., Sumathi S. (2011), *Solution to multi-depot vehicle routing problem using genetic algorithm*, „World Applied Programming”, Vol. 1, No. 3, s. 118–131.
- Syso M.M. (1984), *Optymalizacja kombinatoryczna*, [w:] T. Kasprzak (red.), *Optymalizacja dyskretna. Zastosowania ekonomiczne*, Państwowe Wydawnictwo Ekonomiczne, Warszawa.
- Syso M.M., Deo N., Kowalik J.S. (1995), *Algorytmy optymalizacji dyskretnej*, Wydawnictwo Naukowe PWN, Warszawa.
- Szałek B.Z. (1992), *Logistyka: wstęp do problematyki*, Wydawnictwo Naukowe Uniwersytetu Szczecińskiego, Szczecin.
- Taillard E.D. (1993), *Parallel iterative search methods for vehicle routing problems*, „Networks”, Vol. 23, s. 661–673.
- Tarkowski J., Irestahl B., Lumsden K. (1995), *Transport – logistyka. Towary, informacja, środki*, Instytut Logistyki i Magazynowania, Poznań.
- Tas D., Dellaert N., Woensel van T., Kok de T. (2013), *Vehicle routing problem with stochastic travel times including soft time windows and service costs*, „Computers & Operations Research”, Vol. 40, s. 214–224.
- Tas D., Gendreau M., Dellaert N., Woensel van T., Kok de T. (2014), *Vehicle routing with soft time windows and stochastic travel times: A column generation and branch and price solution approach*, „European Journal of Operational Research”, Vol. 236, s. 789–799.
- Thompson P.M., Psaraftis H.N. (1993), *Cyclic transfer algorithms for multivehicle routing and scheduling problems*, „Operations Research”, Vol. 41, s. 935–946.
- Toth P., Vigo D. (2003), *The granular tabu search and its application to the vehicle-routing problem*, „INFORMS Journal on Computing”, Vol. 15, s. 333–346.
- Trojanowski K. (2005), *Metaheurystyki praktycznie*, Wydawnictwo Wyższej Szkoły Informatyki Stosowanej i Zarządzania, Warszawa.
- Trzaskalik T. (2003), *Wprowadzenie do badań operacyjnych z komputerem*, Polskie Wydawnictwo Ekonomiczne, Warszawa.

- Van Breedam A. (1994), *An analysis of the behavior of heuristics for the vehicle routing problem for a selection of problems with vehicle-related, Customer-related, and time-related constraints*, praca doktorska, Uniwersytet w Antwerpii.
- Van Breedam A. (1995), *Improvement heuristics for the vehicle routing problem based on simulated annealing*, „European Journal of Operational Research”, Vol. 86, s. 480–490.
- Van Breedam A. (2001), *Comparing descent heuristics and metaheuristics for the vehicle routing problem*, „Computers & Operations Research”, Vol. 28, s. 289–315.
- Wagner H. (1980), *Badania operacyjne*, Państwowe Wydawnictwo Ekonomiczne, Warszawa.
- Wang J., Zhou Y., Wang Y., Zhang J., Chen C.L.P., Zheng Z. (2015), *Multiobjective vehicle routing problems with simultaneous delivery and pickup and time windows: Formulation, instances and algorithms*, „IEEE Transactions on Cybernetics”, Vol. 46, No. 3, s. 582–594.
- Wang Z., Geng X., Shao Z. (2009), *An effective simulated annealing algorithm for solving the traveling salesman problem*, „Journal of Computational and Theoretical Nanoscience”, Vol. 6, s. 1680–1686.
- Whitley D., Starkweather T., Fuquay D. (1989), *Scheduling problems and traveling salesmen: The genetic edge recombination operator*, Proceedings 3rd International Conference on Genetic Algorithms (ICGA '89), George Mason University, Fairfax, s. 133–140.
- Whitley D., Starkweather T., Shaner D. (1991), *The traveling salesman and sequence scheduling: Quality solutions using genetic edge recombination*, [w:] L. Davis (red.), *Handbook of genetic algorithms*, Van Nostrand Reinhold, New York, s. 350–372.
- Wierzchoń S.T. (2001), *Sztuczne systemy immunologiczne. Teoria i zastosowania*, Akademicka Oficyna Wydawnicza EXIT, Warszawa.
- Xu J., Kelly J.P. (1998), *A network flow-based tabu search heuristic for the vehicle routing problem*, „Transportation Science”, Vol. 30, s. 379–393.
- Yu B., Yang Z.-Z., Yao B. (2009), *An improved ant colony optimization for vehicle routing problem*, „European Journal of Operational Research”, Vol. 196, s. 171–176.
- Zhang X., Wang J. (2012), *Hybrid ant algorithm and applications for vehicle routing problem*, 2012 International Conference on Solid State Devices and Materials Science, Physics Procedia 25, s. 1892–1899.

