POLITECHNIKA WARSZAWSKA

Wydział Matematyki
i Nauk Informacyjnych

ROZPRAWA DOKTORSKA

mgr inż. Michał Okulewicz

Zastosowanie populacyjnych metaheurystyk uwzględniających rozkład danych problemu do rozwiązywania problemu dynamicznej marszrutyzacji

Promotor
Prof. dr hab. inż. Jacek Mańdziuk

Warszawa, 2016

Mówienie jest wysiłkiem:
nie zdoła człowiek wyrazić wszystkiego słowami.
Koh 1, 8a

Rozprawę dedykuję moim dzieciom:
Magdzie, Asi i (oczekiwanemu) Pawełkowi.

Streszczenie

Rozprawa dotyczy możliwości zastosowania metaheurystycznych algorytmów optymalizacji ciągłej do rozwiązania problemu dynamicznej marszrutyzacji. Problem dynamicznej marszrutyzacji należy do grupy problemów transportowych, w której najbardziej znanym problemem jest problem komiwojażera. Celem problemu dynamicznej marszrutyzacji jest znalezienie takiego podziału zbioru zamówień pomiędzy pojazdy oraz kolejności realizacji zamówień przez te pojazdy, które zminimalizują sumaryczną trasę pojazdów. Dynamizm tego problemu polega na tym, że nowe zamówienia pojawiają się w trakcie procesu optymalizacji. W przeciwieństwie do stochastyczego problemu marszrutyzacji nie są znane ani ich przybliżona lokalizacja ani rozmiar.

Opisywane w literaturze podejścia optymalizujące problem dynamicznej marszrutyzacji traktują go jako ciąg zależnych od siebie statycznych instancji problemu. W obszarze dynamicznej marszrutyzacji były stosowane wyłącznie algorytmy wykorzystujące dyskretne przestrzenie rozwiązań, jednak w literaturze poświęconej statycznym wariantom problemu znajdowały się propozycje ciągłych reprezentacji takich problemów. Zaproponowany w tej rozprawie algorytm, nazwany ContDVRP, wykorzystuje kodowanie ciągłe, nie wymagające modyfikacji algorytmów optymalizacyjnych takich jak Optymalizacja Rojem Cząstek czy Ewolucja Różnicowa. Początkowy wariant kodowania był wprowadzony niezależnie od innych prac w obszarze marszrutyzacji, zaś w finalnej wersji wykorzystuje połączenie propozycji autora rozprawy z wariantami kodowań z prac Ai i Kachitvichyanukula.

Skuteczność proponowanego algorytmu jest weryfikowana eksperymentalnie na zestawie standardowo wykorzystywanych 21 instancji testowych, spopularyzowanych pracami Kilby'ego i Montemanniego. ContDVRP osiąga najwięcej najlepszych średnich wyników spośród podejść prezentowanych w literaturze. Sumaryczna poprawa średnich wyników jest osiągana w obu rodzajach eksperymentów: zarówno tych zatrzymujących proces optymalizacji po zadanej liczbie ewaluacji funkcji jakości, jak i tych wykorzystujących limit czasowy jako kryterium stopu. Oprócz eksperymentalnego potwierdzenia zasadności wykorzystania ciągłego kodowania, jest ono również w rozprawie szczegółowo analizowane.

Rozprawa prezentuje algorytm ContDVRP zarówno w kontekście badań nad optymalizacją problemów dynamicznych metodami metaheurystycznymi, jak i badań nad algorytmami optymalizującymi problemy marszrutyzacji. Rozprawa wykazuje eksperymentalnie,

że sposób kodowania problemu ma większy wpływ na jakość wyniku uzyskiwanego przez poszczególne algorytmy, niż sam wybór algorytmu wykorzystanego do optymalizacji. Rozprawa poszerza wiedzę na temat możliwych do zastosowania ciągłych kodowań i wynikającego z nich stosunku szybkości działania algorytmu do jakości uzyskiwanych wyników. Ponadto: 1) wprowadza alternatywny do standardowo stosowanego algorytmu zachłannego sposób inicjalizacji rozwiązań poprzez poszukiwanie skupisk zamówień, 2) wskazuje najistotniejsze z punktu widzenia jakości finalnego wyniku techniki optymalizacji wspomagające przetwarzanie problemów dynamicznych, 3) poszerza eksperymentalną wiedzę na temat wpływu zrównoleglenia procesu optymalizacji nie tylko na czas, ale i jakość osiąganych rozwiązań.

