

**POLITECHNIKA WARSZAWSKA**

**Wydział Elektroniki  
i Technik Informatycznych**

**ROZPRAWA DOKTORSKA**

mgr inż. Paweł Olender

**Zagadnienia lokalizacyjne ze złożonymi modelami preferencji**

Promotor  
prof. dr hab. Włodzimierz Ogryczak

Warszawa 2014



*Serdecznie dziękuję Panu Profesorowi Włodzimierzowi Ogryczakowi  
za cenne wskazówki, pomoc i wsparcie w kwestiach merytorycznych,  
a także za cierpliwość, wyrozumiałość i ogromną życzliwość.*

*Składam również podziękowania Panu Doktorowi Januszowi Granatowi  
za pomoc w organizacji pracy i przychylność na priorytetowe potraktowanie rozprawy.*

*Duże wyrazy wdzięczności kieruję do przyjaciół, rodziny i bliskich mi osób  
za wszechstronne wsparcie, cierpliwość oraz wyrozumiałość na ograniczenie mojego  
udziału w relacjach międzyludzkich na rzecz zaangażowania w relacje preferencji.*



## Streszczenie

Praca dotyczy dyskretnych wielokryterialnych problemów lokalizacyjnych, ze szczególnym uwzględnieniem modeli preferencji, które pozwalają uzyskiwać rozwiązania optymalne w sensie rozkładu ocen. Podejście takie pozwala osiągać rozwiązania spełniające warunek bezstronności, gdzie rozkład wartości ocen jest ważniejszy od ich przypisania do konkretnych kryteriów. Możliwe jest także generowanie rozwiązań sprawiedliwych w sensie efektywności wyrównującej.

Analizowane podejście wywodzi się od operatora uporządkowanej średniej ważonej (OWA). Prezentowane i proponowane są różne modele optymalizacyjne agregacji OWA i WOWA, zarówno ogólne w postaci zadań PCLM, jak i zadania PL dla specyficznych monotonicznych wag. Zaproponowane modele optymalizacyjne operatora WOWA pozwalają na uwzględnienie wielkości zapotrzebowań, zachowując przy tym własność bezstronności. Badana jest wydajność różnych zadań oraz możliwość jej poprawy za pomocą ograniczeń nadmiarowych. Rozwijana jest także przybliżona metoda rozwiązywania określana mianem przeszukiwania zmiennego sąsiedztwa (VNS). Prezentowane modyfikacje metody skracają wielokrotnie czas rozwiązania, a jednocześnie poprawiają jakość uzyskiwanych rozwiązań.

**Słowa kluczowe:** *optymalizacja wielokryterialna, dyskretne problemy lokalizacyjne, rozkład ocen, efektywność wyrównująca, operator OWA, operator WOWA, metoda VNS*

## Abstract

The dissertation concerns discrete multicriteria location problems and specially preference models that generate optimal solutions in the sense of outcomes distribution. Such an approach allows to obtain impartial solutions, where the distribution of outcomes is more important than their assignment to specific criteria. It is also possible to provide fairness as equitable efficiency.

Analyzed approach is derived from the Ordered Weighted Averaging (OWA) operator. Different OWA and WOWA optimization models are presented and proposed including both general MILP and specific LP formulations for suitable monotone weights. New optimization models for the Weighted Ordered Weighted Averaging (WOWA) operator allow to take into account demand requirements, while preserving the impartiality condition. The performance of different formulations is studied and influence of redundant constraints is analyzed. A heuristic method, so-called Variable Neighborhood Search (VNS), is also developed. The proposed modifications of VNS significantly reduce solution time and simultaneously improve quality of solutions.

**Keywords:** *multicriteria optimization, discrete location problems, ordered operators, outcomes distribution, equitable efficiency, OWA operator, WOWA operator, VNS method*



# Spis treści

<b>1. Wprowadzenie</b>	11
1.1. Sformułowanie problemu	11
1.2. Model preferencji	13
1.3. Motywacja	14
1.4. Cel pracy	15
<b>2. Modele preferencji</b>	17
2.1. Pareto-efektywność	17
2.2. Symetryczna efektywność	19
2.3. Wyrównująca efektywność	20
2.4. Techniki generacji rozwiązań	23
2.4.1. Agregacja uporządkowanych ocen	25
<b>3. Zagadnienia lokalizacyjne — stan sztuki</b>	27
3.1. Problem lokalizacyjny	27
3.2. Modele optymalizacyjne	29
3.2.1. Klasyczne funkcje agregujące	29
3.2.2. Kryterium centro-średniej	30
3.2.3. Porcja najgorszych ocen	30
3.2.4. Kryteria równościowe	33
3.2.5. Problem uporządkowanej średniej	34
3.3. Podsumowanie	34
<b>4. Uporządkowana średnia ważona</b>	35
4.1. Definicja operatora OWA	35
4.2. Parametryczny model preferencji	36
4.3. Złożoność obliczeniowa	40
<b>5. Modele optymalizacyjne OWA</b>	42
5.1. Model PCLM M1	42
5.2. Model PCLM M2	45
5.3. Zależności między modelami M1 i M2	46
5.4. Odchyleniowy model PL	47
5.5. Hybrydowy model PCLM	52
5.5.1. Model PCLM M3	52
5.5.2. Model PCLM M4	58
5.6. Projekt eksperymentów	59

5.7.	Wyniki . . . . .	60
5.7.1.	Wyniki modeli PCLM M1 i M2 . . . . .	62
5.7.2.	Wyniki odchyleniowego modelu PL i porównanie z modelem PCLM . . . . .	65
5.7.3.	Wyniki hybrydowych modeli PCLM M3 i M4 . . . . .	68
5.8.	Wnioski . . . . .	72
<b>6.</b>	<b>Wartościowana uporządkowana średnia ważona . . . . .</b>	<b>74</b>
6.1.	Klasyczna definicja operatora WOWA . . . . .	74
6.2.	Alternatywna definicja operatora WOWA . . . . .	76
<b>7.</b>	<b>Modele optymalizacyjne WOWA . . . . .</b>	<b>82</b>
7.1.	Model PL . . . . .	82
7.2.	Modele PCLM . . . . .	84
7.3.	Ograniczenia nadmiarowe . . . . .	85
7.4.	Projekt eksperymentów . . . . .	86
7.5.	Wyniki . . . . .	87
7.5.1.	Problemy z nierosnącymi wagami preferencji . . . . .	88
7.5.2.	Problemy z niemonotonicznymi lub niemalejącymi wagami preferencji . . . . .	91
7.6.	Wnioski . . . . .	94
<b>8.</b>	<b>Metoda przybliżona . . . . .</b>	<b>96</b>
8.1.	Metaheurystyka przeszukiwania zmiennego sąsiedztwa (VNS) . . . . .	96
8.1.1.	VNS dla kryterium porządkowego . . . . .	97
8.2.	Modyfikacje algorytmu wymiany . . . . .	99
8.2.1.	Relaksacja problemu . . . . .	99
8.2.2.	Modyfikacja sortowania . . . . .	101
8.2.3.	Warunek dominacji . . . . .	102
8.2.4.	Regularyzacja . . . . .	104
8.3.	Implementacja . . . . .	105
8.3.1.	Dodawanie punktu obsługi . . . . .	106
8.3.2.	Usuwanie punktu obsługi . . . . .	107
8.3.3.	Aktualizacja . . . . .	109
8.3.4.	Algorytm wymiany (przeszukiwanie lokalne) . . . . .	109
8.3.5.	Wycena przejścia . . . . .	111
8.3.6.	Przeszukiwanie zmiennego sąsiedztwa . . . . .	112
8.4.	Testy obliczeniowe . . . . .	113
8.4.1.	Problemy o niewielkim rozmiarze . . . . .	114
8.4.2.	Problemy o dużym rozmiarze . . . . .	121
8.5.	Wnioski . . . . .	139
<b>9.</b>	<b>Metoda VNS dla zróżnicowanych wag zapotrzebowania . . . . .</b>	<b>141</b>
9.1.	Metoda VNS z agregacją WOWA . . . . .	141
9.2.	Implementacja . . . . .	143



9.3. Testy obliczeniowe . . . . .	144
9.3.1. Problemy o niewielkim rozmiarze . . . . .	146
9.3.2. Problemy o dużym rozmiarze . . . . .	146
9.4. Wnioski . . . . .	154
<b>10. Podsumowanie . . . . .</b>	<b>156</b>
10.1. Wnioski . . . . .	156
10.2. Podsumowanie twórczego wkładu . . . . .	159
10.3. Dalsze kierunki badań . . . . .	161
<b>Bibliografia . . . . .</b>	<b>163</b>
<b>A. Wyniki modeli PCLM agregacji OWA . . . . .</b>	<b>168</b>
A.1. Wyniki modeli PCLM M1 i M2 agregacji OWA . . . . .	168
A.2. Wyniki modelu PL agregacji OWA . . . . .	172
A.3. Wyniki modeli PCLM M3 i M4 agregacji OWA . . . . .	176
<b>B. Wyniki modeli PCLM agregacji WOWA . . . . .</b>	<b>178</b>
B.1. Wyniki modelu PL agregacji WOWA . . . . .	178
B.2. Wyniki modeli PCLM MW1 i MW2 agregacji WOWA . . . . .	181
<b>C. Wyniki metody VNS z agregacją OWA . . . . .</b>	<b>187</b>
C.1. Wyniki metody VNS z agregacją OWA dla małych problemów . . . . .	187
<b>D. Wyniki metody VNS z agregacją WOWA . . . . .</b>	<b>197</b>
D.1. Wyniki metody VNS z agregacją WOWA dla małych problemów . . . . .	197



# 1. Wprowadzenie

Zagadnienia lokalizacyjno-dystrybucyjne należą do dziedziny badań operacyjnych i dotyczą odpowiedniego rozmieszczenia jednostek przestrzennych w taki sposób, aby spełnić określone założenia i cele. Typowym przykładem może być rozmieszczenie jednostek produkcyjnych uwzględniające położenie źródeł zaopatrzenia i rynków zbytu. Celem w tym przypadku może być minimalizacja kosztów dostarczenia surowców i dystrybucji produktów, czy też maksymalizacja zysku ze sprzedaży produktów u odbiorców, uwzględniająca koszty produkcji i transportu. Jak widać kluczowym elementem w problemach lokalizacyjnych są użyteczności, które wyrażają abstrakcyjne miary odległości między dostawcami a odbiorcami rozpatrywanych usług. Ze względu na charakter użyteczności im mniejsza wartość tym lepiej, dlatego celem jest ich minimalizacja. W przypadku, gdy poszczególni klienci (odbiorcy) są od siebie niezależni, poza globalną efektywnością liczy się także rozkład odległości dla danego schematu lokalizacyjnego [61], a jego sprawiedliwość (równość) staje się dodatkowym kryterium oceny otrzymanego rozwiązania. Takie podejście jest szczególnie istotne przy decyzjach dotyczących lokalizacji obiektów użyteczności publicznej, np. szkół, szpitali, centrów zarządzania kryzysowego, których klienci (obywatele) mają prawo do sprawiedliwego dostępu zgodnie z publicznymi regulacjami.

## 1.1. Sformułowanie problemu

Rzeczywiste problemy decyzyjne charakteryzują się zazwyczaj dużą złożonością. Aby możliwe było ich efektywne rozwiązywanie, potrzebne jest usystematyzowane podejście. W tym celu formułuje się modele decyzyjne, których zadaniem jest opis fragmentu rzeczywistości związanego z badanym zagadnieniem. Opis ten stanowi pewien przybliżony obraz rzeczywistego problemu, w którym uwzględniona jest jego idea i występujące w nim zależności. Zbudowanie modelu jest również konieczne, gdy do jego rozwiązania będzie wykorzystywany komputerowy (automatyczny) system wspomaganie decyzji. Model stanowi wówczas reprezentację problemu w systemie wspomaganie decyzji.

W pracy wykorzystywana jest analiza wielokryterialna i programowanie matematyczne. Podejście wielokryterialne pozwala na uwzględnienie preferencji poszczególnych użytkowników i uzyskanie kompromisowego rozwiązania. Programowanie matematyczne jest z kolei zarówno wygodną, jak i skuteczną metodą modelowania oraz rozwiązywania problemów optymalizacyjnych [90]. Z jednej strony zapewnia zwięzły i precyzyjny sposób modelowania badanych zagadnień, a z drugiej umożliwia ich efektywne rozwiązywanie. Praca dotyczy dyskretnych

problemów lokalizacyjnych, gdzie potencjalne położenia punktów obsługi stanowią zbiory skończone. Z tego względu stosowane jest głównie programowanie całkowitoliczbowe, którego metody osiągnęły znaczny stopień zaawansowania [28].

Zagadnienia lokalizacyjno-dystrybucyjne stanowią przykład problemów rozdziału szeroko rozumianych zasobów, w ramach których rozwijane są metody jak najlepszego wyboru (lokalizacji) i przydziału (alokacji) dostępnych zasobów [75]. W literaturze można znaleźć wiele modeli lokalizacyjnych i przykłady ich szerokiego zastosowania w praktyce [19], w tym studia przypadków rzeczywistych problemów [32, 14]. W pracy rozważany jest dyskretny problem lokalizacyjny, który można ogólnie sformułować w następujący sposób. Dany jest zbiór  $I$  jednostek przestrzennych (np. klientów) oraz zbiór potencjalnych lokalizacji, w których mają zostać umiejscowione obiekty do obsługi klientów. Zależność pomiędzy zbiorem klientów, a zbiorem potencjalnych lokalizacji obiektów jest dowolna, tzn. oba zbiory mogą być rozdzielne, tożsame lub mieć część wspólną będącą ich podzbiorem. W niniejszej pracy bez straty ogólności zakłada się tożsamość obu zbiorów. Problem sprowadza się do wyboru podzbioru lokalizacji zbioru  $I$ , w których umieszczone zostaną punkty obsługi do zaspokojenia zapotrzebowania klientów. Uzyskuje się w ten sposób zbiór  $Q$  możliwych schematów lokalizacyjnych (decyzji określających położenie obiektów podlegających rozmieszczeniu). Dla każdego klienta  $i \in I$  określona jest funkcja  $f_i(\mathbf{x})$  schematu lokalizacyjnego  $\mathbf{x}$ . Jest to indywidualna funkcja oceny, która wyraża wynik (skutek)  $y_i = f_i(\mathbf{x})$  danego rozmieszczenia dla klienta  $i$ . Zazwyczaj wartość ta jest utożsamiana z abstrakcyjnie zdefiniowaną odległością, co może w praktyce oznaczać odległość geograficzną, ale również koszt, czas, czy ogólne niezadowolenie klienta (odwrotność użyteczności). Z tego względu typowo przyjmuje się, że im mniejsza jest ta wartość tym lepiej, dlatego funkcja ta podlega minimalizacji. Problem przybiera więc postać wielokryterialnej optymalizacji

$$\min \{ \mathbf{f}(\mathbf{x}) : \mathbf{x} \in Q \}, \quad (1.1)$$

gdzie

$\mathbf{f} = (f_1, \dots, f_m)$  jest funkcją wektorową, która przekształca przestrzeń decyzji  $X$  (lokalizacji) na przestrzeń ocen  $Y$ ,

$Q \subset X$  stanowi zbiór decyzji dopuszczalnych (spełniających zadane ograniczenia),

$\mathbf{x} \in X$  oznacza decyzję wyrażoną poprzez wektor zmiennych decyzyjnych.

W problemie (1.1) minimalizacji podlega zatem wektor ocen  $\mathbf{y} = (y_1, \dots, y_m)$ , gdzie  $y_i = f_i(\mathbf{x})$ . Wektor  $\mathbf{y} = \mathbf{f}(\mathbf{x})$  nazywamy osiągalnym, jeśli stanowi ocenę rozwiązania dopuszczalnego  $\mathbf{x}$ . Zakładając, że wybór rozwiązania zależy wyłącznie od wektora ocen, problem (1.1) jest równoważny wyborowi najlepszego wektora ocen

$$\min \{ \mathbf{y} : \mathbf{y} \in A \}$$

ze zbioru osiągalnych wektorów ocen  $A = \{ \mathbf{y} \in Y : \mathbf{y} = \mathbf{f}(\mathbf{x}), \mathbf{x} \in Q \}$ .

Funkcje oceny poszczególnych klientów są między sobą przeciwstawne przy ich minimalizacji. Z tego względu zadanie (1.1) istotnie stanowi wielokryterialny problem decyzyjny, którego rozwiązanie wyraża pewnego rodzaju kompromis pomiędzy ocenami klientów. Sposób osiągnięcia tego kompromisu i jego charakter zależą od preferencji decydenta. Cechą charakterystyczną tak przedstawionego problemu jest fakt, że oceny poszczególnych użytkowników są ze sobą porównywalne (wyrażone w tych samych jednostkach). Jest to pewne uproszczenie w stosunku do ogólnej definicji problemu wielokryterialnego.

## 1.2. Model preferencji

W praktyce do rozwiązania problemów wielokryterialnych potrzebna jest jeszcze pewna koncepcja znalezienia i wyboru rozwiązania. Koncepcja ta określana jest mianem modelu preferencji. Model preferencji wskazuje, który z dwóch wektorów ocen jest lepszy, a tym samym pozwala na wybór odpowiadającego mu rozwiązania. Odpowiada to więc pewnej relacji preferencji określonej dla wektorów ocen. Relacja ścisłej preferencji jest zdefiniowana jako

$$y' \prec y'' \Leftrightarrow y' \text{ jest lepszy niż } y''.$$

Gdy dwa wektory ocen są jednakowo dobre, może to być wyrażone za pomocą relacji indyferencji

$$y' \cong y'' \Leftrightarrow y' \text{ jest równie dobry jak } y''.$$

W przypadku niektórych par wektorów ocen można stwierdzić jedynie, że jeden wektor nie jest gorszy niż drugi. Wyraża to relacja słabej preferencji

$$y' \preceq y'' \Leftrightarrow y' \text{ jest nie gorszy niż } y''.$$

W ogólności model preferencji nie musi być spójny (zupełny). To oznacza, że mogą istnieć pary wektorów ocen, które nie są ze sobą porównywalne — nie zachodzi dla nich żadna z powyższych relacji. Do kompletnego wyrażenia modelu preferencji wystarczająca jest w rzeczywistości relacja słabej preferencji [88], za pomocą której mogą być wyrażone dwie pozostałe relacje:

$$y' \prec y'' \Leftrightarrow (y' \preceq y'' \text{ i } y'' \not\preceq y'), \quad (1.2)$$

$$y' \cong y'' \Leftrightarrow (y' \preceq y'' \text{ i } y'' \preceq y'). \quad (1.3)$$

W dalszych rozważaniach model preferencji będzie więc utożsamiany z relacją słabej preferencji, czyli zbiorem relacji  $\preceq$ ,  $\prec$ ,  $\cong$  spełniającym warunki (1.2) i (1.3).

Modele preferencji definiuje się w oparciu o pożądane własności poszukiwanych rozwiązań. Z tych własności wynikają pewne warunki, które powinny spełniać modele preferencji. W pracy rozważane będą tzw. racjonalne relacje preferencji, czyli takie które spełniają 3 następujące

warunki:

— zwrotność

$$\mathbf{y} \preceq \mathbf{y}, \quad (1.4)$$

— przechodniość (tranzytywność)

$$(\mathbf{y}' \preceq \mathbf{y}'' \text{ i } \mathbf{y}'' \preceq \mathbf{y}''') \Rightarrow \mathbf{y}' \preceq \mathbf{y}''', \quad (1.5)$$

— ścisła monotoniczność

$$\mathbf{y} - \varepsilon \mathbf{e}_i \prec \mathbf{y} \text{ dla } \varepsilon > 0, \quad i = 1, \dots, m, \quad (1.6)$$

gdzie  $\mathbf{e}_i$  oznacza  $i$ -ty wektor jednostkowy w przestrzeni ocen  $Y$ .

**Definicja 1.1.** Relację preferencji  $\preceq$  nazywamy racjonalną relacją preferencji, jeśli spełnia warunki (1.4)–(1.6).

Jak wspomniano wcześniej, racjonalna relacja preferencji nie musi spełniać warunku spójności.

Z modelami preferencji ściśle związane jest pojęcie techniki generacji rozwiązań, które określa praktyczne metody znajdowania rozwiązań zgodnych z danymi relacjami preferencji. Jedną z podstawowych technik generacji rozwiązań problemów wielokryterialnych jest skalaryzacja. Polega ona na zdefiniowaniu funkcji skalaryzującej  $s: R^m \rightarrow R$ , która dokonuje agregacji ocen, co prowadzi do zadania optymalizacji jednokryterialnej. W przypadku minimalizacji rozwiązaniem optymalnym takiego zadania jest rozwiązanie o najmniejszej wartości funkcji skalaryzującej. Relacja preferencji jest więc zdefiniowana przez funkcję skalaryzującą, i to od własności tej funkcji zależą uzyskiwane rozwiązania.

W obszarze zagadnień lokalizacyjnych powstały liczne opracowania badające dyskretne problemy lokalizacyjne. Ich cechą charakterystyczną jest fakt, że wiele z nich różni się tylko funkcją skalaryzującą, czyli sposobem agregacji ocen. Dwie najczęściej stosowane agregacje to kryteria średniej i centrum. Są to podejścia w pewnym sensie antagonistyczne. Pierwsze dąży do globalnej efektywności schematu lokalizacyjnego wyrażającej się minimalną sumą (średnią) odległości, a z kolei drugie stara się zapewnić równość poprzez minimalizację maksymalnej odległości. Pomędzy tymi skrajnymi kryteriami znajduje się całe spektrum preferencji, które odpowiada różnego rodzaju rozwiązaniom kompromisowym. Wyznaczenie tych rozwiązań wymaga zastosowania bardziej złożonych modeli preferencji.

### 1.3. Motywacja

Motywacja do podjęcia prac nad złożonymi operatorami modelowania preferencji wynika z rozbieżności pomiędzy oczekiwaniami wobec rozwiązań, a rezultatami dostarczonymi przez systemy wspomaganie decyzji w praktyce. Złożoność rzeczywistych problemów decyzyjnych

wymaga zazwyczaj zastosowania modeli wielokryterialnych, do rozwiązania których potrzebne są funkcje agregujące. Ze względu na złożoność obliczeniową oraz łatwość interpretacji w praktyce stosowane są zazwyczaj najprostsze agregacje typu ważona średnia arytmetyczna czy optymalizacja najgorszej wartości. Nie zapewniają one jednak osiągnięcia rozwiązań o pożądanych, bardziej złożonych własnościach — nie pozwalają wyrażać bardziej złożonych preferencji. Przykładem tego typu własności jest sprawiedliwość w systemach z wieloma agentami, czy też odporność na zmiany w przypadku problemów uwzględniających niepewność otoczenia sytuacji decyzyjnej, co znajduje zastosowanie chociażby w zagadnieniach podziału zasobów i problemach lokalizacyjnych. Wymienione własności mogą być zapewnione przez użycie operatorów porządkowych, takich jak uporządkowana średnia ważona i jej rozszerzenia.

Modele preferencji bazujące na operatorach porządkowych pozwalają na osiągnięcie rozwiązań optymalnych w sensie rozkładu ocen, w tym rozwiązań sprawiedliwych. Uogólniają one także większość popularnych funkcji celu stosowanych w zagadnieniach lokalizacyjnych. Tym samym podejście to pozwala na jednolite potraktowanie wielu problemów lokalizacyjnych, stanowiąc elastyczne podejście do tej klasy problemów. Modele te pozwalają również na definiowanie nowych kryteriów, co prowadzi do nierozważanych wcześniej problemów.

Z drugiej strony uporządkowana średnia ważona nie pozwala na uwzględnienie wielkości zapotrzebowań dla konkretnych klientów, co jest istotnym wymaganiem w wielu zastosowaniach praktycznych. Nie opracowano do tej pory modeli dyskretnych problemów lokalizacyjnych, które pozwoliłyby na osiąganie rozwiązań optymalnych w sensie rozkładu ocen dla dowolnych wag preferencji przy jednoczesnym uwzględnieniu różnych wielkości zapotrzebowań. Opracowanie takiego podejścia znacznie zwiększyłoby elastyczność modelowania badanych zagadnień. Pozwoliłoby to rozszerzyć zakres praktycznych zastosowań opracowanych modeli i metod ich rozwiązywania.

Bezpośrednie zastosowanie operatorów porządkowych może jednak skutkować wzrostem złożoności obliczeniowej, który będzie nieakceptowalny dla problemów o rzeczywistych rozmiarach. Z tego powodu modelowanie tych operatorów musi być zrealizowane w sposób możliwie efektywny.

## 1.4. Cel pracy

Celem pracy jest rozwój i poprawa efektywności istniejących modeli oraz zaproponowanie nowych sformułowań z wykorzystaniem operatorów uporządkowanych średnich ważonych i ich rozszerzeń do operatorów wartościowanych uporządkowanych średnich ważonych, które umożliwiają uzyskanie rozwiązań o pożądanych własnościach dystrybucyjnych dla problemów lokalizacyjnych. Dla tak zdefiniowanych problemów poszukiwane również będą przybliżone metody rozwiązywania pozwalające na efektywne uzyskiwanie rozwiązań dobrej jakości w sensie rozkładu ocen. Szczegółowe cele pracy są następujące:

**Cel 1** Opracowanie parametrycznych modeli optymalizacji wielokryterialnej dla dyskretnych

zagadnień lokalizacyjnych, które uwzględniają zarówno preferencje efektywnościowo-sprawiedliwościowe, jak i zróżnicowane zapotrzebowania.

**Cel 2** Poprawa efektywności parametrycznych modeli optymalizacji wielokryterialnej w sensie najlepszego rozkładu ocen dla dyskretnych problemów lokalizacyjnych poprzez wprowadzenie odpowiednich nadmiarowych ograniczeń.

**Cel 3** Poprawa wydajności i dokładności przybliżonej metody rozwiązywania dyskretnych problemów lokalizacyjnych w sensie najlepszego rozkładu ocen z dowolnymi wagami preferencji.

Osiągnięcie poszczególnych celów będzie wymagało udowodnienia odpowiednich tez, które można wyrazić w następujący sposób:

**Teza 1** Możliwe jest sformułowanie parametrycznych modeli optymalizacji wielokryterialnej dla dyskretnych zagadnień lokalizacyjnych, które uwzględniają zarówno preferencje efektywnościowo-sprawiedliwościowe, jak i zróżnicowane zapotrzebowania.

**Teza 2** Istnieją nadmiarowe ograniczenia, które pozwalają na poprawę efektywności modeli optymalizacji wielokryterialnej w sensie najlepszego rozkładu ocen dla dyskretnych problemów lokalizacyjnych.

**Teza 3** Możliwa jest konstrukcja przybliżonej metody rozwiązywania dyskretnych problemów lokalizacyjnych, która dla dowolnych wag preferencji pozwala efektywnie osiągać rozwiązania dobrej jakości w sensie rozkładu ocen dla problemów o dużych rozmiarach (rzędu kilkuset lokalizacji).

Struktura pracy jest następująca. W następnym rozdziale omówiono racjonalne modele preferencji i techniki generacji, które pozwalają uzyskiwać rozwiązania o zadanych własnościach dystrybucyjnych. W rozdziale 3 sformułowano wielokryterialny dyskretny problem lokalizacyjny w postaci zadania programowania matematycznego oraz przedstawiono modele optymalizacyjne do rozwiązywania tego typu problemów stosowane dotychczas w literaturze. Rozdział 4 poświęcony jest operatorowi uporządkowanej średniej ważonej (OWA). W kolejnym rozdziale analizowane są modele optymalizacyjne operatora OWA i badana wydajność obliczeniowa różnych sformułowań tego operatora dla dyskretnych problemów lokalizacyjnych. W rozdziale 6 przedstawiono rozszerzoną wersję operatora OWA, czyli operator wartościowanej uporządkowanej średniej ważonej (WOWA). W rozdziale 7 badane są modele optymalizacyjne operatora WOWA i ich wydajność obliczeniowa dla dyskretnych problemów lokalizacyjnych. W rozdziale 8 zaprezentowano zmodyfikowaną przybliżoną metodę przeszukiwania zmiennego sąsiedztwa (VNS) z agregacją OWA dla dyskretnych problemów lokalizacyjnych. W rozdziale 9 metoda VNS z poprzedniego rozdziału została rozszerzona dla agregacji WOWA. Ostatni rozdział stanowi podsumowanie pracy.



## 2. Modele preferencji

W problemach wielokryterialnych nie istnieje zazwyczaj jednoznaczne najlepsze rozwiązanie. Poszczególne kryteria oceniają możliwe decyzje z różnych perspektyw, które często są nawzajem przeciwstawne. Ostateczny wybór rozwiązania zależy od preferencji decydenta. Stąd w analizie wielokryterialnej stosuje się modele preferencji, które pozwalają na wybór rozwiązania o pożądanym własnościach. Modele preferencji są definiowane przez odpowiednie relacje preferencji i oba te terminy będą tutaj używane zamiennie. Jak zaznaczono w poprzednim rozdziale, niniejsza praca ograniczona jest do racjonalnych relacji preferencji (definicja 1.1). W tym rozdziale przedstawione są modele preferencji i odpowiadające im własności. Szczególną uwagę zwrócono na relacje preferencji, które zapewniają uzyskanie symetrycznie efektywnych i wyrównująco efektywnych rozwiązań [60]. Relacje te są zgodne z racjonalną relacją preferencji i pozwalają znajdować rozwiązania spełniające własność bezstronności, a także rozwiązania sprawiedliwe w sensie wyrównującym.

### 2.1. Pareto-efektywność

Rozwiązanie problemu decyzyjnego dla spójnej racjonalnej relacji preferencji polega na znalezieniu dopuszczalnego wektora zmiennych decyzyjnych, któremu odpowiada najmniejszy wektor ocen — wybierany jest wektor  $\mathbf{x}^0 \in Q$  taki, że  $\mathbf{f}(\mathbf{x}^0) \preceq \mathbf{y}$  dla każdego  $\mathbf{y} \in A$ .

W ogólności, gdy racjonalna relacja preferencji nie jest spójna, rozwiązanie problemu decyzyjnego polega na znalezieniu dopuszczalnego wektora zmiennych decyzyjnych, któremu odpowiada minimalny (w sensie relacji) wektor ocen — wybierany jest wektor  $\mathbf{x}^0 \in Q$  taki, że nie istnieje wektor  $\mathbf{y} \in A$  spełniający  $\mathbf{y} \prec \mathbf{f}(\mathbf{x}^0)$ . Biorąc pod uwagę warunki, które musi spełniać racjonalna relacja preferencji (patrz definicja 1.1), przy rozwiązywaniu problemu decyzyjnego można wykluczyć pewne wektory ocen, które nie mogą być wektorami minimalnymi w sensie danej racjonalnej relacji preferencji. Jest tak, ponieważ pewne wektory ocen są gorsze od innych dla każdej racjonalnej relacji preferencji. Formalnie wyraża się to za pomocą relacji racjonalnej dominacji.

**Definicja 2.1.** Wektor ocen  $\mathbf{y}'$  (racjonalnie) dominuje  $\mathbf{y}''$ , lub  $\mathbf{y}''$  jest (racjonalnie) dominowany przez  $\mathbf{y}'$  ( $\mathbf{y}' \prec_r \mathbf{y}''$ ) wtedy i tylko wtedy, gdy  $\mathbf{y}' \prec \mathbf{y}''$  dla wszystkich racjonalnych relacji preferencji.

Podobnie można zdefiniować relację racjonalnej indyferencji  $\cong_r$  oraz słabej dominacji

$\preceq_r$ . Relacja dominacji  $\preceq_r$  jest najogólniejszą racjonalną relacją preferencji i każda racjonalna relacja preferencji jest z nią zgodna w tym sensie, że

$$\mathbf{y}' \preceq_r \mathbf{y}'' \Rightarrow \mathbf{y}' \preceq \mathbf{y}''.$$

Ponieważ relacja racjonalnej dominacji zazwyczaj nie jest spójna, istnieje w sensie tej relacji wiele minimalnych wektorów ocen, które są między sobą nieporównywalne. W związku z tym nie istnieje zazwyczaj jeden najlepszy wektor ocen, który dominuje wszystkie pozostałe. Można jednak wyróżnić podzbiór wektorów niezdominowanych  $A_N$ .

**Definicja 2.2.** Wektor ocen  $\mathbf{y} \in A$  nazywamy (racjonalnie) niezdominowanym wtedy i tylko wtedy, gdy nie istnieje  $\mathbf{y}' \in A$  spełniający  $\mathbf{y}' \prec_r \mathbf{y}$ .

Pojęcie niezdominowanych wektorów ocen, które odnosi się do przestrzeni ocen, pozwala na zdefiniowanie rozwiązania Pareto-optimalnego.

**Definicja 2.3.** Wektor dopuszczalny  $\mathbf{x} \in Q$  nazywamy rozwiązaniem efektywnym (sprawnym, Pareto-optimalnym) zadania wielokryterialnego (1.1) wtedy i tylko wtedy, gdy odpowiadający mu wektor ocen  $\mathbf{y} = \mathbf{f}(\mathbf{x})$  jest wektorem niezdominowanym.

Relacja racjonalnej dominacji  $\preceq_r$  może być wyrażona za pomocą nierówności wektorowej. Prawdziwe jest następujące twierdzenie.

**Twierdzenie 2.1.** Dla dowolnych wektorów ocen  $\mathbf{y}', \mathbf{y}''$

$$\mathbf{y}' \preceq_r \mathbf{y}'' \Leftrightarrow y'_i \leq y''_i \quad \text{dla } i = 1, \dots, m.$$

Na podstawie twierdzenia 2.1 optymalność rozwiązania można zdefiniować w oparciu o nierówność wektorową.

**Definicja 2.4.** Wektor dopuszczalny  $\mathbf{x}^0 \in Q$  nazywamy rozwiązaniem efektywnym (sprawnym, Pareto-optimalnym) zadania wielokryterialnego (1.1) wtedy i tylko wtedy, gdy nie istnieje wektor  $\mathbf{x} \in Q$  taki, że  $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^0)$  dla każdego  $i$ , gdzie przynajmniej jedna z nierówności jest spełniona w sposób ścisły.

Tak zdefiniowana efektywność odpowiada powszechnie stosowanej definicji optymalności w sensie Pareto, która rozwiązaniem optymalnym nazywa takie rozwiązanie dopuszczalne, w którym nie można poprawić żadnej oceny, nie pogarszając przy tym innych ocen [82]. Optymalność w sensie Pareto jest powszechnie akceptowanym warunkiem, który powinny spełniać rozwiązania wielokryterialnych problemów decyzyjnych.

## 2.2. Symetryczna efektywność

Gdy poszczególne kryteria wyrażają równoważne (analogiczne) semantycznie oceny, ewaluacja rozwiązania powinna uwzględniać rozkład ocen, a nie przypisanie wartości do konkretnych kryteriów. W tym celu relacja powinna spełniać warunek anonimowości (bezstronności). Relacja spełnia warunek anonimowości, gdy

$$(y_{\tau(1)}, y_{\tau(2)}, \dots, y_{\tau(m)}) \cong (y_1, y_2, \dots, y_m) \quad (2.1)$$

dla dowolnej permutacji  $\tau$  zbioru  $I = \{1, 2, \dots, m\}$ . Własność anonimowości oznacza, że dowolna permutacja wektora ocen nie wpływa na jego jakość (jest równie dobra jak oryginalny wektor ocen) w sensie danej relacji preferencji. Przykładowo w problemie z dwoma kryteriami wektor ocen  $(4, 1)$  jest równie dobry jak wektor  $(1, 4)$ .

**Definicja 2.5.** Relację preferencji  $\preceq_a$  nazywamy anonimowo racjonalną relacją preferencji, jeśli spełnia warunki (1.4)–(1.6) i (2.1).

Na podstawie anonimowo racjonalnej relacji preferencji można, analogicznie jak racjonalną dominację, zdefiniować symetryczną dominację.

**Definicja 2.6.** Wektor ocen  $y'$  dominuje symetrycznie  $y''$ , lub  $y''$  jest symetrycznie dominowany przez  $y'$  ( $y' \prec_a y''$ ) wtedy i tylko wtedy, gdy  $y' \prec y''$  dla wszystkich anonimowo racjonalnych relacji preferencji.

**Definicja 2.7.** Wektor ocen  $y \in A$  nazywamy symetrycznie (racjonalnie) niezdominowanym wtedy i tylko wtedy, gdy nie istnieje  $y' \in A$  spełniający  $y' \prec_a y$ .

**Definicja 2.8.** Wektor dopuszczalny  $x \in Q$  nazywamy rozwiązaniem symetrycznie efektywnym zadania wielokryterialnego (1.1) wtedy i tylko wtedy, gdy odpowiadający mu wektor ocen  $y = f(x)$  jest wektorem symetrycznie niezdominowanym.

Relacja symetrycznej dominacji może być wyrażona za pomocą nierówności wektorowej na uporządkowanych wektorach ocen. Formalnie w tym celu wprowadza się odwzorowanie  $\Theta: R^m \rightarrow R^m$  takie, że składowe wektora  $\Theta(y) = (\theta_1(y), \theta_2(y), \dots, \theta_m(y))$  są uporządkowane nierosnąco  $\theta_1(y) \geq \theta_2(y) \geq \dots \geq \theta_m(y)$  oraz istnieje permutacja  $\tau$  zbioru  $I$ , taka że  $\theta_i(y) = y_{\tau(i)}$  dla  $i = 1, 2, \dots, m$ .

**Twierdzenie 2.2.** Dla dowolnych wektorów ocen  $y', y''$

$$y' \preceq_a y'' \Leftrightarrow \theta_i(y') \leq \theta_i(y'') \quad \text{dla } i = 1, \dots, m.$$

Relacja symetrycznej dominacji jest spójna z relacją racjonalnej dominacji, ale ściślejsza w tym sensie, że każde rozwiązanie symetrycznie efektywne jest jednocześnie rozwiązaniem efektywnym w sensie Pareto, ale nie odwrotnie. Przykładowo oba wektory ocen  $(4, 1)$  i  $(1, 3)$  odpowiadają rozwiązaniom efektywnym w sensie Pareto. Biorąc jednak pod uwagę rozkład

ocen, wektor  $(1, 3)$  symetrycznie dominuje  $(4, 1)$ , gdyż rozkład ocen z wartościami 3 i 1 jest lepszy (minimalizacja) od rozkładu z wartościami 4 i 1.

Dzięki możliwości wyrażenia relacji symetrycznej dominacji w postaci nierówności uporządkowanych wektorów ocen problem poszukiwania rozwiązań symetrycznie efektywnych problemu lokalizacyjnego (1.1) można zastąpić problemem poszukiwania rozwiązań efektywnych w sensie Pareto dla następującego modelu wielokryterialnego

$$\min \{(\theta_1(\mathbf{f}(\mathbf{x})), \theta_2(\mathbf{f}(\mathbf{x})), \dots, \theta_m(\mathbf{f}(\mathbf{x}))) : \mathbf{x} \in Q\}. \quad (2.2)$$

**Twierdzenie 2.3.** Rozmieszczenie  $\mathbf{x} \in Q$  jest symetrycznie efektywnym rozwiązaniem wielokryterialnego problemu lokalizacyjnego (1.1) wtedy i tylko wtedy, gdy jest rozwiązaniem efektywnym wielokryterialnego problemu (2.2).

### 2.3. Wyrównująca efektywność

Zgodnie z teorią równości [79] relacja preferencji powinna dążyć do zmniejszenia rozbieżności ocen. Własność ta opiera się na regule przesunięć wyrównujących (Pigue-Daltona), która wyraża się następująco

$$y_{i'} > y_{i''} \Rightarrow \mathbf{y} - \varepsilon \mathbf{e}_{i'} + \varepsilon \mathbf{e}_{i''} \prec \mathbf{y} \quad \text{dla } 0 < \varepsilon < y_{i'} - y_{i''}. \quad (2.3)$$

Zasada przesunięć wyrównujących oznacza, że poprawiając w odpowiednio małym stopniu ocenę gorszą kosztem oceny lepszej, otrzymamy lepszy wektor ocen. Przez odpowiednio mały stopień rozumie się takie przesunięcie, w wyniku którego początkowo gorsza ocena nie stanie się lepsza od początkowego poziomu oceny lepszej. Przykładowo w dwukryterialnym problemie wektor  $(6, 2)$  można poprawić przez przesunięcie 1 od oceny pierwszej do drugiej, co daje w rezultacie wektor  $(5, 3)$ . Jednak przesunięcie 5 od oceny pierwszej do drugiej doprowadzi do gorszego wektora  $(1, 7)$  o większej rozbieżności ocen.

Zasada przesunięć wyrównujących nie jest sprzeczna z warunkami anonimowych i racjonalnych relacji preferencji. Pozwala to zdefiniować wyrównująco racjonalną relację preferencji podobnie jak racjonalną i anonimową relację preferencji.

**Definicja 2.9.** Relację preferencji  $\preceq_w$  nazywamy wyrównująco (sprawiedliwą) racjonalną relacją preferencji, jeśli spełnia warunki (1.4)–(1.6), (2.1) i (2.3).

Na podstawie wyrównująco racjonalnej relacji preferencji można, podobnie jak racjonalną i symetryczną dominację, zdefiniować wyrównująco dominację.

**Definicja 2.10.** Wektor ocen  $\mathbf{y}'$  wyrównująco dominuje  $\mathbf{y}''$ , lub  $\mathbf{y}''$  jest wyrównująco dominowany przez  $\mathbf{y}'$  ( $\mathbf{y}' \prec_w \mathbf{y}''$ ) wtedy i tylko wtedy, gdy  $\mathbf{y}' \prec \mathbf{y}''$  dla wszystkich wyrównująco racjonalnych relacji preferencji.

**Definicja 2.11.** Wektor ocen  $y \in A$  nazywamy wyrównująco (racjonalnie) niezdominowanym wtedy i tylko wtedy, gdy nie istnieje  $y' \in A$  spełniający  $y' \prec_w y$ .

**Definicja 2.12.** Wektor dopuszczalny  $x \in Q$  nazywamy rozwiązaniem wyrównująco efektywnym zadania wielokryterialnego (1.1) wtedy i tylko wtedy, gdy odpowiadający mu wektor ocen  $y = f(x)$  jest wektorem wyrównująco niezdominowanym.

Relacja wyrównującej dominacji może być wyrażona za pomocą nierówności wektorowej na skumulowanych uporządkowanych wektorach ocen. Do tego celu można wykorzystać odwzorowanie  $\Theta$  wprowadzone przy relacji symetrycznej dominacji, którego wynikiem jest wektor  $\Theta(y) = (\theta_1(y), \theta_2(y), \dots, \theta_m(y))$  ze współrzędnymi o nierosnących wartościach. Następnie dla tak uporządkowanego wektora stosuje się operator agregacji zdefiniowany jako

$$\bar{\theta}_i(y) = \sum_{j=1}^i \theta_j(y) \quad \text{dla } i = 1, \dots, m.$$

W ten sposób otrzymuje się wektor  $\bar{\Theta}(y) = (\bar{\theta}_1(y), \bar{\theta}_2(y), \dots, \bar{\theta}_m(y))$  skumulowanych uporządkowanych ocen, którego kolejne współrzędne wyrażają odpowiednio: największą ocenę, sumę dwóch największych ocen, sumę trzech największych ocen itd.

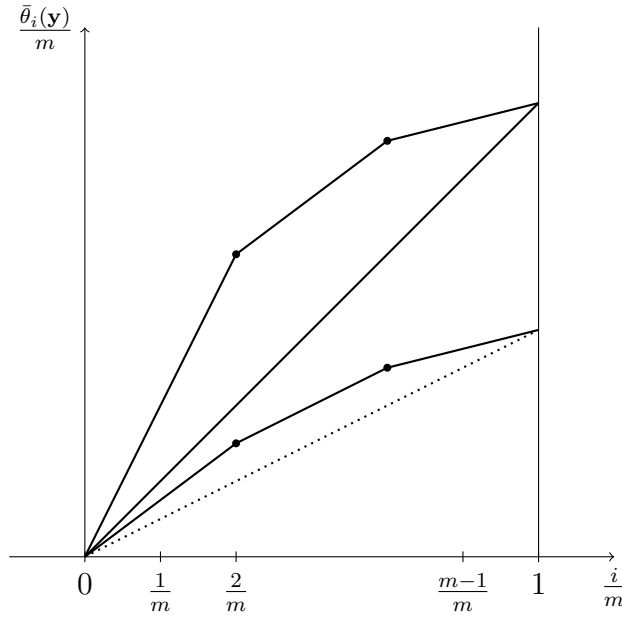
Można zauważyć, że dla dwóch wektorów  $y', y'' \in Y$  zachodzi  $\bar{\Theta}(y') = \bar{\Theta}(y'')$  wtedy i tylko wtedy, gdy mają one identyczny rozkład wartości (czyli  $\Theta(y') = \Theta(y'')$ ). Podobnie nierówność  $\Theta(y') \leq \Theta(y'')$  oznacza spełnienie  $\bar{\Theta}(y') \leq \bar{\Theta}(y'')$ , jednak zależność odwrotna nie jest prawdziwa. Relacje między wektorami  $\bar{\Theta}(y')$  i  $\bar{\Theta}(y'')$  były szczegółowo badane w pracach [50] i [95]. Na podstawie uzyskanych tam wyników autorzy pracy [39] sformułowali następujące twierdzenie.

**Twierdzenie 2.4.** Dla dowolnych wektorów ocen  $y', y''$

$$y' \preceq_w y'' \Leftrightarrow \bar{\theta}_i(y') \leq \bar{\theta}_i(y'') \quad \text{dla } i = 1, \dots, m.$$

Skumulowany uporządkowany wektor ocen można przedstawić graficznie w formie odcinkami liniowej funkcji, która łączy punkty  $(0,0)$  i  $(i/m, \bar{\theta}_i(y)/m)$  dla  $i = 1, 2, \dots, m$ . W szczególności ostatni punkt to  $(1, \bar{\theta}_m(y)/m)$ , który odpowiada średniej na osi rzędnych. Wykres stanowi górną bezwzględna krzywą Lorenza (ang. upper absolute Lorenz curve). Krzywa jest w ogólności wklęsła, a dla wektorów równych ocen przyjmuje postać prostych ukośnych (rysunek 2.1).

W ekonomii krzywych Lorenza używa się do badania zróżnicowania przychodów pewnej populacji. Wykres obrazuje wówczas skumulowane przychody w zależności od części populacji, której te przychody odpowiadają. Zazwyczaj przedstawia się dolną krzywą Lorenza (krzywa wypukła), w której członków populacji porządkuje się według niemalejących przychodów. W pracy, ze względu na odwrotny sposób sortowania, prezentowany wykres stanowi górną krzywą Lorenza. Nie jest to jednak istotna różnica. Ważna jest natomiast różnica między



Rysunek 2.1.  $\bar{\Theta}(\mathbf{y})$  wyrażone w postaci górnych bezwzględnych krzywych Lorenza [38]

krzywą bezwzględną i względną. W przypadku względnej krzywej, gdzie przychody są znormalizowane (na osi rzędnych wartości reprezentują procent przychodów względem przychodu sumarycznego), uwzględniane są tylko rozbieżności, a ignorowane bezwzględne wartości. Bezwzględna krzywa Lorenza (odpowiadająca skumulowanemu uporządkowanemu wektorowi ocen) uwzględnia wartości ocen, dzięki czemu wektor nierównych niewielkich ocen może być lepszy od wektora dużych równych ocen. Pozwala to wyeliminować problem występujący przy bezpośredniej minimalizacji rozbieżności, w którym oceny mogą być pogarszane, aby były do siebie bardziej zbliżone.

Relacja wyrównującej dominacji jest spójna zarówno z relacją racjonalnej dominacji, jak i relacją symetrycznej dominacji, ale jest od nich ściślej. Każde rozwiązanie wyrównująco efektywne jest jednocześnie rozwiązaniem efektywnym w sensie Pareto oraz symetrycznie efektywnym, ale nie odwrotnie. Przykładowo oba wektory ocen  $(4, 1)$  i  $(3, 2)$  odpowiadają rozwiązaniom efektywnym w sensie Pareto jak i symetrycznie efektywnym rozwiązaniom. Uwzględniając jednak przesunięcia wyrównujące, wektor  $(3, 2)$  wyrównująco dominuje  $(4, 1)$ , gdyż zwiększając w wektorze  $(4, 1)$  drugą współrzędną kosztem pierwszej o 1, otrzymamy wektor  $(3, 2)$ . Dominacja jest również widoczna na podstawie twierdzenia 2.4, gdyż  $\bar{\Theta}((3, 2)) = (3, 5) < (4, 5) = \bar{\Theta}((4, 1))$ .

Dzięki możliwości wyrażenia relacji wyrównującej dominacji w postaci nierówności skumulowanych uporządkowanych wektorów ocen problem poszukiwania rozwiązań wyrównująco efektywnych problemu lokalizacyjnego (1.1) można zastąpić problemem poszukiwania rozwiązań efektywnych w sensie Pareto dla następującego modelu wielokryterialnego

$$\min \left\{ \left( \bar{\theta}_1(\mathbf{f}(\mathbf{x})), \bar{\theta}_2(\mathbf{f}(\mathbf{x})), \dots, \bar{\theta}_m(\mathbf{f}(\mathbf{x})) \right) : \mathbf{x} \in Q \right\}. \quad (2.4)$$

**Twierdzenie 2.5.** Rozmieszczenie  $\mathbf{x} \in Q$  jest wyrównująco efektywnym rozwiązaniem wielokryterialnego problemu lokalizacyjnego (1.1) wtedy i tylko wtedy, gdy jest rozwiązaniem efektywnym wielokryterialnego problemu (2.4).

## 2.4. Techniki generacji rozwiązań

Jedną z możliwości znalezienia efektywnego rozwiązania wielokryterialnego zadania jest rozwiązanie jego skalaryzacji. Polega to na rozwiązaniu jednokryterialnego zadania optymalizacji, gdzie kryterium jest wyrażone za pomocą funkcji skalaryzującej  $s: R^m \rightarrow R$

$$\min\{s(f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})) : \mathbf{x} \in Q\}. \quad (2.5)$$

Zadanie 2.5 definiuje więc relację preferencji za pomocą funkcji skalaryzującej

$$\mathbf{y}' \preceq_s \mathbf{y}'' \Leftrightarrow s(\mathbf{y}') \leq s(\mathbf{y}'').$$

Relacja  $\preceq_s$  jest zawsze zwrotna, przechodnia i spójna. Jeśli spełnia dodatkowo warunek ścisłej monotoniczności, to optymalne rozwiązanie zadania (2.5) jest efektywnym rozwiązaniem wielokryterialnego zadania (1.1). Dla ścisłej monotoniczności relacji wystarcza, aby funkcja  $s$  była ściśle monotoniczna. Gdy relacja definiowana przez funkcję  $s$  spełnia tylko warunek słabej monotoniczności

$$\mathbf{y} - \varepsilon \mathbf{e}_i \preceq \mathbf{y} \quad \text{dla } \varepsilon > 0, \quad i = 1, \dots, m, \quad (2.6)$$

wówczas zbiór optymalnych rozwiązań zadania (2.5) zawiera efektywne rozwiązanie wielokryterialnego zadania (1.1). Jeśli optymalne rozwiązanie zadania (2.5) jest jednoznaczne w przestrzeni ocen  $Y$ , to jest ono efektywnym rozwiązaniem wielokryterialnego zadania (1.1).

Jedną z najpopularniejszych funkcji skalaryzujących jest suma poszczególnych ocen, która prowadzi do zadania postaci

$$\min\left\{\sum_{i=1}^m f_i(\mathbf{x}) : \mathbf{x} \in Q\right\}. \quad (2.7)$$

Suma ocen jest funkcją ściśle rosnącą i optymalne rozwiązanie zadania (2.7) jest efektywnym rozwiązaniem wielokryterialnego zadania (1.1).

Do wyznaczania różnych rozwiązań efektywnych powszechnie stosuje się metodę ważenia ocen, która jest szczególnym przypadkiem skalaryzacji za pomocą liniowych ściśle rosnących funkcji, co daje w rezultacie optymalizację postaci

$$\min\left\{\sum_{i=1}^m w_i f_i(\mathbf{x}) : \mathbf{x} \in Q\right\}. \quad (2.8)$$

Dla dowolnych dodatnich wag  $w_i$  optymalne rozwiązanie zadania (2.8) stanowi efektywne

rozwiązanie wielokryterialnego zadania (1.1). Co więcej, ważona suma ocen z odpowiednio dobraćanymi wagami pozwala na wyznaczenie każdego efektywnego rozwiązania wielokryterialnego zadania programowania liniowego. Dodatkową zaletą jest fakt, że jako liniowa funkcja skalaryzująca metoda ważenia nie wprowadza dodatkowych utrudnień obliczeniowych. Niestety możliwość wyznaczenia dowolnego rozwiązania efektywnego dotyczy tylko przypadku liniowego i nie zachodzi dla ogólnych zadań optymalizacji wielokryterialnej, w szczególności dla zadań dyskretnych.

Minimalizacja sumy wartości ocen jest równoznaczna minimalizacji średniej wartości, gdyż optymalizacja nie jest zależna od przeskalowania funkcji celu przez stałą. Analogicznie do poszukiwania najlepszej wartości średniej można minimalizować wartość najgorszą, co prowadzi do zadania optymalizacji minimaksowej

$$\min\left\{\max_{i=1,\dots,m} f_i(\mathbf{x}) : \mathbf{x} \in Q\right\}. \quad (2.9)$$

Funkcja  $\max_{i=1,\dots,m} f_i(\mathbf{x})$  spełnia tylko warunek słabej monotoniczności (2.6), przez co optymalne rozwiązanie zadania (2.9) jest efektywnym rozwiązaniem wielokryterialnego zadania (1.1) tylko wtedy, gdy jest to rozwiązanie jednoznaczne w przestrzeni ocen. W przeciwnym wypadku zbiór optymalnych rozwiązań zadania (2.9) zawiera efektywne rozwiązanie zadania (1.1).

Jeśli relacja definiowana przez funkcję skalaryzującą spełnia także warunek anonimowości, to optymalne rozwiązanie zadania (2.5) jest również symetrycznie efektywnym rozwiązaniem wielokryterialnego zadania (1.1). Relacja jest anonimowa wtedy i tylko wtedy, gdy funkcja  $s$  jest symetryczna, czyli

$$s(y_{\tau(1)}, y_{\tau(2)}, \dots, y_{\tau(m)}) = s(y_1, y_2, \dots, y_m)$$

dla dowolnej permutacji  $\tau$  zbioru  $I = \{1, 2, \dots, m\}$ . Wspomniana wcześniej suma wszystkich ocen  $\sum_{i=1}^m f_i(x)$  jest funkcją symetryczną. Własność symetrii ma też funkcja  $s(\mathbf{y}) = \max_{i=1,\dots,m} y_i$ . Oczywiście w ogólności, podobnie jak w przypadku rozwiązań efektywnych, zbiór optymalnych rozwiązań zadania minimalizacji najgorszej oceny zawiera symetrycznie efektywne rozwiązania zadania (1.1).

W przedstawionym wcześniej zadaniu (2.8) do wyznaczania różnych rozwiązań efektywnych wykorzystywana była skalaryzacja w postaci sumy ocen z różnymi współczynnikami wagowymi dla poszczególnych funkcji oceny. Odpowiada to różnym przekształceniom skal poszczególnych ocen. Takie podejście nie jest możliwe przy poszukiwaniu symetrycznie efektywnych rozwiązań, gdyż łamie ono warunek anonimowości. Przy poszukiwaniu symetrycznie efektywnych rozwiązań wszystkie oceny mogą być jedynie przekształcone za pomocą tej samej ściśle rosnącej funkcji.

**Twierdzenie 2.6.** Dla dowolnej ściśle rosnącej funkcji  $s: R \rightarrow R$ , optymalne rozwiązanie



zadania

$$\min\left\{\sum_{i=1}^m s(f_i(x)) : \mathbf{x} \in Q\right\} \quad (2.10)$$

jest symetrycznie efektywnym rozwiązaniem wielokryterialnego zadania (1.1).

Liniowa funkcja  $s$  w skalaryzacji (2.10) nie zmienia rozwiązania optymalnego. Aby generować różne rozwiązania symetrycznie efektywne, konieczne jest zastosowanie nieliniowych funkcji  $s$ . W ogólnym problemie optymalizacyjnym, gdy dopuszczalne są dowolne oceny, w tym ujemne, może to zawęzić zbiór możliwych do wykorzystania funkcji ze względu na niezachowanie ścisłej monotoniczności na zbiorze dopuszczalnych wartości ocen [40].

Aby optymalne rozwiązanie skalaryzacji (2.5) było również wyrównująco efektywnym rozwiązaniem wielokryterialnego zadania (1.1), relacja definiowana przez funkcję skalaryzującą, poza warunkami zwrotności, przechodniości, ścisłej monotoniczności i anonimowości, powinna spełniać także zasadę przesunięć wyrównujących. Odpowiada to spełnieniu przez funkcję skalaryzującą warunku

$$y_{i'} > y_{i''} \Rightarrow s(\mathbf{y} - \varepsilon \mathbf{e}_{i'} + \varepsilon \mathbf{e}_{i''}) < s(\mathbf{y}) \quad \text{dla } 0 < \varepsilon < y_{i'} - y_{i''}.$$

Dla spełnienia tego warunku wystarcza, aby funkcja skalaryzująca  $s: R^m \rightarrow R$  była ściśle wypukłą.

Funkcja skalaryzująca w postaci sumy ocen nie jest zgodna z regułą przesunięć wyrównujących, przez co nie generuje rozwiązań wyrównująco efektywnych. Możliwe jest jednak wstępne przeskalowanie poszczególnych ocen za pomocą ściśle wypukłej i rosnącej funkcji, a następnie ich zsumowanie, co prowadzi do zadania optymalizacji ze ściśle wypukłą i rosnącą funkcją skalaryzującą. Rozwiązanie optymalne takiego zadania stanowi wyrównująco efektywne rozwiązanie oryginalnego problemu wielokryterialnego. Podobnie jak przy generacji symetrycznie efektywnych rozwiązań, zastosowanie nieliniowych funkcji skalaryzujących wprowadza na nie pewne ograniczenia (kwestie ścisłej monotoniczności w przypadku ujemnych ocen).

#### 2.4.1. Agregacja uporządkowanych ocen

Zgodnie z twierdzeniem 2.3 poszukiwanie symetrycznie efektywnego rozwiązania problemu z oryginalnym wektorem ocen można zastąpić poszukiwaniem efektywnego rozwiązania problemu z uporządkowanym wektorem ocen. Stosując do zadania (2.2) różne techniki generowania rozwiązań efektywnych, otrzymuje się symetrycznie efektywne rozwiązania oryginalnego zadania (1.1).

Jak wspomniano wcześniej, stosowanie różnych wag do poszczególnych ocen łamie warunek anonimowości i przez to metoda ważenia nie może być bezpośrednio stosowana do otrzymywania symetrycznie efektywnych rozwiązań. Można jednak zastosować metodę ważenia do uporządkowanego wektora ocen. Pozwala to na uzyskiwanie różnych rozwiązań

efektywnych zadania (2.2), które z kolei stanowią symetrycznie efektywne rozwiązania wielokryterialnego zadania (1.1). Podejście takie zaproponował Yager [91], określając je mianem uporządkowanej średniej ważonej (ang. Ordered Weighted Averaging, OWA). Agregacja OWA prowadzi do zadania

$$\min\left\{\sum_{i=1}^m w_i \theta_i(\mathbf{f}(\mathbf{x})) : \mathbf{x} \in Q\right\},$$

którego każde rozwiązanie optymalne, dla dowolnych dodatnich wag  $w_i$ , jest symetrycznie efektywnym rozwiązaniem oryginalnego zadania wielokryterialnego (1.1).

Analogiczna sytuacja występuje dla wyrównująco efektywnych rozwiązań. Zgodnie z twierdzeniem 2.5 wyrównująco efektywne rozwiązanie oryginalnego zadania wielokryterialnego można uzyskać jako efektywne rozwiązanie zadania ze skumulowanym uporządkowanym wektorem ocen. Aby otrzymać wyrównująco efektywne rozwiązanie jako optymalne rozwiązanie skalaryzacji, można więc zastosować metodę ważenia ocen do zadania (2.4), otrzymując zadanie postaci

$$\min\left\{\sum_{i=1}^m w_i \bar{\theta}_i(\mathbf{f}(\mathbf{x})) : \mathbf{x} \in Q\right\}.$$

Okazuje się, że zadanie to można wyrazić w postaci agregacji OWA z odpowiednio zmodyfikowanymi wagami, co jest szczegółowo przedstawione w rozdziale 4.

Zadanie optymalizacji uporządkowanej średniej ważonej pozwala więc uzyskiwać rozwiązania najlepsze w sensie rozkładu ocen, zarówno symetrycznie efektywne, jak i wyrównująco efektywne w zależności od zastosowanych wag preferencji.

### 3. Zagadnienia lokalizacyjne — stan sztuki

W rozdziale formalnie przedstawiono wielokryterialny dyskretny problem lokalizacyjny rozważany w niniejszej pracy. Problem ten został wyrażony w postaci zadania programowania matematycznego, a dokładnie zadania programowania całkowitoliczbowego liniowego mieszanego (PCLM). Następnie omówiono najważniejsze modele optymalizacyjne stosowane dotychczas do rozwiązywania tego typu problemów, zwracając szczególną uwagę na możliwość uwzględnienia preferencji efektywnościowo-sprawiedliwościowych i zróżnicowanych wag zapotrzebowania.

#### 3.1. Problem lokalizacyjny

W pracy rozważany jest dyskretny problem lokalizacyjny, który w literaturze funkcjonuje także jako sieciowy problem lokalizacyjny, gdzie punkty obsługi mogą być umieszczone tylko w (wybranych lub wszystkich) wierzchołkach sieci [41]. Problem przedstawia się następująco. Dany jest zbiór  $m$  jednostek przestrzennych (np. klientów) oraz zbiór potencjalnych lokalizacji, w których mają zostać umieszczone obiekty do obsługi klientów. Zgodnie z założeniami pracy zbiory te są ze sobą tożsame. Problem sprowadza się do wyboru  $n$  lokalizacji spośród  $m$  możliwych, w których umieszczone zostaną punkty obsługi do zaspokojenia zapotrzebowania klientów, przy czym  $n \leq m$ . Decyzje wyboru lokalizacji mogą być wyrażone jako zmienne binarne  $x_j$  ( $j = 1, \dots, m$ ), które przyjmują wartość 1, gdy obiekt zostanie umieszczone w lokalizacji  $j$ -tej, a 0 w przeciwnym razie. Uwzględniając liczbę punktów obsługi, zmienne decyzyjne powinny spełniać następujące ograniczenie

$$\sum_{j=1}^m x_j = n.$$

W reprezentacji problemu wykorzystuje się również zmienne  $x'_{ij}$  ( $i, j = 1, \dots, m$ ) określające przypisanie klientów do poszczególnych punktów obsługi. Zmienna  $x'_{ij}$  jest równa 1, gdy klient  $i$ -ty jest przypisany do obiektu  $j$ -tego, a 0 w przeciwnym razie. Zmienne przydziału powinny spełniać poniższe ograniczenia:

$$\sum_{j=1}^m x'_{ij} = 1 \quad \text{dla } i = 1, \dots, m,$$

$$x'_{ij} \leq x_j \quad \text{dla } i, j = 1, \dots, m.$$

Pierwsze ograniczenie zapewnia przypisanie klienta tylko do jednego punktu obsługi, a drugie gwarantuje, że klient zostanie przypisany tylko do istniejących punktów obsługi.

Dla każdego klienta  $i$  ( $i = 1, \dots, m$ ) zdefiniowana jest funkcja  $f_i(\mathbf{x})$  zależna od rozlokowania obiektów. Wyraża ona ocenę  $i$ -tego klienta danego rozmieszczenia i związanych z nim decyzji przydziału. Funkcje oceny wyrażają zazwyczaj abstrakcyjnie zdefiniowane odległości i jako takie podlegają minimalizacji (równoważna interpretacja może odpowiadać czasowi, kosztowi obsługi itp.). Przy braku ograniczeń pojemnościowych i założeniu o tożsamości usług świadczonych przez poszczególne obiekty każdy z klientów będzie przypisany do najbliższego mu punktu obsługi. Prowadzi to do następującej postaci funkcji

$$f_i(\mathbf{x}) = \min_{j=1, \dots, m} \{c_{ij} : x_j = 1\} \quad \text{dla } i = 1, \dots, m,$$

gdzie współczynniki  $c_{ij}$  wyrażają koszt obsługi  $i$ -tego klienta przez  $j$ -ty punkt obsługi. Wykorzystując jawnie zmienne przydziału, funkcje oceny można wyrazić w postaci liniowej

$$f_i(\mathbf{x}) = \sum_{j=1, \dots, m} c_{ij} x'_{ij} \quad \text{dla } i = 1, \dots, m.$$

Podsumowując, wielokryterialny dyskretny problem lokalizacyjny bez ograniczeń pojemnościowych przedstawia się następująco:

$$\min (f_1, f_2, \dots, f_m) \quad (3.1a)$$

$$\text{p.o. } f_i = \sum_{j=1}^m c_{ij} x'_{ij} \quad \text{dla } i = 1, \dots, m, \quad (3.1b)$$

$$\sum_{j=1}^m x_j = n, \quad (3.1c)$$

$$\sum_{j=1}^m x'_{ij} = 1 \quad \text{dla } i = 1, \dots, m, \quad (3.1d)$$

$$x'_{ij} \leq x_j \quad \text{dla } i, j = 1, \dots, m, \quad (3.1e)$$

$$x_j \in \{0, 1\} \quad \text{dla } j = 1, \dots, m, \quad (3.1f)$$

$$x'_{ij} \in \{0, 1\} \quad \text{dla } i, j = 1, \dots, m. \quad (3.1g)$$

Ograniczenia (3.1c)–(3.1g) definiują wspólnie zbiór rozwiązań dopuszczalnych  $Q$  w przestrzeni zmiennych decyzyjnych. Zgodnie z ograniczeniem (3.1b) zbiór rozwiązań dopuszczalnych jest przekształcany w zbiór osiągalnych wektorów ocen. Warto również zauważyć, że dla tak zdefiniowanego problemu bez ograniczeń pojemnościowych zmienne  $x'_{ij}$  nie muszą być jawnie określone jako binarne. Przypisanie klientów do najbliższych punktów obsługi i ograniczenie (3.1e) pozwala zmodyfikować ograniczenie (3.1g) do postaci  $x'_{ij} \geq 0$  dla  $i, j = 1, \dots, m$ .

Problem (3.1) może być dodatkowo modyfikowany poprzez przypisanie klientom wag  $p_i$ , które mogą wyrażać wielkości zapotrzebowań na usługę. W ten sposób zdefiniowany jest

rozkład ocen  $y_i = f_i(\mathbf{x})$  zgodnie z wagami  $p_i$  dla  $i = 1, \dots, m$ . Wagi całkowite mogą być po prostu interpretowane jako powielenie klienta w tej samej lokalizacji geograficznej, co nie zmienia rozkładu ocen. Dzięki temu problem może być zdezagregowany do postaci pierwotnej, gdzie wagi wszystkich klientów są równe  $p_i = 1$ . Podobnie można postąpić z dowolnymi wymiernymi wagami. Takie przekształcenie jest możliwe, jednak w praktyce powoduje zazwyczaj znaczący wzrost rozmiaru problemu (liczby klientów  $m$ ), co często uniemożliwia jego rozwiązanie. W związku z tym pożądanym jest, aby model bezpośrednio uwzględniał wagi wartościujące.

W tym miejscu warto zaznaczyć, że możliwe jest także użycie wag do przeskalowania odległości, czyli zdefiniowania ocen jako  $y_i = p_i f_i(\mathbf{x})$  dla  $i = 1, \dots, m$  z jednostajnym rozkładem (z pojedynczymi klientami w poszczególnych lokalizacjach). Takie podejście jest zasadniczo różne od optymalizacji rozkładu ocen zgodnego z wagami  $p_i$ . W praktyce przeskalowanie odległości może być uwzględnione w funkcjach oceny, co prowadzi do równoważnego problemu bez wag wartościujących z odpowiednio zmodyfikowanymi odległościami (przemnożonymi przez wagi skalujące). W związku z tym zagadnienie to nie będzie specjalnie omawiane w pracy, gdyż może być rozwiązane przez podstawowe sformułowanie problemu lokalizacyjnego.

## 3.2. Modele optymalizacyjne

W ramach badań operacyjnych rozważano różne zagadnienia lokalizacyjne [13, 27, 42], w tym również przypadki dyskretne [51]. Opracowane modele mają jednak pewne funkcjonalne ograniczenia związane z generowaniem rozwiązań kompromisowych, czy też zapewnieniem rozwiązań o odpowiednich własnościach dystrybucyjnych, np. rozwiązań wyrównująco efektywnych [38].

### 3.2.1. Klasyczne funkcje agregujące

Większość klasycznych zadań lokalizacyjnych koncentruje się na minimalizacji sumarycznej (średniej) odległości — tzw. kryterium średniej — lub minimalizacji maksymalnej odległości — tzw. kryterium centrum — pomiędzy klientami a punktami obsługi [54]. Oba te kryteria mogą być bezpośrednio zastosowane dla problemów z wagami wyrażającymi wielkości zapotrzebowań. Zakładając znormalizowane wagi zapotrzebowania  $p_i$ , kryterium średniej wyraża się jako

$$\mu(\mathbf{y}) = \sum_{i=1}^m p_i y_i.$$

W problemie bez wag przyjmuje się wszystkie wagi  $p_i$  równe  $1/m$ , co prowadzi w rezultacie do zwykłej średniej arytmetycznej. Kryterium średniej, będąc równoważnym sumie odległości, dąży do globalnej efektywności, co może prowadzić do dyskryminacji odległych i mało zaludnionych obszarów na korzyść terenów centralnych i gęsto zaludnionych. Z tego powodu przy rozmieszczaniu usług publicznych używane jest zazwyczaj kryterium centrum, które

jest niezależne od współczynników wagowych i minimalizuje odległość (czas obsługi) dla najgorszego przypadku, czyli klienta położonego najdalej od (najbliższego jemu) punktu obsługi

$$M(y) = \max_{i=1,\dots,m} y_i.$$

Ponieważ kryterium centrum dąży do równości (sprawiedliwości), jest ono szczególnie ważne przy lokalizacji systemów usług awaryjnych (np. pogotowie, straż pożarna).

Kryterium centrum jest zgodne z teorią sprawiedliwości Rawlsa [76], szczególnie przy podejściu leksykograficznym [59]. Z drugiej strony kryterium centrum może doprowadzić do znacznego wzrostu sumarycznej odległości, a przez to negatywnie wpłynąć na globalną efektywność rozwiązania. Inną wadą kryterium centrum jest ograniczona możliwość różnicowania rozwiązań, przez co wiele rozwiązań dopuszczalnych, nawet znacznie różniących się od siebie, może być rozwiązaniami optymalnymi w sensie minimaxowym. Standardowe algorytmy optymalizacyjne w przypadku wielu rozwiązań optymalnych wybierają rozwiązanie w sposób losowy. To sprawia, że rozwiązania optymalne kryterium centrum są bardzo niestabilne [78]. Kolejnym problemem jest fakt, że przy pewnych konfiguracjach klientów i potencjalnych lokalizacji punktów obsługi kryterium centrum może być tylko częściowo aktywne. Sytuacja taka występuje, gdy pewien klient jest znacząco oddalony od wszystkich lokalizacji punktów obsługi. Wówczas minimalizacji podlega tylko odległość tego jednego klienta [46], a wszystkie pozostałe są pomijane.

### 3.2.2. Kryterium centro-średniej

W związku z ograniczonymi możliwościami klasycznych funkcji agregujących poszukiwano sposobu na pogodzenie obu kryteriów, co pozwoliłoby na osiągnięcie rozwiązań kompromisowych. Halpern [30] zaproponował podejście w oparciu o minimalizację liniowej kombinacji obu kryteriów — średniej (sumy) odległości i największej odległości

$$H_\lambda(\mathbf{y}) = \lambda M(\mathbf{y}) + (1 - \lambda)\mu(\mathbf{y}),$$

gdzie  $0 \leq \lambda \leq 1$ . Gdy  $\lambda = 0$  otrzymuje się kryterium średniej, a gdy  $\lambda = 1$  kryterium centrum. Kryterium centro-średniej można postrzegać jako skalaryzację (średnią ważoną) dwukryterialnego problemu z ocenami wyrażającymi średnią i maksymalną odległość. Okazuje się jednak, że w przypadku problemów dyskretnych (niewypukłych) zaproponowana metoda ma bardzo ograniczone możliwości znajdowania rozwiązań kompromisowych [58].

### 3.2.3. Porcja najgorszych ocen

Inne kompromisowe podejście zostało zaproponowane przez Slatera [80]. Zastosował on tzw. kryterium  $k$ -centrum, gdzie minimalizuje się sumę  $k$  największych odległości. Podejście

to stanowi naturalne uogólnienie kryterium centrum. Wykorzystując operator porządkowy  $\Theta$ , koncepcję tego kryterium dla dowolnego  $k = 1, \dots, m$  można przedstawić następująco

$$M_{\frac{k}{m}}(\mathbf{y}) = \frac{1}{k} \sum_{i=1}^k \theta_i(\mathbf{y}).$$

W przypadku  $k = 1$  odpowiada to kryterium centrum, gdyż  $M_{\frac{1}{m}}(\mathbf{y}) = \theta_1(\mathbf{y}) = M(\mathbf{y})$ . Dla  $k$  równego liczbie klientów otrzymuje się zadanie z kryterium średniej, ponieważ  $M_{\frac{m}{m}}(\mathbf{y}) = \frac{1}{m} \sum_{i=1}^m \theta_i(\mathbf{y}) = \frac{1}{m} \sum_{i=1}^m y_i = \mu(\mathbf{y})$ . Początkowo metoda była rozwijana dla dyskretnych problemów z pojedynczym punktem obsługi na grafach o strukturze drzewa [80, 1, 2]. Później Peeters [74] analizował problem z pojedynczym punktem obsługi na grafach i opracował pełną klasyfikację problemów o podobnych kryteriach i metody ich rozwiązywania. Zostały zdefiniowane 4 kryteria:  $k$ -tej największej (najmniejszej) średniej, gdzie minimalizacji podlega średnia  $k$  największych (najmniejszych) odległości oraz  $k$ -tego największego (najmniejszego) centrum, gdzie minimalizacji podlega pojedyncza  $k$ -ta największa (najmniejsza) odległość. Zgodnie z tą klasyfikacją wcześniejsze kryterium  $k$ -centrum powinno być raczej określane mianem  $k$ -tej największej średniej. W pracy [83] został zaproponowany algorytm o wielomianowej złożoności obliczeniowej dla kryterium  $k$ -centrum ( $k$ -tej największej średniej) z wieloma punktami obsługi na grafach o strukturze drzewa. Później Ogryczak i Tamir [72] udowodnili, że kryterium  $k$ -tej największej średniej może być wyznaczone za pomocą prostych dodatkowych nierówności liniowych, co pozwoliło na znaczne uproszczenie wielu zadań z tym kryterium. Mianowicie dla dowolnego osiągalnego wektora ocen  $\mathbf{y} \in A$  i dowolnego  $k = 1, \dots, m$  prawdziwa jest równość:

$$M_{\frac{k}{m}}(\mathbf{y}) = \min_{t, d_i, z_i} \left( t + \frac{1}{k} \sum_{i=1}^m d_i \right)$$

$$\begin{aligned} \text{p.o. } d_i &\geq y_i - t && \text{dla } i = 1, \dots, m, \\ d_i &\leq M z_i && \text{dla } i = 1, \dots, m, \\ \sum_{i=1}^m z_i &\leq k - 1, \\ d_i &\geq 0 && \text{dla } i = 1, \dots, m, \\ z_i &\in \{0, 1\} && \text{dla } i = 1, \dots, m. \end{aligned} \tag{3.2}$$

Okazuje się, że zmienne binarne i odpowiadające im ograniczenia mogą zostać wyeliminowane, co prowadzi do zadania postaci:

$$M_{\frac{k}{m}}(\mathbf{y}) = \min_{t, d_i} \left( t + \frac{1}{k} \sum_{i=1}^m d_i \right) \tag{3.3a}$$

$$\text{p.o. } d_i \geq y_i - t \quad \text{dla } i = 1, \dots, m, \tag{3.3b}$$

$$d_i \geq 0 \quad \text{dla } i = 1, \dots, m. \tag{3.3c}$$

Kryterium  $k$ -tej największej średniej było ograniczone do problemów bez wag wartościujących. Co prawda, używano wag do przeskalowania odległości [83], ale nie pozwalały one na uwzględnienie rozkładu klientów (zapotrzebowania). Ogryczak i Zawadzki [73] opracowali parametryczne uogólnienie kryterium  $k$ -tej największej średniej tzw. najgorszą średnią warunkową, które mogło być zastosowane do problemów z wagami wartościującymi, uwzględniając wielkości zapotrzebowań dla najgorszych ocen [25]. Najgorsza średnia warunkowa minimalizuje średnią największych odległości dla zadanej porcji zapotrzebowania. Zakładając znormalizowane wagi zapotrzebowania  $p_i$  dla  $i = 1, \dots, m$ , najgorszą średnią  $\beta$ -warunkową dla dowolnej miary  $0 < \beta \leq 1$  i osiągalnego wektora ocen  $\mathbf{y} \in A$  można wyznaczyć jako:

$$M_\beta(\mathbf{y}) = \min_{t, d_i, z_i} \left( t + \frac{1}{\beta} \sum_{i=1}^m p_i d_i \right)$$

p.o.  $d_i \geq y_i - t$                       dla  $i = 1, \dots, m$ ,

$d_i \leq M z_i$                               dla  $i = 1, \dots, m$ ,

$\sum_{i=1}^m p_i z_i < \beta$ ,

$d_i \geq 0$                                       dla  $i = 1, \dots, m$ ,

$z_i \in \{0, 1\}$                                 dla  $i = 1, \dots, m$ .

Zmienne binarne i związane z nimi ograniczenia mogą zostać wyeliminowane analogicznie jak w przypadku średniej  $k$ -najgorszych ocen, co daje w rezultacie sformułowanie postaci:

$$M_\beta(\mathbf{y}) = \min_{t, d_i} \left( t + \frac{1}{\beta} \sum_{i=1}^m p_i d_i \right)$$

p.o.  $d_i \geq y_i - t$                       dla  $i = 1, \dots, m$ ,                      (3.4)

$d_i \geq 0$                                       dla  $i = 1, \dots, m$ .

Gdy parametr wyrażający wielkość zapotrzebowania dąży do 0, średnia warunkowa dąży do największej wartości, stając się tym samym kryterium centrum. Z drugiej strony, gdy rozważane jest sumaryczne zapotrzebowanie, średnia warunkowa staje się równoważna kryterium średniej. W szczególnym przypadku, gdy wszystkie zapotrzebowania są równe i miara  $\beta = \frac{k}{m}$ , sformułowanie (3.4) pokrywa się ze sformulowaniem dla kryterium  $k$ -tej największej średniej (3.3). Autorzy [73] pokazali, że najgorsza średnia warunkowa generuje znacznie więcej rozwiązań kompromisowych niż wspomniane wcześniej kryterium centro-średniej.

Chociaż najgorsza średnia warunkowa uwzględnia wagi zapotrzebowania, to jednak jej elastyczność jest ograniczona tylko do zapotrzebowania dla najgorszych ocen. Ta wada może być wyeliminowana przez użycie wartościowanej uporządkowanej średniej ważonej (ang. Weighted OWA, WOWA) [84, 86], która pozwala uwzględnić zapotrzebowanie dla dowolnych ocen.



### 3.2.4. Kryteria równościowe

Równość jest w zasadzie nieostrym pojęciem społeczno-ekonomicznym, które wiąże się z pojęciami uczciwości, sprawiedliwości [79], czy też braku zazdrości [26]. Równość można jednak zmierzyć [3] i wiele problemów lokalizacyjnych sformułowano pod kątem maksymalizacji kryterium równości. W tym celu wprowadza się miarę nierówności  $\varrho$  wektora ocen  $y$ , która podlega minimalizacji. Popularnym przykładem takiej miary jest wariancja, która mierzy bezwzględny rozrzut zbioru wartości i była także używana w modelach lokalizacyjnych [44, 8]. W obszarze zagadnień lokalizacyjnych wprowadzono jednak wiele różnych miar nierówności [49]. Jedną z najbardziej popularnych (zwłaszcza w dziedzinie ekonomii) jest indeks Giniego, który również był używany w modelach lokalizacyjnych [45, 9, 47, 22]. Bezpośrednia minimalizacja miary rozbieżności może jednak prowadzić do niekorzystnych efektów. Przykładem może być pogarszanie (zwiększanie) ocen poprzez oddalanie punktów obsługi od klientów, aby względne różnice między ocenami stały się mniejsze (jest to szczególnie widoczne dla względnych miar nierówności) [22]. W rzeczywistości dotyczy to wszystkich miar nierówności wyznaczanych w oparciu o rozrzut wartości [62]. Aby uniknąć tego typu problemów, stosuje się zazwyczaj modele dwukryterialne

$$\min\left\{\left(\frac{1}{m} \sum_{i=1}^m f_i(\mathbf{x}), \varrho(f_i(\mathbf{x}))\right) : \mathbf{x} \in Q\right\},$$

gdzie minimalizacji, obok miary rozbieżności, podlega również średnia ocen. Taki model pozwala także uwzględnić zróżnicowane zapotrzebowania. W modelu mogą być wykorzystywane różne miary rozbieżności: odchylenie standardowe, górne odchylenie standardowe, maksymalne odchylenie, średnia i maksymalna całkowita różnica itp. Nie wszystkie miary zapewniają jednak uzyskanie rozwiązań efektywnych, a tym samym wyrównująco efektywnych w modelach dwukryterialnych. Muszą być spełnione odpowiednie warunki, których nie spełnia np. wariancja [63]. Szczegółowa analiza zagadnienia znajduje się w opracowaniu [64].

Podejściem zgodnym z relacją wyrównującą jest model leksykograficznego centrum [59]

$$\text{lex min}\{(\theta_1(\mathbf{f}(\mathbf{x})), \theta_2(\mathbf{f}(\mathbf{x})), \dots, \theta_m(\mathbf{f}(\mathbf{x}))) : \mathbf{x} \in Q\}, \quad (3.5)$$

gdzie minimalizacji podlega najpierw ocena największa, następnie przy zachowaniu jej wartości minimalizowana jest druga największa ocena, następnie przy zachowaniu poprzednich wartości minimalizowana jest trzecia największa ocena itd. Równoważny problem można otrzymać przez zastosowanie optymalizacji leksykograficznej do skumulowanego uporządkowanego wektora ocen, co może mieć zalety wydajnościowe. W naturalny sposób podejście to stanowi rozszerzoną (regularyzowaną) wersję standardowego kryterium centrum. W [59, 68] opracowano algorytmy rozwiązywania optymalizacji leksykograficznej dla dyskretnych problemów lokalizacyjnych. W pewnym sensie rozwiązanie problemu (3.5) może być traktowane jako rozmieszczenie najbardziej wyrównujące, czasami określane jako najbardziej sprawiedliwe

(ang. max-min fairness) [43]. Bezpośrednie zastosowanie tego podejścia wiąże się więc z pewną utratą elastyczności modelowania preferencji i generowania zróżnicowanych rozwiązań.

Optymalizacja leksykograficzna może być wyrażona (przybliżona) za pomocą operatora uporządkowanej średniej ważonej. Dzieje się tak, gdy różnice między poszczególnymi wagami preferencji dążą do nieskończoności. Poza elastycznością wynikającą z parametrycznego wektora wag preferencji, zaletą agregacji OWA jest mniejsza złożoność obliczeniowa, gdzie rozwiązaniu podlega jeden problem optymalizacyjny, gdy w modelu leksykograficznym trzeba rozwiązać sekwencję zadań optymalizacyjnych.

### **3.2.5. Problem uporządkowanej średniej**

W ostatnim dziesięcioleciu w zagadnieniach lokalizacyjnych badany był nowy problem, oryginalnie zaproponowany przez Nickela [56], określany mianem problemu uporządkowanej średniej (ang. Ordered Median Problem, OMP) [17, 57, 10]. Kryterium problemu uporządkowanej średniej uogólnia większość najpopularniejszych funkcji celu stosowanych w zagadnieniach lokalizacyjnych, przez co stanowi wartościowe i interesujące podejście. W istocie kryterium uporządkowanej średniej odpowiada agregacji OWA z nieznormalizowanymi wagami preferencji. Jest zatem zgodne z wyrównującą relacją preferencji przy założeniu odpowiednio monotonicznych wag preferencji. Dzięki temu pozwala na uzyskiwanie efektywnych i sprawiedliwych schematów lokalizacyjnych. Podobnie jak OWA nie pozwala uwzględniać różnych wielkości zapotrzebowania.

## **3.3. Podsumowanie**

W literaturze opracowano wiele modeli dyskretnych problemów lokalizacyjnych. Wiele z nich, sformułowanych w postaci zadań programowania matematycznego, różni się tylko funkcją celu. Dzięki zastosowaniu wielokryterialnej optymalizacji modele te mogą być ujednoczone przez zastosowanie odpowiedniej funkcji skalaryzującej, a mianowicie uporządkowanej średniej ważonej (OWA) — szczegóły są przedstawione w rozdziale 4. Z jednej strony daje to możliwość jednolitego podejścia do szerokiej gamy dyskretnych zagadnień lokalizacyjnych. Z drugiej strony zwiększa elastyczność modelowania oraz poszukiwania kompromisowych schematów lokalizacyjnych o pożądanym, złożonym własnościach dystrybucyjnych. Jak wspomniano w rozdziale 2, a co jest szczegółowo wyjaśnione w rozdziale 4, operator OWA pozwala na uzyskiwanie symetrycznie efektywnych, a także wyrównująco efektywnych rozwiązań. Co więcej, w przeciwieństwie do optymalizacji leksykograficznej i dwukryterialnych modeli z miami nierówności, operator OWA pozwala na większą elastyczność różnicowania rozwiązań, także w ramach zbiorów rozwiązań symetrycznie i wyrównująco efektywnych. Agregacja OWA nie pozwala natomiast na uwzględnienie zróżnicowanych zapotrzebowań, co było możliwe w części prezentowanych modeli. Aby uwzględnić zróżnicowane zapotrzebowania i zachować elastyczność, w późniejszej części pracy zastosowany będzie operator WOWA.

## 4. Uporządkowana średnia ważona

W rozdziale przedstawiony jest operator uporządkowanej średniej ważonej. W niniejszej pracy jest on stosowany jako parametryczny model preferencji dla dyskretnych zagadnień lokalizacyjnych z równymi wartościami zapotrzebowania poszczególnych klientów. Jego wykorzystanie pozwala na jednolite potraktowanie wielu problemów lokalizacyjnych, a także umożliwia uzyskiwanie rozwiązań optymalnych w sensie rozkładu, w tym symetrycznie i wyrównująco efektywnych.

### 4.1. Definicja operatora OWA

Uporządkowana średnia ważona (ang. Ordered Weighted Averaging, OWA) wprowadzona przez Yagera [91] to średnia ważona, w której wagi są przypisane do wartości uporządkowanych (do wartości największej, drugiej największej itd.) zamiast do konkretnych współrzędnych. Formalnie ujmuje to poniższa definicja.

**Definicja 4.1.** Dany jest  $m$ -wymiarowy wektor wag  $\mathbf{w} = (w_1, \dots, w_m)$ . Operator agregacji OWA wektora  $\mathbf{y} = (y_1, \dots, y_m)$  wyraża się jako

$$A_{\mathbf{w}}(\mathbf{y}) = \sum_{i=1}^m w_i \theta_i(\mathbf{y}), \quad (4.1)$$

gdzie  $\Theta: R^m \rightarrow R^m$  to odwzorowanie takie, że  $\Theta(\mathbf{y}) = (\theta_1(\mathbf{y}), \theta_2(\mathbf{y}), \dots, \theta_m(\mathbf{y}))$  spełnia warunek  $\theta_1(\mathbf{y}) \geq \theta_2(\mathbf{y}) \geq \dots \geq \theta_m(\mathbf{y})$  oraz istnieje permutacja  $\tau$  zbioru  $I$ , taka że  $\theta_i(\mathbf{y}) = y_{\tau(i)}$  dla  $i = 1, 2, \dots, m$ .

Yager zdefiniował operator OWA ze znormalizowanymi wagami, czyli spełniającymi warunki  $0 \leq w_i \leq 1$  dla  $i = 1, 2, \dots, m$  oraz  $\sum_{i=1}^m w_i = 1$ . Dla takich wag operator OWA spełnia tzw. zasadę jednomyślności, która stanowi

$$\text{jeśli } y_i = b \text{ dla } i = 1, 2, \dots, m \Rightarrow A_{\mathbf{w}}(\mathbf{y}) = b,$$

a wartość operatora OWA zawiera się pomiędzy minimalną i maksymalną składową wektora ocen  $\mathbf{y}$ , czyli

$$\min_{i=1, \dots, m} y_i \leq A_{\mathbf{w}}(\mathbf{y}) \leq \max_{i=1, \dots, m} y_i.$$

W pracy nie jest wymagana normalizacja wag preferencji agregacji OWA (4.1), a jedynie ich nieujemność. Z praktycznego punktu widzenia bardzo łatwo wyznaczyć wartość operatora

OWA z wagami znormalizowanymi ( $w_i^{norm}$ ) na podstawie wartości operatora OWA z dowolnymi wagami nieujemnymi, gdyż tą drugą wartość wystarczy podzielić przez sumę wag nieznormalizowanych

$$A_w^{norm}(\mathbf{y}) = \sum_{i=1}^m w_i^{norm} \theta_i(\mathbf{y}) = \sum_{i=1}^m \frac{w_i}{\sum_{i'=1}^m w_{i'}} \theta_i(\mathbf{y}) = \frac{1}{\sum_{i'=1}^m w_{i'}} \sum_{i=1}^m w_i \theta_i(\mathbf{y}) = \frac{A_w(\mathbf{y})}{\sum_{i'=1}^m w_{i'}}.$$

Takie przeskalowanie przez stałą nie zmienia rozwiązania zadania optymalizacji, więc brak normalizacji nie zmniejsza ogólności prowadzonych rozważań.

**Przykład 4.1.** Przykład prezentuje sposób wyznaczania wartości operatora OWA. Rozpatrzmy problem z 5 lokalizacjami. Załóżmy, że osiągalny wektor kosztów  $\mathbf{y} = (1; 3; 2; 4; 5)$ , a wektor wag preferencji  $\mathbf{w} = (0; 2; 2; 1; 0)$ . Uporządkowany wektor kosztów to  $\Theta(\mathbf{y}) = (5; 4; 3; 2; 1)$ , a wartość operatora OWA to  $A_w(\mathbf{y}) = 0 \cdot 5 + 2 \cdot 4 + 2 \cdot 3 + 1 \cdot 2 + 0 \cdot 1 = 16$ .

## 4.2. Parametryczny model preferencji

Operator OWA pozwala wyrazić wiele funkcji celu stosowanych w różnych problemach lokalizacyjnych. Dzięki temu problem lokalizacyjny z agregacją OWA uogólnia wiele badanych dotychczas zagadnień lokalizacyjnych. Stosując odpowiednie wagi preferencji  $\mathbf{w}$ , można modelować większość popularnych kryteriów (w tym te przedstawione w rozdziale 3, przy założeniu równych wag zapotrzebowania):

—  $w_i = 1/m$  dla  $i = 1, \dots, m$  odpowiada kryterium średniej (równoważne sumie ocen, gdy wszystkie wagi równe 1)

$$A_w(\mathbf{y}) = \sum_{i=1}^m \frac{1}{m} \theta_i(\mathbf{y}) = \frac{1}{m} \sum_{i=1}^m y_i = \mu(\mathbf{y}),$$

—  $w_1 = 1$  i  $w_i = 0$  dla  $i = 2, \dots, m$  odpowiada kryterium centrum (maksimum)

$$A_w(\mathbf{y}) = \theta_1(\mathbf{y}) = \max_{i=1, \dots, m} y_i = M(\mathbf{y}),$$

—  $w_1 = 1$  i  $w_i = 1 - \lambda$  dla  $i = 2, \dots, m$  oraz  $0 \leq \lambda \leq 1$  odpowiada kryterium centro-średniej

$$A_w(\mathbf{y}) = \theta_1(\mathbf{y}) + (1 - \lambda) \sum_{i=2}^m \theta_i(\mathbf{y}) = \lambda \theta_1(\mathbf{y}) + (1 - \lambda) \sum_{i=1}^m \theta_i(\mathbf{y}) = H_\lambda(\mathbf{y})$$

—  $w_m = 1$  i  $w_i = 0$  dla  $i = 1, \dots, m - 1$  odpowiada minimum

$$A_w(\mathbf{y}) = \theta_m(\mathbf{y}) = \min_{i=1, \dots, m} y_i.$$

—  $w_1 = \lambda$ ,  $w_m = 1 - \lambda$  i  $w_i = 0$  dla  $i = 2, \dots, m - 1$  oraz  $0 \leq \lambda \leq 1$  odpowiada tzw.

kryterium Hurwicza [34]

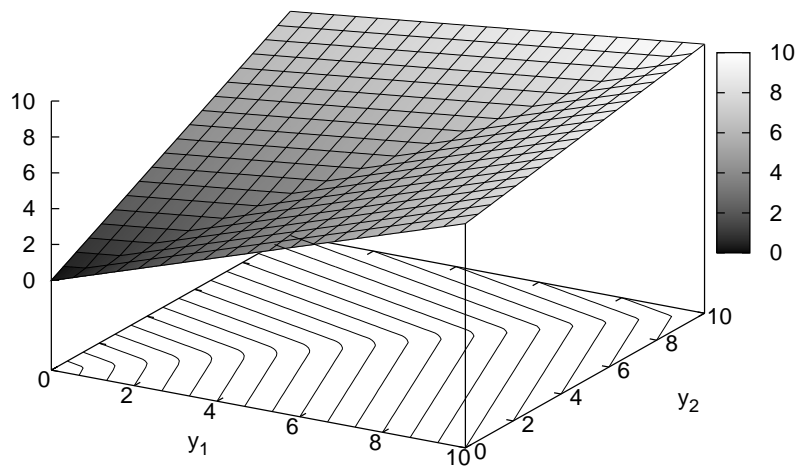
$$A_w(\mathbf{y}) = \lambda \theta_1(\mathbf{y}) + (1 - \lambda) \theta_m(\mathbf{y}) = \lambda \max_{i=1, \dots, m} y_i + (1 - \lambda) \min_{i=1, \dots, m} y_i.$$

Za pomocą operatora OWA można również przybliżyć kryterium leksykograficzne. Gdy różnice między malejącymi wagami dążą do nieskończoności ( $w_1 \gg w_2 \gg \dots \gg w_m$ ), otrzymuje się leksykograficzną minimaksymalizację, a gdy analogiczna sytuacja dotyczy wag rosnących ( $w_1 \ll w_2 \ll \dots \ll w_m$ ), wówczas wynikiem jest leksykograficzna miniminimalizacja. Możliwości agregacji OWA są jednak jeszcze większe, gdyż pozwalają uogólniać dotychczasowe i tworzyć nowe kryteria:

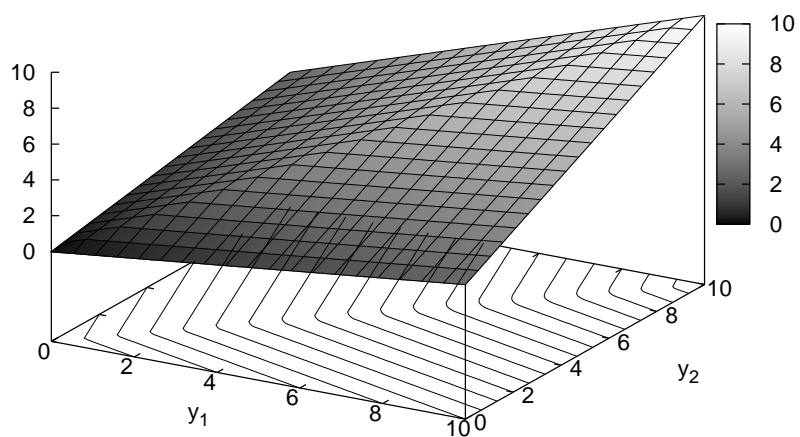
- $w_i = 1/k$  dla  $i = 1, \dots, k$ ,  $w_i = 0$  dla  $i = k + 1, \dots, m$  odpowiada kryterium  $k$ -tej największej średniej, które uogólnia kryterium centrum i wyznacza średnią  $k$  największych ocen (równoważne sumie  $k$  największych ocen, gdy wszystkie wagi równe 1),
- $w_i = 1/k$  dla  $i = m - k + 1, \dots, m$ ,  $w_i = 0$  dla  $i = 1, \dots, m - k$  odpowiada kryterium  $k$ -tej najmniejszej średniej, które uogólnia kryterium minimum i wyznacza średnią  $k$  najmniejszych ocen,
- $w_i = \lambda$  dla  $i = 1, \dots, k$ ,  $w_i = 1 - \lambda$  dla  $i = m - k + 1, \dots, m$  i  $w_i = 0$  dla  $i = k + 1, \dots, m - k$  oraz  $0 \leq \lambda \leq 1$  i  $2k \leq m$  odpowiada uogólnionemu kryterium Hurwicza, które wyznacza kombinację liniową sumy  $k$  największych i najmniejszych ocen,
- $w_i = 1$  dla  $i = k_1 + 1, \dots, m - k_2$ ,  $w_i = 0$  dla  $i = 1, \dots, k_1, m - k_2 + 1, \dots, m$  odpowiada tzw. kryterium zawężonej średniej (ang.  $(k_1 + k_2)$ -trimmed mean), która wyznacza sumę  $m - k_1 - k_2$  pewnych środkowych ocen.

Przedstawione powyżej kryteria i ich kolejne możliwe wariacje pokazują, że agregacja OWA z parametrycznym wektorem wag zapewnia szeroki wachlarz modeli preferencji [92]. W istocie operator OWA pozwala wyrażać kryteria kompromisowe w całym spektrum preferencji od koniunkcji do alternatywy poszczególnych ocen. Uporządkowana średnia ważona jest obszarami liniową funkcją wypukłą, gdy  $w_1 \geq w_2 \geq \dots \geq w_m$  i wklęsłą, gdy  $w_1 \leq w_2 \leq \dots \leq w_m$  (porównaj rysunki 4.1 i 4.2). Możliwe kształty warstwicy agregacji OWA dla różnych wag preferencji są przedstawione na rysunku 4.3. Niestety, podobnie jak metoda ważenia w oryginalnym problemie, tak uporządkowana średnia ważona nie stanowi zupełnej parametryzacji zbioru rozwiązań symetrycznie/wyrównująco efektywnych. W problemach dyskretnych mogą istnieć rozwiązania symetrycznie/wyrównująco efektywne, które nie stanowią rozwiązań optymalnych minimalizacji OWA dla żadnego zbioru dodatnich wag. Obrazuje to następujący przykład.

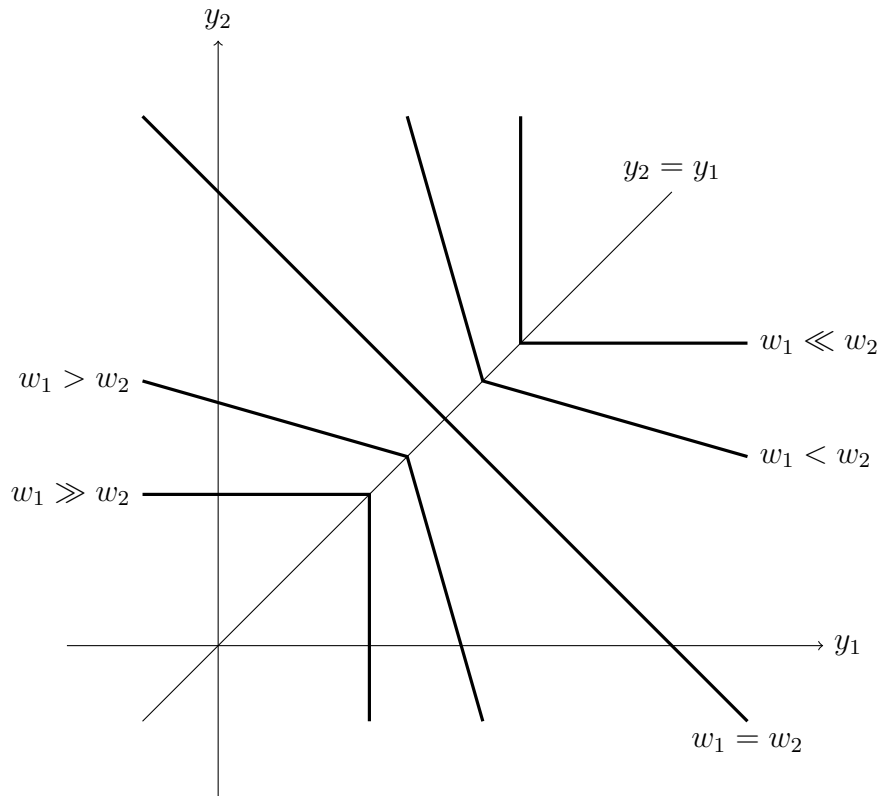
**Przykład 4.2.** Dany jest zbiór trzech wektorów  $y_1 = (2, 9)$ ,  $y_2 = (5, 8)$  i  $y_3 = (7, 7)$ . Każdy z tych wektorów jest wyrównująco niezdominowany (zatem również symetrycznie niezdominowany). Okazuje się jednak, że wektor  $y_2$  nie jest optymalny dla żadnych dodatnich monotonicznych wag agregacji OWA. Gdy  $2w_1 < 5w_2$ , to  $y_1$  jest jednoznacznym rozwiązaniem optymalnym. Jeśli natomiast  $2w_1 > 5w_2$ , to jednoznacznym rozwiązaniem optymalnym jest wektor  $y_3$ . Dla  $2w_1 = 5w_2$  oba wektory są optymalne. Aby wektor  $y_2$  był optymalny, wagi



Rysunek 4.1. Wykres i warstwyce przykładowej agregacji OWA dwóch kryteriów ( $w_1 > w_2$ )



Rysunek 4.2. Wykres i warstwyce przykładowej agregacji OWA dwóch kryteriów ( $w_1 < w_2$ )



Rysunek 4.3. Możliwe kształty warstwic agregacji OWA dwóch kryteriów

musiałyby spełniać nierówności  $w_1 > 3w_2$  i  $w_1 < 2w_2$ , co oczywiście nie jest możliwe dla  $w_1 > w_2 > 0$ .

Stosując operator OWA do agregacji ocen w problemach optymalizacji wielokryterialnej otrzymuje się zadanie

$$\min \left\{ \sum_{i=1}^m w_i \theta_i(\mathbf{y}) : \mathbf{y} = \mathbf{f}(\mathbf{x}), \mathbf{x} \in Q \right\}. \quad (4.2)$$

Jak wynika z rozdziału 2 (podrozdział 2.4.1), rozwiązanie optymalne zadania (4.2) dla dowolnych dodatnich wag  $w_i$  ( $i = 1, \dots, m$ ) stanowi symetrycznie efektywne rozwiązanie bazowego zadania wielokryterialnego (1.1).

W rozdziale 2 pokazano również, że wyrównująco efektywne rozwiązanie zadania wielokryterialnego (1.1) można uzyskać jako rozwiązanie optymalne zadania

$$\min \left\{ \sum_{i=1}^m w_i \bar{\theta}_i(\mathbf{y}) : \mathbf{y} = \mathbf{f}(\mathbf{x}), \mathbf{x} \in Q \right\}$$

z dodatnimi wagami  $w_i$  i operatorem agregacji  $\bar{\theta}_i(\mathbf{y}) = \sum_{j=1}^i \theta_j(\mathbf{y})$  dla  $i = 1, \dots, m$ . Ze względu na definicję operatora kumulacji  $\bar{\Theta}$  powyższe zadanie jest równoważne zadaniu skalaryzacji problemu wielokryterialnego (1.1) przez agregację OWA ze zmodyfikowanymi

wagami. Formalnie zadanie to wyraża się jako

$$\min\left\{\sum_{i=1}^m \bar{w}_i \theta_i(\mathbf{y}) : \mathbf{y} = \mathbf{f}(\mathbf{x}), \mathbf{x} \in Q\right\},$$

gdzie  $\bar{w}_i = \sum_{j=i}^m w_j$  dla  $i = 1, 2, \dots, m$ . Wychodząc od zadania z uporządkowaną średnią ważoną

$$\min\left\{\sum_{i=1}^m w_i \theta_i(\mathbf{y}) : \mathbf{y} = \mathbf{f}(\mathbf{x}), \mathbf{x} \in Q\right\}, \quad (4.3)$$

równoważne zadanie ma postać

$$\min\left\{\sum_{i=1}^m w'_i \bar{\theta}_i(\mathbf{y}) : \mathbf{y} = \mathbf{f}(\mathbf{x}), \mathbf{x} \in Q\right\} \quad (4.4)$$

z wagami określonymi przez zależności  $w'_m = w_m$  i  $w'_i = w_i - w_{i+1}$  dla  $i = 1, 2, \dots, m-1$ . Z warunku na dodatność wag  $w'_i$  dla  $i = 1, 2, \dots, m$  wynika, że każde rozwiązanie problemu OWA (4.3), gdzie wagi spełniają warunek  $w_1 > w_2 > \dots > w_m > 0$ , stanowi wyrównująco efektywne rozwiązanie problemu wielokryterialnego (1.1). Dla zupełności warto zaznaczyć, że agregacja OWA z wagami monotonicznymi jest symetryczną funkcją wypukłą, jednak nie ściśle wypukłą. Mimo to spełnia warunek przesunięć wyrównujących.

Zadanie optymalizacji OWA z odpowiednimi wagami preferencji pozwala więc uzyskiwać rozwiązania o pożądanym, bardziej złożonym własnościach dystrybucyjnych. Dodatkowo wagi preferencji zapewniają uzyskanie rozwiązania spełniającego własność bezstronności, w którym ważniejszy jest rozkład wartości niż ich przypisanie do konkretnych kryteriów. Jeśli wagi preferencji są dodatkowo ściśle monotoniczne (malejące), wówczas osiągnięte są rozwiązania sprawiedliwe (wyrównująco efektywne). Dzięki tym własnościom operator OWA sprawdza się w systemach z wieloma niezależnymi użytkownikami (agentami), których oceny odpowiadają kryteriom sformułowanego modelu.

### 4.3. Złożoność obliczeniowa

Problem optymalizacji wielokryterialnej można podzielić na dwie części: model rzeczowy i model preferencji. W pracy model rzeczowy stanowi dyskretny problem lokalizacyjny. Jak pokazano w rozdziale 3, może być on sformułowany jako zadanie programowania całkowitoliczbowego liniowego mieszanego (PCLM). Agregacja ocen stanowi z kolei model preferencji. Już sam model rzeczowy jest problemem bardzo trudnym — dyskretny problem lokalizacyjny z kryterium średniej jest problemem NP-trudnym [36]. Jak wcześniej pokazano, operator OWA uogólnia kryterium średniej (kryterium średniej jest szczególnym przypadkiem operatora OWA), co oznacza, że dyskretny problem lokalizacyjny z agregacją OWA również jest problemem NP-trudnym. Tym samym dokładne metody rozwiązywania będą w stanie rozwiązywać dyskretny problem lokalizacyjny z agregacją OWA do pewnego (zazwyczaj



niewielkiego) rozmiaru granicznego. Aby granica ta była jak największa, konieczne jest jak najefektywniejsze sformułowanie modelu preferencji w postaci operatora OWA.

W ogólności agregacja OWA może być wyrażona jako zadanie PCLM. Jednak w szczególnych przypadkach może być sformułowana w postaci zadania programowania liniowego (PL). Oczywiście cały problem optymalizacyjny pozostanie zadaniem PCLM, ze względu na dyskretny model rzeczowy. W dalszej części pracy, jeśli nie zaznaczono inaczej, określenia zadań PCLM i PL będą używane w kontekście modelu preferencji.

W kolejnym rozdziale analizowane są różne modele agregacji OWA i badana jest ich wydajność obliczeniowa dla dyskretnych wielokryterialnych problemów lokalizacyjnych. Sprawdzana jest także możliwość poprawy wydajności obliczeniowej zadań optymalizacyjnych poprzez zastosowanie ograniczeń nadmiarowych do ściślejszego opisu zbioru rozwiązań dopuszczalnych. Potencjalny pozytywny wpływ ograniczeń nadmiarowych jest znany w problemach optymalizacji kombinatorycznej [11]. Znalezienie efektywnych nadmiarowych ograniczeń dla konkretnych problemów i sformułowań jest jednak zadaniem nietrywialnym i wymaga zazwyczaj badań eksperymentalnych. Zweryfikowane w ten sposób ograniczenia są powszechnie wykorzystywane przy rozwiązywaniu i dalszej analizie problemów. Przykłady takich ograniczeń można znaleźć także w obszarze problemów lokalizacyjno-dystrybucyjnych, np. nadmiarowe ograniczenia wiążące pojemności i zapotrzebowania w pewnych problemach lokalizacyjnych (ang. Capacitated Plant Location Problem) [33], przy czym te ograniczenia są wprowadzane w modelu rzeczowym i dotyczą problemów, które różnią się nieco od problemów badanych w pracy. Ostatnio nadmiarowe ograniczenia były również analizowane w modelach operatora OWA dla problemów najkrótszej ścieżki w grafie (ang. Shortest Path Problem) oraz doskonałego skojarzenia w grafie (ang. Perfect Matching Problem) [23], przy czym ograniczono się tam do wektora wag preferencji reprezentującego kryterium Hurwicza.

## 5. Modele optymalizacyjne OWA

Agregacja OWA, od momentu jej wprowadzenia, była wykorzystywana do rozwiązywania problemów w różnych dziedzinach [94], w tym m.in. w telekomunikacji [71, 37] oraz w zagadnieniach lokalizacyjnych [57]. Z punktu złożoności obliczeniowej operator porządkujący  $\Theta$  sprawia, że problem staje się nieliniowy, nawet jeśli pierwotny problem był sformułowany w postaci programowania liniowego. Yager [93] pokazał również, że nieliniowość ta może zostać przekształcona do zadania postaci PCLM (ang. Mixed Integer Linear Programming, MILP). Dodatkowo Ogryczak i Śliwiński [67] pokazali, że w przypadku specjalnych zadań z kryterium uporządkowanej średniej ważonej z wagami monotonicznymi, problem może być sformułowany w postaci standardowego zadania PL o wyższym wymiarze (większa liczba zmiennych).

W ramach dyskretnych problemów lokalizacyjnych zostało opracowanych kilka modeli optymalizacyjnych uporządkowanej średniej ważonej (OWA). Część z nich wywodzi się bezpośrednio z operatora OWA [72], a część od tzw. problemów uporządkowanej średniej (ang. Ordered Median Problems, OMP) [10]. Modele te badano na różnych problemach lokalizacyjnych, ale nigdy nie porównano ich wydajności obliczeniowej. W ramach pracy wykonano takie porównanie, a dodatkowo wyniki modeli PCLM zestawiono z rezultatami liniowych modeli OWA dla szczególnych przypadków problemów. Zaproponowano również nowy model optymalizacyjny kryterium OWA będący rozszerzeniem modelu liniowego. Może być on zastosowany dla problemów z dowolnymi wagami preferencji przy zastosowaniu mniejszej liczby zmiennych binarnych w porównaniu do wcześniejszych modeli PCLM. W rozdziale przeanalizowano poszczególne modele optymalizacyjne OWA, rozpatrując przy tym możliwość wprowadzenia do nich nadmiarowych ograniczeń. Porównano poszczególne modele i ich sformułowania różniące się liczbą uwzględnianych ograniczeń.

### 5.1. Model PCLM M1

Nieliniowość operatora  $\Theta$  może być przekształcona do zadania PCLM. Można zauważyć, że największa ocena  $\theta_1(\mathbf{y})$  dowolnego wektora  $\mathbf{y} \in A$  może być łatwo wyznaczona przez zadanie programowania liniowego minimalizacji:

$$\begin{aligned} \theta_1(\mathbf{y}) = \min_{\hat{y}_1} \quad & \hat{y}_1 \\ \text{p.o.} \quad & \hat{y}_1 \geq y_i \quad \text{dla } i = 1, \dots, m. \end{aligned} \tag{5.1}$$

Zgodnie z propozycją Yagera [93] w podobny sposób można wyznaczyć dowolną  $k$ -tą największą ocenę  $\theta_k(\mathbf{y})$ , przy czym wymaga to wprowadzenia zmiennych binarnych. Dla dowolnego  $\mathbf{y} \in A$  i dowolnego  $k = 1, 2, \dots, m$  można sformułować zadanie:

$$\theta_k(\mathbf{y}) = \min_{\hat{y}_k, z_{ik}} \hat{y}_k \quad (5.2a)$$

$$\text{p.o. } \hat{y}_k + M z_{ik} \geq y_i \quad \text{dla } i = 1, \dots, m, \quad (5.2b)$$

$$\sum_{i=1}^m z_{ik} \leq k - 1, \quad (5.2c)$$

$$z_{ik} \in \{0, 1\} \quad \text{dla } i = 1, \dots, m. \quad (5.2d)$$

gdzie  $M$  jest odpowiednio dużą stałą (nie mniejszą niż największa możliwa różnica pomiędzy składowymi wektora ocen). Sformułowanie wyszukuje  $k$ -tą największą wartość  $\hat{y}_k$ , wykorzystując zmienne binarne  $z_{ik}$  do relaksacji części ograniczeń (5.2b), co umożliwi pominięcie  $k - 1$  największych wartości przy minimalizacji zmiennej  $\hat{y}_k$ . Zmienna  $z_{ik}$  jest równa 1, jeśli  $\hat{y}_k$  jest mniejsze od  $y_i$ , a równa 0 w przeciwnym przypadku. Można zauważyć, że dla  $k = 1$  wszystkie zmienne binarne  $z_{1i}$  są równe 0, co prowadzi do zadania PL (5.1).

Wykorzystując zadanie (5.2), optymalizację OWA można sformułować jako zadanie PCLM:

$$\min_{\hat{y}_k, z_{ik}, y_i} \sum_{k=1}^m w_k \hat{y}_k, \quad (5.3a)$$

$$\text{p.o. } \hat{y}_k + M z_{ik} \geq y_i \quad \text{dla } i, k = 1, \dots, m, \quad (5.3b)$$

$$\sum_{i=1}^m z_{ik} \leq k - 1 \quad \text{dla } k = 1, \dots, m, \quad (5.3c)$$

$$z_{ik} \in \{0, 1\} \quad \text{dla } i, k = 1, \dots, m, \quad (5.3d)$$

$$\mathbf{y} \in A, \quad (5.3e)$$

gdzie  $w_k$  jest  $k$ -tą wagą agregacji OWA. To sformułowanie agregacji OWA wprowadza  $O(m^2)$  zmiennych binarnych  $z_{ik}$  występujących w  $m$  ograniczeniach wielokrotnego wyboru (ang. multiple choice constraints) (5.3c) oraz  $m$  zmiennych ciągłych  $\hat{y}_k$ , z których każda jest zdefiniowana przez  $m$  odpowiednich nierówności (5.3b). Tym samym sformułowanie wprowadza  $O(m^2)$  ograniczeń.

Oryginalny model (5.3), wprowadzony w [93], zawierał dodatkowe ograniczenia

$$\hat{y}_k \geq \hat{y}_{k+1} \quad \text{dla } k = 1, \dots, m - 1 \quad (5.4)$$

reprezentujące jawne nierosnące uporządkowanie zmiennych  $\hat{y}_k$ . Jednak ze względu na minimalizację z nieujemnymi wagami  $w_k$ , ograniczenia te są nadmiarowe w tym sensie, że istnieją rozwiązania optymalne zadania (5.3), które spełniają te ograniczenia (gdy wszystkie wagi  $w_k$

dla  $k = 1, \dots, m$  są dodatnie, wówczas każde rozwiązanie optymalne spełnia te ograniczenia). Interesujące było jednak pytanie, czy ograniczenia te mogą poprawić wydajność obliczeniową modelu. W związku z tym zbadany został ich wpływ na czasy osiągnięte przez sformułowanie (5.3). Rozpatrzono również inne ograniczenia nadmiarowe — ich nadmiarowość pokazuje poniższe twierdzenie.

**Twierdzenie 5.1.** Zadanie (5.3) posiada rozwiązanie optymalne, które spełnia ograniczenia:

(i) niemalejącego uporządkowania zmiennych binarnych  $z_{ik}$  dotyczących  $i$ -tej oceny

$$z_{ik} \leq z_{ik+1} \quad \text{dla} \quad i = 1, \dots, m; k = 1, \dots, m - 1. \quad (5.5)$$

(ii) zbilansowania sumy ocen oryginalnego wektora wyjściowego  $y$  i uporządkowanego wektora  $\hat{y}$

$$\sum_{k=1}^m \hat{y}_k = \sum_{i=1}^m y_i. \quad (5.6)$$

*Dowód.* W ramach dowodu przeanalizowane zostanie wybrane rozwiązanie optymalne zadania (5.3) i pokazane, że spełnia ono ograniczenia (i)–(ii).

Zgodnie ze wcześniejszymi obserwacjami zadanie (5.3) posiada rozwiązanie optymalne, gdzie zmienne  $\hat{y}_k^*$  odpowiadają  $k$ -tym największym składowym wektora ocen  $y^* \in A$ . Dla dowolnego  $i = 1, \dots, m$  i  $k = 1, \dots, m - 1$  rozważono zależność między zmiennymi  $\hat{y}_k^*$  i  $y_i^*$ . Zgodnie z ograniczeniem (5.3b), gdy  $\hat{y}_k^*$  jest mniejsza od  $y_i^*$ , to zmienna  $z_{ik}^*$  jest równa 1. W takim przypadku, ze względu na nierosnące uporządkowanie składowych  $\hat{y}^*$ , także zmienna  $\hat{y}_{k+1}^*$  jest mniejsza od  $y_i^*$ , a tym samym  $z_{ik+1}^*$  jest równe 1. Gdy  $\hat{y}_k^*$  jest nie mniejsza od  $y_i^*$ , to zmienna  $z_{ik}^*$  jest równa 0 (ze względu na minimalizację). Z kolei zmienna  $z_{ik+1}^*$  jako zmienna binarna może być równa 0 lub 1. Stąd wynika, że w obu rozpatrywanych przypadkach  $z_{ik}^* \leq z_{ik+1}^*$  dla dowolnego  $i = 1, \dots, m$  i  $k = 1, \dots, m - 1$ , czyli analizowane rozwiązanie optymalne spełnia ograniczenie (5.5). Spełnienie ograniczenia (5.6) wynika wprost z  $\hat{y}^* = \Theta(y^*)$ .

Podsumowując, zadanie (5.3) posiada rozwiązanie optymalne, które spełnia ograniczenia (i)–(ii). □

Gdy wszystkie wagi  $w_k$  ( $k = 1, \dots, m$ ) są dodatnie, wówczas w każdym rozwiązaniu optymalnym zmienne  $\hat{y}_k^*$  odpowiadają  $k$ -tym największym ocenom. Z twierdzenia (5.1) wynika wówczas, że każde rozwiązanie optymalne zadania (5.3) spełnia ograniczenia (5.5)–(5.6) (podobnie jak ograniczenie (5.4)). Warto również zaznaczyć, że wektory zmiennych  $\hat{y}$  oraz  $z$  mają charakter pomocniczy w odniesieniu do całego problemu lokalizacyjnego. Rozpatrując zbiór wektorów osiągalnych  $y \in A$  i odpowiadający mu zbiór rozwiązań dopuszczalnych określony przez zmienne decyzje  $x \in Q$ , można stwierdzić, że zbiory te nie zależą od ograniczeń (5.4)–(5.6).

Model (5.3) będzie dalej oznaczany jako M1. Aby zbadać wpływ ograniczeń nadmiarowych na wydajność obliczeniową modelu M1, porównano wyniki czasowe sformułowań różniących się obecnością poszczególnych ograniczeń. Rozpatrzono pięć sformułowań:

- M1<sub>1</sub> — sformułowanie bez ograniczeń nadmiarowych,
- M1<sub>2</sub> — sformułowanie z jednym nadmiarowym ograniczeniem (5.4),
- M1<sub>3</sub> — sformułowanie z dwoma nadmiarowymi ograniczeniami (5.4), (5.6),
- M1<sub>4</sub> — sformułowanie z jednym nadmiarowym ograniczeniem (5.5),
- M1<sub>5</sub> — sformułowanie z dwoma nadmiarowymi ograniczeniami (5.5), (5.6).

## 5.2. Model PCLM M2

Wiele modeli matematycznego programowania całkowitoliczbowego operatora OWA było rozwijanych w ramach zagadnień lokalizacyjnych określanych mianem problemów uporządkowanej średniej (ang. Ordered Median Problems, OMP) [57]. Zaczynając od kwadratowego zadania całkowitoliczbowego, przez model PCLM z  $O(m^3)$  zmiennymi binarnymi, aż do modelu PCLM o  $O(m^2)$  zmiennych binarnych i ograniczeniach [10]. Ten ostatni model, po uwzględnieniu używanej w pracy notacji, prezentuje się następująco:

$$\min_{\hat{y}_k, s_{ik}, y_i} \sum_{k=1}^m w_k \hat{y}_k, \quad (5.7a)$$

$$\text{p.o. } \hat{y}_k \geq \hat{y}_{k+1} \quad \text{dla } k = 1, \dots, m-1, \quad (5.7b)$$

$$\hat{y}_k + M(1 - s_{ik}) \geq y_i \quad \text{dla } i, k = 1, \dots, m, \quad (5.7c)$$

$$\sum_{i=1}^m s_{ik} = 1 \quad \text{dla } k = 1, \dots, m, \quad (5.7d)$$

$$\sum_{k=1}^m s_{ik} = 1 \quad \text{dla } i = 1, \dots, m, \quad (5.7e)$$

$$\sum_{k=1}^m \hat{y}_k = \sum_{i=1}^m y_i, \quad (5.7f)$$

$$s_{ik} \in \{0, 1\} \quad \text{dla } i, k = 1, \dots, m, \quad (5.7g)$$

$$\mathbf{y} \in A, \quad (5.7h)$$

gdzie  $m^2$  zmiennych binarnych  $s_{ik}$  stanowi przypisanie oryginalnych ocen  $y_i$  do ich uporządkowanych odpowiedników  $\hat{y}_k$ . Zmienna  $s_{ik}$  jest równa 1, jeśli ocena  $y_i$  jest  $k$ -tą największą oceną i wynosi 0 w przeciwnym przypadku.

Sformułowanie (5.7) bazuje na połączeniu zadania alokacji i sortowania, przy czym zostało zmodyfikowane przez przekształcenie zaczerpnięte z kwadratowego zagadnienia przydziału (ang. Quadratic Assignment Problem, QAP), co pozwoliło na ograniczenie liczby zmiennych binarnych z  $m^3$  do  $m^2$ . Sortowanie jest zapewnione przez ograniczenia (5.7b), (5.7d) i (5.7e). W tym przypadku  $m-1$  ograniczeń (5.7b), porządkujących wektor ocen, jest koniecznych. Jednak, przy minimalizacji z nieujemnymi wagami, nadmiarowe jest ograniczenie bilansujące (5.7f) (analogicznie jak w modelu M1). Można je w takim razie pominąć. Jednak warto zaznaczyć, że pełne sformułowanie uwzględniające ograniczenia (5.7f), może być zastosowane

zarówno do minimalizacji, jak i maksymalizacji. W związku z tym dla tego modelu, określonego dalej mianem M2, rozpatrzono dwa sformułowania:

- M2<sub>1</sub> — pełne sformułowanie (5.7),
- M2<sub>2</sub> — sformułowanie (5.7) bez ograniczenia nadmiarowego (5.7f).

### 5.3. Zależności między modelami M1 i M2

W modelu M2 ograniczenia (5.7c) pozostają prawdziwe we wzmocnionej postaci

$$\hat{y}_k + M(1 - \sum_{j=k}^m s_{ij}) \geq y_i \quad \text{dla } i, k = 1, \dots, m, \quad (5.8)$$

gdyż  $\hat{y}_k$  jest większy równy nie tylko od przypisanej mu wartości  $y_i$ , ale także od wartości przypisanych  $\hat{y}_j$  dla  $j > k$ . Wyrażenie  $1 - \sum_{j=k}^m s_{ij}$  przyjmuje wartość 1, gdy wartość  $\hat{y}_k$  jest mniejsza od  $y_i$ . Zależność ta jest analogiczna do ograniczenia (5.3b) w modelu M1. I faktycznie podstawiając

$$z_{ik} = 1 - \sum_{j=k}^m s_{ij} \quad \text{dla } i, k = 1, \dots, m$$

do (5.8), otrzymuje się ograniczenie (5.3b) z modelu M1.

Można również wyznaczyć zależność odwrotną pomiędzy zmiennymi binarnymi, gdzie zmienne  $s_{ik}$  można wyrazić za pomocą zmiennych  $z_{ik}$  jako

$$s_{ik} = \begin{cases} z_{ik+1} - z_{ik} & \text{dla } i = 1, \dots, m; k = 1, \dots, m-1, \\ 1 - z_{ik} & \text{dla } i = 1, \dots, m; k = m. \end{cases} \quad (5.9)$$

Podstawiając teraz (5.9) do ograniczenia (5.7d) dla  $k = 1, \dots, m-1$  otrzymujemy

$$1 = \sum_{i=1}^m (z_{ik+1} - z_{ik}) = \sum_{i=1}^m z_{ik+1} - \sum_{i=1}^m z_{ik} \leq k - \sum_{i=1}^m z_{ik}, \quad (5.10)$$

gdyż  $k+1$  największa wartość jest mniejsza od co najwyżej  $k$  wartości. Podobnie dla  $k = m$  otrzymujemy

$$1 = \sum_{i=1}^m (1 - z_{ik}) = m - \sum_{i=1}^m z_{ik},$$

a biorąc pod uwagę minimalizację

$$\sum_{i=1}^m z_{ik} \leq m - 1. \quad (5.11)$$

Tym samym zależności (5.10) i (5.11) dają w rezultacie ograniczenie (5.3c) z modelu M1.

Podstawiając (5.9) do ograniczenia (5.7e) uzyskujemy z kolei

$$\sum_{k=1}^{m-1} (z_{ik+1} - z_{ik}) + (1 - z_{im}) = 1 \quad \text{dla } i = 1, \dots, m,$$

czyli  $z_{i1} = 0$  dla  $i = 1, \dots, m$ , co jest już zapewnione przez ograniczenie (5.3c). Uwzględniając powyższe, można stwierdzić, że model M2 ze wzmocnionym ograniczeniem (5.8) stanowi model M1, gdzie zmienne  $s_{ik}$  zostały zastąpione zmiennymi  $z_{ik}$ .

## 5.4. Odchyleniowy model PL

Operator porządkujący  $\Theta$  używany w agregacji OWA jest nieliniowy i w ogólności prowadzi do złożonych modeli obliczeniowych. Jednak w szczególnym przypadku, gdy wagi są monotoniczne, agregacja OWA jest odcinkami liniową funkcją wypukłą, której minimalizacja może być wyrażona w postaci programowania liniowego [67]. Wynikiem jest tzw. model „odchyleniowy”, który wykorzystuje liniową reprezentację skumulowanego uporządkowanego wektora ocen  $\bar{\Theta}_k(\mathbf{y}) = (\bar{\theta}_1(\mathbf{y}), \bar{\theta}_2(\mathbf{y}), \dots, \bar{\theta}_m(\mathbf{y}))$ .

Na początku należy zauważyć, że poszczególne składowe skumulowanego uporządkowanego wektora ocen  $\bar{\theta}_k(\mathbf{y}) = \sum_{i=1}^k \theta_i(\mathbf{y})$  dla  $k = 1, \dots, m$  — reprezentujące odpowiednio najgorszą (największą) ocenę, sumę dwóch najgorszych ocen, sumę trzech najgorszych ocen itd. — mogą być wyrażone w następującej postaci

$$\bar{\theta}_k(\mathbf{y}) = k\theta_k(\mathbf{y}) + \sum_{i=1}^{k-1} (\theta_i(\mathbf{y}) - \theta_k(\mathbf{y})) \quad \text{dla } k = 1, \dots, m.$$

W tym sformułowaniu  $k$ -ta największa ocena  $\theta_k(\mathbf{y})$  stanowi pewnego rodzaju wartość bazową, do której dodawane są różnice (odchylenia) w stosunku do ocen od niej większych. Teraz wychodząc od zadania optymalizacyjnego (5.2) na  $k$ -tą największą ocenę  $\theta_k(\mathbf{y})$ , można je przekształcić w taki sposób, aby wyznaczało sumę  $k$  największych ocen  $\bar{\theta}_k(\mathbf{y})$ . W tym celu wprowadza się zmienne jawnie reprezentujące nieujemne odchylenia ocen od wartości bazowej. Dla dowolnego  $\mathbf{y} \in A$  oraz  $k = 1, \dots, m$  można sformułować zadanie:

$$\bar{\theta}_k(\mathbf{y}) = \min_{\hat{y}_k, d_{ik}, z_{ik}} \left( k\hat{y}_k + \sum_{i=1}^m d_{ik} \right) \quad (5.12a)$$

$$\text{p.o. } \hat{y}_k + d_{ik} \geq y_i \quad \text{dla } i = 1, \dots, m, \quad (5.12b)$$

$$d_{ik} \leq Mz_{ik} \quad \text{dla } i = 1, \dots, m, \quad (5.12c)$$

$$\sum_{i=1}^m z_{ik} \leq k - 1, \quad (5.12d)$$

$$d_{ik} \geq 0 \quad \text{dla } i = 1, \dots, m, \quad (5.12e)$$

$$z_{ik} \in \{0, 1\} \quad \text{dla } i = 1, \dots, m. \quad (5.12f)$$

Zgodnie z ograniczeniami (5.12c) i (5.12d) dla danego  $k$  co najwyżej  $k - 1$  zmiennych  $d_{ik}$  ( $i = 1, \dots, m$ ) jest niezerowa. Stąd co najwyżej  $k - 1$  ograniczeń (5.12b) jest zrelaksowanych w tym sensie, że  $\hat{y}_k$  jest mniejsza od co najwyżej  $k - 1$  ocen  $y_i$  ( $i = 1, \dots, m$ ). Tym samym, ponieważ funkcja celu podlega minimalizacji,  $\hat{y}_k$  jest faktycznie  $k$ -tą największą oceną. Z kolei zmienne  $d_{ik}$  wyrażają różnice pomiędzy ocenami  $y_i$  ( $i = 1, \dots, m$ ) i oceną  $\hat{y}_k$ , o ile  $y_i$  jest większa od  $\hat{y}_k$  (gdy  $y_i$  jest nie większa od  $k$ -tej największej wartości, wówczas  $d_{ik} = 0$ ).

Okazuje się, że zmienne binarne w zadaniu (5.12) mogą być pominięte, co prowadzi w konsekwencji do znacznie łatwiejszego zadania. Zmienia się przy tym charakter części zmiennych, co znajduje odzwierciedlenie w notacji, gdzie zmienne  $\hat{y}_k$  są zastępowane przez zmienne  $t_k$  dla  $k = 1, \dots, m$  — szczegółowy charakter zmian jest wyjaśniony poniżej. Jak pokazano w [67], każda z wielkości  $\bar{\theta}_k(\mathbf{y})$  dla dowolnego  $\mathbf{y} \in A$  oraz  $k = 1, \dots, m$  może być wyznaczona jako wartość optymalna następującego zadania PL:

$$\bar{\theta}_k(\mathbf{y}) = \min_{t_k, d_{ik}} \left( kt_k + \sum_{i=1}^m d_{ik} \right) \quad (5.13a)$$

$$\text{p.o. } d_{ik} \geq y_i - t_k \quad \text{dla } i = 1, \dots, m, \quad (5.13b)$$

$$d_{ik} \geq 0 \quad \text{dla } i = 1, \dots, m. \quad (5.13c)$$

Takie uproszczenie jest możliwe, gdyż dla dowolnego  $k = 1, \dots, m$  każde rozwiązanie dopuszczalne zadania (5.13), w którym więcej niż  $k - 1$  zmiennych  $d_{ik}$  jest większych od 0, można zastąpić rozwiązaniem dopuszczalnym z nie gorszą wartością funkcji celu z co najwyżej  $k - 1$  zmiennymi  $d_{ik}$  większymi niż 0. Tym samym dla każdego rozwiązania optymalnego zadania (5.13) istnieje odpowiadające mu rozwiązanie optymalne zadania (5.12). Szczegółowy dowód znajduje się w [67]. Można również zauważyć, że zadanie (5.13) jest równoważne zadaniu (3.3) minimalizacji średniej  $k$  największych wartości. Wystarczy podzielić wartość funkcji celu (5.13a) przez  $k$ , aby otrzymać wartość średnią  $k$  największych kosztów obsługi (3.3a). Analogicznie zadanie (5.12) jest równoważne zadaniu (3.2).

Na podstawie dowodu z [67] można stwierdzić, że gdy w rozwiązaniu optymalnym zadania (5.13) dla dowolnego  $k = 1, \dots, m$  co najwyżej  $k - 1$  zmiennych  $d_{ik}$  jest większych od 0, to zmienna  $t_k$  stanowi  $k$ -tą największą ocenę. Istnieją jednak rozwiązania optymalne, gdzie zmienna  $t_k$  nie spełnia tego warunku (stąd zastąpienie zmiennych  $\hat{y}_k$  przez zmienne  $t_k$ ). W tym momencie warto dokładniej przeanalizować zadanie (5.13), co pozwoli na precyzyjniejsze zaprezentowanie struktury zbioru rozwiązań optymalnych. Z zadania (5.13) wynika, że zmienne  $d_{ik} = \max(y_i - t_k, 0) = (y_i - t_k)_+$ . Funkcja celu może być więc sformułowana w postaci  $g(t_k) = kt_k + \sum_{i=1}^m (y_i - t_k)_+$ . Można teraz pokazać następującą własność rozwiązań optymalnych.

**Lemat 5.1.** Dla dowolnego  $\mathbf{y} \in A$  oraz  $k = 1, \dots, m$  każde rozwiązanie optymalne zadania



(5.13) spełnia warunek

$$\begin{aligned} \theta_{k+1}(\mathbf{y}) &\leq t_k^* \leq \theta_k(\mathbf{y}) & \text{dla } k = 1, \dots, m-1, \\ t_m^* &\leq \theta_m(\mathbf{y}). \end{aligned} \quad (5.14)$$

*Dowód.* Poniżej zostanie pokazane, że dla dowolnego  $\mathbf{y} \in A$  wartość funkcji  $g(t_k) = kt_k + \sum_{i=1}^m (y_i - t_k)_+$  jest stała i równa  $\bar{\theta}_k(\mathbf{y})$ , gdy  $\theta_{k+1}(\mathbf{y}) \leq t_k \leq \theta_k(\mathbf{y})$  dla  $k = 1, \dots, m-1$  lub gdy  $t_k \leq \theta_k(\mathbf{y})$  dla  $k = m$ .

Jak wspomniano wcześniej zadanie (5.13) posiada rozwiązanie optymalne, gdzie  $t_k^*$  jest  $k$ -tą największą oceną, czyli  $t_k^* = \theta_k(\mathbf{y})$  i wartość funkcji celu to  $\bar{\theta}_k(\mathbf{y})$ . Tym samym  $g(\theta_k(\mathbf{y})) = \bar{\theta}_k(\mathbf{y})$ . Pochodna lewostronna funkcji  $g(t_k) = kt_k + \sum_{i=1}^m (y_i - t_k)_+$  w punkcie  $\theta_k(\mathbf{y})$  wynosi

$$g'_-(\theta_k(\mathbf{y})) = k + \sum_{i=1}^k (-1) = 0.$$

Podobnie pochodna  $g'(t_k) = 0$  dla  $t_k \in (\theta_{k+1}(\mathbf{y}), \theta_k(\mathbf{y}))$  oraz pochodna prawostronna  $g'_+(\theta_{k+1}(\mathbf{y})) = 0$  dla  $k = 1, \dots, m-1$ . Gdy  $k = m$ , wówczas  $g'(t_m) = 0$  dla  $t_m \in (-\infty, \theta_m(\mathbf{y}))$ . Funkcja  $g(t_k)$  jest przedziałami liniową funkcją ciągłą. Ciągłość i zerowa pochodna oznacza, że wartość funkcji  $g(t_k)$  jest stała, gdy  $t_k \in [\theta_{k+1}(\mathbf{y}), \theta_k(\mathbf{y})]$  dla  $k = 1, \dots, m-1$  lub gdy  $t_k \in (-\infty, \theta_k(\mathbf{y})]$  dla  $k = m$ . Ponieważ  $g(\theta_k(\mathbf{y})) = \bar{\theta}_k(\mathbf{y})$  i  $\theta_k(\mathbf{y})$  jest prawym końcem rozpatrywanego przedziału, to wartość funkcji  $g(t_k)$  w całym tym przedziale wynosi  $\bar{\theta}_k(\mathbf{y})$  — jest to oczywiście wartość optymalna. Tym samym nierówność (5.14) jest prawdziwa dla każdego rozwiązania optymalnego zadania (5.13).  $\square$

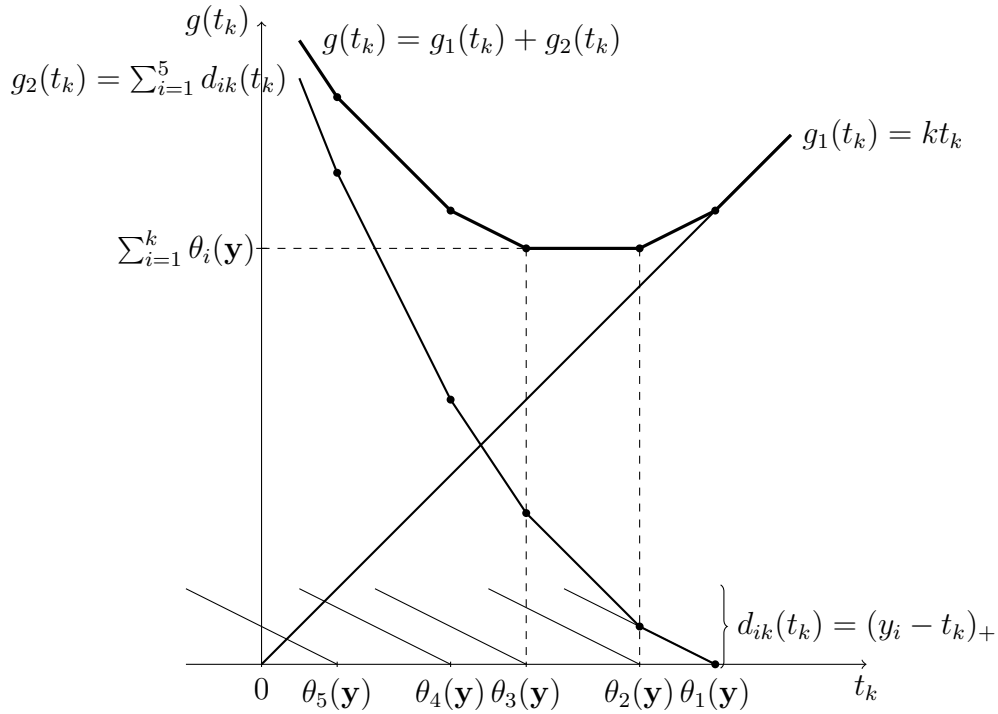
**Przykład 5.1.** Dla zobrazowania lematu 5.1 w ramach przykładu przedstawiony jest wykres funkcji  $g(t_k) = kt_k + \sum_{i=1}^m (y_i - t_k)_+$  i jej poszczególnych funkcji składowych. Niech  $\mathbf{y} \in A$  będzie wektorem ocen odpowiadającym kosztom obsługi 5 lokalizacji, a  $k = 2$ . Wówczas przy minimalizacji funkcja  $g(t_2)$  osiąga wartość optymalną, równą  $\theta_1(\mathbf{y}) + \theta_2(\mathbf{y})$ , dla dowolnego  $t_2 \in [\theta_3(\mathbf{y}), \theta_2(\mathbf{y})]$ . Ilustruje to wykres funkcji na rysunku 5.1. Dla pozostałych  $k$  wykresy funkcji są bardzo zbliżone.

Uporządkowany wektor ocen może być wyrażony jako różnice  $\theta_k(\mathbf{y}) = \bar{\theta}_k(\mathbf{y}) - \bar{\theta}_{k-1}(\mathbf{y})$  dla  $k = 2, \dots, m$  i  $\theta_1(\mathbf{y}) = \bar{\theta}_1(\mathbf{y})$ . Stąd, jak pokazano w rozdziale 4, zadanie z agregacją OWA (4.3) z wagami  $w_k$  może być zastąpione zadaniem (4.4) ze skumulowanym uporządkowanym wektorem ocen oraz zmodyfikowanymi wagami  $w'_m = w_m$  i  $w'_k = w_k - w_{k+1}$  dla  $k = 1, 2, \dots, m-1$ . Zakładając, że wagi  $w_k$  są monotoniczne i nieujemne, czyli spełniają warunek  $w_1 \geq w_2 \geq \dots \geq w_m \geq 0$ , zadanie optymalizacji OWA może być sformułowane w postaci:

$$\min_{t_k, d_{ik}, y_i} \sum_{k=1}^m w'_k (kt_k + \sum_{i=1}^m d_{ik}) \quad (5.15a)$$

$$\text{p.o. } d_{ik} \geq y_i - t_k \quad \text{dla } i, k = 1, \dots, m, \quad (5.15b)$$

$$d_{ik} \geq 0 \quad \text{dla } i, k = 1, \dots, m, \quad (5.15c)$$



Rysunek 5.1. Przykładowy wykres funkcji  $g(t_k) = kt_k + \sum_{i=1}^m (y_i - t_k)_+$  i jej funkcji składowych dla pięcioelementowego wektora ocen  $\mathbf{y} \in A$  przy  $k = 2$

$$\mathbf{y} \in A. \quad (5.15d)$$

Zadanie to zawiera tylko  $m^2$  dodatkowych ograniczeń liniowych w stosunku do podstawowego problemu lokalizacyjnego. Jeśli dodatkowo wagi  $w_k$  dla  $k = 1, 2, \dots, m$  są ściśle monotoniczne i dodatnie, to rozwiązania optymalne zadania (5.15) stanowią wyrównująco efektywne rozwiązania problemu lokalizacyjnego.

Sprawdzono wpływ ograniczeń nadmiarowych na czasy rozwiązania uzyskiwane przez model (5.15). Rozpatrzone zostały 4 dodatkowe ograniczenia, których nadmiarowy charakter pokazuje poniższe twierdzenie.

**Twierdzenie 5.2.** Zadanie (5.15) posiada rozwiązanie optymalne, które spełnia poniższe ograniczenia:

(i) nieujemności zmiennych  $t_k$

$$t_k \geq 0 \quad \text{dla} \quad k = 1, \dots, m, \quad (5.16)$$

(ii) nierosnącego uporządkowania zmiennych  $t_k$

$$t_k \geq t_{k+1} \quad \text{dla} \quad k = 1, \dots, m - 1, \quad (5.17)$$

(iii) niemalejącego uporządkowania zmiennych odchyłeń  $d_{ik}$

$$d_{ik} \leq d_{ik+1} \quad \text{dla } i = 1, \dots, m, k = 1, \dots, m-1, \quad (5.18)$$

(iv) niemalejącego uporządkowania sumy zmiennych odchyłeń  $d_{ik}$  od poszczególnych wartości bazowych

$$\sum_{i=1}^m d_{ik} \leq \sum_{i=1}^m d_{ik+1} \quad \text{dla } k = 1, \dots, m-1. \quad (5.19)$$

*Dowód.* W ramach dowodu rozpatrzone zostanie szczególne rozwiązanie optymalne zadania (5.15), które spełnia wszystkie rozważane ograniczenia.

Zadanie (5.15) posiada rozwiązanie optymalne, w którym zmienne  $t_k^*$  odpowiadają  $k$ -tym największym składowym wektora ocen  $\mathbf{y}^* \in A$ . Poszczególne oceny wyrażają koszty obsługi poszczególnych lokalizacji i koszty te są nieujemne. Czyli  $t_k^* = \theta_k(\mathbf{y}^*) \geq 0$  dla  $k = 1, \dots, m$ , a więc rozwiązanie optymalne spełnia ograniczenie (5.16). Z kolei spełnienie ograniczenia (5.17) wynika wprost z definicji  $k$ -tej największej oceny. Dalej z ograniczenia (5.17) wynika  $y_i^* - t_k^* \leq y_i^* - t_{k+1}^*$  dla dowolnego  $i, k = 1, \dots, m$ . Pamiętając, że zmienne odchyłeń można wyrazić jako  $d_{ik} = \max(y_i - t_k, 0)$  dla dowolnego  $i, k = 1, \dots, m$ , prawdziwa jest zależność  $d_{ik}^* = \max(y_i^* - t_k^*, 0) \leq \max(y_i^* - t_{k+1}^*, 0) = d_{ik+1}^*$ . Tym samym spełnione jest ograniczenie (5.18). Bezpośrednio ze spełnienia ograniczenia (5.18) wynika, że badane rozwiązanie optymalne spełnia również ograniczenie (5.19), które stanowi relaksację ograniczenia (5.18).

Ostatecznie zadanie (5.15) posiada rozwiązanie optymalne, które spełnia ograniczenia (i)–(iv).  $\square$

Po dodaniu dowolnego ograniczenia (5.16)–(5.19) do zadania (5.15), jego zbiór rozwiązań dopuszczalnych określony przez wszystkie zmienne decyzyjne (w tym  $t_k$  i  $d_{ik}$  dla  $i, k = 1, \dots, m$ ) będzie stanowił podzbiór pierwotnego zbioru dopuszczalnego. Warto jednak zauważyć, że zmienne  $t_k$  i  $d_{ik}$  ( $i, k = 1, \dots, m$ ) stanowią zmienne pomocnicze w odniesieniu do całego zadania lokalizacyjnego, które składa się z modelu rzeczowego i modelu preferencji. Eliminując z rozważań te zmienne pomocnicze (odnosząc się do modelu rzeczowego), z twierdzenia 5.2 wynika, że zbiór osiągalnych wektorów ocen  $\mathbf{y} \in A$ , a tym samym zbiór decyzji dopuszczalnych  $\mathbf{x} \in Q$ , nie zależy od obecności ograniczeń (5.16)–(5.19). Analiza obliczeniowa została przeprowadzona dla pięciu następujących sformułowań:

- MLP<sub>1</sub> — sformułowanie (5.15) bez nadmiarowych ograniczeń,
- MLP<sub>2</sub> — sformułowanie (5.15) z nadmiarowym ograniczeniem (5.16),
- MLP<sub>3</sub> — sformułowanie (5.15) z nadmiarowym ograniczeniem (5.17),
- MLP<sub>4</sub> — sformułowanie (5.15) z nadmiarowym ograniczeniem (5.18),
- MLP<sub>5</sub> — sformułowanie (5.15) z nadmiarowym ograniczeniem (5.19).

## 5.5. Hybrydowy model PCLM

Zadanie (5.15) opiera się na minimalizacji górnych ograniczeń funkcji  $kt_k + \sum_{i=1}^m d_{ik}$  dla każdego  $k = 1, 2, \dots, m$ . Ograniczenia te są przeskalowywane w funkcji celu (5.15a) przez wagi  $w'_k$ . Gdy wagi preferencji  $w_k$  nie spełniają warunku monotoniczności, część wag  $w'_k$  dla  $k = 1, 2, \dots, m$  jest ujemna. Wówczas zadanie (5.15) staje się nieograniczone i nie może być wykorzystane do optymalizacji OWA. Aby zadanie optymalizacji pozwalało na ujemne wagi  $w'_k$  ( $k = 1, 2, \dots, m$ ), konieczne jest użycie ograniczenia dolnego na funkcję  $kt_k + \sum_{i=1}^m d_{ik}$ .

### 5.5.1. Model PCLM M3

W zadaniu (5.13) dla danego  $\mathbf{y} \in A$  oraz  $k = 1, \dots, m$  minimalizacji podlega górne ograniczenie funkcji, co można jawnie wyrazić w postaci:

$$\begin{aligned} \bar{\theta}_k(\mathbf{y}) &= \min_{\varrho_k, t_k, d_{ik}} \varrho_k \\ \text{p.o.} \quad & kt_k + \sum_{i=1}^m d_{ik} \leq \varrho_k, \\ & d_{ik} \geq y_i - t_k \quad \text{dla } i = 1, \dots, m, \\ & d_{ik} \geq 0 \quad \text{dla } i = 1, \dots, m. \end{aligned}$$

Sumę  $k$ -największych wartości  $\bar{\theta}_k(\mathbf{y})$  można podobnie wyznaczyć poprzez maksymalizację dolnego ograniczenia, przy czym wymagane jest do tego użycie zmiennych binarnych. Dla dowolnego  $\mathbf{y} \in A$  oraz  $k = 1, \dots, m$  zadanie takie ma postać:

$$\bar{\theta}_k(\mathbf{y}) = \max_{\varrho_k, t'_k, d'_{ik}, z_{ik}} \varrho_k \quad (5.20a)$$

$$\text{p.o.} \quad \varrho_k \leq kt'_k + \sum_{i=1}^m d'_{ik}, \quad (5.20b)$$

$$t'_k + d'_{ik} \leq y_i + M(1 - z_{ik}) \quad \text{dla } i = 1, \dots, m, \quad (5.20c)$$

$$d'_{ik} \leq Mz_{ik} \quad \text{dla } i = 1, \dots, m, \quad (5.20d)$$

$$\sum_{i=1}^m z_{ik} = k, \quad (5.20e)$$

$$z_{ik} \in \{0, 1\} \quad \text{dla } i = 1, \dots, m. \quad (5.20f)$$

Jak widać, dla dowolnego  $k = 1, \dots, m$  używanych jest  $m$  zmiennych binarnych  $z_{ik}$  do wyznaczenie sumy  $k$  największych wartości. Zmienne te decydują, które z ograniczeń (5.20c) są relaksowane poprzez dodanie dużej stałej  $M$ , a także które zmienne  $d'_{ik}$  są niezerowe (zgodnie z ograniczeniem (5.20d)). Jeżeli zmienna  $z_{ik} = 1$ , wówczas dla odpowiadającego indeksu  $i$  ograniczenie (5.20c) staje się aktywne, a zmienna  $d'_{ik}$  może być większa od 0. Tym samym  $t'_k + d'_{ik}$  staje się równe  $y_i$  (zgodnie z maksymalizacją). Zadanie sprowadza się więc do wyboru  $k$  zmiennych  $z_{ik}$  ( $i = 1, \dots, m$ ), które przyjmą wartość 1, tak aby odpowiednie sumy  $t'_k + d'_{ik}$

były jak największe. Z tego wynika, że wartość 1 przyjmują te zmienne  $z_{ik}$ , które odpowiadają  $k$  największym ocenom  $y_i$ . Z kolei  $k$  aktywnych ograniczeń (5.20c) sprawia, że zmienna  $t'_k$  jest nie większa niż  $k$ -ta największa wartość z ocen  $y_i$ , a zmienne  $d'_{ik}$  dopełniają zmienną  $t'_k$  do odpowiadających  $k$  największych wartości spośród ocen  $y_i$ .

Następnie można pokazać, że dla dowolnego  $\mathbf{y} \in A$  oraz  $k = 1, \dots, m$  zadanie (5.20) można nieco uprościć, a mianowicie ograniczenie (5.20c) można zastąpić analogicznym ograniczeniem bez składnika ze zmiennymi binarnymi. Zmienia się przez to charakter zmiennych  $t'_k$ , ale zadanie nadal poprawnie wyznacza sumę  $k$  największych ocen. Pokazuje to poniższe twierdzenie.

**Twierdzenie 5.3.** Dla dowolnego  $k = 1, \dots, m$  sumę  $k$  największych ocen wektora  $\mathbf{y} \in A$  można wyznaczyć za pomocą następującego zadania maksymalizacji:

$$\bar{\theta}_k(\mathbf{y}) = \max_{q_k, t'_k, d'_{ik}, z_{ik}} q_k \quad (5.21a)$$

$$\text{p.o. } q_k \leq kt'_k + \sum_{i=1}^m d'_{ik}, \quad (5.21b)$$

$$t'_k + d'_{ik} \leq y_i \quad \text{dla } i = 1, \dots, m, \quad (5.21c)$$

$$d'_{ik} \leq Mz_{ik} \quad \text{dla } i = 1, \dots, m, \quad (5.21d)$$

$$\sum_{i=1}^m z_{ik} = k, \quad (5.21e)$$

$$z_{ik} \in \{0, 1\} \quad \text{dla } i = 1, \dots, m. \quad (5.21f)$$

*Dowód.* W celu uzasadnienia twierdzenia zostanie pokazane, że wartość optymalna zadania (5.21) jest taka sama jak zadania (5.20). Na początku należy zauważyć, że zadanie (5.21) stanowi restrykcję zadania (5.20). Każde rozwiązanie dopuszczalne zadania (5.21) jest również rozwiązaniem dopuszczalnym zadania (5.20). Niech  $I_k^1 = \{i: z_{ik} = 1\}$ , co oznacza podzbiór indeksów  $i$ , dla których zmienne  $z_{ik}$  są równe 1 dla dowolnego rozwiązania dopuszczalnego. Z kolei  $I_k^0 = I \setminus I_k^1$ . Dla  $i \in I_k^1$  ograniczenia (5.20c) są tożsame z ograniczeniami (5.21c). Stąd dowolne rozwiązanie dopuszczalne zadania (5.20) spełnia ograniczenia (5.21c) dla  $i \in I_k^1$ . Jeśli rozwiązanie dopuszczalne zadania (5.20) spełnia dodatkowo ograniczenia (5.21c) dla  $i \in I_k^0$ , to jest również rozwiązaniem dopuszczalnym zadania (5.21). W przeciwnym przypadku istnieje rozwiązanie dopuszczalne zadania (5.20), w którym dla pewnej liczby  $s > 0$  indeksów  $i \in I_k^0$  ( $s \leq m - k$ ) zachodzi nierówność  $t'_k + d'_{ik} > y_i$ . Ponieważ  $d'_{ik} \leq 0$  dla  $i \in I_k^0$  (zgodnie z ograniczeniem (5.20d)) to  $t'_k > y_i$  dla  $s$  indeksów  $i \in I_k^0$ . Każde takie rozwiązanie można jednak zastąpić alternatywnym rozwiązaniem o nie gorszej wartości funkcji celu, które narusza tylko  $s - 1$  ograniczeń (5.21c) dla  $i \in I_k^0$ . Mianowicie można wyznaczyć

$$\delta = \min_{\substack{i \in I_k^0, \\ y_i < t'_k}} (t'_k - y_i) \quad (5.22)$$

i podstawić zmienne  $\check{t}'_k = t'_k - \delta$  oraz  $\check{d}'_{ik} = d'_{ik} + \delta$  dla tych  $i$ , dla których  $\check{d}'_{ik}$  spełnia ograniczenia (5.20d). Biorąc pod uwagę, że oceny  $y_i$  ( $i = 1, \dots, m$ ) są nieujemne, z (5.22) wynika  $\delta \leq t'_k$ . Z tego oraz ze spełnienia ograniczenia (5.20c) dla  $i \in I_k^1$  można stwierdzić, że

$$\check{d}'_{ik} = d'_{ik} + \delta \leq y_i - t'_k + \delta \leq y_i \leq M \quad \text{dla } i \in I_k^1. \quad (5.23)$$

Z (5.23) wynika, że  $\check{d}'_{ik}$  spełnia ograniczenie (5.20d) co najmniej dla  $i \in I_k^1$ . Tym samym zmienne  $\check{d}'_{ik}$  można podstawić dla  $r \geq k$  indeksów  $i$ . W ten sposób otrzymuje się

$$k\check{t}'_k + \sum_{i=1}^m \check{d}'_{ik} = kt'_k + \sum_{i=1}^m d'_{ik} + (r - k)\delta \geq kt'_k + \sum_{i=1}^m d'_{ik}.$$

To oznacza, że dla każdego rozwiązania dopuszczalnego zadania (5.20), zmniejszając stopniowo liczbę naruszonych ograniczeń (5.21c) dla  $i \in I_k^0$ , można znaleźć odpowiadające rozwiązanie dopuszczalne zadania (5.21) z nie gorszą wartością funkcji celu. Podsumowując, zadanie (5.21), podobnie jak zadanie (5.20), wyznacza sumę  $k$  największych ocen.  $\square$

Z twierdzenia (5.3) wynika również, że jednym z rozwiązań optymalnych zadania (5.21) dla dowolnego  $k = 1, \dots, m$  jest rozwiązanie, w którym  $t_k^* = \theta_m(\mathbf{y})$ , a zmienne  $d_{ik}^* = y_i - \theta_m(\mathbf{y})$  dla  $k$  indeksów  $i$  odpowiadających największym ocenom  $y_i$  oraz  $d_{ik}^* = 0$  dla pozostałych  $i$ . W ogólności prawdziwe jest następujące spostrzeżenie.

**Lemat 5.2.** Dla dowolnego  $\mathbf{y} \in A$ ,  $k = 1, \dots, m$  i dowolnej wartości  $\zeta \in [0, \theta_m(\mathbf{y})]$  zadanie (5.21) posiada rozwiązanie optymalne, w którym  $t_k^* = \zeta$ .

*Dowód.* Jak zauważono powyżej, jednym z rozwiązań optymalnych zadania (5.21) jest rozwiązanie, gdzie  $t_k^* = \theta_m(\mathbf{y})$ . Poniżej zostanie pokazane, że dla dowolnego  $k = 1, \dots, m$  wartość funkcji celu jest stała dla  $t'_k \in [0, \theta_m(\mathbf{y})]$ .

Funkcję celu zadania (5.21) można wyrazić w postaci

$$g(t'_k) = kt'_k + \sum_{i \in I_k^1} \min(y_i - t'_k, M) + \sum_{i \in I_k^0} \min(y_i - t'_k, 0), \quad (5.24)$$

gdzie  $I_k^1 = \{i: z_{ik} = 1\}$ , a  $I_k^0 = I \setminus I_k^1$ . Oczywiście  $|I_k^1| = k$ , a  $|I_k^0| = m - k$ . Dla dowolnego  $t'_k \in [0, \theta_m(\mathbf{y})]$  funkcję (5.24) można uprościć do postaci

$$g(t'_k) = kt'_k + \sum_{i \in I_k^1} (y_i - t'_k). \quad (5.25)$$

W rozpatrywanym przedziale pochodna funkcji  $g(t'_k)$  wynosi

$$g'(t'_k) = k + \sum_{i \in I_k^1} (-1) = 0.$$

Z tego wynika, że wartość funkcji  $g(t'_k)$  jest stała na przedziale  $[0, \theta_m(\mathbf{y})]$ .

Podsumowując, dla dowolnego  $\mathbf{y} \in A$ ,  $k = 1, \dots, m$  i dowolnej wartości  $\zeta \in [0, \theta_m(\mathbf{y})]$  zadanie (5.21) posiada rozwiązanie optymalne z  $t_k^* = \zeta$ .  $\square$

Wykorzystując zadanie (5.21), można teraz sformułować model OWA, który bazuje na funkcji  $kt_k + \sum_{i=1}^m d_{ik}$  i pozwala na ujemne wagi  $w'_k$  (niemonotoniczne wagi  $w_k$ ):

$$\min_{\varrho_k, t_k, d_{ik}, t'_k, d'_{ik}, z_{ik}, y_i} \sum_{k=1}^m w'_k \varrho_k \quad (5.26a)$$

$$\text{p.o.} \quad kt_k + \sum_{i=1}^m d_{ik} \leq \varrho_k \quad \text{dla } k = 1, \dots, m, \quad (5.26b)$$

$$t_k + d_{ik} \geq y_i, \quad d_{ik} \geq 0 \quad \text{dla } i, k = 1, \dots, m, \quad (5.26c)$$

$$\varrho_k \leq kt'_k + \sum_{i=1}^m d'_{ik} \quad \text{dla } k = 1, \dots, m, \quad (5.26d)$$

$$t'_k + d'_{ik} \leq y_i \quad \text{dla } i, k = 1, \dots, m, \quad (5.26e)$$

$$d'_{ik} \leq Mz_{ik} \quad \text{dla } i, k = 1, \dots, m, \quad (5.26f)$$

$$\sum_{i=1}^m z_{ik} = k \quad \text{dla } k = 1, \dots, m, \quad (5.26g)$$

$$z_{ik} \in \{0, 1\} \quad \text{dla } i, k = 1, \dots, m, \quad (5.26h)$$

$$\mathbf{y} \in A. \quad (5.26i)$$

Nie ma jednak potrzeby jednoczesnego używania zarówno ograniczenia dolnego, jak i górnego dla każdego  $k = 1, \dots, m$ . Można pominąć wyznaczenie ograniczenia górnego dla tych  $k$ , gdzie  $w'_k < 0$  oraz wyznaczenie ograniczenia dolnego dla tych  $k$ , gdzie  $w'_k \geq 0$ . Tym samym w zadaniu (5.26) ograniczenia (5.26b), (5.26c) mogą zostać pominięte dla tych  $k$ , dla których  $w'_k < 0$ . Co ważniejsze ograniczenia (5.26d)–(5.26h) mogą zostać pominięte dla tych  $k$ , dla których  $w'_k \geq 0$ . To zmniejsza również zdecydowanie liczbę zmiennych decyzyjnych, gdyż zmienne  $t_k, d_{ik}$  będą określone tylko dla tych  $k$ , gdzie  $w'_k \geq 0$ , a zmienne  $t'_k, d'_{ik}$  i  $z_{ik}$  dla tych  $k$ , gdzie  $w'_k < 0$ . Liczba zmiennych ciągłych będzie więc taka sama jak w modelu (5.15). Uwzględniając powyższe obserwacje, zadanie przedstawia się następująco:

$$\min_{\varrho_k, t_k, d_{ik}, t'_k, d'_{ik}, z_{ik}, y_i} \sum_{k=1}^m w'_k \varrho_k \quad (5.27a)$$

$$\text{p.o.} \quad kt_k + \sum_{i=1}^m d_{ik} \leq \varrho_k \quad \text{dla } k = 1, \dots, m; w'_k \geq 0, \quad (5.27b)$$

$$t_k + d_{ik} \geq y_i, \quad d_{ik} \geq 0 \quad \text{dla } i, k = 1, \dots, m; w'_k \geq 0, \quad (5.27c)$$

$$\varrho_k \leq kt'_k + \sum_{i=1}^m d'_{ik} \quad \text{dla } k = 1, \dots, m; w'_k < 0, \quad (5.27d)$$

$$t'_k + d'_{ik} \leq y_i \quad \text{dla } i, k = 1, \dots, m; w'_k < 0, \quad (5.27e)$$

$$d'_{ik} \leq Mz_{ik} \quad \text{dla } i, k = 1, \dots, m; w'_k < 0, \quad (5.27f)$$

$$\sum_{i=1}^m z_{ik} = k \quad \text{dla } k = 1, \dots, m; w'_k < 0, \quad (5.27g)$$

$$z_{ik} \in \{0, 1\} \quad \text{dla } i, k = 1, \dots, m; w'_k < 0, \quad (5.27h)$$

$$\mathbf{y} \in A. \quad (5.27i)$$

Wyraźnie widać teraz podział zadania na część liniową z ograniczeniami (5.27b)–(5.27c) oraz część całkowitoliczbową liniową mieszaną z ograniczeniami (5.27d)–(5.27h). Model ten można więc określić mianem hybrydowego. Część liniowa, określona dla nieujemnych wag  $w'_k$ , stanowi w rzeczywistości model liniowy OWA (5.15). Druga część tego zadania, ze względu na zmienne binarne, jest oczywiście znacznie kosztowniejsza obliczeniowo. Warto jednak zauważyć, że liczba zmiennych binarnych jest proporcjonalna do liczby ujemnych wag  $w'_k$ . Definiując ten zbiór jako  $K^- = \{k: w'_k < 0, k = 1, \dots, m\}$ , liczba zmiennych binarnych jest równa  $|K^-|m$ . Uwzględniając, że wagi preferencji  $w_k \geq 0$ , a wagi  $w'_k$  są zdefiniowane jako  $w'_m = w_m, w'_k = w_k - w_{k+1}$ , liczebność zbioru  $K^-$  może wynosić co najwyżej  $m - 1$ . Model ten może być więc alternatywą dla innych modeli PCLM agregacji OWA, gdzie liczba zmiennych binarnych jest rzędu  $m^2$ . Z drugiej strony zadanie to zawiera jednak  $O(m^2)$  zmiennych ciągłych, gdy inne modele PCLM wprowadzały  $O(m)$  zmiennych ciągłych. Co gorsza, większa jest także liczba ograniczeń — co prawda, ich liczba w obu przypadkach jest rzędu  $m^2$ , ale w przypadku tego modelu współczynnik proporcjonalności jest większy.

W związku z powyższym model (5.27) wydaje się interesujący zwłaszcza dla problemów z niewielką liczbą ujemnych wag  $w'_k$ . W praktyce dotyczy to np. sytuacji, gdy w funkcji celu chce się pominąć pewną część najgorszych kryteriów. Odpowiada to problemom typu TC4 (zawężonej średniej), gdzie tylko jedna waga  $w'_k$  jest mniejsza od 0.

Dla modelu rozpatrzono ograniczenia nadmiarowe, przy czym tylko takie, które dotyczą części całkowitoliczbowej zadania — ograniczenia nadmiarowe w części liniowej są badane w ramach odchyleniowego modelu liniowego OWA (5.15).

**Twierdzenie 5.4.** Zadanie (5.27) posiada rozwiązanie optymalne, które spełnia ograniczenia:

(i) nieujemności zmiennych  $d'_{ik}$

$$d'_{ik} \geq 0 \quad \text{dla } i, k = 1, \dots, m; w'_k < 0, \quad (5.28)$$

(ii) nieujemności zmiennych  $t'_k$

$$t'_k \geq 0 \quad \text{dla } k = 1, \dots, m; w'_k < 0, \quad (5.29)$$

(iii) niemalejącego uporządkowania zmiennych binarnych  $z_{ik}$  dotyczących  $i$ -tej oceny

$$z_{ik} \leq z_{ik'} \quad \text{dla } i = 1, \dots, m; k \in \{K^- \setminus \max\{K^-\}\}; k' = \text{suc}(k), \quad (5.30)$$

gdzie  $\text{suc}(k) = \min\{k': k' \in K^- \wedge k' > k\}$  to funkcja następnika w ramach zbioru  $K^-$ .



*Dowód.* Ze względu na strukturę zadanie (5.27) posiada rozwiązanie optymalne, które spełnia ograniczenia (5.28)–(5.29) wtedy i tylko wtedy, gdy istnieje rozwiązanie optymalne zadania (5.21) dla dowolnego  $k = 1, \dots, m$ , które spełnia warunki (5.28)–(5.29). W ramach dowodu rozpatrzone zostanie szczególne rozwiązanie optymalne zadania (5.21) dla dowolnego  $k = 1, \dots, m$ . Z kolei aby udowodnić spełnienie ograniczenia (5.30), przeanalizowana zostanie zależność między rozwiązaniami optymalnymi zadania (5.21) dla różnych  $k, k' = 1, \dots, m$  ( $k' > k$ ).

Z lematu (5.2) wynika, że dla dowolnego  $\mathbf{y} \in A$ ,  $k = 1, \dots, m$  oraz dowolnej wartości  $\zeta \in [0, \theta_m(\mathbf{y})]$  istnieje rozwiązanie optymalne zadania (5.21), gdzie  $t_k^* = \zeta$ . Niech w rozpatrywanym rozwiązaniu optymalnym  $t_k^* = \theta_m(\mathbf{y})$ . Wówczas dla dowolnego  $k = 1, \dots, m$

$$d_{ik}^* = \begin{cases} \min(y_i - \theta_m(\mathbf{y}), 0) = 0 & \text{dla } i \in I_k^0, \\ \min(y_i - \theta_m(\mathbf{y}), M) = y_i - \theta_m(\mathbf{y}) \geq 0 & \text{dla } i \in I_k^1. \end{cases}$$

Tym samym badane rozwiązanie optymalne spełnia ograniczenie (5.28). Ograniczenie (5.29) wynika bezpośrednio z przyjęcia  $t_k^* = \theta_m(\mathbf{y}) \geq 0$ .

Nierówność (5.30) jest analogiczna do ograniczenia (5.5) dla modelu M1. W tym przypadku należy jednak uwzględnić fakt, że zmienne binarne są określone tylko dla  $k \in K^-$ , a ograniczenie (5.30) jest określone tylko gdy  $|K^-| \geq 2$ . Z postaci maksymalizowanej funkcji celu (5.25) można wywnioskować, że dla dowolnego  $k = 1, \dots, m$  indeksy  $i \in I_k^1$  będą odpowiadały  $k$  największym ocenom wektora  $\mathbf{y}$ . Stąd jeśli  $y_i$  jest jedną z  $k$  największych ocen, wówczas  $z_{ik}^* = 1$ . Rozpatrując zadanie (5.21) dla tego samego  $\mathbf{y}$  oraz dowolnego  $k' > k$ , oczywiste jest, że  $y_i$  jest również jedną z  $k'$  największych ocen i tym samym  $z_{ik'}^* = 1$ . To oznacza, że dla tego samego  $\mathbf{y} \in A$  i dowolnych  $k, k' = 1, \dots, m$  ( $k' > k$ ) rozwiązania optymalne zadań (5.21) spełniają nierówności  $z_{ik}^* \leq z_{ik'}^*$  dla dowolnego  $i = 1, \dots, m$ . Tym samym rozwiązanie optymalne zadania (5.27), gdzie  $t_k^* = \theta_m(\mathbf{y}^*)$  dla wszystkich  $k \in K^-$ , spełnia ograniczenie (5.30).

Ostatecznie zadanie (5.27) posiada rozwiązanie optymalne, które spełnia ograniczenia (i)–(iii).  $\square$

Należy zauważyć, że tak jak w poprzednich modelach, także w zadaniu (5.27) zmienne  $t_k'$ ,  $d_{ik}'$  i  $z_{ik}$  są zmiennymi pomocniczymi dla dowolnych  $i, k = 1, \dots, m$  (oczywiście tak samo jak zmienne  $t_k$  i  $d_{ik}$  w części liniowej zadania, która odpowiada odchyleniowemu modelowi OWA). Zbiór rozwiązań dopuszczalnych zadania (5.27) określony przez zmienne decyzyjne  $\mathbf{x} \in Q$  (czy zbiór wektorów osiągalnych  $\mathbf{y} \in A$ ) jest niezależny od ograniczeń (5.28)–(5.30). W ramach testów sprawdzono następujące sformułowania:

- M3<sub>1</sub> — sformułowanie (5.27) bez ograniczeń nadmiarowych,
- M3<sub>2</sub> — sformułowanie (5.27) z jednym nadmiarowym ograniczeniem (5.28),
- M3<sub>3</sub> — sformułowanie (5.27) z jednym nadmiarowym ograniczeniem (5.29),
- M3<sub>4</sub> — sformułowanie (5.27) z jednym nadmiarowym ograniczeniem (5.30),

— M3<sub>5</sub> — sformułowanie (5.27) z dwoma nadmiarowymi ograniczeniami (5.29), (5.30).

### 5.5.2. Model PCLM M4

Zadanie (5.21) wyznacza sumę  $k$ -największych wartości poprzez maksymalizację dolnego ograniczenia funkcji  $kt'_k + \sum_{i=1}^m d'_{ik}$ . Zadanie to można nieco zmodyfikować ograniczając liczbę zmiennych i upraszczając strukturę części ograniczeń.

**Twierdzenie 5.5.** Dla dowolnego  $k = 1, \dots, m$  sumę  $k$  największych ocen wektora  $\mathbf{y} \in A$  można wyznaczyć za pomocą następującego zadania maksymalizacji:

$$\bar{\theta}_k(\mathbf{y}) = \max_{\varrho_k, y'_{ik}, z_{ik}} \varrho_k \quad (5.31a)$$

$$\text{p.o. } \varrho_k \leq \sum_{i=1}^m y'_{ik}, \quad (5.31b)$$

$$y'_{ik} \leq y_i \quad \text{dla } i = 1, \dots, m, \quad (5.31c)$$

$$y'_{ik} \leq M z_{ik} \quad \text{dla } i = 1, \dots, m, \quad (5.31d)$$

$$\sum_{i=1}^m z_{ik} = k, \quad (5.31e)$$

$$z_{ik} \in \{0, 1\} \quad \text{dla } i = 1, \dots, m. \quad (5.31f)$$

*Dowód.* Poniżej pokazane zostanie, że wartość optymalna zadania (5.31) jest taka sama jak zadania (5.21).