Ostatnie rozdziały rozprawy są poświęcone analizie dynamizmu problemu oraz wskazaniu przyczyn sukcesu podejść polegających na optymalizacji ciągu instancji statycznych, bez uwzględniania możliwej zmienności problemu. W tym zakresie rozprawa stawia tezę, że możliwa jest dalsza poprawa działania algorytmu ContDVRP w oparciu o estymację spodziewanej wielkości sumy zamówień na podstawie dostępnych danych problemu. Eksperymenty udowodniły słuszność tej tezy, a analiza ciągu kolejnych proponowanych wyników wykazała poprawę ich stabilności względem bazowych eksperymentów. Większa stabilność cząstkowych wyników dotyczyła mniejszej zmienności przypisań zamówień do pojazdów i liczby wykorzystywanych w rozwiązaniu pojazdów.

Abstract

This thesis presents the possibility of using continuous metaheuristic algorithms for solving Dynamic Vehicle Routing Problem (DVRP). DVRP belongs to a group of transportation problems, which most famous representative is a Traveling Salesman Problem (TSP). The goal of the DVRP is to find an assignment of requests to vehicles, and a route for each of the vehicles, which minimize the total route length. Dynamic nature of the DVRP comes from the fact that some of the requests become known only during the optimization process. DVRP differs from the Stochastic Vehicle Routing Problem (SVRP), in which the approximate location and volume of the requests are known.

Literature methods to solving DVRP process it as a sequence of dependent static Vehicle Routing Problem (VRP) instances. Although, continuous problem encoding has been utilized for the VRP, no such approach has been proposed for the DVRP. This thesis introduces such an algorithm, denoted ContDVRP, which by utilizing a continuous search space allows for a direct application of population based metaheuristics, like Particle Swarm Optimization (PSO) or Differential Evolution (DE). The initial encoding has been introduced independently from research on the VRP, while the final version integrates the author's approach with the methods of Ai and Kachitvichyanukul.

Effectiveness of the algorithm is tested on a set of 21 benchmark instances, converted from a static VRP instances by Kilby et al. and Montemanni et al. ContDVRP has obtained the largest number of significantly better average results than other algorithms presented in the literature. Best average result is achieved on two types of experiments: the one using a limit of fitness function evaluations, and the one using time limit to stop the optimization process. Besides experimental proof of abilities of continuous DVRP encoding its properties are thoroughly analyzed.

The thesis presents ContDVRP as both the example of a dynamic optimization algorithm and a method of solving vehicle routing problems. Thesis experimentally proves that the choice of the encoding is a more crucial factor than the choice of the optimization algorithm in terms of achieving a certain solution quality on a given instance. This thesis extends knowledge about the features on the continuous encodings and their computation time to quality ratio. Additionally: (1) it introduces solving capacitated clustering problem as an alternative method to greedy construction of initial candidate solutions, (2) points out the most important optimization techniques in terms of results quality, (3) extends

experimental knowledge on impact of parallel processing on computation time and quality of the achieved results.

Final chapters of this thesis are dedicated to the analysis of the DVRP dynamics and its features which allowed for a success of the approach of solving DVRP as a sequence of static instances (without explicit accounting for the growth of the problem during the process). The thesis states, and experimentally proves, that it is possible to improve the quality of solution by estimating the expected sum of requests' volume. The optimization process accounting for the expected growth of the number of requests shows a greater stability of the candidate solutions' sequences. That stability has been measured by the number of requests reassignments and changes in the number of needed vehicles.

Podziękowania

W pierwszej kolejności chciałbym podziękować mojemu promotorowi, profesorowi Jackowi Mańdziukowi, za zainteresowanie problemem oraz dalsze wyzwania naukowe i motywację do poprawiania osiągniętych rezultatów, a także duże wsparcie w przygotowywaniu i redakcji artykułów naukowych.