Na podstawie lematu 5.2 wiadomo, że dla dowolnego  $\mathbf{y} \in A$ ,  $k = 1, \dots, m$  oraz dowolnej wartości  $\zeta \in [0, \theta_m(\mathbf{y})]$  zadanie (5.21) posiada rozwiązanie optymalne z  $t_k^* = \zeta$ . Tym samym po podstawieniu za  $t'_k$  dowolnej wartości z przedziału  $[0, \theta_m(\mathbf{y})]$  zadanie (5.21) nadal wyznacza sumę  $k$  największych ocen wektora  $\mathbf{y} \in A$ . Podstawiając  $t'_k = 0$  do zadania (5.21), otrzymuje się zadanie (5.31), gdzie zmienne  $d'_{ik}$  są zastąpione przez zmienne  $y'_{ik}$ . Ta zmiana w notacji wynika ze zmiany charakteru zmiennych, które stanowiąc dodatnie odchylenia  $k$  największych ocen od wartości bazowej — ustalonej w zadaniu (5.31) na 0 — reprezentują faktycznie wartości  $k$  największych ocen.

Podsumowując, wartość optymalna zadania (5.31), podobnie jak zadania (5.21), wyznacza sumę  $k$  największych ocen wektora  $\mathbf{y} \in A$ .  $\square$

Z postaci zadania (5.31) widać teraz, że ograniczenia (5.31c) i (5.31d) dla dowolnego  $k = 1, \dots, m$  stanowią linearyzację ograniczenia

$$y'_{ik} \leq y_i z_{ik} \quad \text{dla } i = 1, \dots, m.$$

Podobnie jak zadanie (5.21) we wcześniejszym rozdziale, tak zadanie (5.31) można

zastosować do wyznaczenia sumy  $k$  największych ocen wektora  $y$  dla tych  $k$ , dla których  $w'_k < 0$ . Tym samym cały model optymalizacji OWA formułuje się jako:

$$\min_{\varrho_k, t_k, d_{ik}, y'_{ik}, z_{ik}, y_i} \sum_{k=1}^m w'_k \varrho_k \quad (5.32a)$$

$$\text{p.o.} \quad kt_k + \sum_{i=1}^m d_{ik} \leq \varrho_k \quad \text{dla } k = 1, \dots, m; w'_k \geq 0, \quad (5.32b)$$

$$t_k + d_{ik} \geq y_i, \quad d_{ik} \geq 0 \quad \text{dla } i, k = 1, \dots, m; w'_k \geq 0, \quad (5.32c)$$

$$\varrho_k \leq \sum_{i=1}^m y'_{ik} \quad \text{dla } k = 1, \dots, m; w'_k < 0, \quad (5.32d)$$

$$y'_{ik} \leq y_i \quad \text{dla } i, k = 1, \dots, m; w'_k < 0, \quad (5.32e)$$

$$y'_{ik} \leq M z_{ik} \quad \text{dla } i, k = 1, \dots, m; w'_k < 0, \quad (5.32f)$$

$$\sum_{i=1}^m z_{ik} = k \quad \text{dla } k = 1, \dots, m; w'_k < 0, \quad (5.32g)$$

$$z_{ik} \in \{0, 1\} \quad \text{dla } i, k = 1, \dots, m; w'_k < 0, \quad (5.32h)$$

$$y \in A. \quad (5.32i)$$

Zadanie to w porównaniu do zadania (5.27) ma nieco mniejszą liczbę zmiennych ciągłych — o tyle ile jest ujemnych wag  $w'_k$ . Prostsza strukturę mają także ograniczenia (5.32d) i (5.32e) w stosunku do odpowiadających im ograniczeń (5.27d) i (5.27e). Zadanie (5.32) można zatem traktować jako drugą wersję modelu hybrydowego.

Dla modelu (5.32), oznaczanego dalej jako M4, zbadano wpływ ograniczenia nadmiarowego (5.30) (dowód nadmiarowości analogiczny jak w twierdzeniu 5.4). W związku z tym w ramach testów wykorzystano dwa sformułowania:

- M4<sub>1</sub> — sformułowanie (5.32) bez ograniczeń nadmiarowych,
- M4<sub>2</sub> — sformułowanie (5.32) z jednym nadmiarowym ograniczeniem (5.30).

## 5.6. Projekt eksperymentów

W ramach testów porównano czasy osiągane przez różne sformułowania OWA w postaci zadań PCLM, a dla wag monotonicznych również w postaci zadań PL. Aby porównać wydajność obliczeniową modeli, rozwiązano za ich pomocą różne problemy lokalizacyjne i skonfrontowano czasy uzyskiwane przez poszczególne sformułowania. Ogólny schemat testów jest analogiczny do idei przedstawionej w [17].

Aby badania odbywały się na zróżnicowanych problemach, wybrano podstawowe parametry charakteryzujące problemy lokalizacyjne i określono dla nich zbiory wartości dopuszczalnych. Następnie na podstawie kombinacji wartości tych parametrów zdefiniowano instancje problemów lokalizacyjnych. Uwzględnione parametry to rozmiar problemu (liczba lokalizacji), liczba punktów obsługi do rozmieszczenia oraz typ problemu określony przez wektor wag

agregacji OWA. Podobnie jak w [66] rozważono następujące zbiory dopuszczalnych wartości poszczególnych parametrów ( $\lceil a \rceil$  oznacza najmniejszą liczbę całkowitą nie mniejszą niż  $a$ , a  $\lfloor a \rfloor$  największą liczbę całkowitą nie większą niż  $a$ ):

- rozmiar problemu — siedem wariantów:  $m = 8, 10, 12, 15, 20, 25, 30$ ,
- liczba punktów obsługi — proporcjonalna do rozmiaru problemu  $m$ , cztery warianty w ramach każdego rozmiaru:  $n = \lceil \frac{m}{4} \rceil, \lceil \frac{m}{3} \rceil, \lceil \frac{m}{2} \rceil, \lceil \frac{m}{2} + 1 \rceil$ ,
- typ problemu — Dwanaście wariantów zdefiniowanych przez odpowiednie wektory wag preferencji w (patrz tabela 5.1). Osiem pierwszych typów pojawiło się w pozycji [10]. Kryterium średniej (TC1) i kryterium centrum (TC2) to najpopularniejsze kryteria w zagadnieniach lokalizacyjnych. Kryterium  $k$ -centrów i  $(k_1 + k_2)$ -zawężonej średniej nie są tak powszechne, ale również stosowane w literaturze przedmiotu. Problemy typu TC5–TC8 można traktować jako sztuczne i są przede wszystkim używane do sprawdzenia wydajności obliczeniowej dla nierównomiernie rozłożonych wag preferencji. Ostatnie cztery typy TC9–TC12 dotyczą problemów z monotonicznymi wagami. Dla wag malejących (TC9, TC11) są to problemy wypukłe, a co za tym idzie prostsze przy minimalizacji. Natomiast dla wag rosnących (TC10, TC12) problemy są wklęsłe, przez co trudniejsze przy minimalizacji. Te ostatnie cztery typy problemów mogą być traktowane jako rozszerzenia podejścia pesymistycznego min max (TC9, TC11) i optymistycznego min min (TC10, TC12).

Dla każdej wartości rozmiaru problemu zostało wygenerowanych 15 losowych macierzy kosztów z zerami na głównej przekątnej i pozostałymi elementami wylosowanymi z jednostajnego rozkładu dyskretnego w przedziale  $[1, 100]$ . Macierze te zostały przypisane do każdej kombinacji wartości parametrów, tworząc w ten sposób zbiór problemów testowych.

Modele PCLM zostały przetestowane na mniejszych problemach (8, 10 lokalizacji). Część modeli zastosowano także dla problemów z 12 i 15 lokalizacjami, co wykorzystano m.in. do porównania z rezultatami modelu liniowego, który był badany na wszystkich wymienionych rozmiarach problemów (ale tylko dla problemów z wagami malejącymi i nierosnącymi).

Procedura eksperymentalna została zaimplementowana w języku C++, a do rozwiązywania problemów wykorzystano pakiet IBM ILOG CPLEX Optimization Studio (w tym solver cplex) w wersji 12.4 [35]. Eksperymenty obliczeniowe zostały wykonane na maszynie z procesorem Intel Core2 Duo 2.53GHz (mobilny) i 3 GB pamięci RAM. Nałożono również limit czasowy 600 sekund jako maksymalny czas rozwiązywania pojedynczej instancji problemu.

## 5.7. Wyniki

Porównanie wydajności obliczeniowej zostało wykonane w oparciu o średnie czasy rozwiązania problemów w ramach poszczególnych typów, rozmiarów i punktów obsługi, czyli uśredniając po instancjach macierzy kosztów. Pełne wyniki znajdują się w dodatku A w tabelach A.1–A.3. Dla większej przejrzystości poniższa analiza wyników prezentuje rezultaty w formie graficznej uśrednione dodatkowo po wariantach liczby punktów obsługi.

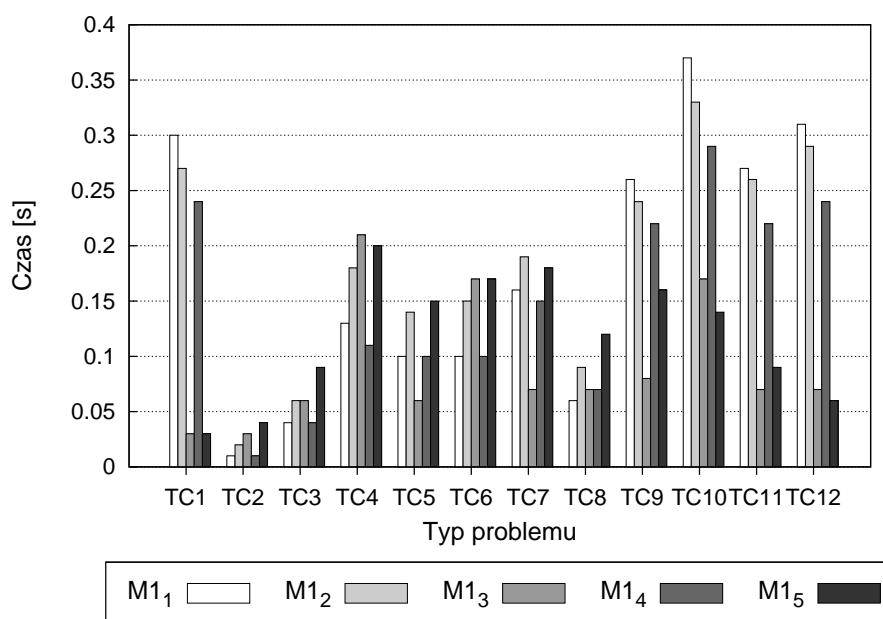
Tabela 5.1. Typy problemów i definiujące je wektory wag preferencji w zależne od liczby lokalizacji  $m$  i ewentualnie liczby punktów obsługi  $n$

typ	nazwa/opis kryterium	wektor wag $w$	parametry
TC1	średniej (ang. $N$ -median)	$(\underbrace{1, \dots, 1}_m)$	
TC2	centrum (ang. $N$ -center)	$(1, \underbrace{0, \dots, 0}_{m-1})$	
TC3	$k$ -centrów (ang. $k$ -centra)	$(\underbrace{1, \dots, 1}_k, 0, \dots, 0)$	$k = \lfloor \frac{m}{3} \rfloor$
TC4	$k_1 + k_2$ zawężonej średniej (ang. $k_1 + k_2$ -trimmed mean)	$(\underbrace{0, \dots, 0}_{k_1}, 1, \dots, 1, \underbrace{0, \dots, 0}_{k_2})$	$k_1 = \lceil \frac{m}{10} \rceil,$ $k_2 = \lceil n + \frac{m}{10} \rceil$
TC5	ciąg naprzemiennych 0 i 1, zaczynający się od 1	$(1, 0, 1, 0, 1, 0, \dots)$	
TC6	ciąg naprzemiennych 0 i 1, zaczynający się od 0	$(0, 1, 0, 1, 0, 1, \dots)$	
TC7	ciąg sekwencji (1, 1, 0)	$(1, 1, 0, 1, 1, 0, \dots)$	
TC8	ciąg sekwencji (1, 0, 0)	$(1, 0, 0, 1, 0, 0, \dots)$	
TC9	ciąg zaczynający się od wartości $m$ i malejący o 1	$(m, m - 1, \dots, 2, 1)$	
TC10	jak TC9, ale w odwrotnej kolejności (ciąg rosnący)	$(1, 2, \dots, m - 1, m)$	
TC11	ciąg zaczynający się od $3m$ i malejący przedziałami liniowo, najpierw $k$ wag o 3, następnie $k$ wag o 2 i reszta o 1	$(3m, \underbrace{3(m-1), \dots, 3(m-k)}_k,$ $\underbrace{3(m-k)-2, \dots, 3(m-k)-2k}_k,$ $3m - 5k - 1, 3m - 5k - 2, \dots)$	$k = \lfloor \frac{m}{3} \rfloor$
TC12	jak TC11, ale w odwrotnej kolejności (ciąg rosnący)	$(\dots, 3m - 5k - 2, 3m - 5k - 1,$ $\underbrace{3(m-k) - 2k, \dots, 3(m-k) - 2}_k,$ $\underbrace{3(m-k), \dots, 3(m-1), 3m}_k)$	$k = \lfloor \frac{m}{3} \rfloor$

### 5.7.1. Wyniki modeli PCLM M1 i M2

Aby zbadać wpływ ograniczeń nadmiarowych na wydajność obliczeniową zadań PCLM modeli M1 i M2, porównano sformułowania z różnymi ograniczeniami w ramach poszczególnych modeli. Szczegółowe wyniki dla modeli mieszanego programowania całkowitoliczbowego M1 i M2 można znaleźć w dodatku A.1.

Analizując wyniki pierwszego modelu (rysunek 5.2), można zauważyć, że dodanie ograniczeń nadmiarowych prowadzi do polepszenia wydajności obliczeniowej w przypadku problemów z wszystkimi wagami różnymi od 0 (TC1, TC9–TC12). Różnice dla sformułowania M1<sub>2</sub> (z ograniczeniem nadmiarowym (5.4)) w stosunku do sformułowania M1<sub>1</sub> są niewielkie i mieszczą się w granicach ok. 20%. Znacznie większa poprawa dotyczy sformułowania M1<sub>3</sub> (z ograniczeniami nadmiarowymi (5.4), (5.6)). Tutaj czas rozwiązania skrócił się kilkukrotnie, a dla kryterium średniej (TC1) ponad 10-krotnie. Sformułowanie M1<sub>4</sub> (z ograniczeniem nadmiarowym (5.5)) osiąga zbliżone wyniki do sformułowania M1<sub>2</sub>, przy czym skrócenie czasu sięga 35%. Z kolei sformułowanie M1<sub>5</sub> (z ograniczeniami nadmiarowymi (5.5), (5.6)) osiąga czasy zbliżone lub nieco gorsze niż sformułowanie M1<sub>3</sub>.



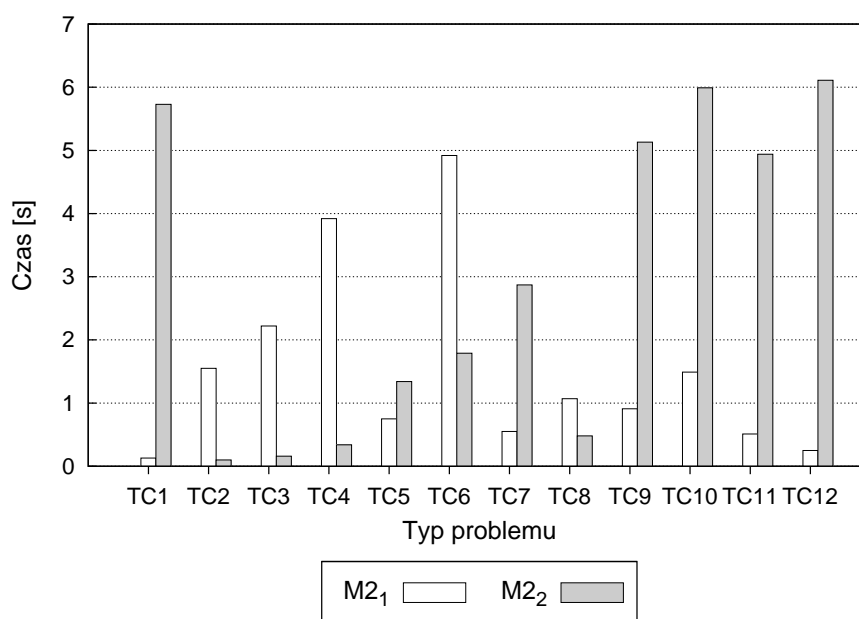
Rysunek 5.2. Średni czas rozwiązania problemu przez model M1 dla 8 lokalizacji

Nieco inaczej wygląda sytuacja w przypadku problemów, które skupiają się na minimalizacji większych wartości składowych wektora wyjściowego (TC2–TC4). Dla kryterium centrum (TC2) nadmiarowe ograniczenia nie zmieniają lub pogarszają wydajność modelu. Sformułowania M1<sub>3</sub> i M1<sub>5</sub> osiągają czasy ok. dwukrotnie gorsze w stosunku do sformułowania podstawowego. W przypadku sformułowań M1<sub>2</sub> i M1<sub>4</sub> czasy rozwiązania nie ulegają praktycznie zmianie. Dla problemów typu TC3 i TC4 zależność jest nieco bardziej złożona. Przy najmniejszych rozmiarach (8, 10 lokalizacji) sformułowania z nadmiarowymi ograniczeniami

osiągają czasy gorsze od sformułowania podstawowego (jedno sformułowanie  $M1_4$  osiąga czasy zbliżone). W miarę wzrostu rozmiaru, a szczególnie dla małej liczby punktów obsługi, sformułowania  $M1_3$  i  $M1_4$  zaczynają osiągać lepsze czasy niż sformułowanie podstawowe.

Dla pozostałych typów problemów (TC5–TC8) trudno określić wyraźną tendencję zmian. W szczególności, porównując problemy TC5 i TC6, dla których wektory wag stanowią naprzemienne ciągi 0 i 1, przy czym w TC5 ciąg zaczyna się od 1, a w TC6 od 0, można zauważyć przeciwny wpływ ograniczeń nadmiarowych na uzyskiwane czasy rozwiązań sformułowania  $M1_3$ . Sformułowania  $M1_2$  i  $M1_5$  osiągają nieco gorsze czasy, a z kolei  $M1_4$  jest zbliżone do sformułowania bez ograniczeń nadmiarowych.

Analiza wyników drugiego modelu ujawnia w dużej części podobne rezultaty jak dla modelu pierwszego (rysunek 5.3), przy czym część różnic i zależności jest jeszcze bardziej widoczna. W większym stopniu skraca się czas po dodaniu ograniczeń nadmiarowych dla problemów z wszystkimi wagami różnymi od 0 (nawet ok. 45-krotnie dla TC1), ale bardziej i konsekwentnie wydłuża się czas dla problemów TC2–TC4 (nawet ok. 15-krotnie dla TC2). Wyniki są również analogiczne dla problemów TC5–TC8, gdzie wpływ ograniczeń nadmiarowych nie jest jednoznaczny.

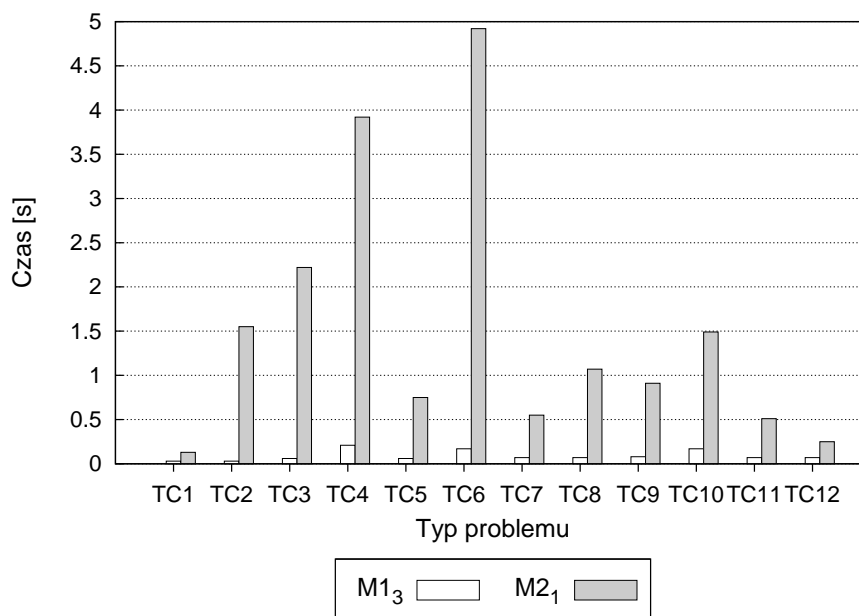


Rysunek 5.3. Średni czas rozwiązania problemu przez model M2 dla 8 lokalizacji

Aby porównać bezpośrednio modele M1 i M2, skonfrontowano ze sobą odpowiednie sformułowania tych modeli, a mianowicie sformułowania z i bez ograniczeń nadmiarowych.

Przy uwzględnieniu ograniczeń nadmiarowych porównano sformułowanie  $M1_3$  z  $M2_1$ . Wybór sformułowania  $M1_3$  w ramach modelu M1 wynikał z mniejszej liczby ograniczeń nadmiarowych rzędu  $m$  w stosunku do sformułowania  $M1_5$  o liczbie ograniczeń nadmiarowych rzędu  $m^2$  — a te dwa sformułowania uzyskiwały najlepsze wyniki w ramach modelu M1. Rysunek 5.4 prezentuje wyniki dla 8 lokalizacji. Jak widać, dla wszystkich typów problemów

pierwszy model pokazuje znacznie lepszą wydajność niż model drugi. Dla typów TC1 i TC12 czas rozwiązania jest krótszy odpowiednio ok. 5 i 4 razy. Dla typów TC7, TC10, TC11 model M1 ma czas krótszy ok. 8–9 razy. Dla pozostałych typów czas rozwiązania skrócił się o ponad jeden rząd wielkości — dla części problemów poprawa była 30–45-krotna (TC2–TC3, TC6).



Rysunek 5.4. Średni czas rozwiązania problemu przez sformułowania z nadmiarowymi ograniczeniami dla 8 lokalizacji

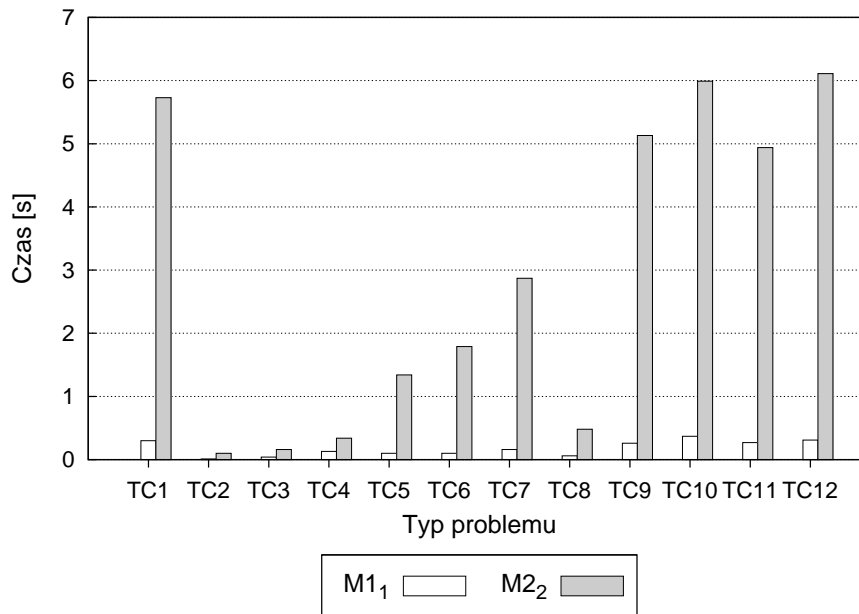
Podobna sytuacja występuje dla sformułowań bez ograniczeń nadmiarowych, gdzie porównaniu poddano sformułowania M1<sub>1</sub> i M2<sub>2</sub> (rysunek 5.5). Tu również dla wszystkich typów problemów model pierwszy jest znacznie wydajniejszy niż jego drugi odpowiednik i dla większości typów skrócenie czasu przekracza jeden rząd wielkości.

Dla 10 lokalizacji różnice między modelami rosną i dla znacznej części problemów model M1 osiąga czasy o ponad dwa rzędy wielkości krótsze niż model M2. Na podstawie uzyskanych wyników można stwierdzić, że model M1 jest wydajniejszy niż model M2.

Ponieważ, jak wspomniano wcześniej, specyficzne problemy z odpowiednio monotonicznymi (malejącymi w przypadku minimalizacji) wagami OWA mogą zostać sformułowane w postaci standardowego programowania liniowego, sprawdzono jak w takich przypadkach radzą sobie modele PCLM. Mówiąc precyzyjniej, interesujące było, czy modele PCLM osiągają lepszą wydajność dla problemów o specyficznej strukturze (wypukłych przy minimalizacji). W tym celu porównano czasy rozwiązania dla problemów z wagami malejącymi (prostsze — TC9, TC11) i wagami rosnącymi (problemy trudniejsze — TC10, TC12). Graficznie, dla 8 lokalizacji, wyniki można przeanalizować na wcześniejszych rysunkach 5.2 i 5.3.

Analiza modelu M1 pokazuje (rysunek 5.2), że w przypadku problemów TC9 i TC10 sformułowania M1<sub>1</sub>–M1<sub>4</sub> uzyskują średnio lepsze czasy rozwiązania dla wag malejących





Rysunek 5.5. Średni czas rozwiązania problemu przez sformułowania bez nadmiarowych ograniczeń dla 8 lokalizacji

(TC9). Różnice maleją jednak wraz ze wzrostem liczby punktów obsługi. Sformułowanie  $M1_5$  osiąga natomiast średnio nieco krótsze czasy dla wag rosnących (TC10). W przypadku problemów TC11 i TC12, sformułowania  $M1_1$ ,  $M1_2$  i  $M1_4$  osiągają średnio nieco krótszy czas rozwiązania dla wag malejących, jednak różnice są jeszcze mniejsze niż dla problemów typu TC9 i TC10. Dla sformułowania  $M1_3$  średni czas rozwiązania można uznać za równy dla obu rodzajów problemów, a dla sformułowania  $M1_5$  różnica względna jeszcze się powiększyła na korzyść wag rosnących (problemy teoretycznie trudniejsze).

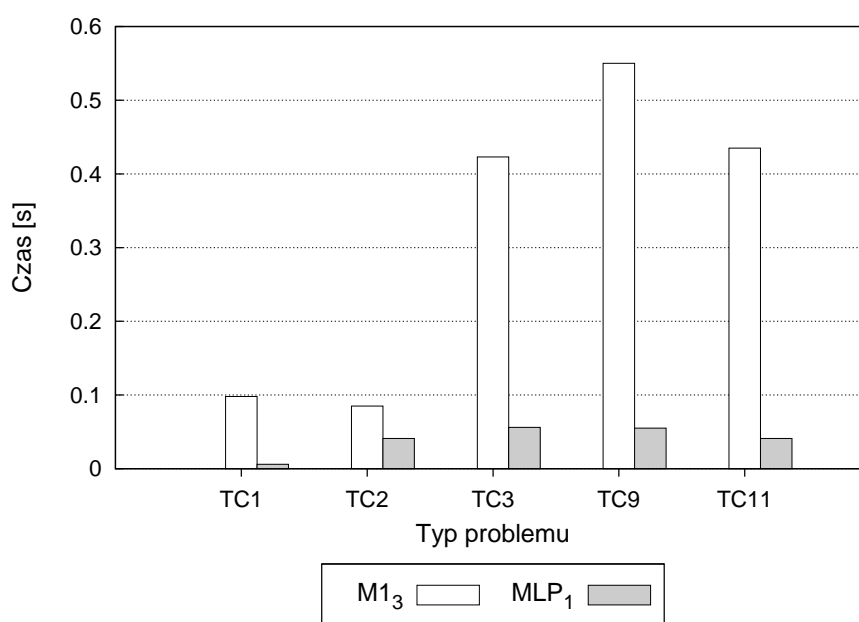
W przypadku modelu M2 (rysunek 5.3) sformułowanie  $M2_2$ , bez ograniczeń nadmiarowych, również liczy się nieco krócej w sensie średnim dla wag malejących. Dotyczy to zarówno problemów TC9–TC10, jak i TC11–TC12. Podobnie jest dla sformułowania  $M2_1$  przy porównaniu problemów TC9 i TC10. Inaczej wygląda jednak sytuacja dla sformułowania  $M2_1$  w przypadku problemów TC11 i TC12. Tutaj to problem z wagami rosnącymi ma krótszy czas rozwiązania i dotyczy to wszystkich wariantów liczby punktów obsługi. W sensie średnim czas rozwiązania dla wag rosnących jest ok. 2-krotnie krótszy niż dla wag malejących.

Bardzo zbliżony jest charakter różnic dla problemów z 10 lokalizacjami. Przeprowadzona analiza sugeruje, że modele PCLM nie gwarantują lepszej wydajności dla problemów z wagami malejącymi, nawet w sensie średnim.

### 5.7.2. Wyniki odchyleniowego modelu PL i porównanie z modelem PCLM

Modele PCLM nie gwarantują lepszej wydajności dla prostszych problemów z odpowiednio monotonicznymi wagami. Dla części takich instancji osiągały jednak nieco lepsze czasy niż dla problemów teoretycznie trudniejszych. Aby porównać bezpośrednio tę poprawę z potencjalnym

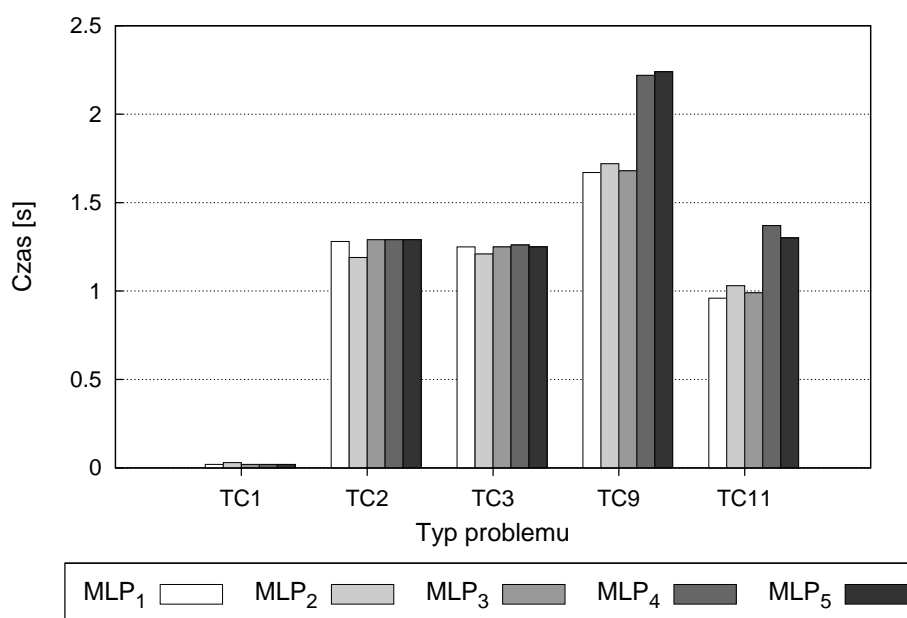
zyskiem, który można osiągnąć, stosując podejście dedykowane do specyficznej struktury zadania, zestawiono ze sobą czasy rozwiązania modelu całkowitoliczbowego mieszanego i sformułowania liniowego OWA. Skonfrontowano czasy rozwiązania najwydajniejszego rozwiązującego modelu PCLM dla wag malejących (sformułowanie  $M1_3$ ) i modelu liniowego. Porównanie przeprowadzono dla problemów z wagami malejącymi (TC9, TC11) oraz wagami nierosnącymi (TC1–TC3). Ze względu na krótkie czasy rozwiązania sformułowań liniowych oraz ograniczoną dokładność pomiaru czasu porównanie przeprowadzono dla nieco większych problemów (12 i 15 lokalizacji). Szczegółowe wyniki sformułowania liniowego zamieszczone są w dodatku A.2. Na rysunku 5.6 zaprezentowano wyniki porównania dla 12 lokalizacji. Jak widać, nawet najlepszy rozpatrywany model mieszanego programowania całkowitoliczbowego osiąga znacznie gorsze czasy niż sformułowanie liniowe OWA. Czasy dla modelu liniowego są ok. 10 razy krótsze dla wag malejących (TC9, TC11). W przypadku kryterium średniej (TC1) różnica jest ok. 15-krotna, a dla kryterium  $k$ -centrów (TC3) ok. 8-krotna. Dla 15 lokalizacji przewaga modelu liniowego jest jeszcze większa. Najmniejsza różnica występuje dla kryterium centrum TC2. Czasy modelu  $M1_3$  są tutaj około 2 razy dłuższe. Biorąc pod uwagę, że dla tego kryterium nadmiarowe ograniczenia zazwyczaj wydłużają czas rozwiązania, można oczekiwać, że czasy sformułowania  $M1_1$  (bez ograniczeń nadmiarowych) powinny być jeszcze bardziej zbliżone do modelu liniowego. I faktycznie czasy sformułowania  $M1_1$  są prawie identyczne jak modelu liniowego OWA (patrz szczegółowe wyniki w dodatkach A.1, A.2). Jednak ten wyjątek również potwierdza lepszą wydajność modelu liniowego, gdyż dla kryterium centrum wszystkie zmienne binarne w modelu PCLM (5.3) są równe 0 zgodnie z ograniczeniem (5.3c). Tym samym model (5.3) degeneruje się do modelu liniowego.



Rysunek 5.6. Średni czas rozwiązania problemu przez sformułowanie  $M1_3$  oraz sformułowanie liniowe OWA dla wag nierosnących dla 12 lokalizacji

Przeprowadzone porównanie pokazuje, że nawet jeśli modele PCLM są bardziej wydajne dla niektórych instancji problemów z odpowiednio monotonicznymi wagami niż dla ogólnego przypadku OWA, to i tak mają znacznie gorszą wydajność niż modele z liniowym sformułowaniem kryterium OWA dla tych specyficznych (prostszych) przypadków.

Wiedząc, że liniowe sformułowanie OWA osiąga znacznie krótsze czasy niż modele mieszanego programowania liniowego oraz że ograniczenia nadmiarowe mogą znacznie poprawić wydajność tych drugich, zbadano wpływ ograniczeń nadmiarowych na liniowe modele OWA. Porównano czasy rozwiązania pięciu sformułowań przedstawionych w podrozdziale 5.4 dla problemów z malejącymi wagami (TC9, TC11) oraz dodatkowo dla problemów TC1–TC3 z wagami nierosnącymi. Pełne wyniki dla sformułowań liniowych zamieszczone są w dodatku A.2. Na rysunku 5.7 przedstawiono wyniki dla 30 lokalizacji. Można zauważyć, że w przypadku typów TC1–TC3 wszystkie liniowe sformułowania ( $MLP_1$ – $MLP_5$ ) osiągają zbliżone czasy rozwiązania. Analizując wyniki sformułowań  $MLP_1$ – $MLP_3$ , nie widać również znaczących różnic dla dwóch pozostałych typów problemów. Sformułowania  $MLP_4$  i  $MLP_5$  uzyskują około 30%–40% dłuższe czasy w sensie średnim dla problemów typu TC9 i TC11. Pomijając niejednoznaczne wyniki dla najmniejszych problemów ze względu na ograniczoną dokładność pomiaru, rezultaty uzyskane dla innych rozmiarów (liczby lokalizacji) są analogiczne. Uzyskane wyniki sugerują, że ograniczenia nadmiarowe nie poprawiają wydajności modeli optymalizacyjnych z liniowym sformułowaniem operatora OWA.

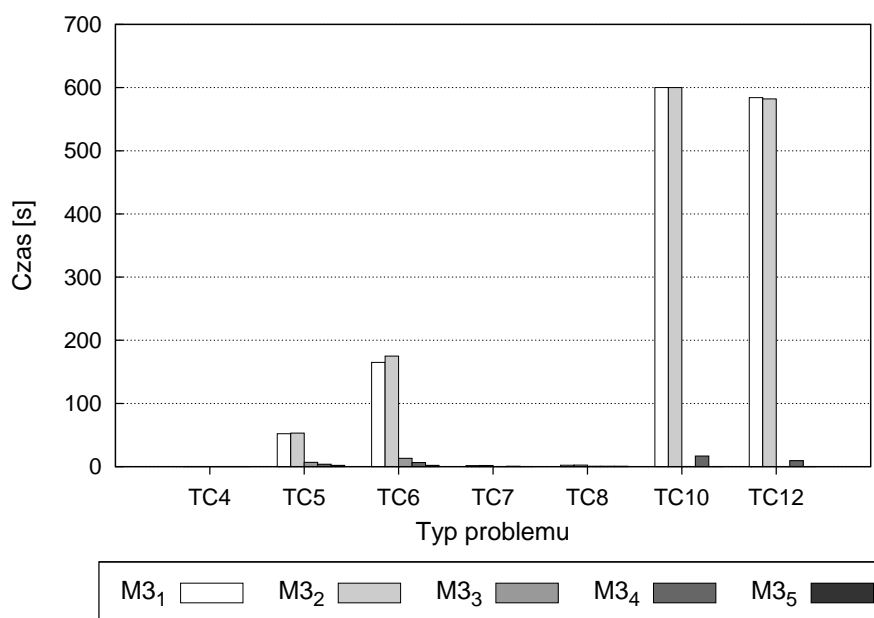


Rysunek 5.7. Średni czas rozwiązania problemów przez liniowe sformułowania OWA dla 30 lokalizacji

### 5.7.3. Wyniki hybrydowych modeli PCLM M3 i M4

W ramach testów obliczeniowych zbadano czasy uzyskiwane przez modele M3 i M4 dla problemów z wagami niemonotonicznymi lub rosnącymi. W przypadku problemów z wagami nierosnącymi modele M3 i M4 upraszczają się do modelu PL, którego rezultaty omówiono powyżej. Szczegółowe wyniki modeli M3 i M4 znajdują się w dodatku A.3.

Analiza wyników modelu M3 pokazuje duże zróżnicowanie czasów rozwiązania zarówno w zależności od typu problemu, jak i obecności części ograniczeń nadmiarowych. Na rysunku 5.8 przedstawiono wyniki dla modelu M3 dla 8 lokalizacji uśrednione dodatkowo po wariantach liczby punktów obsługi. To co konsekwentnie dotyczy wszystkich typów, to minimalne różnice czasowe między sformułowaniami  $M3_1$  i  $M3_2$  (z ograniczeniem nadmiarowym (5.28)). Świadczy to o braku istotnego wpływu ograniczenia (5.28) na czas rozwiązania.



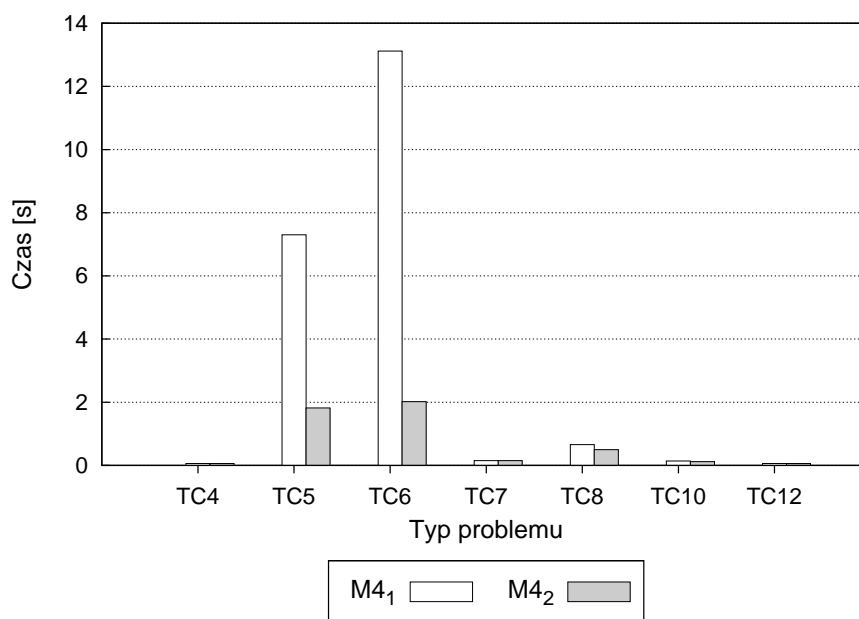
Rysunek 5.8. Średni czas rozwiązania problemu przez model M3 dla 8 lokalizacji

Zgodnie z oczekiwaniami najlepsze czasy rozwiązania osiągnęte są dla problemów zawężonej średniej TC4. Najkrótszy czas osiąga sformułowanie  $M3_1$  (bez ograniczeń nadmiarowych). Praktycznie identyczny czas ma sformułowanie  $M3_4$  (z nadmiarowym ograniczeniem (5.30)), które dla kryterium TC4 jest tożsame z  $M3_1$ , gdyż ograniczenie (5.30) jest określone tylko wtedy, gdy istnieją co najmniej dwie ujemne wagi  $w'_k$ . Ograniczenie (5.29) w  $M3_3$  wydłuża nieco czas rozwiązania dla TC4, jednak pogorszenie to wynosi tylko ok. 20%.

Dla pozostałych typów problemów duży wpływ na czas rozwiązania mają ograniczenia (5.29) i (5.30), które pozwalają na jego znaczne skrócenie. Sformułowanie  $M3_3$  (z ograniczeniem nadmiarowym (5.29)) osiąga czasy ok. 10 razy krótsze w stosunku do sformułowania bez ograniczeń nadmiarowych dla problemów typu TC5, TC6, a sformułowanie  $M4_4$  (z ograniczeniem nadmiarowym (5.30)) skraca ten czas jeszcze ok. dwukrotnie. W nieco

mniejszym stopniu, ale również kilkakrotnie, ograniczenia te skracają czas dla kryteriów TC7, TC8. Ciekawa sytuacja ma miejsce dla problemów typu TC10, TC12. Sformułowanie  $M3_4$  (z ograniczeniem (5.30)) ma czasy o ponad jeden rząd wielkości krótsze w porównaniu do sformułowania podstawowego. Z kolei sformułowanie  $M3_3$  (z ograniczeniem (5.29)) osiąga czasy nawet o ponad trzy rzędy wielkości krótsze niż sformułowanie podstawowe. Sformułowanie  $M3_5$  (z dwoma ograniczeniami nadmiarowymi (5.29), (5.30)), pozwala zazwyczaj na jeszcze większe skrócenie czasu niż sformułowania z pojedynczymi ograniczeniami, ale zdecydowanie mniejsze niż iloczyn wielokrotności skrócenia w ramach pojedynczych ograniczeń.

Analiza modelu M4 pokazuje nieco mniejsze zróżnicowanie czasowe ze względu na obecność ograniczenia nadmiarowego. Na rysunku 5.9 przedstawiono wyniki modelu M4 dla 8 lokalizacji. Ograniczenie nadmiarowe (5.30) w największym stopniu wpływa na problemy typu TC5, TC6. Skraca ono czas rozwiązania ok. 4–6 razy. Dla pozostałych typów problemów, pomijając problemy zawężonej średniej, gdzie ograniczenie to nie jest określone, również skraca ono czas rozwiązania, ale w niewielkim stopniu.

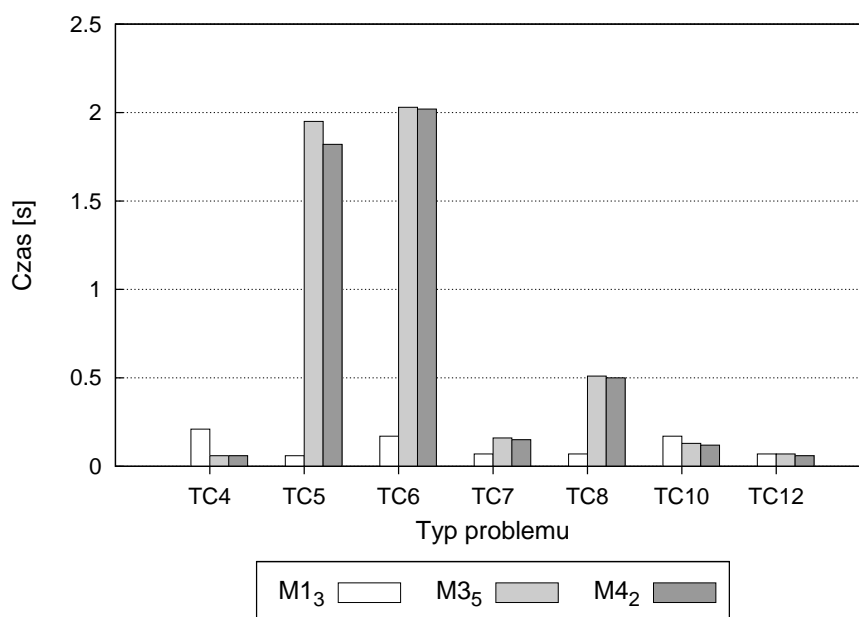


Rysunek 5.9. Średni czas rozwiązania problemu przez model M4 dla 8 lokalizacji

Mniejszy wpływ ograniczenia nadmiarowego w modelu M4 w stosunku do modelu M3 wynika w dużym stopniu ze znacząco lepszych czasów rozwiązań sformułowania podstawowego w przypadku modelu M4. Analiza wyników pokazuje, że sformułowanie  $M4_1$ , bez ograniczeń nadmiarowych, uzyskuje wyniki prawie identyczne jak sformułowanie  $M3_3$ , z ograniczeniem nadmiarowym (5.29). Te z kolei, jak wcześniej przedstawiono, uzyskuje czasy wielokrotnie krótsze niż sformułowanie  $M3_1$  dla problemów typu TC5–TC8, TC10 i TC12.

Okazuje się również, że sformułowanie  $M4_2$  (z jednym ograniczeniem nadmiarowym),

uzyskuje bardzo zbliżone wyniki do sformułowania  $M3_5$  (z dwoma ograniczeniami nadmiarowymi). Te dwa sformułowania uzyskują najlepsze czasy w ramach modeli hybrydowych dla zdecydowanej większości problemów (jedynie dla problemów zawężonej średniej sformułowanie  $M3_1$  radzi sobie nieco lepiej). Ich wyniki zestawiono z wynikami modelu  $M1$ , który wcześniej okazał się zdecydowanie wydajniejszy od modelu  $M2$ . Na początku do porównania wykorzystano sformułowanie  $M1_3$ , które wydaje się najbardziej atrakcyjne w ramach modelu  $M1$ , biorąc pod uwagę wyniki zbiorcze. Uzyskiwało ono najlepsze lub niewiele gorsze od najlepszych czasy rozwiązania dla rozpatrywanych typów problemów. Na rysunku 5.10 przedstawiono takie porównanie dla 8 lokalizacji. Widać, że sformułowanie  $M1_3$  uzyskuje znacznie lepsze czasy dla problemów typu TC5–TC8. Inaczej jest jednak dla typu TC4, gdzie zgodnie z oczekiwaniem krótsze czasy osiągają sformułowania modeli  $M3$  i  $M4$ . Co ciekawe sformułowania modeli  $M3$  i  $M4$  osiągają również lepsze czasy dla problemów typu TC10 oraz zbliżone rezultaty dla problemów typu TC12. Jest to o tyle interesujące, że dla tych dwóch typów liczba ujemnych wag  $w'_k$  jest największa ( $m - 1$  wag  $w'_k$  jest ujemnych), a przez to część dyskretna zadania jest największa (duża liczba zmiennych binarnych). Te krótkie czasy wynikają prawdopodobnie z braku wag zerowych, co poprawia stabilność przeszukiwania zbioru rozwiązań dopuszczalnych — mniejsze są różnice wartości funkcji celu dla różnych rozwiązań dopuszczalnych, co wygładza w pewnym sensie funkcję celu.



Rysunek 5.10. Średni czas rozwiązania problemu przez najlepsze sformułowania modeli  $M1$ ,  $M3$  i  $M4$  dla 8 lokalizacji

Następnie przeprowadzono dokładniejsze porównanie rezultatów modeli hybrydowych z wcześniejszymi modelami PCLM dla problemów typu TC4 i TC10. Najlepszą wersję modelu hybrydowego  $M4_2$  zestawiono ze sformułowaniem  $M1_3$ , a także dodatkowo z wersjami  $M1_4$  i  $M1_5$ , które uzyskują jeszcze lepsze wyniki odpowiednio dla typów TC4 i TC10. W tabeli

5.2 przedstawiono średnie czasy rozwiązań problemów z 8, 10, 12 i 15 lokalizacjami dla poszczególnych wariantów liczby punktów obsługi. Widać teraz wyraźnie przewagę modelu hybrydowego nad modelem M1. Dla kryterium zawężonej średniej (TC4) dotyczy to zwłaszcza problemów z mniejszą liczbą punktów obsługi, gdzie przy 15 lokalizacjach M4<sub>2</sub> ma czasy ponad dwa rzędy wielkości krótsze niż M1<sub>3</sub> oraz ponad rząd wielkości krótsze niż M1<sub>4</sub>. Dla problemów typu TC10 różnice są nieco mniejsze, ale nadal znaczne. W stosunku do M1<sub>3</sub> czasy są kilkukrotnie, a nawet o jeden rząd wielkości krótsze, natomiast w stosunku do M1<sub>5</sub> kilkukrotnie do jednego rzędu wielkości krótsze.

Tabela 5.2: Porównanie najbardziej efektywnych modeli PCLM agregacji OWA dla problemów typu TC4 i TC10 (górny indeks przed czasem oznacza liczbę instancji spośród 15, które przekroczyły ustalony limit czasu 600 s )

Problem			CPU[s]			
typ	m	n	M1 <sub>3</sub>	M1 <sub>4</sub>	M1 <sub>5</sub>	M4 <sub>2</sub>
TC4	8	2	0,35	0,20	0,27	0,08
		3	0,29	0,14	0,23	0,06
		4	0,15	0,06	0,19	0,06
		5	0,06	0,02	0,11	0,05
	10	3	1,85	0,89	1,13	0,15
		4	1,33	0,56	0,91	0,13
		5	0,67	0,23	0,76	0,11
		6	0,28	0,08	0,46	0,10
	12	3	11,80	3,81	6,34	0,64
		4	7,86	1,97	5,53	0,39
		6	2,11	0,28	2,28	0,37
		7	1,33	0,10	1,52	0,33
	15	4	162,75	56,47	79,38	0,90
		5	94,90	25,88	49,06	0,81
		8	11,70	2,39	10,61	0,77
9		5,56	0,59	6,22	0,73	
TC10	8	2	0,30	0,35	0,24	0,21
		3	0,22	0,39	0,19	0,14
		4	0,11	0,26	0,10	0,09
		5	0,04	0,17	0,04	0,04
	10	3	1,81	1,74	0,91	0,66
		4	0,79	1,56	0,63	0,35
		5	0,46	0,85	0,33	0,19
		6	0,16	0,49	0,16	0,11
	12	3	11,79	18,84	8,83	3,64
		4	10,95	14,66	7,11	1,61

kontynuacja na następnej stronie

Problem			CPU[s]			
typ	m	n	M1 <sub>3</sub>	M1 <sub>4</sub>	M1 <sub>5</sub>	M4 <sub>2</sub>
		6	1,77	3,91	1,26	0,46
		7	0,54	1,52	0,49	0,20
	15	4	156,83	<sup>1</sup> 213,24	112,38	37,00
		5	101,02	<sup>1</sup> 154,77	62,39	7,54
		8	11,19	15,21	6,41	0,91
		9	3,05	6,47	2,59	0,54

Zaprezentowane wyniki świadczą o użyteczności nowych modeli dla części typów problemów. Dotyczy to zwłaszcza kryterium zawężonej średniej, która wydaje się kryterium o największym znaczeniu praktycznym spośród kryteriów, do których nie można zastosować modelu liniowego ze względu na to, że wagi nie spełniają odpowiedniego warunku monotoniczności.

## 5.8. Wnioski

W rozdziale studiowano modele optymalizacyjne z agregacją OWA dla dyskretnych problemów lokalizacyjnych. Badano różne sformułowania operatora uporządkowanej średniej ważonej. Przeanalizowano i porównano zaczerpnięte z literatury dwa modele PCLM agregacji OWA oraz liniowy model OWA dla specyficznych wag preferencji. Zaproponowano również nowy hybrydowy model PCLM agregacji OWA, a dokładnie jego dwie wersje. Stanowi on rozszerzenie modelu liniowego, do którego dodano część dyskretną, co pozwala na jego zastosowanie dla dowolnych wag preferencji. Przeprowadzono eksperymenty w celu porównania wydajności obliczeniowej poszczególnych modeli i ich różnych sformułowań z dodatkowymi ograniczeniami.

Na podstawie otrzymanych wyników można stwierdzić, że nadmiarowe ograniczenia w modelach mieszanego programowania całkowitoliczbowego OWA mogą znacznie poprawić efektywność i wydajność obliczeniową dla pewnych typów problemów lokalizacyjnych (pewnych klas wektorów wag OWA). Do tej klasy zaliczają się problemy ze wszystkimi niezerowymi wagami preferencji. Po drugie, dla badanych rozmiarów problemów model M1 okazał się znacznie efektywniejszy niż model M2. Potwierdza to analizę teoretyczną, która dowodzi, że model M1 odpowiada modelowi M2 ze wzmocnionym ograniczeniem i częściowo zmodyfikowanymi zmiennymi. Poza tym jeśli badany problem pozwala na sformułowanie kryterium OWA w standardowej postaci liniowej, to należy to wykorzystać, gdyż znacznie zwiększy to jego wydajność obliczeniową. Wydaje się również, że ograniczenia nadmiarowe w liniowym sformułowaniu OWA nie zwiększają jego efektywności, a dla niektórych typów problemów mogą wydłużać czas rozwiązania. Warto tutaj podkreślić, że dotyczy to ograniczeń bezpośrednio związanych z częścią liniową, stanowiącą model preferencji. Nie wyklucza to potencjalnego pozytywnego wpływu ograniczeń nadmiarowych, które dotyczyłyby części



dyskretnej odpowiadającej rzeczowemu modelowi lokalizacyjnemu — takie ograniczenia nie były tutaj rozpatrywane.

Zaproponowane nowe modele udowadniają swoją użyteczność dla części typów problemów, gdzie wagi preferencji nie spełniają odpowiedniego warunku monotoniczności (warunku nierosnących wartości). Widoczne jest to szczególnie dla kryterium zawężonej średniej, gdzie nowe sformułowania uzyskują wyniki znacznie lepsze niż najlepsze ze znanych dotychczas modeli PCLM dla dyskretnych zadań lokalizacyjnych. Jest to tym ważniejsze, że kryterium zawężonej średniej wydaje się mieć największe znaczenie praktyczne spośród pozostałych kryteriów, w których wagi preferencji nie spełniają warunku nierosnących wartości. Efektywność nowych modeli jest zdecydowanie gorsza dla problemów typu TC5–TC8. Są to jednak raczej kryteria sztuczne, o małej wartości praktycznej. Zaskakująca jest również dobra efektywność tych modeli dla problemów o wagach rosnących, czyli teoretycznie najtrudniejszych. Przykładem są problemy typu TC10, gdzie czasy nowych modeli są krótsze niż czasy modeli PCLM z literatury.

## 6. Wartościowana uporządkowana średnia ważona

W zagadnieniach lokalizacyjnych zbiorów odbiorców często nie jest jednorodny, tzn. poszczególni odbiorcy różnią się między sobą. Przykładem takiej sytuacji może być problem lokalizacyjny, gdzie odbiorcami są miasta o różnej liczbie mieszkańców. Liczba mieszkańców determinuje zróżnicowane wielkości zapotrzebowania poszczególnych miast. Tym samym w modelu pojawia się potrzeba zróżnicowania zapotrzebowania odbiorców. Jak wynika z przytoczonego przykładu, zapotrzebowanie jest ściśle związane z konkretnym odbiorcą. Bezpośrednie zastosowanie agregacji OWA nie jest możliwe, gdyż ze względu na własność anonimowości uporządkowana średnia ważona nie pozwala na przypisanie różnych charakterystyk do poszczególnych kryteriów. Aby umożliwić jej zastosowanie, problem musiałby zostać zdezagregowany do równoważnych odbiorców, np. do pojedynczych mieszkańców. Taka operacja, choć teoretycznie możliwa, prowadzi jednak to znacznego wzrostu rozmiaru problemu, a w konsekwencji do niemożliwości jego rozwiązania.

Torra [84] opracował operator agregacji, który pozwala uwzględnić wagi wartościujące przypisane konkretnym kryteriom, spełniając jednocześnie warunek bezstronności w sensie rozkładu. Tym operatorem jest wartościowana uporządkowana średnia ważona (ang. Weighted Ordered Averaging, WOWA). Zastosowanie agregacji WOWA w problemach lokalizacyjnych umożliwia poszukiwanie schematu rozmieszczenia optymalnego w sensie rozkładu z jednoczesnym uwzględnieniem zróżnicowanych zapotrzebowań poszczególnych odbiorców.

### 6.1. Klasyczna definicja operatora WOWA

Pierwotną definicję operatora agregacji WOWA przedstawił Torra [84].

**Definicja 6.1.** Dane są  $m$ -wymiarowe wektory wag  $\mathbf{w} = (w_1, \dots, w_m)$  i  $\mathbf{p} = (p_1, \dots, p_m)$ , gdzie wektor  $\mathbf{w}$  stanowi wagi preferencji, a wektor  $\mathbf{p}$  wagi wartościujące. Oba wektory wag są znormalizowane, czyli  $0 \leq w_i \leq 1$ ,  $0 \leq p_i \leq 1$  dla  $i = 1, \dots, m$  oraz  $\sum_{i=1}^m w_i = 1$ ,  $\sum_{i=1}^m p_i = 1$ . Operator agregacji WOWA wektora  $\mathbf{y} = (y_1, \dots, y_m)$  wyraża się w postaci

$$A_{\mathbf{w}, \mathbf{p}}(\mathbf{y}) = \sum_{i=1}^m \omega_i \theta_i(\mathbf{y}), \quad (6.1)$$

gdzie wagi  $\omega_i$  są określone jako

$$\omega_i = w^* \left( \sum_{k \leq i} p_{\tau(k)} \right) - w^* \left( \sum_{k < i} p_{\tau(k)} \right), \quad (6.2)$$

gdzie  $w^*$  jest niemalejącą funkcją, która łączy punkty  $(\frac{i}{m}, \sum_{k \leq i} w_k)$  oraz punkt  $(0, 0)$ , a  $\tau$  to permutacja porządkująca wektor  $y$  (tzn.  $y_{\tau(i)} = \theta_i(y)$ ). Funkcja  $w^*$  musi reprezentować linię prostą, jeśli punkty można interpolować w taki sposób.

Dzięki ostatniemu wymaganiu powyższej definicji, gdy wszystkie wagi preferencji są równe ( $w_i = 1/m$  dla  $i = 1, \dots, m$ ), agregacja WOWA jest równoważna średniej ważonej z wagami  $p_i$ . Z kolei gdy równe są wszystkie wagi wartościujące ( $p_i = 1/m$  dla  $i = 1, \dots, m$ ), operator WOWA staje się równoważny operatorowi OWA. Widać więc, że agregacja WOWA uogólnia zarówno zwykłą średnią ważoną, jak i agregację OWA. Podobnie jak OWA, operator WOWA spełnia również zasadę jednomyślności, a jego wartość zawiera się pomiędzy minimalną i maksymalną oceną. Sam operator WOWA stanowi szczególny przypadek całki Choqueta [29].

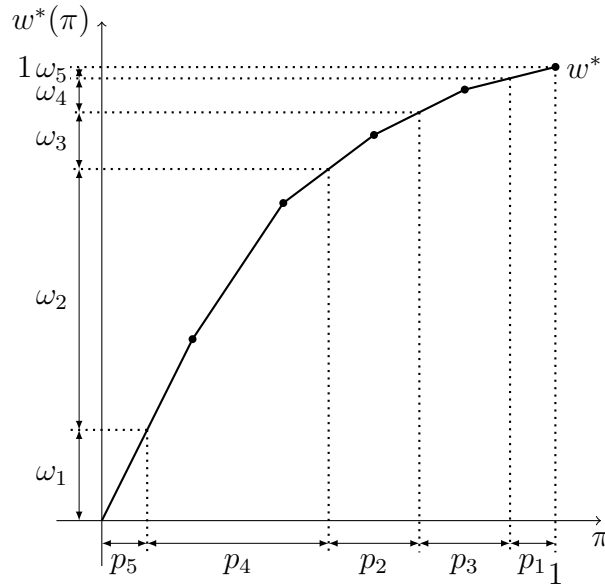
Poza wymaganiem na linię prostą, gdy punkty  $(\frac{i}{m}, \sum_{k \leq i} w_k)$  faktycznie leżą na jednej prostej, definicja operatora WOWA jest elastyczna w kwestii sposobu łączenia punktów  $(\frac{i}{m}, \sum_{k \leq i} w_k)$ , co pozwala stosować różne interpolacje funkcji  $w^*$  [85]. W pracy ograniczono się do najprostszej interpolacji w postaci funkcji przedziałami liniowej. Warto jednak zaznaczyć, że funkcja interpolująca może być zbudowana na dowolnej liczbie przedziałów, nie koniecznie  $m$ . Dzięki temu dowolna funkcja nieliniowa może być wystarczająco dobrze aproksymowana funkcją przedziałami liniową z odpowiednią liczbą przedziałów. W związku z tym w pracy rozważany będzie wektor wag preferencji  $w$  o wymiarze  $\tilde{m}$  niekoniecznie równym  $m$ .

Poniższy przykład pozwala prześledzić procedurę wyznaczania wartości operatora WOWA.

**Przykład 6.1.** W ramach przykładu rozważany jest problem z 5 lokalizacjami. Niech wektor kosztów obsługi wynosi  $y = (1; 3; 2; 4; 5)$ , a znormalizowane wagi zapotrzebowania  $p = (0,1; 0,2; 0,2; 0,4; 0,1)$ . Czyli lokalizacja druga i trzecia ma 2 razy większe zapotrzebowanie, a lokalizacja czwarta ma 4 razy większe zapotrzebowanie niż lokalizacja pierwsza (lokalizacja piąta ma takie samo zapotrzebowanie jak pierwsza). Z kolei wektor wag preferencji to  $w = (0,4; 0,3; 0,15; 0,1; 0,05)$ . Na podstawie wektora  $w$  można wyznaczyć funkcję interpolującą  $w^*$ . Zgodnie z definicją będzie ona łączyła punkty  $(0; 0)$ ,  $(1/5; 0,4)$ ,  $(2/5; 0,7)$ ,  $(3/5; 0,85)$ ,  $(4/5; 0,95)$ ,  $(1; 1)$ . Zakładając przedziałami liniowy charakter funkcji, można ją wyrazić w postaci

$$w^*(\pi) = \begin{cases} \frac{0,4}{0,2}\pi & \text{dla } 0 \leq \pi \leq 0,2, \\ 0,4 + \frac{0,3}{0,2}(\pi - 0,2) & \text{dla } 0,2 < \pi \leq 0,4, \\ 0,7 + \frac{0,15}{0,2}(\pi - 0,4) & \text{dla } 0,4 < \pi \leq 0,6, \\ 0,85 + \frac{0,1}{0,2}(\pi - 0,6) & \text{dla } 0,6 < \pi \leq 0,8, \\ 0,95 + \frac{0,05}{0,2}(\pi - 0,8) & \text{dla } 0,8 < \pi \leq 1. \end{cases}$$

Uporządkowany wektor kosztów to  $(5; 4; 3; 2; 1)$ , a odpowiadający mu wektor wag zapotrzebowania  $(0,1; 0,4; 0,2; 0,2; 0,1)$ . Można teraz wyznaczyć wagi  $\omega_i$  ( $i = 1, \dots, m$ )



Rysunek 6.1. Wagi  $\omega_i$  agregacji WOVA zdefiniowane w oparciu o funkcję  $w^*$  z przykładu 6.1

agregacji WOVA zgodnie z (6.2). Mianowicie:

$$\omega_1 = w^*(p_5) = w^*(0,1) = 0,2,$$

$$\omega_2 = w^*(p_5 + p_4) - w^*(p_5) = 0,775 - 0,2 = 0,575,$$

$$\omega_3 = w^*(p_5 + p_4 + p_2) - w^*(p_5 + p_4) = 0,9 - 0,775 = 0,125,$$

$$\omega_4 = w^*(p_5 + p_4 + p_2 + p_3) - w^*(p_5 + p_4 + p_2) = 0,975 - 0,9 = 0,075,$$

$$\omega_5 = w^*(p_5 + p_4 + p_2 + p_3 + p_1) - w^*(p_5 + p_4 + p_2 + p_3) = 1 - 0,975 = 0,025.$$

Procedura wyznaczania wag  $\omega_i$  została również zobrazowana na rysunku 6.1. Ostatecznie wartość operatora WOVA, zgodnie z (6.1), wynosi  $A_{\mathbf{w},\mathbf{p}}(\mathbf{y}) = 0,2 \cdot 5 + 0,575 \cdot 4 + 0,125 \cdot 3 + 0,075 \cdot 2 + 0,025 \cdot 1 = 3,85$ .

## 6.2. Alternatywna definicja operatora WOVA

Oryginalna definicja operatora WOVA bazuje na wyznaczaniu wag  $\omega_i$  ( $i = 1, \dots, m$ ) w oparciu o funkcję  $w^*$  dla uporządkowanych składowych wektora  $\mathbf{y}$ . Agregację WOVA można zdefiniować w sposób alternatywny, gdzie wyznacza się pewnego rodzaju średnie ważone uporządkowanych składowych wektora  $\mathbf{y}$  w ramach równych porcji zapotrzebowania dla poszczególnych wag preferencji  $w_i$  ( $i = 1, \dots, \tilde{m}$ ).

Funkcję  $w^*$  można wyrazić jako  $w^*(\alpha) = \int_0^\alpha g(\xi) d\xi$  z funkcją tworzącą  $g$ . Definiując punkty graniczne zgodnie z wagami wartościującymi  $\beta_i = \sum_{k \leq i} p_{\tau(k)}$  oraz  $\beta_0 = 0$ ,

poszczególne wagi  $\omega_i$  można wyznaczyć jako

$$\omega_i = \int_0^{\beta_i} g(\xi) d\xi - \int_0^{\beta_{i-1}} g(\xi) d\xi = \int_{\beta_{i-1}}^{\beta_i} g(\xi) d\xi.$$

Cała agregacja WOWA wyraża się więc jako

$$A_{\mathbf{w},\mathbf{p}}(\mathbf{y}) = \sum_{i=1}^m \theta_i(\mathbf{y}) \int_{\beta_{i-1}}^{\beta_i} g(\xi) d\xi = \int_0^1 g(\xi) F_{\mathbf{y}}^{(-1)}(\xi) d\xi, \quad (6.3)$$

gdzie  $F_{\mathbf{y}}^{(-1)}$  jest funkcją schodkową  $F_{\mathbf{y}}^{(-1)}(\xi) = \theta_i(\mathbf{y})$  dla  $\beta_{i-1} < \xi \leq \beta_i$ . Funkcja ta może być również sformalizowana w następujący sposób. Na początku definiuje się lewostronnie ciągłą prawostronną dystrybuantę

$$F_{\mathbf{y}}(d) = \sum_{i \in I} p_i \delta_i(d), \quad \text{gdzie} \quad \delta_i(d) = \begin{cases} 1 & \text{jeśli } y_i \geq d \\ 0 & \text{w p.p.} \end{cases}, \quad (6.4)$$

która dla dowolnej wartości rzeczywistej  $d$  stanowi miarę wielkości zapotrzebowania odpowiadającą ocenom większym lub równym  $d$ . Na tej podstawie można zdefiniować funkcję kwantylową  $F_{\mathbf{y}}^{(-1)}$  jako lewostronnie ciągłą odwrotność dystrybuanty  $F_{\mathbf{y}}$

$$F_{\mathbf{y}}^{(-1)}(\xi) = \sup \{ \eta : F_{\mathbf{y}}(\eta) \geq \xi \} \quad \text{dla} \quad 0 < \xi \leq 1. \quad (6.5)$$

Funkcja ta dla danej wartości  $\xi \in (0, 1]$  oznacza minimalny koszt obsługi dla co najmniej  $\xi$ -tej części sumarycznego zapotrzebowania. Inaczej mówiąc, co najmniej  $\xi$ -ta część sumarycznego zapotrzebowania jest obsługiwana z kosztem większym lub równym  $F_{\mathbf{y}}^{(-1)}(\xi)$ .

Agregacja WOWA zdefiniowana zgodnie z równaniem (6.3) jest bardzo ogólnym sformułowaniem operatora, który może być również zastosowany do przypadku ciągłego. Funkcja tworząca  $g$  zależy bezpośrednio od zastosowanej interpolacji funkcji  $w^*$ . Jak założono przy definicji klasycznej, w pracy będzie rozważana najprostsza interpolacja  $w^*$  w postaci funkcji przedziałami liniowej. Funkcja ta może być jednak zbudowana na dowolnej liczbie przedziałów, nie koniecznie  $m$ , dzięki czemu dowolna funkcja nieliniowa może być wystarczająco dobrze aproksymowana funkcją przedziałami liniową z odpowiednią liczbą przedziałów  $\tilde{m}$ . Dowolna przedziałami liniowa funkcja interpolująca  $w^*$  może być wyrażona za pomocą schodkowej funkcji tworzącej

$$g(\xi) = \tilde{m} w_k \quad \text{dla} \quad (k-1)/\tilde{m} < \xi \leq k/\tilde{m}, \quad k = 1, \dots, \tilde{m}. \quad (6.6)$$

Podstawiając (6.6) do równania (6.3), otrzymuje się

$$A_{\mathbf{w},\mathbf{p}}(\mathbf{y}) = \int_0^1 g(\xi) F_{\mathbf{y}}^{(-1)}(\xi) d\xi = \sum_{k=1}^{\tilde{m}} w_k \tilde{m} \int_{(k-1)/\tilde{m}}^{k/\tilde{m}} F_{\mathbf{y}}^{(-1)}(\xi) d\xi. \quad (6.7)$$

Równanie (6.7) stanowi formalną definicję agregacji WOWA dla  $m$ -wymiarowego wektora ocen  $\mathbf{y}$  z  $m$ -wymiarowym wektorem wag wartościujących  $\mathbf{p}$  oraz  $\tilde{m}$ -wymiarowym wektorem wag preferencji  $\mathbf{w}$ . Poniższy przykład pozwala prześledzić procedurę wyznaczania wartości operatora WOWA zgodnie z (6.7).

**Przykład 6.2.** Wartość operatora WOWA zostanie wyznaczona dla tego samego problemu co w przykładzie 6.1, gdzie wektor kosztów obsługi  $\mathbf{y} = (1; 3; 2; 4; 5)$ , znormalizowane wagi zapotrzebowania  $\mathbf{p} = (0,1; 0,2; 0,2; 0,4; 0,1)$ , a wektor wag preferencji  $\mathbf{w} = (0,4; 0,3; 0,15; 0,1; 0,05)$ . Na podstawie wektora kosztów  $\mathbf{y}$  i wag zapotrzebowania  $\mathbf{p}$  można wyznaczyć dystrybuantę oraz funkcję kwantylową zgodnie odpowiednio z (6.4) i (6.5):

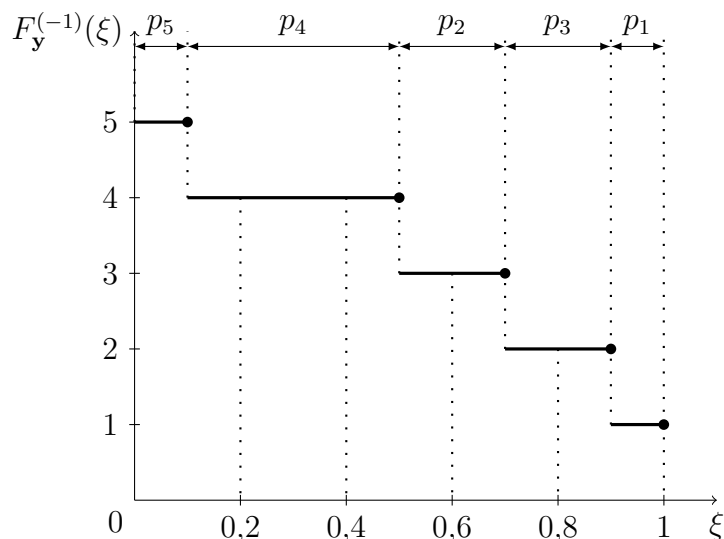
$$F_{\mathbf{y}}(d) = \begin{cases} 1 & \text{dla } d \leq 1, \\ 0,9 & \text{dla } 1 < d \leq 2, \\ 0,7 & \text{dla } 2 < d \leq 3, \\ 0,5 & \text{dla } 3 < d \leq 4, \\ 0,1 & \text{dla } 4 < d \leq 5, \\ 0 & \text{dla } d > 5, \end{cases} \quad F_{\mathbf{y}}^{(-1)}(\xi) = \begin{cases} 5 & \text{dla } 0 < \xi \leq 0,1, \\ 4 & \text{dla } 0,1 < \xi \leq 0,5, \\ 3 & \text{dla } 0,5 < \xi \leq 0,7, \\ 2 & \text{dla } 0,7 < \xi \leq 0,9, \\ 1 & \text{dla } 0,9 < \xi \leq 1. \end{cases}$$

Wykorzystując funkcję kwantylową, można teraz wyznaczyć średnie uporządkowanego wektora kosztów w ramach równych porcji zapotrzebowania  $1/\tilde{m}$ . W rozważanym problemie wymiary wektora wag preferencji i wektora ocen są równe ( $\tilde{m} = m = 5$ ). Poszukiwane średnie odpowiadają całkom  $\int_{(k-1)/5}^{k/5} F_{\mathbf{y}}^{(-1)}(\xi) d\xi$  dla  $k = 1, \dots, 5$ :

$$\begin{aligned} \int_0^{1/5} F_{\mathbf{y}}^{(-1)}(\xi) d\xi &= 0,1 \cdot 5 + 0,1 \cdot 4 = 0,9, \\ \int_{1/5}^{2/5} F_{\mathbf{y}}^{(-1)}(\xi) d\xi &= 0,2 \cdot 4 = 0,8, \\ \int_{2/5}^{3/5} F_{\mathbf{y}}^{(-1)}(\xi) d\xi &= 0,1 \cdot 4 + 0,1 \cdot 3 = 0,7, \\ \int_{3/5}^{4/5} F_{\mathbf{y}}^{(-1)}(\xi) d\xi &= 0,1 \cdot 3 + 0,1 \cdot 2 = 0,5, \\ \int_{4/5}^1 F_{\mathbf{y}}^{(-1)}(\xi) d\xi &= 0,1 \cdot 2 + 0,1 \cdot 1 = 0,3. \end{aligned}$$

Funkcja kwantylowa  $F_{\mathbf{y}}^{(-1)}(\xi)$ , na podstawie której wyznaczone są poszczególne całki, została przedstawiona na rysunku 6.2. Ostatecznie, zgodnie z (6.7), wartość operatora WOWA wynosi  $A_{\mathbf{w},\mathbf{p}}(\mathbf{y}) = 5 \cdot (0,4 \cdot 0,9 + 0,3 \cdot 0,8 + 0,15 \cdot 0,7 + 0,1 \cdot 0,5 + 0,05 \cdot 0,3) = 3,85$ .

W praktyce można sobie wyobrazić nieco inną procedurę wyznaczenia wartości operatora WOWA — zwłaszcza jeśli chce się ją policzyć ręcznie. Można zmodyfikować wektor kosztów w taki sposób, aby poszczególne oceny odpowiadały równym porcjom zapotrzebowania,



Rysunek 6.2. Funkcja kwantylowa  $F_{\mathbf{y}}^{(-1)}(\xi)$  do wyznaczenia wartości operatora WOWA z przykładu 6.2

których całkowita wielokrotność będzie równa  $1/\tilde{m}$ . Inaczej mówiąc, nowe porcje zapotrzebowania poszczególnych ocen powinny stanowić dzielnik wartości  $1/\tilde{m}$ . Taka modyfikacja jest możliwa, jeśli dla wszystkich wag zapotrzebowania  $p_i$  ( $i = 1, \dots, m$ ) i wartości  $1/\tilde{m}$  istnieje wspólny dzielnik. Warunkiem wystarczającym jego istnienia jest wymierność wag zapotrzebowania i wartości  $1/\tilde{m}$ . Biorąc pod uwagę charakter problemów lokalizacyjnych, założenie o wymierności poszczególnych wag wydaje się całkowicie uzasadnione. Omawiana modyfikacja prowadzi do zwielokrotnienia poszczególnych składowych wektora kosztów  $\mathbf{y}$  zgodnie z odpowiadającymi im wagami zapotrzebowania  $\mathbf{p}$ . Każda składowa  $y_i$  będzie zwielokrotniona tyle razy, ile ustalona porcja zapotrzebowania mieści się w  $p_i$  dla  $i = 1, \dots, m$ . Dla analizowanego problemu wektor kosztów można przekształcić tak, aby pojedyncza ocena odpowiadała 0,1 zapotrzebowania. W rezultacie otrzymuje się wektor  $\check{\mathbf{y}} = (1; 3; 3; 2; 2; 4; 4; 4; 4; 5)$  o 10 składowych. W stosunku do oryginalnego wektora kosztów druga i trzecia ocena zostały powtórzone dwukrotnie, a czwarta czterokrotnie. Oryginalne wagi preferencji  $w$  można teraz zastosować do wektora  $\check{\mathbf{y}}$  w taki sposób, że pierwsza waga będzie dotyczyć średniej dwóch największych ocen, druga waga będzie dotyczyć średniej dwóch kolejnych największych ocen itd. W ten sposób otrzymuje się  $A_{w,\mathbf{p}}(\mathbf{y}) = 0,4 \cdot 4,5 + 0,3 \cdot 4 + 0,15 \cdot 3,5 + 0,1 \cdot 2,5 + 0,05 \cdot 1,5 = 3,85$ .

Jak wspomniano na początku rozdziału, rozwiązania najlepsze w sensie rozkładu ocen dla problemów ze zróżnicowanymi wagami zapotrzebowania można również uzyskiwać z wykorzystaniem operatora OWA, przy czym konieczna jest do tego transformacja problemu. Charakter transformacji jest nieco zbliżony do drugiego podejścia wyznaczania wartości operatora WOWA w ramach przykładu 6.2. Ponieważ operator OWA nie pozwala na przypisanie różnych charakterystyk do poszczególnych kryteriów, problem musi być zdezagregowany do równych porcji zapotrzebowania. W sensie logicznym polega to na zwielokrotnieniu poszczególnych odbiorców w ramach swoich lokalizacji zgodnie z wagami zapotrzebowania  $\mathbf{p}$ . W praktyce

przekształcenie składa się z kilku kroków. Najpierw należy wyznaczyć porcję zapotrzebowania  $q$ , która będzie dzielnikiem wszystkich wag zapotrzebowania  $p_i$  ( $i = 1, \dots, m$ ). Dzielnik taki istnieje, gdy wszystkie wagi zapotrzebowania są wymierne, co wydaje się rozsądnym założeniem. Ze względów wydajnościowych porcja ta powinna być jak największa, aby wzrost rozmiaru problemu był jak najmniejszy. W najlepszym podejściu poszukiwana porcja zapotrzebowania  $q$  powinna więc być odpowiednikiem największego wspólnego dzielnika (NWD) wag  $p_i$ . Po wyznaczeniu tej porcji należy zwieliokrotnić poszczególnych odbiorców  $p_i/q$  razy dla  $i = 1, \dots, m$ , co wiąże się z przekształceniem macierzy kosztów obsługi. Macierz można naiwnie rozszerzyć w obu wymiarach. Jednak spowodowałoby to sztuczne zwiększenie liczby możliwych lokalizacji punktów obsługi, a w rezultacie jeszcze większy wzrost rozmiaru problemu. Ponieważ zdezagregowanie odpowiada wyodrębnieniu poszczególnych odbiorców w ramach pojedynczych lokalizacji, liczba możliwych lokalizacji nie ulega zmianie. Wystarczające jest zatem zwieliokrotnienie wierszy macierzy, które odpowiadają wektorom kosztów obsługi dla poszczególnych odbiorców. W zdezagregowanym problemie osiągalne wektory kosztów  $\check{y} \in A$  będą  $(1/q)$ -wymiarowe, więc należy jeszcze pamiętać o przekształceniu wektora wag preferencji  $w$ . Ten  $\tilde{m}$ -wymiarowy wektor trzeba również zdezagregować do wymiaru  $1/q$ . Nowe wagi  $\check{w}$  można wyznaczyć podobnie jak wagi  $\omega$  operatora WOWA we wzorze (6.2). Uwzględniając równe porcje zapotrzebowania  $q$ , wagi w problemie zdezagregowanym mogą być wyrażone jako

$$\check{w}_{i'} = w^*(i'q) - w^*((i' - 1)q) \quad \text{dla } i' = 1, \dots, \frac{1}{q},$$

gdzie  $w^*$  jest niemalejącą funkcją, która podobnie jak w klasycznej definicji operatora WOWA łączy punkty  $(\frac{i}{\tilde{m}}, \sum_{k \leq i} w_k)$  dla  $i = 1, \dots, \tilde{m}$  oraz punkt  $(0, 0)$ .

**Przykład 6.3.** W przykładzie zaprezentowane są ostatnie kroki procedury wyznaczania wartości agregacji WOWA za pomocą operatora OWA zastosowanego do zdezagregowanego problemu. Rozpatrzmy ten sam problem co w przykładach 6.1 i 6.2. Załóżmy również, że po pierwsze znaleziono wcześniej porcję zapotrzebowania  $q = 0,1$  będącą odpowiednikiem NWD wszystkich wag zapotrzebowania, a po drugie przekształcono macierz kosztów obsługi, zwieliokrotniając  $p_i/q$  razy wiersze dla  $i = 1, \dots, 5$ . W wyniku tego otrzymano wektor kosztów obsługi  $\check{y} = (1; 3; 3; 2; 2; 4; 4; 4; 4; 5)$  — taki jak w drugiej części przykładu 6.2. Należy teraz wyznaczyć wagi  $\check{w}_{i'}$  dla  $i' = 1, \dots, 10$ . W tym przypadku jest to szczególnie proste, gdyż  $1/\tilde{m} = 0,2$  jest wielokrotnością  $q = 0,1$ , więc wystarczy podzielić wagi na  $1/(\tilde{m}q)$  części, co daje w rezultacie  $\check{w} = (0,2; 0,2; 0,15; 0,15; 0,075; 0,075; 0,05; 0,05; 0,025; 0,025)$ . Ostateczna wartość agregacji to oczywiście  $0,2 \cdot 5 + 0,2 \cdot 4 + 0,15 \cdot 4 + 0,15 \cdot 4 + 0,075 \cdot 4 + 0,075 \cdot 3 + 0,05 \cdot 3 + 0,05 \cdot 2 + 0,025 \cdot 2 + 0,025 \cdot 1 = 3,85$ .

Jak widać, sama dezagregacja problemu wymaga dodatkowych obliczeń. Co gorsza, operacja taka może prowadzić do znacznego wzrostu liczby odbiorców, a tym samym rozmiaru



problemu, co w konsekwencji może uniemożliwić jego rozwiązanie. W przedstawionym przykładzie liczba klientów wzrosła tylko dwukrotnie, ale łatwo wyobrazić sobie przykłady, gdzie liczba ta wzrasta kilkadziesiąt razy, np. gdy waga zapotrzebowania jednego odbiorcy jest znacznie mniejsza od wag pozostałych odbiorców lub gdy wagi zapotrzebowania dwóch klientów stanowią odwrotności liczb pierwszych. Zastosowanie operatora WOWA pozwala wyeliminować trudności wynikające z transformacji problemu.

Operator WOWA był wykorzystywany w różnych zagadnieniach wielokryterialnego wspomaganie decyzji [87, 65, 69, 70], w tym przy agregacji metadanych [15, 55]. Według wiedzy autora nie był jednak badany w celu uzyskiwania rozwiązań optymalnych w sensie rozkładu wartości w problemach lokalizacyjnych ze zróżnicowanymi wagami zapotrzebowania.

## 7. Modele optymalizacyjne WOWA

Operator agregacji WOWA pozwala uzyskiwać rozwiązania zgodne z zasadą bezstronności, uwzględniając jednocześnie zróżnicowane wielkości zapotrzebowań bez konieczności dezagregowania problemu do pojedynczych porcji zapotrzebowania. W obszarze zagadnień lokalizacyjnych nie było do tej pory modeli, które pozwalałyby na uzyskiwanie rozwiązań optymalnych w sensie rozkładu ocen dla dowolnych wag preferencji z jednoczesnym bezpośrednim uwzględnieniem zróżnicowanych zapotrzebowań. W rozdziale tym przedstawiono modele optymalizacyjne agregacji WOWA dla dyskretnych problemów lokalizacyjnych. Przedstawiono zadanie PL agregacji WOWA dla nierosnących wag preferencji, a także zadanie PCLM dla dowolnego wektora wag preferencji. Oczywiście w obu tych przypadkach całe zadanie optymalizacyjne pozostaje zadaniem PCLM ze względu na dyskretny rzeczowy model lokalizacyjny. Zbadano efektywność obliczeniową zaproponowanych modeli, a także możliwość poprawy ich wydajności przez zastosowanie nadmiarowych ograniczeń. Eksperymenty przeprowadzono zarówno na małych, jak i dużych problemach lokalizacyjnych.

### 7.1. Model PL

Zastosowanie operatora WOWA do agregacji ocen w problemach optymalizacji wielokryterialnej prowadzi do zadania

$$\min\{A_{\mathbf{w},\mathbf{p}}(\mathbf{y}) : \mathbf{y} = \mathbf{f}(\mathbf{x}), \mathbf{x} \in Q\}. \quad (7.1)$$

Zastępując w równaniu (6.7) całki na przedziałach całkami lewostronnymi otrzyma się

$$A_{\mathbf{w},\mathbf{p}}(\mathbf{y}) = \sum_{k=1}^{\tilde{m}} \tilde{m}w_k(L(\mathbf{y}, \mathbf{p}, \frac{k}{\tilde{m}}) - L(\mathbf{y}, \mathbf{p}, \frac{k-1}{\tilde{m}})), \quad (7.2)$$

gdzie  $L(\mathbf{y}, \mathbf{p}, \alpha)$  jest wyznaczone przez lewostronną całkę  $F_{\mathbf{y}}^{(-1)}$

$$L(\mathbf{y}, \mathbf{p}, 0) = 0 \quad \text{oraz} \quad L(\mathbf{y}, \mathbf{p}, \alpha) = \int_0^\alpha F_{\mathbf{y}}^{(-1)}(\xi) d\xi \quad \text{dla } 0 < \alpha \leq 1. \quad (7.3)$$

W szczególności  $L(\mathbf{y}, \mathbf{p}, 1) = \int_0^1 F_{\mathbf{y}}^{(-1)}(\xi) d\xi = \sum_{i=1}^m p_i y_i$ , co stanowi średnią ważoną. Wykres funkcji  $L(\mathbf{y}, \mathbf{p}, \alpha)$  (ze względu na  $\alpha$ ) przyjmuje postać tzw. górnej bezwzględnej krzywej Lorenza. Zgodnie z równaniem (7.3) wartość funkcji  $L(\mathbf{y}, \mathbf{p}, \alpha)$  dla  $0 \leq \alpha \leq 1$  można

wyznaczyć w oparciu o zadanie optymalizacji

$$L(\mathbf{y}, \mathbf{p}, \alpha) = \max_{u_i} \left\{ \sum_{i=1}^m y_i u_i : \sum_{i=1}^m u_i = \alpha, \quad 0 \leq u_i \leq p_i \quad \forall i \right\}. \quad (7.4)$$

Przedstawione zadanie optymalizacji jest zadaniem PL dla ustalonego wektora ocen  $\mathbf{y}$ , ale dla wektora ocen jako zmiennych jest zadaniem nieliniowym. Można jednak zdefiniować zadanie dualne do zadania (7.4), które będzie zadaniem PL. W tym celu należy wprowadzić zmienną dualną  $t$  odpowiadającą ograniczeniu  $\sum_{i=1}^m u_i = \alpha$  oraz zmienne  $d_i$  odpowiadające górnym ograniczeniom zmiennych  $u_i$ . W wyniku otrzymuje się zadanie dualne w postaci

$$L(\mathbf{y}, \mathbf{p}, \alpha) = \min_{t, d_i} \left\{ \alpha t + \sum_{i=1}^m p_i d_i : t + d_i \geq y_i, \quad d_i \geq 0 \quad \forall i \right\} \quad (7.5a)$$

$$= \min_t \left\{ \alpha t + \sum_{i=1}^m p_i \max\{y_i - t, 0\} \right\}, \quad (7.5b)$$

gdzie optymalna wartość  $t^*$  jest  $\alpha$ -kwantylem rozkładu zmiennych  $y_i$  zgodnie z miarami  $p_i$ . Zadanie (7.5a) niesie ze sobą następujący wniosek.

**Twierdzenie 7.1.** Dla dowolnego wektora  $\mathbf{y}$  nierówność  $L(\mathbf{y}, \mathbf{p}, \alpha) \leq \varrho$  jest spełniona wtedy i tylko wtedy, gdy istnieją  $t$  i  $d_i$  ( $i = 1, \dots, m$ ) takie, że

$$\alpha t + \sum_{i=1}^m p_i d_i \leq \varrho \quad \text{oraz} \quad t + d_i \geq y_i, \quad d_i \geq 0 \quad \forall i.$$

Zgodnie z równaniem 7.2 standardowa agregacja WOWA ( $\tilde{m} = m$ ) przyjmuje postać

$$A_{\mathbf{w}, \mathbf{p}}(\mathbf{y}) = \sum_{k=1}^m m w_k \left( L(\mathbf{y}, \mathbf{p}, \frac{k}{m}) - L(\mathbf{y}, \mathbf{p}, \frac{k-1}{m}) \right) = \sum_{k=1}^m w'_k L(\mathbf{y}, \mathbf{p}, \frac{k}{m}), \quad (7.6)$$

gdzie  $w'_m = m w_m$ ,  $w'_k = m(w_k - w_{k+1})$ . Gdy wagi preferencji są nierosnące  $w_1 \geq w_2 \geq \dots \geq w_m$ , wagi  $w'_m$  w powyższym sformułowaniu agregacji WOWA są nieujemne. To pozwala sformułować następujący wniosek.

**Twierdzenie 7.2.** Zadanie minimalizacji WOWA (7.1) z nierosnącymi wagami preferencji  $w_1 \geq w_2 \geq \dots \geq w_m$  może być sformułowane jako zadanie PL z dodatkowymi ograniczeniami:

$$\begin{aligned} \min_{\varrho_k, t_k, d_{ik}, y_i} \quad & \sum_{k=1}^m w'_k \varrho_k \\ \text{p.o.} \quad & \frac{k}{m} t_k + \sum_{i=1}^m p_i d_{ik} \leq \varrho_k \quad \text{dla } k = 1, \dots, m, \\ & t_k + d_{ik} \geq y_i, \quad d_{ik} \geq 0 \quad \text{dla } i, k = 1, \dots, m, \\ & \mathbf{y} \in A. \end{aligned}$$

## 7.2. Modele PCLM

W ogólnym przypadku, gdy wagi preferencji  $w_i$  agregacji WOWA nie są monotoniczne, część współczynników  $w'_k$  w równaniu (7.6) może być ujemna. Z tego powodu minimalizacja górnej granicy  $\varrho_k$  nie jest wystarczająca. Dla ujemnych współczynników odpowiadające im funkcje Lorenza należy również ograniczyć od dołu.

Zgodnie z równaniem (7.5b), uwzględniając fakt, że wartość optymalna  $t^*$  odpowiada kwantylowi, a więc jednej z wartości  $y_i$ , nierówność  $L(\mathbf{y}, \mathbf{p}, \alpha) \geq \varrho$  jest prawdziwa wtedy i tylko wtedy, gdy

$$\varrho \leq \alpha y_{i'} + \sum_{i=1}^m p_i \max\{y_i - y_{i'}, 0\} \quad \text{dla } i' = 1, \dots, m.$$

**Twierdzenie 7.3.** Dla dowolnego wektora  $\mathbf{y}$  wartość  $\varrho$  spełnia nierówność  $L(\mathbf{y}, \mathbf{p}, \alpha) \geq \varrho$  wtedy i tylko wtedy, gdy istnieją  $z_{ii'}$  oraz  $\bar{d}_{ii'}$  ( $i', i = 1, \dots, m$ ) takie, że:

$$\begin{aligned} \varrho &\leq \alpha y_{i'} + \sum_{i=1}^m p_i \bar{d}_{ii'} && \text{dla } i' = 1, \dots, m, \\ \bar{d}_{ii'} &\leq y_i - y_{i'} + M z_{ii'} && \text{dla } i' \neq i = 1, \dots, m, \\ \bar{d}_{ii'} &\leq M(1 - z_{ii'}) && \text{dla } i' \neq i = 1, \dots, m, \\ \bar{d}_{ii} &= 0 && \text{dla } i = 1, \dots, m, \\ z_{ii'} &\in \{0, 1\} && \text{dla } i' \neq i = 1, \dots, m. \end{aligned}$$

Te obserwacje pozwalają na sformułowanie ogólnego zadania optymalizacji WOWA w postaci zadania PCLM.

**Twierdzenie 7.4.** Dowolne zadanie optymalizacji agregacji WOWA (7.1) może być sformułowane jako zadanie PCLM z dodatkowymi ograniczeniami i zmiennymi binarnymi:

$$\min_{\varrho_k, t_k, d_{ik}, y_i, \bar{d}_{ii'}, z_{ii'}} \sum_{k=1}^m w'_k \varrho_k \quad (7.7a)$$

$$\text{p.o. } \frac{k}{m} t_k + \sum_{i=1}^m p_i d_{ik} \leq \varrho_k \quad \text{dla } k = 1, \dots, m, \quad (7.7b)$$

$$t_k + d_{ik} \geq y_i, \quad d_{ik} \geq 0 \quad \text{dla } i, k = 1, \dots, m, \quad (7.7c)$$

$$\varrho_k \leq \frac{k}{m} y_{i'} + \sum_{i=1}^m p_i \bar{d}_{ii'} \quad \text{dla } i', k = 1, \dots, m, \quad (7.7d)$$

$$\bar{d}_{ii'} \leq y_i - y_{i'} + M z_{ii'} \quad \text{dla } i' \neq i = 1, \dots, m, \quad (7.7e)$$

$$\bar{d}_{ii'} \leq M(1 - z_{ii'}) \quad \text{dla } i' \neq i = 1, \dots, m, \quad (7.7f)$$

$$\bar{d}_{ii} = 0 \quad \text{dla } i = 1, \dots, m, \quad (7.7g)$$

$$z_{ii'} \in \{0, 1\} \quad \text{dla } i' \neq i = 1, \dots, m, \quad (7.7h)$$

$$\mathbf{y} \in A. \quad (7.7i)$$

Pełne sformułowanie ze wszystkimi ograniczeniami (7.7b)–(7.7i) stanowi poprawne ogólne zadanie optymalizacji agregacji WOWA. Nie ma jednak potrzeby używania zarówno górnych, jak i dolnych ograniczeń dla każdego  $k$ . Z zadania mogą zostać wyeliminowane górne ograniczenia (7.7b)–(7.7c) dla wag  $w'_k < 0$  oraz dolne ograniczenia (7.7d) dla wag  $w'_k \geq 0$ . Ograniczenia (7.7e)–(7.7h) mogą być pominięte tylko wtedy, gdy wszystkie wagi  $w'_k \geq 0$ . Model ten będzie dalej oznaczany jako MW1.

Zmienne binarne  $z_{ii'}$  z ograniczeniami (7.7e)–(7.7h) reprezentują porównanie parami wartości  $y_i$  i  $y_{i'}$ . Mówiąc precyzyjnie, gdy  $y_i < y_{i'}$  to  $z_{ii'} = 1$ , a w przeciwnym wypadku  $z_{ii'} = 0$ . Liczbę zmiennych binarnych i ograniczeń można zmniejszyć, korzystając z własności symetrii zmiennych  $\bar{d}_{ii'}$  i  $\bar{d}_{i'i}$ . Dzięki temu ograniczenia (7.7e)–(7.7h) można zastąpić następująco:

$$\bar{d}_{ii'} \leq y_i - y_{i'} + M z_{ii'} \quad \text{dla } i', i = 1, \dots, m; i < i', \quad (7.8a)$$

$$\bar{d}_{ii'} \leq M(1 - z_{ii'}) \quad \text{dla } i', i = 1, \dots, m; i < i', \quad (7.8b)$$

$$\bar{d}_{i'i} \leq y_{i'} - y_i + \bar{d}_{ii'} \quad \text{dla } i', i = 1, \dots, m; i < i', \quad (7.8c)$$

$$\bar{d}_{ii} = 0 \quad \text{dla } i = 1, \dots, m, \quad (7.8d)$$

$$z_{ii'} \in \{0, 1\} \quad \text{dla } i', i = 1, \dots, m; i < i'. \quad (7.8e)$$

Model ten będzie dalej oznaczany jako MW2.

Dla obu powyższych modeli rozpatrzone zostało nadmiarowe ograniczenie na nieujemne zmienne  $\bar{d}_{ii'}$ , czyli

$$\bar{d}_{ii'} \geq 0 \quad \text{dla } i, i' = 1, \dots, m. \quad (7.9)$$

W ramach testów obliczeniowych porównano cztery sformułowania:

- MW1<sub>1</sub> – sformułowanie (7.7a)–(7.7i),
- MW1<sub>2</sub> – sformułowanie (7.7a)–(7.7i) z nadmiarowym ograniczeniem (7.9),
- MW2<sub>1</sub> – sformułowanie (7.7a)–(7.7c), (7.7i), (7.8a)–(7.8e),
- MW2<sub>2</sub> – sformułowanie (7.7a)–(7.7c), (7.7i), (7.8a)–(7.8e) z nadmiarowym ograniczeniem (7.9).

### 7.3. Ograniczenia nadmiarowe

W celu poprawy wydajności obliczeniowej zaproponowanych zadań PCLM rozpatrzono dodanie nadmiarowych ograniczeń.

Zbadano ograniczenia wynikające z relacji przechodniości porównania ocen parami. Relacja przechodniości oznacza, że gdy  $y_i < y_{i'}$  i  $y_{i'} < y_{i''}$  to  $y_i < y_{i''}$ . Jest to równoważne ograniczeniu na zmienne binarne

$$z_{ii''} \geq z_{ii'} + z_{i'i''} - 1 \quad \text{dla } i, i', i'' = 1, \dots, m; i < i' < i''. \quad (7.10)$$

Ograniczenie (7.10) można traktować jako dolne ograniczenie na zmienne binarne wynikające z relacji przechodniości. Analogicznie można dodać ograniczenie górne, które odpowiada następującej zależności: jeśli  $y_i \geq y_{i'}$  i  $y_{i'} \geq y_{i''}$  to  $y_i \geq y_{i''}$ . Równoważne ograniczenie to

$$z_{ii''} \leq z_{ii'} + z_{i'i''} \quad \text{dla} \quad i, i', i'' = 1, \dots, m; i < i' < i''. \quad (7.11)$$

Należy jednak zwrócić uwagę, że relacja przechodniości generuje bardzo dużą liczbę ograniczeń, co może mieć z drugiej strony negatywny wpływ na efektywność obliczeniową modeli.

Rozważano również ograniczenia na wartość funkcji  $L(\mathbf{y}, \mathbf{p}, \alpha)$ . Jednym z nich jest ograniczenie na maksymalny przyrost wartości  $L(\mathbf{y}, \mathbf{p}, \alpha)$ . Funkcja  $L(\mathbf{y}, \mathbf{p}, k/m)$  dla danego  $k$  reprezentuje średnią ważoną  $k/m$  największych kosztów  $\mathbf{y}$  z rozkładem wag  $\mathbf{p}$ . Biorąc pod uwagę nierosnące uporządkowanie ocen, różnica między średnimi ważonymi  $(k+1)/m$  a  $k/m$  największych kosztów może być co najwyżej taka, jak między średnimi ważonymi  $k/m$  a  $(k-1)/m$  największych kosztów dla każdego  $k = 2, \dots, m-1$ . Uwzględniając przypadek graniczny dla  $k = 1$ , ograniczenia te wyrażają się następująco:

$$\begin{aligned} \rho_{k+1} &\leq 2\rho_k - \rho_{k-1} & \text{dla} \quad k = 2, \dots, m-1, \\ \rho_2 &\leq 2\rho_1. \end{aligned} \quad (7.12)$$

Wartości funkcji  $L(\mathbf{y}, \mathbf{p}, \alpha)$  można również dodatkowo ograniczyć od dołu. Ze względu na nierosnące uporządkowanie ocen wartość  $L(\mathbf{y}, \mathbf{p}, k/m)$  stanowi co najmniej  $k/m$ -tą część całości, czyli wartości funkcji  $L(\mathbf{y}, \mathbf{p}, 1)$ , co z kolei odpowiada średniej ważonej wszystkich ocen dla  $k = 1, \dots, m$ . Formalnie ograniczenie to oznacza

$$\rho_k \geq \frac{k}{m} \sum_{i=1}^m p_i y_i \quad \text{dla} \quad k = 1, \dots, m. \quad (7.13)$$

Powyższe ograniczenia będą dalej oznaczane jako:

- $c_1$  — ograniczenie (7.10),
- $c_2$  — ograniczenie (7.11),
- $c_3$  — ograniczenie (7.12),
- $c_4$  — ograniczenie (7.13).

## 7.4. Projekt eksperymentów

W celu sprawdzenia wydajności modeli WOWA przeprowadzono testy obliczeniowe podobne jak dla modeli OWA. Wykorzystano te same instancje problemów, wygenerowane jako kombinacje poszczególnych parametrów (patrz 5.6). Pojawił się także dodatkowy parametr  $\mathbf{p}$ , odpowiadający wektorowi wag zapotrzebowania.

Wagi zapotrzebowania wygenerowano zgodnie z rozkładem Zipfa, który kojarzony jest przede wszystkim z rozkładem częstości występowania słów w korpusach językowych [48].

W swojej pracy Zipf [96] rozkład ten odniósł także do wielkości populacji miast, a zależność w tym obszarze została również wcześniej zauważona przez Auerbacha [4]<sup>1</sup>. Tym samym wykorzystanie tego rozkładu w niniejszej pracy, dotyczącej problemów lokalizacyjnych, wydaje się w pełni uzasadnione. Rozkład Zipfa występuje także w innych zagadnieniach, np. w rozkładzie wielkości firm [5], które również mogą stanowić przedmiot praktycznych zastosowań problemów lokalizacyjnych. Chociaż ostatnie prace [20] pokazują, że na podstawie dokładniejszych danych (dokładniejsze uwzględnienie mniejszych miejscowości) rozkład populacji miast odpowiada raczej rozkładowi logarytmicznie normalnemu, to autorzy tych prac przyznają jednocześnie, że dla części największych miast pokrywa się on praktycznie z rozkładem Zipfa. Takie przybliżenie wydaje się zupełnie wystarczające na potrzeby niniejszej pracy.

W rozkładzie Zipfa wielkość obiektu (zjawiska) jest odwrotnie proporcjonalna do pozycji w rankingu tych obiektów według ich wielkości, gdzie na pierwszej pozycji znajduje się obiekt największy. Formalnie oznacza to  $p_i \sim 1/i^b$ , gdzie  $p_i$  oznacza wielkość obiektu znajdującego się na  $i$ -tej pozycji rankingu. Wykładnik  $b$  jest bardzo zbliżony do jedności i zazwyczaj dla uproszczenia przyjmuje się dokładnie  $b = 1$  (takie założenie przyjęto również w niniejszej pracy). Założono również, że lokalizacje odpowiadają kolejności w rankingu, a więc że ich kolejność odpowiada nierosnącemu (malejącemu) zapotrzebowaniu. Uwzględniając normalizację, wagi zapotrzebowania zostały więc wyznaczone jako

$$p_i = \frac{1}{i \sum_{i=1}^m \frac{1}{i}} \quad \text{dla } i = 1, \dots, m.$$

Procedura eksperymentalna została zaimplementowana w języku C++, a do rozwiązywania problemów wykorzystano pakiet IBM ILOG CPLEX Optimization Studio (w tym solver cplex) w wersji 12.4 [35]. Eksperymenty obliczeniowe zostały wykonane na maszynie z procesorem Intel Core2 Duo 2.53GHz (mobilny) i 3 GB pamięci RAM. Nałożono również limit czasowy 600 sekund jako maksymalny czas rozwiązywania pojedynczej instancji problemu.

## 7.5. Wyniki

Wydajność obliczeniowa modeli została porównana w oparciu o średnie czasy rozwiązania problemów w ramach poszczególnych typów, rozmiarów i liczby punktów obsługi, uśredniając po instancjach macierzy kosztów. Pełne wyniki znajdują się w dodatku B w tabelach B.1–B.4. Górny indeks przed czasem rozwiązania oznacza liczbę instancji problemów spośród 15, które przekroczyły dopuszczalny czasowy limit 600 sekund. Przypadki, gdy żadna z 15 instancji nie została rozwiązana w ramach dopuszczalnego limitu, zostały oznaczone przez „-”. W celu zobrazowania najważniejszych zależności poniższa analiza wyników prezentuje rezultaty w formie graficznej uśrednione dodatkowo po wariantach liczby punktów obsługi.

<sup>1</sup> Źródła za [20]

### 7.5.1. Problemy z nierosnącymi wagami preferencji

W przypadku problemów, w których wagi preferencji  $w_k$  ( $k = 1, \dots, m$ ) są nierosnące, co oznacza jednocześnie, że wszystkie wagi  $w'_k$  ( $k = 1, \dots, m$ ) są nieujemne, oba modele MW1 oraz MW2 sprowadzają się do podstawowego modelu liniowego MWLP.

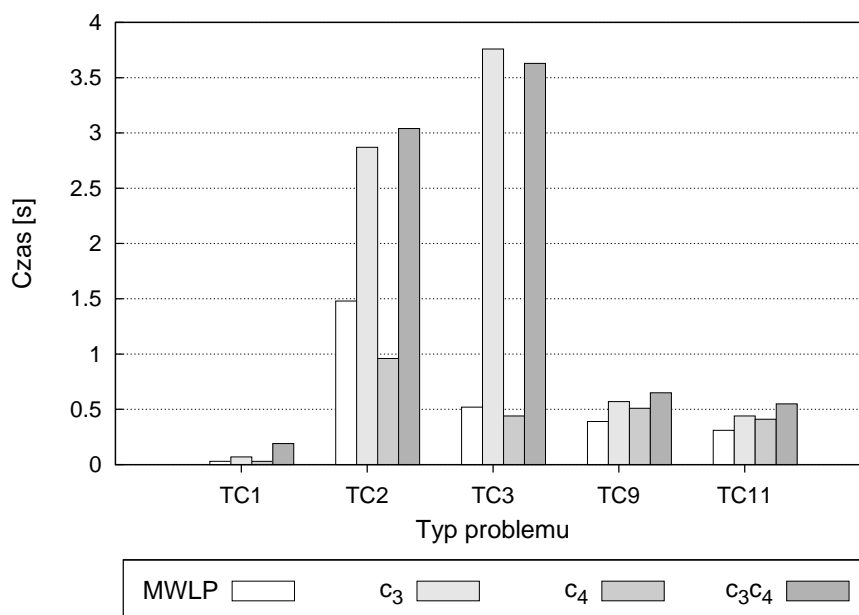
W ramach testów sprawdzono efektywność obliczeniową podstawowego modelu liniowego, a także zbadano wpływ nadmiarowych ograniczeń  $c_3$  i  $c_4$  na osiągnięte czasy rozwiązania. Szczegółowe wyniki modelu liniowego (od 8 do 30 lokalizacji) znajdują się w tabeli B.1, a na rysunku 7.1 przedstawiono uśrednione wyniki dla 30 lokalizacji.

Wyniki pokazują, że model MWLP dobrze radzi sobie z rozwiązywaniem problemów o rozmiarze do 30 lokalizacji. Maksymalne czasy sformułowania bez nadmiarowych ograniczeń są rzędu pojedynczych sekund, przy czym dotyczy to problemów z kryterium centrum. Dla pozostałych typów problemów z nierosnącymi wagami preferencji czasy są jeszcze krótsze, osiągając minimalne wartości dla kryterium średniej. Nadmiarowe ograniczenie  $c_3$  pogarsza czasy rozwiązania dla wszystkich typów rozpatrywanych problemów. Dla typów TC1–TC3 jest to znaczne pogorszenie, nawet kilkukrotne. Dotyczy to zarówno przypadków, gdy  $c_3$  jest jedynym nadmiarowym ograniczeniem, jak i sformułowania z dwoma nadmiarowymi ograniczeniami. Inna sytuacja jest w przypadku ograniczenia  $c_4$ . Dla typów TC1, TC9 i TC11 wpływ tego ograniczenia jest negatywny, ale w znacznie mniejszym stopniu niż ograniczenia  $c_3$ . Natomiast dla problemów typu TC2 i TC3 sformułowanie z tym ograniczeniem osiąga krótsze czasy niż sformułowanie podstawowe. Różnice czasowe są jednak niewielkie, zwłaszcza dla typu TC3.

Wyniki uzyskane dla problemów do 30 lokalizacji sugerują, że nadmiarowe ograniczenia w liniowym sformułowaniu WOVA nie pozwalają na znaczną poprawę wydajności, a czasem mogą ją zdecydowanie pogorszyć. Co prawda, ograniczenie  $c_4$  skróciło czas rozwiązania dla pewnych typów małych problemów, ale tylko w niewielkim stopniu. Ograniczenie  $c_4$  stanowi dolne ograniczenie na górną granicę funkcji  $L(\mathbf{y}, \mathbf{p}, \alpha)$ , a więc jego kierunek jest zgodny z ograniczeniem występującym w bazowym modelu liniowym agregacji WOVA. To prawdopodobnie sprawia, że ograniczenia  $c_4$  nie komplikuje znacząco struktury zbioru dopuszczalnego, a dla części problemów ma pozytywny wpływ na algorytm rozwiązywania. Z kolei wydaje się, że ograniczenia  $c_3$  utrudnia w większym stopniu strukturę zbioru dopuszczalnego, co prowadzi w konsekwencji do gorszych czasów rozwiązania.

Sprawdzono także jak wygląda sytuacja dla większych problemów. Zbadano wydajność modelu liniowego i wpływ nadmiarowego ograniczenia  $c_4$  dla problemów z biblioteki OR o rozmiarze 100 i 200 lokalizacji. Wyniki zostały przedstawione w tabeli 7.1.





Rysunek 7.1. Średni czas rozwiązania problemów z nierosnącymi wagami preferencji i ze zróżnicowanymi wagami zapotrzebowania dla 30 lokalizacji przez model MWLP z liniowym sformułowaniem agregacji WOVA

Tabela 7.1: Statystyki czasowe modelu PL agregacji WOVA dla problemów o rozmiarze 100 i 200 lokalizacji ze zróżnicowanymi wagami zapotrzebowania (wyniki sformułowania podstawowego MWLP oraz sformułowania z dodanym ograniczeniem nadmiarowym; przez „-” zostały oznaczone przypadki, gdy nie udało się rozwiązać problemu w ramach limitu 600 s)

typ	Problem			CPU[s]	
	nazwa	m	n	MWLP	c <sub>4</sub>
TC1	pmed1	100	5	0,56	0,63
	pmed2		10	0,29	0,37
	pmed3		10	0,34	0,42
	pmed4		20	0,22	0,33
	pmed5		33	0,22	0,29
	pmed6	200	5	17,31	24,16
	pmed7		10	2,29	2,82
	pmed8		20	1,93	2,68
	pmed9		40	1,72	2,5
	pmed10		67	1,38	2,17
TC2	pmed1	100	5	-	-
	pmed2		10	-	-
	pmed3		10	-	-
	pmed4		20	-	-
	pmed5		33	-	-
	pmed6	200	5	-	-
	pmed7		10	-	-

kontynuacja na następnej stronie

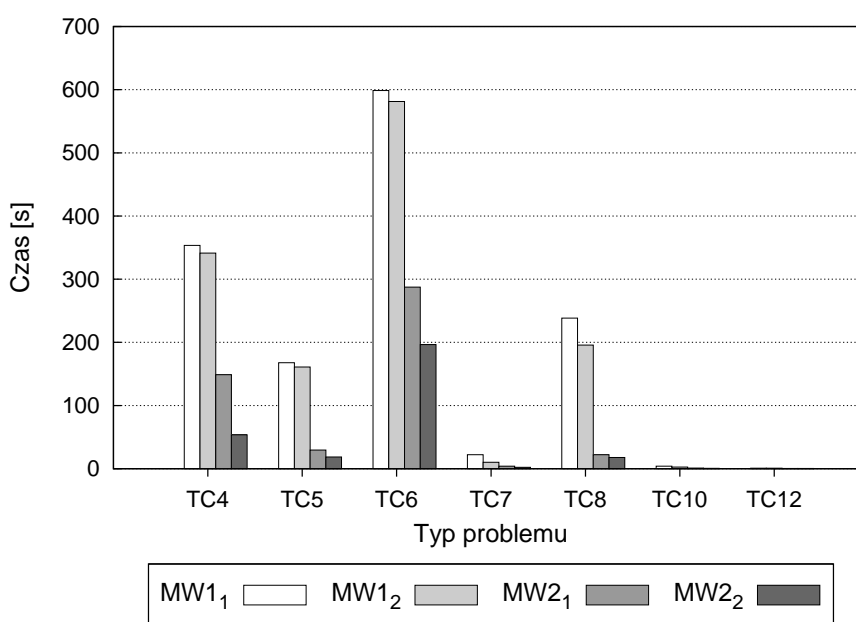
typ	Problem			CPU[s]	
	nazwa	m	n	MWLP	c <sub>4</sub>
TC3	pmed8		20	–	–
	pmed9		40	–	–
	pmed10		67	–	–
	pmed1	100	5	27,01	30,56
	pmed2		10	–	–
	pmed3		10	88,1	99,58
	pmed4		20	–	–
	pmed5		33	36,05	62,73
	pmed6	200	5	–	–
	pmed7		10	–	–
TC9	pmed8		20	–	–
	pmed9		40	–	–
	pmed10		67	274,44	255,76
	pmed1	100	5	24,48	38,28
	pmed2		10	81,17	100,87
	pmed3		10	35,03	44,78
	pmed4		20	35,37	53,77
	pmed5		33	4,06	16,87
	pmed6	200	5	–	–
	pmed7		10	–	–
TC11	pmed8		20	–	–
	pmed9		40	–	–
	pmed10		67	–	–
	pmed1	100	5	7,44	13,54
	pmed2		10	15,98	37,85
	pmed3		10	22,52	30,16
	pmed4		20	17,44	31,36
	pmed5		33	4,09	13,27
	pmed6	200	5	–	–
	pmed7		10	–	–
pmed8		20	–	–	
pmed9		40	–	–	
pmed10		67	576,76	–	

Widać, że poza kryterium średnim (TC1), udało się rozwiązać tylko nieliczne problemy o 200 lokalizacjach. W ramach problemów o 100 lokalizacjach najgorsze wyniki dotyczyły typów TC2 i TC3. Dla TC2 nie udało się rozwiązać żadnego problemu w ramach ustalonego limitu 600 s, a dla TC3 rozwiązano 3 z 5 problemów. To dla tych typów problemów nadmiarowe ograniczenie c<sub>4</sub> skróciło nieco czas rozwiązania części małych problemów (do 30 lokalizacji). Jednak w przypadku dużych problemów, które udało się rozwiązać, nadmiarowe ograniczenie wydłużyło czas rozwiązania nawet dla typu TC3. To pokazuje, że nadmiarowe ograniczenia w liniowym sformułowaniu WOVA zazwyczaj w większym stopniu utrudniają zadanie niż uściślają jego opis, co prowadzi do dłuższych czasów rozwiązania.

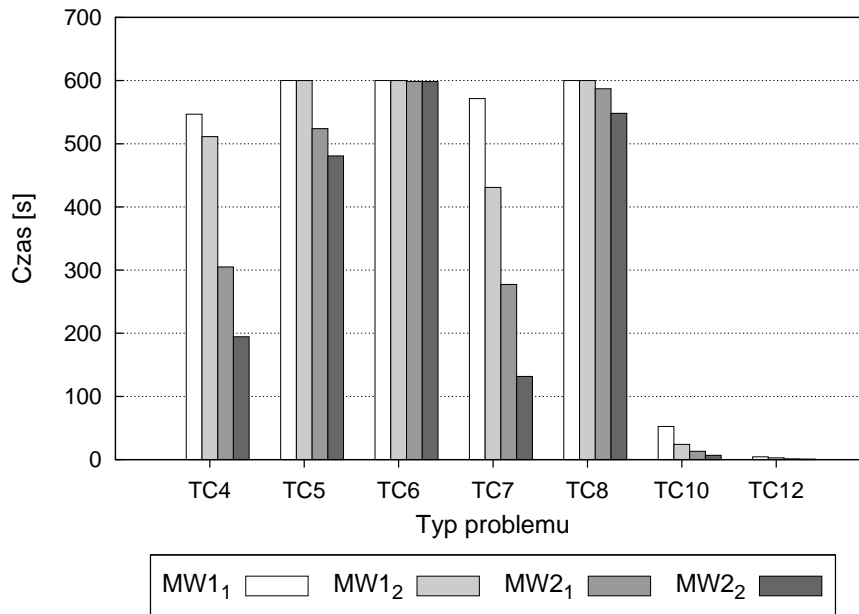
### 7.5.2. Problemy z niemonotonicznymi lub niemalejącymi wagami preferencji

Gdy wagi preferencji nie są odpowiednio monotoniczne (nierosnące), w sformułowaniu zadania konieczne są zmienne binarne i związane z nimi ograniczenia. Prowadzi to do modeli PCLM, których złożoność obliczeniowa jest znacznie większa niż modeli liniowych. Ich wydajność została sprawdzona na problemach z 8 i 10 lokalizacjami.

Wstępnie porównano ze sobą modele MW1 i MW2 oraz zbadano wpływ dodatkowego ograniczenia (7.9). Wyniki dla problemów ze zróżnicowanymi zapotrzebowaniami znajdują się w tabeli B.2 (w dodatku B). Wyniki dla 10 lokalizacji (uśrednione po wariantach liczby punktów obsługi) przedstawione są na rysunku 7.2. Widać, że model MW2 osiąga znacząco krótsze czasy rozwiązania niż model MW1. Wielkość różnic zależy od typu problemu, ale w wielu przypadkach przekraczają one rząd wielkości. Dotyczy to obu wersji modelu, tzn. z i bez ograniczenia (7.9). Samo ograniczenie również w zdecydowanej większości przypadków skraca czas rozwiązania, chociaż jego wpływ jest mniejszy i nie przekracza zazwyczaj jedynokrotności. Poza tym widać, że nawet przy tak małych problemach (8, 10 lokalizacji) czas rozwiązania jest długi. Dotyczy to zwłaszcza typów TC4–TC8. Model MW1 nie rozwiązał znacznej części problemów z 10 lokalizacjami w ramach ustalonego limitu czasu. Lepiej poradził sobie model MW2, chociaż on również nie rozwiązał wszystkich problemów, zwłaszcza z większą liczbą punktów obsługi. Pewnym pocieszeniem jest fakt, że z tych problemów tylko problemy typu TC4 wydają się mieć duże znaczenie praktyczne (ze względu na charakter wag preferencji). Ciekawy jest również fakt, że znacznie lepsze czasy rozwiązania dotyczą problemów typu TC10 i TC12, gdzie wagi preferencji są rosnące. Być może wynika to z faktu, że wszystkie wagi preferencji w tych problemach są niezerowe.



Rysunek 7.2. Średni czas rozwiązania problemów ze zróżnicowanymi wagami zapotrzebowania dla 10 lokalizacji przez modele MW1 i MW2

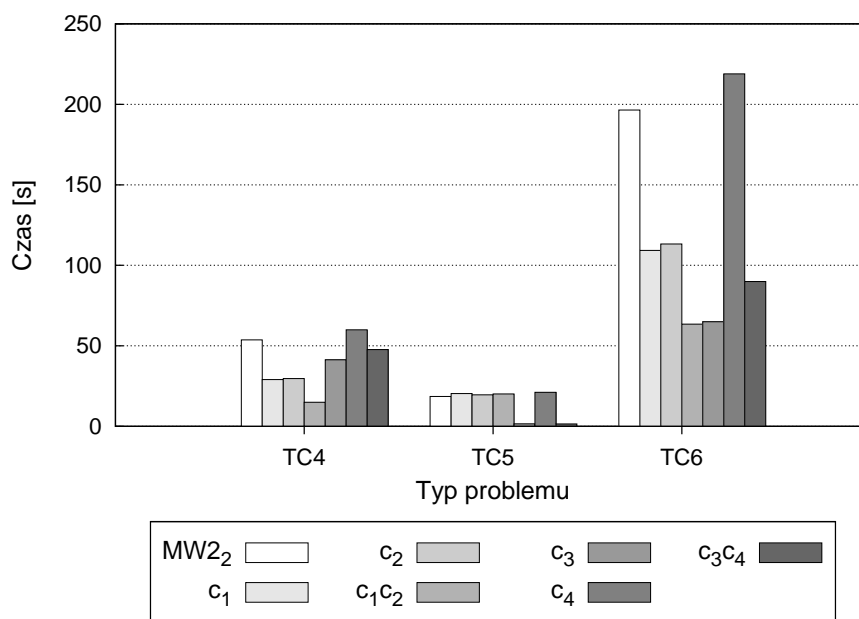


Rysunek 7.3. Średni czas rozwiązania problemów z równymi wagami zapotrzebowania dla 10 lokalizacji przez modele MW1 i MW2

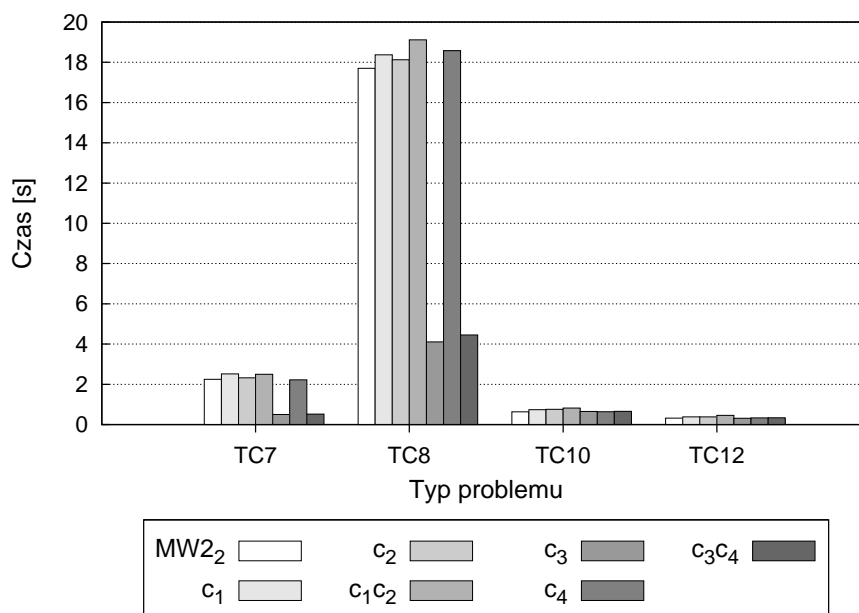
Dla porównania sprawdzono również wpływ rozkładu wartości wag zapotrzebowania na wydajność modeli. W tym celu rozwiązano te same problemy z równymi wagami zapotrzebowania, czyli takie, które można rozwiązać za pomocą modeli z agregacją OWA. Szczegółowe wyniki znajdują się w tabeli B.3, a na rysunku 7.3 przedstawiono uśrednione wyniki dla 10 lokalizacji. Widać, że czasy znacznie się pogorszyły, chociaż w największym stopniu dla problemów typu TC5–TC8.

W związku z niewielką wydajnością obliczeniową modeli PCLM agregacji WOVA sprawdzono możliwość jej poprawy przez dodanie ograniczeń nadmiarowych. Do tego celu wybrane zostało sformułowanie MW2<sub>2</sub>, które osiągało najlepsze czasy rozwiązania. Następnie dodawano do niego zaproponowane wcześniej ograniczenia nadmiarowe i za pomocą tak otrzymanych nowych sformułowań rozwiązywano problemy testowe. W sumie sprawdzono 6 sformułowań: 4 z pojedynczym dodatkowym ograniczeniem, 2 z dwoma dodatkowymi ograniczeniami. Szczegółowe wyniki znajdują się w tabeli B.4. Na rysunkach 7.4 i 7.5 przedstawiono wyniki dla 10 lokalizacji uśrednione dodatkowo po wariantach liczby punktów obsługi. Symbolami ograniczeń oznaczone zostały wyniki sformułowania MW2<sub>2</sub> z dodanymi tymi ograniczeniami.

Po pierwsze, można zauważyć, że problemy typu TC10 i TC12 (rosnące wagi) są rozwiązywane w krótkim czasie przez podstawowe sformułowanie MW2<sub>2</sub>, a wpływ ograniczeń nadmiarowych jest znikomy (przynajmniej dla tak małych problemów), poza pojedynczymi przypadkami wydłużenia czasu dla najmniejszego rozmiaru. Analizując dalej wpływ pojedynczych ograniczeń, pozytywnie wyróżnia się ograniczenie  $c_3$  na maksymalny przyrost wartości



Rysunek 7.4. Średni czas rozwiązania problemów typu TC4–TC6 ze zróżnicowanymi wagami zapotrzebowania dla 10 lokalizacji przez model MW<sub>2</sub><sub>2</sub> z nadmiarowymi ograniczeniami



Rysunek 7.5. Średni czas rozwiązania problemów typu TC7, TC8, TC10, TC12 ze zróżnicowanymi wagami zapotrzebowania dla 10 lokalizacji przez model MW<sub>2</sub><sub>2</sub> z nadmiarowymi ograniczeniami

funkcji  $L(\mathbf{y}, \mathbf{p}, \alpha)$ . Jest to szczególnie widoczne dla problemów typu TC5–TC8. Dodanie ograniczenia  $c_3$  skutkuje tutaj kilkukrotnym skróceniem średniego czasu rozwiązania w stosunku do wyjściowego sformułowania. Wyniki te są jednakowe dla różnej liczby punktów obsługi. Nieco inaczej wygląda sytuacja dla problemów typu TC4. Tutaj dodanie ograniczenia  $c_3$  poprawia średni czas rozwiązania tylko dla największej liczby punktów obsługi. Dla problemów typu TC4 nieco lepsze wyniki osiągają ograniczenia  $c_1$  i  $c_2$  wynikające z relacji przechodniości. Sformułowanie z ograniczeniem  $c_4$  osiąga czasy rozwiązania zbliżone do sformułowania bazowego MW2<sub>2</sub> dla wszystkich typów rozpatrywanych problemów. Rozważając sformułowania z dwoma nadmiarowymi ograniczeniami, widać, że tylko niewielka część problemów uzyskała lepsze czasy rozwiązania w porównaniu do sformułowań z pojedynczymi nadmiarowymi ograniczeniami. Jednoczesne dodanie ograniczeń  $c_1$  i  $c_2$  poprawia nieco czas rozwiązania w stosunku do pojedynczego ograniczenia dla problemów typu TC4 i TC6. Warto jednak zauważyć, że liczba ograniczeń  $c_1$  i  $c_2$  jest rzędu  $m^3$  (dokładna liczba ograniczeń każdego typu to  $\binom{m}{3}$ ), natomiast ograniczeń  $c_3$  rzędu  $m$ . To oznacza, że w miarę wzrostu rozmiaru problemu ograniczenia  $c_3$  powinny w mniejszym stopniu utrudniać zadanie niż ograniczenia  $c_1$  i  $c_2$ .

Przedstawione wyniki pokazują, że w przypadku modelu PCLM agregacji WOWA, inaczej niż w przypadku sformułowania liniowego, to ograniczenie  $c_3$  pozwala skrócić czas rozwiązania dla części problemów. W bazowym modelu PCLM agregacji WOWA wyznaczana jest zarówno dolna, jak i górna granica funkcji  $L(\mathbf{y}, \mathbf{p}, \alpha)$ . W tym przypadku to jednak granica dolna bardziej utrudnia model, gdyż jest związana ze zmiennymi binarnymi. W związku z tym można przypuszczać, że ograniczenie  $c_3$ , będąc górnym ograniczeniem granicy dolnej, daje możliwość ściślejszego opisu istotnego obszaru zbioru rozwiązań dopuszczalnych, a tym samym większej poprawy obliczeniowej efektywności modelu.

## 7.6. Wnioski

W rozdziale badano modele optymalizacyjne dla dyskretnych wielokryterialnych problemów lokalizacyjnych z operatorem agregacji WOWA. Modele takie pozwalają uzyskiwać rozwiązania optymalne w sensie rozkładu ocen, uwzględniając przy tym zróżnicowane wielkości zapotrzebowania. Dla nierosnących wag preferencji agregację WOWA sformułowano w postaci zadania PL. Sformułowanie to jest analogiczne do liniowego modelu OWA i bazuje na funkcji  $L(\mathbf{y}, \mathbf{p}, \alpha)$ , która wyraża średnią ważoną największych kosztów obsługi w ramach ustalonej porcji zapotrzebowania  $\alpha$ . W ogólności, gdy warunek monotoniczności wag nie jest spełniony, zaproponowano model rozszerzony. Wymagało to dodania zmiennych binarnych i odpowiednich ograniczeń, co doprowadziło do zadań PCLM agregacji WOWA, które mogą być stosowane dla dowolnych wag preferencji. Ta uniwersalność jest jednak okupiona znacznym wzrostem złożoności obliczeniowej, co zdecydowanie ogranicza maksymalne rozmiary problemów, które mogą być rozwiązywane za pomocą tych modeli. Sprawdzone możliwości

poprawy wydajności obliczeniowej modeli WOWA poprzez dodanie nadmiarowych ograniczeń. Przeprowadzono testy obliczeniowe, aby zbadać wydajność poszczególnych sformułowań, a także wpływ ograniczeń nadmiarowych.

Modele z liniowym sformułowaniem agregacji WOWA bardzo dobrze radzą sobie z niewielkimi problemami (do 30 lokalizacji), które rozwiązują w czasie od kilku setnych do kilku sekund. Dla dużych rozmiarów (rzędu 100 lokalizacji) zadanie PL nieźle radzi sobie z problemami o wszystkich niezerowych monotonicznych wagach (TC9, TC11) i bardzo dobrze z kryterium średniej (TC1). Znacznie gorzej jest z problemami typu TC2 i TC3 — przy 100 lokalizacjach udało się rozwiązać 3 z 5 problemów TC3 i żadnego problemu typu TC2. Badane ograniczenia nadmiarowe dla liniowego sformułowania WOWA nie poprawiają w ogólności wydajności obliczeniowej. Co prawda, jedno ograniczenie skróciło w niewielkim stopniu czasy rozwiązania dla części małych problemów, jednak dla analogicznych typów problemów o dużych rozmiarach przekroczono limit czasowy lub uzyskano dłuższe czasy rozwiązania niż czasy sformułowania bez nadmiarowego ograniczenia. Wyniki te są analogiczne do rezultatów uzyskanych w przypadku modeli z agregacją OWA z rozdziału 5, gdzie nadmiarowe ograniczenia również nie poprawiały wydajności obliczeniowej liniowych sformułowań agregacji.

Porównanie dwóch zadań PCLM agregacji WOWA dla dowolnych wag preferencji pokazuje lepszą wydajność modelu MW2 z ograniczoną liczbą zmiennych binarnych. Jednak nawet ten lepszy model przekroczył limit czasowy 600 s dla części problemów o rozmiarze zaledwie 10 lokalizacji. Niektóre z zaproponowanych ograniczeń nadmiarowych pozwoliły na kilkukrotne skrócenie czasu rozwiązań i uzyskanie rozwiązań optymalnych dla wszystkich problemów z 10 lokalizacjami. Niemniej jednak niemożliwe było rozwiązanie wszystkich problemów dla nieco większego rozmiaru. To pokazuje, że w przypadku niemonotonicznych wag preferencji modele optymalizacyjne z operatorem WOWA pozwalają na rozwiązanie problemów o maksymalnie kilkunastu lokalizacjach.

## 8. Metoda przybliżona

Ze względu na złożoność omawianych problemów lokalizacyjnych dokładne algorytmy rozwiązywania mogą być stosowane tylko dla niewielkich instancji, których rozmiary odbiegają zdecydowanie od problemów rzeczywistych. Aby rozwiązać problemy o dużych rozmiarach, stosuje się metody przybliżone, które chociaż nie gwarantują uzyskania rozwiązania optymalnego, to jednak dostarczają rezultat przybliżony w akceptowalnym czasie. Stanowią więc pewnego rodzaju kompromis pomiędzy jakością a efektywnością w sensie wymaganego czasu obliczeń. Heurystyki są również przydatne przy projektowaniu algorytmów dokładnych, gdyż dostarczają ograniczeń (w naszym przypadku ograniczenia górnego), które pozwalają na zawężenie przestrzeni rozwiązań przy poszukiwaniu rozwiązania optymalnego. W pracy badana była metoda bazująca na metaheurystyce przeszukiwania zmiennego sąsiedztwa (ang. Variable Neighbourhood Search, VNS).

### 8.1. Metaheurystyka przeszukiwania zmiennego sąsiedztwa (VNS)

Metaheurystyka VNS została pierwotnie przedstawiona przez Hansena i Mladenovica [52]. W oparciu o nią autorzy zaproponowali metodę przybliżoną rozwiązywania problemu lokalizacyjnego z kryterium średniej [31] oraz z kryterium centrum [53]. Metoda VNS była także stosowana do rozwiązywania problemów lokalizacyjnych z kryterium średniej z ograniczonymi pojemnościami punktów obsługi [24]. VNS określa sposób przeszukiwania przestrzeni rozwiązań, starając się przy tym uciekać z optimów lokalnych. Podejście to opiera się na badaniu tzw. sąsiedztw za pomocą algorytmu przeszukiwania lokalnego. Odbywa się to w sposób systematyczny, gdzie algorytm pozostaje w tym samym rozwiązaniu, dopóki nie znajdzie lepszego, po czym przenosi się do tego nowego rozwiązania. W ten sposób rozwiązanie jest stopniowo polepszane w miarę działania algorytmu.

W sposób formalny metoda może być przedstawiona następująco. Niech  $\gamma$  oznacza rozwiązanie problemu lokalizacyjnego, czyli podzbiór  $n$  lokalizacji, gdzie zostaną umieszczone punkty obsługi, spośród zbioru  $I$  wszystkich potencjalnych lokalizacji. Wówczas całą przestrzeń rozwiązań problemu można wyrazić jako  $\Gamma = \{\gamma : \gamma \subset I, |\gamma| = n\}$ . Jej licznosc wynosi  $\binom{m}{n}$ . Odległość między dwoma dowolnymi rozwiązaniami definiuje się jako liczbę punktów obsługi umiejscowionych w innych lokalizacjach. Na tej podstawie można zdefiniować symetryczną funkcję odległości jako

$$\rho(\gamma_1, \gamma_2) = |\gamma_1 \setminus \gamma_2| = |\gamma_2 \setminus \gamma_1|, \forall \gamma_1, \gamma_2 \in \Gamma.$$



W oparciu o powyższe można zdefiniować struktury sąsiedztw  $\mathcal{N}_r$  dla  $r = 1, \dots, r_{max}$ , gdzie  $r_{max} \leq n$ , a dalej sąsiedztwo  $\mathcal{N}_r(\gamma)$  jako zbiór rozwiązań oddalonych od rozwiązania  $\gamma$  o  $r$ . Formalnie wyraża się to jako

$$\gamma' \in \mathcal{N}_r(\gamma) \Leftrightarrow \rho(\gamma, \gamma') = r.$$

Sąsiedztwa charakteryzują się pewnymi własnościami:

- liczebność — zmiana rozwiązania o  $r$  wymaga usunięcia  $r$  aktualnych lokalizacji oraz wybranie  $r$  nowych na ich miejsce, czyli

$$|\mathcal{N}_r(\gamma)| = \binom{n}{r} \binom{m-n}{r}.$$

- rozłączność — pozwala na systematyczne przeglądanie sąsiedztw bez powielania sprawdzania tych samych rozwiązań

$$\mathcal{N}_{r_1}(\gamma) \cap \mathcal{N}_{r_2}(\gamma) = \emptyset, \forall r_1, r_2, r_1 \neq r_2.$$

- zupełność — zabezpiecza przed pominięciem któregoś z rozwiązań

$$\gamma \cup \bigcup_{r=1}^n \mathcal{N}_r(\gamma) = \Gamma.$$

Metoda VNS bazuje na algorytmie wymiany, który ocenia skutki zmiany lokalizacji wybranych punktów obsługi. Startując z rozwiązania początkowego, heurystyka wybiera w sposób losowy rozwiązanie z danego sąsiedztwa, a następnie sprawdza wszystkie możliwości wymiany pojedynczego punktu obsługi i realizuje najlepszą z nich, o ile polepszy to wybrane wcześniej rozwiązanie. Powtarzając ten proces znajduje optimum lokalne, które zapamiętuje jako nowe rozwiązanie, jeśli jest lepsze niż aktualne. Algorytm 1 przedstawia ogólny schemat metody VNS.

Rozwiązanie początkowe może zostać wyznaczone w sposób losowy lub zachłanny. Przy podejściu zachłannym lokalizowane są kolejne punkty obsługi (od pierwszego do  $n$ -tego), które dają najlepsze rozwiązanie, przy założeniu, że wcześniej umieszczone punkty obsługi nie są już przemieszczane.

### 8.1.1. VNS dla kryterium porządkowego

Na podstawie [31] została opracowana przez Domínguez metoda VNS do problemów z funkcją celu bazującą na uporządkowanej średniej ważonej [17]. Podstawową trudność w metodzie przedstawionej w [17] w porównaniu do jej bazowej wersji z [31] stanowi wyznaczenie zmiany wartości funkcji celu, opartej na uporządkowanej średniej ważonej, po dokonaniu wymiany punktów obsługi. W przypadku problemu z kryterium średnim nowa

---

**Algorytm 1.** Ogólny schemat metody VNS

---

```
1: Inicjalizacja
2:   Znajdź rozwiązanie początkowe  $\gamma_{opt}$ 
3: Krok główny
4:    $r \leftarrow 1$ 
5:   while ( $r \leq r_{max}$ ) i (opcjonalny dodatkowy warunek stopu nie jest spełniony) do
6:     Losowanie rozwiązania
7:     Wylosuj rozwiązanie  $\gamma_{cur}$  z  $r$ -tego sąsiedztwa  $\mathcal{N}_r(\gamma_{opt})$ 
8:     Przeszukiwanie lokalne
9:     Poprawiaj rozwiązanie  $\gamma_{cur}$ , wymieniając pojedynczy punkt obsługi, aż osiągniesz
        minimum lokalne
10:    Decyzja o zmianie rozwiązania
11:    if  $\gamma_{cur}$  lepsze niż  $\gamma_{opt}$  then
12:       $\gamma_{opt} \leftarrow \gamma_{cur}$ 
13:       $r \leftarrow 1$ 
14:    else
15:       $r \leftarrow r + 1$ 
16:    end if
17:  end while
```

---

wartość może być wyliczona niewielkim nakładem obliczeń na podstawie wartości przed zmianą. W problemie z operatorem porządkowym, ze względu na proces sortowania, pojedyncza wymiana punktów obsługi może skutkować jednak znacznie większymi zmianami w (posortowanym) wektorze kosztów, na podstawie którego wyznaczana jest wartość funkcji celu. W związku z tym w metodzie z [17] aktualizowany i sortowany jest cały wektor kosztów, co w konsekwencji skutkuje znacznie większą złożonością obliczeniową metody.

Podstawowym elementem przedstawianej metody, który decyduje również o jej złożoności, jest algorytm wymiany stanowiący jednocześnie przeszukiwanie lokalne. Algorytm bazuje na heurystyce zaproponowanej przez Whitakera [89] i został zaimplementowany między innymi w [31], [77]. Składają się na niego dwa podstawowe kroki. Pierwszy z nich to procedura *przejście*, która dla danego dodawanego nowego punktu obsługi znajduje najlepszy do usunięcia dotychczasowy punkt obsługi. Drugi krok to *aktualizacja* wektora kosztów oraz najbliższego i drugiego najbliższego punktu obsługi dla każdego klienta. Aby znaleźć najlepszy do usunięcia punkt obsługi, w procedurze *przejście* wyznaczana jest wartość funkcji celu dla każdej możliwej wymiany pojedynczego punktu obsługi. W związku z tym złożoność tej procedury dla funkcji opartej na uporządkowanej średniej ważonej wynosi  $O(nm \log m)$  w porównaniu do złożoności  $O(m)$  dla problemu z kryterium średnim [31]. To decyduje również o złożoności całego algorytmu wymiany, którego złożoność wynika z  $(m - n)$ -krotnego wywołania procedury *wycena przejścia* i jest równa  $O(nm^2 \log m)$  dla problemu z kryterium porządkowym oraz  $O(m^2)$  dla kryterium średniej.

Jak widać złożoność algorytmu dla problemów z kryterium uporządkowanej średniej jest znacznie większa niż metody dedykowanej dla problemów z kryterium średniej. Z tego powodu czasy rozwiązania dla problemów o dużych rozmiarach (rzędu kilkuset) znacznie wzrastają.

Można również zauważyć, że czas rozwiązania szybko rośnie wraz ze wzrostem liczby punktów obsługi do rozmieszczenia. Z jednej strony wynika to z faktu wzrostu liczności całego zbioru rozwiązań dopuszczalnych, którego moc wynosi  $\binom{m}{n}$  i rośnie wraz z  $n$  aż do  $n = m/2$ . Z drugiej strony większa liczba punktów obsługi oznacza wzrost liczności najbliższego sąsiedztwa (różniącego się lokalizacją jednego punktu obsługi), co przekłada się bezpośrednio na liczbę operacji w algorytmie wymiany. W metodzie przedstawionej w [17] sprawdzane są wszystkie możliwe wymiany, a więc w każdym kroku rozpatrywanych jest  $(m - n)$  potencjalnych lokalizacji do dodania punktu obsługi oraz  $n$  potencjalnych lokalizacji do usunięcia punktu obsługi, co daje w rezultacie  $(m - n)n$  wymian do sprawdzenia (bezpośrednich/najbliższych sąsiadów do odwiedzenia). Pomysłem na poprawienie metody, który wymieniany był w [17] jako kierunek przyszłych badań, było opracowanie algorytmu wymiany, który nie odwiedzałby bezpośrednio wszystkich możliwych najbliższych sąsiadów. Poza tym w pracy starano się również ograniczyć i uprościć procedurę wyznaczania wartości funkcji celu, która wymaga kosztownego procesu sortowania wektora kosztów. Aby zrealizować te cele opracowano trzy modyfikacje metody, które zostaną przedstawione poniżej.

Oprócz zmian mających na celu przyspieszenie metody, dokonano również modyfikacji pod kątem poprawy jakości uzyskiwanych rozwiązań. Rozwiązania uzyskiwane metodą z literatury cechowały się słabą jakością dla problemów z kryterium centrum. Zaproponowane zmiany miały na celu polepszenie jakości dla tego typu problemów i ewentualnie jemu zbliżonych, nie pogarszając przy tym dobrej jakości rozwiązań dla problemów pozostałych typów.

## 8.2. Modyfikacje algorytmu wymiany

W celu poprawy efektywności algorytmu wymiany zostały opracowane warunki, które pozwalają na ograniczenie przeszukiwania w nieobiecujących kierunkach oraz ograniczają nakład obliczeń przy ocenie rozwiązania. Aby natomiast poprawić jakość metody, dodano warunek regularyzacyjny przy porównywaniu rozwiązań.

### 8.2.1. Relaksacja problemu

Relaksacja problemu miała na celu ograniczenie liczby sprawdzanych sąsiednich rozwiązań. Chodziło o to, aby na jak najwcześniejszym etapie odrzucić nieperspektywiczne rozwiązania bez konieczności ich dokładnego wyznaczenia. Przez nieperspektywiczne rozumiemy takie rozwiązania, które nie będą lepsze od najlepszego aktualnie znanego rozwiązania.

Jak wspomniano wcześniej, w algorytmie wymiany sprawdzane są skutki zastąpienia jednego punktu obsługi przez nowo dodawany punkt obsługi, przy czym najpierw ustala się punkt dodawany, a następnie na tej podstawie wybiera się punkt do usunięcia. Jak łatwo zauważyć, przy takim podejściu dodawany punkt obsługi wpływa na wszystkie rozwiązania, które można otrzymać przy tej wymianie. W szczególności może się okazać, że nowy punkt obsługi nie jest w stanie poprawić aktualnego najlepszego rozwiązania. W związku z tym

warto takie przypadki wykrywać i odpowiednio na nie reagować. W tym celu proces wymiany podzielony został na dwa etapy: dodanie nowego punktu obsługi i usunięcie aktualnego punktu obsługi, a pomiędzy nimi dodany został dodatkowy warunek sprawdzający perspektywiczność potencjalnych nowych rozwiązań.

Po dodaniu nowego punktu obsługi (przed usunięciem) wyznaczana jest nowa wartość funkcji celu (OWA). Jeśli wartość ta nie jest mniejsza od wartości aktualnego rozwiązania, to wówczas nie ma sensu szukać punktu obsługi do usunięcia, bo wartość funkcji celu może się już tylko pogorszyć. Przy sprawdzaniu pierwszego sąsiada (a w ogólności do momentu aż nie znajdzie się lepszego sąsiada) nowa wartość funkcji celu może być tylko równa lub mniejsza od pierwotnej (bo dodanie nowego punktu obsługi nie może doprowadzić do gorszego rozwiązania). Jednak po znalezieniu lepszego sąsiada, odpowiadająca mu wartość funkcji celu zastąpi dotychczasową wartość graniczną, a przez to pozwoli na eliminację większej liczby nieobiecujących rozwiązań.

Mechanizm tej modyfikacji jest zbliżony do mechanizmu metody podziału i ograniczeń. Dodanie nowego punktu obsługi jest w rzeczywistości relaksacją pierwotnego problemu. Wyznaczona wartość funkcji celu stanowi więc granicę dolną. Oczywiście granica ta ma charakter lokalny i odnosi się do rozwiązań, które mogą zostać uzyskane przez usunięcie jednego (nadmiarowego w stosunku do ograniczeń) punktu obsługi. Jeśli więc wartość tej granicy jest większa od aktualnie najlepszego rozwiązania, które można potraktować jako granicę górną, to przeszukiwanie rozpatrywanego zbioru nie ma sensu, gdyż nie znajdzie się rozwiązania lepszego od aktualnego. Dodatkowo każde znalezione lepsze rozwiązanie będzie poprawiało (zmniejszało) granicę górną, a przez to lepiej ograniczało badane relaksacje.

Sprawdzenie takie jest bardziej korzystne (eliminuje większą liczbę nieperspektywicznych rozwiązań) dla problemów w wektorem wag z wieloma wagami równymi 0, gdyż po dodaniu nowego punktu wartość funkcji celu poprawi się tym bardziej, im więcej ocen z niezerowymi wagami ulegnie poprawie. W przypadku np. problemu z kryterium centrum rozwiązanie będzie miało szansę być lepsze wtedy i tylko wtedy, gdy po dodaniu nowego punktu obsługi polepszy się ocena najgorsza.

Modyfikacja ta wprowadza dodatkowe wyznaczenie wartości funkcji celu (a więc również uporządkowanego wektora kosztów), które będzie wykonywane po dodaniu nowego punktu obsługi. Z drugiej jednak strony, wyznaczony w ten sposób wektor będzie mógł być wykorzystany przy szukaniu najlepszego do usunięcia punktu obsługi. Wektor ten będzie już uwzględniał zmiany kosztów związane z dodanym punktem obsługi, a przez to mniejsza liczba składowych wektora kosztów będzie ulegać zmianie przy badaniu każdego potencjalnego punktu obsługi do usunięcia. Uwzględniając kolejne modyfikacje, zmniejszy to nakład obliczeń potrzebny do wyceny poszczególnych wymian.

## 8.2.2. Modyfikacja sortowania

Proponowany w [17] algorytm wymiany za każdym razem sortuje cały wektor kosztów w celu wyznaczenia wartości funkcji celu. Nie jest to efektywne, gdyż w wielu przypadkach zmianie ulega tylko niewielka część składowych wektora. W skrajnym przypadku przy pojedynczej wymianie mogą zmienić się tylko dwie składowe wektora kosztów (przy założeniu FSS/SSS — free/smallest self-service; przy braku tego założenia wektor kosztów mógłby nawet pozostać niezmienny). Zmodyfikowane sortowanie pozwala na wyznaczenie uporządkowanego wektora kosztów mniejszym nakładem obliczeń, wykorzystując do tego informacje z aktualnego rozwiązania.

Propozycja polega na zastąpieniu „nadmiarowego” sortowania przez pojedynczy krok sortowania przez scalanie. Warto zauważyć, że koszty, które nie uległy zmianie przy rozpatrywanej wymianie, zostały już uporządkowane przy wyznaczeniu wartości funkcji celu aktualnego rozwiązania. Można więc je zapamiętać, aby nie było konieczne ich ponowne sortowanie. Należy jedynie zapamiętać składowe wektora, które uległy zmianie, aby ich nie uwzględniać w nowym rozwiązaniu. W tym celu potrzebna będzie również permutacja odwzorowująca uporządkowany wektor kosztów na oryginalny wektor kosztów. Dodatkowo trzeba oczywiście zapamiętać nowe koszty, a następnie uporządkować je zgodnie z porządkiem całego wektora (czyli nierosnąco). To co pozostaje, to scalić te dwa wektory, zachowując porządek i pamiętając o nieuwzględnianiu starych kosztów.

Zmniejszenie nakładu obliczeń zależy od liczby składowych wektora kosztów, które ulegają zmianie przy zadanej wymianie, ponieważ determinują one wielkość wektora, który będzie podlegał sortowaniu. Można więc rozważyć, ile kosztów (przypisań) się zmienia przy wymianie jednego punktu obsługi. Dodanie punktu obsługi powoduje średnio  $m/(n+1)$  zmian, bo średnio tyle lokalizacji przypada na punkt obsługi po dodaniu jednego dodatkowego i tyle średnio zostanie przypisanych do nowego punktu obsługi. Analogicznie usunięcie również powoduje zmianę  $m/(n+1)$  składowych wektora kosztów, bo tyle lokalizacji trzeba będzie przypisać do pozostałych punktów obsługi. W związku z tym, gdy zmiany wynikające z dodawania i usuwania punktu obsługi są rozpatrywane oddzielnie, wymiar wektora, który podlega sortowaniu, wynosi średnio  $m/(n+1)$ . Gdy zmiany dotyczą całej wymiany (jednoczesnego dodania i usunięcia punktu obsługi), to wymiar wektora, który podlega sortowaniu, wynosi średnio  $2m/(n+1)$  (przy założeniu, że koszty przy dodawaniu i usuwaniu zmieniają się niezależnie; w przypadku gdy punkt dodawany i usuwany znajdują się blisko siebie, wymiar wektora może być mniejszy, gdyż część klientów przypisanych wcześniej do usuwanego punktu obsługi zostanie przydzielona do dodawanego punktu obsługi). Z kolei operacja scalania ma złożoność liniową proporcjonalną do sumy wymiarów dwóch scalanych wektorów: wektora kosztów i wektora kosztów zmienionych. Średnia liczba operacji jest więc proporcjonalna do  $m + m/(n+1)$  lub  $m + 2m/(n+1)$ , czyli jej złożoność wynosi  $O(m)$ .

Bezpośrednio z powyższej analizy wynika również, że zmniejszenie nakładu obliczeniowego będzie rosło wraz ze wzrostem liczby punktów obsługi. Może więc w pewnej części

niwelować wzrost liczności zbioru rozwiązań dopuszczalnych.

Jak wspomniano wcześniej, do realizacji tej modyfikacji wymagane jest wyznaczenie permutacji określającej odwzorowanie uporządkowanego wektora kosztów na wektor oryginalny. Może to być jednak realizowane przy okazji sortowania i scalania wektorów kosztów z pomijalnie małym nakładem obliczeń bez wpływu na asymptotyczną złożoność algorytmu. Wymagana jest jedynie dodatkowa pamięć na zapisanie tej permutacji.

Ponieważ w trakcie algorytmu operujemy głównie na uporządkowanym wektorze kosztów, a na podstawie permutacji możemy z niego wyznaczyć oryginalny wektor kosztów, więc ten ostatni nie jest bezpośrednio przechowywany.

### 8.2.3. Warunek dominacji

Trzecia modyfikacja wynika po części z podobnych obserwacji jak modyfikacja druga. Proponowany w literaturze algorytm wymiany wyznacza wartość funkcji celu dla każdego rozpatrywanego rozwiązania (każdej wymiany). W przypadku gdy liczba zmieniających się kosztów jest niewielka, wydaje się, że wcześniej warto sprawdzić perspektywiczność nowego rozwiązania. Celem tej modyfikacji jest odrzucenie rozwiązań nieperspektywicznych bez kosztownego wyznaczania wartości funkcji celu.

Warunkiem decydującym o odrzuceniu rozpatrywanego rozwiązania jest warunek dominacji posortowanych wektorów kosztów. Czyli jeśli posortowany nowy wektor kosztów jest większy (lub równy) od aktualnego posortowanego wektora kosztów po współrzędnych, to wartość funkcji celu nowego rozwiązania nie będzie lepsza od aktualnego rozwiązania. Innymi słowy, nowe rozwiązanie nie jest rozwiązaniem symetrycznie efektywnym (lub jest równe aktualnemu) i można je pominąć w dalszych rozważaniach.

Oczywiście naiwne (bezpośrednie) porównanie całego wektora kosztów nie ma sensu, gdyż — podobnie jak przy wyznaczaniu funkcji celu — wymagałoby to posortowania go w całości. Okazuje się jednak, że wystarczy porównać tylko te koszty, które uległy zmianie przy zadanej wymianie punktów obsługi.

Dla uzasadnienia poprawności tego warunku (modyfikacji) użyjemy powszechnie znanej własności dotyczącej posortowanych wektorów. Jeśli dla dwóch wektorów  $\mathbf{y}'$ ,  $\mathbf{y}''$  jeden z nich jest nie większy niż drugi, czyli zachodzi  $y'_i \leq y''_i$  po współrzędnych, to zależność ta zachodzi również dla odpowiadających im wektorów powstałych przez posortowanie wektorów  $\mathbf{y}'$  i  $\mathbf{y}''$ . Formalnie przedstawia to poniższy lemat (patrz również Dijkstra [16]).

**Lemat 8.1.** Jeśli wektory  $\mathbf{y}', \mathbf{y}'' \in R^m$  spełniają warunek  $y'_i \leq y''_i$  dla każdego  $i = 1, \dots, m$ , to zachodzi również  $\theta_i(\mathbf{y}') \leq \theta_i(\mathbf{y}'')$  dla  $i = 1, \dots, m$ , gdzie  $\theta_i(\mathbf{y}') = y'_{\tau'(i)}$ ,  $\theta_i(\mathbf{y}'') = y''_{\tau''(i)}$  oraz

$$\theta_1(\mathbf{y}') \geq \theta_2(\mathbf{y}') \geq \dots \geq \theta_m(\mathbf{y}'),$$

$$\theta_1(\mathbf{y}'') \geq \theta_2(\mathbf{y}'') \geq \dots \geq \theta_m(\mathbf{y}'').$$

*Dowód.* Rozpatrzmy  $\theta_i(\mathbf{y}') = y'_{\tau'(i)}$  dla dowolnego ustalonego  $i = 1, \dots, m$ . Zgodnie z definicją operatora porządkowego  $y'_{\tau'(i)} \leq y'_{\tau'(j)}$  dla każdego  $j = 1, \dots, i - 1$ . Z założenia wynika  $y'_{\tau'(j)} \leq y''_{\tau'(j)}$  dla każdego  $j$  oraz  $y'_{\tau'(i)} \leq y''_{\tau'(i)}$ . W związku z tym istnieje co najmniej  $i$  składowych wektora  $\mathbf{y}''$  nie mniejszych od  $y'_{\tau'(i)}$ , czyli  $y'_{\tau'(i)} \leq \theta_i(\mathbf{y}'')$ , a więc  $\theta_i(\mathbf{y}') \leq \theta_i(\mathbf{y}'')$ .  $\square$

Wykorzystując lemat 8.1, proponowany warunek, który sprawdza, czy nowy wektor kosztów jest symetrycznie zdominowany przez aktualny wektor kosztów, przebiega w następujący sposób. Dla pojedynczej wymiany (ustalonej pary dodawanego i usuwanego punktu obsługi) bada się część wektora kosztów, która uległa zmianie po wykonaniu wymiany. Zapamiętuje się stare koszty i ich nowe odpowiedniki, sortuje tak powstałe wektory (o mniejszym wymiarze) i porównuje ze sobą. Jeśli powstały w ten sposób wektor zmienionych nowych kosztów jest zdominowany przez wektor zmienionych starych kosztów (tzn. nowe koszty są nie mniejsze niż stare i co najmniej jeden jest większy), to znaczy, że pełnowymiarowy nowy wektor kosztów jest symetrycznie zdominowany przez wektor kosztów aktualnego rozwiązania. Jest tak, ponieważ pozostałe składowe wektora kosztów są takie same w obu wektorach, przez co aktualny wektor kosztów dominuje w sensie Pareto nowy wektor kosztów. Bezpośrednio z lematu wynika więc również dominacja symetryczna. Dokładne wyznaczanie takiego rozwiązania nie ma oczywiście sensu (podobnie jest w przypadku otrzymania takiego samego wektora kosztów jak aktualny). Warto również zauważyć, że posortowanie nowego wektora kosztów będzie także wykorzystane przy drugiej modyfikacji (podpunkt 8.2.2), gdyby okazało się, że warunek dominacji nie jest spełniony.

Efektywność tej modyfikacji, podobnie jak poprzedniej, zależy od liczby kosztów, które ulegają zmianie. Ponieważ modyfikacja ta jest stosowana przy oddzielnym rozpatrywaniu dodawania i usuwania punktu obsługi, więc zarówno po dodaniu, jak i usunięciu punktu obsługi wymiar wektora, który podlega sortowaniu, wynosi średnio  $m/(n + 1)$ . Następnie te dwa wektory są ze sobą scalane z zachowaniem porządku, co można zrealizować ze złożonością liniową proporcjonalną do sumy wymiarów tych wektorów, czyli  $2m/(n + 1)$ . Zapamiętane wektory starych kosztów są już uporządkowane, więc wystarczy je tylko scalić.

Wadą tej modyfikacji jest fakt, że zmniejsza nakład obliczeń tylko w przypadku, gdy nowe rozwiązanie jest zdominowane przez dotychczasowe. Wówczas nie trzeba porządkować całego wektora kosztów i wyznaczać funkcji celu. W przeciwnym wypadku modyfikacja nie przyniesie jednak żadnego zysku, a ponieważ sama wymaga dodatkowych obliczeń, to może wręcz wydłużyć czas rozwiązania. Jej zysk będzie przez to niwelowany i sumarycznie może się okazać, że nie wniesie ona żadnej poprawy. Taka możliwość powinna być jednak minimalizowana dzięki temu, że część rezultatów tej modyfikacji (posortowany wektor zmienionych nowych kosztów) będzie wykorzystana przy zmodyfikowanym sortowaniu. Rozważając dodatkowe nakłady związane z proponowaną modyfikacją, należy jeszcze pamiętać o dodatkowej pamięci wymaganej do zapisania starych i nowych kosztów (przed i po wymianie).

#### 8.2.4. Regularyzacja

Dodanie regularyzacji miało na celu poprawę jakości uzyskiwanych rozwiązań. Dotyczyło to zwłaszcza problemów z kryterium centrum, dla których dokładność rozwiązań była znacznie gorsza niż dla problemów pozostałych typów.

Ze względu na specyfikę heurystyki rozwiązanie jest modyfikowane w kolejnych iteracjach tylko w niewielkim stopniu — wymieniany jest pojedynczy punkt obsługi. W problemach, gdzie znaczna liczba wag preferencji jest zerowa, porównanie wartości funkcji celu jest często niewystarczające do wyboru bardziej obiecującego rozwiązania. Jeśli wartości funkcji celu są takie same, to w danej chwili oba rozwiązania są równie dobre. Wybór rozwiązania ma jednak znaczenie dla dalszego przebiegu metody, gdyż wpływa na rozwiązania otrzymywane w kolejnych iteracjach.

Gdy część wag jest zerowa, funkcja celu nie jest ściśle monotoniczna ze względu na wektor ocen i poprawa ocen odpowiadających zerowym wagom nie polepszy wartości funkcji celu. W związku z tym może się zdarzać, że pomimo znalezienia lepszego rozwiązania w sensie wektora ocen nie zastąpi ono aktualnego rozwiązania, gdyż wartość funkcji celu nie uległa poprawie. Może mieć to negatywny wpływ na rozwiązania uzyskiwane w kolejnych iteracjach metody. Łatwo wyobrazić sobie rozwiązanie, w którym wymiana pojedynczego punktu obsługi poprawia wektor kosztów, ale nie poprawia wartości funkcji celu, którą można z kolei poprawić przy wymianie dwóch punktów obsługi. Rozwiązanie takie odpowiada pewnemu optimum lokalnemu. W takim przypadku wylosowane będzie rozwiązanie z innego sąsiedztwa i tam rozpocznie się nowe przeszukiwanie. Sytuacja taka zdarza się tym częściej, im większa jest liczba wag zerowych. W konsekwencji poszukiwanie może być za mało dokładne — metoda przeszukiwania nie jest wystarczająco zintensyfikowana dla problemów z dużą liczbą zerowych wag. Uwzględnienie poprawy wektora ocen pozwoliłoby na kontynuowanie przeszukiwania w aktualnym sąsiedztwie i znalezienie lepszego rozwiązania zarówno ze względu na wektor ocen, jak i wartość funkcji celu.

W ogólności uwzględnienie poprawy wektora ocen oczywiście nie daje pewności, że znalezione ostatecznie rozwiązanie będzie lepsze od tego, które zostałyby znalezione, gdyby bazować tylko na wartości funkcji celu. Wydaje się jednak, że postępowanie takie zwiększa szansę na uzyskanie lepszego rozwiązania. Zależy to również od sposobu uwzględnienia wektora ocen przy ocenie rozwiązania. Obiecujące wydaje się np. uwzględnianie poprawy tylko tych ocen, które w uporządkowanym (nierosnąco) wektorze znajdują się na pozycjach nie mniejszych niż oceny uwzględniane w funkcji celu. Daje to szansę, że w kolejnych iteracjach poprawione oceny zastąpią te, na podstawie których wyznaczana jest wartość funkcji celu.

W związku z powyższymi obserwacjami w proponowanej metodzie dodano warunek regularyzacji jako dodatkowy warunek przy porównywaniu dwóch rozwiązań z jednakowymi wartościami funkcji celu. Warunek ten polega na leksykograficznym porównaniu uporządkowanych wektorów ocen od zadanej pozycji składowej. Pozycja, od której rozpoczyna się porównywanie kolejnych składowych wektora kosztów, jest ustalana na podstawie wektora



wag. W pracy badano pozycję zdefiniowaną jako numer pierwszej zerowej składowej następującej po pierwszej niezerowej składowej wektora wag. Przykładowo, dla wektora wag  $w = (0, 1, 1, 0, 1, 0)$  będzie to pozycja czwarta. Porównywane byłyby zatem koszty znajdujące się na pozycjach czwartej, piątej i szóstej. Jako lepsze wybierane jest rozwiązanie, którego porównywana część uporządkowanego wektora kosztów jest leksykograficznie mniejsza.

Jak łatwo zauważyć, regularyzacji podlegają tylko problemy, w których przynajmniej jedna waga jest zerowa (i jest to waga występująca po wadze niezerowej). Problemy np. z kryterium średnim nie podlegają więc regularyzacji. Wydaje się to rozsądne, gdyż wszystkie oceny w takim problemie są uwzględniane w funkcji celu.

Koszt tej modyfikacji jest niewielki, gdyż wektory będą porównywane tylko w przypadku równych wartości funkcji celu. Pozycja, od której rozpoczyna się porównanie wektorów, jest wyznaczana w czasie liniowym proporcjonalnym do rozmiaru problemu  $m$  i operacja ta jest wykonywana jednokrotnie po wczytaniu problemu. Jej koszt jest więc pomijalny. Wymagane jest zapamiętanie uporządkowanego wektora kosztów i permutacji odwzorowującej go na wektor oryginalny dla aktualnie najlepszego rozwiązania. Metoda może oczywiście wykonywać się dłużej, ze względu na większą intensyfikację przeszukiwania w pojedynczej iteracji.

### 8.3. Implementacja

W podrozdziale przedstawiona jest propozycja implementacji metody przeszukiwania zmiennego sąsiedztwa z uwzględnieniem modyfikacji zaproponowanych w poprzednim punkcie. Przeglądanie i aktualizacja kosztów odbywa się na uporządkowanym wektorze kosztów (i zgodnie z jego porządkiem).

Poniżej wyjaśniona jest notacja używana w opisie algorytmu:

- $c$  macierz kosztów obsługi użytkowników ( $c(i, j)$  — koszt spełnienia zapotrzebowania  $i$ -tego klienta przez  $j$ -ty punkt obsługi dla  $i, j = 1, \dots, m$ ),
- $w$  wektor wag preferencji operatora OWA ( $w(k)$  —  $k$ -ta waga dla  $k = 1, \dots, m$ ),
- $scost$  uporządkowany wektor kosztów ( $scost(k)$  —  $k$ -ty największy koszt obsługi dla  $k = 1, \dots, m$ ),
- $\tau$  permutacja określająca odwzorowanie uporządkowanego wektora kosztów na oryginalny wektor kosztów ( $\tau(k)$  — indeks użytkownika o  $k$ -tym największym koszcie obsługi dla  $k = 1, \dots, m$ ),
- $f$  wartość funkcji celu,
- $d1$  wektor indeksów najbliższych punktów obsługi ( $d1(i)$  — indeks najbliższego punktu obsługi  $i$ -tego klienta dla  $i = 1, \dots, m$ ),
- $d2$  wektor indeksów drugich najbliższych punktów obsługi ( $d2(i)$  — indeks drugiego najbliższego punktu obsługi  $i$ -tego klienta dla  $i = 1, \dots, m$ ),
- $goin$  indeks punktu obsługi dodawanego do aktualnego rozwiązania,

- gout* indeks punktu obsługi usuwanego z aktualnego rozwiązania,
- b* flaga binarna przyjmująca wartość prawdy, gdy nowe rozwiązanie jest lepsze od aktualnego oraz fałszu w przeciwnym przypadku,
- scost'* wektor starych kosztów, które ulegają zmianie przy dodawaniu lub usuwaniu punktu obsługi ( $scost'(l)$  — stary koszt przed zmianą przy dodawaniu lub usuwaniu punktu obsługi dla  $l$  zmiennego dynamicznie, przy czym  $l_{max} = m$ ),
- scost''* wektor nowych kosztów, które pojawiły się przy dodawaniu lub usuwaniu punktu obsługi ( $scost''(l)$  — nowy koszt po zmianie przy dodawaniu lub usuwaniu punktu obsługi dla  $l$  zmiennego dynamicznie, przy czym  $l_{max} = m$ ),
- $\tau''$  wektor indeksów klientów, którym zmieniły się koszty przy dodawaniu lub usuwaniu punktu obsługi ( $\tau''(l)$  — indeks klienta, któremu zmienił się koszt przy dodawaniu lub usuwaniu punktu obsługi).

W ramach algorytmów wywoływane są również funkcje pomocnicze. Funkcja SORTUJ sortuje nierosnąco (zgodnie z założeniami) koszty obsługi, a gdy jest wywoływana z dodatkowym parametrem — będącym wektorem indeksów klientów, który odpowiada wektorowi kosztów — aktualizuje go zgodnie z porządkiem sortowania kosztów. Funkcja SCALKOLEJNO łączy ze sobą dwa posortowane wektory kosztów w taki sposób, aby wynik również był posortowany (nierosnąco). Można to interpretować jako pojedynczy krok sortowania przez scalanie. Analogicznie jak poprzednio, gdy funkcja ta jest wywoływana z dodatkowymi parametrami (indeksami klientów), łączy je ona zgodnie z kolejnością wynikającą z sortowania kosztów. Wynikowy uporządkowany wektor kosztów jest przypisywany do pierwszego parametru, a wynikowy wektor indeksów klientów do drugiego parametru. Funkcja OBLICZOWA wyznacza wartość funkcji celu, a ponieważ otrzymuje ona jako parametry uporządkowany wektor kosztów oraz wektor wag, stanowi w istocie iloczyn skalarny tych dwóch wektorów. Funkcja CZYLEPSZE sprawdza, czy nowe rozwiązanie jest lepsze od aktualnie najlepszego i zwraca logiczną wartość prawdy w takiej sytuacji. W przeciwnym razie zwraca logiczną wartość fałszu. W prezentowanej w algorytmach metodzie z regularyzacją funkcja CZYLEPSZE przyjmuje jako parametry wartości funkcji celu i uporządkowane wektory kosztów nowego i aktualnie najlepszego rozwiązania. Gdy nowa wartość funkcji celu okazuje się taka sama jak wartość aktualnie najlepsza, funkcja porównuje leksykograficznie odpowiednie części uporządkowanych wektorów kosztów obu rozwiązań. W podstawowej wersji metody funkcja ta przyjmuje jako parametry tylko wartości funkcji celu i sprawdza po prostu, czy nowa wartość jest mniejsza niż wartość aktualnie najlepsza.

Dodatkowo za pomocą \* oznaczane są parametry wyjściowe procedur, czyli takie, które są wyznaczane lub modyfikowane przez te procedury.

### 8.3.1. Dodawanie punktu obsługi

Algorytm 2 przedstawia procedurę dodawania punktu obsługi DODAJ wykorzystywaną w algorytmie wymiany. W pętli sprawdzane są koszty, które ulegają zmianie wskutek dodania

nowego punktu obsługi *goin*. Zapamiętywane są stare koszty (linia 5) do testu dominacji oraz nowe koszty wraz z indeksem klienta, którego zmiana dotyczy (linia 6), do wyznaczenia nowego uporządkowanego wektora kosztów i permutacji. Zmieniony koszt jest również usuwany z aktualnego uporządkowanego wektora kosztów (linia 7). Aktualizowane są także najbliższe i drugie najbliższe punkty obsługi, jednak tylko w takim stopniu, w jakim jest to potrzebne dla procedury usuwania punktu obsługi. Dlatego chociaż w linii 8 zmieniany jest najbliższy punkt obsługi, a więc zmianie ulega również drugi najbliższy, to jednak ten ostatni nie jest aktualizowany, gdyż informacja ta nie będzie wykorzystywana w procedurze usuwania (koszt tego klienta nie ulegnie zmianie przy usuwaniu, ponieważ dodawany właśnie punkt obsługi nie będzie rozpatrywany przy usuwaniu w tej iteracji). Następnie zmienione koszty są sortowane i scalane z zachowaniem porządku z kosztami, które nie uległy zmianie. Na koniec wyznaczana jest nowa wartość funkcji celu.

---

**Algorytm 2.** Procedura dodawania punktu obsługi

---

```

1: function DODAJ( $m, n, c, w, scost^*, \tau^*, d1^*, d2^*, goin, scost'^*, scost''^*, f_{new}^*$ )
2:    $l \leftarrow 0$ 
3:   for  $k = 1, \dots, m$  do
4:     if  $c(\tau^*(k), goin) < scost^*(k)$  then
5:        $scost'^*(l) \leftarrow scost^*(k)$ 
6:        $\tau''(l) \leftarrow \tau^*(k); scost''^*(l) \leftarrow c(\tau^*(k), goin)$ 
7:        $scost^*(k) \leftarrow null$ 
8:        $d1^*(\tau^*(k)) \leftarrow goin$ 
9:     else
10:      if  $c(\tau^*(k), goin) < c(\tau^*(k), d2^*(\tau^*(k)))$  then
11:         $d2^*(\tau^*(k)) \leftarrow goin$ 
12:      end if
13:    end if
14:     $l \leftarrow l + 1$ 
15:  end for
16:  /* Oblicz wartość funkcji celu po dodaniu punktu obsługi */
17:  SORTUJ( $scost''^*, \tau''$ )
18:  SCALKOLEJNO( $scost^*, \tau^*, scost''^*, \tau''$ )
19:   $f_{new}^* \leftarrow OBLICZOWA(scost^*, w)$ 
20: end function

```

---

Zarówno funkcja SCALKOLEJNO, jak i OBLICZOWA ma złożoność liniową  $O(m)$ . Pesymistyczna złożoność funkcji DODAJ wynika więc ze złożoności sortowania i wynosi  $O(m \log m)$ . W rzeczywistości jednak wektor podlegający sortowaniu będzie znacznie krótszy niż  $m$  i średnio będzie wynosił  $m/(n + 1)$ .