Dziękuję uczestnikom Seminarium Inteligencji Obliczeniowej na Wydziale Matematyki i Nauk Informacyjnych Politechniki Warszawskiej, zwłaszcza Jankowi Karwowskiemu, Karolowi Walędzikowi oraz Maćkowi Świechowskiemu, za ciekawe pytania i pomocne wskazówki.

Dziękuję wszystkim wykładowcom, z którymi miałem przyjemność mieć zajęcia w ciągu całego toku moich studiów. W szczególny sposób chciałbym podziękować doktorowi Konstantemu Junoszy-Szaniawskiemu i profesorowi Jarosławowi Arabasowi za wpływ jaki mieli na mój sposób patrzenia na problemy grafowe oraz zadania optymalizacji.

Dziękuję Wydziałowi Matematyki i Nauk Informacyjnych PW oraz Instytutowi Podstaw Informatyki PAN za wsparcie sprzętowe i edukacyjne prowadzonych badań. Dziękuję również Ośrodkowi Badań dla Biznesu oraz absolwentom Wydziału MiNI PW za umożliwienie zapoznania się z procesami biznesowymi i praktyką dużej firmy kurierskiej oraz wewnętrznego transportu materiałów budowlanych, co pozwoliło mi dokładniej zrozumieć możliwości i ograniczenia modelu rozważanego w tej rozprawie.

Przeprowadzenie badań przedstawionych w tej rozprawie nie byłoby możliwe w tak dużym zakresie, gdyby nie wsparcie finansowe w postacie stypendium doktoranckiego w ramach projektu POKL.04.01.01–00–051/10–00 *Technologie informacyjne: badania i ich interdyscyplinarne zastosowania* oraz grantu DEC-2012/07/B/ST6/01527 Narodowego Centrum Nauki.

Na koniec dziękuję swojej żonie Ani, za cierpliwość i przeczytanie wstępnych wersji tej rozprawy.

Spis treści

Skróty wykorzystywane w rozprawie

Skrót	Nazwa polska	Nazwa angielska
2MPSO	Dwufazowa wielorojowa optymalizacja rojem cząstek	Two-Phase Multi-Swarm Optimization
2PSO	Dwufazowa optymalizacja rojem cząstek	Two-Phase PSO
ACO	Algorytm mrówkowy	Ant Colony Optimization
ACS	Algorytm mrówkowy	Ant Colony System
CIL	Ciąg identyfikatorów lokalizacji	Location identifiers series
CIZ	Ciąg identyfikatorów zamówień	Requests identifiers series
CCP	Grupowanie z ograniczeniami	Capacitated Clustering Problem
ContDVRP	Optymalizator DVRP wykorzystujący ciągłe kodowanie	Continuous DVRP optimizer
DAPSO	Dynamiczna adaptująca się optymalizacja rojem cząstek	Dynamic Adapted PSO
DE	Ewolucja różnicowa	Differential Evolution
DVRP	Problem dynamicznej marszrutyzacji	Dynamic Vehicle Routing Problem
EH	Ewolucyjna hiperheurystyka	Evolutionary Hyperheuristic
GA	Algorytm genetyczny	Genetic Algorithm
LNS	Przeszukiwanie dużego sąsiedztwa	Large Neighbourhood Search
M-VRPDR	Algorytm memtyczny dla VRPwDR	Memetic Algorithm for VRPwDR
MAPSO	Adaptacyjna wielorojowa optymalizacja cząsteczkowa	Multi-Adaptive PSO
MEMSO	Heterogeniczny optymalizator wielorojowy	Multi-Environmental Multi-Swarm Optimizer

Skrót	Nazwa polska	Nazwa angielska
N_{TS}	Liczba kroków czasowych	Number of time slices
PSO	Optymalizacja rojem cząstek	Particle Swarm Optimization
T_{AC}	Bufor czasowy	Advanced commitment time
T_{CO}	Czas odcięcia	Cut–off time
TS	Przeszukiwanie z tabu	Tabu Search
TSP	Problem komiwojażera	Traveling Salesman Problem
VNS	Metoda zmiennego sąsiedztwa	Variable Neighbourhood Search
VRP	Problem marszrutyzacji	Vehicle Routing Problem
VRPwDR	Problem marszrutyzacji z dynamicznymi zamówieniami	VRP with Dynamic Requests
WCZ	Wektor centroidów skupisk zamówień	Requests clusters centers vector
WPP	Wektor przyporządkowań do pojazdów	Requests-to-vehicles assignment vector
WRZ	Wektor rang zamówień	Requests ranks vector

Wprowadzenie

Rozprawa poświęcona jest rozwiązywaniu problemu dynamicznej marszrutyzacji, rozpatrywanego jako dynamiczny problem optymalizacyjny rozwiązywany metodami metaheurystycznymi. Problemy transportowe (takie jak problem komiwojażera, problem chińskiego listonosza, problem marszrutyzacji czy problem kolejkowania zamówień) mają nie tylko interesujące własności matematyczne, ale również duże znaczenie praktyczne [37].

Statyczny problem marszrutyzacji, w wariancie poszukiwania sumarycznej optymalnej trasy do floty pojazdów o ustalonej wielkości, pojawił się w literaturze w 1959 roku, jako efekt prac praktycznych nad optymalizacją dostaw zapasów paliwa do stacji benzynowych [28]. Rozwój możliwości obliczeniowych, algorytmów aproksymacyjnych i heurystycznych wraz z rozwojem systemów informacji geograficznej, możliwości komunikacji bezprzewodowej i lokalizacji pojazdów w czasie rzeczywistym doprowadziły w 1988 roku do wyodrębnienia się problemu dynamicznej marszrutyzacji [102,103]. Problem został spopularyzowany dzięki wprowadzeniu do literatury (w 1998) zestawu problemów testowych, utworzonych z istniejących wcześniej instancji dla statycznej wersji systemu [59], oraz opublikowaniu prac (w 2005) poświęconych adaptacji algorytmu mrówkowego i adaptacyjnemu stochastycznemu losowemu przeszukiwaniu[1], prezentujących pierwsze referencyjne wyniki na tym zbiorze [75, 76].

W literaturze przedmiotu poświęconej problemom marszrutyzacji wiele prac skupia się na wprowadzaniu modeli mających za zadanie najlepiej odpowiadać potrzebom praktycznym [31, 105, 116, 121]. Kolejny istotny obszar badań stanowi dostosowywanie i usprawnianie istniejących heurystyk [23, 32, 78, 89], metaheurystyk [47, 58, 71, 84] oraz prezentacji wariantów problemów marszrutyzacji jako zagadnień programowania liniowego [132]. Osobny obszar prac stanowi również sama budowa systemów wspomagania decyzji i ich miejsce w procesach biznesowych związanych z planowaniem i zatwierdzaniem tras pojazdów [48–50, 118].

Równolegle z rozwojem tematyki problemów transportowych prowadzone były badania w drugim obszarze, w którym osadzona jest ta rozprawa, czyli rozwiązywaniu dynamicznych problemów optymalizacyjnych. Rozwój ten dotyczył klasyfikacji takich problemów i technik stosowanych do ich optymalizacji [26, 29, 130] oraz tworzenia pro-

[1] ang. *Greedy Randomized Adaptive Search Procedure* (GRASP)

blemów testowych, które pozwalałyby na weryfikację skuteczności poszczególnych metod [13, 65, 124, 126, 136].

Zgodnie z aktualnym stanem wiedzy w momencie rozpoczęcia badań przedstawionych w tej rozprawie, najskuteczniejszymi podejściami do problemu dynamicznej marszrutyzacji były: algorytm genetyczny operujący na kolejności zamówień do obsłużenia [47] oraz optymalizacja rojem cząstek, wykorzystująca przetwarzanie rozproszone, operująca na wektorach przypisań zamówień do pojazdów [55]. Oba te podejścia wykorzystywały dyskretne kodowanie problemu, co wymuszało konstrukcję specjalnego operatora krzyżowania w algorytmie genetycznym oraz dyskretyzację operatora aktualizacji położenia w optymalizacji rojem cząstek. Ponadto, w celu konstrukcji tras początkowych oraz ich aktualizacji przy zmianie stanu problemu, oba te podejścia korzystały wyłącznie z zachłannego wstawiania zamówień.