### 8.3.2. Usuwanie punktu obsługi

Algorytm 3 przedstawia procedurę usuwania punktu obsługi USUŃ. Jej zadaniem jest znalezienie dla zadanego dodawanego punktu obsługi najlepszego punktu do usunięcia, o ile nowe rozwiązanie będzie lepsze od aktualnie najlepszego.

W procedurze rozpatrywany jest każdy punkt obsługi znajdujący się w aktualnym rozwiązaniu. Dla każdego takiego punktu obsługi wyznaczany jest uporządkowany wektor kosztów ulegających zmianie. Następnie, za pomocą warunku dominacji (linia 19), sprawdzane jest, czy otrzymane po usunięciu tego punktu rozwiązanie ma szansę być lepsze od rozwiązania aktualnego. Jeśli tak, wyznaczany jest cały uporządkowany wektor kosztów i wartość funkcji celu. Jeśli otrzymane rozwiązanie jest lepsze od aktualnie najlepszego, zapamiętuje się nową wartość funkcji celu, uporządkowany wektor kosztów wraz z permutacją oraz rozpatrywany punkt obsługi. Ustawia się również flagę informującą o poprawieniu rozwiązania.

---

### Algorytm 3. Procedura usuwania punktu obsługi

---

```

1: function USUN( $m, n, c, w, x, scost_{cur}, \tau_{cur}, d1, d2, scost'_d, scost''_d, scost^*_{best}, \tau^*_{best}, f^*_{best}, goout^*, b^*$ )
2:   for  $goout \leftarrow x(1)$  to  $x(n)$  do
3:      $(scost_{new}, \tau_{new}) \leftarrow (scost_{cur}, \tau_{cur})$ 
4:     Opróżnij:  $scost'_{tmp}, scost''_{tmp}$ 
5:      $l \leftarrow 0$ 
6:     for  $k = 1, \dots, m$  do
7:       if  $d1(\tau(k)) = goout$  then
8:          $scost'_{tmp}(l) \leftarrow scost_{cur}(k)$ 
9:          $\tau''_u(l) \leftarrow \tau^*(k); scost''_{tmp}(l) \leftarrow c(\tau(k), d2(\tau(k)))$ 
10:         $scost_{new}(k) \leftarrow null$ 
11:         $l \leftarrow l + 1$ 
12:       end if
13:     end for
14:     SORTUJ( $scost''_{tmp}, \tau''_u$ )
15:      $scost''_u \leftarrow scost''_{tmp}$ 
16:     SCALKOLEJNO( $scost'_{tmp}, scost'_d$ )
17:     SCALKOLEJNO( $scost''_{tmp}, scost''_d$ )
18:     /* Sprawdź czy  $scost'_{tmp}$  dominuje  $scost''_{tmp}$  */
19:     if  $\neg(scost'_{tmp} \leq scost''_{tmp})$  then
20:       SCALKOLEJNO( $scost_{new}, \tau_{new}, scost''_u, \tau''_u$ )
21:        $f_{new} \leftarrow OBLICZOWA(scost_{new}, w)$ 
22:       if CZYLEPSZY( $f^*_{best}, f_{new}, scost^*_{best}, scost_{new}$ ) then
23:          $(f^*_{best}, scost^*_{best}, \tau^*_{best}) \leftarrow (f_{new}, scost_{new}, \tau_{new})$ 
24:          $goout^* \leftarrow goout$ 
25:          $b^* \leftarrow true$ 
26:       end if
27:     end if
28:   end for
29: end function

```

---

Dla pojedynczego punktu obsługi, który jest rozpatrywany do usunięcia, złożoność obliczeniowa jest analogiczna do złożoności procedury dodawania. Pesymistycznie określona jest ona przez operację sortowania i wynosi  $O(m \log m)$ . Ponieważ badanych jest  $n$  punktów do usunięcia, to pesymistyczna złożoność procedury usuwania jest równa  $O(nm \log m)$ . Jednak, podobnie jak w procedurze dodawania, wektor podlegający sortowaniu będzie zazwyczaj

znacznie krótszy niż  $m$  i średnio będzie wynosić  $m/(n+1)$ . W praktyce nakład obliczeń będzie więc znacznie mniejszy.

### 8.3.3. Aktualizacja

Procedura AKTUALIZUJ przedstawiona w Algorytmie 4 ma za zadanie dokonać zmian najbliższych i drugich najbliższych punktów obsługi zgodnie z wymianą punktu *goout* na *goin*. Informacje te są potrzebne przy dodawaniu i usuwaniu punktów obsługi. Proponowana procedura w odróżnieniu od przedstawionej w [17] nie aktualizuje wektora kosztów, który jest zapamiętywany przy okazji wyznaczania wartości funkcji celu w innych procedurach.

---

#### Algorytm 4. Procedura aktualizacji

---

```

1: function AKTUALIZUJ( $m, n, c, x, goin, goout, d1^*, d2^*$ )
2:   for  $i = 1, \dots, m$  do
3:     if  $d1^*(i) = goout$  then
4:       if  $c(i, goin) < c(i, d2^*(i))$  then
5:          $d1^*(i) \leftarrow goin$ ;
6:       else
7:          $d1^*(i) \leftarrow d2^*(i)$ 
8:          $d2^*(i) \leftarrow$  Znajdź drugi najbliższy punkt obsługi klienta  $i$ 
9:       end if
10:    else
11:      if  $c(i, goin) < c(i, d1^*(i))$  then
12:         $d2^*(i) \leftarrow d1^*(i)$ ;
13:         $d1^*(i) \leftarrow goin$ ;
14:      else
15:        if  $c(i, goin) < c(i, d2^*(i))$  then
16:           $d2^*(i) \leftarrow goin$ 
17:        else
18:          if  $d2^*(i) = goout$  then
19:             $d2^*(i) \leftarrow$  Znajdź drugi najbliższy punkt obsługi klienta  $i$ 
20:          end if
21:        end if
22:      end if
23:    end if
24:  end for
25: end function

```

---

Pesymistyczna złożoność pojedynczej iteracji pętli w procedurze jest proporcjonalna do  $n - 1$ , gdy trzeba wyznaczyć drugi najbliższy punkt obsługi. Pesymistyczna złożoność procedury AKTUALIZUJ wynosi więc  $O(m(n - 1)) = O(mn)$ .

### 8.3.4. Algorytm wymiany (przeszukiwanie lokalne)

W Algorytmie 5 przedstawiono algorytm wymiany, którego zadaniem jest znalezienie optimum lokalnego przy zadanym rozwiązaniu początkowym. Główna różnica w porównaniu do algorytmu przedstawionego w [17] dotyczy sposobu przeglądu sąsiadów, gdzie wykorzystujemy opisane wcześniej procedury. Zmiany wynikają z modyfikacji opisanej w części 8.2.1.

---

**Algorytm 5.** Algorytm wymiany

---

```
1: Inicjalizacja
2:   Znajdź rozwiązanie początkowe  $x_{opt}$ ;
3:   Wyznacz wektor kosztów, a na jego podstawie uporządkowany nierosnąco wektor  $scost_{opt}$ 
   i odpowiadającą mu permutację  $\tau_{opt}$  oraz wartość funkcji celu  $f_{opt}$ ;
4:   Wyznacz najbliższe  $d1_{opt}$  i drugie najbliższe  $d2_{opt}$  punkty obsługi dla każdej lokalizacji  $i =$ 
    $1, \dots, m$ ;
5: loop ▷ Iteracja
6:   Przegląd sąsiadów
7:      $(f_{best}, scost_{best}, \tau_{best}) \leftarrow (f_{opt}, scost_{opt}, \tau_{opt})$ 
8:      $b \leftarrow \text{false}$ 
9:     for  $goin \leftarrow x_{opt}(n+1)$  to  $x_{opt}(m)$  do
10:       $(d1_{new}, d2_{new}, scost_{new}, \tau_{new}) \leftarrow (d1_{opt}, d2_{opt}, scost_{opt}, \tau_{opt})$ 
11:      Opróżnij:  $scost', scost''$ 
12:      Dodaj punkt obsługi  $goin$  do rozwiązania:
13:      DODAJ( $m, n, c, w, scost_{new}, \tau_{new}, d1_{new}, d2_{new}, goin, scost', scost'', f_{new}$ )
14:      if CZYLEPSZY( $f_{best}, f_{new}, scost_{best}, scost_{new}$ ) then
15:        USUŃ( $m, n, c, w, x_{opt}, scost_{new}, \tau_{new}, d1_{new}, d2_{new},$ 
           $scost_{best}, \tau_{best}, f_{best}, scost', scost'', goout, b_u$ )
16:        if  $b_u$  then ▷ Zapamiętaj najlepszą parę punktów obsługi do wymiany
17:           $goin_{opt} \leftarrow goin$ 
18:           $goout_{opt} \leftarrow goout$ 
19:           $b \leftarrow \text{true}$ 
20:        end if
21:      end if
22:    end for
23:    Zatrzymanie
24:    if  $\neg b$  then ▷ Jeśli brak lepszych sąsiadów, przerwij
25:      Stop
26:    end if
27:    Aktualizacja
28:     $(f_{opt}, scost_{opt}, \tau_{opt}) \leftarrow (f_{best}, scost_{best}, \tau_{best})$ 
29:    Zaktualizuj  $x_{opt}$ : wymień  $goout_{opt}$  na  $goin_{opt}$ 
30:    Zaktualizuj najbliższy i drugi najbliższy punkt obsługi:
31:    AKTUALIZUJ( $m, n, c, x_{opt}, goin_{opt}, goout_{opt}, d1_{opt}, d2_{opt}$ )
32: end loop
```

---

Pesymistyczna złożoność pojedynczej iteracji algorytmu wymiany wynika z maksymalnie  $(m-n)$ -krotnego wywołania procedury USUŃ i wynosi  $O((m-n)nm \log m) = O(nm^2 \log m)$ . W tym pesymistycznym przypadku złożoność jest taka sama jak w [17]. Jednak w praktyce procedura USUŃ nie będzie wywoływana maksymalną liczbę razy oraz sama będzie wymagała mniejszej ilości obliczeń.

### 8.3.5. Wycena przejścia

W metodzie przeszukiwania zmiennego sąsiedztwa wykorzystujemy również, odpowiednio zmodyfikowaną, procedurę PRZEJŚCIE z [17] (w źródle określana jako Modified Move). Stosujemy ją do wyceny przejścia przy losowaniu rozwiązania z danego sąsiedztwa, gdy należy znaleźć najlepszy punkt do usunięcia dla danego dodawanego punktu obsługi, bez względu na jakość nowego rozwiązania (Algorytm 6).

---

#### Algorytm 6. Procedura wyceny przejścia

---

```

1: function PRZEJŚCIE( $m, n, c, w, x, scost^*, \tau^*, f^*, d1, d2, goin, goout^*$ )
2:   Inicjalizacja
3:    $f^* \leftarrow \infty$ 
4:   Najlepszy do usunięcia
5:   for  $goout \leftarrow x(1)$  to  $x(n)$  do
6:      $(scost_{new}, \tau_{new}) \leftarrow (scost^*, \tau^*)$ 
7:     Opróżnij  $scost'', \tau''$ 
8:      $l \leftarrow 0$ 
9:     for  $k = 1, \dots, m$  do
10:      if  $d1(\tau_{new}(k)) = goout$  then
11:         $scost''(l) \leftarrow \min\{c(\tau_{new}(k), goin), c(\tau_{new}(k), d2(\tau_{new}(k)))\}$ 
12:         $\tau''(l) \leftarrow \tau_{new}(k)$ 
13:         $scost_{new}(k) \leftarrow null$ 
14:      else
15:        if  $c(\tau_{new}(k), goin) < scost_{new}(k)$  then
16:           $scost''(l) \leftarrow c(\tau_{new}(k), goin)$ 
17:           $\tau''(l) \leftarrow \tau_{new}(k)$ 
18:           $scost_{new}(k) \leftarrow null$ 
19:        end if
20:      end if
21:       $l \leftarrow l + 1$ 
22:    end for
23:    SORTUJ( $scost'', \tau''$ )
24:    SCALKOLEJNO( $scost_{new}, \tau_{new}, scost'', \tau''$ )
25:     $f_{new} \leftarrow$  OBLICZOWA( $scost_{new}, w$ )
26:    if CZYLEPSZY( $f^*, f_{new}, scost_{best}, scost_{new}$ ) then
27:       $f^* \leftarrow f_{new}$ 
28:       $(scost_{best}, \tau_{best}) \leftarrow (scost_{new}, \tau_{new})$ 
29:       $goout^* \leftarrow goout$ 
30:    end if
31:  end for
32:   $(scost^*, \tau^*) \leftarrow (scost_{best}, \tau_{best})$ 
33: end function

```

---

Złożoność procedury PRZEJŚCIE jest analogiczna do złożoności procedury USUŃ i w przypadku pesymistycznym wynosi  $O(nm \log m)$ .

### 8.3.6. Przeszukiwanie zmiennego sąsiedztwa

Dla zupełności rozważań Algorytm 7 prezentuje całą metodę przeszukiwania zmiennego sąsiedztwa, która co do idei jest taka sama jak w [17]. Podobnie jak tam, losowanie rozwiązania nie gwarantuje jego otrzymania z badanego sąsiedztwa, ze względu na losowy charakter wyboru dodawanego punktu obsługi, który może zostać ponownie dodany po jego wcześniejszym usunięciu.

---

#### Algorytm 7. Metoda przeszukiwania zmiennego sąsiedztwa

---

```

1: Inicjalizacja
2:   Znajdź rozwiązanie początkowe  $x_{opt}$ 
3:   Wyznacz  $d1_{opt}, d2_{opt}, scost_{opt}, \tau_{opt}$  i  $f_{opt}$ 
4:    $(f_{cur}, x_{cur}, d1_{cur}, d2_{cur}, scost_{cur}, \tau_{cur}) \leftarrow (f_{opt}, x_{opt}, d1_{opt}, d2_{opt}, scost_{opt}, \tau_{opt})$ 
5: Krok główny
6:    $r \leftarrow 1$ 
7:   while ( $r \leq r_{max}$ ) i (warunek stopu nie jest spełniony) do
8:     Losowanie rozwiązania
9:     for  $j \leftarrow 1, \dots, r$  do ▷ wymień  $r$  punktów obsługi
10:      Wylusuj punkt obsługi  $goin$  do dodania
11:      Wyznacz usuwany punkt obsługi  $goout, scost_{cur}, \tau_{cur}$  i  $f_{cur}$  za pomocą procedury:
12:      PRZEJŚCIE( $m, n, c, w, x_{cur}, scost_{cur}, \tau_{cur}, f_{cur}, d1_{cur}, d2_{cur}, goin, goout$ )
13:      Zaktualizuj  $x_{cur}$ : wymień  $goout$  na  $goin$ 
14:      Wyznacz  $d1_{cur}$  i  $d2_{cur}$  za pomocą procedury:
15:      AKTUALIZUJ( $m, n, c, x_{cur}, goin, goout, d1_{cur}, d2_{cur}$ )
16:     end for
17:     Przeszukiwanie lokalne
18:     Zastosuj algorytm wymiany (bez inicjalizacji) z  $x_{cur}, d1_{cur}, d2_{cur}, scost_{cur}, \tau_{cur}, f_{cur}$ 
      jako parametrami wejściowymi i wyjściowymi
19:     Decyzja o zmianie rozwiązania
20:     if CZYLEPSZY( $f_{opt}, f_{cur}, scost_{opt}, scost_{cur}$ ) then
21:        $(f_{opt}, x_{opt}, d1_{opt}, d2_{opt}, scost_{opt}, \tau_{opt}) \leftarrow (f_{cur}, x_{cur}, d1_{cur}, d2_{cur}, scost_{cur}, \tau_{cur})$ 
22:        $r \leftarrow 1$ 
23:     else
24:        $(f_{cur}, x_{cur}, d1_{cur}, d2_{cur}, scost_{cur}, \tau_{cur}) \leftarrow (f_{opt}, x_{opt}, d1_{opt}, d2_{opt}, scost_{opt}, \tau_{opt})$ 
25:        $r \leftarrow r + 1$ 
26:     end if
27:   end while

```

---

Wylusowane rozwiązanie stanowi rozwiązanie początkowe przeszukiwania lokalnego, które realizowane jest za pomocą algorytmu wymiany. Gdy znalezione w ten sposób rozwiązanie okaże się lepsze niż dotychczasowe, wówczas je zastępuje i kolejne przeszukiwanie będzie się toczyć w jego najbliższym sąsiedztwie (linie 21–22). W przeciwnym przypadku zwiększa się odległość sąsiedztwa, w którym będą wykonywane następne przeszukiwania (linia 25). Dzieje się tak do momentu osiągnięcia maksymalnej ustalonej odległości  $r_{max}$  (która maksymalnie



może wynosić  $n$ ) lub spełnienia alternatywnego warunku stopu, np. maksymalnej liczby iteracji (linia 7).

## 8.4. Testy obliczeniowe

W celu sprawdzenia skuteczności zaproponowanych modyfikacji, zarówno tych wpływających na szybkość, jak i dokładność metody, zdecydowano się na sprawdzenie czterech wersji zmienionej metody VNS (tabela 8.1). Dwie wersje nie uwzględniają regularyzacji i obrazują tylko zmiany szybkości metody. Dwie kolejne wersje uwzględniają regularyzację, pokazując zmiany szybkościowe i jakościowe — wersje te są oznaczone dolnym indeksem  $r$ . Dodatkowo

		warunek dominacji	
		brak	obecny
regula-	brak	VNS'	VNS''
ryzacja	obecna	VNS' <sub>r</sub>	VNS'' <sub>r</sub>

Tabela 8.1. Wersje zmodyfikowanej metody VNS

w ramach tych dwóch grup wersje są zróżnicowane ze względu na modyfikacje szybkościowe. Jedna z wersji obejmuje wszystkie wymienione wcześniej modyfikacje szybkościowe (wersja ta oznaczana jest jako VNS'', a dla wersji z regularyzacją VNS''<sub>r</sub>). Druga wersja, oznaczana jako VNS' (i odpowiednio VNS'<sub>r</sub>), nie uwzględnia warunku dominacji (podpunkt 8.2.3), który wykorzystywany jest w procedurze usuwania punktu obsługi — algorytm 3, linia 19 (przez to nie jest również potrzebne zapamiętywanie wartości kosztów przed zmianą). Zmienione w ten sposób metody porównane zostały z oryginalną metodą przedstawioną w [17]. Dla jednoznaczności porównania oryginalna metoda została zaimplementowana i rozwiązano za jej pomocą te same instancje problemów. Proponowane zmiany wydajnościowe nie mają wpływu na jakość rozwiązania, a jedynie na szybkość algorytmu. Wersje metody VNS w ramach swojej grupy (z regularyzacją lub bez) dają te same rozwiązania, o ile początkowe i kolejno losowane rozwiązania będą takie same. W związku z tym zdecydowano się na ustawienie stałego, takiego samego we wszystkich wersjach, początkowego ziarna funkcji losującej. Dzięki temu czasy rozwiązania poszczególnych wersji w ramach każdej z dwóch grup dotyczą dokładnie tej samej drogi przeszukiwania przestrzeni rozwiązań. To również oznacza, że przy analizie jakości otrzymywanych rozwiązań wystarczy rozpatrzyć dwie wersje różniące się obecnością warunku regularyzacji, które będą po prostu oznaczane jako VNS i VNS<sub>r</sub>.

Poszczególne wersje metody VNS oraz procedura eksperymentalna została zaimplementowana w języku C++. Eksperymenty obliczeniowe zostały wykonane na maszynie z procesorem Intel Core2 Duo 2.53GHz (mobilny) i 4 GB pamięci RAM. Testy obliczeniowe zostały przeprowadzone w sposób analogiczny jak w pracy [17]. Metoda została sprawdzona na dwóch zbiorach problemów. Pierwsza grupa to wygenerowane problemy o niewielkich rozmiarach.

Druga grupa to problemy o dużych rozmiarach, które zostały zaczerpnięte z biblioteki OR (ang. OR-Library<sup>1</sup>).

#### 8.4.1. Problemy o niewielkim rozmiarze

Pierwsza część eksperymentów została wykonana na problemach o niewielkich rozmiarach. Wykorzystano do tego te same instancje problemów, które były używane przy testach modeli programowania matematycznego. Szczegółowy opis instancji testowych i sposobu ich generowania znajduje się w podrozdziale 5.6. W ramach analizy wydajnościowej, ze względu na ograniczoną dokładność pomiaru czasu, każdą instancję rozwiązano 100 razy i na tej podstawie wyznaczono średni czas rozwiązania problemu. Wszystkie problemy testowe zostały rozwiązane za pomocą pięciu wersji metody VNS i dla dwóch różnych sposobów inicjalizacji (zachłannej i losowej). Uwzględniając kombinacje parametrów, liczbę problemów, które zostały rozwiązane w ramach eksperymentów, można wyznaczyć jako 7 (warianty rozmiaru) x 4 (warianty liczby punktów obsługi) x 12 (warianty typów problemu) x 15 (instancje macierzy kosztów) x 5 (5 wersji metody) x 2 (2 sposoby inicjalizacji) x 100 (100 naliczeń problemu) = 5040000.

Statystyki czasowe dla wszystkich problemów znajdują się w dodatku C w tabeli C.1. Czasy rozwiązań najmniejszych problemów (dla  $m = 8$ , częściowo dla  $m = 10$ ) są poniżej lub na granicy dokładności pomiaru czasu (0,0001 s). Ich porównanie jest więc mało wiarygodne. Z tego powodu przy analizie skupiono się na nieco większych problemach. W tabelach 8.2 i 8.3 przedstawione są szczegółowe wyniki odpowiednio dla 15 i 30 lokalizacji. Dla zobrazowania głównych zależności czasu rozwiązania tych problemów, uśrednione dodatkowo po liczbie punktów obsługi, przedstawione są także na rysunkach 8.1 (15 lokalizacji, inicjalizacja zachłanna) i 8.2 (30 lokalizacji, inicjalizacja losowa).

Tabela 8.2: Statystyki czasowe rozwiązań problemów z 15 lokalizacjami

Problem		inicjalizacja zachłanna					inicjalizacja losowa				
		CPU[s]					CPU[s]				
typ	n	VNS	VNS'	VNS''	VNS' <sub>r</sub>	VNS'' <sub>r</sub>	VNS	VNS'	VNS''	VNS' <sub>r</sub>	VNS'' <sub>r</sub>
TC1	4	0,0008	0,0005	0,0014	0,0005	0,0014	0,0008	0,0005	0,0014	0,0005	0,0014
	5	0,0010	0,0006	0,0015	0,0006	0,0015	0,0011	0,0006	0,0015	0,0006	0,0015
	8	0,0016	0,0009	0,0018	0,0009	0,0018	0,0016	0,0009	0,0017	0,0009	0,0017
	9	0,0016	0,0010	0,0018	0,0010	0,0018	0,0017	0,0009	0,0017	0,0009	0,0017
TC2	4	0,0008	0,0003	0,0007	0,0005	0,0014	0,0008	0,0003	0,0006	0,0005	0,0014
	5	0,0010	0,0004	0,0008	0,0007	0,0017	0,0010	0,0003	0,0006	0,0006	0,0016
	8	0,0015	0,0005	0,0009	0,0010	0,0019	0,0016	0,0005	0,0008	0,0009	0,0017
	9	0,0016	0,0006	0,0010	0,0010	0,0019	0,0017	0,0005	0,0009	0,0009	0,0017
TC3	4	0,0008	0,0004	0,0011	0,0005	0,0014	0,0008	0,0004	0,0011	0,0005	0,0014
	5	0,0010	0,0005	0,0012	0,0006	0,0015	0,0011	0,0005	0,0012	0,0006	0,0015

kontynuacja na następnej stronie

<sup>1</sup> <http://people.brunel.ac.uk/mastjib/jeb/info.html>

Problem		inicjalizacja zachłanna					inicjalizacja losowa				
typ	n	CPU[s]					CPU[s]				
		VNS	VNS'	VNS''	VNS' <sub>r</sub>	VNS'' <sub>r</sub>	VNS	VNS'	VNS''	VNS' <sub>r</sub>	VNS'' <sub>r</sub>
TC4	8	0,0016	0,0008	0,0015	0,0009	0,0018	0,0017	0,0007	0,0013	0,0009	0,0017
	9	0,0017	0,0008	0,0015	0,0010	0,0018	0,0016	0,0007	0,0013	0,0009	0,0017
	4	0,0007	0,0005	0,0013	0,0005	0,0014	0,0008	0,0005	0,0012	0,0005	0,0014
	5	0,0010	0,0005	0,0013	0,0006	0,0015	0,0010	0,0005	0,0013	0,0006	0,0015
	8	0,0013	0,0006	0,0012	0,0008	0,0016	0,0013	0,0006	0,0011	0,0008	0,0015
TC5	9	0,0014	0,0006	0,0011	0,0009	0,0016	0,0014	0,0006	0,0010	0,0008	0,0016
	4	0,0008	0,0005	0,0014	0,0005	0,0014	0,0008	0,0005	0,0014	0,0005	0,0014
	5	0,0010	0,0006	0,0013	0,0006	0,0016	0,0011	0,0006	0,0014	0,0006	0,0016
	8	0,0017	0,0009	0,0019	0,0010	0,0019	0,0017	0,0009	0,0018	0,0009	0,0018
	9	0,0016	0,0007	0,0013	0,0010	0,0018	0,0016	0,0007	0,0012	0,0009	0,0017
TC6	4	0,0007	0,0005	0,0012	0,0005	0,0014	0,0008	0,0005	0,0013	0,0005	0,0014
	5	0,0009	0,0006	0,0015	0,0006	0,0016	0,0010	0,0006	0,0015	0,0007	0,0016
	8	0,0014	0,0006	0,0012	0,0009	0,0017	0,0015	0,0006	0,0011	0,0008	0,0016
	9	0,0014	0,0009	0,0017	0,0009	0,0017	0,0015	0,0009	0,0016	0,0009	0,0017
	4	0,0008	0,0005	0,0015	0,0005	0,0015	0,0008	0,0005	0,0014	0,0005	0,0014
TC7	5	0,0010	0,0006	0,0015	0,0006	0,0015	0,0011	0,0006	0,0016	0,0006	0,0016
	8	0,0017	0,0009	0,0019	0,0010	0,0019	0,0017	0,0009	0,0018	0,0009	0,0018
	9	0,0017	0,0008	0,0014	0,0010	0,0018	0,0017	0,0007	0,0013	0,0009	0,0017
	4	0,0008	0,0005	0,0013	0,0005	0,0014	0,0008	0,0005	0,0013	0,0005	0,0014
	5	0,0011	0,0007	0,0016	0,0007	0,0017	0,0011	0,0006	0,0016	0,0007	0,0017
TC8	8	0,0016	0,0009	0,0018	0,0010	0,0020	0,0016	0,0009	0,0018	0,0010	0,0019
	9	0,0017	0,0007	0,0012	0,0010	0,0018	0,0016	0,0006	0,0010	0,0009	0,0017
	4	0,0007	0,0005	0,0013	0,0005	0,0013	0,0008	0,0005	0,0014	0,0005	0,0014
	5	0,0010	0,0006	0,0015	0,0006	0,0015	0,0011	0,0006	0,0016	0,0006	0,0016
	8	0,0017	0,0009	0,0019	0,0009	0,0018	0,0017	0,0009	0,0018	0,0009	0,0017
TC9	9	0,0017	0,0010	0,0018	0,0010	0,0018	0,0016	0,0009	0,0017	0,0009	0,0017
	4	0,0007	0,0005	0,0013	0,0005	0,0013	0,0008	0,0005	0,0014	0,0005	0,0014
	5	0,0010	0,0006	0,0015	0,0006	0,0015	0,0011	0,0006	0,0016	0,0006	0,0016
	8	0,0017	0,0009	0,0019	0,0009	0,0018	0,0017	0,0009	0,0018	0,0009	0,0017
	9	0,0017	0,0010	0,0018	0,0010	0,0018	0,0016	0,0009	0,0017	0,0009	0,0017
TC10	4	0,0007	0,0005	0,0013	0,0005	0,0013	0,0007	0,0005	0,0014	0,0005	0,0014
	5	0,0009	0,0006	0,0014	0,0006	0,0014	0,0010	0,0006	0,0016	0,0006	0,0016
	8	0,0015	0,0009	0,0019	0,0009	0,0019	0,0016	0,0009	0,0019	0,0010	0,0018
	9	0,0016	0,0010	0,0019	0,0010	0,0018	0,0016	0,0010	0,0018	0,0010	0,0018
	4	0,0008	0,0005	0,0014	0,0005	0,0014	0,0008	0,0005	0,0014	0,0005	0,0014
TC11	5	0,0010	0,0006	0,0015	0,0006	0,0015	0,0011	0,0006	0,0016	0,0006	0,0016
	8	0,0017	0,0009	0,0019	0,0009	0,0018	0,0017	0,0009	0,0018	0,0009	0,0017
	9	0,0017	0,0010	0,0018	0,0010	0,0018	0,0016	0,0009	0,0017	0,0009	0,0017
	4	0,0008	0,0005	0,0014	0,0005	0,0014	0,0008	0,0005	0,0014	0,0005	0,0014
	5	0,0010	0,0006	0,0015	0,0006	0,0015	0,0011	0,0007	0,0016	0,0007	0,0016
TC12	8	0,0016	0,0009	0,0018	0,0009	0,0018	0,0017	0,0009	0,0018	0,0009	0,0018
	9	0,0016	0,0010	0,0018	0,0010	0,0018	0,0016	0,0009	0,0017	0,0009	0,0017
	4	0,0008	0,0005	0,0014	0,0005	0,0014	0,0008	0,0005	0,0014	0,0005	0,0014
	5	0,0010	0,0006	0,0015	0,0006	0,0015	0,0011	0,0007	0,0016	0,0007	0,0016
	8	0,0016	0,0009	0,0018	0,0009	0,0018	0,0017	0,0009	0,0018	0,0009	0,0018

Tabela 8.3: Statystyki czasowe rozwiązań problemów z 30 lokalizacjami

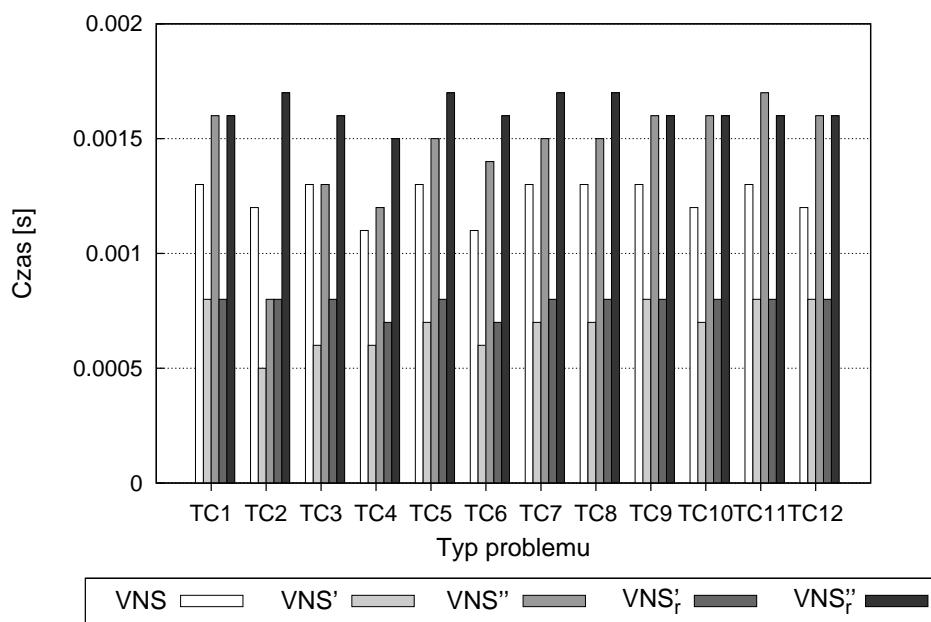
Problem		inicjalizacja zachłanna					inicjalizacja losowa				
typ	n	CPU[s]					CPU[s]				
		VNS	VNS'	VNS''	VNS' <sub>r</sub>	VNS'' <sub>r</sub>	VNS	VNS'	VNS''	VNS' <sub>r</sub>	VNS'' <sub>r</sub>
TC1	8	0,0141	0,0052	0,0099	0,0052	0,0099	0,0159	0,0055	0,0105	0,0055	0,0105
	10	0,0173	0,0062	0,0106	0,0062	0,0106	0,0210	0,0070	0,0119	0,0070	0,0119
	15	0,0274	0,0097	0,0139	0,0097	0,0138	0,0301	0,0099	0,0140	0,0101	0,0140
	16	0,0268	0,0097	0,0135	0,0097	0,0134	0,0292	0,0098	0,0135	0,0100	0,0135
TC2	8	0,0141	0,0021	0,0036	0,0060	0,0117	0,0154	0,0019	0,0030	0,0057	0,0109
	10	0,0180	0,0025	0,0041	0,0068	0,0119	0,0188	0,0022	0,0033	0,0068	0,0117
	15	0,0246	0,0043	0,0065	0,0098	0,0141	0,0238	0,0037	0,0053	0,0098	0,0137
	16	0,0229	0,0045	0,0067	0,0101	0,0142	0,0241	0,0041	0,0058	0,0097	0,0133
TC3	8	0,0147	0,0040	0,0077	0,0051	0,0099	0,0166	0,0041	0,0079	0,0054	0,0102
	10	0,0201	0,0051	0,0089	0,0071	0,0123	0,0215	0,0050	0,0086	0,0069	0,0117
	15	0,0281	0,0073	0,0106	0,0099	0,0141	0,0301	0,0072	0,0103	0,0100	0,0139
	16	0,0275	0,0072	0,0103	0,0099	0,0138	0,0294	0,0070	0,0099	0,0099	0,0134
TC4	8	0,0131	0,0046	0,0090	0,0050	0,0098	0,0141	0,0045	0,0090	0,0051	0,0101
	10	0,0176	0,0055	0,0098	0,0063	0,0113	0,0192	0,0056	0,0098	0,0067	0,0118
	15	0,0227	0,0065	0,0099	0,0088	0,0130	0,0243	0,0063	0,0094	0,0089	0,0129
	16	0,0229	0,0065	0,0096	0,0090	0,0129	0,0243	0,0062	0,0090	0,0091	0,0129
TC5	8	0,0146	0,0047	0,0089	0,0053	0,0102	0,0166	0,0049	0,0093	0,0056	0,0107
	10	0,0187	0,0051	0,0089	0,0066	0,0113	0,0214	0,0055	0,0093	0,0071	0,0121
	15	0,0274	0,0098	0,0140	0,0104	0,0148	0,0291	0,0096	0,0138	0,0109	0,0151
	16	0,0273	0,0062	0,0090	0,0098	0,0137	0,0290	0,0058	0,0083	0,0102	0,0138
TC6	8	0,0138	0,0051	0,0098	0,0056	0,0107	0,0159	0,0055	0,0106	0,0059	0,0114
	10	0,0170	0,0061	0,0107	0,0068	0,0118	0,0199	0,0067	0,0116	0,0077	0,0132
	15	0,0237	0,0055	0,0082	0,0092	0,0133	0,0268	0,0054	0,0079	0,0096	0,0135
	16	0,0242	0,0093	0,0132	0,0109	0,0153	0,0270	0,0094	0,0134	0,0107	0,0147
TC7	8	0,0144	0,0053	0,0101	0,0054	0,0104	0,0163	0,0055	0,0106	0,0058	0,0110
	10	0,0189	0,0067	0,0115	0,0070	0,0120	0,0216	0,0071	0,0122	0,0075	0,0127
	15	0,0279	0,0066	0,0097	0,0104	0,0148	0,0305	0,0067	0,0096	0,0101	0,0140
	16	0,0262	0,0095	0,0133	0,0112	0,0155	0,0286	0,0097	0,0134	0,0110	0,0148
TC8	8	0,0141	0,0052	0,0099	0,0054	0,0105	0,0165	0,0056	0,0107	0,0060	0,0116
	10	0,0182	0,0044	0,0076	0,0071	0,0122	0,0217	0,0049	0,0083	0,0072	0,0123
	15	0,0259	0,0050	0,0075	0,0099	0,0141	0,0291	0,0049	0,0072	0,0101	0,0140
	16	0,0249	0,0048	0,0072	0,0098	0,0136	0,0289	0,0049	0,0071	0,0100	0,0136
TC9	8	0,0147	0,0052	0,0100	0,0052	0,0100	0,0163	0,0054	0,0103	0,0054	0,0104
	10	0,0183	0,0064	0,0110	0,0064	0,0110	0,0219	0,0071	0,0122	0,0072	0,0122
	15	0,0287	0,0102	0,0145	0,0102	0,0145	0,0310	0,0102	0,0145	0,0104	0,0144
	16	0,0280	0,0101	0,0141	0,0101	0,0141	0,0300	0,0100	0,0139	0,0102	0,0138
TC10	8	0,0130	0,0054	0,0103	0,0053	0,0103	0,0143	0,0055	0,0107	0,0055	0,0107
	10	0,0181	0,0073	0,0125	0,0073	0,0125	0,0200	0,0076	0,0131	0,0077	0,0131
	15	0,0265	0,0110	0,0155	0,0110	0,0154	0,0284	0,0111	0,0156	0,0113	0,0155
	16	0,0277	0,0116	0,0159	0,0116	0,0159	0,0296	0,0118	0,0161	0,0120	0,0161
TC11	8	0,0153	0,0055	0,0105	0,0055	0,0105	0,0163	0,0055	0,0105	0,0055	0,0105

kontynuacja na następnej stronie

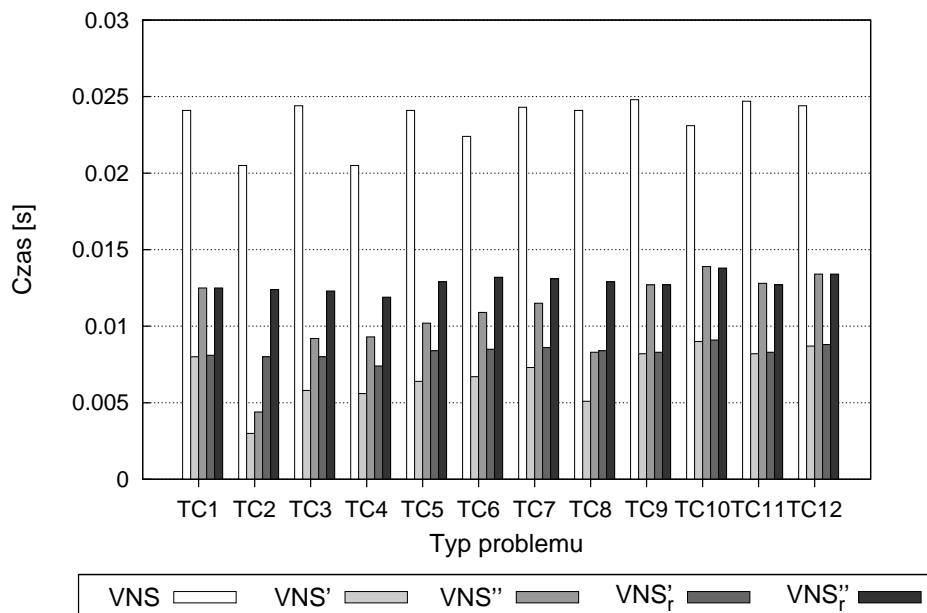
		inicjalizacja zachłanna					inicjalizacja losowa				
Problem typ	n	CPU[s]					CPU[s]				
		VNS	VNS'	VNS''	VNS' <sub>r</sub>	VNS'' <sub>r</sub>	VNS	VNS'	VNS''	VNS' <sub>r</sub>	VNS'' <sub>r</sub>
TC12	10	0,0179	0,0063	0,0109	0,0063	0,0108	0,0217	0,0071	0,0122	0,0072	0,0121
	15	0,0288	0,0102	0,0146	0,0102	0,0145	0,0310	0,0102	0,0145	0,0104	0,0144
	16	0,0280	0,0101	0,0141	0,0101	0,0141	0,0300	0,0100	0,0139	0,0102	0,0139
	8	0,0136	0,0052	0,0099	0,0052	0,0099	0,0159	0,0057	0,0109	0,0057	0,0110
	10	0,0177	0,0066	0,0113	0,0066	0,0113	0,0209	0,0073	0,0125	0,0074	0,0125
	15	0,0280	0,0107	0,0150	0,0106	0,0150	0,0303	0,0107	0,0151	0,0109	0,0151
	16	0,0294	0,0114	0,0157	0,0114	0,0157	0,0303	0,0110	0,0151	0,0112	0,0150

Porównując czasy różnych wersji metody VNS, można zaobserwować kilka wyraźnych zależności. Po pierwsze, wersja VNS' osiąga najkrótsze czasy dla prawie wszystkich problemów (pomijając pewne niejednoznaczności dla najmniejszych rozmiarów). Po drugie, oryginalna wersja VNS z literatury osiąga krótsze czasy niż wersja VNS'' dla problemów o mniejszym rozmiarze. W miarę wzrostu liczby lokalizacji różnica maleje i dla problemów o rozmiarze 15–20 lokalizacji czasy obu wersji są do siebie zbliżone. Dla większych rozmiarów wersja VNS'' osiąga już znacznie krótsze czasy. Sytuacja ta dotyczy wszystkich typów problemów, przy czym dla kryterium centrum (typ TC2) przewaga wersji VNS'' objawia się już dla mniejszych rozmiarów (dla 15 lokalizacji). W miarę wzrostu rozmiaru problemu maleje również różnica między wersją VNS'' a wersją VNS' bez warunku dominacji. Oczywiście rośnie również różnica między wersją VNS a wersją VNS'. Wyniki są analogiczne dla obu sposobów inicjalizacji (porównaj rysunki 8.1 i 8.2). Podobna sytuacja dotyczy wersji z regularyzacją. Dla problemów ze wszystkimi wagami niezerowymi (TC1, TC9–TC12) czasy wersji z regularyzacją są praktycznie identyczne jak wersji bez regularyzacji, co potwierdza fakt, że warunek regularyzacji nie jest aktywny dla tego typu problemów. Dla problemów pozostałych typów (TC2–TC8) czasy wersji z regularyzacją są dłuższe, przy czym największe różnice występują dla problemów z kryterium centrum (TC2). Potwierdza to wcześniejsze oczekiwania, że warunek regularyzacji będzie najczęściej aktywny dla problemów typu TC2. Większe czasy wynikają oczywiście z większej intensywności przeszukiwania w poszczególnych iteracjach. Poza tym ogólna charakterystyka wyników jest praktycznie taka sama jak opisana powyżej dla wersji bez regularyzacji. Różnica jest taka, że zwiększone czasy dla kryterium centrum upodabniają się do czasów rozwiązania pozostałych typów problemów.

Na podstawie analizy otrzymanych wyników można stwierdzić, że zaproponowane modyfikacje metody VNS z agregacją OWA mogą skrócić czas rozwiązania problemów z dowolnymi wagami preferencji. Zmiany wymagają pewnych dodatkowych kosztów, jednak uzyskane zyski są znacząco większe — przynajmniej dla modyfikacji dotyczącej relaksacji i sortowania. Wątpliwości można mieć do warunku dominacji (podrozdział 8.2.3), gdyż wersja metody VNS''/VNS''<sub>r</sub> z tym warunkiem, dla wszystkich dotychczasowych problemów, miała czas gorszy niż wersja VNS'/VNS'<sub>r</sub> (warunek dominacji wymaga zapamiętania wartości kosztów przed zmianą). Częściowo może być to również spowodowane nakładaniem się obszarów zysku



Rysunek 8.1. Średni czas działania metody VNS dla poszczególnych typów problemów z 15 lokalizacjami przy inicjalizacji zachłannej.



Rysunek 8.2. Średni czas działania metody VNS dla poszczególnych typów problemów z 30 lokalizacjami przy inicjalizacji losowej.

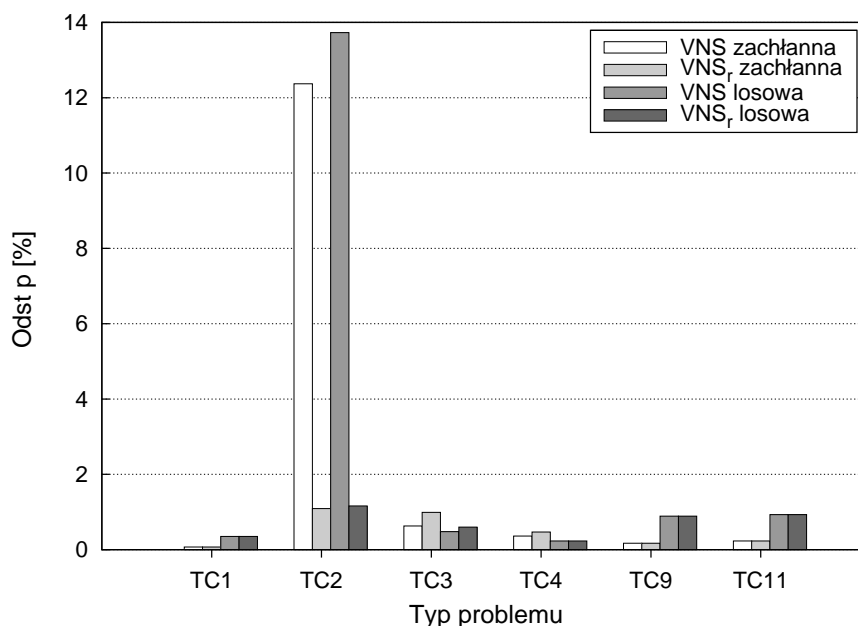
warunku dominacji z modyfikacją dotyczącą relaksacji. Mówiąc precyzyjniej, część nieperspektywicznych rozwiązań, które zostałyby wyeliminowane przez warunek dominacji, jest już wcześniej eliminowanych przez sprawdzenie wartości funkcji celu problemu zrelaksowanego. Warto jednak zauważyć, że wraz ze wzrostem rozmiaru problemu rozbieżności czasowe między nowymi wersjami różniącymi się obecnością warunku dominacji maleją. Skłania to do sprawdzenia omawianych modyfikacji dla problemów o większych, bardziej realistycznych rozmiarach. Poza tym czasy rozwiązania dotychczasowych problemów, dla wszystkich wersji metody VNS, są liczone w setnych częściach sekundy, co w praktyce oznacza, że dla takich rozmiarów wszystkie wersje można uznać za jednakowo dobre.

W ramach testów sprawdzono także jakość rozwiązań znajdowanych przez metodę przybliżoną. Za kryterium jakościowe przyjęto odstęp od rozwiązania optymalnego (najlepszego) wyznaczany zgodnie ze wzorem

$$\text{odstęp} = \frac{f - f_{opt}}{f_{opt}} \times 100\%, \quad (8.1)$$

gdzie  $f$  oznacza wartość funkcji celu znalezionego rozwiązania, a  $f_{opt}$  wartość funkcji celu rozwiązania optymalnego. Analizę przeprowadzono dla tych typów problemów, dla których udało się znaleźć rozwiązania optymalne wszystkich instancji problemów do rozmiaru 30 lokalizacji. Dotyczyło to problemów z nierosnącymi wagami preferencji (TC1–TC3, TC9, TC11), dla których rozwiązania optymalne zostały znalezione przez liniowe sformułowania agregacji OWA, a także problemów typu TC4, który udało się rozwiązać za pomocą hybrydowych modeli agregacji OWA. Dla powyższych kryteriów wyznaczono odstępów rozwiązań metody przybliżonej od rozwiązań optymalnych. Rozważono dwie wersje metody przybliżonej różniące się obecnością warunku regularyzacji (modyfikacje szybkościowe nie wpływają na znajdowane rozwiązania). Przez VNS oznaczono wyniki dla wersji bez regularyzacji, a przez VNS<sub>r</sub> wyniki wersji z regularyzacją. Szczegółowe wyniki znajdują się w dodatku C w tabeli C.2. W poszczególnych wierszach znajdują się statystyki dla 15 instancji macierzy kosztów w ramach kombinacji typu problemu, rozmiaru problemu i wariantu liczby punktów obsługi. Na rysunku 8.3 przedstawiono wyniki dla problemów z 30 lokalizacjami, uśrednione dodatkowo po wariantach liczby punktów obsługi.

Analiza wyników pokazuje, że poza kryterium centrum (TC2) metoda VNS znajduje rozwiązania dobrej jakości. Dla wielu problemów uzyskano rozwiązania optymalne, a średnie odstępów dla zdecydowanej większości kombinacji typów, rozmiarów i wariantów liczby punktów obsługi wynoszą 1–2%. Dla problemów typu TC3 i TC4 warunek regularyzacji ma niewielki wpływ i wersje różniące się obecnością tego warunku mają zbliżone rezultaty (dla problemów typu TC1, TC9, TC11 regularyzacja nie jest aktywna). Odmienna sytuacja ma miejsce dla kryterium centrum. Dla problemów typu TC2 rozwiązania uzyskane przez wersję metody VNS bez regularyzacji w większym stopniu odbiegają od rozwiązań optymalnych. Wydaje się także, że odstępów zwiększają się wraz ze wzrostem liczby lokalizacji i liczby

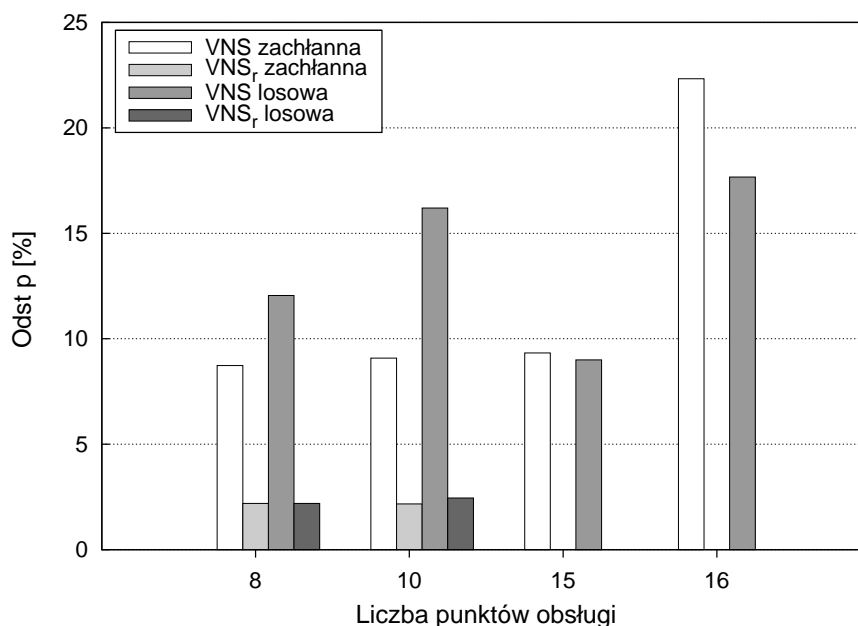


Rysunek 8.3. Średni odstęp [%] rozwiązań metody VNS z agregacją OWA dla problemów z 30 lokalizacjami (pojedynczy słupek stanowi średni odstęp dla 60 problemów — 4 warianty liczby punktów obsługi i 15 instancji macierzy kosztów)

punktów obsługi. W przypadku kryterium centrum regularyzacja znacznie poprawia jakość uzyskiwanych rozwiązań. Wersja metody VNS z regularyzacją osiąga rozwiązania o kilkakrotnie mniejszych odstępach od rozwiązań optymalnych i częściej, zwłaszcza dla większej liczby punktów obsługi, znajduje rozwiązania optymalne. Widać to na rysunku 8.4, gdzie przedstawione są wyniki metody VNS dla problemów typu TC2 z 30 lokalizacjami. Oczywiście warunek regularyzacji nie gwarantuje uzyskania lepszych rezultatów i dla części problemów, zwłaszcza o mniejszych rozmiarach, zdarza się, że rozwiązania uzyskiwane przez wersję z regularyzacją charakteryzują się gorszą jakością.

Podsumowując, metoda VNS z agregacją OWA pozwala znajdować rozwiązania dobrej jakości dla niewielkich problemów typu TC1, TC3, TC4, TC9 i TC11. Dla problemów typu TC3 i TC4 regularyzacja ma znikomy wpływ na uzyskiwane rozwiązania (dla problemów typu TC1, TC9 i TC11 warunek regularyzacji nie jest aktywny). Dla problemów typu TC2 metoda VNS bez regularyzacji znajduje rozwiązania o znacząco gorszej jakości i wydaje się, że ich jakość spada wraz ze wzrostem rozmiaru i liczby punktów obsługi. W tym przypadku zdecydowanie pomagają regularyzacja, która pozwala na znaczącą poprawę jakości uzyskiwanych rozwiązań. Wydaje się również, że poprawa jest większa dla problemów o większym rozmiarze i większej liczbie punktów obsługi. Skłania to do sprawdzenia zmodyfikowanej metody na problemach o dużych rozmiarach.





Rysunek 8.4. Średni odstęp [%] metody VNS z agregacją OWA dla problemów z kryterium centrum z 30 lokalizacjami (pojedynczy słupek stanowi średni odstęp dla 15 instancji problemów)

#### 8.4.2. Problemy o dużym rozmiarze

Druga część eksperymentów dotyczyła problemów o dużych rozmiarach. Do przetestowania wprowadzonych modyfikacji wykorzystano problemy z biblioteki badań operacyjnych (tzw. OR-Library [7]). Dokładnie chodzi o problemy lokalizacyjne bez ograniczeń pojemnościowych z kryterium średniej<sup>2</sup>. Na stronie dostępne są pliki z danymi poszczególnych instancji, a także instrukcja ich przygotowania — mówiąc w skrócie, na podstawie danych z plików należy wygenerować graf z krawędziami o podanych kosztach, a następnie na tej podstawie wyznaczyć odległości między wszystkimi parami wierzchołków za pomocą algorytmu Floyd-Warshalla [12, rozdział 25]. W wyniku otrzymania się pełną macierz kosztów dla danej instancji problemu. W sumie jest 40 instancji problemów (oznaczone jako *pmed1*, ..., *pmed40*). Rozpatrywanych jest 9 różnych rozmiarów — wartości liczby lokalizacji ( $m = 100, 200, 300, \dots, 900$ ). Dla poszczególnych rozmiarów wyznaczane są różne wartości liczby punktów obsługi: 5, 10,  $m/10$ ,  $m/5$ ,  $m/3$  (jeśli nie są to wartości całkowite to są zaokrąglane). Przy czym dla  $m = 700$  są rozpatrywane cztery pierwsze wartości, a dla  $m = \{800, 900\}$  trzy pierwsze wartości. Na podstawie danych z biblioteki OR określono trzy zbiory problemów, różniące się wektorem wag OWA: problemy z kryterium średniej (TC1), problemy z kryterium centrum (TC2) oraz problemy z zawężoną średnią (TC4).

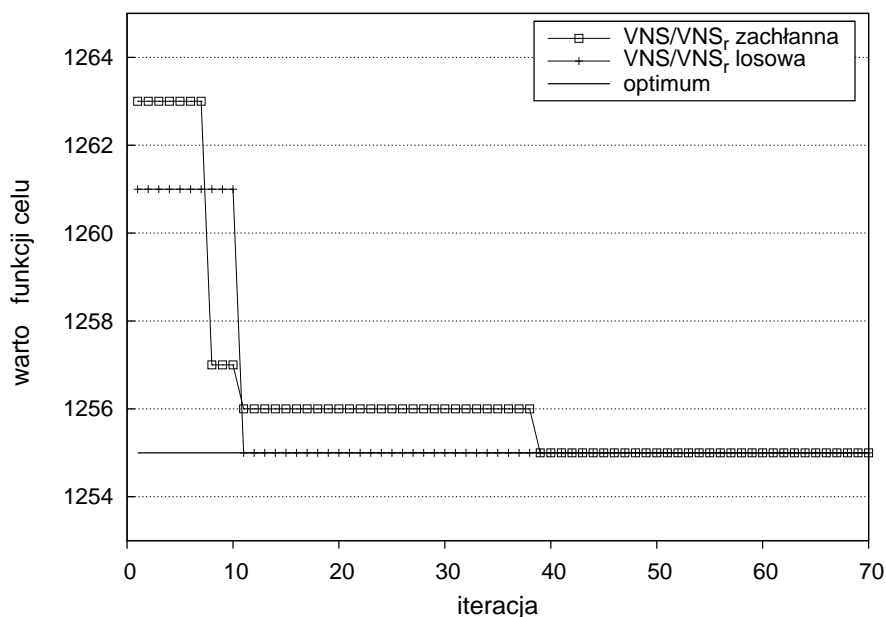
W pierwszej kolejności prezentowane są wyniki dla problemów z kryterium średniej. Ze względu na znaczny wzrost czasu rozwiązania dodany został dodatkowy warunek stopu w postaci maksymalnej liczby iteracji. W pracy [17] rozważano limit 50 iteracji, który

<sup>2</sup> <http://people.brunel.ac.uk/mastjib/jeb/orlib/pmedinfo.html>

zmieniono ostatecznie na 5 iteracji argumentując szybką zbieżnością metody (używano tam inicjalizacji zachłannej). Faktycznie, największa poprawa następuje w pierwszej iteracji — wyjątek stanowią nieliczne przypadki, w których zazwyczaj rozwiązanie początkowe okazało się rozwiązaniem optymalnym. Szybka zbieżność występuje również w problemach z małą liczbą punktów obsługi i zazwyczaj najlepsze rozwiązanie, które często okazuje się rozwiązaniem optymalnym, jest znajdowane już w pierwszej iteracji. W takim przypadku ograniczenie na liczbę iteracji nie jest konieczne, ponieważ szybko zadziała podstawowy warunek stopu związany z najdalszym sąsiedztwem. Dla problemów z większą liczbą punktów obsługi zbieżność w dalszych iteracjach nie jest już jednak taka szybka. Co prawda, względna wartość poprawy, w stosunku do wartości funkcji celu, nie jest zazwyczaj duża, jednak stopień zmniejszenia odstępów od rozwiązania optymalnego można uznać za znaczący — szczególnie jeśli priorytetem jest jakość rozwiązania. Na rysunku 8.5 przedstawiony jest przebieg metody VNS dla problemu *pmed10* z kryterium średniej z pominięciem rozwiązania początkowego. Dwa wykresy (warunek regularyzacji nie jest aktywny dla kryterium średniej) obrazują poprawę wartości funkcji celu w kolejnych iteracjach dla dwóch sposobów inicjalizacji. W obu przypadkach metoda osiąga zaznaczone rozwiązanie optymalne, przy czym w przypadku inicjalizacji zachłannej następuje to w 39 iteracji, a w przypadku inicjalizacji losowej w 11 iteracji. Odstęp od rozwiązania optymalnego, który po pierwszej iteracji wynosi  $8/1255$  (0,64%) dla inicjalizacji zachłannej oraz  $6/1255$  (0,48%) dla inicjalizacji losowej, jest w całości niwelowany dla tego problemu. W związku z powyższym efektywność wprowadzonych modyfikacji zbadana została dla ograniczenia 50 iteracji (warto również zaznaczyć, że ze względu na wzrost mocy obliczeniowej czasy osiągane obecnie dla 50 iteracji są lepsze od czasów raportowanych w [17] dla 5 iteracji).

W tabeli 8.4 przedstawiono czasy rozwiązań problemów z kryterium średniej. Ponieważ dla kryterium średniej warunek regularyzacji nie jest aktywny, czasy wersji różniące się obecnością warunku regularyzacji są równe (minimalne różnice mogą wynikać z szczegółów technicznych implementacji lub z niedokładności pomiaru). W związku z tym poniższa charakterystyka wyników odnosi się do wersji bez regularyzacji, ale należy pamiętać, że dotyczy to również wersji z regularyzacją.

Jak widać czasy zmodyfikowanych metod są znacznie krótsze od metody oryginalnej, nawet o jeden rząd wielkości. W ogólności różnice rosną wraz ze wzrostem rozmiaru problemu — liczby lokalizacji. Szczególny wzrost zysku z wprowadzonych modyfikacji można zaobserwować przy wzroście liczby punktów obsługi — zwłaszcza dla wersji z warunkiem dominacji. Dla przypadków z większą liczbą punktów obsługi (od  $m/5$ , a dla większych rozmiarów już od  $m/10$ ) czasy rozwiązania są ok. 6–13 razy krótsze. Potwierdza to oczekiwania przedstawione przy analizie modyfikacji (patrz 8.2.3 i 8.2.2). Dotyczy to obu sposobów inicjalizacji, przy czym dla inicjalizacji losowej występują problemy, których czas rozwiązania za pomocą zmodyfikowanej metody jest 14, a nawet ok. 16 razy krótszy. Średnio na zbiorze 40 problemów wersje VNS' i VNS'' uzyskują rozwiązania odpowiednio ok. 5,3 i 6 razy szybciej w przypadku



Rysunek 8.5. Przebieg metody VNS/VNS<sub>r</sub> dla problemu *pmed10* z kryterium średniej (bez rozwiązania początkowego).

inicjalizacji zachłannej oraz ok. 6 i 6,7 razy szybciej przy inicjalizacji losowej. Spostrzeżenia te potwierdzają hipotezę wysuniętą przy omawianiu wyników dla małych problemów, że wprowadzone modyfikacje pozwalają na znaczne skrócenie czasu rozwiązywania problemów lokalizacyjnych z kryterium uporządkowanej średniej ważonej.

Tabela 8.4: Statystyki czasowe metody VNS z agregacją OWA dla problemów z biblioteki OR z kryterium średniej (TC1)

Problem			inicjalizacja zachłanna					inicjalizacja losowa				
nazwa	m	n	CPU[s]					CPU[s]				
			VNS	VNS'	VNS''	VNS' <sub>r</sub>	VNS'' <sub>r</sub>	VNS	VNS'	VNS''	VNS' <sub>r</sub>	VNS'' <sub>r</sub>
pmed1	100	5	0,12	0,04	0,08	0,04	0,08	0,11	0,04	0,07	0,04	0,07
pmed2		10	0,41	0,11	0,16	0,11	0,16	0,51	0,12	0,19	0,12	0,19
pmed3		10	0,33	0,09	0,14	0,08	0,13	0,42	0,11	0,16	0,11	0,16
pmed4		20	2,09	0,44	0,52	0,45	0,51	2,23	0,49	0,54	0,49	0,55
pmed5		33	4,69	0,99	0,88	0,96	0,89	6,83	1,36	1,24	1,37	1,25
pmed6	200	5	0,49	0,16	0,32	0,16	0,32	0,59	0,16	0,31	0,16	0,30
pmed7		10	2,57	0,55	0,83	0,55	0,84	1,61	0,30	0,46	0,30	0,46
pmed8		20	12,69	2,20	2,55	2,25	2,57	14,01	2,18	2,59	2,27	2,57
pmed9		40	36,85	6,20	5,34	6,34	5,40	36,03	5,31	4,72	5,60	4,66
pmed10		67	41,28	7,18	4,88	7,30	4,94	61,01	9,51	6,79	10,01	6,71
pmed11	300	5	0,90	0,27	0,48	0,27	0,48	1,18	0,32	0,58	0,32	0,56
pmed12		10	3,68	0,76	1,09	0,75	1,09	4,62	0,90	1,30	0,90	1,30
pmed13		30	37,95	6,07	6,04	6,04	6,01	66,97	9,95	9,98	9,94	9,96
pmed14		60	159,22	25,99	20,20	26,02	20,14	124,92	17,70	14,03	17,78	14,01
pmed15		100	137,11	22,82	12,94	22,91	12,90	210,22	30,41	18,08	30,96	18,03
pmed16	400	5	1,87	0,58	1,02	0,58	1,01	1,89	0,43	0,75	0,43	0,75
pmed17		10	5,34	0,98	1,41	0,99	1,41	6,47	1,09	1,55	1,09	1,54
pmed18		40	151,91	22,07	20,47	21,95	20,40	191,41	26,57	24,67	27,01	24,44
pmed19		80	248,90	36,32	24,86	37,18	24,73	341,07	44,85	31,26	45,92	31,29
pmed20		133	432,21	64,36	34,38	66,17	34,24	514,94	65,67	36,07	67,46	36,06

kontynuacja na następnej stronie

Problem			inicjalizacja zachłanna					inicjalizacja losowa				
			CPU[s]					CPU[s]				
nazwa	m	n	VNS	VNS'	VNS''	VNS' <sub>r</sub>	VNS'' <sub>r</sub>	VNS	VNS'	VNS''	VNS' <sub>r</sub>	VNS'' <sub>r</sub>
pmed21	500	5	1,91	0,52	0,85	0,52	0,85	3,13	0,77	1,27	0,77	1,28
pmed22		10	8,49	1,77	2,62	1,79	2,61	11,46	2,13	3,05	2,14	3,06
pmed23		50	417,29	59,25	51,83	60,96	51,40	340,36	45,25	38,27	45,18	38,25
pmed24		100	705,17	102,00	65,45	103,19	65,18	805,54	104,99	67,41	104,25	67,66
pmed25		167	676,77	100,52	53,58	101,24	53,39	894,65	113,27	60,93	112,15	60,96
pmed26	600	5	3,75	1,04	1,76	1,02	1,75	4,02	0,91	1,55	0,92	1,56
pmed27		10	16,46	3,60	5,03	3,59	5,03	24,19	4,25	5,92	4,24	5,93
pmed28		60	404,84	56,13	46,51	56,65	46,53	831,06	112,34	92,04	112,19	92,24
pmed29		120	967,10	146,15	95,46	147,52	95,25	1425,63	182,10	120,54	182,17	120,87
pmed30		200	1053,87	155,10	77,15	156,36	77,21	1374,68	167,97	86,74	168,37	87,07
pmed31	700	5	3,78	1,10	1,72	1,08	1,73	5,77	1,36	2,10	1,35	2,08
pmed32		10	17,97	3,72	5,07	3,70	5,09	26,19	4,99	6,79	4,97	6,78
pmed33		70	851,54	121,52	92,71	123,31	92,50	1004,37	129,50	101,12	130,13	101,39
pmed34		140	1353,85	186,18	108,91	190,09	108,46	1999,12	236,82	139,66	237,67	139,91
pmed35	800	5	6,25	1,93	3,07	1,91	3,07	7,97	2,08	3,30	2,07	3,30
pmed36		10	23,65	4,72	6,43	4,71	6,44	33,83	6,36	8,72	6,34	8,71
pmed37		80	1663,19	235,02	174,90	235,33	174,84	1736,29	215,07	163,33	217,44	163,55
pmed38	900	5	10,18	3,31	5,34	3,27	5,38	10,76	2,77	4,45	2,74	4,44
pmed39		10	25,31	4,85	6,60	4,81	6,64	37,53	6,00	8,17	5,98	8,14
pmed40		90	1639,72	238,38	166,01	237,55	166,27	2199,19	262,41	189,94	262,91	191,65

Wyniki dla dużych problemów umożliwiają również porównanie wersji VNS' i VNS'' (z warunkiem dominacji). Można zaobserwować, że dla mniejszych problemów (100 lokalizacji) i problemów z małą liczbą punktów obsługi wersja VNS' osiąga krótsze czasy niż wersja VNS''. Jednak wraz ze wzrostem rozmiaru problemu wersja VNS'' zaczyna osiągać lepsze czasy dla instancji z większą liczbą punktów obsługi. Można przypuszczać, że przy rosnącym rozmiarze problemu większa liczba punktów obsługi może być rozumiana jako wielkość proporcjonalna do liczby wszystkich lokalizacji. W badanych problemach wersja VNS'' dla najmniejszej rozpatrywanej proporcjonalnej liczby punktów obsługi  $m/10$  osiąga krótsze czasy rozwiązania niż wersja VNS' dla problemów z co najmniej 400 lokalizacjami.

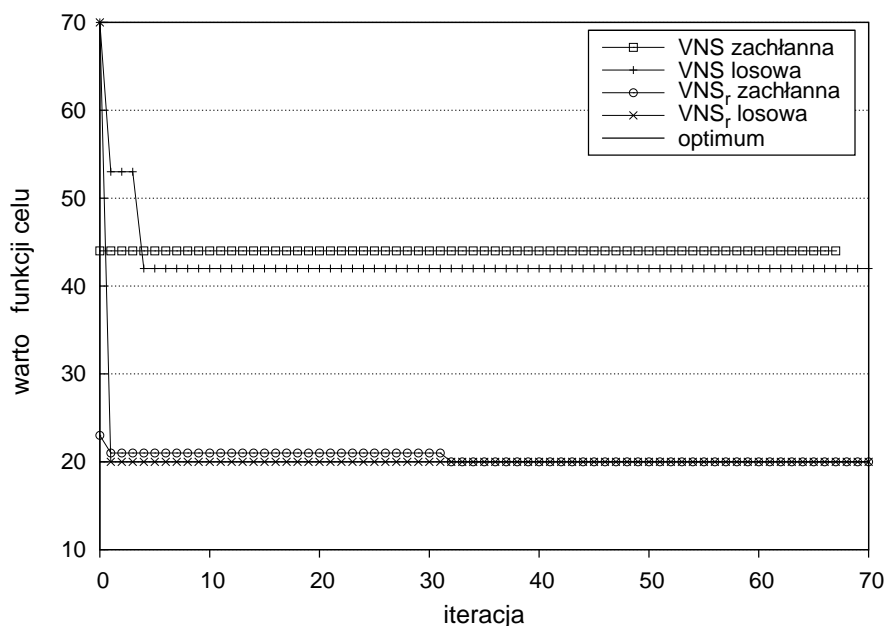
Warto również zauważyć, że ponieważ czas rozwiązania problemów rośnie szybko wraz ze wzrostem liczby punktów obsługi, to bezwzględne oszczędności czasu wersji VNS'' są znacznie większe niż te dla wersji VNS'. Inaczej mówiąc, wersja z warunkiem dominacji osiąga krótsze czasy rozwiązania dla problemów, które wykonują się długo (mają liczniejsze zbiory rozwiązań dopuszczalnych), a wersja bez warunku dominacji osiąga lepsze czasy rozwiązania dla problemów z mniejszymi zbiorami rozwiązań dopuszczalnych. To powoduje, że sumaryczny czas rozwiązania wszystkich problemów (a tym samym średni czas rozwiązania problemu z biblioteki OR) jest znacząco krótszy dla wersji VNS''. Dłuższy czas wykonania wersji VNS'' dla problemów z mniejszymi zbiorami rozwiązań dopuszczalnych wynika oczywiście z faktu, że koszt warunku dominacji przewyższa jego zysk dla tych instancji problemów. W związku z powyższym można powiedzieć, że warunek dominacji jest uzasadniony i pozwala skrócić czas rozwiązania problemów z kryterium średniej, w których liczba punktów obsługi stanowi znaczący procent wszystkich lokalizacji.

Jako następne przeanalizowane zostaną problemy z kryterium centrum. Tutaj również dodany został warunek ograniczający liczbę iteracji do 50. Taki sam limit używany był w [17], gdyż zbieżność metody VNS bez regularyzacji dla problemów z kryterium centrum jest powolna. Odstęp od rozwiązania optymalnego maleje zazwyczaj stopniowo w kolejnych iteracjach. Wyjątkiem są problemy z małą liczbą punktów obsługi lub sytuacje, w których metoda utyka szybko w minimum lokalnym. W przypadku inicjalizacji zachłannej zdarzają się przypadki, gdy rozwiązanie początkowe okazuje się najlepszym znalezionym. Nie wynika to jednak z szybkiej zbieżności metody, a właśnie z utknięcia w minimum lokalnym, które jest zazwyczaj dużo gorsze od rozwiązania optymalnego. Metoda z regularyzacją prezentuje się o wiele lepiej. Charakteryzuje się szybszą zbieżnością i często osiąga rozwiązania optymalne lub bliskie optymalnym. W tym przypadku można by nawet rozważyć zmniejszenie limitu liczby iteracji. Z drugiej strony ze względu na niewielkie bezwzględne wartości funkcji celu (w porównaniu np. do kryterium średniej) nawet małe bezwzględne odstępstwa stanowią znaczący procent wartości optymalnej. Niewielkie poprawy w dalszych iteracjach mogą więc zauważalnie zmniejszyć odstęp od rozwiązania optymalnego.

Na rysunku 8.6 przedstawiony jest przebieg metody VNS dla problemu *pmed10* z kryterium centrum (razem z rozwiązaniem początkowym). Widać, że metoda bez regularyzacji dosyć szybko zatrzymuje się w minimum lokalnym. Wersja z inicjalizacją losową poprawia wartość funkcji celu od 70 (rozwiązanie początkowe) do 42, przy czym najlepszą wartość osiąga po raz pierwszy po 4 iteracji. W przypadku inicjalizacji zachłannej początkowe rozwiązanie o wartości funkcji celu 44 okazuje się najlepszym znalezionym. Przy obu inicjalizacjach znalezione rozwiązanie jest ponad dwa razy gorsze od rozwiązania optymalnego (optymalna wartość f. celu wynosi 20). Metoda z regularyzacją przy inicjalizacji losowej, zaczynając od wartości początkowej 70, osiąga wartość optymalną już w 1 iteracji. Wersja z inicjalizacją zachłanną rozpoczyna od wartości 24 i również znajduje rozwiązanie optymalne, przy czym osiąga je w 32 iteracji. Warto zauważyć, że już po 1 iteracji wartość funkcji celu znalezionego rozwiązania różni się tylko o 1 od wartości optymalnej. Stanowi to jednak 5% odstęp, co można uznać za wartość znaczącą, zwłaszcza porównując z wynikami uzyskiwanymi dla kryterium średniej.

W tabeli 8.5 przedstawiono czasy rozwiązań problemów z kryterium centrum. Widać, że czasy rozwiązania za pomocą zmodyfikowanych wersji są znacznie krótsze od wersji oryginalnej. Duże różnice występują też między wersjami z regularyzacją i bez niej.

Dla metody bez regularyzacji, której rozwiązania są identyczne jak metody oryginalnej, poprawa jest nawet lepsza niż w przypadku kryterium średniej mimo tego, że czasy rozwiązania (również wersji oryginalnej) są zdecydowanie mniejsze niż problemów z kryterium średniej. Potwierdza to oczekiwania przedstawione w opisie modyfikacji (patrz 8.2.1). Zdecydowana większość problemów została rozwiązana w czasie o ponad rząd wielkości mniejszym (z nielicznymi wyjątkami dotyczącymi problemów z małą liczbą punktów obsługi). Duża część problemów ma czasy rozwiązania ponad 30, a nawet ponad 40 razy krótsze niż metoda



Rysunek 8.6. Przebieg metody VNS dla problemu *pmed10* z kryterium centrum.

oryginalna przy inicjalizacji zachłannej. Przy inicjalizacji losowej poprawa jest jeszcze większa i dla niektórych problemów przekracza nawet dwa rzędy wielkości. Średnio dla 40 badanych problemów wersje VNS', VNS'' pozwoliły na skrócenie czasu odpowiednio 29,4 i 22,6 razy przy inicjalizacji zachłannej oraz 43,9 i 35,4 razy przy inicjalizacji losowej.

Metoda z regularyzacją ma czasy znacznie dłuższe niż zmodyfikowana metoda bez regularyzacji, ale i tak zazwyczaj kilkukrotnie krótsze niż metoda oryginalna. Dłuższe czasy wynikają z większej intensywności przeszukiwania w poszczególnych iteracjach, co może też prowadzić do wyboru i losowania innych rozwiązań w kolejnych iteracjach. Gorsze czasy są więc skutkiem przeszukiwania większej liczby rozwiązań, co z kolei wpływa pozytywnie na jakość uzyskiwanych rozwiązań. W większości przypadków czasy rozwiązania są ok. 3–4 razy krótsze niż metody oryginalnej. Zdarzają się również czasy zbliżone do czasów metody oryginalnej (zazwyczaj dla małych problemów, czy małej liczby punktów obsługi), a w jednym przypadku czas nowej metody jest nawet nieco dłuższy niż oryginalnie (*pmed21*). Oczywiście są również problemy, gdzie nowa metoda osiąga czas ok. 6-krotnie krótszy, a w jednym przypadku nawet 12-krotnie krótszy (*pmed22*). Średnio dla 40 badanych problemów wersje VNS'\_r, VNS''\_r pozwoliły na skrócenie czasu odpowiednio 3,4 i 3 razy przy inicjalizacji zachłannej oraz 3,6 i 3,3 razy przy inicjalizacji losowej.

Tabela 8.5: Statystyki czasowe metody VNS z agregacją OWA dla problemów z biblioteki OR z kryterium centrum (min-max, TC2)

Problem			inicjalizacja zachłanna					inicjalizacja losowa				
			CPU[s]					CPU[s]				
nazwa	m	n	VNS	VNS'	VNS''	VNS'_r	VNS''_r	VNS	VNS'	VNS''	VNS'_r	VNS''_r
pmed1	100	5	0,08	0,01	0,02	0,03	0,07	0,12	0,01	0,02	0,06	0,12

kontynuacja na następnej stronie

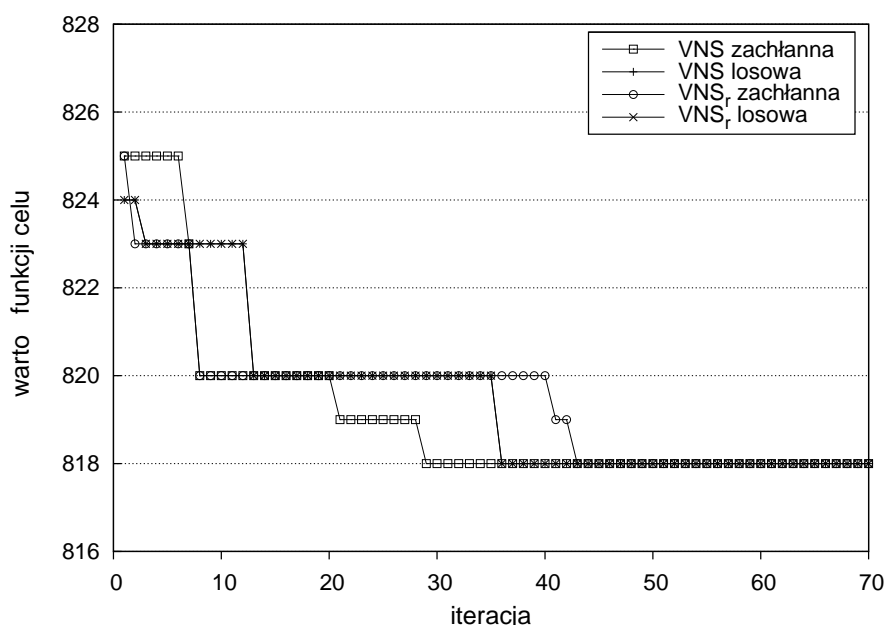
Problem			inicjalizacja zachłanna					inicjalizacja losowa				
			CPU[s]					CPU[s]				
nazwa	m	n	VNS	VNS'	VNS''	VNS' <sub>r</sub>	VNS'' <sub>r</sub>	VNS	VNS'	VNS''	VNS' <sub>r</sub>	VNS'' <sub>r</sub>
pmed2		10	0,50	0,03	0,04	0,13	0,21	0,43	0,02	0,03	0,19	0,31
pmed3		10	0,35	0,02	0,04	0,18	0,29	0,67	0,03	0,04	0,13	0,22
pmed4		20	0,99	0,05	0,06	0,40	0,51	1,73	0,05	0,08	0,24	0,30
pmed5		33	1,80	0,07	0,09	0,52	0,57	1,69	0,06	0,08	0,54	0,57
pmed6	200	5	0,34	0,04	0,07	0,11	0,21	0,81	0,09	0,14	0,11	0,22
pmed7		10	2,92	0,14	0,19	0,77	1,25	1,32	0,05	0,07	0,38	0,61
pmed8		20	5,52	0,16	0,23	1,54	1,98	5,91	0,15	0,19	1,92	2,52
pmed9		40	9,62	0,21	0,28	2,99	3,28	11,05	0,19	0,26	2,61	2,77
pmed10		67	11,93	0,63	0,84	3,83	3,20	13,31	0,47	0,62	4,88	4,15
pmed11	300	5	1,47	0,14	0,18	0,26	0,47	1,26	0,09	0,11	0,20	0,38
pmed12		10	3,17	0,17	0,25	0,60	0,97	5,52	0,21	0,28	0,56	0,86
pmed13		30	10,22	0,34	0,49	5,55	6,31	20,54	0,38	0,48	5,19	5,85
pmed14		60	32,78	0,62	0,82	8,83	8,40	34,45	0,31	0,38	9,54	8,99
pmed15		100	39,35	0,99	1,30	14,49	10,90	43,33	0,95	1,18	12,44	9,36
pmed16	400	5	1,64	0,17	0,24	0,56	0,95	0,90	0,07	0,10	0,39	0,67
pmed17		10	3,20	0,20	0,29	1,09	1,58	4,11	0,15	0,18	1,15	1,73
pmed18		40	33,64	0,94	1,25	11,71	12,32	44,52	0,82	1,04	12,66	12,82
pmed19		80	74,21	1,19	1,53	21,95	18,33	76,22	1,16	1,43	22,19	19,38
pmed20		133	95,74	2,91	3,65	37,08	25,22	106,63	2,24	2,72	34,68	24,48
pmed21	500	5	1,05	0,16	0,24	0,92	1,57	3,67	0,30	0,39	0,64	1,09
pmed22		10	19,77	0,92	1,22	1,65	2,61	5,47	0,22	0,28	1,71	2,62
pmed23		50	82,54	1,40	1,79	24,96	25,05	82,86	1,29	1,61	25,09	25,09
pmed24		100	139,13	2,78	3,48	41,93	32,48	149,90	2,43	2,97	40,43	31,27
pmed25		167	185,45	4,87	5,96	61,31	39,73	198,70	2,91	3,49	71,90	47,47
pmed26	600	5	3,53	0,39	0,56	1,86	3,35	4,70	0,42	0,57	1,24	2,11
pmed27		10	5,43	0,41	0,59	3,11	4,58	10,96	0,43	0,54	3,40	4,97
pmed28		60	142,73	2,51	3,17	46,61	43,33	143,61	2,33	2,83	43,63	40,84
pmed29		120	231,05	5,09	6,21	75,91	59,12	236,77	3,62	4,36	72,68	54,99
pmed30		200	329,18	7,55	9,10	96,96	60,79	338,07	4,14	4,88	100,22	60,36
pmed31	700	5	5,60	0,62	0,86	1,25	1,98	7,16	0,60	0,74	1,26	1,92
pmed32		10	7,56	0,64	0,93	5,43	8,05	13,35	0,56	0,68	4,60	6,44
pmed33		70	249,44	3,62	4,48	69,04	60,28	229,27	1,96	2,30	69,36	61,16
pmed34		140	390,29	7,51	9,29	120,09	86,24	396,10	3,14	3,65	119,77	86,52
pmed35	800	5	7,15	0,80	1,07	1,80	2,91	8,57	0,68	0,82	1,47	2,36
pmed36		10	22,01	1,32	1,77	6,99	9,92	14,42	0,63	0,80	7,52	10,59
pmed37		80	318,21	6,43	7,89	87,33	76,02	332,01	3,22	3,75	104,29	95,29
pmed38	900	5	10,32	1,33	1,91	2,36	3,86	5,80	0,56	0,74	2,23	3,60
pmed39		10	24,33	1,57	2,12	7,93	11,21	15,74	0,69	0,84	6,28	8,67
pmed40		90	470,70	6,76	8,20	142,85	118,83	504,65	5,62	6,60	155,93	129,78

Porównanie wersji różniących się obecnością warunku dominacji pozwala stwierdzić, że w przypadku kryterium centrum wersja VNS', bez warunku dominacji, radzi sobie lepiej niż wersja VNS''. Faktycznie dla każdego badanego problemu wersja VNS' uzyskała nie gorszy czas niż wersja VNS''. Nieco inaczej wygląda sytuacja dla metody z regularyzacją. Wersja VNS'<sub>r</sub> radzi sobie lepiej niż wersja VNS''<sub>r</sub> w nieco więcej niż połowie problemów. Widać również wyraźnie, że wersja VNS''<sub>r</sub> osiąga krótsze czasy dla problemów większych i z większą liczbą punktów obsługi. To może również potwierdzać podejrzenie, że warunek dominacji pokrywa się częściowo z modyfikacją dotyczącą relaksacji. Ponieważ w przypadku kryterium centrum dla metody bez regularyzacji warunek dotyczący relaksacji odrzuca bardzo dużą część nieperspektywicznych rozwiązań (wszystkie, które nie poprawiają największego kosztu), to

warunek dominacji zadziała w bardzo niewielu przypadkach. Z tego powodu zysk warunku dominacji będzie mniejszy niż jego koszty. W metodzie z regularyzacją, która uwzględni dodatkowo zmiany w wektorze ocen, warunek dotyczący relaksacji działa już dla mniejszej liczby rozwiązań i część rozwiązań przepuszczonych przez ten warunek jest odrzucana przez warunek dominacji. Dlatego wersja  $VNS_r''$  dla części problemów osiąga krótsze czasy niż wersja bez warunku dominacji.

Porównując czasy metody dla różnych sposobów inicjalizacji, można zauważyć, że wersja bez regularyzacji osiąga znacząco krótsze czasy rozwiązania przy inicjalizacji losowej. W przypadku regularyzacji przewaga ta nie jest taka wyraźna. Z kolei oryginalna wersja z literatury zdaje się uzyskiwać nieco lepsze czasy przy inicjalizacji zachłannej.

Dalej przedstawione są wyniki uzyskane dla problemów z kryterium zawężonej średniej. Tutaj również dodano ograniczenie w postaci maksymalnej liczby 50 iteracji z uzasadnieniem analogicznym jak w kryterium średniej. Przebieg metody VNS dla problemu *pmed10* z kryterium zawężonej średniej (bez rozwiązania początkowego) jest przedstawiony na rysunku 8.7. W przypadku inicjalizacji losowej regularyzacja nie wpłynęła na rozwiązanie tego problemu i obie wersje metody osiągają najlepsze znalezione rozwiązanie w 36 iteracji. W przypadku inicjalizacji zachłannej wersja bez regularyzacji osiąga najlepsze rozwiązanie w 29 iteracji, a wersja z regularyzacją w iteracji 43.



Rysunek 8.7. Przebieg metody VNS dla problemu *pmed10* z kryterium zawężonej średniej (bez rozwiązania początkowego).

W tabeli 8.6 przedstawiono czasy rozwiązań problemów z kryterium zawężonej średniej. Wyniki są zbliżone do rezultatów uzyskanych dla problemów z kryterium średniej. Czasy rozwiązania dla zmodyfikowanych metod są znacznie krótsze niż dla metody oryginalnej. Dotyczy to zarówno wersji bez regularyzacji, jak i z regularyzacją, przy czym ta druga ma



zazwyczaj czasy nieco gorsze niż ta pierwsza. Zysk rośnie umiarkowanie wraz ze wzrostem liczby lokalizacji oraz rośnie szybko ze wzrostem liczby punktów obsługi. Podobnie jak przy kryterium średniej dla problemów z większą liczbą punktów obsługi czas rozwiązania skrócił się około 6–13 razy przy inicjalizacji zachłannej, a przy inicjalizacji losowej różnice są jeszcze większe. Średnio na zbiorze 40 problemów czasy rozwiązania wersji VNS' i VNS'' skróciły się ok. 5,7 i 6 razy przy inicjalizacji zachłannej oraz odpowiednio 6,4 i 6,7 razy przy inicjalizacji losowej. Niewiele gorzej jest dla metody z regularyzacją. Średnie czasy rozwiązania wersji VNS'<sub>r</sub> i VNS''<sub>r</sub> skróciły się ok. 5,5 razy przy inicjalizacji zachłannej oraz odpowiednio 6 i 6,3 razy przy inicjalizacji losowej.

Podobnie jak dla kryterium średniej wygląda również porównanie wersji różniących się obecnością warunku dominacji. Wersja VNS'/VNS'<sub>r</sub>, bez warunku dominacji, osiąga krótsze czasy dla problemów z małą liczbą punktów obsługi lub o mniejszym rozmiarze. W miarę wzrostu rozmiaru dla problemów z większą liczbą punktów obsługi lepsze czasy zaczyna osiągać wersja VNS''/VNS''<sub>r</sub>.

Tabela 8.6: Statystyki czasowe metody VNS z agregacją OWA dla problemów z biblioteki OR z kryterium zawężonej średniej (TC4)

Problem			inicjalizacja zachłanna					inicjalizacja losowa				
			CPU[s]					CPU[s]				
nazwa	m	n	VNS	VNS'	VNS''	VNS' <sub>r</sub>	VNS'' <sub>r</sub>	VNS	VNS'	VNS''	VNS' <sub>r</sub>	VNS'' <sub>r</sub>
pmed1	100	5	0,12	0,04	0,08	0,04	0,09	0,10	0,03	0,06	0,03	0,06
pmed2		10	0,53	0,13	0,21	0,12	0,20	0,56	0,13	0,21	0,13	0,21
pmed3		10	0,47	0,10	0,17	0,11	0,17	0,50	0,11	0,19	0,11	0,18
pmed4		20	1,66	0,32	0,38	0,41	0,47	1,95	0,36	0,43	0,38	0,44
pmed5		33	5,23	0,90	0,92	1,05	1,01	4,37	0,75	0,81	0,89	0,88
pmed6	200	5	0,51	0,17	0,32	0,16	0,33	0,46	0,12	0,22	0,14	0,26
pmed7		10	1,51	0,32	0,51	0,33	0,52	1,80	0,32	0,52	0,33	0,52
pmed8		20	8,68	1,44	1,69	1,43	1,78	7,53	1,11	1,34	1,30	1,50
pmed9		40	33,87	5,48	4,93	5,79	5,71	43,45	6,57	6,34	7,16	6,50
pmed10		67	33,15	5,15	3,98	7,39	6,14	38,66	5,53	4,49	6,64	4,99
pmed11	300	5	0,93	0,27	0,47	0,27	0,50	1,03	0,27	0,48	0,27	0,47
pmed12		10	3,09	0,63	0,93	0,64	0,98	4,59	0,87	1,31	0,89	1,32
pmed13		30	40,64	6,30	6,65	6,53	7,09	65,82	9,89	10,32	10,18	10,44
pmed14		60	126,86	17,96	15,38	16,46	14,88	223,35	31,11	26,47	33,83	27,93
pmed15		100	128,13	16,94	11,40	21,45	14,95	159,66	21,15	14,21	24,31	15,50
pmed16	400	5	1,77	0,53	0,94	0,58	1,06	1,86	0,43	0,77	0,44	0,77
pmed17		10	6,22	1,11	1,58	1,07	1,57	7,13	1,30	1,89	1,26	1,81
pmed18		40	160,57	22,43	21,77	23,73	25,08	174,49	22,91	22,35	29,18	28,35
pmed19		80	283,98	37,88	28,40	34,95	28,23	300,90	35,09	26,21	47,67	35,15
pmed20		133	324,18	40,77	25,88	43,30	28,59	386,91	42,61	27,75	48,49	30,34
pmed21	500	5	1,91	0,52	0,84	0,52	0,86	2,78	0,63	1,02	0,64	1,02
pmed22		10	13,19	2,69	3,90	2,70	4,05	11,43	2,09	2,98	2,10	2,97
pmed23		50	401,00	56,58	52,44	47,07	47,59	470,75	61,36	56,66	45,25	40,29
pmed24		100	632,83	83,78	60,46	84,38	64,45	847,58	100,71	73,09	100,53	71,46
pmed25		167	682,74	80,82	52,11	103,02	66,77	996,25	105,39	67,62	117,99	72,30
pmed26	600	5	4,02	1,12	1,87	1,12	1,93	4,04	0,92	1,56	1,00	1,69
pmed27		10	15,12	3,25	4,60	3,28	4,72	18,45	3,37	4,69	3,39	4,71
pmed28		60	409,60	52,75	46,10	59,31	56,23	563,44	67,81	59,97	86,29	75,40
pmed29		120	846,46	98,45	73,45	114,48	90,30	1209,52	131,70	98,01	143,97	103,15
pmed30		200	1221,11	142,79	88,61	168,83	107,95	1339,81	130,77	79,45	184,34	106,28

kontynuacja na następnej stronie

Problem			inicjalizacja zachłanna					inicjalizacja losowa				
			CPU[s]					CPU[s]				
nazwa	m	n	VNS	VNS'	VNS''	VNS' <sub>r</sub>	VNS'' <sub>r</sub>	VNS	VNS'	VNS''	VNS' <sub>r</sub>	VNS'' <sub>r</sub>
pmed31	700	5	4,75	1,47	2,32	1,46	2,40	6,90	1,80	2,81	1,80	2,84
pmed32		10	17,72	3,67	5,03	3,67	5,22	27,91	5,32	7,33	4,64	6,37
pmed33		70	1238,82	173,00	140,98	169,79	147,69	1138,02	138,98	115,45	143,47	116,80
pmed34		140	1723,28	221,96	145,12	186,20	128,69	2243,60	226,04	153,60	226,55	146,82
pmed35	800	5	4,88	1,44	2,29	1,44	2,36	7,63	1,91	3,09	1,91	3,12
pmed36		10	24,36	4,97	6,92	4,97	7,16	42,92	7,88	10,79	6,95	9,56
pmed37		80	1191,32	152,33	122,22	167,36	143,14	1563,34	172,78	139,49	184,20	145,44
pmed38	900	5	9,61	3,01	4,86	2,99	5,00	10,70	2,76	4,38	2,74	4,40
pmed39		10	31,98	6,68	9,11	6,68	9,50	37,24	6,05	8,16	6,02	8,18
pmed40		90	1883,77	249,09	187,82	309,92	251,56	2325,65	259,54	197,89	307,23	230,29

W pracy zbadano także jakość rozwiązań uzyskiwanych za pomocą metody przeszukiwania zmiennego sąsiedztwa. Sprawdzone bezpośrednie zmiany jakościowe wynikające z dodania regularyzacji, ale także pośredni wpływ zmian wydajnościowych. Te drugie nie wpływają bezpośrednio na jakość rozwiązania, ale pozwalają, jak pokazują wyniki, na znaczne skrócenie czasu obliczeń. To sugeruje z kolei możliwość uzyskania lepszych rozwiązań w tym samym czasie poprzez zwiększenie maksymalnej dopuszczalnej liczby iteracji lub kilkukrotne uruchomienie metody. Aby to zweryfikować, każdy problem z biblioteki OR został naliczony 10 razy (z limitem 50 iteracji), a uzyskane wyniki porównane z rozwiązaniami optymalnymi lub najlepszymi znanymi dotychczas rozwiązaniami (w przypadku problemów z kryterium zawężonej średniej). Na tej podstawie wyznaczono statystyki jakościowe metody VNS. Podobna procedura testowa była zastosowana w pracy [81], w której badano dwie heurystyki bazujące na algorytmach genetycznych do rozwiązywania dyskretnych problemów z kryterium porządkowym (jedna HGA1 stosowała kodowanie binarne, a druga HGA2 całkowitoliczbowe). W pracy [81] przeprowadzono także porównanie metod HGA1 i HGA2 z metodą VNS z pracy [18], które wskazywało na lepszą jakość rozwiązań uzyskiwanych za pomocą metod HGA1 i HGA2. Należy jednak zaznaczyć, że w pracy [18] metoda VNS była naliczana tylko jednokrotnie. W niniejszej pracy porównano wyniki jakościowe zmodyfikowanej metody VNS z wynikami metod HGA1 i HGA2 przy tej samej (10-krotnej) liczbie naliczeń. Pozwoliło to ocenić dokładność metody VNS z warunkiem regularyzacji i większą liczbą iteracji w odniesieniu do innych metod przybliżonych z literatury.

Jako jedno z kryteriów jakościowych zastosowano odstęp od rozwiązania optymalnego (najlepszego) wyznaczany zgodnie ze wzorem (8.1). Rozwiązania optymalne dla problemów z kryterium średniej zaczerpnięte zostały z biblioteki OR [7] (problemy z biblioteki OR były rozwiązywane np. w [6]), a dla problemów z kryterium centrum z [21]. W przypadku problemów z kryterium zawężonej średniej nie są znane rozwiązania optymalne. Z tego powodu do zbadania jakości metody VNS wykorzystano najlepsze znane dotychczas rozwiązania, które zaczerpnięto ze wspomnianej wcześniej pracy dotyczącej heurystyk opartych na algorytmach genetycznych [81].

W tabeli 8.7 przedstawiono statystyki jakościowe rozwiązań dla problemów z kryterium

średniej (w tym przypadku regularyzacja nie jest aktywna, więc wyniki dotyczą wersji z obu grup). Pierwsze 4 kolumny charakteryzują problem: pierwsza zawiera identyfikator problemu z biblioteki OR, druga liczbę lokalizacji, trzecia liczbę punktów obsługi, a czwarta wartość funkcji celu rozwiązania optymalnego. W następnych kolumnach dla każdego z dwóch sposobów inicjalizacji wyznaczono liczbę naliczeń, w których udało się znaleźć rozwiązanie optymalne oraz maksymalny, średni i minimalny odstęp uzyskany w ramach 10 naliczeń. Jak widać uzyskiwane rozwiązania charakteryzują się bardzo dobrą jakością. Dla prawie wszystkich problemów, z wyjątkiem jednego problemu przy inicjalizacji zachłannej i dwóch przy inicjalizacji losowej, nawet maksymalny odstęp jest mniejszy od 1%, a średni (uśredniony dodatkowo po 40 problemach) dla obu sposobów inicjalizacji jest mniejszy od 0,1%. Patrząc na odstęp minimalny widać, że przy inicjalizacji zachłannej udało się znaleźć rozwiązanie optymalne dla 34 z 40 problemów, a przy inicjalizacji losowej optymalnie rozwiązano 36 problemów. Zbliżony jest również odstęp minimalny uśredniony po wszystkich problemach, który przy inicjalizacji zachłannej wynosi 0,013%, a przy inicjalizacji losowej 0,01%.

Dla porównania heurystyki HGA1 i HGA2 z pracy [81] znalazły odpowiednio 24 i 22 rozwiązania optymalne, a ich minimalne odchylenie uśrednione po 40 problemach wynosiło 0,111% i 0,176%, czyli było ok. 10 razy gorsze. Różnica w odchyleniu nie wynika tylko z mniejszej liczby znalezionych rozwiązań optymalnych. Uśredniając minimalny odstęp tylko po problemach nierozwiązanych optymalnie, otrzymujemy dla metody VNS z inicjalizacją zachłanną i losową odpowiednio 0,087% i 0,105%, a dla metod HGA1, HGA2 odpowiednio 0,277% oraz 0,391%.

Tabela 8.7: Statystyki jakościowe metody VNS z agregacją OWA dla problemów z biblioteki OR z kryterium średniej (TC1)

Problem				inicjalizacja zachłanna				inicjalizacja losowa			
				# opt.	odstęp [%]			# opt.	odstęp [%]		
nazwa	m	n	optimum	znal.	max.	śr.	min.	znal.	max.	śr.	min.
pmed1	100	5	5819	10	0	0	0	10	0	0	0
pmed2		10	4093	6	0,293	0,117	0	8	0,293	0,059	0
pmed3		10	4250	10	0	0	0	10	0	0	0
pmed4		20	3034	10	0	0	0	9	0,396	0,040	0
pmed5		33	1355	10	0	0	0	9	0,221	0,022	0
pmed6	200	5	7824	10	0	0	0	10	0	0	0
pmed7		10	5631	3	0,249	0,174	0	4	0,249	0,107	0
pmed8		20	4445	10	0	0	0	10	0	0	0
pmed9		40	2734	4	0,622	0,300	0	5	0,475	0,212	0
pmed10		67	1255	7	0,080	0,024	0	10	0	0	0
pmed11	300	5	7696	10	0	0	0	10	0	0	0
pmed12		10	6634	10	0	0	0	10	0	0	0
pmed13		30	4374	10	0	0	0	10	0	0	0
pmed14		60	2968	0	0,034	0,034	0,034	5	0,101	0,024	0

kontynuacja na następnej stronie

				inicjalizacja zachłanna				inicjalizacja losowa			
Problem				# opt.	odstęp [%]			# opt.	odstęp [%]		
nazwa	m	n	optimum	znal.	max.	śr.	min.	znal.	max.	śr.	min.
pmed15		100	1729	2	0,116	0,052	0	2	0,174	0,064	0
pmed16	400	5	8162	10	0	0	0	10	0	0	0
pmed17		10	6999	10	0	0	0	6	0,057	0,023	0
pmed18		40	4809	2	0,042	0,033	0	2	0,125	0,052	0
pmed19		80	2845	0	0,422	0,176	0,035	1	0,246	0,116	0
pmed20		133	1789	5	0,168	0,050	0	4	0,224	0,084	0
pmed21	500	5	9138	10	0	0	0	10	0	0	0
pmed22		10	8579	5	1,049	0,516	0	9	1,049	0,105	0
pmed23		50	4619	10	0	0	0	10	0	0	0
pmed24		100	2961	10	0	0	0	5	0,371	0,128	0
pmed25		167	1828	1	0,274	0,120	0	0	0,602	0,263	0,109
pmed26	600	5	9917	10	0	0	0	7	0,071	0,021	0
pmed27		10	8307	10	0	0	0	9	0,036	0,004	0
pmed28		60	4498	0	0,245	0,138	0,044	1	0,222	0,080	0
pmed29		120	3033	1	0,099	0,053	0	1	0,429	0,148	0
pmed30		200	1989	0	0,452	0,332	0,201	0	0,754	0,543	0,251
pmed31	700	5	10086	10	0	0	0	7	0,010	0,003	0
pmed32		10	9297	4	0,043	0,026	0	8	0,043	0,009	0
pmed33		70	4700	1	0,255	0,177	0	4	0,340	0,074	0
pmed34		140	3013	0	0,398	0,285	0,166	1	0,332	0,163	0
pmed35	800	5	10400	10	0	0	0	9	1,163	0,116	0
pmed36		10	9934	10	0	0	0	6	0,564	0,108	0
pmed37		80	5057	3	0,119	0,077	0	0	0,119	0,065	0,020
pmed38	900	5	11060	10	0	0	0	9	0,407	0,041	0
pmed39		10	9423	10	0	0	0	10	0	0	0
pmed40		90	5128	0	0,195	0,111	0,039	0	0,215	0,099	0,039
Średnia					0,129	0,070	0,013		0,232	0,069	0,010

Zaprezentowane wyniki potwierdzają przypuszczenie, że w przypadku problemów z kryterium średniej metoda VNS umożliwia uzyskiwanie bardzo dobrych jakościowo rozwiązań — optymalnych lub minimalnie gorszych od optymalnych. Dotyczy to obu typów inicjalizacji.

W tabelach 8.8 i 8.9 przedstawiono wyniki jakościowe dla problemów z kryterium centrum.

Tabela 8.8 zawiera wyniki jakościowe metody bez regularyzacji. Tutaj wyniki są znacznie gorsze niż dla kryterium średniej. Zdarzają się rozwiązania ponad dwukrotnie, a nawet trzykrotnie gorsze od optymalnego. Średni odstęp uśredniony po wszystkich problemach wynosi ok. 45–50%. Rzadko znajdowane są również rozwiązania optymalne, które przy obu sposobach inicjalizacji osiągnięto 8 razy. Osiągnięte rozwiązania optymalne dotyczą zazwyczaj problemów mniejszych i z małą liczbą punktów obsługi. Uśredniony minimalny błąd wyniósł przy inicjalizacji zachłannej 44,06%, a przy inicjalizacji losowej 31,9%. Dla problemów z kryterium centrum heurystyki HGA1 i HGA2 z [81] radzą sobie nieco lepiej, rozwiązując optymalnie odpowiednio 11 i 8 problemów. Uśredniony odstęp minimalny wyniósł w ich

przypadku 28,39% oraz 24,79%. Uzyskane wyniki pokazują, że jakość rozwiązań metody bez regularyzacji dla problemów z kryterium centrum nie jest zadowalająca.

Tabela 8.8: Statystyki jakościowe metody VNS z agregacją OWA bez regularyzacji dla problemów z biblioteki OR z kryterium centrum (TC2)

Problem				inicjalizacja zachłanna				inicjalizacja losowa			
				# opt. znal.	odstęp [%]			# opt. znal.	odstęp [%]		
nazwa	m	n	optimum		max.	śr.	min.		max.	śr.	min.
pmed1	100	5	127	5	3,15	0,63	0	5	4,72	0,87	0
pmed2		10	98	0	6,12	4,29	4,08	1	5,10	3,78	0
pmed3		10	93	0	17,20	6,88	3,23	0	7,53	4,30	2,15
pmed4		20	74	0	13,51	8,92	2,70	0	14,86	8,38	2,70
pmed5		33	48	2	39,58	13,75	0	0	52,08	15,83	4,17
pmed6	200	5	84	2	9,52	7,02	0	3	8,33	4,29	0
pmed7		10	64	1	7,81	5,00	0	0	14,06	9,84	4,69
pmed8		20	55	0	30,91	28,00	14,55	0	30,91	19,82	3,64
pmed9		40	37	0	97,30	60,27	37,84	0	56,76	40,81	27,03
pmed10		67	20	0	120,00	120,00	120,00	0	150,00	90,00	55,00
pmed11	300	5	59	0	8,47	3,73	1,69	2	15,25	5,93	0
pmed12		10	51	0	50,98	15,49	1,96	0	17,65	10,59	3,92
pmed13		30	36	0	52,78	52,78	52,78	0	63,89	38,61	13,89
pmed14		60	26	0	130,77	86,15	57,69	0	76,92	64,62	42,31
pmed15		100	18	0	116,67	109,44	100,00	0	155,56	107,22	77,78
pmed16	400	5	47	10	0	0	0	5	17,02	2,55	0
pmed17		10	39	0	15,38	11,79	2,56	0	15,38	9,74	5,13
pmed18		40	28	0	78,57	78,57	78,57	0	78,57	53,57	42,86
pmed19		80	18	0	72,22	72,22	72,22	0	94,44	77,78	66,67
pmed20		133	13	0	146,15	146,15	146,15	0	207,69	150,77	115,38
pmed21	500	5	40	1	20,00	11,75	0	2	7,50	4,50	0
pmed22		10	38	0	18,42	12,63	5,26	0	23,68	16,84	7,89
pmed23		50	22	0	68,18	49,09	45,45	0	68,18	59,55	45,45
pmed24		100	15	0	126,67	116,67	100,00	0	106,67	88,00	73,33
pmed25		167	11	0	145,45	145,45	145,45	0	227,27	130,00	109,09
pmed26	600	5	38	0	5,26	4,47	2,63	0	15,79	6,32	2,63
pmed27		10	32	0	12,50	9,69	9,38	0	21,88	13,44	9,38
pmed28		60	18	0	77,78	63,89	61,11	0	94,44	62,22	50,00
pmed29		120	13	0	176,92	150,77	92,31	0	176,92	112,31	76,92
pmed30		200	9	0	222,22	222,22	222,22	0	222,22	176,67	122,22
pmed31	700	5	30	2	6,67	5,33	0	0	10,00	6,33	3,33
pmed32		10	29	0	13,79	12,07	6,90	0	20,69	11,03	6,90
pmed33		70	15	0	113,33	76,67	60,00	0	100,00	76,67	53,33
pmed34		140	11	0	145,45	145,45	145,45	0	145,45	132,73	118,18
pmed35	800	5	30	0	20,00	12,33	3,33	1	20,00	6,67	0
pmed36		10	27	0	11,11	11,11	11,11	0	22,22	15,93	7,41
pmed37		80	15	0	73,33	73,33	73,33	0	93,33	73,33	60,00

kontynuacja na następnej stronie

				inicjalizacja zachłanna				inicjalizacja losowa			
Problem				# opt. znal.	odstęp [%]			# opt. znal.	odstęp [%]		
nazwa	m	n	optimum		max.	śr.	min.		max.	śr.	min.
pmed38	900	5	29	2	6,90	3,10	0	1	41,38	17,93	0
pmed39		10	23	0	17,39	15,65	13,04	0	21,74	14,78	8,70
pmed40		90	13	0	92,31	71,54	69,23	0	84,62	72,31	53,85
Średnia					59,77	51,11	44,06		65,27	45,42	31,90

Zdecydowanie lepsze rezultaty osiągnęła metoda z regularyzacją, której wyniki zaprezentowane są w tabeli 8.9. We wszystkich problemach, poza jednym wyjątkiem, nawet maksymalny odstęp jest mniejszy od 10%, a średni odstęp (uśredniony dodatkowo po 40 problemach) dla obu sposobów inicjalizacji wynosi ok. 2,5%. Często znajdowane są również rozwiązania optymalne, gdyż w ramach każdego sposobu inicjalizacji rozwiązano optymalnie 30 z 40 problemów. Uśredniony minimalny odstęp wyniósł przy inicjalizacji zachłannej 1,21%, a przy inicjalizacji losowej 1,27%. W porównaniu do metody bez regularyzacji zmalał więc ok. 25–30 razy, a w stosunku do metod HGA1 i HGA2 ok. 20 razy. Porównując uzyskane wyniki z wynikami dla kryterium średniej, nadal widać dużą różnicę, gdyż uśredniony odstęp minimalny jest ok. dwa rzędy wielkości gorszy dla kryterium centrum. Należy jednak pamiętać o odmiennej specyfice kryterium centrum, gdzie wartość funkcji celu zależy tylko od pojedynczej oceny. Można więc uznać, że metoda z regularyzacją osiąga zadowalającą jakość rozwiązań dla problemów z kryterium centrum.

Tabela 8.9: Statystyki jakościowe metody VNS z agregacją OWA z regularyzacją dla problemów z biblioteki OR z kryterium centrum (TC2)

				inicjalizacja zachłanna				inicjalizacja losowa			
Problem				# opt. znal.	odstęp [%]			# opt. znal.	odstęp [%]		
nazwa	m	n	optimum		max.	śr.	min.		max.	śr.	min.
pmed1	100	5	127	8	3,15	0,39	0	9	0,79	0,08	0
pmed2		10	98	0	5,10	3,27	2,04	2	5,10	3,27	0
pmed3		10	93	0	5,38	2,26	1,08	0	3,23	1,94	1,08
pmed4		20	74	8	1,35	0,27	0	4	6,76	1,89	0
pmed5		33	48	10	0	0	0	10	0	0	0
pmed6	200	5	84	3	4,76	2,74	0	3	8,33	3,45	0
pmed7		10	64	7	4,69	1,25	0	3	6,25	3,13	0
pmed8		20	55	5	3,64	1,45	0	5	3,64	1,82	0
pmed9		40	37	8	5,41	0,81	0	10	0	0	0
pmed10		67	20	10	0	0	0	10	0	0	0
pmed11	300	5	59	1	1,69	1,53	0	1	5,08	1,86	0
pmed12		10	51	1	3,92	2,94	0	1	1,96	1,76	0
pmed13		30	36	0	2,78	2,78	2,78	3	2,78	1,94	0
pmed14		60	26	4	3,85	2,31	0	5	3,85	1,92	0
pmed15		100	18	10	0	0	0	10	0	0	0

kontynuacja na następnej stronie

				inicjalizacja zachłanna				inicjalizacja losowa			
Problem				# opt.	odstęp [%]			# opt.	odstęp [%]		
nazwa	m	n	optimum	znal.	max.	śr.	min.	znal.	max.	śr.	min.
pmed16	400	5	47	1	2,13	1,91	0	9	2,13	0,21	0
pmed17		10	39	10	0	0	0	7	2,56	0,77	0
pmed18		40	28	3	3,57	2,50	0	4	3,57	2,14	0
pmed19		80	18	0	5,56	5,56	5,56	0	5,56	5,56	5,56
pmed20		133	13	1	7,69	6,92	0	3	7,69	5,38	0
pmed21	500	5	40	10	0	0	0	7	2,50	0,75	0
pmed22		10	38	0	5,26	3,95	2,63	0	5,26	3,42	2,63
pmed23		50	22	0	4,55	4,55	4,55	0	4,55	4,55	4,55
pmed24		100	15	1	6,67	6,00	0	5	6,67	3,33	0
pmed25		167	11	10	0	0	0	8	9,09	1,82	0
pmed26	600	5	38	4	5,26	1,84	0	6	5,26	1,58	0
pmed27		10	32	10	0	0	0	10	0	0	0
pmed28		60	18	5	5,56	2,78	0	1	5,56	5,00	0
pmed29		120	13	10	0	0	0	9	7,69	0,77	0
pmed30		200	9	0	11,11	11,11	11,11	0	11,11	11,11	11,11
pmed31	700	5	30	10	0	0	0	9	3,33	0,33	0
pmed32		10	29	1	6,90	3,45	0	0	3,45	3,45	3,45
pmed33		70	15	0	6,67	6,67	6,67	0	6,67	6,67	6,67
pmed34		140	11	10	0	0	0	10	0	0	0
pmed35	800	5	30	10	0	0	0	9	3,33	0,33	0
pmed36		10	27	2	3,70	2,96	0	0	3,70	3,70	3,70
pmed37		80	15	3	6,67	4,67	0	1	6,67	6,00	0
pmed38	900	5	29	10	0	0	0	10	0	0	0
pmed39		10	23	0	4,35	4,35	4,35	0	4,35	4,35	4,35
pmed40		90	13	0	7,69	7,69	7,69	0	7,69	7,69	7,69
Średnia					3,48	2,47	1,21		4,15	2,55	1,27

W tabelach 8.10 i 8.11 przedstawiono wyniki jakościowe dla problemów z kryterium zawężonej średniej. Jak wspomniano wcześniej rozwiązania optymalne dla tych problemów nie są znane. Uzyskane rezultaty zostały odniesione do najlepszych znanych rozwiązań (kolumna NZR). W związku z tym w tabelach występują odstęp ujemne, które oznaczają, że metoda VNS w zaprojektowanym eksperymencie zdołała poprawić dotychczasowe rozwiązania. W tabelach znajduje się również informacja o tym ile razy udało się poprawić rozwiązanie w ramach danej instancji problemu (obok liczby określającej ile razy udało się osiągnąć najlepsze znane rozwiązanie).

Statystyki jakościowe metody bez regularyzacji dla problemów z kryterium zawężonej średniej są zaprezentowane w tabeli 8.10. Uzyskane wyniki świadczą o tym, że jakość otrzymywanych rozwiązań jest bardzo dobra. W przypadku inicjalizacji zachłannej dla 31 problemów osiągnięto najlepsze znane rozwiązanie, a dla 8 problemów je poprawiono. To oznacza, że tylko w przypadku 1 problemu wyniki są gorsze niż znane najlepsze rozwiązania (odstęp minimalny dla tego problemu jest znacznie mniejszy niż 0,1%). Jeszcze lepiej wygląda

sytuacja przy inicjalizacji losowej. Tutaj również osiągnięto najlepsze znane rozwiązania 31 problemów, zresztą tych samych co przy inicjalizacji zachłannej. W pozostałych 9 problemach poprawiono najlepsze rozwiązania. Rozwiązania 4 problemów poprawiono w minimalnie lepszym stopniu niż przy inicjalizacji zachłannej. Oznacza to, że metoda VNS bez regularyzacji z inicjalizacją losową w ogólności uzyskała lepsze wyniki niż znane dotychczas rozwiązania (dla każdego problemu znalazła nie gorsze rozwiązanie). Metody HGA1 i HGA2 nie poprawiły żadnego najlepszego dotychczasowego rozwiązania i osiągnęły odpowiednio 31 i 25 najlepszych rozwiązań. Po uwzględnieniu poprawionych przez metodę VNS rozwiązań będzie to odpowiednio 27 i 22 najlepszych rozwiązań.

Tabela 8.10: Statystyki jakościowe metody VNS z agregacją OWA bez regularyzacji dla problemów z biblioteki OR z kryterium zawężonej średniej (TC4)

Problem				inicjalizacja zachłanna				inicjalizacja losowa			
				# opt./ popr.	odstęp [%]			# opt./ popr.	odstęp [%]		
nazwa	m	n	NZR		max.	śr.	min.		max.	śr.	min.
pmed1	100	5	4523	9/0	0,022	0,002	0	9/0	0,022	0,002	0
pmed2		10	2987	10/0	0	0	0	10/0	0	0	0
pmed3		10	3067	9/0	0,228	0,023	0	10/0	0	0	0
pmed4		20	2137	10/0	0	0	0	10/0	0	0	0
pmed5		33	818	10/0	0	0	0	10/0	0	0	0
pmed6	200	5	6064	2/0	0,247	0,198	0	6/0	0,247	0,099	0
pmed7		10	4206	10/0	0	0	0	10/0	0	0	0
pmed8		20	3182	9/0	0,063	0,006	0	9/0	0,063	0,006	0
pmed9		40	1807	9/0	0,111	0,011	0	10/0	0	0	0
pmed10		67	818	10/0	0	0	0	10/0	0	0	0
pmed11	300	5	5979	10/0	0	0	0	8/0	0,017	0,003	0
pmed12		10	5021	10/0	0	0	0	10/0	0	0	0
pmed13		30	3133	10/0	0	0	0	9/0	0,351	0,035	0
pmed14		60	1949	0/10	-0,154	-0,154	-0,154	0/10	-0,154	-0,154	-0,154
pmed15		100	1133	1/9	0	-0,265	-0,353	0/10	-0,177	-0,318	-0,353
pmed16	400	5	6341	10/0	0	0	0	9/0	0,394	0,039	0
pmed17		10	5381	3/0	0,595	0,416	0	9/0	0,595	0,059	0
pmed18		40	3437	9/0	0,029	0,003	0	10/0	0	0	0
pmed19		80	1922	2/7	0,052	-0,036	-0,104	3/6	0,156	-0,021	-0,104
pmed20		133	1146	4/0	0,349	0,148	0	5/0	0,349	0,140	0
pmed21	500	5	7245	10/0	0	0	0	10/0	0	0	0
pmed22		10	6685	2/0	0,957	0,534	0	5/0	1,032	0,223	0
pmed23		50	3306	10/0	0	0	0	4/0	0,060	0,024	0
pmed24		100	2004	1/9	0	-0,125	-0,200	0/10	-0,050	-0,145	-0,200
pmed25		167	1148	4/0	0,174	0,070	0	3/0	0,784	0,157	0
pmed26	600	5	7787	10/0	0	0	0	4/0	0,848	0,258	0
pmed27		10	6444	10/0	0	0	0	10/0	0	0	0
pmed28		60	3202	0/0	0,468	0,222	0,031	0/1	0,437	0,122	-0,062
pmed29		120	2005	1/8	0,150	-0,140	-0,299	1/9	0	-0,209	-0,349

kontynuacja na następnej stronie



				inicjalizacja zachłanna				inicjalizacja losowa			
Problem				# opt./ popr.	odstęp [%]			# opt./ popr.	odstęp [%]		
nazwa	m	n	NZR		max.	śr.	min.		max.	śr.	min.
pmed30		200	1295	0/5	0,386	0,046	-0,077	0/1	0,618	0,185	-0,077
pmed31	700	5	8046	10/0	0	0	0	8/0	0,124	0,025	0
pmed32		10	7278	1/0	0,027	0,025	0	5/0	0,027	0,014	0
pmed33		70	3403	10/0	0	0	0	3/0	0,294	0,100	0
pmed34		140	2023	0/7	0,099	-0,059	-0,148	0/5	0,247	-0,010	-0,247
pmed35	800	5	8191	10/0	0	0	0	9/0	1,013	0,101	0
pmed36		10	7796	4/0	0,308	0,185	0	7/0	0,693	0,168	0
pmed37		80	3598	1/0	0,083	0,044	0	1/0	0,195	0,097	0
pmed38	900	5	8720	10/0	0	0	0	9/0	0,780	0,078	0
pmed39		10	7360	10/0	0	0	0	10/0	0	0	0
pmed40		90	3710	9/1	0	-0,005	-0,054	1/3	0,162	0,046	-0,081
Średnia					0,105	0,029	-0,034		0,228	0,028	-0,041

Podobne wyniki otrzymano dla metody z regularyzacją, której statystyki jakościowe są zaprezentowane w tabeli 8.11. W przypadku inicjalizacji zachłannej dla 29 problemów osiągnięto najlepsze znane rozwiązanie, a dla 9 problemów je poprawiono. Tutaj w przypadku 2 problemów wyniki są gorsze niż znane najlepsze rozwiązania (odstęp minimalny dla tych problemów jest znacznie mniejszy niż 0,1%). W przypadku inicjalizacji losowej wyniki są jeszcze lepsze, gdyż dla 31 problemów osiągnięto najlepsze znane rozwiązania, a dla pozostałych 9 je poprawiono. Poprawione zostały rozwiązania tych samych problemów co przy inicjalizacji zachłannej, przy czym w przypadku 4 problemów osiągnięte rozwiązania były minimalnie lepsze przy inicjalizacji losowej. Wynika stąd, że również metoda z regularyzacją z inicjalizacją losową w ogólności uzyskała lepsze wyniki niż znane dotychczas rozwiązania (dla każdego problemu znalazła nie gorsze rozwiązanie).

Tabela 8.11: Statystyki jakościowe metody VNS z agregacją OWA z regularyzacją dla problemów z biblioteki OR z kryterium zawężonej średniej (TC4)

				inicjalizacja zachłanna				inicjalizacja losowa			
Problem				# opt./ popr.	odstęp [%]			# opt./ popr.	odstęp [%]		
nazwa	m	n	NZR		max.	śr.	min.		max.	śr.	min.
pmed1	100	5	4523	9/0	0,022	0,002	0	9/0	0,022	0,002	0
pmed2		10	2987	10/0	0	0	0	10/0	0	0	0
pmed3		10	3067	9/0	0,228	0,023	0	8/0	0,228	0,046	0
pmed4		20	2137	10/0	0	0	0	10/0	0	0	0
pmed5		33	818	10/0	0	0	0	10/0	0	0	0
pmed6	200	5	6064	2/0	0,247	0,198	0	6/0	0,247	0,099	0
pmed7		10	4206	10/0	0	0	0	10/0	0	0	0
pmed8		20	3182	10/0	0	0	0	10/0	0	0	0
pmed9		40	1807	10/0	0	0	0	10/0	0	0	0
pmed10		67	818	10/0	0	0	0	9/0	0,367	0,037	0

kontynuacja na następnej stronie

Problem				inicjalizacja zachłanna				inicjalizacja losowa			
				# opt./ popr.	odstęp [%]			# opt./ popr.	odstęp [%]		
nazwa	m	n	NZR		max.	śr.	min.		max.	śr.	min.
pmed11	300	5	5979	10/0	0	0	0	8/0	0,017	0,003	0
pmed12		10	5021	10/0	0	0	0	10/0	0	0	0
pmed13		30	3133	10/0	0	0	0	9/0	0,351	0,035	0
pmed14		60	1949	1/9	0	-0,139	-0,154	1/8	0,103	-0,113	-0,154
pmed15		100	1133	1/9	0	-0,247	-0,353	0/10	-0,088	-0,309	-0,353
pmed16	400	5	6341	10/0	0	0	0	9/0	0,394	0,039	0
pmed17		10	5381	3/0	0,595	0,416	0	9/0	0,595	0,059	0
pmed18		40	3437	10/0	0	0	0	10/0	0	0	0
pmed19		80	1922	2/8	0	-0,047	-0,104	1/6	0,260	0,016	-0,104
pmed20		133	1146	6/0	0,524	0,140	0	4/0	0,698	0,253	0
pmed21	500	5	7245	10/0	0	0	0	10/0	0	0	0
pmed22		10	6685	2/0	0,957	0,471	0	6/0	0,823	0,197	0
pmed23		50	3306	4/0	0,181	0,091	0	5/0	0,181	0,036	0
pmed24		100	2004	0/10	-0,100	-0,160	-0,200	0/9	0,050	-0,115	-0,200
pmed25		167	1148	2/0	0,174	0,096	0	3/0	0,261	0,105	0
pmed26	600	5	7787	10/0	0	0	0	4/0	0,848	0,258	0
pmed27		10	6444	10/0	0	0	0	10/0	0	0	0
pmed28		60	3202	0/1	0,344	0,116	-0,031	2/1	0,312	0,078	-0,062
pmed29		120	2005	3/7	0	-0,140	-0,299	0/10	-0,050	-0,229	-0,349
pmed30		200	1295	1/4	0,154	0,015	-0,077	0/4	0,386	0,069	-0,077
pmed31	700	5	8046	10/0	0	0	0	8/0	0,124	0,025	0
pmed32		10	7278	0/0	0,027	0,027	0,027	1/0	0,027	0,025	0
pmed33		70	3403	7/0	0,059	0,015	0	7/0	0,235	0,035	0
pmed34		140	2023	1/6	0,099	-0,069	-0,198	1/7	0,049	-0,138	-0,247
pmed35	800	5	8191	10/0	0	0	0	9/0	1,013	0,101	0
pmed36		10	7796	4/0	0,308	0,185	0	4/0	0,770	0,277	0
pmed37		80	3598	0/0	0,139	0,064	0,028	2/0	0,195	0,078	0
pmed38	900	5	8720	10/0	0	0	0	9/0	0,780	0,078	0
pmed39		10	7360	10/0	0	0	0	10/0	0	0	0
pmed40		90	3710	3/7	0	-0,043	-0,081	0/5	0,135	-0,005	-0,135
Średnia					0,099	0,025	-0,036		0,233	0,026	-0,042

Porównując bezpośrednio ze sobą wersje bez i z regularyzacją, można zaobserwować niewielkie różnice. Przy inicjalizacji zachłannej wersja z regularyzacją nie znajduje 2 najlepszych rozwiązań, które osiąga wersja bez regularyzacji, ale za to poprawia 1 najlepsze rozwiązanie, którego nie znajduje wersja bez regularyzacji i polepsza 2 rozwiązania w minimalnie lepszym stopniu. Przy inicjalizacji losowej jedyna różnica to minimalnie lepsza poprawa 1 rozwiązania przez metodę z regularyzacją. W ogólności metoda z regularyzacją przy inicjalizacji losowej osiągnęła najlepsze rozwiązania problemów z kryterium zawężonej średniej ze wszystkich badanych wersji. Różnice wprowadzone przez regularyzację dla tego kryterium są jednak na tyle małe, że obie wersje można uznać za jednakowo dobre.

Podsumowując, w przypadku problemów z kryterium zawężonej średniej, podobnie jak dla kryterium średniej, rozwiązania uzyskiwane za pomocą metody VNS charakteryzują się bardzo dobrą jakością. Dotyczy to zarówno metody bez, jak i z regularyzacją oraz obu sposobów inicjalizacji, przy czym minimalnie lepsze okazały się wyniki metody z inicjalizacją losową.

## 8.5. Wnioski

W rozdziale zaprezentowano zmodyfikowaną metodę przeszukiwania zmiennego sąsiedztwa dla dyskretnych problemów lokalizacyjnych z kryterium uporządkowanej średniej ważonej. Wprowadzone zmiany dotyczą zarówno aspektów wydajnościowych, jak i jakościowych.

W ramach modyfikacji wydajnościowych zmieniony został algorytm wymiany, który podzielono na dwa etapy. Dzięki temu po części relaksacyjnej dodano warunek, który pozwolił na ograniczenie przeszukiwania przestrzeni rozwiązań w nieobiecujących kierunkach. Zwykle sortowanie całego wektora kosztów przy wyznaczaniu nowych rozwiązań zastąpiono mechanizmem uaktualniania uporządkowanego wektora kosztów. Zaproponowany mechanizm znacząco zmniejszył koszt obliczeniowy ewaluacji jakości rozwiązań uzyskiwanych w kolejnych iteracjach metody. Opracowano również dodatkowy warunek, bazujący na dominacji posortowanych wektorów, umożliwiającą odrzucenie części gorszych rozwiązań bez dokładnego wyznaczenia wartości funkcji celu.

W celu poprawy jakości uzyskiwanych rozwiązań dodano warunek regularyzacyjny. Jest on aktywny dla pewnych typów problemów i zwiększa szansę na wybór bardziej obiecującego rozwiązania przy porównywaniu rozwiązań o równych wartościach funkcji celu.

W kontekście uzyskanych wyników można powiedzieć, że wszystkie zaproponowane modyfikacje są użyteczne i spełniają swoje zadania. W związku z tym metoda z regularyzacją i wszystkimi trzema modyfikacjami jakościowymi wydaje się być wersją uniwersalną z najlepszym stosunkiem wydajności do jakości otrzymywanych rozwiązań. Wątpliwy dla małych problemów warunek dominacji znacząco zmniejsza czas dla dużych problemów z większą liczbą punktów obsługi. Oczywiście wiedząc a priori, że problemy do rozwiązania są niewielkie lub mają małą liczbę punktów obsługi, można zastosować wersję metody bez warunku dominacji. Zwłaszcza jeśli nawet niewielkie bezwzględne różnice czasu są istotne. Regularyzacja, będąc aktywna, wydłuża czas rozwiązania, jednak zazwyczaj w niewielkim stopniu. Tam gdzie zmiany są znaczące (kryterium centrum), zdecydowana poprawa jakości uzasadnia tę modyfikację.

Przeprowadzone testy wskazują, że zmodyfikowana metoda pozwala na kilkukrotne skrócenie czasu obliczeń. Dotyczy to zwłaszcza problemów o większych wymiarach, a w szczególności instancji z większą liczbą punktów obsługi. Poprawa w ich przypadku często przekracza jeden rząd wielkości w stosunku do metody oryginalnej. Mniejsza poprawa (ok. 3–4 razy) dotyczy kryterium centrum i wynika z kosztów regularyzacji. W przypadku wersji bez regularyzacji poprawa wydajności przekracza nawet dwa rzędy wielkości, jednak otrzymywane

rozwiązania charakteryzują się niską jakością. Dyskwalifikuje to wersję bez regularyzacji do rozwiązywania problemów z kryterium centrum. To samo dotyczy także pierwotnej wersji VNS, jak również innych ogólnych przybliżonych metod rozwiązywania problemów z kryterium porządkowym z literatury (np. HGA1, HGA2). W tym kontekście kilkukrotne skrócenie czasu przy zdecydowanej poprawie jakości należy uznać za bardzo korzystne. Tym bardziej, że czasy dla kryterium centrum, pomimo wzrostu, pozostały krótsze niż np. dla kryterium średniej czy zawężonej średniej. Znacznie krótsze czasy rozwiązania i gorsza jakość rozwiązań w wersji bez regularyzacji potwierdzają również fakt, że przeszukiwanie w wersji oryginalnej było zbyt słabo zintensyfikowane. Regularyzacja pozwoliła na większą intensyfikację w poszczególnych iteracjach metody.

Dzięki zmianom wydajnościowym możliwe jest wykonanie w tym samym czasie większej liczby iteracji metody przybliżonej lub kilkukrotne jej uruchomienie, co pozwala na uzyskanie lepszego rozwiązania. Bezpośrednio na jakość wpływa warunek regularyzacji. Porównanie z innymi metodami z literatury pokazuje, że otrzymane rozwiązania charakteryzują się bardzo dobrą jakością. W przypadku problemów z kryterium średniej i zawężonej średniej uzyskane rozwiązania są optymalne lub tylko minimalnie gorsze. W największym stopniu poprawione zostały rozwiązania problemów z kryterium centrum. Co prawda, ich odchylenia względne są większe niż w problemach pozostałych typów, jednak wynika to w dużym stopniu z innej specyfiki problemu. Tym samym zaproponowana metoda VNS z regularyzacją jest faktycznie uniwersalną metodą rozwiązywania problemów z agregacją uporządkowanej średniej ważonej. Pozwala uzyskiwać dobre jakościowo wyniki w akceptowalnym czasie dla dowolnie zdefiniowanego kryterium.

Warto również pamiętać, że uzyskane lepsze rozwiązanie metody przybliżonej stanowi również lepszą granicę (górną), która powinna pozwolić na ściślejsze ograniczenie przeszukiwanego zbioru w metodach dokładnych.

## 9. Metoda VNS dla zróżnicowanych wag zapotrzebowania

Zaprezentowana w poprzednim rozdziale zmodyfikowana metoda VNS dla dyskretnych problemów lokalizacyjnych z agregacją OWA udowodniła swoją użyteczność, pozwalając na uzyskiwanie dobrych jakościowo rozwiązań w rozsądnym czasie dla dowolnie zdefiniowanego kryterium. Metoda ta była jednak ograniczona do problemów z równymi wagami zapotrzebowania. Gdy poszczególne lokalizacje charakteryzują się różnymi zapotrzebowaniami rozumianymi np. jako liczba odbiorców w danej lokalizacji, metoda ta nie może być zastosowana w sposób bezpośredni. Aby możliwe było jej wykorzystanie, podobnie jak dla dokładnych modeli optymalizacyjnych, konieczna byłaby dezagregacja problemu do lokalizacji z równymi porcjami zapotrzebowania. Jak omawiano w rozdziale 6, taka operacja prowadzi zazwyczaj do znacznego wzrostu rozmiaru problemu, a w konsekwencji do znacznego utrudnienia lub wręcz uniemożliwienia rozwiązania problemu. Aby tego uniknąć, w tym rozdziale zaproponowano metodę VNS dla problemów z agregacją WOVA, która pozwala uwzględniać zróżnicowane zapotrzebowania poszczególnych klientów w sposób bezpośredni.

### 9.1. Metoda VNS z agregacją WOVA

Ogólna idea metody VNS z agregacją WOVA jest taka sama jak metody z agregacją OWA. Podstawowa różnica dotyczy sposobu wyznaczania wartości funkcji celu. Do wyznaczenia wartości operatora WOVA wykorzystywany jest wektor wag wartościujących, który wyraża wielkości zapotrzebowania poszczególnych lokalizacji i stanowi dodatkowy parametr wejściowy. Procedura obliczania agregacji WOVA jest realizowana zgodnie z definicją 6.1, gdzie oba wektory wag (preferencji i zapotrzebowania) podlegają normalizacji (takiemu przeskalowaniu aby suma składowych w ramach każdego z wektorów była równa 1).

Jednak nie wszystkie modyfikacje metody VNS z agregacją OWA można zastosować w przypadku agregacji WOVA. Gdy poszczególne lokalizacje mają różne wielkości zapotrzebowania, warunek dominacji posortowanego wektora kosztów (patrz podrozdział 8.2.3) nie jest wystarczający do odrzucenia badanego rozwiązania. Wynika to z faktu, że zwykła dominacja po współrzędnych posortowanego wektora kosztów nie gwarantuje lepszej wartości operatora WOVA. Pokazuje to następujący przykład.

**Przykład 9.1.** Rozpatrzmy problem z 5 lokalizacjami, którym odpowiada następujący wektor wag zapotrzebowania  $\mathbf{p} = (0,1; 0,2; 0,2; 0,4; 0,1)$ . Rozważmy dalej dwa wektory kosztów

$\mathbf{y} = (1; 3; 2; 4; 5)$  oraz  $\mathbf{y}' = (1; 3; 2; 5; 3)$ . Dla uproszczenia założymy równe wagi preferencji  $\mathbf{w} = (0,2; 0,2; 0,2; 0,2; 0,2)$ .

Widać, że wektor  $\mathbf{y}'$  dominuje symetrycznie wektor  $\mathbf{y}$ , gdyż  $\Theta(\mathbf{y}) = (5; 4; 3; 2; 1) \geq (5; 3; 3; 2; 1) = \Theta'(\mathbf{y}')$ . Aby obliczyć wartości agregacji WOWA dla obu wektorów kosztów, wykorzystamy fakt równych wag preferencji, co sprowadza agregację WOWA do zwykłej średniej ważonej. W związku z tym

$$A_{w,p}(\mathbf{y}) = \sum_{i=1}^5 p_i y_i = 0,1 \cdot 1 + 0,2 \cdot 3 + 0,2 \cdot 2 + 0,4 \cdot 4 + 0,1 \cdot 5 = 3,2,$$

natomiast

$$A_{w,p}(\mathbf{y}') = \sum_{i=1}^5 p_i y_i = 0,1 \cdot 1 + 0,2 \cdot 3 + 0,2 \cdot 2 + 0,4 \cdot 5 + 0,1 \cdot 3 = 3,4,$$

a więc  $A_{w,p}(\mathbf{y}) < A_{w,p}(\mathbf{y}')$ . Stąd wynika, że wartość agregacji WOWA jest gorsza dla wektora  $\mathbf{y}'$ , pomimo że dominuje on symetrycznie wektor  $\mathbf{y}$ .

Pozostałe modyfikacje szybkościowe zaproponowane w podrozdziale 8.2 dla metody VNS z agregacją OWA są adekwatne również dla metody VNS dla problemów ze zróżnicowanymi wagami zapotrzebowania. Dodanie jednego nadmiarowego punktu obsługi nadal stanowi lokalną relaksację problemu. Wartość funkcji celu tej relaksacji stanowi ograniczenie dolne dla problemów, które można otrzymać poprzez usunięcie jednego punktu obsługi. Na tej podstawie można więc wyeliminować z dokładnego wyznaczania część rozwiązań, których wartość funkcji celu nie będzie lepsza od aktualnego rozwiązania. Wagi zapotrzebowania nie mają także wpływu na uporządkowanie wektora kosztów. Pozwala to wykorzystać modyfikację sortowania do ograniczenia nakładu obliczeń przy wyznaczaniu nierosnącego wektora kosztów.

Modyfikacji wymaga warunek regularyzacji. W przypadku zróżnicowanych wag zapotrzebowania poszczególne kwantyle rzędu  $1/\tilde{m}, 2/\tilde{m}, \dots, 1$  nie odpowiadają jednoznacznie kolejnym największym kosztom. Z tego względu proste leksykograficzne porównanie wektorów kosztów od zadanej pozycji, które było wykonywane w metodzie VNS z agregacją OWA, nie jest w tej sytuacji odpowiednie. W metodzie z agregacją WOWA należy porównać funkcje kwantylowe  $F_{\mathbf{y}}^{(-1)}$  badanych wektorów kosztów  $\mathbf{y}$ . Sposób ich porównania jest zbliżony do tego, który był stosowany w metodzie VNS z agregacją OWA. Mianowicie porównanie rozpoczyna się od kwantyli ustalonego rzędu, który odpowiada wyznaczonej wcześniej pozycji w wektorze wag preferencji  $\mathbf{w}$ . Podobnie jak przy agregacji OWA pozycja ta jest określona jako współrzędna pierwszej wagi zerowej po wagach niezerowych w wektorze  $\mathbf{w}$ . Od wyznaczonego w ten sposób argumentu porównuje się wartości funkcji kwantylowych. Porównywane są one w sposób leksykograficzny i jako lepsza wybierana jest leksykograficznie mniejsza (od ustalonego argumentu). Ponieważ są to funkcje schodkowe, to ich porównanie wystarczy przeprowadzić w punktach granicznych, które są określone przez sumaryczne wartości wag

zapotrzebowania odbiorców z  $k$  największymi kosztami dla kolejnych  $k$ . Oczywiście podobnie jak dla metody VNS z agregacją OWA warunek regularyzacji nie gwarantuje uzyskania lepszego rozwiązania. Wydaje się również, że zróżnicowane wagi zapotrzebowania spowodują większe zróżnicowanie wartości funkcji celu, co może sprawić, że regularyzacja będzie rzadziej wykonywana, gdyż warunek regularyzacji jest wykorzystywany tylko wtedy, gdy wartości funkcji celu są równe.

## 9.2. Implementacja

Zaproponowane wcześniej modyfikacje metody VNS dla OWA pozwoliły na znaczne ograniczenie skali zmian implementacyjnych przy dostosowaniu metody do agregacji WOVA. W szczególności wynika to z dodania wyznaczania permutacji odwzorowującej uporządkowany wektor kosztów na oryginalny wektor kosztów. Permutacja ta pozwala w prosty sposób odwoływać się do wag zapotrzebowania, które stanowią dodatkowy parametr wejściowy w rozwiązywanych problemach lokalizacyjnych. W ramach implementacji należało uwzględnić wektor wag zapotrzebowania  $p$  przy wyznaczaniu wartości funkcji celu, czyli wartości agregacji WOVA. Z punktu logicznego to podstawowa różnica w stosunku do metody z poprzedniego rozdziału. Modyfikacji wymagał także warunek regularyzacji, w którym ze względu na zróżnicowane wagi zapotrzebowania porównaniu należy poddać funkcje kwantylowe.

Algorytm 8 przedstawia procedurę wyznaczania wartości operatora WOVA. Oznaczenia są analogiczne jak w metodzie VNS dla OWA, przy czym pojawia się także wspomniany wcześniej wektor wag zapotrzebowania  $p$  oraz wymiar wektora wag preferencji  $\tilde{m}$ , który w ogólności może być różny od  $m$ . Używane są również zmienne pomocnicze: sumaryczna wartość zapotrzebowania  $\bar{p}$  pewnej liczby największych ocen, sumaryczne zapotrzebowanie  $\bar{q}$  pewnej liczby równych porcji zapotrzebowania odpowiadających pojedynczej wadze preferencji oraz waga  $\omega$  agregacji WOVA. Procedura w kolejnych iteracjach głównej pętli wyznacza kolejne wagi  $\omega$  agregacji WOVA, które następnie wymnaża przez uporządkowany wektor kosztów  $scost$ . Wagi agregacji są wyznaczone zgodnie ze wzorem (6.2). Złożoność obliczeniowa procedury OBLICZWOWA wynosi  $O(m + \tilde{m})$ , gdyż taka jest sumaryczna liczba iteracji wewnętrznej pętli (parametrem procedury jest posortowany wcześniej wektor kosztów). Zatem podobnie jak w przypadku procedury OBLICZOWA jest to złożoność liniowa, przy czym większy jest współczynnik proporcjonalności. Faktyczna implementacja wyznaczania wartości operatora WOVA odbiega nieznacznie od tej prezentowanej w algorytmie 8. Różnice dotyczą kwestii normalizacji i mają na celu zachowanie maksymalnej dokładności obliczeń. Nie ma to jednak istotnego znaczenia dla prezentowanej analizy.

W ramach implementacji przedstawiona jest także procedura porównania rozwiązań w metodzie VNS z agregacją WOVA z regularyzacją. Procedura ta jest przedstawiona w algorytmie 9 i ma przede wszystkim na celu zaprezentowanie warunku regularyzacji dla problemów ze zróżnicowanymi wagami zapotrzebowania. Zadaniem funkcji CZYLEPSZE jest odpowiedź na

---

**Algorytm 8.** Procedura wyznaczania wartości operatora WOWA

---

```
1: function OBLICZWOWA( $m, \tilde{m}, w, p, scost, \tau, f^*$ )
2:    $i \leftarrow 1$ 
3:    $(f^*, \bar{p}, \bar{p}_{prev}, \bar{q}) \leftarrow (0, 0, 0, i/\tilde{m})$ 
4:   for  $k = 1, \dots, m$  do
5:      $\omega \leftarrow 0$ 
6:      $\bar{p} \leftarrow \bar{p} + p(\tau(k))$ 
7:     while  $(\bar{q} < \bar{p})$  do
8:        $\omega \leftarrow \omega + \tilde{m} \cdot w(i) \cdot (\bar{q} - \bar{p}_{prev})$ 
9:        $\bar{p}_{prev} \leftarrow \bar{q}$ 
10:       $i \leftarrow i + 1$ 
11:       $\bar{q} \leftarrow i/\tilde{m}$ 
12:     end while
13:      $\omega \leftarrow \omega + \tilde{m} \cdot w(i) \cdot (\bar{p} - \bar{p}_{prev})$ 
14:      $\bar{p}_{prev} \leftarrow \bar{p}$ 
15:      $f^* \leftarrow f^* + \omega \cdot scost(k)$ 
16:   end for
17: end function
```

---

pytanie, czy nowe rozwiązanie (oznaczane indeksem dolnym *new*) jest lepsze od rozwiązania aktualnego. Gdy wartości funkcji celu obu rozwiązań są równe, wykonywany jest warunek regularyzacji. Staje się on aktywny, gdy parametr  $\bar{q}$  stanowiący argument funkcji kwantylowych, jest mniejszy od 1 (linia 7). Oznacza to, że w wektorze wag preferencji w istnieje waga zerowa po wagach niezerowych. Wówczas rozpoczyna się procedura porównywania wartości funkcji kwantylowych. W kolejnych iteracjach porównywane są koszty odpowiadające kwantylom w punktach granicznych obu rozwiązań, zaczynając właśnie od kwantyli rzędu  $\bar{q}$ . Pętle wyznaczające kwantyle rzędu  $\bar{q}$  mogą mieć co najwyżej  $m$  iteracji, a główna pętla co najwyżej  $2m$  iteracji. Tym samym asymptotyczna złożoność warunku regularyzacji jest liniowa  $O(m)$ . Warto również pamiętać, że warunek regularyzacji jest wykonywany tylko wtedy, gdy wartości funkcji celu są równe. W pozostałych przypadkach funkcja CZYLEPSZE ma złożoność stałą  $O(1)$ .

### 9.3. Testy obliczeniowe

W ramach testów zbadano dwie wersje metody VNS z agregacją WOWA: bez i z regularyzacją, oznaczone odpowiednio jako  ${}^w\text{VNS}$  i  ${}^w\text{VNS}_r$ . Procedura testowa była zbliżona do tej zastosowanej przy metodzie VNS z agregacją OWA w podrozdziale 8.4. Metoda przybliżona została przetestowana na małych i dużych problemach — tych samych, które były wykorzystywane w poprzednich rozdziałach. Wagi zapotrzebowania wygenerowano zgodnie z rozkładem Zipfa, tak jak przy dokładnych modelach optymalizacyjnych w rozdziale 7.



---

**Algorytm 9.** Procedura porównania rozwiązań z warunkiem regularyzacyjnym dla WOWA

---

```
1: function CZYLEPSZE( $m, \bar{q}, f, f_{new}, scost, scost_{new}, p, p_{new}, \tau, \tau_{new}$ )
2:   if  $f_{new} < f$  then
3:     return true
4:   else if  $f_{new} > f$  then
5:     return false
6:   else
7:     if  $\bar{q} = 1$  then
8:       return false
9:     end if
10:     $(\bar{p}, \bar{p}_{new}, k, k_{new}) \leftarrow (0, 0, 0, 0)$ 
11:    while  $\bar{p} \leq \bar{q}$  do
12:       $k \leftarrow k + 1$ 
13:       $\bar{p} \leftarrow \bar{p} + p(\tau(k))$ 
14:    end while
15:    while  $\bar{p}_{new} \leq \bar{q}$  do
16:       $k_{new} \leftarrow k_{new} + 1$ 
17:       $\bar{p}_{new} \leftarrow \bar{p}_{new} + p_{new}(\tau_{new}(k_{new}))$ 
18:    end while
19:    loop
20:      if  $scost_{new}(k_{new}) < scost(k)$  then
21:        return true
22:      else if  $scost_{new}(k_{new}) > scost(k)$  then
23:        return false
24:      else
25:        if  $k = m$  and  $k_{new} = m$  then
26:          return false
27:        end if
28:        if  $\bar{p} < \bar{p}_{new}$  then
29:           $k \leftarrow k + 1$ 
30:           $\bar{p} \leftarrow \bar{p} + p(\tau(k))$ 
31:        else if  $\bar{p} > \bar{p}_{new}$  then
32:           $k_{new} \leftarrow k_{new} + 1$ 
33:           $\bar{p}_{new} \leftarrow \bar{p}_{new} + p_{new}(\tau_{new}(k_{new}))$ 
34:        else
35:           $k \leftarrow k + 1$ 
36:           $\bar{p} \leftarrow \bar{p} + p(\tau(k))$ 
37:           $k_{new} \leftarrow k_{new} + 1$ 
38:           $\bar{p}_{new} \leftarrow \bar{p}_{new} + p_{new}(\tau_{new}(k_{new}))$ 
39:        end if
40:      end if
41:    end loop
42:  end if
43: end function
```

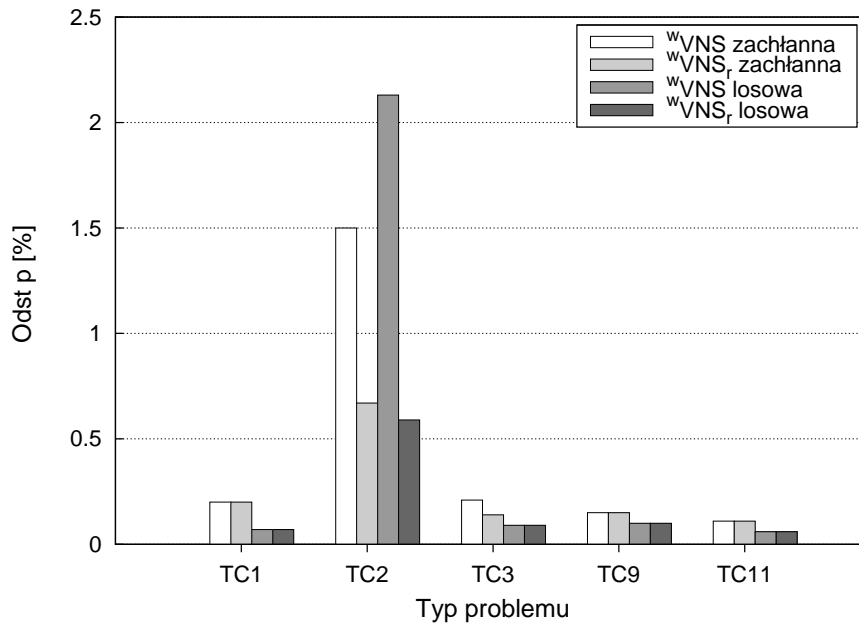
---

### 9.3.1. Problemy o niewielkim rozmiarze

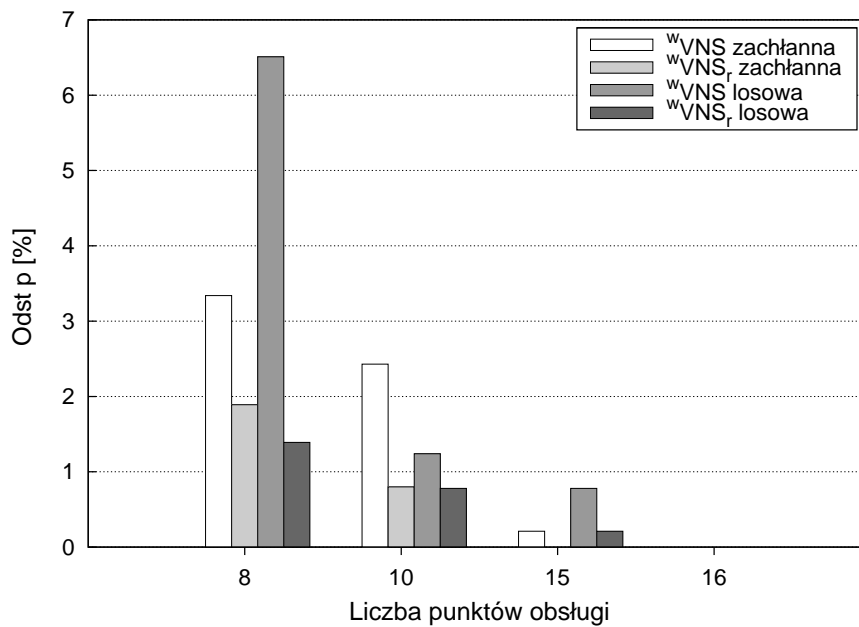
Na małych problemach sprawdzono dokładność metody VNS z agregacją WOWA. Wykorzystano do tego problemy z nierosnącymi wagami preferencji (TC1–TC3, TC9, TC11), dla których za pomocą modeli programowania matematycznego udało się uzyskać rozwiązania optymalne dla rozmiarów do 30 lokalizacji. Szczegółowe wyniki znajdują się w dodatku D w tabeli D.1. Poszczególne wiersze zawierają maksymalne, średnie i minimalne odstępów od rozwiązań optymalnych dla 15 instancji różnych macierzy kosztów w ramach poszczególnych typów, rozmiarów oraz wariantów liczby punktów obsługi. Podobnie jak przy metodzie VNS z agregacją OWA, także tutaj widać, że jakościowo najgorsze rozwiązania są uzyskiwane dla kryterium centrum. Średnie odstępów dla problemów typu TC2 są kilkukrotnie gorsze niż dla pozostałych typów z monotonicznymi wagami preferencji. Warto jednak zauważyć, że wyniki metody VNS z agregacją WOWA są lepsze — znalezione rozwiązania w mniejszym stopniu odbiegają od rozwiązań optymalnych — niż wyniki metody VNS z agregacją OWA. Znaczna część problemów została rozwiązana optymalnie. Poza kryterium centrum zdecydowana większość badanych problemów uzyskała średni odstęp poniżej 1% dla 15 instancji macierzy kosztów w ramach poszczególnych kombinacji typów, rozmiarów i liczby punktów obsługi. Nieco większe odstępów (do kilku procent) dotyczą problemów z kryterium centrum. Te różnice obrazuje również rysunek 9.1, który przedstawia wyniki dla problemów z 30 lokalizacjami uśrednione dodatkowo po wariantach liczby punktów obsługi. Analizując dalej wpływ regularyzacji, widać, że dla badanych typów problemów jest ona aktywna tylko dla TC2 i TC3, a istotny wpływ ma jedynie na kryterium centrum (TC2). Na rysunku 9.2 przedstawione są wyniki dla problemów z kryterium centrum z 30 lokalizacjami. W tym przypadku metoda z regularyzacją osiąga nie gorsze wyniki, a czasami kilkukrotnie lepsze, niż metoda bez regularyzacji. Należy jednak zaznaczyć, że warunek regularyzacji nie gwarantuje lepszego rozwiązania i dla części problemów o mniejszym rozmiarze wersja z regularyzacją osiąga gorsze wyniki. Prawdopodobnie wynika to z faktu, że dla mniejszych problemów zbyt szybko aktywuje się warunek stopu związany z maksymalnym sąsiedztwem. Być może dla mniejszych problemów korzystne byłoby określenie minimalnej liczby iteracji metody VNS. Poza tym tak małe problemy można rozwiązać za pomocą metod dokładnych lub ewentualnie rozwiązać kilkukrotnie dany problem za pomocą metody przybliżonej. Wydaje się, że regularyzacja pozwala osiągać lepsze rozwiązania dla większych problemów, chociaż jej wpływ jest mniejszy niż w przypadku metody VNS z agregacją OWA. Po części wynika to z ogólnie lepszych rezultatów metody VNS z agregacją WOWA w stosunku do metody VNS z agregacją OWA.

### 9.3.2. Problemy o dużym rozmiarze

Druga część testów została przeprowadzona na problemach o dużych rozmiarach pochodzących z biblioteki OR. Podobnie jak dla metody z agregacją OWA rozważono 3 typy kryteriów: średniej, centrum oraz zawężonej średniej. Ze względu na brak optymalnych rozwiązań dużych



Rysunek 9.1. Średni odstęp [%] metody VNS z agregacją WOWA dla problemów z 30 lokalizacjami



Rysunek 9.2. Średni odstęp [%] metody VNS z agregacją WOWA dla problemów kryterium centrum z 30 lokalizacjami

problemów ze zróżnicowanymi wagami zapotrzebowania nie przeprowadzono pełnej analizy jakościowej. Z rozważanych tutaj 3 kryteriów, za pomocą modelu PL agregacji WOWA (podrozdział 7.5.1) udało się jednak uzyskać optymalne rozwiązania problemów z kryterium średniej o rozmiarach 100 i 200 lokalizacji. Spośród tych 10 problemów (*pmed1–pmed10*) metoda przybliżona przy inicjalizacji zachłannej znalazła optymalne rozwiązania dla 9 spośród nich, a w przypadku jednego problemu uzyskany odstęp był mniejszy od 0,01%. Przy inicjalizacji losowej optymalne rozwiązania zostały uzyskane dla wszystkich 10 problemów.

W ramach analizy wydajnościowej oprócz problemów ze zróżnicowanymi wagami zapotrzebowania rozwiązano również problemy z równymi wagami zapotrzebowania. Celem tego była ocena, w jakim stopniu dopasowanie do agregacji WOWA wpłynęło na efektywność badanej metody. Dwie wersje metody VNS z agregacją WOWA, czyli  ${}^w\text{VNS}$  i  ${}^w\text{VNS}_r$ , porównano z odpowiadającymi im wersjami metody VNS z agregacją OWA bez warunku dominacji, a mianowicie z  $\text{VNS}'$  oraz  $\text{VNS}'_r$ . Zastosowano te same początkowe ziarna funkcji losujących, co zapewniło tę samą drogę przeszukiwania przestrzeni rozwiązań oraz uzyskanie tego samego rozwiązania. Potwierdzają to również wartości funkcji celu, które są równe z dokładnością do normalizacji wag (preferencji i zapotrzebowania). W drugiej kolejności porównano czasy rozwiązania problemów z równymi oraz ze zróżnicowanymi wagami zapotrzebowania. To porównanie ma jednak bardziej charakter poglądowy, gdyż różne wagi zapotrzebowania zmieniają diametralnie proces przeszukiwania przestrzeni rozwiązań. Dlatego w tym przypadku chodzi raczej o porównanie zbiorcze, które ma pokazać, jaka jest ogólna tendencja i skala zmian czasów rozwiązania przy uwzględnieniu zróżnicowanych zapotrzebowań.

Czasy osiągane przez metodę VNS z agregacją WOWA przedstawiono w tabelach 9.1–9.3.

W tabeli 9.1 znajdują się czasy rozwiązania problemów z kryterium średniej. Ponieważ regularyzacja nie jest aktywna dla kryterium średniej, to czasy wersji bez i z warunkiem regularyzacji są praktycznie identyczne. Jak widać ogólna charakterystyka czasowa, zarówno dla równych, jak i zróżnicowanych wag zapotrzebowania, jest podobna do charakterystyki metody z agregacją OWA — czasy rosną wraz ze wzrostem rozmiaru oraz jeszcze szybciej ze wzrostem liczby punktów obsługi. Dla małej liczby punktów obsługi (5, 10) zdecydowana większość problemów jest rozwiązywana w czasie kilku sekund. W ramach wszystkich 40 problemów najdłuższe czasy są poniżej 450 s.

Aby sprawdzić, w jakim stopniu uwzględnienie wartości wag zapotrzebowania wpłynęło na szybkość metody VNS, porównano czasy tej nowej metody dla problemów z równymi wagami zapotrzebowania z czasami metody VNS z agregacją OWA (patrz tabela 8.4). Porównanie czasów pokazuje, że metoda VNS z agregacją WOWA osiąga czasy gorsze o ok. 30%–75%, przy czym generalnie różnice rosną wraz ze wzrostem liczby punktów obsługi. W ramach 40 problemów czas rozwiązania wzrósł średnio o ok. 50%. Różnice dotyczą konsekwentnie wersji bez jak i z regularyzacją dla obu typów inicjalizacji.

Porównując czasy rozwiązań problemów ze zróżnicowanymi wagami zapotrzebowania z czasami rozwiązań problemów z równymi wagami zapotrzebowania, nie widać jednoznacznej

zależności. Zmiana rozkładu wartości wag zapotrzebowania w części problemów powoduje skrócenie, a w innych wydłużenie czasu rozwiązania. W zdecydowanej większości problemów różnice mieszczą się zakresie od 50% skrócenia do jednokrotnego wydłużenia czasu rozwiązania. Ogólnie można powiedzieć, że czasy są zbliżone z lekką tendencją wzrostową dla problemów ze zróżnicowanymi wagami zapotrzebowania.

Tabela 9.1: Statystyki czasowe metody VNS z agregacją WOVA dla problemów z biblioteki OR z kryterium średniej (TC1)

Problem			równe wagi zapotrzebowania				zróżnicowane wagi zapotrzebowania			
			inicjalizacja zachłanna		inicjalizacja losowa		inicjalizacja zachłanna		inicjalizacja losowa	
nazwa	m	n	CPU[s]				CPU[s]			
			<sup>w</sup> VNS	<sup>w</sup> VNS <sub>r</sub>	<sup>w</sup> VNS	<sup>w</sup> VNS <sub>r</sub>	<sup>w</sup> VNS	<sup>w</sup> VNS <sub>r</sub>	<sup>w</sup> VNS	<sup>w</sup> VNS <sub>r</sub>
pmed1	100	5	0,06	0,06	0,05	0,06	0,04	0,05	0,05	0,05
pmed2		10	0,16	0,15	0,19	0,25	0,14	0,14	0,15	0,18
pmed3		10	0,12	0,12	0,15	0,20	0,25	0,26	0,14	0,14
pmed4		20	0,71	0,70	0,77	0,76	0,57	0,52	0,63	0,64
pmed5		33	1,60	1,54	2,25	2,21	1,74	1,73	1,79	1,79
pmed6	200	5	0,22	0,22	0,22	0,24	0,18	0,18	0,17	0,18
pmed7		10	0,79	0,79	0,43	0,44	0,49	0,50	0,52	0,53
pmed8		20	3,43	3,44	3,47	3,44	2,17	2,22	2,10	2,13
pmed9		40	10,15	10,17	8,89	8,83	8,81	8,97	9,71	9,86
pmed10		67	11,95	11,98	16,21	16,12	24,39	24,76	24,75	24,97
pmed11	300	5	0,36	0,35	0,43	0,42	0,37	0,37	0,31	0,31
pmed12		10	1,09	1,08	1,30	1,30	1,26	1,28	1,14	1,16
pmed13		30	9,70	9,63	16,07	16,07	8,23	8,33	13,73	13,83
pmed14		60	43,30	42,87	30,05	30,00	40,63	39,09	54,71	55,11
pmed15		100	38,79	38,40	53,14	52,97	49,26	49,68	78,30	78,34
pmed16	400	5	0,77	0,77	0,58	0,58	0,57	0,58	0,72	0,72
pmed17		10	1,41	1,40	1,57	1,57	1,77	1,79	2,10	2,14
pmed18		40	36,05	35,93	43,69	43,54	32,78	33,15	42,01	42,42
pmed19		80	62,07	61,80	77,17	76,66	89,06	89,42	98,35	99,33
pmed20		133	112,32	111,81	115,34	114,51	122,87	123,33	183,85	184,82
pmed21	500	5	0,68	0,69	1,03	1,03	0,95	0,97	1,09	1,11
pmed22		10	2,55	2,54	3,06	3,06	3,84	3,89	3,40	3,45
pmed23		50	100,16	99,66	75,01	74,77	62,48	63,03	58,77	59,36
pmed24		100	175,70	174,82	179,64	179,04	146,09	148,47	138,28	141,13
pmed25		167	175,22	174,18	197,97	196,68	249,67	250,10	395,99	396,15
pmed26	600	5	1,36	1,36	1,21	1,21	1,43	1,45	1,42	1,43
pmed27		10	5,10	5,11	6,09	6,08	4,08	4,15	4,25	4,35
pmed28		60	93,72	93,64	186,73	186,44	107,45	108,50	111,72	112,85
pmed29		120	251,12	251,21	311,09	311,05	296,08	296,87	278,79	281,69
pmed30		200	270,28	269,20	292,92	291,82	412,96	417,19	415,95	418,41
pmed31	700	5	1,42	1,42	1,79	1,79	2,13	2,16	2,56	2,61

kontynuacja na następnej stronie

Problem			równe wagi zapotrzebowania				zróżnicowane wagi zapotrzebowania			
			inicjalizacja zachłanna		inicjalizacja losowa		inicjalizacja zachłanna		inicjalizacja losowa	
nazwa	m	n	CPU[s]				CPU[s]			
			<sup>w</sup> VNS	<sup>w</sup> VNS <sub>r</sub>	<sup>w</sup> VNS	<sup>w</sup> VNS <sub>r</sub>	<sup>w</sup> VNS	<sup>w</sup> VNS <sub>r</sub>	<sup>w</sup> VNS	<sup>w</sup> VNS <sub>r</sub>
pmed32		10	5,25	5,25	7,09	7,08	5,70	5,80	6,28	6,39
pmed33		70	204,64	204,32	216,60	215,82	197,93	200,12	221,27	223,97
pmed34		140	322,86	321,92	406,38	404,73	432,43	434,88	378,37	380,67
pmed35	800	5	2,52	2,52	2,75	2,74	3,01	3,05	2,34	2,38
pmed36		10	6,65	6,66	9,02	9,01	7,71	7,84	8,94	9,12
pmed37		80	392,50	392,07	361,33	360,24	259,77	262,87	266,68	269,33
pmed38	900	5	4,30	4,30	3,64	3,63	3,84	3,91	3,62	3,68
pmed39		10	6,77	6,78	8,47	8,45	10,40	10,59	10,41	10,59
pmed40		90	398,10	398,42	440,13	439,15	418,63	423,44	367,37	370,53

W tabeli 9.2 znajdują się czasy rozwiązania problemów z kryterium centrum. Podobnie jak dla kryterium średniej także tutaj ogólna charakterystyka czasowa jest podobna do metody VNS z agregacją OWA — czasy rosną ze wzrostem rozmiaru i liczbą punktów obsługi. Widoczne są również wyraźne różnice związane z warunkiem regularyzacji. Czasy wersji bez regularyzacji sięgają maksymalnie kilkunastu sekund, podczas gdy wersje z regularyzacją dla największych problemów zbliżają się do ok. 300 s.

Na początku porównano metodę VNS z agregacją WOWA do metody z agregacją OWA (tabela 8.5) dla problemów z równymi wagami zapotrzebowania. W przypadku wersji z regularyzacją wyniki są bardzo zbliżone do tych dla kryterium średniej. Czasy rozwiązania dla agregacji WOWA wzrastają o ok. 30%–75%, przy czym większy wzrost dotyczy problemów z większą liczbą punktów obsługi (poza wyjątkami najmniejszych problemów, których czasy rozwiązania są zbliżone do dokładności pomiaru). W ramach 40 problemów czas rozwiązania wzrósł średnio o ok. 50%. W nieco mniejszym stopniu wzrosły czasy dla wersji bez regularyzacji. Poza kilkoma wyjątkami dla najmniejszych problemów wzrost mieści się w granicach 20%–50%, przy czym średnia dla wszystkich problemów wyniosła ok. 40%. Podobnie jak dla kryterium średniej widoczna jest także tendencja większego wzrostu dla większej liczby punktów obsługi, chociaż z większą liczbą nieregularności.

Następnie porównano czasy rozwiązań dla problemów ze zróżnicowanymi wagami zapotrzebowania z czasami dla problemów z równymi wagami zapotrzebowania. W zdecydowanej większości przypadków zróżnicowane wagi zapotrzebowania doprowadziły do wydłużenia czasów rozwiązania, przy czym w większości przypadków wzrosty nie przekraczały jednokrotności. W ramach 40 problemów czas rozwiązania wzrósł średnio o ok. 60% przy inicjalizacji zachłannej oraz o ok. 85% przy inicjalizacji losowej.

Tabela 9.2: Statystyki czasowe metody VNS z agregacją WOVA dla problemów z biblioteki OR z kryterium centrum (TC2)

Problem			równe wagi zapotrzebowania				zróżnicowane wagi zapotrzebowania			
			inicjalizacja zachłanna		inicjalizacja losowa		inicjalizacja zachłanna		inicjalizacja losowa	
nazwa	m	n	CPU[s]				CPU[s]			
			<sup>w</sup> VNS	<sup>w</sup> VNS <sub>r</sub>	<sup>w</sup> VNS	<sup>w</sup> VNS <sub>r</sub>	<sup>w</sup> VNS	<sup>w</sup> VNS <sub>r</sub>	<sup>w</sup> VNS	<sup>w</sup> VNS <sub>r</sub>
pmed1	100	5	0,01	0,06	0,02	0,08	0,03	0,05	0,04	0,10
pmed2		10	0,04	0,19	0,03	0,27	0,04	0,19	0,07	0,16
pmed3		10	0,03	0,25	0,05	0,21	0,07	0,18	0,12	0,23
pmed4		20	0,06	0,59	0,07	0,40	0,18	0,66	0,14	0,70
pmed5		33	0,10	0,80	0,09	0,84	0,25	1,46	0,24	1,48
pmed6	200	5	0,06	0,14	0,11	0,15	0,11	0,37	0,11	0,35
pmed7		10	0,18	1,11	0,07	0,54	0,41	0,83	0,14	2,53
pmed8		20	0,23	2,40	0,21	2,96	0,47	3,52	0,82	4,01
pmed9		40	0,30	4,96	0,29	4,27	0,58	6,43	0,47	5,74
pmed10		67	0,95	6,49	0,74	8,18	1,21	9,63	0,70	8,43
pmed11	300	5	0,18	0,33	0,12	0,27	0,36	0,41	0,25	0,97
pmed12		10	0,22	0,85	0,28	0,78	0,45	1,90	0,41	4,19
pmed13		30	0,50	8,76	0,55	8,25	0,60	11,54	0,92	11,13
pmed14		60	0,91	14,52	0,45	15,82	1,46	18,50	1,02	19,79
pmed15		100	1,50	24,28	1,44	21,06	1,66	34,32	1,91	33,50
pmed16	400	5	0,23	0,74	0,10	0,52	0,43	1,90	0,26	0,81
pmed17		10	0,27	1,54	0,20	1,66	0,68	2,73	0,59	6,68
pmed18		40	1,34	18,94	1,16	20,59	1,62	25,25	1,52	23,81
pmed19		80	1,72	36,37	1,70	37,60	3,54	44,85	1,70	55,03
pmed20		133	4,35	62,43	3,36	59,77	3,43	65,28	4,35	82,60
pmed21	500	5	0,20	1,21	0,38	0,84	0,67	1,17	0,58	1,76
pmed22		10	1,18	2,33	0,29	2,42	1,05	8,95	1,34	6,95
pmed23		50	1,97	40,41	1,85	40,65	3,41	51,10	2,49	43,92
pmed24		100	4,06	70,23	3,63	68,13	4,99	97,28	4,17	93,91
pmed25		167	7,26	104,60	4,38	123,15	8,52	153,43	3,88	166,06
pmed26	600	5	0,50	2,45	0,53	1,65	0,95	4,64	0,57	2,68
pmed27		10	0,53	4,41	0,56	4,82	1,03	10,11	1,76	17,42
pmed28		60	3,50	76,31	3,31	71,23	5,34	67,14	4,31	77,90
pmed29		120	7,36	127,94	5,32	122,98	8,74	141,62	3,70	158,59
pmed30		200	11,06	165,97	6,10	171,65	11,79	279,49	6,06	248,33
pmed31	700	5	0,79	1,63	0,77	1,65	0,78	3,00	1,36	3,37
pmed32		10	0,80	7,62	0,72	6,48	1,69	13,36	1,73	13,23
pmed33		70	5,05	113,63	2,73	114,43	7,62	119,80	6,04	133,69
pmed34		140	10,80	203,34	4,49	203,17	11,04	230,45	5,84	258,87
pmed35	800	5	1,00	2,35	0,88	1,95	1,22	3,47	1,06	2,52
pmed36		10	1,68	9,82	0,80	10,60	1,81	15,46	2,06	21,70
pmed37		80	8,97	144,66	4,51	171,99	10,23	191,54	8,16	201,93
pmed38	900	5	1,61	3,06	0,69	2,93	1,80	9,25	1,36	4,53

kontynuacja na następnej stronie

Problem			równe wagi zapotrzebowania				zróżnicowane wagi zapotrzebowania			
			inicjalizacja zachłanna		inicjalizacja losowa		inicjalizacja zachłanna		inicjalizacja losowa	
			CPU[s]				CPU[s]			
nazwa	m	n	<sup>w</sup> VNS	<sup>w</sup> VNS <sub>r</sub>	<sup>w</sup> VNS	<sup>w</sup> VNS <sub>r</sub>	<sup>w</sup> VNS	<sup>w</sup> VNS <sub>r</sub>	<sup>w</sup> VNS	<sup>w</sup> VNS <sub>r</sub>
pmed39		10	1,98	11,10	0,88	8,82	2,41	36,90	2,53	15,32
pmed40		90	9,34	236,58	7,87	257,76	13,61	259,06	9,93	290,22

W tabeli 9.3 znajdują się czasy rozwiązania problemów z kryterium zawężonej średniej. Ogólnie wyniki są zbliżone do kryterium średniej. Przy małej liczbie punktów obsługi (5, 10) większość czasów mieści się w granicach kilku sekund. Dla większych rozmiarów i większej liczby punktów obsługi maksymalne czasy sięgają ok. 500 s. Można zauważyć, że charakter wyników problemów z równymi wagami zapotrzebowania jest nieco inny niż problemów ze zróżnicowanymi wagami zapotrzebowania. Dla tych pierwszych widoczne są pewne różnice między wersją bez i z regularyzacją, podobnie jak miało to miejsce w metodzie z agregacją OWA. Z kolei dla problemów ze zróżnicowanymi wagami zapotrzebowania, poza pojedynczymi wyjątkami, nie występują praktycznie różnice czasowe pomiędzy wersjami bez i z regularyzacją. Wydaje się, że może to wynikać z faktu zróżnicowania wyników przez wagi zapotrzebowania, co w konsekwencji wyeliminowało (lub zdecydowanie ograniczyło) przypadki równych wartości funkcji celu, a tym samym aktywność warunku regularyzacji.

Porównanie czasów rozwiązań problemów z równymi wagami zapotrzebowania metody VNS z agregacją WOVA z czasami metody VNS z agregacją OWA (tabela 8.6) jest praktycznie identyczne jak dla kryterium średniej. Czasy rozwiązania metody uwzględniającej wartości wag zapotrzebowania są dłuższe o ok. 30%–75%, a w ramach 40 problemów średnio o ok. 50%. Różnice rosną wraz z liczbą punktów obsługi. Dotyczy to zarówno wersji bez, jak i z regularyzacją oraz obu typów inicjalizacji.