W obszarze problemów marszrutyzacji znane było podejście wykorzystujące optymalizację rojem cząstek, operującą w ciągłej przestrzeni przeszukiwań [1, 2]. W obszarze optymalizacji oraz problemów dynamicznych zidentyfikowane były: istotność wpływu kodowania na dynamizm problemu i osiągane wyniki [15, 110] oraz rozbieżność między praktycznymi problemami dyskretnymi a ciągłymi instancjami testowymi wykorzystywanymi do porównywania algorytmów [65, 108, 109, 136].

Cele badawcze

Badania przedstawione w tej rozprawie przebiegały w dwóch etapach. Celem pierwszego etapu było zaprojektowanie algorytmu rozwiązującego problem dynamicznej marszrutyzacji za pomocą standardowej wersji optymalizacji rojem cząstek. Celem drugiego etapu była szczegółowa analiza wyników i powiązanie ich z cechami poszczególnych instancji testowych. Cele badawcze pierwszego etapu można podsumować w następujący sposób:

- zaproponowanie takiego kodowania problemu marszrutyzacji, które umożliwi stosowanie algorytmów optymalizacji ciągłej bez ich modyfikacji i tworzenia specjalnych operatorów,
- wprowadzenie szybkiego algorytmu heurystycznego konstruującego trasy początkowe, który operuje na zamówieniach w sposób globalny,
- **poprawa średnich wyników dla testowych instancji problemu względem wyników literaturowych**.

Po zrealizowaniu powyższych celów badawczych zostały sformułowane kolejne cele, które można podzielić na 3 grupy i podsumować w następujący sposób:

1. Analiza zaproponowanego algorytmu:
    - **wskazanie komponentów zaproponowanego algorytmu, które mają największy wpływ na jakość osiąganych wyników**,
    - weryfikacja zasadności wykorzystania przetwarzania równoległego,

- zbadanie zależności pomiędzy wariantem zastosowanego kodowania problemu a jakością wyników i czasem optymalizacji.

2. Analiza dynamiki problemu dynamicznej marszrutyzacji i jej konsekwencji:
   - zmierzenie stopnia dynamizmu problemu dynamicznej marszrutyzacji oraz stabilności wskazywanych rozwiązań,
   - wyjaśnienie przyczyn skuteczności podejść wykorzystujących jedynie przekazywanie poprzednich rozwiązań jako techniki poprawiającej jakość optymalizacji problemów dynamicznych,
   - wskazanie przyczyn osiągania dobrych wyników, pomimo znajdowania dla pośrednich stanów problemu optimów odległych od tego znajdowanego dla końcowego stanu problemu,
   - **zaproponowanie metody wykorzystującej znajomość dynamiki problemu do poprawy osiąganych wyników**.
3. Analiza cech problemu marszrutyzacji i jej konsekwencji:
   - wskazanie powiązania między algorytmem a osiąganymi wynikami dla danych własności instancji problemu,
   - pokazanie silnego związku między typem kodowania problemu a wynikami osiąganymi przez algorytm, który operuje na rozwiązaniach z przestrzeni przeszukiwania indukowanej tym typem kodowania,
   - **zaproponowanie metody wykorzystującej cechy problemu do wyboru algorytmu optymalizacyjnego, właściwego dla danego rodzaju problemu**.

Hipotezy badawcze

Rozprawa stawia 3 hipotezy, które są eksperymentalnie weryfikowane poprzez realizację założonych celów badawczych:

H1. Możliwe jest zaproponowanie ciągłego kodowania problemu (dynamicznej) marszrutyzacji, które umożliwi zastosowanie dowolnych populacyjnych algorytmów optymalizacji ciągłej i osiągnięcie dobrej jakości wyników.