Różnice czasów metody VNS z agregacją WOVA między problemami z równymi i ze zróżnicowanymi wagami zapotrzebowania są analogiczne jak dla kryterium średniej. W zależności od instancji problemu część czasów rozwiązania dla zróżnicowanych wag zapotrzebowania wzrosła, a część zmalała, przy czym w większości przypadków różnice mieszczą się w zakresie od 50% skrócenia do jednokrotnego wydłużenia. Generalnie można powiedzieć, że czasy są zbliżone z niewielką tendencją wzrostową dla zróżnicowanych wag zapotrzebowania przy inicjalizacji zachłannej.



Tabela 9.3: Statystyki czasowe metody VNS z agregacją WOVA dla problemów z biblioteki OR z kryterium zawężonej średniej (TC4)

Problem			równe wagi zapotrzebowania				zróżnicowane wagi zapotrzebowania			
			inicjalizacja zachłanna		inicjalizacja losowa		inicjalizacja zachłanna		inicjalizacja losowa	
nazwa	m	n	CPU[s]				CPU[s]			
			<sup>w</sup> VNS	<sup>w</sup> VNS <sub>r</sub>	<sup>w</sup> VNS	<sup>w</sup> VNS <sub>r</sub>	<sup>w</sup> VNS	<sup>w</sup> VNS <sub>r</sub>	<sup>w</sup> VNS	<sup>w</sup> VNS <sub>r</sub>
pmed1	100	5	0,06	0,06	0,04	0,04	0,04	0,04	0,04	0,04
pmed2		10	0,20	0,18	0,21	0,20	0,11	0,12	0,13	0,13
pmed3		10	0,16	0,16	0,17	0,18	0,17	0,17	0,15	0,15
pmed4		20	0,52	0,62	0,59	0,60	0,53	0,52	0,59	0,60
pmed5		33	1,53	1,63	1,32	1,49	1,63	1,62	1,67	1,68
pmed6	200	5	0,22	0,22	0,16	0,18	0,18	0,18	0,18	0,19
pmed7		10	0,47	0,46	0,48	0,48	0,57	0,58	0,54	0,55
pmed8		20	2,20	2,21	1,77	1,99	1,82	1,86	1,66	1,68
pmed9		40	8,83	9,48	11,19	11,59	7,78	7,87	8,67	8,76
pmed10		67	8,50	12,32	9,65	10,98	13,53	13,63	17,62	17,73
pmed11	300	5	0,36	0,36	0,36	0,36	0,37	0,38	0,35	0,35
pmed12		10	0,90	0,90	1,28	1,27	1,13	1,15	1,05	1,06
pmed13		30	10,27	10,28	15,98	16,25	11,35	11,46	10,50	10,62
pmed14		60	30,59	26,91	52,36	55,91	33,65	33,76	28,56	28,76
pmed15		100	29,40	35,87	36,45	41,09	54,11	54,09	62,51	62,88
pmed16	400	5	0,71	0,77	0,59	0,59	0,53	0,53	0,61	0,62
pmed17		10	1,59	1,51	1,88	1,79	3,30	3,35	1,98	2,00
pmed18		40	36,40	38,46	37,83	48,04	36,64	37,03	37,92	38,31
pmed19		80	63,98	58,97	59,46	79,60	55,30	55,51	71,59	71,97
pmed20		133	70,12	74,32	73,47	82,21	92,70	124,28	115,15	115,17
pmed21	500	5	0,70	0,69	0,84	0,85	1,10	1,12	1,06	1,08
pmed22		10	3,87	3,88	3,01	3,02	3,21	3,25	3,78	3,83
pmed23		50	94,43	78,78	101,65	74,27	60,43	60,87	69,42	70,03
pmed24		100	144,56	142,31	172,32	170,15	128,48	128,65	132,10	132,69
pmed25		167	142,28	176,19	183,76	203,37	247,53	247,70	275,64	288,12
pmed26	600	5	1,46	1,47	1,21	1,32	2,04	2,08	1,49	1,51
pmed27		10	4,65	4,68	4,81	5,34	4,51	4,59	4,13	4,20
pmed28		60	88,30	98,27	112,26	141,94	124,13	125,33	103,63	104,55
pmed29		120	169,55	195,14	223,30	242,81	230,26	232,00	317,14	318,55
pmed30		200	249,32	291,06	225,94	323,92	330,84	330,72	456,30	440,62
pmed31	700	5	1,92	1,92	2,37	2,38	2,39	2,43	2,84	2,89
pmed32		10	5,20	5,20	7,54	6,58	5,52	5,60	5,53	5,62
pmed33		70	290,75	283,93	231,46	256,76	175,13	176,43	193,81	195,37
pmed34		140	382,23	319,48	385,80	382,48	328,43	329,29	376,92	376,62
pmed35	800	5	1,88	1,88	2,54	2,53	3,29	3,33	2,32	2,35
pmed36		10	7,00	7,03	11,15	9,86	7,29	7,43	7,82	7,98
pmed37		80	254,53	279,05	290,31	306,26	235,19	237,01	240,84	242,62
pmed38	900	5	3,93	3,93	3,62	3,62	3,17	3,22	4,43	4,48

kontynuacja na następnej stronie

Problem			równe wagi zapotrzebowania				zróżnicowane wagi zapotrzebowania			
			inicjalizacja zachłanna		inicjalizacja losowa		inicjalizacja zachłanna		inicjalizacja losowa	
nazwa	m	n	CPU[s]				CPU[s]			
			<sup>w</sup> VNS	<sup>w</sup> VNS <sub>r</sub>	<sup>w</sup> VNS	<sup>w</sup> VNS <sub>r</sub>	<sup>w</sup> VNS	<sup>w</sup> VNS <sub>r</sub>	<sup>w</sup> VNS	<sup>w</sup> VNS <sub>r</sub>
pmed39		10	9,50	9,43	8,51	8,52	14,40	14,65	9,27	9,43
pmed40		90	418,86	515,95	434,68	512,82	293,37	295,78	375,88	378,38

## 9.4. Wnioski

Zaproponowana w rozdziale metoda VNS z agregacją WOWA pozwala na rozwiązywanie dyskretnych problemów lokalizacyjnych ze zróżnicowanymi wielkościami zapotrzebowania. Za jej pomocą można rozwiązywać problemy o dużych rozmiarach.

Analiza jakościowa została przeprowadzona na zbiorze niewielkich problemów do rozmiaru 30 lokalizacji, dla których udało się znaleźć rozwiązania optymalne za pomocą modeli programowania matematycznego z wcześniejszych rozdziałów. Rozwiązania znalezione przez metodę przybliżoną charakteryzują się niewielkimi odstępami od rozwiązań optymalnych. Dotyczy to nawet wersji VNS bez regularyzacji. Wyjątek stanowią problemy z kryterium centrum, dla których wersja metody VNS bez regularyzacji znajduje rozwiązania z kilkukrotnie większymi odstępami. W tym wypadku zdecydowanie pomaga warunek regularyzacji, który dla większości problemów zmniejsza znacząco odstęp, co widoczne jest zwłaszcza dla większych problemów (w ramach tych niewielkich problemów). Ze względu na znajomość optymalnych rozwiązań tylko 10 dużych problemów z kryterium średniej nie przeprowadzono pełnej analizy jakościowej dla dużych rozmiarów. Jednak analizując tych kilka problemów o znanych rozwiązaniach optymalnych, okazuje się, że przybliżona metoda znalazła rozwiązania optymalne 9/10 przypadków przy inicjalizacji zachłannej oraz 10/10 przy inicjalizacji losowej. Uwzględniając te rezultaty oraz biorąc pod uwagę fakt, że pod względem jakościowym rozwiązania metody VNS z agregacją WOWA są lepsze niż rozwiązania metody VNS z agregacją OWA dla niewielkich problemów, można przypuszczać, że również dla dużych problemów metoda z agregacją WOWA znajduje rozwiązania dobrej jakości.

Na dużych problemach sprawdzono wydajność zaproponowanej metody. Ogólna charakterystyka czasowa metody dla problemów z biblioteki OR jest podobna do charakterystyki metody z agregacją OWA. Oprócz oczywistego wzrostu dla większych rozmiarów jeszcze bardziej widoczny jest wzrost czasu dla problemów z większą liczbą punktów obsługi. Maksymalne czasy mieszczą się jednak w granicach kilku minut.

Porównując czasy metody z operatorem WOWA z czasami metody z operatorem OWA dla problemów z równymi wagami zapotrzebowania, można określić koszty wydajnościowe związane z uwzględnieniem wartości zapotrzebowań odbiorców. Dla każdego z trzech badanych kryteriów (średniej, centrum, zawężonej średniej) czasy metody z agregacją WOWA wzrosły o ok. 30%–75%, przy czym różnice zwiększały się wraz ze wzrostem liczby punktów

obsługi. W ramach 40 problemów czas rozwiązania wydłużył się średnio o ok. 50%. Nieco mniejsze wzrosty dotyczyły wersji bez regularyzacji dla kryterium centrum. Rezultaty wydają się zadowalające, szczególnie jeśli weźmiemy pod uwagę alternatywny scenariusz dezagregacji, który prowadzi zazwyczaj do zdecydowanie większego zwielokrotnienia rozmiaru problemu, a tym samym czasu rozwiązania. Poza tym należy pamiętać, że zmodyfikowana metoda VNS z agregacją OWA przedstawiona we wcześniejszych rozdziałach osiąga czasy kilkukrotnie krótsze niż metoda z literatury. Biorąc pod uwagę stopień wydłużenia czasu metody z agregacją WOWA w stosunku do zmodyfikowanej metody z agregacją OWA, ta pierwsza i tak będzie zazwyczaj kilkukrotnie krótsza (choć z mniejszym współczynnikiem) niż metoda z agregacją OWA z literatury. Starano się również ocenić różnice pomiędzy czasami rozwiązań problemów z równymi i ze zróżnicowanymi wagami zapotrzebowania. Różnice zależą silnie od konkretnych instancji problemów, jednak ich zakres nie jest duży — od 50% skrócenia do jednokrotnego wydłużenia czasu rozwiązania. Wydaje się, że czasy są zbliżone, być może z nieco dłuższymi czasami dla zróżnicowanych wag zapotrzebowania w przypadku niektórych wersji metody. Nieco dłuższe czasy dla zróżnicowanych wag zapotrzebowania dotyczą zwłaszcza kryterium centrum. Uzyskane wyniki pokazują, że zaproponowana metoda pozwala na efektywne rozwiązywanie dużych dyskretnych problemów lokalizacyjnych ze zróżnicowanymi zapotrzebowaniami i uzyskiwanie rozwiązań o dobrej jakości w sensie rozkładu ocen.

## 10. Podsumowanie

### 10.1. Wnioski

W pracy badano dyskretne wielokryterialne problemy lokalizacyjne, ze szczególnym uwzględnieniem modeli preferencji, które pozwalają uzyskiwać rozwiązania optymalne w sensie rozkładu ocen. Takie podejście jest ważne ze względu na specyfikę problemu, którego rozwiązania powinny spełniać warunek bezstronności, gdzie rozkład wartości ocen jest ważniejszy od ich przypisania do konkretnych kryteriów. Badane techniki umożliwiają także zapewnienie sprawiedliwości w sensie efektywności wyrównującej, gdzie dąży się do polepszenia ocen gorszych kosztem ocen lepszych, co przybliży do pełnej równości. Jest to szczególnie ważne przy rozmieszczaniu obiektów użyteczności publicznej, które zgodnie z regulacjami prawnymi powinny zapewniać jednakową obsługę wszystkim obywatelom (klientom).

Analizowane podejście wywodzi się od operatora uporządkowanej średniej ważonej (OWA), którego wykorzystanie prowadzi do tzw. problemu uporządkowanej średniej (OMP). Oprócz pożądaných własności dystrybucyjnych podejście to pozwala również na uogólnienie wielu kryteriów stosowanych dotychczas w różnych problemach lokalizacyjnych, w tym dwóch najczęściej stosowanych: kryterium średniej i kryterium centrum. Są to cele w pewnym sensie antagonistyczne, a możliwość ich modelowania przez pojedynczą technikę świadczy o jej dużej elastyczności. Co więcej, uporządkowana średnia ważona (OWA) pozwala na wyrażenie kryteriów kompromisowych w całym spektrum preferencji od koniunkcji do alternatywy poszczególnych ocen. Ze względu na warunek bezstronności operator OWA nie pozwala jednak na uwzględnienie wag wartościujących dla konkretnych kryteriów. W problemie lokalizacyjnym nie można zatem bezpośrednio uwzględnić wielkości zapotrzebowań, które wyrażają charakterystykę poszczególnych lokalizacji (klientów). Co prawda, możliwe jest teoretyczne zdezagregowanie do pojedynczych jednostek zapotrzebowania i utożsamienie ich z klientami, jednak prowadzi to zazwyczaj do drastycznego zwiększenia rozmiaru problemu, co zazwyczaj uniemożliwia jego rozwiązanie, a tym samym czyni niepraktycznym omawiane przekształcenie. Popularne kryterium średniej umożliwia bezpośrednio uwzględnienie wag wartościujących, co prowadzi do średniej ważonej. Jednak średnia ważona nie zapewnia warunku bezstronności. Z kolei w przypadku kryterium centrum wagi wartościujące nie wpływają na uzyskiwane rozwiązania, co czyni je nieistotnymi przy tym podejściu. Aby uwzględnić bezpośrednio wagi wartościujące oraz jednocześnie zachować własność bezstronności w sensie rozkładu ocen, zastosowano w pracy operator wartościowanej uporządkowanej średniej ważonej (WOWA), będący rozszerzeniem operatora OWA. Doprowadziło to do nowej klasy problemów, które można

określić mianem WOMP, a będących rozszerzeniem klasy OMP. Dla tych nowych problemów opracowano modele optymalizacyjne programowania matematycznego, które pozwalają na uzyskiwanie optymalnych rozwiązań agregacji WOWA. W przypadku preferencji zgodnych z relacją wyrównującą, co odpowiada malejącym wagom preferencji, zaproponowane modele WOWA stanowią zadania PL — precyzyjnie mówiąc, takie sformułowanie jest możliwe dla wag nierosnących. W ogólnym przypadku, dla dowolnych nieujemnych wag preferencji, modele WOWA zostały sformułowane w postaci zadań PCLM. Sformułowane zadania umożliwiają kompleksowe potraktowanie szerokiej klasy dyskretnych wielokryterialnych problemów lokalizacyjnych oraz dają nieosiągalną dotychczas elastyczność ich modelowania z jednoczesnym uwzględnieniem własności dystrybucyjnych i wielkości zapotrzebowań.

Większa elastyczność i nowe możliwości modelowania preferencji, które stwarza operator OWA i jego rozszerzona wersja WOWA, są jednak okupione większą złożonością obliczeniową. Ponieważ zagadnienia lokalizacyjne nawet dla podstawowych kryteriów, zwłaszcza w przypadku dyskretnym, są problemami trudnymi, zastosowanie operatorów porządkowych musi być zrealizowane możliwie efektywnie. W pracy analizowano różne sformułowania operatorów porządkowych i możliwości poprawy ich wydajności przez zastosowanie nadmiarowych ograniczeń.

W przypadku operatora OWA szczegółowa analiza została przeprowadzona dla zaczerpniętych z literatury dwóch ogólnych modeli OWA w postaci zadań PCLM i jednego modelu PL dla nierosnących wag preferencji. Zaproponowano także nowe hybrydowe modele OWA w postaci zadań PCLM będące rozszerzeniem modelu PL, które można stosować z dowolnymi wagami preferencji. Pokazano teoretyczne własności rozwiązań optymalnych powyższych zadań. Wyprowadzono zależność pomiędzy dwoma sformułowaniami PCLM z literatury oraz przeprowadzono obszerne wydajnościowe testy obliczeniowe na dużej liczbie zróżnicowanych problemów, które obejmowały różne typy kryteriów zdefiniowane przez wektor wag preferencji. Uzyskane wyniki pokazały, że odpowiednie nadmiarowe ograniczenia mogą znacząco zwiększyć wydajność obliczeniową modeli PCLM dla pewnych typów problemów. Dotyczy to klasy problemów o niezerowych wszystkich wagach preferencji. Ponadto rozważany zbiór testowych instancji wskazuje, że jeden z modeli jest znacznie efektywniejszy niż drugi dla wszystkich rozważanych typów problemów. To potwierdza intuicyjne przypuszczenie oparte na charakterze teoretycznej zależności pomiędzy ograniczeniami w obu modelach. Wydajność nowych modeli hybrydowych OWA zależy silnie od typu problemu. Modele te okazały się użyteczne dla części typów problemów, w tym szczególnie dla kryterium zawężonej średniej. Osiągane tam czasy były zdecydowanie lepsze od czasów osiąganych przez modele PCLM z literatury. Jest to tym ważniejsze, że kryterium zawężonej średniej wydaje się najbardziej praktycznym spośród badanych problemów o niemonotonicznych wagach. Należy jednak zaznaczyć, że dla pewnych typów problemów (TC5–TC8) wydajność modeli hybrydowych była znacznie gorsza niż modeli PCLM z literatury. Przy czym te problemy wydają się mieć małą wartość praktyczną. Gdy wagi preferencji są nierosnące, co ma miejsce dla preferencji

zgodnych z wyrównującą efektywnością, która wymaga ściśle malejących wag, zadanie optymalizacji OWA można sformułować w postaci liniowej. Osiągnięte rezultaty potwierdzają, że wydajność modeli liniowych jest znacznie lepsza niż ogólnych modeli OWA w postaci zadań PCLM i nawet dla małych rozmiarów czasy rozwiązań wersji liniowych są ok. 1–2 rzędy wielkości krótsze. To jednoznacznie sugeruje użycie takich sformułowań dla problemów z monotonicznymi wagami preferencji. Wydaje się również, że ograniczenia nadmiarowe nie pomagają w przypadku sformułowań liniowych i dla części problemów mogą wydłużać czas rozwiązania.

Testy wydajnościowe modeli z operatorem WOVA zostały przeprowadzone analogicznie jak dla modeli OWA. Również tutaj zadanie PL dla monotonicznych wag preferencji osiąga znacznie lepsze rezultaty niż ogólne modele PCLM. Podobnie jak poprzednio można także uznać, że ograniczenia nadmiarowe nie poprawiają wydajności sformułowania liniowego WOVA. Testy modeli PCLM dla dowolnych wag preferencji pokazują lepszą wydajność modelu z ograniczoną liczbą zmiennych binarnych. Dodanie do niego zaproponowanych ograniczeń nadmiarowych pozwala jeszcze kilkukrotnie skrócić czasy rozwiązań pewnych typów problemów. Pomimo tej poprawy zadania PCLM agregacji WOVA umożliwiają rozwiązanie problemów tylko do kilkunastu lokalizacji.

Jak pokazują powyższe wyniki, ze względu na złożoność dyskretnych problemów lokalizacyjnych uniwersalne dokładne metody pozwalają na rozwiązywanie tylko niewielkich problemów. Dotyczy to również problemów, które można rozwiązać w krótszym czasie przy zastosowaniu nadmiarowych ograniczeń. Do rozwiązania problemów o rzeczywistych rozmiarach należy zastosować metody przybliżone.

Rozwijana w pracy metoda przeszukiwania zmiennego sąsiedztwa (VNS) okazała się skuteczna dla średnich i dużych problemów (rzędu kilkuset lokalizacji). Wprowadzono modyfikacje, głównie do algorytmu wymiany, które ograniczyły znacząco przeszukiwanie w nieobiecujących kierunkach, odrzucając część rozwiązań na podstawie oszacowanych możliwie najlepszych wartości funkcji celu. Zmniejszono również znacząco koszt obliczeniowy dokładnego wyznaczenia rozwiązania, a zwłaszcza odpowiadającej mu wartości funkcji celu, wykorzystując informacje z poprzedniego rozwiązania. Z drugiej strony dodano warunek regularyzacyjny, który pozwala poprawić jakość uzyskiwanych rozwiązań dla pewnych typów problemów, zwłaszcza dla kryterium centrum. Porównanie zmodyfikowanej metody VNS w stosunku do metody oryginalnej, ale także do innych przybliżonych metod z literatury, pokazało bardzo dobrą jakość uzyskiwanych rozwiązań, a zwłaszcza zdecydowaną poprawę dla kryterium centrum, dla którego odstęp od rozwiązania optymalnego został poprawiony ok. 20–30 razy. To sprawia, że zmodyfikowana metoda VNS z agregacją OWA stanowi uniwersalną przybliżoną metodę, która pozwala na uzyskiwanie w rozsądnym czasie dobrych jakościowo rozwiązań w sensie rozkładu ocen dla dyskretnych problemów lokalizacyjnych.

Zaproponowana rozszerzona metoda VNS z agregacją WOVA również uzyskuje dobre rezultaty. Chociaż czasy rozwiązania rosną w stosunku do metody z agregacją OWA, to jest

to wzrost niewielki, wynoszący średnio ok. 50%. Wydaje się to dobrym rezultatem, szczególnie biorąc pod uwagę alternatywny scenariusz polegający na zdezagregowaniu problemu do równych porcji zapotrzebowania, który wymaga dodatkowych nakładów na transformację problemu i może prowadzić do wielokrotnego wzrostu rozmiaru problemu.

## 10.2. Podsumowanie twórczego wkładu

Zasadniczo wszystkie cele postawione w pracy zostały osiągnięte.

W ramach celu 1 zastosowano operator wartościowanej uporządkowanej średniej ważonej WOWA, przedstawiony w rozdziale 6, do dyskretnego wielokryterialnego problemu lokalizacyjnego, co doprowadziło do nowej klasy dyskretnych problemów lokalizacyjnych, będących wartościowanym rozszerzeniem problemu uporządkowanej średniej (OMP). Takie podejście stwarza niedostępną wcześniej możliwość uwzględnienia wielkości zapotrzebowań przy jednoczesnym dążeniu do osiągnięcia rozwiązań optymalnych w sensie rozkładu ocen. W rozdziale 7, dla tak zdefiniowanych problemów, opracowano modele optymalizacyjne agregacji WOWA w oparciu o programowanie matematyczne. Dla nierosnących wag preferencji agregację WOWA sformułowano w postaci zadania PL. Sformułowanie to jest analogiczne do modelu wykorzystywanego do wspomagania decyzji w warunkach ryzyka i uogólnia liniowy model OWA. W ogólności, gdy warunek monotoniczności wag nie jest spełniony, model musi być odpowiednio rozszerzony. Wymagało to dodania zmiennych binarnych i odpowiednich ograniczeń, co doprowadziło do zadania PCLM, które może być stosowane dla dowolnych wag preferencji — przedstawiono dwie wersje zadań PCLM agregacji WOWA. Zaproponowano także nadmiarowe ograniczenia do ściślejszego opisu zbioru rozwiązań dopuszczalnych badanych problemów. Tym samym obserwacje z rozdziału 6 i parametryczne modele opracowane w rozdziale 7 dowodzą, że możliwe jest sformułowanie parametrycznych modeli optymalizacji wielokryterialnej dla dyskretnych zagadnień lokalizacyjnych, które uwzględniają zarówno preferencje efektywnościowo-sprawiedliwościowe, jak i zróżnicowane zapotrzebowania — czyli tezę 1 niniejszej pracy.

W ramach celu 2 zbadano różne modele agregacji OWA (rozdział 5) i WOWA (rozdział 7), a także wpływ nadmiarowych ograniczeń na ich wydajność. Przeprowadzono obszerne testy obliczeniowe, które pozwoliły na porównanie wydajności poszczególnych modeli. W przypadku modeli OWA uzyskane wyniki pokazały, że model M1 wywodzący się bezpośrednio z operatora OWA jest wydajniejszy od modelu M2 oraz że nadmiarowe ograniczenia mogą znacząco zwiększyć wydajność modeli PCLM dla części typów problemów. Dotyczy to zwłaszcza problemów o niezerowych wszystkich wagach preferencji. Należy jednak zaznaczyć, że dla części problemów, szczególnie tych, gdzie minimalizacji podlegają tylko największe odległości, czasy rozwiązań uległy pogorszeniu. W obszarze celu 2 zaproponowano także hybrydowe modele OWA, stanowiące rozszerzenie sformułowania liniowego, które uzyskały lepsze czasy dla części typów problemów, a szczególnie dla kryterium zawężonej średniej. W przypadku

agregacji WOWA zbadano modele zaproponowane w ramach celu 1. Spośród dwóch modeli PCLM lepszy okazał się model MW2. Część zaproponowanych nadmiarowych ograniczeń pozwoliła na kilkukrotne skrócenie czasu rozwiązań modelu MW2 dla niektórych typów problemów. Testy potwierdziły również znacznie lepszą wydajność liniowych sformułowań operatorów OWA i WOWA, ale także pokazały, że nadmiarowe ograniczenia nie poprawiają w ogólności ich wydajności. Uzyskane wyniki stanowią więc częściowe potwierdzenie tezy 2, gdyż wśród badanych nadmiarowych ograniczeń są takie, które poprawiają efektywność modeli optymalizacji wielokryterialnej w sensie najlepszego rozkładu ocen dla dyskretnych problemów lokalizacyjnych, przy czym dotyczy to modeli PCLM. Warto przy tym zaznaczyć, że w pracy badane były ograniczenia nadmiarowe dotyczące samego sformułowania operatorów OWA/WOWA, a nie całego zadania włącznie z dyskretną częścią rzeczowego modelu lokalizacyjnego. To wydaje się uzasadniać brak poprawy wydajności sformułowań PL operatorów OWA/WOWA, w przypadku których dodanie ograniczeń nadmiarowych nie poprawi dolnego oszacowania (relaksacji liniowej), a może tylko wpłynąć na inne elementy algorytmu rozwiązywania — taki wpływ okazuje się niewystarczający do poprawy efektywności modeli. To również tłumaczy brak poprawy modelu M1 agregacji OWA dla kryterium centrum, który degeneruje się do modelu liniowego. W przypadku kryterium centrum i jemu zbliżonych podobna sytuacja może mieć miejsce również dla innych modeli (np. modelu M2), które chociaż nie degenerują się do modeli liniowych, to jednak znacznie się upraszczają dla problemów z dużą liczbą zerowych wag preferencji. W takich przypadkach nadmiarowe ograniczenia w większym stopniu utrudniają (sztucznie) strukturę problemu niż uściślają jego opis. Hipotezę tę zdają się potwierdzać znacznie krótsze czasy rozwiązania problemów ze znaczną przewagą zerowych wag preferencji (np. typy TC2, TC3). Wydaje się, że te obserwacje tym bardziej pozwalają na stwierdzenie prawdziwości tezy 2 dla modeli PCLM operatorów OWA/WOWA, pamiętając o wyłączeniu z niej specyficznych problemów ze zdegenerowaną strukturą.

W ramach celu 3 w rozdziale 8 zaproponowano zmienioną przybliżoną metodę VNS do rozwiązywania problemów z kryterium OWA o średnich i dużych rozmiarach. Przeprowadzono również szczegółowe testy obliczeniowe dla różnorodnych problemów. Zaproponowano zarówno modyfikacje poprawiające wydajność metody, jak i jakość uzyskiwanych rozwiązań. Modyfikacje wydajnościowe ograniczyły przeszukiwanie w nieperspektywicznych kierunkach i zmniejszyły koszt obliczeniowy oceny nowego rozwiązania. W celu poprawy jakości dodano warunek regularyzacyjny, który zintensyfikował proces przeszukiwania dla części typów problemów. Dotyczyło to zwłaszcza kryterium centrum, gdzie osiągnięto zdecydowaną poprawę jakości — odstęp od rozwiązania optymalnego został zmniejszony kilkadziesiąt razy. Warunek regularyzacyjny wydłuża oczywiście czas rozwiązania, ale zazwyczaj tylko w niewielkim stopniu. Tam gdzie zmiany są większe (kryterium centrum), zdecydowana poprawa jakości uzasadnia zwiększone koszty obliczeniowe, tym bardziej że ostateczne czasy i tak są kilkakrotnie krótsze niż w metodzie oryginalnej. Warto również dodać, że dla problemów, gdzie



wszystkie wagi preferencji są niezerowe (np. kryterium średniej), warunek regularyzacji nie jest aktywny i zupełnie nie wpływa na rozwiązanie (w tym na wydajność). Rozwiązania uzyskiwane dla tych problemów cechowały się bardzo małym odstępem od rozwiązań optymalnych także dla oryginalnej metody i w ich przypadku warunek regularyzacji nie jest potrzebny. Dzięki zaproponowanym zmianom opracowana metoda VNS skróciła wielokrotnie czas rozwiązania, a jednocześnie poprawiła jakość uzyskiwanych rozwiązań. Tym samym stanowi ona uniwersalną przybliżoną metodę pozwalającą efektywnie uzyskiwać rozwiązania optymalne lub bliskie optymalnym w sensie rozkładu ocen dla dowolnych wag preferencji. W rozdziale 9 zaproponowano także metodę VNS z agregacją WOWA. Pozwala ona na uzyskiwanie dobrych jakościowo rozwiązań w sensie rozkładu ocen dla problemów lokalizacyjnych ze zróżnicowanymi wagami zapotrzebowania. Wprowadzone wcześniej modyfikacje ułatwiły adaptację metody do operatora WOWA. Chociaż czasy metody wzrosły w stosunku do metody z operatorem OWA, to dla dużych problemów z biblioteki OR nie przekraczają one kilku minut. Metody opracowane w rozdziałach 8 i 9 stanowią jednocześnie potwierdzenie tezy 3, że możliwa jest konstrukcja przybliżonej metody rozwiązywania dyskretnych problemów lokalizacyjnych, która dla dowolnych wag preferencji pozwala efektywnie osiągać rozwiązania dobrej jakości w sensie rozkładu ocen dla problemów o dużych rozmiarach (rzędu kilkuset lokalizacji).

### **10.3. Dalsze kierunki badań**

Przeprowadzone badania pokazują, że uniwersalne metody rozwiązywania (za pomocą solverów) zadań optymalizacyjnych w sensie rozkładu ocen dla dyskretnych problemów lokalizacyjnych radzą sobie jedynie z problemami o niewielkim rozmiarze. Dotyczy to zwłaszcza zadań PCLM, które pozwalają na zastosowanie dowolnych wag preferencji. Zasadne wydaje się więc opracowanie specjalistycznych metod dokładnych dedykowanych dyskretnym problemom lokalizacyjnym, które wykorzystując specyficzne własności tych zagadnień, umożliwiłyby uzyskanie rozwiązań optymalnych dla problemów o większych rozmiarach.

Dokładniejszej analizie warto również poddać sformułowania dla monotonicznych wag preferencji, które zawężają w pewnym stopniu zakres dopuszczalnych preferencji, jednak nadal zapewniają dużą elastyczność. Tym bardziej, że monotoniczne wagi są zgodne z relacją wyrównującą, która pozwala na uzyskiwanie rozwiązań sprawiedliwych (wyrównująco efektywnych), co jest ważne ze względu na charakter zagadnień lokalizacyjnych. Ograniczenie do tego typu preferencji wynika więc często z uwarunkowań zewnętrznych, niezależnych od kwestii elastyczności modelowania i wydajności metod rozwiązywania. Jednocześnie, ze względu na wypukłość (przy minimalizacji), są to zadania PL zapewniające znacznie lepszą wydajność niż zadania PCLM, a przez to pozwalające na rozwiązywanie problemów o większych rozmiarach.

Interesujące wydaje się porównanie przybliżonej metody VNS z popularną w ostatnich

latach heurystyką TA (ang. Threshold Accepting). Zaproponowany w pracy warunek regularyzacji może stanowić także punkt wyjścia do opracowania hybrydowej metody czerpiącej z obu heurystyk.

Niezależnie od powyższych kierunków dalszych badań wskazane jest również rozszerzenie badanych technik na przypadek dyskretnych problemów lokalizacyjnych z ograniczoną pojemnością punktów obsługi. Dotyczy to zarówno modeli optymalizacyjnych programowania matematycznego, jak i przybliżonej metody VNS.

## Bibliografia

- [1] G. Andreatta, F. M. Mason.  $k$ -eccentricity and absolute  $k$ -centrum of a probabilistic tree. *European Journal of Operational Research*, 19(1):114–117, 1985.
- [2] G. Andreatta, F. M. Mason. Properties of the  $k$ -centra in a tree network. *Networks*, 15(1):21–25, 1985.
- [3] A. B. Atkinson. On the measurement of inequality. *Journal of Economic Theory*, 2(3):244–263, 1970.
- [4] F. Auerbach. Das gesetz der bevölkerungskonzentration. *Petermanns Geographische Mitteilungen*, 59:73–76, 1913.
- [5] R. L. Axtell. Zipf distribution of US firm sizes. *Science*, 293(5536):1818–1820, 2001.
- [6] J. E. Beasley. A note on solving large  $p$ -median problems. *European Journal of Operational Research*, 21(2):270–273, 1985.
- [7] J. E. Beasley. OR-library: distributing test problems by electronic mail. *Journal of the Operational Research Society*, 41(11):1069–1072, 1990.
- [8] O. Berman. Mean-variance location problems. *Transportation Science*, 24(4):287–293, 1990.
- [9] O. Berman, E. H. Kaplan. Equity maximizing facility location schemes. *Transportation Science*, 24(2):137–144, 1990.
- [10] N. Boland, P. Domínguez-Marín, S. Nickel, J. Puerto. Exact procedures for solving the discrete ordered median problem. *Computers & Operations Research*, 33(11):3270–3300, 2006.
- [11] R. Bosch, M. Trick. Integer programming. E. K. Burke, G. Kendall, redaktorzy, *Search Methodologies*, strony 69–95. Springer, 2005.
- [12] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein. *Wprowadzenie do Algorytmów*. Klasyka Informatyki. Wydawnictwa Naukowo-Techniczne, 2004.
- [13] J. Current, H. Min, D. Schilling. Multiobjective analysis of facility location decisions. *European Journal of Operational Research*, 49(3):295–307, 1990.
- [14] P. Czyżak, A. Jaskiewicz. A multiobjective metaheuristic approach to the location of petrol stations by the capital budgeting model. *Control and Cybernetics*, 25:177–187, 1996.
- [15] E. Damiani, S. De Capitani di Vimercati, P. Samarati, M. Viviani. A WOVA-based aggregation technique on trust values connected to metadata. *Electronic Notes in Theoretical Computer Science*, 157(3):131–142, 2006.
- [16] E. D. Dijkstra. The ladder theorem. <https://www.cs.utexas.edu/users/EWD/ewd12xx/EWD1258a.PDF>, 3 1997.
- [17] P. Domínguez-Marín. *The Discrete Ordered Median Problem: Models and Solution Methods*. Springer, 2003.
- [18] P. Domínguez-Marín, S. Nickel, P. Hansen, N. Mladenović. Heuristic procedures for solving the discrete ordered median problem. *Annals of Operations Research*, 136(1):145–173, 2005.

- [19] Z. Drezner, H. W. Hamacher, redaktorzy. *Facility Location: Applications and Theory*. Springer, Berlin, 2004.
- [20] J. Eeckhout. Gibrat's law for (all) cities. *The American Economic Review*, 94(5):1429–1451, 2004.
- [21] S. Elloumi, M. Labbé, Y. Pochet. A new formulation and resolution method for the p-center problem. *INFORMS Journal on Computing*, 16(1):84–94, 2004.
- [22] E. Erkut. Inequality measures for location problems. *Location Science*, 1(3):199–217, 1993.
- [23] E. Fernández, M. A. Pozo, J. Puerto. Ordered weighted average combinatorial optimization: Formulations and their properties. *Discrete Applied Mathematics*, 169:97–118, 2014.
- [24] K. Fleszar, K. S. Hindi. An effective VNS for the capacitated p-median problem. *European Journal of Operational Research*, 191(3):612–622, 2008.
- [25] K. Fleszar, W. Ogryczak. On generalized OWA approach to support location and routing decisions. A. Jaskiewicz, M. Kaczmarek, J. Żak, M. Kubiak, redaktorzy, *Advanced OR and AI Methods in Transportation, Proceedings of 10th EWGT Meeting and 16th Mini-EURO Conference*, strony 746–751, Poznań, 2005.
- [26] D. K. Foley. Resource allocation and the public sector. *Yale Economics Essays, Spring 1967*, 7(1):45–98, 1967.
- [27] R. L. Francis, J. A. White, L. F. McGinnis. *Facility Layout and Location: An Analytical Approach*. Prentice-Hall Englewood Cliffs, NJ, 1974.
- [28] R. Garfinkel, G.L. Nemhauser. *Programowanie Całkowitoliczbowe*. PWN, 1978.
- [29] M. Grabisch, S. A. Orlovski, R. R. Yager. Fuzzy aggregation of numerical preferences. R. Slowiński, redaktor, *Fuzzy Sets in Decision Analysis, Operations Research and Statistics*, strony 31–68. Springer, 1998.
- [30] J. Halpern. Finding minimal center-median convex combination (cent-dian) of a graph. *Management Science*, 24(5):535–544, 1978.
- [31] P. Hansen, N. Mladenović. Variable neighborhood search for the p-median. *Location Science*, 5(4):207–226, 1997.
- [32] M. Hapke, A. Jaskiewicz, J. Żak. The design of the physical distribution system with the application of the multiple objective mathematical programming. Case study. T. Trzaskalik, J. Michnik, redaktorzy, *Multiple Objective and Goal Programming*, strony 297–309. Springer, 2002.
- [33] K. S. Hindi, K. Pieńkosz. Efficient solution of large scale, single-source, capacitated plant location problems. *Journal of the Operational Research Society*, 50(3):268–274, 1999.
- [34] L. Hurwicz. Optimality criteria for decision making under ignorance. *Cowles Commission Discussion Papers: Statistics*, strona 370, 1951.
- [35] IBM. *IBM ILOG CPLEX Optimization Studio*. <http://www-03.ibm.com/software/products/en/ibmilogcplexoptistud/>.
- [36] O. Kariv, S. L. Hakimi. An algorithmic approach to network location problems. II: The p-medians. *SIAM Journal on Applied Mathematics*, 37(3):539–560, 1979.
- [37] M. Köppen, K. Yoshida, M. Tsuru, Y. Oie. Annealing heuristic for fair wireless channel allocation by exponential ordered-ordered weighted averaging operator maximization. *11th Annual International Symposium on Applications and the Internet, SAINT 2011, Munich, Germany, 18-21*

- July, 2011, *Proceedings*, strony 538–543. IEEE Computer Society, 2011.
- [38] M. M. Kostreva, W. Ogryczak. Equitable approaches to location problems. J.-C. Thill, redaktor, *Spatial Multicriteria Decision Making and Analysis: A Geographic Information Sciences Approach*, strony 103–126. Ashgate, Brookfield, 1999.
- [39] M. M. Kostreva, W. Ogryczak. Linear optimization with multiple equitable criteria. *RAIRO - Operations Research*, 33(3):275–297, 1999.
- [40] M. M. Kostreva, W. Ogryczak, A. Wierzbicki. Equitable aggregations and multiple criteria analysis. *European Journal of Operational Research*, 158(2):362–377, 2004.
- [41] M. Labbé, D.e Peeters, J.-F. Thisse. Location on networks. M. O. Ball, T. L. Magnanti, C. L. Monma, G. L. Nemhauser, redaktorzy, *Handbooks in Operations Research and Management Science: Network Routing*, wolumen 8, strony 551–624. North-Holland, Amsterdam, 1996.
- [42] R. F. Love, J. J. G. Morris, G. O. Wesolowsky. *Facilities Location: Models & Methods*. Publications in Operations Research. North-Holland, 1988.
- [43] H. Luss. *Equitable Resource Allocation: Models, Algorithms and Applications*. Wiley, 2012.
- [44] O. Maimon. The variance equity measure in locational decision theory. *Annals of Operations Research*, 6(5):147–160, 1986.
- [45] O. Maimon. An algorithm for the Lorenz measure in locational decisions on trees. *Journal of Algorithms*, 9(4):583–596, 1988.
- [46] J. Malczewski, W. Ogryczak. A multiobjective approach to the reorganization of health-service areas: a case study. *Environment and Planning A*, 20(11):1461–1470, 1988.
- [47] M. B. Mandell. Modelling effectiveness-equity trade-offs in public service delivery systems. *Management Science*, 37(4):467–482, 1991.
- [48] C. D. Manning, H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT press, 1999.
- [49] M. T. Marsh, D. A. Schilling. Equity measurement in facility location analysis: a review and framework. *European Journal of Operational Research*, 74(1):1–17, 1994.
- [50] A. W. Marshall, I. Olkin. *Inequalities: Theory of Majorization and its Applications*. Mathematics in Science and Engineering. Academic Press, New York, 1979.
- [51] P. B. Mirchandani, R. L. Francis. *Discrete Location Theory*. Wiley, New York, 1990.
- [52] N. Mladenović, P. Hansen. Variable neighborhood search. *Computers & Operations Research*, 24(11):1097–1100, 1997.
- [53] N. Mladenović, M. Labbé, P. Hansen. Solving the p-center problem with tabu search and variable neighborhood search. *Networks*, 42(1):48–64, 2003.
- [54] R. L. Morrill, J. Symons. Efficiency and equity aspects of optimum location. *Geographical Analysis*, 9(3):215–225, 1977.
- [55] D. Nettleton, J. Muñiz. Processing and representation of meta-data for sleep apnea diagnosis with an artificial intelligence approach. *International Journal of Medical Informatics*, 63(1):77–89, 2001.
- [56] S. Nickel. Discrete ordered Weber problems. B. Fleischmann, R. Lasch, U. Derigs, W. Domschke, U. Rieder, redaktorzy, *Operations Research Proceedings 2000*, strony 71–76. Springer, 2001.
- [57] S. Nickel, J. Puerto. *Location Theory: A Unified Approach*. Springer, Berlin, 2005.

- [58] W. Ogryczak. On cent-dians of general networks. *Location Science*, 5(1):15–28, 1997.
- [59] W. Ogryczak. On the lexicographic minimax approach to location problems. *European Journal of Operational Research*, 100(3):566–585, 1997.
- [60] W. Ogryczak. *Wielokryterialna Optymalizacja Liniowa i Dyskretna*. Wydawnictwa Uniwersytetu Warszawskiego, Warszawa, 1997.
- [61] W. Ogryczak. On the distribution approach to location problems. *Computers & Industrial Engineering*, 37(3):595–612, 1999.
- [62] W. Ogryczak. Inequality measures and equitable approaches to location problems. *European Journal of Operational Research*, 122(2):374–391, 2000.
- [63] W. Ogryczak. Multicriteria models for fair resource allocation. *Control and Cybernetics*, 36(2):303–332, 2007.
- [64] W. Ogryczak. Inequality measures and equitable locations. *Annals of Operations Research*, 167(1):61–86, 2009.
- [65] W. Ogryczak. Ordered weighted enhancement of preference modeling in the reference point method for multiple criteria optimization. *Soft Computing*, 14(5):435–450, 2010.
- [66] W. Ogryczak, P. Olender. On MILP models for the OWA optimization. *Journal of Telecommunications and Information Technology*, 2012(2):5–12, 2012.
- [67] W. Ogryczak, T. Śliwiński. On solving linear programs with the ordered weighted averaging objective. *European Journal of Operational Research*, 148(1):80–91, 2003.
- [68] W. Ogryczak, T. Śliwiński. On direct methods for lexicographic min-max optimization. *Computational Science and Its Applications - ICCSA 2006*, wolumen 3982 serii *Lecture Notes in Computer Science*, strony 802–811, 2006.
- [69] W. Ogryczak, T. Śliwiński. On optimization of the importance weighted OWA aggregation of multiple criteria. *Computational Science and Its Applications - ICCSA 2007, International Conference, Kuala Lumpur, Malaysia, August 26-29, 2007. Proceedings, Part I*, strony 804–817. Springer, 2007.
- [70] W. Ogryczak, T. Śliwiński. On efficient WOWA optimization for decision support under risk. *International Journal of Approximate Reasoning*, 50(6):915–928, 2009.
- [71] W. Ogryczak, T. Śliwiński, A. Wierzbicki. Fair resource allocation schemes and network dimensioning problems. *Journal of Telecommunications and Information Technology*, 2003(3):34–42, 2003.
- [72] W. Ogryczak, A. Tamir. Minimizing the sum of the  $k$  largest functions in linear time. *Information Processing Letters*, 85(3):117–122, 2003.
- [73] W. Ogryczak, M. Zawadzki. Conditional median: a parametric solution concept for location problems. *Annals of Operations Research*, 110(1-4):167–181, 2002.
- [74] P. H. Peeters. Some new algorithms for location problems on networks. *European Journal of Operational Research*, 104(2):299–309, 1998.
- [75] K. Piękosz. *Wybrane Modele i Metody Optymalizacji Alokacji Zasobów*. Oficyna Wydawnicza PW, Warszawa, 2010.
- [76] J. Rawls. *A Theory of Justice*. Belknap Press, 1999.
- [77] M. G. C. Resende, R. F. Werneck. A fast swap-based local search procedure for location problems.

- Annals of Operations Research*, 150(1):205–230, 2007.
- [78] D. Richard, H. Beguin, D. Peeters. The location of fire stations in a rural environment: A case study. *Environment and Planning A*, 22(1):39–52, 1990.
- [79] A. Sen. *On Economic Inequality*. Oxford University Press, 1973.
- [80] P. J. Slater. Centers to centroids in graphs. *Journal of Graph Theory*, 2(3):209–222, 1978.
- [81] Z. Stanimirović, J. Kratica, D. Dugošija. Genetic algorithms for solving the discrete ordered median problem. *European Journal of Operational Research*, 182(3):983–1001, 2007.
- [82] R. E. Steuer. *Multiple Criteria Optimization: Theory, Computation and Application*. Wiley, 1986.
- [83] A. Tamir. The k-centrum multi-facility location problem. *Discrete Applied Mathematics*, 109(3):293–307, 2001.
- [84] V. Torra. The weighted OWA operator. *International Journal of Intelligent Systems*, 12(2):153–166, 1997.
- [85] V. Torra. The WOWA operator and the interpolation function  $W^*$ : Chen and Otto's interpolation method revisited. *Fuzzy Sets and Systems*, 113(3):389–396, 2000.
- [86] V. Torra, Y. Narukawa. *Modeling Decisions: Information Fusion and Aggregation Operators*. Springer, 2007.
- [87] A. Valls, V. Torra. Using classification as an aggregation tool in MCDM. *Fuzzy Sets and Systems*, 115(1):159–168, 2000.
- [88] P. Vincke. *Multicriteria Decision-Aid*. Wiley, 1992.
- [89] R. A. Whitaker. A fast algorithm for the greedy interchange for large-scale clustering and median location problems. *INFOR: Information Systems and Operational Research*, 21:95–108, 1983.
- [90] H. P. Williams. *Model Building in Mathematical Programming*. Wiley, 1978.
- [91] R. R. Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1):183–190, 1988.
- [92] R. R. Yager. Families of OWA operators. *Fuzzy Sets and Systems*, 59(2):125–148, 1993.
- [93] R. R. Yager. Constrained OWA aggregation. *Fuzzy Sets and Systems*, 81(1):89–101, 1996.
- [94] R. R. Yager, J. Kacprzyk, G. Beliakov, redaktorzy. *Recent Developments in the Ordered Weighted Averaging Operators: Theory and Practice*, wolumen 265 serii *Studies in Fuzziness and Soft Computing*. Springer, 2011.
- [95] H. P. Young. *Equity: In Theory and Practice*. Princeton University Press, 1994.
- [96] G. K. Zipf. *Human Behaviour and the Principles of Least Effort*. Addison-Wesley, 1949.

## A. Wyniki modeli PCLM agregacji OWA

### A.1. Wyniki modeli PCLM M1 i M2 agregacji OWA

Tabela A.1: Statystyki czasowe modeli PCLM M1 i M2 agregacji OWA (górny indeks przed czasem oznacza liczbę instancji spośród 15, które przekroczyły ustalony limit czasu 600 s; modele nie naliczane dla danych instancji problemów oznaczone są przez „x”)

Problem			CPU[s]						
typ	m	n	M1 <sub>1</sub>	M1 <sub>2</sub>	M1 <sub>3</sub>	M1 <sub>4</sub>	M1 <sub>5</sub>	M2 <sub>1</sub>	M2 <sub>2</sub>
TC1	8	2	0,42	0,37	0,04	0,28	0,04	0,19	4,05
		3	0,37	0,33	0,03	0,29	0,03	0,26	5,37
		4	0,25	0,24	0,02	0,21	0,02	0,03	7,24
		5	0,17	0,15	0,01	0,17	0,02	0,04	6,25
	10	3	2,28	1,77	0,07	1,03	0,07	0,94	71,93
		4	1,50	1,06	0,05	0,93	0,04	0,80	128,99
		5	0,83	0,81	0,03	0,65	0,04	0,08	188,66
	12	6	0,50	0,47	0,02	0,41	0,02	0,12	<sup>1</sup> 208,67
		3	24,57	13,54	0,23	6,51	0,19	x	x
		4	11,37	9,08	0,07	5,25	0,08	x	x
		6	1,76	1,63	0,05	1,28	0,05	x	x
	15	7	0,95	1,06	0,04	0,83	0,05	x	x
		4	<sup>4</sup> 278,23	<sup>1</sup> 135,97	1,10	59,95	0,86	x	x
		5	88,29	50,63	0,33	27,74	0,39	x	x
		8	4,01	4,32	0,08	2,89	0,08	x	x
TC2	8	9	2,48	2,57	0,06	1,60	0,07	x	x
		2	0,02	0,02	0,07	0,01	0,06	2,55	0,11
		3	0,02	0,02	0,03	0,02	0,04	1,37	0,09
		4	0,01	0,01	0,02	0,02	0,03	1,04	0,09
	10	5	0,01	0,01	0,02	0,01	0,02	1,22	0,10
		3	0,03	0,04	0,07	0,03	0,08	20,66	0,37
		4	0,03	0,03	0,05	0,03	0,06	8,70	0,30
		5	0,02	0,02	0,04	0,02	0,04	5,82	0,39
	12	6	0,02	0,01	0,03	0,01	0,03	8,04	0,40
		3	0,06	0,07	0,13	0,06	0,14	x	x
		4	0,05	0,06	0,11	0,05	0,10	x	x
		6	0,04	0,03	0,06	0,03	0,07	x	x
	15	7	0,03	0,02	0,04	0,02	0,05	x	x

kontynuacja na następnej stronie



Problem			CPU[s]							
typ	m	n	M1 <sub>1</sub>	M1 <sub>2</sub>	M1 <sub>3</sub>	M1 <sub>4</sub>	M1 <sub>5</sub>	M2 <sub>1</sub>	M2 <sub>2</sub>	
TC3	15	4	0,10	0,11	0,19	0,10	0,23	x	x	
		5	0,11	0,09	0,20	0,11	0,18	x	x	
		8	0,06	0,05	0,11	0,06	0,10	x	x	
		9	0,05	0,04	0,06	0,05	0,08	x	x	
		8	2	0,04	0,06	0,08	0,04	0,10	2,44	0,16
	8	3	0,05	0,06	0,06	0,05	0,10	2,17	0,15	
		4	0,04	0,06	0,06	0,04	0,09	2,09	0,17	
		5	0,04	0,05	0,06	0,04	0,08	2,18	0,15	
		10	3	0,16	0,33	0,18	0,16	0,38	47,07	1,35
		4	0,18	0,30	0,17	0,16	0,37	76,99	1,91	
	5	5	0,14	0,25	0,16	0,14	0,33	57,34	2,03	
		6	0,12	0,20	0,14	0,11	0,25	<sup>1</sup> 142,72	2,17	
		12	3	0,63	1,17	0,45	0,57	1,16	x	x
		4	0,68	1,18	0,48	0,59	1,27	x	x	
		6	0,41	0,76	0,43	0,41	0,86	x	x	
TC4	15	7	0,33	0,61	0,33	0,33	0,73	x	x	
		4	3,90	6,59	1,78	2,45	4,89	x	x	
		5	4,13	6,33	2,29	2,68	4,99	x	x	
		8	1,03	1,80	1,08	0,84	1,99	x	x	
		9	0,72	1,41	0,79	0,68	1,54	x	x	
	8	2	0,25	0,29	0,35	0,20	0,27	4,29	0,56	
		3	0,18	0,24	0,29	0,14	0,23	5,33	0,44	
		4	0,07	0,13	0,15	0,06	0,19	4,05	0,28	
		5	0,02	0,05	0,06	0,02	0,11	2,00	0,09	
		10	3	1,68	1,74	1,85	0,89	1,13	98,67	7,82
	4	4	1,01	1,07	1,33	0,56	0,91	<sup>2</sup> 268,55	7,58	
		5	0,39	0,47	0,67	0,23	0,76	<sup>1</sup> 175,29	4,41	
		6	0,10	0,23	0,28	0,08	0,46	144,02	2,00	
		12	3	14,92	8,71	11,80	3,81	6,34	x	x
		4	7,84	5,84	7,86	1,97	5,53	x	x	
TC5	15	6	0,50	1,61	2,11	0,28	2,28	x	x	
		7	0,10	1,05	1,33	0,10	1,52	x	x	
		4	<sup>2</sup> 259,21	118,69	162,75	56,47	79,38	x	x	
		5	<sup>1</sup> 141,47	87,95	94,90	25,88	49,06	x	x	
		8	10,51	9,70	11,70	2,39	10,61	x	x	
	8	9	2,08	4,00	5,56	0,59	6,22	x	x	
		2	0,12	0,17	0,09	0,11	0,20	0,77	0,86	
		3	0,13	0,19	0,07	0,11	0,17	0,98	1,25	
		4	0,10	0,12	0,04	0,09	0,14	0,57	1,74	
		5	0,07	0,09	0,04	0,07	0,10	0,69	1,52	
	10	3	0,48	0,71	0,32	0,40	0,61	9,22	14,50	
		4	0,35	0,54	0,15	0,32	0,48	6,00	17,51	
		5	0,28	0,37	0,12	0,24	0,41	8,35	13,38	
		6	0,18	0,23	0,06	0,17	0,25	7,38	14,34	

kontynuacja na następnej stronie

Problem			CPU[s]						
typ	m	n	M1 <sub>1</sub>	M1 <sub>2</sub>	M1 <sub>3</sub>	M1 <sub>4</sub>	M1 <sub>5</sub>	M2 <sub>1</sub>	M2 <sub>2</sub>
TC6	12	3	1,52	2,10	0,72	1,26	2,08	x	x
		4	1,19	1,78	0,52	1,07	1,81	x	x
		6	0,57	0,83	0,20	0,57	0,82	x	x
		7	0,44	0,59	0,15	0,41	0,63	x	x
	15	4	13,48	18,91	4,40	9,46	16,50	x	x
		5	6,09	9,09	2,45	5,03	7,49	x	x
		8	1,10	2,19	0,46	1,09	1,75	x	x
		9	0,77	1,36	0,19	0,70	1,07	x	x
	8	2	0,15	0,21	0,28	0,13	0,22	4,14	0,92
		3	0,13	0,18	0,20	0,13	0,20	5,17	2,19
		4	0,09	0,13	0,14	0,09	0,17	6,13	2,46
		5	0,04	0,06	0,06	0,05	0,10	4,25	1,59
	10	3	0,57	0,86	0,94	0,50	0,77	76,66	16,76
		4	0,44	0,70	0,84	0,43	0,65	128,22	23,27
		5	0,19	0,32	0,51	0,33	0,59	170,81	27,52
		6	0,18	0,27	0,39	0,27	0,52	148,19	31,28
	12	3	2,24	3,46	4,31	1,46	2,52	x	x
		4	2,41	2,86	3,24	1,51	2,17	x	x
		6	0,67	1,51	1,72	1,10	2,01	x	x
		7	0,45	0,72	1,13	0,70	1,90	x	x
15	4	25,15	33,34	39,67	13,96	22,79	x	x	
	5	15,62	23,96	31,62	9,58	16,27	x	x	
	8	2,87	4,53	5,90	3,45	6,92	x	x	
	9	2,55	4,13	3,98	3,92	6,68	x	x	
TC7	8	2	0,18	0,23	0,10	0,16	0,20	0,53	1,90
		3	0,21	0,23	0,09	0,21	0,22	0,92	3,23
		4	0,14	0,17	0,05	0,13	0,18	0,45	3,46
		5	0,10	0,11	0,03	0,10	0,12	0,32	2,90
	10	3	0,69	0,85	0,24	0,55	0,72	5,71	24,74
		4	0,58	0,61	0,18	0,56	0,63	4,65	38,54
		5	0,46	0,56	0,13	0,39	0,53	6,52	43,25
		6	0,26	0,38	0,07	0,27	0,34	6,11	33,47
	12	3	3,82	4,19	0,72	1,71	2,37	x	x
		4	2,61	2,48	0,47	1,71	1,95	x	x
		6	0,81	1,11	0,21	0,75	1,00	x	x
		7	0,60	0,83	0,15	0,53	0,68	x	x
15	4	43,39	29,13	3,31	17,76	24,90	x	x	
	5	12,59	15,15	1,62	9,28	10,85	x	x	
	8	1,62	2,28	0,34	1,40	2,11	x	x	
	9	1,20	1,75	0,21	0,98	1,33	x	x	
TC8	8	2	0,07	0,12	0,10	0,07	0,15	0,95	0,40
		3	0,08	0,10	0,08	0,08	0,14	1,56	0,46
		4	0,07	0,10	0,06	0,06	0,11	1,21	0,54
		5	0,04	0,05	0,03	0,04	0,07	0,55	0,51

kontynuacja na następnej stronie

Problem			CPU[s]							
typ	m	n	M1 <sub>1</sub>	M1 <sub>2</sub>	M1 <sub>3</sub>	M1 <sub>4</sub>	M1 <sub>5</sub>	M2 <sub>1</sub>	M2 <sub>2</sub>	
TC9	10	3	0,23	0,37	0,28	0,25	0,46	17,16	5,82	
		4	0,18	0,27	0,15	0,19	0,34	9,69	6,00	
		5	0,18	0,24	0,12	0,14	0,30	16,14	5,28	
		6	0,14	0,18	0,09	0,12	0,21	11,59	4,31	
	12	3	0,70	1,06	0,65	0,64	1,31	x	x	
		4	0,55	0,96	0,50	0,64	1,14	x	x	
		6	0,25	0,54	0,23	0,29	0,63	x	x	
		7	0,25	0,35	0,17	0,25	0,49	x	x	
	15	4	2,56	5,95	2,55	2,76	5,72	x	x	
		5	2,02	3,87	2,03	2,03	3,82	x	x	
		8	0,70	1,37	0,68	0,59	1,59	x	x	
		9	0,41	0,82	0,20	0,41	0,90	x	x	
	8	2	0,33	0,30	0,11	0,26	0,23	0,87	4,19	
		3	0,33	0,32	0,09	0,26	0,18	1,11	5,60	
		4	0,23	0,22	0,07	0,20	0,13	0,82	5,95	
		5	0,16	0,14	0,04	0,16	0,10	0,84	4,79	
		10	3	1,27	1,03	0,28	0,94	0,67	7,71	82,97
			4	1,01	0,94	0,23	0,81	0,54	8,84	133,78
			5	0,70	0,69	0,17	0,51	0,41	12,63	117,79
			6	0,42	0,42	0,12	0,44	0,28	17,03	<sup>2</sup> 140,23
		12	3	10,62	7,04	0,96	2,59	2,16	x	x
			4	6,05	4,90	0,66	2,62	1,99	x	x
			6	1,44	1,47	0,38	1,31	0,83	x	x
			7	0,95	0,92	0,20	0,80	0,63	x	x
15	4	<sup>2</sup> 183,27	64,22	4,59	27,41	16,77	x	x		
	5	42,77	26,02	3,51	16,57	11,36	x	x		
	8	3,59	3,24	0,70	2,38	1,77	x	x		
	9	2,02	2,29	0,41	1,51	1,27	x	x		
TC10	8	2	0,54	0,46	0,30	0,35	0,24	1,64	3,90	
		3	0,47	0,40	0,22	0,39	0,19	2,06	6,70	
		4	0,29	0,29	0,11	0,26	0,10	1,74	7,64	
		5	0,17	0,17	0,04	0,17	0,04	0,51	5,71	
	10	3	24,97	4,31	1,81	1,74	0,91	42,92	69,82	
		4	5,49	2,32	0,79	1,56	0,63	92,17	113,03	
		5	1,72	1,38	0,46	0,85	0,33	61,58	148,37	
		6	0,64	0,58	0,16	0,49	0,16	33,44	<sup>2</sup> 203,23	
	12	3	<sup>5</sup> 303,48	30,62	11,79	18,84	8,83	x	x	
		4	121,74	24,36	10,95	14,66	7,11	x	x	
		6	6,70	5,33	1,77	3,91	1,26	x	x	
		7	1,63	1,70	0,54	1,52	0,49	x	x	
15	4	<sup>14</sup> 583,42	<sup>3</sup> 353,9	156,83	<sup>1</sup> 213,24	112,38	x	x		
	5	<sup>10</sup> 465,72	<sup>1</sup> 244,66	101,02	<sup>1</sup> 154,77	62,39	x	x		
	8	29,43	25,71	11,19	15,21	6,41	x	x		
	9	11,15	7,47	3,05	6,47	2,59	x	x		

kontynuacja na następnej stronie

Problem			CPU[s]							
typ	m	n	M1 <sub>1</sub>	M1 <sub>2</sub>	M1 <sub>3</sub>	M1 <sub>4</sub>	M1 <sub>5</sub>	M2 <sub>1</sub>	M2 <sub>2</sub>	
TC11	8	2	0,35	0,34	0,09	0,26	0,13	0,54	3,87	
		3	0,32	0,32	0,08	0,25	0,10	0,62	5,29	
		4	0,23	0,24	0,06	0,21	0,07	0,45	5,76	
		5	0,17	0,15	0,04	0,15	0,05	0,45	4,84	
	10	3	1,49	1,13	0,25	0,91	0,35	4,42	78,21	
		4	1,22	1,00	0,20	0,77	0,27	4,25	135,66	
		5	0,73	0,71	0,15	0,56	0,19	5,80	<sup>1</sup> 152,3	
	6	6	0,45	0,45	0,09	0,40	0,13	7,12	<sup>2</sup> 173,24	
		12	3	12,41	8,80	0,72	2,74	1,04	x	x
	4	4	6,60	5,23	0,52	2,66	0,83	x	x	
		6	1,53	1,47	0,32	1,16	0,43	x	x	
		7	0,90	0,99	0,17	0,75	0,28	x	x	
	15	4	<sup>2</sup> 202,64	63,93	3,14	28,74	4,60	x	x	
		5	51,34	29,14	2,54	18,07	2,26	x	x	
		8	2,97	3,42	0,59	2,19	0,72	x	x	
		9	2,14	2,20	0,38	1,56	0,47	x	x	
	TC12	8	2	0,44	0,40	0,12	0,28	0,10	0,32	3,99
			3	0,35	0,36	0,08	0,31	0,07	0,37	5,75
4			0,26	0,25	0,04	0,22	0,03	0,17	7,71	
5			0,17	0,15	0,03	0,16	0,02	0,13	7,00	
10		3	5,14	2,09	0,42	1,27	0,28	2,48	66,42	
		4	3,63	1,58	0,19	1,00	0,16	2,38	137,57	
		5	1,05	0,79	0,10	0,73	0,10	1,87	199,40	
6		6	0,51	0,45	0,05	0,41	0,05	0,65	<sup>2</sup> 256,87	
		12	3	69,09	19,77	1,47	10,81	0,97	x	x
4		4	23,17	11,09	0,71	7,50	0,51	x	x	
		6	2,35	2,12	0,20	1,60	0,19	x	x	
		7	1,13	1,25	0,10	0,91	0,12	x	x	
15		4	<sup>8</sup> 396,14	223,48	5,59	120,54	2,90	x	x	
		5	<sup>1</sup> 146,68	<sup>1</sup> 96,21	2,61	47,09	1,40	x	x	
		8	5,38	5,59	0,27	3,51	0,27	x	x	
	9	2,71	2,92	0,15	1,90	0,17	x	x		

## A.2. Wyniki modelu PL agregacji OWA

Tabela A.2: Statystyki czasowe modelu PL agregacji OWA

Problem			CPU[s]				
typ	m	n	MLP <sub>1</sub>	MLP <sub>2</sub>	MLP <sub>3</sub>	MLP <sub>4</sub>	MLP <sub>5</sub>
TC1	8	2	0,00	0,01	0,00	0,01	0,01
		3	0,01	0,00	0,00	0,01	0,00
		4	0,00	0,00	0,00	0,00	0,00

kontynuacja na następnej stronie

Problem			CPU[s]				
typ	m	n	MLP <sub>1</sub>	MLP <sub>2</sub>	MLP <sub>3</sub>	MLP <sub>4</sub>	MLP <sub>5</sub>
		5	0,01	0,00	0,00	0,00	0,00
	10	3	0,01	0,01	0,01	0,01	0,01
		4	0,00	0,01	0,00	0,01	0,01
		5	0,00	0,00	0,01	0,00	0,01
		6	0,01	0,00	0,00	0,01	0,00
	12	3	0,01	0,01	0,01	0,01	0,01
		4	0,01	0,01	0,00	0,01	0,01
		6	0,00	0,01	0,01	0,01	0,01
		7	0,00	0,01	0,01	0,00	0,00
	15	4	0,02	0,02	0,02	0,02	0,02
		5	0,01	0,01	0,01	0,01	0,01
		8	0,01	0,01	0,01	0,01	0,01
		9	0,01	0,01	0,01	0,01	0,01
	20	5	0,02	0,02	0,03	0,02	0,03
		7	0,01	0,01	0,01	0,01	0,02
		10	0,01	0,01	0,01	0,01	0,01
		11	0,01	0,01	0,01	0,01	0,01
	25	7	0,02	0,02	0,02	0,03	0,03
		9	0,02	0,02	0,02	0,02	0,02
		13	0,01	0,01	0,01	0,01	0,02
		14	0,01	0,01	0,01	0,01	0,02
	30	8	0,04	0,04	0,04	0,04	0,05
		10	0,02	0,02	0,02	0,02	0,03
		15	0,02	0,01	0,02	0,01	0,02
		16	0,02	0,02	0,01	0,01	0,02
TC2	8	2	0,02	0,02	0,02	0,02	0,02
		3	0,02	0,02	0,02	0,02	0,02
		4	0,01	0,01	0,02	0,01	0,01
		5	0,01	0,01	0,01	0,01	0,01
	10	3	0,03	0,03	0,03	0,03	0,03
		4	0,03	0,04	0,03	0,03	0,03
		5	0,02	0,02	0,02	0,02	0,02
		6	0,01	0,02	0,01	0,02	0,01
	12	3	0,06	0,06	0,06	0,06	0,05
		4	0,05	0,05	0,05	0,05	0,05
		6	0,03	0,04	0,04	0,04	0,04
		7	0,02	0,02	0,03	0,03	0,03
	15	4	0,11	0,11	0,11	0,11	0,10
		5	0,11	0,11	0,11	0,11	0,11
		8	0,06	0,06	0,06	0,06	0,06
		9	0,04	0,04	0,04	0,05	0,05
	20	5	0,31	0,31	0,31	0,31	0,32
		7	0,28	0,29	0,29	0,29	0,30
		10	0,21	0,22	0,22	0,22	0,23

kontynuacja na następnej stronie

Problem			CPU[s]					
typ	m	n	MLP <sub>1</sub>	MLP <sub>2</sub>	MLP <sub>3</sub>	MLP <sub>4</sub>	MLP <sub>5</sub>	
TC3		11	0,18	0,19	0,19	0,18	0,18	
		25	0,73	0,73	0,73	0,73	0,71	
			9	0,65	0,65	0,64	0,65	0,66
			13	0,57	0,56	0,56	0,56	0,49
			14	0,44	0,44	0,44	0,44	0,39
		30	8	2,21	2,21	2,19	2,21	1,96
			10	1,44	1,47	1,47	1,47	1,09
			15	0,85	0,85	0,85	0,85	0,93
			16	0,63	0,63	0,63	0,63	0,76
		8	2	0,02	0,02	0,02	0,02	0,02
			3	0,02	0,02	0,02	0,02	0,02
			4	0,02	0,02	0,02	0,02	0,02
			5	0,01	0,01	0,02	0,02	0,01
		10	3	0,03	0,04	0,03	0,03	0,03
			4	0,03	0,04	0,03	0,03	0,03
			5	0,03	0,04	0,03	0,03	0,03
			6	0,03	0,03	0,03	0,03	0,03
		12	3	0,05	0,05	0,05	0,05	0,05
			4	0,06	0,06	0,06	0,05	0,05
			6	0,06	0,06	0,06	0,06	0,06
			7	0,05	0,05	0,05	0,05	0,05
		15	4	0,10	0,10	0,10	0,11	0,10
			5	0,12	0,12	0,12	0,12	0,11
			8	0,10	0,10	0,09	0,10	0,09
			9	0,07	0,07	0,07	0,08	0,07
	20	5	0,31	0,32	0,32	0,32	0,28	
		7	0,36	0,37	0,37	0,37	0,38	
		10	0,34	0,35	0,35	0,35	0,34	
		11	0,28	0,28	0,29	0,28	0,27	
	25	7	0,70	0,69	0,69	0,69	0,70	
		9	0,92	0,92	0,93	0,92	0,88	
		13	0,64	0,63	0,63	0,63	0,55	
		14	0,49	0,49	0,49	0,49	0,52	
	30	8	1,43	1,43	1,45	1,43	1,45	
		10	1,87	1,87	1,87	1,88	1,79	
		15	0,85	0,85	0,86	0,86	0,83	
		16	0,86	0,85	0,86	0,86	0,77	
TC9	8	2	0,02	0,02	0,03	0,02	0,02	
		3	0,02	0,02	0,02	0,02	0,02	
		4	0,02	0,02	0,02	0,02	0,02	
		5	0,01	0,01	0,01	0,02	0,02	
	10	3	0,04	0,04	0,05	0,04	0,04	
		4	0,04	0,03	0,04	0,04	0,04	
		5	0,03	0,03	0,03	0,03	0,04	

kontynuacja na następnej stronie

Problem			CPU[s]				
typ	m	n	MLP <sub>1</sub>	MLP <sub>2</sub>	MLP <sub>3</sub>	MLP <sub>4</sub>	MLP <sub>5</sub>
		6	0,02	0,02	0,03	0,03	0,03
	12	3	0,07	0,07	0,08	0,08	0,07
		4	0,06	0,07	0,07	0,08	0,06
		6	0,05	0,05	0,06	0,06	0,06
		7	0,04	0,04	0,05	0,05	0,05
	15	4	0,13	0,13	0,16	0,15	0,14
		5	0,12	0,12	0,14	0,14	0,13
		8	0,07	0,08	0,09	0,09	0,09
		9	0,06	0,06	0,07	0,07	0,07
	20	5	0,32	0,33	0,40	0,39	0,34
		7	0,32	0,34	0,42	0,39	0,34
		10	0,25	0,27	0,35	0,34	0,27
		11	0,22	0,23	0,30	0,29	0,24
	25	7	0,75	0,77	1,00	0,92	0,72
		9	0,77	0,80	1,01	0,97	0,80
		13	0,55	0,55	0,72	0,77	0,57
		14	0,44	0,45	0,60	0,65	0,48
	30	8	2,12	2,03	2,63	2,53	2,19
		10	2,10	2,18	2,80	2,73	2,14
		15	1,28	1,31	1,77	1,91	1,32
		16	1,17	1,18	1,67	1,80	1,24
TC11	8	2	0,02	0,02	0,02	0,02	0,02
		3	0,02	0,02	0,02	0,02	0,02
		4	0,01	0,01	0,02	0,02	0,02
		5	0,01	0,01	0,01	0,01	0,01
	10	3	0,03	0,03	0,04	0,03	0,03
		4	0,03	0,02	0,03	0,03	0,03
		5	0,03	0,03	0,03	0,03	0,03
		6	0,02	0,02	0,02	0,02	0,02
	12	3	0,05	0,05	0,06	0,06	0,06
		4	0,05	0,05	0,06	0,05	0,05
		6	0,03	0,04	0,05	0,05	0,04
		7	0,03	0,03	0,03	0,04	0,04
	15	4	0,10	0,11	0,13	0,12	0,12
		5	0,09	0,10	0,11	0,11	0,10
		8	0,05	0,05	0,06	0,06	0,06
		9	0,04	0,04	0,05	0,05	0,05
	20	5	0,23	0,23	0,28	0,26	0,25
		7	0,22	0,23	0,28	0,26	0,23
		10	0,18	0,18	0,22	0,22	0,19
		11	0,12	0,13	0,17	0,16	0,14
	25	7	0,54	0,54	0,74	0,64	0,60
		9	0,49	0,50	0,69	0,62	0,53
		13	0,31	0,32	0,44	0,43	0,36

kontynuacja na następnej stronie

Problem			CPU[s]				
typ	m	n	MLP <sub>1</sub>	MLP <sub>2</sub>	MLP <sub>3</sub>	MLP <sub>4</sub>	MLP <sub>5</sub>
		14	0,26	0,25	0,38	0,37	0,31
	30	8	1,27	1,39	1,78	1,56	1,35
		10	1,29	1,27	1,79	1,64	1,34
		15	0,67	0,71	1,01	1,11	0,76
		16	0,63	0,60	0,89	0,90	0,68

### A.3. Wyniki modeli PCLM M3 i M4 agregacji OWA

Tabela A.3: Statystyki czasowe modeli PCLM M3 i M4 agregacji OWA (górny indeks przed czasem oznacza liczbę instancji spośród 15, które przekroczyły ustalony limit czasu 600 s; przypadki, gdy wszystkie 15 instancji przekroczyło limit czasu, oznaczone są przez „-”)

Problem			CPU[s]						
typ	m	n	M3 <sub>1</sub>	M3 <sub>2</sub>	M3 <sub>3</sub>	M3 <sub>4</sub>	M3 <sub>5</sub>	M4 <sub>1</sub>	M4 <sub>2</sub>
TC4	8	2	0,05	0,05	0,06	0,05	0,06	0,08	0,08
		3	0,05	0,05	0,07	0,05	0,07	0,07	0,06
		4	0,05	0,05	0,06	0,05	0,06	0,06	0,06
	10	5	0,05	0,04	0,05	0,05	0,05	0,05	0,05
		3	0,08	0,09	0,15	0,09	0,15	0,15	0,15
		4	0,09	0,09	0,14	0,09	0,14	0,13	0,13
TC5	8	5	0,11	0,11	0,11	0,11	0,12	0,11	0,11
		6	0,10	0,10	0,10	0,10	0,10	0,09	0,10
		2	52,48	51,39	10,22	3,97	2,69	10,98	2,62
	10	3	53,16	54,28	7,02	3,63	2,39	7,58	2,10
		4	50,14	53,76	5,06	3,61	1,48	5,74	1,38
		5	52,74	52,91	4,68	4,26	1,23	4,91	1,20
TC6	8	3	-	-	-	167,56	103,15	-	108,89
		4	-	-	-	160,29	83,34	-	96,73
		5	-	-	-	157,81	72,70	-	82,83
	10	6	-	-	-	152,06	56,88	-	58,92
		2	148,20	151,27	18,73	6,36	2,53	19,88	2,73
		3	177,86	189,57	14,01	6,42	2,02	14,05	1,88
TC7	8	4	170,15	199,93	11,29	6,65	2,25	11,41	2,17
		5	163,80	159,00	8,66	6,44	1,32	7,13	1,30
		3	-	-	-	294,08	91,91	-	96,32
	10	4	-	-	-	321,08	84,96	-	90,87
		5	-	-	-	313,26	72,61	-	74,96
		6	-	-	-	297,26	68,36	-	70,64

kontynuacja na następnej stronie



Problem			CPU[s]						
typ	m	n	M3 <sub>1</sub>	M3 <sub>2</sub>	M3 <sub>3</sub>	M3 <sub>4</sub>	M3 <sub>5</sub>	M4 <sub>1</sub>	M4 <sub>2</sub>
TC8	10	3	<sup>1</sup> 346,19	<sup>2</sup> 393,42	12,78	22,12	6,25	10,21	6,11
		4	<sup>1</sup> 327,38	<sup>1</sup> 364,44	6,66	20,24	4,35	6,34	4,61
		5	310,17	260,00	5,54	23,00	3,08	4,32	3,49
		6	207,92	226,05	2,03	19,51	1,53	1,93	1,63
	8	2	2,57	2,85	0,83	0,80	0,65	0,76	0,62
		3	2,33	2,30	0,93	0,69	0,58	0,89	0,59
		4	2,17	2,40	0,64	0,63	0,49	0,62	0,45
TC10	10	3	<sup>3</sup> 416,97	<sup>2</sup> 416,28	45,31	25,34	11,81	43,93	12,12
		4	<sup>1</sup> 287,87	311,06	36,27	24,31	9,77	31,67	9,37
		5	291,73	289,73	34,71	25,30	9,44	30,62	9,10
		6	293,54	310,41	26,53	24,76	8,06	27,82	7,68
	8	2	–	–	0,28	8,88	0,22	0,26	0,21
		3	–	–	0,18	14,44	0,15	0,16	0,14
		4	–	–	0,11	19,69	0,09	0,09	0,09
		5	–	–	0,06	24,00	0,05	0,05	0,04
		6	–	–	0,06	24,00	0,05	0,05	0,04
	TC12	10	3	–	–	2,68	<sup>4</sup> 490,57	0,68	3,36
4			–	–	0,65	<sup>14</sup> 593,68	0,40	0,54	0,35
5			–	–	0,30	–	0,21	0,26	0,19
6			–	–	0,15	–	0,13	0,13	0,11
8		2	<sup>12</sup> 536,72	<sup>11</sup> 528,55	0,13	3,47	0,11	0,12	0,10
		3	–	–	0,09	7,50	0,07	0,06	0,08
		4	–	–	0,05	11,19	0,05	0,05	0,05
		5	–	–	0,03	15,50	0,03	0,03	0,03
10	3	–	–	0,38	195,78	0,32	0,52	0,27	
	4	–	–	0,18	<sup>1</sup> 361,19	0,17	0,16	0,14	
	5	–	–	0,09	<sup>7</sup> 518,86	0,09	0,09	0,09	
	6	–	–	0,07	<sup>14</sup> 589,66	0,06	0,06	0,06	

## B. Wyniki modeli PCLM agregacji WOWA

### B.1. Wyniki modelu PL agregacji WOWA

Tabela B.1: Statystyki czasowe modelu PL agregacji WOWA ze zróżnicowanymi wagami zapotrzebowania (wyniki sformułowania podstawowego MWLP oraz sformułowań z dodanymi ograniczeniami nadmiarowymi)

Problem			CPU[s]			
typ	m	n	MWLP	c <sub>3</sub>	c <sub>4</sub>	c <sub>3</sub> c <sub>4</sub>
TC1	8	2	0,01	0,01	0,01	0,01
		3	0,00	0,01	0,00	0,00
		4	0,00	0,00	0,00	0,00
		5	0,00	0,00	0,00	0,01
	10	3	0,00	0,01	0,00	0,01
		4	0,01	0,01	0,00	0,01
		5	0,00	0,01	0,00	0,01
		6	0,00	0,01	0,00	0,01
	12	3	0,01	0,01	0,01	0,02
		4	0,01	0,01	0,01	0,01
		6	0,00	0,01	0,01	0,01
		7	0,01	0,01	0,01	0,01
	15	4	0,01	0,02	0,01	0,04
		5	0,01	0,01	0,01	0,03
		8	0,01	0,01	0,01	0,02
		9	0,01	0,01	0,01	0,02
	20	5	0,02	0,06	0,03	0,10
		7	0,01	0,03	0,02	0,06
		10	0,01	0,02	0,01	0,05
		11	0,01	0,01	0,01	0,04
25	7	0,02	0,05	0,03	0,14	
	9	0,02	0,04	0,02	0,13	
	13	0,01	0,02	0,01	0,08	
	14	0,02	0,02	0,01	0,08	
30	8	0,04	0,12	0,05	0,26	
	10	0,04	0,08	0,04	0,23	
	15	0,02	0,03	0,02	0,13	
	16	0,02	0,03	0,02	0,13	

kontynuacja na następnej stronie

Problem			CPU[s]				
typ	m	n	MWLP	c <sub>3</sub>	c <sub>4</sub>	c <sub>3</sub> c <sub>4</sub>	
TC2	8	2	0,02	0,04	0,02	0,04	
		3	0,02	0,03	0,02	0,03	
		4	0,02	0,03	0,02	0,03	
		5	0,02	0,02	0,02	0,03	
	10	3	0,03	0,07	0,04	0,08	
		4	0,03	0,05	0,04	0,06	
		5	0,02	0,05	0,03	0,06	
		6	0,02	0,04	0,03	0,05	
	12	3	0,05	0,11	0,06	0,12	
		4	0,05	0,10	0,06	0,11	
		6	0,05	0,09	0,06	0,09	
		7	0,04	0,09	0,06	0,09	
	15	4	0,11	0,24	0,09	0,23	
		5	0,11	0,20	0,07	0,22	
		8	0,09	0,17	0,05	0,15	
		9	0,07	0,15	0,05	0,14	
	20	5	0,35	0,65	0,24	0,66	
		7	0,36	0,51	0,19	0,52	
		10	0,30	0,40	0,13	0,44	
		11	0,29	0,46	0,14	0,41	
	25	7	0,83	1,54	0,60	1,67	
		9	0,80	1,31	0,48	1,39	
		13	0,68	1,04	0,29	0,98	
		14	0,56	0,84	0,23	0,89	
	30	8	1,81	4,00	1,60	4,55	
		10	1,78	3,73	1,23	3,75	
		15	1,29	1,91	0,57	2,11	
		16	1,05	1,84	0,45	1,76	
	TC3	8	2	0,02	0,04	0,02	0,05
			3	0,02	0,04	0,02	0,04
4			0,02	0,04	0,02	0,04	
5			0,02	0,03	0,01	0,04	
10		3	0,03	0,07	0,03	0,08	
		4	0,03	0,09	0,03	0,08	
		5	0,03	0,06	0,03	0,07	
		6	0,01	0,04	0,02	0,05	
12		3	0,05	0,15	0,06	0,16	
		4	0,05	0,15	0,06	0,17	
		6	0,05	0,11	0,05	0,14	
		7	0,03	0,10	0,03	0,11	
15		4	0,09	0,27	0,07	0,27	
		5	0,08	0,25	0,06	0,26	
		8	0,04	0,18	0,04	0,19	
		9	0,03	0,13	0,03	0,13	

kontynuacja na następnej stronie

Problem			CPU[s]					
typ	m	n	MWLP	c <sub>3</sub>	c <sub>4</sub>	c <sub>3</sub> c <sub>4</sub>		
TC9	20	5	0,25	0,94	0,17	0,95		
		7	0,24	0,94	0,17	0,96		
		10	0,18	0,63	0,13	0,67		
		11	0,13	0,54	0,10	0,60		
	25	7	0,51	2,88	0,39	2,86		
		9	0,51	2,87	0,34	2,80		
		13	0,21	1,17	0,16	1,13		
		14	0,12	0,75	0,11	0,72		
	30	8	0,81	5,82	0,69	5,67		
		10	0,81	5,32	0,60	5,02		
		15	0,25	1,98	0,26	1,97		
		16	0,21	1,92	0,19	1,85		
	8	2	2	0,02	0,02	0,02	0,02	
			3	0,01	0,01	0,01	0,01	
			4	0,01	0,01	0,01	0,01	
			5	0,01	0,01	0,01	0,01	
			10	0,02	0,03	0,03	0,03	
		4	3	0,02	0,02	0,02	0,02	
			5	0,01	0,02	0,02	0,02	
			6	0,01	0,01	0,01	0,01	
			12	3	0,06	0,07	0,06	0,07
			4	0,05	0,07	0,06	0,07	
	6	3	0,03	0,03	0,04	0,04		
		7	0,02	0,03	0,03	0,03		
		15	4	0,07	0,10	0,09	0,10	
		5	0,07	0,09	0,07	0,09		
	8	4	0,04	0,06	0,05	0,06		
		9	0,03	0,04	0,03	0,05		
		20	5	0,21	0,29	0,26	0,31	
		7	0,16	0,20	0,19	0,22		
		10	0,09	0,13	0,12	0,14		
	11	0,08	0,11	0,11	0,12			
25		7	0,46	0,62	0,55	0,67		
9		0,41	0,54	0,48	0,64			
13		0,12	0,20	0,18	0,24			
14		0,09	0,16	0,15	0,17			
30	8	0,73	0,97	0,84	1,05			
	10	0,56	0,84	0,74	0,95			
	15	0,14	0,25	0,24	0,32			
	16	0,12	0,22	0,22	0,28			
TC11	8	2	0,02	0,02	0,02	0,02		
		3	0,01	0,01	0,01	0,01		
		4	0,01	0,01	0,01	0,01		
		5	0,01	0,01	0,01	0,01		

kontynuacja na następnej stronie

Problem			CPU[s]			
typ	m	n	MWLP	c <sub>3</sub>	c <sub>4</sub>	c <sub>3</sub> c <sub>4</sub>
	10	3	0,02	0,02	0,02	0,03
		4	0,02	0,02	0,02	0,02
		5	0,01	0,01	0,01	0,02
		6	0,01	0,01	0,01	0,01
	12	3	0,04	0,05	0,05	0,06
		4	0,04	0,04	0,04	0,05
		6	0,02	0,03	0,03	0,03
		7	0,02	0,03	0,02	0,03
	15	4	0,06	0,08	0,07	0,08
		5	0,05	0,07	0,07	0,09
		8	0,03	0,05	0,04	0,06
		9	0,02	0,04	0,03	0,04
	20	5	0,18	0,22	0,20	0,25
		7	0,13	0,17	0,16	0,20
		10	0,07	0,11	0,10	0,13
		11	0,06	0,09	0,09	0,12
	25	7	0,33	0,43	0,38	0,47
		9	0,32	0,42	0,38	0,48
		13	0,10	0,15	0,15	0,20
		14	0,08	0,13	0,13	0,16
	30	8	0,50	0,68	0,63	0,81
		10	0,50	0,67	0,59	0,81
		15	0,13	0,21	0,22	0,29
		16	0,11	0,20	0,20	0,27

## B.2. Wyniki modeli PCLM MW1 i MW2 agregacji WOWA

Tabela B.2: Statystyki czasowe modeli PCLM agregacji WOWA ze zróżnicowanymi wagami zapotrzebowania (górnym indeksem przed czasem to liczba instancji spośród 15, które przekroczyły ustalony limit czasu 600 s; przypadki, gdy wszystkie 15 instancji przekroczyło limit czasu, oznaczone są przez „-”)

Problem			CPU[s]			
typ	m	n	MW1 <sub>1</sub>	MW1 <sub>2</sub>	MW2 <sub>1</sub>	MW2 <sub>2</sub>
TC4	8	2	2,99	1,41	0,63	0,39
		3	18,01	6,29	1,45	0,83
		4	76,19	26,70	3,38	1,97
		5	<sup>10</sup> 516,71	<sup>6</sup> 364,93	11,80	6,64
	10	3	78,17	59,77	8,54	4,12
		4	<sup>2</sup> 141,83	<sup>1</sup> 156,31	24,45	8,07
		5	<sup>14</sup> 594,06	<sup>11</sup> 549,54	131,12	26,22
		6	-	-	<sup>6</sup> 431,12	<sup>1</sup> 176,46

kontynuacja na następnej stronie

Problem			CPU[s]					
typ	m	n	MW1 <sub>1</sub>	MW1 <sub>2</sub>	MW2 <sub>1</sub>	MW2 <sub>2</sub>		
TC5	8	2	5,08	4,01	1,25	0,84		
		3	5,50	4,56	1,24	0,86		
		4	5,48	3,84	1,38	0,86		
		5	4,37	4,25	0,92	0,75		
		10	3	141,67	95,26	23,90	19,01	
	10	4	172,59	158,08	23,52	21,34		
		5	<sup>1</sup> 168,07	<sup>2</sup> 201,03	23,54	17,27		
		6	<sup>2</sup> 188,8	<sup>3</sup> 189,2	47,05	16,54		
		TC6	8	2	13,68	11,85	2,71	1,78
				3	36,35	23,83	4,45	3,66
4	178,59			63,66	7,18	6,38		
5	<sup>7</sup> 391,16			138,80	8,32	8,30		
10	3			<sup>12</sup> 594,46	<sup>10</sup> 544,71	101,00	63,31	
10	4	–	<sup>13</sup> 580,75	<sup>1</sup> 187,92	119,40			
	5	–	–	<sup>3</sup> 290,17	199,29			
	6	–	–	<sup>13</sup> 570,7	<sup>2</sup> 403,8			
	TC7	8	2	1,29	0,94	0,62	0,35	
3			0,88	0,80	0,45	0,30		
4			0,70	0,65	0,29	0,23		
5			0,48	0,44	0,21	0,18		
10			3	21,46	13,62	5,38	2,76	
10		4	42,25	11,66	4,59	2,04		
		5	18,36	9,71	2,91	1,95		
		6	6,17	5,90	2,69	2,23		
		TC8	8	2	6,01	2,60	1,06	0,79
				3	5,81	6,80	1,18	0,78
4	1,92			2,24	0,69	0,56		
5	3,45			2,56	0,61	0,65		
10	3			<sup>4</sup> 274,03	<sup>2</sup> 217,87	23,41	17,24	
10	4		<sup>4</sup> 248,97	<sup>4</sup> 249,46	25,13	16,98		
	5		<sup>3</sup> 206,18	<sup>3</sup> 194,49	20,74	18,27		
	6		<sup>3</sup> 224,4	<sup>1</sup> 121,01	19,66	18,30		
	TC10		8	2	1,19	0,75	0,50	0,28
				3	0,68	0,54	0,26	0,22
4		0,48		0,40	0,22	0,19		
5		0,27		0,27	0,17	0,14		
10		3		6,62	4,61	1,19	0,77	
10		4	4,84	3,25	1,24	0,78		
		5	2,90	1,77	0,78	0,58		
		6	1,54	1,08	0,49	0,37		
		TC12	8	2	0,51	0,44	0,35	0,21
				3	0,26	0,33	0,19	0,14
4	0,22			0,27	0,17	0,12		
5	0,15			0,19	0,11	0,10		

kontynuacja na następnej stronie

Problem			CPU[s]			
typ	m	n	MW1 <sub>1</sub>	MW1 <sub>2</sub>	MW2 <sub>1</sub>	MW2 <sub>2</sub>
	10	3	1,25	1,02	0,59	0,43
		4	0,87	0,86	0,44	0,35
		5	0,56	0,58	0,30	0,28
		6	0,34	0,50	0,20	0,24

Tabela B.3: Statystyki czasowe modeli PCLM agregacji WOWA z jednakowymi wagami zapotrzebowania (górnny indeks przed czasem to liczba instancji spośród 15, które przekroczyły ustalony limit czasu 600 s; przypadki, gdy wszystkie 15 instancji przekroczyło limit czasu, oznaczone są przez „-”)

Problem			CPU[s]					
typ	m	n	MW1 <sub>1</sub>	MW1 <sub>2</sub>	MW2 <sub>1</sub>	MW2 <sub>2</sub>		
TC4	8	2	9,26	4,51	1,12	0,88		
		3	24,03	17,42	2,41	1,48		
		4	<sup>11</sup> 508,67	<sup>4</sup> 339,21	21,64	5,85		
		5	-	-	<sup>2</sup> 319,41	58,89		
	10	3	<sup>8</sup> 415,32	<sup>4</sup> 352,73	49,94	13,66		
		4	<sup>13</sup> 571,53	<sup>10</sup> 492,69	119,42	23,03		
		5	-	-	<sup>9</sup> 450,72	141,19		
		6	-	-	-	-		
		TC5	8	2	26,04	21,21	8,77	5,90
				3	41,98	40,93	13,25	8,99
4	50,18			36,33	9,61	7,07		
5	126,42			86,67	15,39	10,51		
10	3			-	-	<sup>11</sup> 553,17	<sup>5</sup> 462,55	
	4	-	-	<sup>11</sup> 530,15	<sup>6</sup> 523,86			
	5	-	-	<sup>10</sup> 546,78	<sup>8</sup> 502,62			
	6	-	-	<sup>8</sup> 465,71	<sup>8</sup> 434,12			
TC6	8	2	136,96	78,75	21,25	18,34		
		3	222,45	114,64	26,05	19,71		
		4	<sup>3</sup> 360,05	<sup>2</sup> 251,39	34,77	28,04		
		5	<sup>3</sup> 296,89	<sup>1</sup> 198,74	28,14	22,21		
		10	3	-	-	-	-	
	4		-	-	<sup>14</sup> 594,28	<sup>14</sup> 594,17		
	5		-	-	-	-		
	6		-	-	-	-		
	TC7		8	2	7,24	2,77	1,22	1,00
		3		10,30	3,74	1,05	1,05	
4		7,56		4,02	0,95	0,84		
5		7,36		2,61	0,63	0,62		
10		3		<sup>11</sup> 573,17	<sup>2</sup> 334,06	172,71	95,74	
		4	<sup>12</sup> 544,79	<sup>4</sup> 359,78	232,49	112,27		
		5	<sup>14</sup> 581,47	<sup>10</sup> 508,8	<sup>6</sup> 399,7	139,13		

kontynuacja na następnej stronie

Problem			CPU[s]				
typ	m	n	MW1 <sub>1</sub>	MW1 <sub>2</sub>	MW2 <sub>1</sub>	MW2 <sub>2</sub>	
TC8	8	6	<sup>14</sup> 586,54	<sup>10</sup> 520,65	<sup>4</sup> 303,92	179,74	
		2	24,89	15,24	6,25	3,40	
		3	52,70	27,82	9,76	5,85	
		4	67,45	32,48	11,34	7,10	
		5	77,93	34,21	9,60	3,98	
		10	3	–	–	<sup>14</sup> 598,19	<sup>11</sup> 565,79
TC10	10	4	–	–	<sup>12</sup> 555,31	<sup>10</sup> 518,69	
		5	–	–	–	<sup>12</sup> 559,32	
		6	–	–	<sup>14</sup> 594,6	<sup>12</sup> 548,61	
		8	2	3,02	2,00	0,91	0,65
		3	2,94	1,61	0,75	0,67	
	4	3,57	1,45	0,77	0,47		
	5	1,69	0,93	0,39	0,30		
	10	3	45,72	21,31	14,18	7,45	
	4	45,81	25,00	15,22	7,95		
	5	<sup>1</sup> 84,65	25,81	14,18	7,67		
TC12	8	6	33,76	24,30	9,43	4,07	
		2	1,06	0,65	0,63	0,23	
		3	0,82	0,46	0,42	0,22	
		4	0,51	0,29	0,30	0,18	
	5	0,25	0,25	0,17	0,15		
	10	3	8,89	5,32	2,23	1,20	
	4	5,39	3,35	1,61	0,99		
	5	2,42	1,46	0,74	0,61		
6	0,88	0,98	0,39	0,48			

Tabela B.4: Statystyki czasowe modelu PCLM MW<sub>2</sub> agregacji WOVA z nadmiarowymi ograniczeniami ze zróżnicowanymi wagami zapotrzebowania (górnny indeks przed czasem oznacza liczbę instancji spośród 15, które przekroczyły ustalony limit czasu 600 s; przypadki, gdy wszystkie 15 instancji przekroczyło limit czasu, oznaczone są przez „–” )

Problem			CPU[s]							
typ	m	n	MW2 <sub>2</sub>	c <sub>1</sub>	c <sub>2</sub>	c <sub>1</sub> c <sub>2</sub>	c <sub>3</sub>	c <sub>4</sub>	c <sub>3</sub> c <sub>4</sub>	
TC4	8	2	0,39	0,43	0,38	0,43	0,63	0,37	0,72	
		3	0,83	0,83	0,85	0,90	1,53	0,89	1,78	
		4	1,97	1,61	1,51	1,24	4,07	2,02	3,46	
		5	6,64	4,67	4,22	1,87	5,28	7,16	4,89	
		10	3	4,12	4,47	3,94	3,49	8,77	4,10	10,61
		4	8,07	9,58	10,21	9,13	16,45	7,47	19,38	
TC5	8	5	26,22	24,26	22,19	14,17	31,81	32,67	47,29	
		6	<sup>1</sup> 176,46	77,86	82,20	32,88	108,25	<sup>1</sup> 195,52	113,34	
		2	0,84	0,84	0,87	0,90	0,30	0,90	0,33	
		3	0,86	1,04	0,99	0,97	0,30	0,87	0,31	

kontynuacja na następnej stronie



Problem			CPU[s]						
typ	m	n	MW <sub>2</sub>	c <sub>1</sub>	c <sub>2</sub>	c <sub>1</sub> c <sub>2</sub>	c <sub>3</sub>	c <sub>4</sub>	c <sub>3</sub> c <sub>4</sub>
TC6	10	4	0,86	0,97	0,98	0,94	0,43	0,91	0,36
		5	0,75	0,89	0,81	0,71	0,29	0,78	0,19
		3	19,01	18,99	17,92	18,46	1,28	18,43	1,42
		4	21,34	21,03	17,61	21,93	1,74	24,67	1,47
		5	17,27	20,61	21,38	19,18	1,91	18,34	1,78
	6	16,54	20,83	21,27	20,92	1,31	23,20	1,19	
	8	2	1,78	1,64	1,59	1,63	0,83	1,85	1,20
	3	3,66	2,37	2,57	2,02	1,95	3,12	2,05	
	4	6,38	4,39	5,14	2,47	2,84	5,92	3,45	
	5	8,30	4,99	4,85	2,60	3,17	8,63	3,05	
TC7	10	3	63,31	55,66	51,80	42,82	15,67	75,97	16,24
		4	119,40	69,93	80,91	50,29	27,57	121,48	38,25
		5	199,29	111,63	119,99	67,19	66,10	205,66	89,80
	6	<sup>2</sup> 403,8	199,69	200,33	93,59	150,59	<sup>6</sup> 472,57	215,57	
	8	2	0,35	0,35	0,34	0,39	0,23	0,33	0,26
	3	0,30	0,30	0,30	0,36	0,17	0,30	0,17	
4	0,23	0,30	0,27	0,27	0,14	0,25	0,16		
5	0,18	0,20	0,21	0,25	0,13	0,18	0,14		
TC8	10	3	2,76	3,37	2,98	3,32	0,69	2,56	0,73
		4	2,04	2,70	2,51	2,59	0,53	2,47	0,58
		5	1,95	2,11	1,91	2,27	0,46	1,98	0,42
	6	2,23	1,91	1,86	1,84	0,35	1,88	0,34	
	8	2	0,79	0,67	0,69	0,71	0,47	0,72	0,54
	3	0,78	0,75	0,75	0,69	0,32	0,73	0,33	
4	0,56	0,64	0,57	0,56	0,19	0,68	0,21		
5	0,65	0,62	0,60	0,60	0,22	0,63	0,24		
TC10	10	3	17,24	20,07	20,17	20,59	6,73	20,32	7,08
		4	16,98	19,22	17,00	17,79	4,91	16,83	5,61
		5	18,27	16,24	16,42	18,08	3,17	16,85	2,94
	6	18,30	17,96	18,92	20,03	1,61	20,33	2,17	
	8	2	0,28	0,28	0,28	0,31	0,30	0,29	0,32
	3	0,22	0,22	0,22	0,26	0,21	0,23	0,23	
4	0,19	0,21	0,21	0,25	0,18	0,19	0,19		
5	0,14	0,16	0,16	0,18	0,14	0,14	0,14		
TC12	10	3	0,77	0,98	1,01	1,01	0,86	0,88	0,88
		4	0,78	0,94	0,93	0,97	0,80	0,78	0,84
		5	0,58	0,66	0,67	0,73	0,58	0,56	0,57
	6	0,37	0,38	0,42	0,57	0,36	0,34	0,36	
	8	2	0,21	0,23	0,24	0,26	0,21	0,21	0,21
	3	0,14	0,19	0,15	0,19	0,17	0,16	0,17	
4	0,12	0,15	0,17	0,16	0,12	0,13	0,13		
5	0,10	0,13	0,11	0,14	0,10	0,12	0,12		
10	3	0,43	0,48	0,49	0,56	0,41	0,43	0,45	
	4	0,35	0,41	0,47	0,52	0,35	0,37	0,38	

kontynuacja na następnej stronie

Problem			CPU[s]						
typ	m	n	MW2 <sub>2</sub>	c <sub>1</sub>	c <sub>2</sub>	c <sub>1</sub> c <sub>2</sub>	c <sub>3</sub>	c <sub>4</sub>	c <sub>3</sub> c <sub>4</sub>
		5	0,28	0,33	0,34	0,43	0,27	0,29	0,30
		6	0,24	0,29	0,23	0,32	0,19	0,24	0,25

## C. Wyniki metody VNS z agregacją OWA

### C.1. Wyniki metody VNS z agregacją OWA dla małych problemów

Tabela C.1: Statystyki czasowe metody VNS z agregacją OWA dla problemów o małym rozmiarze (od 8 do 30 lokalizacji)

Problem			inicjalizacja zachłanna					inicjalizacja losowa				
			CPU[s]					CPU[s]				
typ	m	n	VNS	VNS'	VNS''	VNS' <sub>r</sub>	VNS'' <sub>r</sub>	VNS	VNS'	VNS''	VNS' <sub>r</sub>	VNS'' <sub>r</sub>
TC1	8	2	0,0001	0,0001	0,0002	0,0001	0,0002	0,0001	0,0001	0,0003	0,0001	0,0003
		3	0,0001	0,0001	0,0003	0,0001	0,0003	0,0001	0,0001	0,0003	0,0001	0,0003
		4	0,0001	0,0001	0,0003	0,0001	0,0003	0,0001	0,0001	0,0003	0,0001	0,0003
		5	0,0002	0,0001	0,0003	0,0001	0,0003	0,0001	0,0001	0,0003	0,0001	0,0003
	10	3	0,0002	0,0002	0,0005	0,0002	0,0005	0,0002	0,0002	0,0005	0,0002	0,0005
		4	0,0003	0,0002	0,0006	0,0002	0,0006	0,0003	0,0002	0,0006	0,0002	0,0006
		5	0,0003	0,0002	0,0006	0,0003	0,0006	0,0003	0,0003	0,0006	0,0002	0,0006
		6	0,0003	0,0003	0,0006	0,0003	0,0006	0,0003	0,0003	0,0006	0,0003	0,0006
	12	3	0,0003	0,0002	0,0007	0,0002	0,0007	0,0003	0,0002	0,0008	0,0003	0,0008
		4	0,0004	0,0003	0,0008	0,0003	0,0008	0,0004	0,0003	0,0009	0,0003	0,0009
		6	0,0006	0,0004	0,0010	0,0004	0,0010	0,0006	0,0004	0,0010	0,0004	0,0010
		7	0,0007	0,0005	0,0010	0,0005	0,0010	0,0007	0,0005	0,0010	0,0005	0,0010
	15	4	0,0008	0,0005	0,0014	0,0005	0,0014	0,0008	0,0005	0,0014	0,0005	0,0014
		5	0,0010	0,0006	0,0015	0,0006	0,0015	0,0011	0,0006	0,0015	0,0006	0,0015
		8	0,0016	0,0009	0,0018	0,0009	0,0018	0,0016	0,0009	0,0017	0,0009	0,0017
		9	0,0016	0,0010	0,0018	0,0010	0,0018	0,0017	0,0009	0,0017	0,0009	0,0017
	20	5	0,0022	0,0011	0,0027	0,0011	0,0027	0,0026	0,0012	0,0030	0,0012	0,0030
		7	0,0035	0,0016	0,0033	0,0016	0,0033	0,0039	0,0017	0,0035	0,0017	0,0035
		10	0,0054	0,0023	0,0042	0,0024	0,0042	0,0056	0,0023	0,0041	0,0023	0,0041
		11	0,0058	0,0025	0,0043	0,0025	0,0043	0,0060	0,0025	0,0042	0,0025	0,0041
	25	7	0,0066	0,0027	0,0055	0,0027	0,0055	0,0079	0,0030	0,0060	0,0029	0,0060
		9	0,0092	0,0036	0,0066	0,0036	0,0066	0,0105	0,0038	0,0068	0,0038	0,0068
		13	0,0123	0,0048	0,0075	0,0048	0,0075	0,0141	0,0050	0,0078	0,0052	0,0078
		14	0,0129	0,0051	0,0078	0,0052	0,0078	0,0143	0,0053	0,0079	0,0054	0,0079
30	8	0,0141	0,0052	0,0099	0,0052	0,0099	0,0159	0,0055	0,0105	0,0055	0,0105	
	10	0,0173	0,0062	0,0106	0,0062	0,0106	0,0210	0,0070	0,0119	0,0070	0,0119	
	15	0,0274	0,0097	0,0139	0,0097	0,0138	0,0301	0,0099	0,0140	0,0101	0,0140	
	16	0,0268	0,0097	0,0135	0,0097	0,0134	0,0292	0,0098	0,0135	0,0100	0,0135	
TC2	8	2	0,0001	0,0001	0,0002	0,0001	0,0003	0,0001	0,0001	0,0002	0,0001	0,0003
		3	0,0001	0,0001	0,0002	0,0001	0,0003	0,0001	0,0001	0,0002	0,0001	0,0003
		4	0,0001	0,0001	0,0002	0,0001	0,0003	0,0001	0,0001	0,0002	0,0001	0,0003
		5	0,0001	0,0001	0,0002	0,0001	0,0003	0,0001	0,0001	0,0002	0,0001	0,0003
	10	3	0,0002	0,0001	0,0003	0,0002	0,0005	0,0002	0,0001	0,0003	0,0002	0,0005
		4	0,0003	0,0002	0,0003	0,0002	0,0006	0,0003	0,0001	0,0003	0,0002	0,0006
		5	0,0003	0,0002	0,0003	0,0003	0,0007	0,0003	0,0001	0,0003	0,0002	0,0006
		6	0,0003	0,0002	0,0004	0,0003	0,0006	0,0003	0,0002	0,0003	0,0003	0,0006
12	3	0,0003	0,0002	0,0004	0,0003	0,0008	0,0003	0,0001	0,0004	0,0002	0,0008	
	4	0,0005	0,0002	0,0005	0,0003	0,0010	0,0004	0,0002	0,0004	0,0003	0,0009	

kontynuacja na następnej stronie

			inicjalizacja zachłanna					inicjalizacja losowa					
Problem			CPU[s]					CPU[s]					
typ	m	n	VNS	VNS'	VNS''	VNS' <sub>r</sub>	VNS'' <sub>r</sub>	VNS	VNS'	VNS''	VNS' <sub>r</sub>	VNS'' <sub>r</sub>	
TC3	6	6	0,0007	0,0003	0,0005	0,0005	0,0010	0,0006	0,0002	0,0004	0,0004	0,0009	
		7	0,0007	0,0003	0,0006	0,0005	0,0010	0,0006	0,0003	0,0005	0,0004	0,0009	
	15	4	0,0008	0,0003	0,0007	0,0005	0,0014	0,0008	0,0003	0,0006	0,0005	0,0014	
		5	0,0010	0,0004	0,0008	0,0007	0,0017	0,0010	0,0003	0,0006	0,0006	0,0016	
		8	0,0015	0,0005	0,0009	0,0010	0,0019	0,0016	0,0005	0,0008	0,0009	0,0017	
	9	5	0,0016	0,0006	0,0010	0,0010	0,0019	0,0017	0,0005	0,0009	0,0009	0,0017	
		20	5	0,0027	0,0007	0,0014	0,0014	0,0032	0,0027	0,0006	0,0012	0,0013	0,0030
		7	0,0035	0,0008	0,0014	0,0019	0,0039	0,0038	0,0007	0,0012	0,0017	0,0036	
	10	7	0,0048	0,0011	0,0019	0,0024	0,0042	0,0051	0,0010	0,0016	0,0022	0,0039	
		11	0,0053	0,0013	0,0021	0,0025	0,0043	0,0056	0,0012	0,0018	0,0024	0,0040	
		25	7	0,0080	0,0014	0,0025	0,0034	0,0071	0,0081	0,0012	0,0020	0,0032	0,0066
	9	9	0,0094	0,0016	0,0026	0,0038	0,0070	0,0099	0,0014	0,0022	0,0038	0,0068	
		13	0,0113	0,0023	0,0037	0,0052	0,0081	0,0118	0,0021	0,0031	0,0052	0,0078	
		14	0,0120	0,0026	0,0041	0,0053	0,0080	0,0125	0,0024	0,0036	0,0053	0,0077	
	30	8	0,0141	0,0021	0,0036	0,0060	0,0117	0,0154	0,0019	0,0030	0,0057	0,0109	
		10	0,0180	0,0025	0,0041	0,0068	0,0119	0,0188	0,0022	0,0033	0,0068	0,0117	
		15	0,0246	0,0043	0,0065	0,0098	0,0141	0,0238	0,0037	0,0053	0,0098	0,0137	
		16	0,0229	0,0045	0,0067	0,0101	0,0142	0,0241	0,0041	0,0058	0,0097	0,0133	
	8	2	0,0001	0,0001	0,0002	0,0001	0,0003	0,0001	0,0001	0,0002	0,0001	0,0003	
		3	0,0001	0,0001	0,0002	0,0001	0,0003	0,0001	0,0001	0,0002	0,0001	0,0003	
		4	0,0001	0,0001	0,0003	0,0001	0,0003	0,0001	0,0001	0,0002	0,0001	0,0003	
		5	0,0002	0,0001	0,0003	0,0001	0,0003	0,0002	0,0001	0,0002	0,0001	0,0003	
		10	3	0,0002	0,0002	0,0004	0,0002	0,0005	0,0002	0,0001	0,0004	0,0002	0,0005
	4	4	0,0003	0,0002	0,0005	0,0002	0,0006	0,0003	0,0002	0,0004	0,0002	0,0006	
		5	0,0003	0,0002	0,0005	0,0003	0,0006	0,0003	0,0002	0,0005	0,0002	0,0006	
		6	0,0004	0,0002	0,0005	0,0003	0,0006	0,0003	0,0002	0,0005	0,0003	0,0006	
		12	3	0,0003	0,0002	0,0006	0,0002	0,0007	0,0003	0,0002	0,0006	0,0002	0,0007
		4	0,0005	0,0003	0,0007	0,0003	0,0009	0,0004	0,0003	0,0007	0,0003	0,0009	
	6	6	0,0006	0,0004	0,0008	0,0004	0,0010	0,0006	0,0004	0,0008	0,0004	0,0010	
		7	0,0007	0,0005	0,0009	0,0005	0,0011	0,0007	0,0004	0,0008	0,0005	0,0010	
		15	4	0,0008	0,0004	0,0011	0,0005	0,0014	0,0008	0,0004	0,0011	0,0005	0,0014
	5	5	0,0010	0,0005	0,0012	0,0006	0,0015	0,0011	0,0005	0,0012	0,0006	0,0015	
		8	0,0016	0,0008	0,0015	0,0009	0,0018	0,0017	0,0007	0,0013	0,0009	0,0017	
		9	0,0017	0,0008	0,0015	0,0010	0,0018	0,0016	0,0007	0,0013	0,0009	0,0017	
	20	5	0,0026	0,0010	0,0024	0,0012	0,0030	0,0027	0,0010	0,0024	0,0013	0,0030	
		7	0,0039	0,0013	0,0026	0,0017	0,0036	0,0041	0,0012	0,0024	0,0017	0,0034	
		10	0,0056	0,0018	0,0032	0,0024	0,0042	0,0055	0,0017	0,0029	0,0023	0,0040	
		11	0,0058	0,0019	0,0033	0,0026	0,0044	0,0058	0,0018	0,0030	0,0024	0,0040	
	25	7	0,0079	0,0024	0,0049	0,0029	0,0060	0,0082	0,0023	0,0046	0,0029	0,0059	
		9	0,0103	0,0030	0,0053	0,0041	0,0075	0,0106	0,0027	0,0048	0,0038	0,0068	
		13	0,0137	0,0040	0,0062	0,0053	0,0082	0,0138	0,0037	0,0056	0,0051	0,0078	
		14	0,0142	0,0044	0,0067	0,0056	0,0083	0,0142	0,0040	0,0060	0,0054	0,0079	
	30	8	0,0147	0,0040	0,0077	0,0051	0,0099	0,0166	0,0041	0,0079	0,0054	0,0102	
		10	0,0201	0,0051	0,0089	0,0071	0,0123	0,0215	0,0050	0,0086	0,0069	0,0117	
		15	0,0281	0,0073	0,0106	0,0099	0,0141	0,0301	0,0072	0,0103	0,0100	0,0139	
		16	0,0275	0,0072	0,0103	0,0099	0,0138	0,0294	0,0070	0,0099	0,0099	0,0134	
	TC4	8	2	0,0001	0,0001	0,0002	0,0001	0,0003	0,0001	0,0001	0,0003	0,0001	0,0003
			3	0,0001	0,0001	0,0003	0,0001	0,0003	0,0001	0,0001	0,0003	0,0001	0,0003
4			0,0001	0,0001	0,0003	0,0001	0,0003	0,0001	0,0001	0,0002	0,0001	0,0003	
5			0,0001	0,0001	0,0002	0,0001	0,0003	0,0001	0,0001	0,0002	0,0001	0,0003	
10			3	0,0002	0,0002	0,0005	0,0002	0,0005	0,0002	0,0002	0,0005	0,0002	0,0005
4		4	0,0002	0,0002	0,0005	0,0002	0,0006	0,0002	0,0002	0,0005	0,0002	0,0006	
		5	0,0003	0,0002	0,0005	0,0002	0,0006	0,0003	0,0002	0,0004	0,0002	0,0005	
		6	0,0003	0,0002	0,0004	0,0002	0,0006	0,0003	0,0002	0,0004	0,0002	0,0005	
		12	3	0,0003	0,0002	0,0007	0,0002	0,0007	0,0003	0,0002	0,0007	0,0002	0,0007

kontynuacja na następnej stronie

			inicjalizacja zachłanna					inicjalizacja losowa				
Problem			CPU[s]					CPU[s]				
typ	m	n	VNS	VNS'	VNS''	VNS' <sub>r</sub>	VNS'' <sub>r</sub>	VNS	VNS'	VNS''	VNS' <sub>r</sub>	VNS'' <sub>r</sub>
TC5		4	0,0004	0,0003	0,0007	0,0003	0,0008	0,0004	0,0003	0,0007	0,0003	0,0008
		6	0,0005	0,0003	0,0007	0,0004	0,0009	0,0005	0,0003	0,0007	0,0004	0,0009
		7	0,0006	0,0003	0,0006	0,0004	0,0009	0,0005	0,0003	0,0006	0,0004	0,0009
	15	4	0,0007	0,0005	0,0013	0,0005	0,0014	0,0008	0,0005	0,0012	0,0005	0,0014
		5	0,0010	0,0005	0,0013	0,0006	0,0015	0,0010	0,0005	0,0013	0,0006	0,0015
		8	0,0013	0,0006	0,0012	0,0008	0,0016	0,0013	0,0006	0,0011	0,0008	0,0015
		9	0,0014	0,0006	0,0011	0,0009	0,0016	0,0014	0,0006	0,0010	0,0008	0,0016
	20	5	0,0022	0,0011	0,0025	0,0012	0,0028	0,0025	0,0011	0,0028	0,0012	0,0030
		7	0,0035	0,0014	0,0030	0,0016	0,0034	0,0037	0,0014	0,0030	0,0017	0,0035
		10	0,0046	0,0018	0,0032	0,0021	0,0039	0,0047	0,0017	0,0030	0,0021	0,0037
		11	0,0048	0,0018	0,0032	0,0023	0,0039	0,0049	0,0017	0,0030	0,0022	0,0038
	25	7	0,0068	0,0026	0,0055	0,0030	0,0063	0,0073	0,0026	0,0054	0,0029	0,0061
		9	0,0083	0,0030	0,0055	0,0035	0,0065	0,0091	0,0030	0,0055	0,0036	0,0066
		13	0,0105	0,0035	0,0055	0,0046	0,0073	0,0112	0,0033	0,0053	0,0046	0,0072
		14	0,0109	0,0036	0,0057	0,0049	0,0076	0,0114	0,0033	0,0052	0,0049	0,0074
	30	8	0,0131	0,0046	0,0090	0,0050	0,0098	0,0141	0,0045	0,0090	0,0051	0,0101
		10	0,0176	0,0055	0,0098	0,0063	0,0113	0,0192	0,0056	0,0098	0,0067	0,0118
		15	0,0227	0,0065	0,0099	0,0088	0,0130	0,0243	0,0063	0,0094	0,0089	0,0129
		16	0,0229	0,0065	0,0096	0,0090	0,0129	0,0243	0,0062	0,0090	0,0091	0,0129
	8	2	0,0001	0,0001	0,0002	0,0001	0,0002	0,0001	0,0001	0,0003	0,0001	0,0003
		3	0,0001	0,0001	0,0003	0,0001	0,0003	0,0001	0,0001	0,0003	0,0001	0,0003
		4	0,0001	0,0001	0,0003	0,0001	0,0003	0,0001	0,0001	0,0003	0,0001	0,0003
		5	0,0001	0,0001	0,0003	0,0001	0,0003	0,0002	0,0001	0,0003	0,0001	0,0003
	10	3	0,0002	0,0002	0,0005	0,0002	0,0005	0,0002	0,0002	0,0005	0,0002	0,0005
		4	0,0003	0,0002	0,0005	0,0002	0,0006	0,0003	0,0002	0,0005	0,0002	0,0006
		5	0,0003	0,0003	0,0007	0,0003	0,0007	0,0003	0,0002	0,0006	0,0002	0,0006
		6	0,0004	0,0002	0,0005	0,0003	0,0006	0,0003	0,0002	0,0005	0,0003	0,0006
	12	3	0,0003	0,0002	0,0008	0,0002	0,0007	0,0003	0,0002	0,0008	0,0003	0,0008
		4	0,0004	0,0003	0,0008	0,0003	0,0008	0,0004	0,0003	0,0008	0,0003	0,0009
		6	0,0006	0,0004	0,0009	0,0005	0,0010	0,0007	0,0004	0,0008	0,0004	0,0010
		7	0,0007	0,0005	0,0010	0,0005	0,0011	0,0007	0,0005	0,0010	0,0005	0,0010
	15	4	0,0008	0,0005	0,0014	0,0005	0,0014	0,0008	0,0005	0,0014	0,0005	0,0014
		5	0,0010	0,0006	0,0013	0,0006	0,0016	0,0011	0,0006	0,0014	0,0006	0,0016
		8	0,0017	0,0009	0,0019	0,0010	0,0019	0,0017	0,0009	0,0018	0,0009	0,0018
		9	0,0016	0,0007	0,0013	0,0010	0,0018	0,0016	0,0007	0,0012	0,0009	0,0017
	20	5	0,0023	0,0012	0,0028	0,0012	0,0028	0,0027	0,0013	0,0030	0,0013	0,0031
		7	0,0036	0,0016	0,0034	0,0017	0,0034	0,0041	0,0017	0,0036	0,0018	0,0037
		10	0,0054	0,0018	0,0032	0,0024	0,0043	0,0056	0,0017	0,0030	0,0024	0,0041
		11	0,0056	0,0025	0,0043	0,0025	0,0044	0,0059	0,0024	0,0042	0,0026	0,0044
	25	7	0,0070	0,0025	0,0050	0,0028	0,0057	0,0080	0,0026	0,0053	0,0030	0,0062
	9	0,0098	0,0030	0,0055	0,0036	0,0065	0,0107	0,0030	0,0054	0,0038	0,0068	
	13	0,0133	0,0036	0,0056	0,0051	0,0079	0,0142	0,0034	0,0053	0,0053	0,0079	
	14	0,0130	0,0053	0,0081	0,0057	0,0085	0,0145	0,0055	0,0082	0,0058	0,0085	
30	8	0,0146	0,0047	0,0089	0,0053	0,0102	0,0166	0,0049	0,0093	0,0056	0,0107	
	10	0,0187	0,0051	0,0089	0,0066	0,0113	0,0214	0,0055	0,0093	0,0071	0,0121	
	15	0,0274	0,0098	0,0140	0,0104	0,0148	0,0291	0,0096	0,0138	0,0109	0,0151	
	16	0,0273	0,0062	0,0090	0,0098	0,0137	0,0290	0,0058	0,0083	0,0102	0,0138	
TC6	8	2	0,0001	0,0001	0,0002	0,0001	0,0003	0,0001	0,0001	0,0003	0,0001	0,0003
		3	0,0001	0,0001	0,0002	0,0001	0,0003	0,0001	0,0001	0,0003	0,0001	0,0003
		4	0,0001	0,0001	0,0003	0,0001	0,0003	0,0001	0,0001	0,0003	0,0001	0,0003
		5	0,0001	0,0001	0,0002	0,0001	0,0003	0,0001	0,0001	0,0002	0,0001	0,0003
10	3	0,0002	0,0001	0,0004	0,0002	0,0005	0,0002	0,0001	0,0005	0,0002	0,0005	
	4	0,0002	0,0002	0,0006	0,0002	0,0006	0,0002	0,0002	0,0006	0,0002	0,0006	
	5	0,0003	0,0002	0,0005	0,0002	0,0006	0,0003	0,0002	0,0004	0,0002	0,0005	
	6	0,0003	0,0002	0,0005	0,0003	0,0006	0,0003	0,0002	0,0005	0,0002	0,0005	

kontynuacja na następnej stronie

			inicjalizacja zachłanna					inicjalizacja losowa					
Problem			CPU[s]					CPU[s]					
typ	m	n	VNS	VNS'	VNS''	VNS' <sub>r</sub>	VNS'' <sub>r</sub>	VNS	VNS'	VNS''	VNS' <sub>r</sub>	VNS'' <sub>r</sub>	
TC7	12	3	0,0003	0,0002	0,0007	0,0002	0,0007	0,0003	0,0002	0,0007	0,0002	0,0008	
		4	0,0004	0,0003	0,0008	0,0003	0,0009	0,0004	0,0003	0,0009	0,0003	0,0009	
		6	0,0006	0,0004	0,0009	0,0004	0,0010	0,0006	0,0004	0,0009	0,0004	0,0009	
	7	0,0006	0,0004	0,0008	0,0005	0,0009	0,0006	0,0004	0,0007	0,0004	0,0009		
		15	4	0,0007	0,0005	0,0012	0,0005	0,0014	0,0008	0,0005	0,0013	0,0005	0,0014
			5	0,0009	0,0006	0,0015	0,0006	0,0016	0,0010	0,0006	0,0015	0,0007	0,0016
	8		0,0014	0,0006	0,0012	0,0009	0,0017	0,0015	0,0006	0,0011	0,0008	0,0016	
	9	0,0014	0,0009	0,0017	0,0009	0,0017	0,0015	0,0009	0,0016	0,0009	0,0017		
		20	5	0,0021	0,0010	0,0024	0,0011	0,0025	0,0025	0,0011	0,0027	0,0012	0,0029
			7	0,0035	0,0014	0,0029	0,0016	0,0033	0,0039	0,0014	0,0029	0,0017	0,0035
	10		0,0048	0,0022	0,0040	0,0023	0,0042	0,0052	0,0022	0,0039	0,0024	0,0042	
	11	0,0049	0,0016	0,0028	0,0023	0,0040	0,0051	0,0016	0,0026	0,0023	0,0039		
		25	7	0,0066	0,0027	0,0057	0,0029	0,0060	0,0076	0,0029	0,0060	0,0031	0,0064
			9	0,0085	0,0035	0,0064	0,0038	0,0070	0,0099	0,0037	0,0067	0,0040	0,0073
	13		0,0118	0,0048	0,0076	0,0052	0,0080	0,0130	0,0049	0,0078	0,0053	0,0081	
	14	0,0121	0,0034	0,0053	0,0052	0,0078	0,0133	0,0032	0,0049	0,0052	0,0077		
		30	8	0,0138	0,0051	0,0098	0,0056	0,0107	0,0159	0,0055	0,0106	0,0059	0,0114
			10	0,0170	0,0061	0,0107	0,0068	0,0118	0,0199	0,0067	0,0116	0,0077	0,0132
	15		0,0237	0,0055	0,0082	0,0092	0,0133	0,0268	0,0054	0,0079	0,0096	0,0135	
	16	0,0242	0,0093	0,0132	0,0109	0,0153	0,0270	0,0094	0,0134	0,0107	0,0147		
		8	2	0,0001	0,0001	0,0002	0,0001	0,0003	0,0001	0,0001	0,0003	0,0001	0,0003
			3	0,0001	0,0001	0,0003	0,0001	0,0003	0,0001	0,0001	0,0003	0,0001	0,0003
	4		0,0001	0,0001	0,0003	0,0001	0,0003	0,0001	0,0001	0,0003	0,0001	0,0003	
	5	0,0002	0,0001	0,0003	0,0001	0,0003	0,0001	0,0001	0,0003	0,0001	0,0003		
		10	3	0,0002	0,0002	0,0005	0,0002	0,0005	0,0002	0,0002	0,0005	0,0002	0,0005
			4	0,0003	0,0002	0,0005	0,0002	0,0006	0,0003	0,0002	0,0005	0,0002	0,0006
	5		0,0003	0,0003	0,0007	0,0003	0,0007	0,0003	0,0003	0,0006	0,0003	0,0006	
	6	0,0004	0,0003	0,0006	0,0003	0,0006	0,0003	0,0003	0,0006	0,0003	0,0006		
		12	3	0,0003	0,0002	0,0007	0,0002	0,0007	0,0003	0,0002	0,0008	0,0002	0,0008
			4	0,0004	0,0003	0,0009	0,0003	0,0009	0,0004	0,0003	0,0009	0,0003	0,0009
	6		0,0006	0,0004	0,0009	0,0004	0,0010	0,0006	0,0004	0,0009	0,0004	0,0010	
	7	0,0007	0,0005	0,0011	0,0005	0,0011	0,0007	0,0005	0,0010	0,0005	0,0010		
		15	4	0,0008	0,0005	0,0015	0,0005	0,0015	0,0008	0,0005	0,0014	0,0005	0,0014
			5	0,0010	0,0006	0,0015	0,0006	0,0015	0,0011	0,0006	0,0016	0,0006	0,0016
	8		0,0017	0,0009	0,0019	0,0010	0,0019	0,0017	0,0009	0,0018	0,0009	0,0018	
	9	0,0017	0,0008	0,0014	0,0010	0,0018	0,0017	0,0007	0,0013	0,0009	0,0017		
		20	5	0,0023	0,0011	0,0027	0,0012	0,0029	0,0027	0,0012	0,0029	0,0013	0,0030
			7	0,0035	0,0016	0,0033	0,0016	0,0034	0,0040	0,0017	0,0035	0,0017	0,0036
	10		0,0056	0,0024	0,0044	0,0026	0,0046	0,0057	0,0023	0,0041	0,0025	0,0043	
	11	0,0059	0,0021	0,0037	0,0026	0,0044	0,0059	0,0020	0,0033	0,0025	0,0042		
		25	7	0,0068	0,0026	0,0052	0,0027	0,0055	0,0081	0,0027	0,0055	0,0030	0,0061
			9	0,0092	0,0036	0,0066	0,0038	0,0069	0,0104	0,0038	0,0068	0,0040	0,0071
	13		0,0124	0,0035	0,0055	0,0050	0,0076	0,0138	0,0035	0,0053	0,0051	0,0077	
	14	0,0134	0,0054	0,0082	0,0057	0,0085	0,0144	0,0053	0,0080	0,0058	0,0084		
		30	8	0,0144	0,0053	0,0101	0,0054	0,0104	0,0163	0,0055	0,0106	0,0058	0,0110
			10	0,0189	0,0067	0,0115	0,0070	0,0120	0,0216	0,0071	0,0122	0,0075	0,0127
	15		0,0279	0,0066	0,0097	0,0104	0,0148	0,0305	0,0067	0,0096	0,0101	0,0140	
	16	0,0262	0,0095	0,0133	0,0112	0,0155	0,0286	0,0097	0,0134	0,0110	0,0148		
8		2	0,0000	0,0001	0,0002	0,0001	0,0002	0,0000	0,0001	0,0002	0,0001	0,0003	
		3	0,0001	0,0001	0,0003	0,0001	0,0003	0,0001	0,0001	0,0003	0,0001	0,0003	
	4	0,0001	0,0001	0,0003	0,0001	0,0003	0,0001	0,0001	0,0003	0,0001	0,0003		
5	0,0001	0,0001	0,0002	0,0001	0,0003	0,0001	0,0001	0,0002	0,0001	0,0003			
	10	3	0,0002	0,0002	0,0005	0,0002	0,0005	0,0002	0,0002	0,0005	0,0002	0,0005	
		4	0,0003	0,0002	0,0004	0,0002	0,0006	0,0003	0,0002	0,0004	0,0002	0,0006	
5		0,0003	0,0002	0,0005	0,0003	0,0006	0,0003	0,0002	0,0005	0,0003	0,0006		

kontynuacja na następnej stronie

			inicjalizacja zachłanna					inicjalizacja losowa					
Problem			CPU[s]					CPU[s]					
typ	m	n	VNS	VNS'	VNS''	VNS' <sub>r</sub>	VNS'' <sub>r</sub>	VNS	VNS'	VNS''	VNS' <sub>r</sub>	VNS'' <sub>r</sub>	
TC9	12	6	0,0004	0,0003	0,0006	0,0003	0,0007	0,0004	0,0003	0,0006	0,0003	0,0006	
		3	0,0003	0,0002	0,0007	0,0003	0,0008	0,0003	0,0002	0,0007	0,0003	0,0008	
		4	0,0004	0,0003	0,0008	0,0003	0,0009	0,0004	0,0003	0,0008	0,0003	0,0009	
		6	0,0006	0,0003	0,0007	0,0004	0,0010	0,0006	0,0003	0,0007	0,0004	0,0010	
		7	0,0007	0,0004	0,0008	0,0005	0,0011	0,0007	0,0004	0,0007	0,0005	0,0010	
		4	0,0008	0,0005	0,0013	0,0005	0,0014	0,0008	0,0005	0,0013	0,0005	0,0014	
	15	5	0,0011	0,0007	0,0016	0,0007	0,0017	0,0011	0,0006	0,0016	0,0007	0,0017	
		8	0,0016	0,0009	0,0018	0,0010	0,0020	0,0016	0,0009	0,0018	0,0010	0,0019	
		9	0,0017	0,0007	0,0012	0,0010	0,0018	0,0016	0,0006	0,0010	0,0009	0,0017	
	20	5	0,0024	0,0011	0,0025	0,0012	0,0029	0,0027	0,0011	0,0026	0,0013	0,0032	
		7	0,0036	0,0016	0,0034	0,0017	0,0035	0,0040	0,0017	0,0036	0,0018	0,0038	
		10	0,0053	0,0023	0,0042	0,0025	0,0045	0,0056	0,0023	0,0042	0,0025	0,0045	
	25	11	0,0056	0,0017	0,0029	0,0026	0,0044	0,0058	0,0016	0,0027	0,0025	0,0042	
		7	0,0072	0,0024	0,0049	0,0028	0,0058	0,0083	0,0025	0,0051	0,0030	0,0062	
		9	0,0098	0,0039	0,0070	0,0042	0,0076	0,0108	0,0038	0,0070	0,0042	0,0076	
	30	13	0,0128	0,0028	0,0044	0,0051	0,0079	0,0139	0,0027	0,0041	0,0051	0,0077	
		14	0,0137	0,0034	0,0052	0,0053	0,0080	0,0142	0,0031	0,0046	0,0054	0,0079	
		8	0,0141	0,0052	0,0099	0,0054	0,0105	0,0165	0,0056	0,0107	0,0060	0,0116	
	TC9	10	8	0,0182	0,0044	0,0076	0,0071	0,0122	0,0217	0,0049	0,0083	0,0072	0,0123
			15	0,0259	0,0050	0,0075	0,0099	0,0141	0,0291	0,0049	0,0072	0,0101	0,0140
		16	0,0249	0,0048	0,0072	0,0098	0,0136	0,0289	0,0049	0,0071	0,0100	0,0136	
		8	2	0,0001	0,0001	0,0003	0,0001	0,0003	0,0001	0,0001	0,0003	0,0001	0,0003
			3	0,0001	0,0001	0,0003	0,0001	0,0003	0,0001	0,0001	0,0003	0,0001	0,0003
			4	0,0001	0,0001	0,0003	0,0001	0,0003	0,0001	0,0001	0,0003	0,0001	0,0003
		10	5	0,0001	0,0001	0,0003	0,0001	0,0003	0,0002	0,0001	0,0003	0,0001	0,0003
			3	0,0002	0,0002	0,0005	0,0002	0,0006	0,0002	0,0002	0,0005	0,0002	0,0005
	4		0,0003	0,0002	0,0006	0,0002	0,0006	0,0003	0,0002	0,0006	0,0002	0,0006	
	TC9	12	5	0,0003	0,0002	0,0006	0,0003	0,0006	0,0003	0,0002	0,0006	0,0002	0,0006
			6	0,0004	0,0003	0,0006	0,0003	0,0006	0,0003	0,0003	0,0006	0,0003	0,0006
			3	0,0003	0,0002	0,0007	0,0002	0,0007	0,0003	0,0003	0,0008	0,0002	0,0008
		15	4	0,0004	0,0003	0,0009	0,0003	0,0009	0,0004	0,0003	0,0009	0,0003	0,0009
			6	0,0006	0,0004	0,0010	0,0004	0,0010	0,0006	0,0004	0,0010	0,0004	0,0010
			7	0,0007	0,0005	0,0011	0,0005	0,0010	0,0007	0,0005	0,0010	0,0005	0,0010
		20	4	0,0007	0,0005	0,0013	0,0005	0,0013	0,0008	0,0005	0,0014	0,0005	0,0014
			5	0,0010	0,0006	0,0015	0,0006	0,0015	0,0011	0,0006	0,0016	0,0006	0,0016
	8		0,0017	0,0009	0,0019	0,0009	0,0018	0,0017	0,0009	0,0018	0,0009	0,0017	
	25	9	0,0017	0,0010	0,0018	0,0010	0,0018	0,0016	0,0009	0,0017	0,0009	0,0017	
		5	0,0025	0,0013	0,0029	0,0012	0,0029	0,0027	0,0013	0,0031	0,0013	0,0031	
		7	0,0038	0,0017	0,0035	0,0017	0,0035	0,0041	0,0017	0,0035	0,0017	0,0035	
		10	0,0060	0,0026	0,0047	0,0026	0,0047	0,0059	0,0024	0,0043	0,0024	0,0043	
		11	0,0061	0,0026	0,0046	0,0026	0,0045	0,0060	0,0025	0,0042	0,0025	0,0042	
	30	7	0,0074	0,0029	0,0060	0,0029	0,0060	0,0081	0,0029	0,0060	0,0030	0,0060	
		9	0,0094	0,0037	0,0066	0,0036	0,0066	0,0108	0,0038	0,0069	0,0039	0,0069	
		13	0,0131	0,0052	0,0080	0,0052	0,0080	0,0144	0,0052	0,0080	0,0053	0,0080	
		14	0,0135	0,0054	0,0082	0,0054	0,0082	0,0145	0,0054	0,0081	0,0055	0,0080	
		8	0,0147	0,0052	0,0100	0,0052	0,0100	0,0163	0,0054	0,0103	0,0054	0,0104	
	TC10	10	10	0,0183	0,0064	0,0110	0,0064	0,0110	0,0219	0,0071	0,0122	0,0072	0,0122
			15	0,0287	0,0102	0,0145	0,0102	0,0145	0,0310	0,0102	0,0145	0,0104	0,0144
16			0,0280	0,0101	0,0141	0,0101	0,0141	0,0300	0,0100	0,0139	0,0102	0,0138	
2			0,0001	0,0001	0,0002	0,0001	0,0002	0,0000	0,0001	0,0003	0,0001	0,0003	
8		3	0,0001	0,0001	0,0003	0,0001	0,0003	0,0001	0,0001	0,0003	0,0001	0,0003	
		4	0,0001	0,0001	0,0003	0,0001	0,0003	0,0001	0,0001	0,0003	0,0001	0,0003	
		5	0,0002	0,0001	0,0003	0,0001	0,0003	0,0002	0,0001	0,0003	0,0001	0,0003	
		3	0,0002	0,0002	0,0005	0,0002	0,0005	0,0002	0,0002	0,0005	0,0002	0,0005	
4	0,0002	0,0002	0,0006	0,0002	0,0006	0,0003	0,0002	0,0006	0,0002	0,0006			

kontynuacja na następnej stronie

Problem			inicjalizacja zachłanna					inicjalizacja losowa				
typ	m	n	VNS	VNS'	CPU[s]	VNS' <sub>r</sub>	VNS'' <sub>r</sub>	VNS	VNS'	CPU[s]	VNS' <sub>r</sub>	VNS'' <sub>r</sub>
TC11	5	5	0,0003	0,0003	0,0006	0,0002	0,0006	0,0003	0,0002	0,0006	0,0002	0,0006
		6	0,0003	0,0003	0,0006	0,0003	0,0006	0,0004	0,0003	0,0006	0,0003	0,0006
	12	3	0,0003	0,0002	0,0007	0,0002	0,0007	0,0003	0,0003	0,0008	0,0002	0,0008
		4	0,0004	0,0003	0,0008	0,0003	0,0008	0,0004	0,0003	0,0009	0,0003	0,0009
	6	6	0,0006	0,0005	0,0011	0,0005	0,0011	0,0006	0,0004	0,0011	0,0004	0,0010
		7	0,0007	0,0005	0,0011	0,0005	0,0011	0,0007	0,0005	0,0010	0,0005	0,0010
	15	4	0,0007	0,0005	0,0013	0,0005	0,0013	0,0007	0,0005	0,0014	0,0005	0,0014
		5	0,0009	0,0006	0,0014	0,0006	0,0014	0,0010	0,0006	0,0016	0,0006	0,0016
	8	8	0,0015	0,0009	0,0019	0,0009	0,0019	0,0016	0,0009	0,0019	0,0010	0,0018
		9	0,0016	0,0010	0,0019	0,0010	0,0018	0,0016	0,0010	0,0018	0,0010	0,0018
	20	5	0,0021	0,0012	0,0028	0,0011	0,0028	0,0025	0,0013	0,0031	0,0013	0,0031
		7	0,0035	0,0017	0,0036	0,0017	0,0036	0,0039	0,0018	0,0039	0,0018	0,0038
	10	10	0,0055	0,0026	0,0046	0,0026	0,0046	0,0056	0,0025	0,0045	0,0026	0,0045
		11	0,0059	0,0028	0,0048	0,0028	0,0047	0,0059	0,0027	0,0045	0,0027	0,0045
	25	7	0,0071	0,0032	0,0066	0,0032	0,0066	0,0075	0,0031	0,0064	0,0031	0,0064
		9	0,0092	0,0041	0,0074	0,0041	0,0074	0,0103	0,0042	0,0078	0,0043	0,0078
	13	13	0,0127	0,0057	0,0088	0,0057	0,0087	0,0139	0,0058	0,0089	0,0059	0,0088
		14	0,0133	0,0060	0,0091	0,0061	0,0090	0,0144	0,0061	0,0092	0,0063	0,0091
	30	8	0,0130	0,0054	0,0103	0,0053	0,0103	0,0143	0,0055	0,0107	0,0055	0,0107
		10	0,0181	0,0073	0,0125	0,0073	0,0125	0,0200	0,0076	0,0131	0,0077	0,0131
	15	15	0,0265	0,0110	0,0155	0,0110	0,0154	0,0284	0,0111	0,0156	0,0113	0,0155
		16	0,0277	0,0116	0,0159	0,0116	0,0159	0,0296	0,0118	0,0161	0,0120	0,0161
	8	2	0,0001	0,0001	0,0003	0,0001	0,0003	0,0001	0,0001	0,0003	0,0001	0,0003
		3	0,0001	0,0001	0,0003	0,0001	0,0003	0,0001	0,0001	0,0003	0,0001	0,0003
	4	4	0,0001	0,0001	0,0003	0,0001	0,0003	0,0001	0,0001	0,0003	0,0001	0,0003
		5	0,0002	0,0001	0,0003	0,0001	0,0003	0,0002	0,0001	0,0003	0,0001	0,0003
	10	3	0,0002	0,0002	0,0005	0,0002	0,0005	0,0002	0,0002	0,0005	0,0002	0,0005
		4	0,0003	0,0002	0,0006	0,0002	0,0006	0,0003	0,0002	0,0006	0,0002	0,0006
	5	5	0,0003	0,0003	0,0006	0,0003	0,0006	0,0003	0,0002	0,0006	0,0002	0,0006
		6	0,0004	0,0003	0,0006	0,0003	0,0006	0,0004	0,0003	0,0006	0,0003	0,0006
	12	3	0,0003	0,0002	0,0008	0,0002	0,0008	0,0003	0,0002	0,0008	0,0003	0,0008
		4	0,0004	0,0003	0,0009	0,0003	0,0009	0,0004	0,0003	0,0009	0,0003	0,0009
	6	6	0,0006	0,0004	0,0010	0,0004	0,0010	0,0006	0,0004	0,0010	0,0004	0,0010
		7	0,0007	0,0005	0,0010	0,0005	0,0010	0,0007	0,0005	0,0010	0,0005	0,0010
	15	4	0,0008	0,0005	0,0014	0,0005	0,0014	0,0008	0,0005	0,0014	0,0005	0,0014
		5	0,0010	0,0006	0,0015	0,0006	0,0015	0,0011	0,0006	0,0016	0,0006	0,0016
	8	8	0,0017	0,0009	0,0019	0,0009	0,0018	0,0017	0,0009	0,0018	0,0009	0,0017
		9	0,0017	0,0010	0,0018	0,0010	0,0018	0,0016	0,0009	0,0017	0,0009	0,0017
	20	5	0,0025	0,0013	0,0030	0,0012	0,0030	0,0027	0,0013	0,0031	0,0013	0,0031
		7	0,0038	0,0017	0,0036	0,0017	0,0035	0,0041	0,0017	0,0036	0,0017	0,0035
	10	10	0,0060	0,0026	0,0047	0,0026	0,0046	0,0059	0,0024	0,0043	0,0024	0,0042
		11	0,0061	0,0026	0,0046	0,0026	0,0046	0,0060	0,0025	0,0042	0,0025	0,0042
	25	7	0,0072	0,0029	0,0059	0,0029	0,0059	0,0081	0,0030	0,0061	0,0030	0,0061
		9	0,0093	0,0037	0,0066	0,0036	0,0066	0,0107	0,0038	0,0070	0,0039	0,0069
	13	13	0,0129	0,0051	0,0079	0,0051	0,0078	0,0144	0,0052	0,0080	0,0053	0,0080
		14	0,0133	0,0054	0,0081	0,0054	0,0081	0,0145	0,0054	0,0081	0,0055	0,0080
	30	8	0,0153	0,0055	0,0105	0,0055	0,0105	0,0163	0,0055	0,0105	0,0055	0,0105
		10	0,0179	0,0063	0,0109	0,0063	0,0108	0,0217	0,0071	0,0122	0,0072	0,0121
15	15	0,0288	0,0102	0,0146	0,0102	0,0145	0,0310	0,0102	0,0145	0,0104	0,0144	
	16	0,0280	0,0101	0,0141	0,0101	0,0141	0,0300	0,0100	0,0139	0,0102	0,0139	
8	2	0,0001	0,0001	0,0002	0,0001	0,0002	0,0001	0,0001	0,0003	0,0001	0,0003	
	3	0,0001	0,0001	0,0003	0,0001	0,0003	0,0001	0,0001	0,0003	0,0001	0,0003	
4	4	0,0001	0,0001	0,0003	0,0001	0,0003	0,0001	0,0001	0,0003	0,0001	0,0003	
	5	0,0002	0,0001	0,0003	0,0001	0,0003	0,0002	0,0001	0,0003	0,0001	0,0003	
10	3	0,0002	0,0002	0,0005	0,0002	0,0005	0,0002	0,0002	0,0005	0,0002	0,0005	

kontynuacja na następnej stronie



			inicjalizacja zachłanna					inicjalizacja losowa				
Problem			CPU[s]					CPU[s]				
typ	m	n	VNS	VNS'	VNS''	VNS <sub>r</sub> '	VNS <sub>r</sub> ''	VNS	VNS'	VNS''	VNS <sub>r</sub> '	VNS <sub>r</sub> ''
	4		0,0003	0,0002	0,0006	0,0002	0,0006	0,0003	0,0002	0,0006	0,0002	0,0006
	5		0,0003	0,0002	0,0006	0,0003	0,0006	0,0003	0,0002	0,0006	0,0003	0,0006
	6		0,0003	0,0003	0,0006	0,0003	0,0006	0,0003	0,0003	0,0006	0,0003	0,0006
12	3		0,0003	0,0002	0,0007	0,0002	0,0007	0,0003	0,0003	0,0008	0,0003	0,0008
	4		0,0004	0,0003	0,0008	0,0003	0,0008	0,0004	0,0003	0,0009	0,0003	0,0009
	6		0,0006	0,0005	0,0010	0,0004	0,0010	0,0006	0,0004	0,0010	0,0004	0,0010
	7		0,0007	0,0005	0,0011	0,0005	0,0011	0,0007	0,0005	0,0010	0,0005	0,0010
15	4		0,0008	0,0005	0,0014	0,0005	0,0014	0,0008	0,0005	0,0014	0,0005	0,0014
	5		0,0010	0,0006	0,0015	0,0006	0,0015	0,0011	0,0007	0,0016	0,0007	0,0016
	8		0,0016	0,0009	0,0018	0,0009	0,0018	0,0017	0,0009	0,0018	0,0009	0,0018
	9		0,0016	0,0010	0,0018	0,0010	0,0018	0,0016	0,0009	0,0017	0,0009	0,0017
20	5		0,0022	0,0012	0,0028	0,0012	0,0028	0,0025	0,0013	0,0030	0,0013	0,0030
	7		0,0038	0,0018	0,0037	0,0018	0,0036	0,0041	0,0018	0,0037	0,0018	0,0037
	10		0,0055	0,0024	0,0044	0,0025	0,0044	0,0057	0,0024	0,0043	0,0024	0,0043
	11		0,0058	0,0026	0,0045	0,0026	0,0045	0,0061	0,0026	0,0044	0,0026	0,0043
25	7		0,0066	0,0028	0,0057	0,0028	0,0057	0,0079	0,0030	0,0062	0,0031	0,0062
	9		0,0091	0,0038	0,0068	0,0038	0,0068	0,0106	0,0040	0,0073	0,0040	0,0072
	13		0,0127	0,0053	0,0082	0,0053	0,0082	0,0140	0,0054	0,0083	0,0055	0,0083
	14		0,0131	0,0056	0,0084	0,0056	0,0084	0,0144	0,0057	0,0085	0,0058	0,0084
30	8		0,0136	0,0052	0,0099	0,0052	0,0099	0,0159	0,0057	0,0109	0,0057	0,0110
	10		0,0177	0,0066	0,0113	0,0066	0,0113	0,0209	0,0073	0,0125	0,0074	0,0125
	15		0,0280	0,0107	0,0150	0,0106	0,0150	0,0303	0,0107	0,0151	0,0109	0,0151
	16		0,0294	0,0114	0,0157	0,0114	0,0157	0,0303	0,0110	0,0151	0,0112	0,0150

Tabela C.2: Statystyki jakościowe metody VNS z agregacją OWA dla problemów o małym rozmiarze (od 8 do 30 lokalizacji); VNS oznacza wersje bez regularyzacji (VNS, VNS', VNS''), a VNS<sub>r</sub> wersje z regularyzacją (VNS<sub>r</sub>', VNS<sub>r</sub>'')

odstęp [%]			inicjalizacja zachłanna						inicjalizacja losowa					
Problem			VNS			VNS <sub>r</sub>			VNS			VNS <sub>r</sub>		
typ	m	n	max.	śr.	min.	max.	śr.	min.	max.	śr.	min.	max.	śr.	min.
TC1	8	2	0	0	0	0	0	0	10,64	0,71	0	10,64	0,71	0
		3	3,39	0,23	0	3,39	0,23	0	2,02	0,13	0	2,02	0,13	0
		4	12,90	0,86	0	12,90	0,86	0	0	0	0	0	0	0
		5	0	0	0	0	0	0	0	0	0	0	0	0
	10	3	12,90	0,86	0	12,90	0,86	0	2,04	0,14	0	2,04	0,14	0
		4	18,18	1,21	0	18,18	1,21	0	3,23	0,22	0	3,23	0,22	0
		5	0	0	0	0	0	0	0	0	0	0	0	0
		6	0	0	0	0	0	0	0	0	0	0	0	0
	12	3	18,26	2,04	0	18,26	2,04	0	18,26	1,22	0	18,26	1,22	0
		4	4,95	0,33	0	4,95	0,33	0	7,96	0,53	0	7,96	0,53	0
		6	0	0	0	0	0	0	0	0	0	0	0	0
		7	0	0	0	0	0	0	0	0	0	0	0	0
	15	4	0	0	0	0	0	0	4,60	0,31	0	4,60	0,31	0
		5	16,98	1,33	0	16,98	1,33	0	16,98	1,13	0	16,98	1,13	0
		8	0	0	0	0	0	0	0	0	0	0	0	0
		9	0	0	0	0	0	0	0	0	0	0	0	0
	20	5	6,48	0,70	0	6,48	0,70	0	20,00	2,22	0	20,00	2,22	0
		7	3,61	0,24	0	3,61	0,24	0	3,61	0,35	0	3,61	0,35	0
		10	2,08	0,14	0	2,08	0,14	0	0	0	0	0	0	0
		11	0	0	0	0	0	0	4,17	0,28	0	4,17	0,28	0
	25	7	6,74	0,65	0	6,74	0,65	0	0	0	0	0	0	0
		9	0	0	0	0	0	0	1,54	0,10	0	1,54	0,10	0
		13	0	0	0	0	0	0	0	0	0	0	0	0
		14	0	0	0	0	0	0	0	0	0	0	0	0

kontynuacja na następnej stronie

odstęp [%]			inicjalizacja zachłanna						inicjalizacja losowa						
Problem			VNS			VNS <sub>r</sub>			VNS			VNS <sub>r</sub>			
typ	m	n	max.	śr.	min.	max.	śr.	min.	max.	śr.	min.	max.	śr.	min.	
TC2	30	8	0	0	0	0	0	0	5,65	0,43	0	5,65	0,43	0	
		10	1,64	0,28	0	1,64	0,28	0	7,14	0,96	0	7,14	0,96	0	
		15	0	0	0	0	0	0	0	0	0	0	0	0	0
		16	0	0	0	0	0	0	0	0	0	0	0	0	0
	8	2	48,00	4,87	0	48,00	3,20	0	14,29	1,49	0	14,29	1,09	0	
		3	11,43	0,76	0	11,43	0,76	0	11,43	0,76	0	37,50	3,26	0	
		4	0	0	0	0	0	0	50,00	5,24	0	0	0	0	0
		5	0	0	0	0	0	0	0	0	0	0	0	0	0
	10	3	0	0	0	0	0	0	0	0	0	20,00	1,33	0	
		4	10,00	0,67	0	50,00	4,58	0	9,09	0,61	0	18,75	1,25	0	
		5	0	0	0	0	0	0	0	0	0	0	0	0	0
	6	88,89	5,93	0	0	0	0	0	0	0	0	0	0	0	0
		12	3	73,68	5,85	0	0	0	0	24,14	2,07	0	0	0	0
		4	27,27	2,12	0	12,50	0,83	0	27,27	1,82	0	0	0	0	0
	6	0	0	0	0	0	0	0	17,65	1,18	0	0	0	0	0
		7	0	0	0	0	0	0	0	0	0	0	0	0	0
		15	4	17,39	2,65	0	28,57	3,06	0	17,39	3,41	0	0	0	0
	5	22,22	1,96	0	22,22	2,51	0	22,22	1,96	0	22,22	1,96	0	0	0
		8	0	0	0	0	0	0	83,33	5,56	0	0	0	0	0
		9	66,67	5,78	0	0	0	0	0	0	0	0	0	0	0
	20	5	14,29	2,04	0	20,00	3,36	0	22,22	5,89	0	14,29	3,27	0	
		7	15,39	2,52	0	15,39	2,43	0	25,00	4,45	0	15,39	2,21	0	
		10	150,00	10,74	0	0	0	0	0	0	0	0	0	0	0
	11	150,00	10,00	0	0	0	0	0	25,00	1,67	0	0	0	0	0
25		7	33,33	10,04	0	15,39	2,21	0	33,33	6,97	0	8,33	2,03	0	
9		37,50	5,24	0	14,29	0,95	0	28,57	5,99	0	25,00	2,41	0		
13	75,00	7,67	0	0	0	0	0	40,00	4,89	0	0	0	0	0	
	14	133,33	21,67	0	0	0	0	66,67	8,44	0	0	0	0	0	
	30	8	38,46	8,73	0	15,39	2,19	0	38,46	12,05	0	15,39	2,19	0	
10	57,14	9,08	0	12,50	2,17	0	0	42,86	16,20	0	14,29	2,45	0		
	15	66,67	9,33	0	0	0	0	75,00	9,00	0	0	0	0	0	
	16	150,00	22,33	0	0	0	0	133,33	17,67	0	0	0	0	0	
TC3	8	2	23,29	1,55	0	23,29	1,78	0	26,00	1,73	0	26,00	1,73	0	
		3	4,35	0,29	0	4,35	0,29	0	42,50	3,24	0	42,50	3,24	0	
		4	0	0	0	0	0	0	23,33	1,56	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		10	3	0	0	0	0	0	0	0	0	0	0	0	0
		4	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	2,38	0,16	0	0	0	0	0	2,38	0,16	0	0	0	0	0
		6	0	0	0	0	0	0	0	0	0	0	0	0	0
		12	3	17,95	1,20	0	17,95	1,20	0	22,22	2,99	0	22,22	2,99	0
	4	15,69	1,05	0	7,14	0,48	0	0	0	0	0	0	0	0	0
		6	0	0	0	0	0	0	0	0	0	0	0	0	0
		7	0	0	0	0	0	0	0	0	0	0	0	0	0
15	4	4,92	0,33	0	4,92	0,33	0	0	0	0	0	0	0	0	
	5	7,69	0,63	0	7,69	0,63	0	19,44	1,30	0	19,44	1,30	0		
	8	0	0	0	0	0	0	0	0	0	0	0	0	0	
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	20	5	16,22	1,42	0	42,31	4,82	0	3,16	0,29	0	5,56	0,66	0	
	7	10,17	0,68	0	14,89	1,78	0	7,32	0,80	0	0	0	0	0	
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	11	0	0	0	0	0	0	0	0	0	0	0	0	0	
	25	7	15,25	1,79	0	15,25	1,79	0	4,44	0,30	0	8,07	0,54	0	
9	6,78	0,99	0	6,78	0,74	0	0	4,26	0,28	0	4,26	0,45	0		
	13	0	0	0	0	0	0	0	0	0	0	0	0	0	

kontynuacja na następnej stronie

odstęp [%]			inicjalizacja zachłanna						inicjalizacja losowa						
Problem			VNS			VNS <sub>r</sub>			VNS			VNS <sub>r</sub>			
typ	m	n	max.	śr.	min.	max.	śr.	min.	max.	śr.	min.	max.	śr.	min.	
TC4	30	14	0	0	0	12,50	0,83	0	0	0	0	0	0	0	
		8	15,07	1,58	0	15,63	2,41	0	5,33	1,05	0	5,33	0,76	0	
		10	5,80	0,94	0	9,26	1,56	0	11,54	0,87	0	11,54	1,63	0	
		15	0	0	0	0	0	0	0	0	0	0	0	0	0
		16	0	0	0	0	0	0	0	0	0	0	0	0	0
	8	2	0	0	0	0	0	0	0	0	0	0	0	0	0
		3	0	0	0	0	0	0	0	9,30	0,62	0	9,30	0,62	0
		4	0	0	0	0	0	0	0	0	0	0	0	0	0
		5	0	0	0	0	0	0	0	0	0	0	0	0	0
		10	3	4,35	0,29	0	4,35	0,29	0	3,51	0,23	0	3,51	0,23	0
	10	4	0	0	0	29,41	2,44	0	0	0	0	0	0	0	0
		5	0	0	0	0	0	0	0	0	0	0	0	0	0
		6	0	0	0	0	0	0	0	0	0	0	0	0	0
		12	3	4,23	0,28	0	4,23	0,28	0	4,23	0,28	0	4,23	0,28	0
		4	10,00	0,67	0	10,00	1,10	0	0	0	0	0	0	0	0
	15	6	0	0	0	0	0	0	0	0	0	0	0	0	0
		7	0	0	0	0	0	0	0	0	0	0	0	0	0
		4	8,16	0,79	0	8,16	0,79	0	0	0	0	0	0	0	0
		5	0	0	0	0	0	0	0	2,78	0,19	0	2,78	0,19	0
		8	0	0	0	0	0	0	0	0	0	0	0	0	0
20	9	0	0	0	0	0	0	0	0	0	0	0	0	0	
	5	0	0	0	0	0	0	0	1,25	0,08	0	1,25	0,08	0	
	7	0	0	0	0	0	0	0	10,26	0,68	0	10,26	0,68	0	
	10	0	0	0	0	0	0	0	0	0	0	0	0	0	
	11	0	0	0	0	0	0	0	0	0	0	0	0	0	
25	7	6,58	0,60	0	6,58	0,60	0	1,59	0,11	0	0	0	0	0	
	9	0	0	0	0	0	0	0	7,50	0,50	0	0	0	0	
	13	0	0	0	0	0	0	0	0	0	0	0	0	0	
	14	0	0	0	0	0	0	0	0	0	0	0	0	0	
	8	8,97	1,03	0	8,97	1,03	0	3,03	0,20	0	3,03	0,20	0	0	
TC9	30	10	6,25	0,42	0	6,45	0,85	0	6,25	0,70	0	6,25	0,70	0	
		15	0	0	0	0	0	0	0	0	0	0	0	0	
		16	0	0	0	0	0	0	0	0	0	0	0	0	0
		8	2	0	0	0	0	0	0	4,58	0,31	0	4,58	0,31	0
		3	0	0	0	0	0	0	0	1,07	0,07	0	1,07	0,07	0
	10	4	6,11	0,41	0	6,11	0,41	0	0	0	0	0	0	0	0
		5	0	0	0	0	0	0	0	0	0	0	0	0	0
		3	0	0	0	0	0	0	0	5,26	0,66	0	5,26	0,66	0
		4	21,36	1,42	0	21,36	1,42	0	8,78	0,59	0	8,78	0,59	0	0
		5	0	0	0	0	0	0	0	0	0	0	0	0	0
	12	6	0	0	0	0	0	0	0	0	0	0	0	0	0
		3	20,53	1,43	0	20,53	1,43	0	29,14	3,37	0	29,14	3,37	0	0
		4	4,73	0,44	0	4,73	0,44	0	0	0	0	0	0	0	0
		6	5,91	0,39	0	5,91	0,39	0	5,91	0,39	0	5,91	0,39	0	0
		7	9,88	0,66	0	9,88	0,66	0	0	0	0	0	0	0	0
	15	4	8,16	0,54	0	8,16	0,54	0	0	0	0	0	0	0	0
		5	14,89	1,10	0	14,89	1,10	0	14,89	0,99	0	14,89	0,99	0	0
		8	4,11	0,27	0	4,11	0,27	0	0	0	0	0	0	0	0
		9	0	0	0	0	0	0	0	0	0	0	0	0	0
		5	7,28	0,66	0	7,28	0,66	0	19,09	1,27	0	19,09	1,27	0	0
20	7	14,40	1,55	0	14,40	1,55	0	0	0	0	0	0	0	0	
	10	1,46	0,10	0	1,46	0,10	0	0	0	0	0	0	0	0	
	11	2,17	0,14	0	2,17	0,14	0	2,17	0,14	0	2,17	0,14	0	0	
	7	9,75	1,04	0	9,75	1,04	0	0	0	0	0	0	0	0	
	9	6,94	0,47	0	6,94	0,47	0	0,08	0,01	0	0,08	0,01	0	0	

kontynuacja na następnej stronie



## D. Wyniki metody VNS z agregacją WOWA

### D.1. Wyniki metody VNS z agregacją WOWA dla małych problemów

Tabela D.1: Statystyki jakościowe rozwiązań problemów o małym rozmiarze (od 8 do 30 lokalizacji) ze zróżnicowanymi wagami zapotrzebowania

odstęp [%]			inicjalizacja zachłanna						inicjalizacja losowa						
Problem			<sup>w</sup> VNS			<sup>w</sup> VNS <sub>r</sub>			<sup>w</sup> VNS			<sup>w</sup> VNS <sub>r</sub>			
typ	m	n	max.	śr.	min.	max.	śr.	min.	max.	śr.	min.	max.	śr.	min.	
TC1	8	2	8,24	0,55	0	8,24	0,55	0	0	0	0	0	0	0	
		3	0	0	0	0	0	0	0	0	0	0	0	0	
		4	0	0	0	0	0	0	0	0	0	0	0	0	0
		5	0	0	0	0	0	0	0	0	0	0	0	0	0
	10	3	0	0	0	0	0	0	0	0	0	0	0	0	0
		4	0	0	0	0	0	0	0	0	0	0	0	0	0
		5	12,86	0,86	0	12,86	0,86	0	12,86	0,86	0	12,86	0,86	0	0
		6	0	0	0	0	0	0	0	0	0	0	0	0	0
	12	3	0	0	0	0	0	0	0	0	0	0	0	0	0
		4	1,02	0,07	0	1,02	0,07	0	3,06	0,20	0	3,06	0,20	0	0
		6	0	0	0	0	0	0	0	0	0	0	0	0	0
		7	3,61	0,24	0	3,61	0,24	0	0	0	0	0	0	0	0
	15	4	0	0	0	0	0	0	0	0	0	0	0	0	0
		5	13,43	0,98	0	13,43	0,98	0	0	0	0	0	0	0	0
		8	0	0	0	0	0	0	0	0	0	0	0	0	0
		9	0	0	0	0	0	0	0	0	0	0	0	0	0
	20	5	0	0	0	0	0	0	0	0	0	0	0	0	0
		7	7,19	0,48	0	7,19	0,48	0	0	0	0	0	0	0	0
		10	0	0	0	0	0	0	0	0	0	0	0	0	0
		11	0	0	0	0	0	0	0	0	0	0	0	0	0
25	7	0	0	0	0	0	0	0,80	0,05	0	0,80	0,05	0	0	
	9	2,99	0,31	0	2,99	0,31	0	1,19	0,11	0	1,19	0,11	0	0	
	13	0	0	0	0	0	0	0	0	0	0	0	0	0	
	14	0	0	0	0	0	0	0	0	0	0	0	0	0	
30	8	3,66	0,53	0	3,66	0,53	0	0,92	0,06	0	0,92	0,06	0	0	
	10	3,28	0,28	0	3,28	0,28	0	3,28	0,22	0	3,28	0,22	0	0	
	15	0	0	0	0	0	0	0	0	0	0	0	0	0	
	16	0	0	0	0	0	0	0	0	0	0	0	0	0	
TC2	8	2	15,17	1,01	0	15,17	1,01	0	29,41	2,76	0	0	0	0	
		3	0	0	0	0	0	0	26,95	2,76	0	26,95	2,76	0	
		4	0	0	0	0	0	0	26,04	1,74	0	26,04	1,74	0	
		5	0	0	0	0	0	0	0	0	0	0	0	0	
	10	3	0	0	0	0	0	0	5,05	0,41	0	5,05	0,41	0	
		4	0,34	0,02	0	0	0	0	0	0	0	0	0	0	0
		5	0	0	0	43,46	2,90	0	0	0	0	43,46	2,90	0	
		6	0	0	0	0	0	0	0	0	0	0	0	0	0
	12	3	13,92	2,27	0	13,92	2,86	0	12,79	0,85	0	12,79	0,85	0	

kontynuacja na następnej stronie

odstęp [%]			inicjalizacja zachłanna						inicjalizacja losowa						
Problem			<sup>w</sup> VNS			<sup>w</sup> VNS <sub>r</sub>			<sup>w</sup> VNS			<sup>w</sup> VNS <sub>r</sub>			
typ	m	n	max.	śr.	min.	max.	śr.	min.	max.	śr.	min.	max.	śr.	min.	
TC3	15	4	15,88	1,61	0	57,09	4,94	0	8,75	0,58	0	8,75	0,58	0	
		6	0	0	0	0	0	0	8,43	0,56	0	0	0	0	
		7	0	0	0	0	0	0	2,32	0,15	0	0	0	0	
	20	4	0	0	0	11,87	0,79	0	12,74	1,33	0	7,15	0,48	0	
		5	27,98	2,89	0	8,33	0,76	0	12,33	1,35	0	12,33	1,03	0	
		8	0	0	0	0	0	0	10,94	0,73	0	0	0	0	
	25	9	0	0	0	0	0	0	0	0	0	0	0	0	
		5	23,82	1,88	0	12,40	1,11	0	23,82	2,69	0	13,00	0,97	0	
		7	11,54	1,11	0	20,34	2,47	0	13,21	1,95	0	5,47	0,53	0	
	30	10	0	0	0	0	0	0	0	0	0	0	0	0	
		11	0	0	0	0	0	0	0	0	0	0	0	0	
		7	33,33	4,52	0	11,21	1,66	0	15,02	2,70	0	14,87	2,80	0	
	30	9	7,86	1,40	0	14,09	2,34	0	14,29	3,54	0	14,09	2,30	0	
		13	6,00	0,61	0	0	0	0	0	0	0	0	0	0	
		14	0	0	0	0	0	0	0	0	0	0	0	0	
	30	8	23,66	3,34	0	13,23	1,89	0	31,54	6,51	0	9,23	1,39	0	
		10	17,63	2,43	0	6,71	0,80	0	6,49	1,24	0	6,71	0,78	0	
		15	3,21	0,21	0	0	0	0	8,45	0,78	0	3,21	0,21	0	
	30	16	0	0	0	0	0	0	0	0	0	0	0	0	
		8	2	0	0	0	0	0	21,08	1,41	0	21,08	1,41	0	
		3	0	0	0	0	0	0	4,94	0,33	0	4,94	0,33	0	
	30	4	0	0	0	0	0	0	0	0	0	0	0	0	
		5	0	0	0	0	0	0	0	0	0	0	0	0	
		10	3	24,06	1,99	0	24,06	1,99	0	1,49	0,10	0	1,49	0,10	0
	30	4	1,64	0,11	0	1,64	0,11	0	6,86	0,46	0	6,86	0,46	0	
		5	0	0	0	0	0	0	0	0	0	0	0	0	
		6	0	0	0	0	0	0	0	0	0	0	0	0	
	30	12	3	12,55	0,84	0	12,55	0,84	0	12,60	0,84	0	12,60	0,84	0
		4	5,10	0,78	0	5,10	0,64	0	22,29	1,75	0	22,29	1,75	0	
		6	0	0	0	0	0	0	0	0	0	0	0	0	
	30	7	3,61	0,24	0	3,61	0,24	0	0	0	0	0	0	0	
		15	4	0	0	0	0	0	0	0	0	0	0	0	
		5	11,02	0,73	0	11,02	0,73	0	11,02	0,76	0	11,02	0,76	0	
	30	8	0	0	0	0	0	0	0	0	0	0	0	0	
		9	0	0	0	0	0	0	0	0	0	0	0	0	
		20	5	2,63	0,32	0	2,63	0,32	0	10,92	1,05	0	10,92	1,55	0
	30	7	0	0	0	0	0	0	3,93	0,26	0	3,93	0,26	0	
		10	0	0	0	0	0	0	0	0	0	0	0	0	
		11	0	0	0	0	0	0	0	0	0	0	0	0	
	30	25	7	10,07	0,72	0	0,68	0,05	0	7,09	0,52	0	7,09	0,47	0
		9	2,73	0,29	0	1,62	0,11	0	0	0	0	0	0	0	
		13	0	0	0	0	0	0	0	0	0	0	0	0	
	30	14	0	0	0	0	0	0	0	0	0	0	0	0	
		8	8,11	0,80	0	8,11	0,54	0	0	0	0	0	0	0	
		10	0,43	0,03	0	0,43	0,03	0	3,28	0,34	0	3,28	0,34	0	
	30	15	0	0	0	0	0	0	0	0	0	0	0	0	
		16	0	0	0	0	0	0	0	0	0	0	0	0	
		TC9	8	2,86	0,19	0	2,86	0,19	0	1,23	0,08	0	1,23	0,08	0
30	3	0	0	0	0	0	0	0	0	0	0	0	0		
	4	0	0	0	0	0	0	0	0	0	0	0	0		
	5	0	0	0	0	0	0	0	0	0	0	0	0		
30	10	3	0,35	0,02	0	0,35	0,02	0	0	0	0	0	0		
	4	0	0	0	0	0	0	0	0	0	0	0	0		
	5	12,68	0,85	0	12,68	0,85	0	12,68	0,85	0	12,68	0,85	0		
30	6	0	0	0	0	0	0	0	0	0	0	0			

kontynuacja na następnej stronie

odstep [%]			inicjalizacja zachłanna						inicjalizacja losowa						
Problem			<sup>w</sup> VNS			<sup>w</sup> VNS <sub>r</sub>			<sup>w</sup> VNS			<sup>w</sup> VNS <sub>r</sub>			
typ	m	n	max.	śr.	min.	max.	śr.	min.	max.	śr.	min.	max.	śr.	min.	
TC11	12	3	0	0	0	0	0	0	0	0	0	0	0	0	
		4	22,43	1,50	0	22,43	1,50	0	1,45	0,10	0	1,45	0,10	0	
		6	10,30	0,69	0	10,30	0,69	0	0	0	0	0	0	0	
	7	4	4,40	0,29	0	4,40	0,29	0	0	0	0	0	0	0	
		15	4	1,63	0,11	0	1,63	0,11	0	0	0	0	0	0	0
			5	15,59	1,29	0	15,59	1,29	0	0	0	0	0	0	0
	8		0	0	0	0	0	0	0	0	0	0	0	0	
	9	0	0	0	0	0	0	0	0	0	0	0	0	0	
		20	5	4,20	0,28	0	4,20	0,28	0	4,20	0,28	0	4,20	0,28	0
			7	5,18	0,35	0	5,18	0,35	0	4,26	0,28	0	4,26	0,28	0
	10		1,34	0,09	0	1,34	0,09	0	8,60	0,57	0	8,60	0,57	0	
	11	0	0	0	0	0	0	0	0	0	0	0	0	0	
		25	7	3,08	0,39	0	3,08	0,39	0	3,08	0,21	0	3,08	0,21	0
			9	10,36	0,69	0	10,36	0,69	0	3,78	0,25	0	3,78	0,25	0
	13		0	0	0	0	0	0	0	0	0	0	0	0	
	14	0	0	0	0	0	0	0	0	0	0	0	0	0	
		30	8	3,76	0,43	0	3,76	0,43	0	3,44	0,23	0	3,44	0,23	0
			10	2,45	0,19	0	2,45	0,19	0	2,45	0,16	0	2,45	0,16	0
	15		0	0	0	0	0	0	0	0	0	0	0	0	
	16	0	0	0	0	0	0	0	0	0	0	0	0	0	
		8	2	5,46	0,36	0	5,46	0,36	0	5,67	0,38	0	5,67	0,38	0
			3	0	0	0	0	0	0	0	0	0	0	0	0
	4		0	0	0	0	0	0	0	0	0	0	0	0	
	5	0	0	0	0	0	0	0	0	0	0	0	0	0	
		10	3	0	0	0	0	0	0	0	0	0	0	0	0
			4	0	0	0	0	0	0	0	0	0	0	0	0
	5		12,68	0,85	0	12,68	0,85	0	12,68	0,85	0	12,68	0,85	0	
	6	0	0	0	0	0	0	0	0	0	0	0	0	0	
		12	3	0	0	0	0	0	0	3,28	0,22	0	3,28	0,22	0
			4	0	0	0	0	0	0	1,49	0,10	0	1,49	0,10	0
	6		0	0	0	0	0	0	0	0	0	0	0	0	
	7	4,40	0,29	0	4,40	0,29	0	0	0	0	0	0	0	0	
		15	4	2,03	0,14	0	2,03	0,14	0	0	0	0	0	0	0
			5	15,46	1,32	0	15,46	1,32	0	0	0	0	0	0	0
	8		0	0	0	0	0	0	0	0	0	0	0	0	
	9	0	0	0	0	0	0	0	0	0	0	0	0	0	
		20	5	4,84	0,32	0	4,84	0,32	0	4,84	0,35	0	4,84	0,35	0
			7	7,78	0,52	0	7,78	0,52	0	1,80	0,12	0	1,80	0,12	0
	10		1,34	0,09	0	1,34	0,09	0	8,63	0,58	0	8,63	0,58	0	
	11	0	0	0	0	0	0	0	0	0	0	0	0	0	
		25	7	4,28	0,43	0	4,28	0,43	0	1,49	0,10	0	1,49	0,10	0
			9	0	0	0	0	0	0	3,77	0,45	0	3,77	0,45	0
	13		0	0	0	0	0	0	0	0	0	0	0	0	
	14	0	0	0	0	0	0	0	0	0	0	0	0	0	
		30	8	2,78	0,35	0	2,78	0,35	0	3,09	0,21	0	3,09	0,21	0
			10	0,60	0,07	0	0,60	0,07	0	0,48	0,03	0	0,48	0,03	0
	15		0	0	0	0	0	0	0	0	0	0	0	0	
	16	0	0	0	0	0	0	0	0	0	0	0	0	0	