H2. Możliwa jest poprawa jakości osiąganych wyników względem podstawowego proponowanego podejścia i wyników literaturowych oraz zwiększenie stabilności ciągu rozwiązań uzyskiwanych dla cząstkowych stanów problemu, poprzez predykcję końcowego stanu zadania dynamicznej marszrutyzacji w oparciu o aktualnie dostępne dane.

H3. Możliwy jest wybór, spośród puli różnych algorytmów, tego z podejść, które ma największą szansę na osiągnięcie najlepszego wyniku dla danej instancji problemu na podstawie danych o początkowo dostępnych zamówieniach.

Opublikowane częściowe wyniki

Część wyników zaprezentowanych w tej rozprawie została wcześniej opublikowane w pracach współautorskich autora i promotora rozprawy.

Pierwszy artykuł [83] wprowadza ciągłe kodowanie przypisania zamówień do pojazdu oraz kolejności zamówień w pojeździe, pierwszą wersję funkcji jakości opartą na analizie skupień zamówień i prezentuje wyniki proponowanego algorytmu ContDVRP (oznaczanego pierwotnie jako 2PSO) na przykładzie algorytmu PSO. Kolejny artykuł [84] rozszerza kodowanie przypisania zamówień do pojazdu, opiera stosowaną funkcję jakości o długości tras aproksymowane algorytmem 2–OPT oraz osadza ContDVRP (oznaczany w kolejnej wersji jako 2MPSO) w środowisku równoległych procesów optymalizacyjnych. Rozprawa poszerza analizę komponentów algorytmu oraz wprowadza kilka jego modyfikacji, skutkujących dalszą poprawą osiąganych wyników. Trzeci artykuł [85] prezentuje heurystykę podziału zamówień pomiędzy pojazdy, opartą na algorytmie Kruskala znajdującym minimalne drzewo rozpinające. Artykuł ten wprowadza również pierwotną wersję metody wykorzystującej założenia dotyczące dynamiki problemu do poprawy jakości wyników. Czwarty artykuł [87] pokazuje różnice w działaniu algorytmów wynikające z zastosowania różnego rodzaju kodowań. Artykuł ten wprowadza również metodę hiperheurystyczną, w której w oparciu o wyniki algorytmów i skwantyfikowaną charakterystykę instancji problemu budowany jest model liniowy decydujący o wyborze algorytmu, który optymalizuje daną instancję. Piąty artykuł [86], znajdujący się obecnie w recenzji[2] , analizuje wpływ poszczególnych komponentów ContDVRP na osiągane wyniki. W przygotowaniu jest również artykuł [3] , który prezentuje ogólność ciągłego kodowania względem algorytmu optymalizacyjnego oraz metodę poprawy wyników poprzez uwzględnienie estymacji finalnej sumy rozmiarów zamówień [88].

Układ rozprawy

Rozprawa jest podzielona na dwie główne części. **Część I** stanowi ogólne wprowadzenie do tematyki optymalizacji, ze szczególnym uwzględnieniem optymalizacji dynamicznej. **Część II** koncentruje się na samym problemie dynamicznej marszrutyzacji, osadzając go zarówno w kontekście innych problemów transportowych jak i omówionych w części I problemów dynamicznych.

Poniżej przedstawiony jest szczegółowy układ rozprawy.

Rozdział 1 przedstawia klasyfikację problemów optymalizacyjnych. Szczególna uwaga poświęcona jest oddzieleniu dyskretnego lub ciągłego typu problemu od stosowanego typu

[2]Treść artykułu jest dostępna pod adresem:
`http://www.mini.pw.edu.pl/ okulewiczm/downloads/badania/ASOC_OptimizationModulesDVRP.pdf`

[3]Treść artykułu jest dostępna pod adresem:
`http://www.mini.pw.edu.pl/ okulewiczm/downloads/badania/EvCo_ContinuousDVRP.pdf`

przestrzeni przeszukiwań, w której pracuje algorytm optymalizacyjny. Ponadto, w celu późniejszego sklasyfikowania problemu dynamicznej marszrutyzacji, rozdzielane są pojęcia dostępności danych oraz celu optymalizacji.

Rozdział 2 prezentuje przegląd literatury w zakresie technik wykorzystywanych w optymalizacji problemów dynamicznych. Główną jego część stanowi omówienie modyfikacji metaheurystyk populacyjnych stosowanych do problemów statycznych (zwłaszcza metod ewolucyjnych), które poprawiają jakość ich działania dla problemów dynamicznych. Ponadto omawiane są metody optymalizacji odpornej, jako rozszerzanie metod programowania liniowego czy kwadratowego dla problemów dynamicznych.

Rozdział 3 wprowadza problem dynamicznej marszrutyzacji, prezentuje jego model matematyczny oraz powiązane z nim parametry operacyjne. **Rozdział 4** prezentuje najpopularniejsze algorytmy heurystyczne, wykorzystywane jako pomocnicze operatory przeszukiwania przestrzeni rozwiązań, poprawiające wyniki wskazywane przez metody metaheurystyczne. Szczegółowo omówione są metoda 2–OPT oraz modyfikacja algorytmu Kruskala, wykorzystywane w algorytmie ContDVRP. **Rozdział 5** porządkuje rodzaje przestrzeni przeszukiwań wykorzystywanych w metodach rozwiązujących problem dynamicznej marszrutyzacji. Porównywane są rozmiary przestrzeni przeszukiwań oraz zalety i wady różnych reprezentacji. **Rozdział 6** opisuje populacyjne algorytmy optymalizacyjne oraz sposoby ich zastosowania w problemie dynamicznej marszrutyzacji. Zaprezentowana jest również podstawowa technika wykorzystywana w tych algorytmach dla poprawy wyników w problemie dynamicznym, czyli transfer poprzednich rozwiązań.

Rozdział 7 prezentuje szczegółową analizę cech instancji problemów znajdujących się w zbiorze testowym. Na podstawie tej analizy prezentowane są metoda hiperheurystyczna oraz metoda predykcyjna. Metoda hiperheurystyczna ma za zadanie estymować względną jakość rozwiązań zwracanych przez różne algorytmy dla zadanego typu problemu i na tej podstawie decyduje o wyborze algorytmu przed rozpoczęciem procesu optymalizacji. Metoda predykcyjna ma za zadanie estymować spodziewaną sumę wielkości zamówień i na tej podstawie kierować algorytm w stronę rozwiązań, które podlegają mniejszym zmianom wraz z prezentacją kolejnych zamówień.

Rozdział 8 prezentuje proponowaną metodę ContDVRP w sposób całościowy. W opisie uwzględniony jest zarówno sposób wykorzystania omówionych wcześniej algorytmów heurystycznych i metaheurystycznych, ich parametryzacja, jak i wskazówki implementacyjne. **Podrozdział 8.4** prezentuje najistotniejsze wyniki numeryczne oraz ich analizę.

Rozdział 9 zawiera podsumowanie oraz prezentuje potencjalne dalsze kierunki badań w tym obszarze. **Dodatek A** prezentuje wyniki dostrajania parametrów operacyjnych problemu oraz algorytmu ContDVRP. **Dodatek B** zawiera wizualizacje i definicje tras najlepszych rozwiązań znalezionych przez ContDVRP. Sześć spośród tych rozwiązań, to wciąż najlepsze znane wyniki literaturowe, natomiast trzynaście innych ustępuje jedynie ostatnio opublikowanym wynikom algorytmu memetycznego.

Część I

Wprowadzenie do dynamicznych problemów optymalizacyjnych

Rozdział 1

Kategorie problemów optymalizacyjnych

W tym rozdziale przedstawiony jest sposób kategoryzacji problemów optymalizacyjnych w celu ujednolicenia pojęć występujących w dalszej części rozprawy. Najpierw zaprezentowane są obecne w literaturze kategoryzacje dynamicznych problemów optymalizacyjnych. Następnie pojęcia dotyczące kategoryzacji są zebrane w spójną całość i uzupełnione niezbędnymi modyfikacjami pozwalającymi prawidłowo opisać, omawiany w głównej części rozprawy, problem dynamicznej marszrutyzacji.

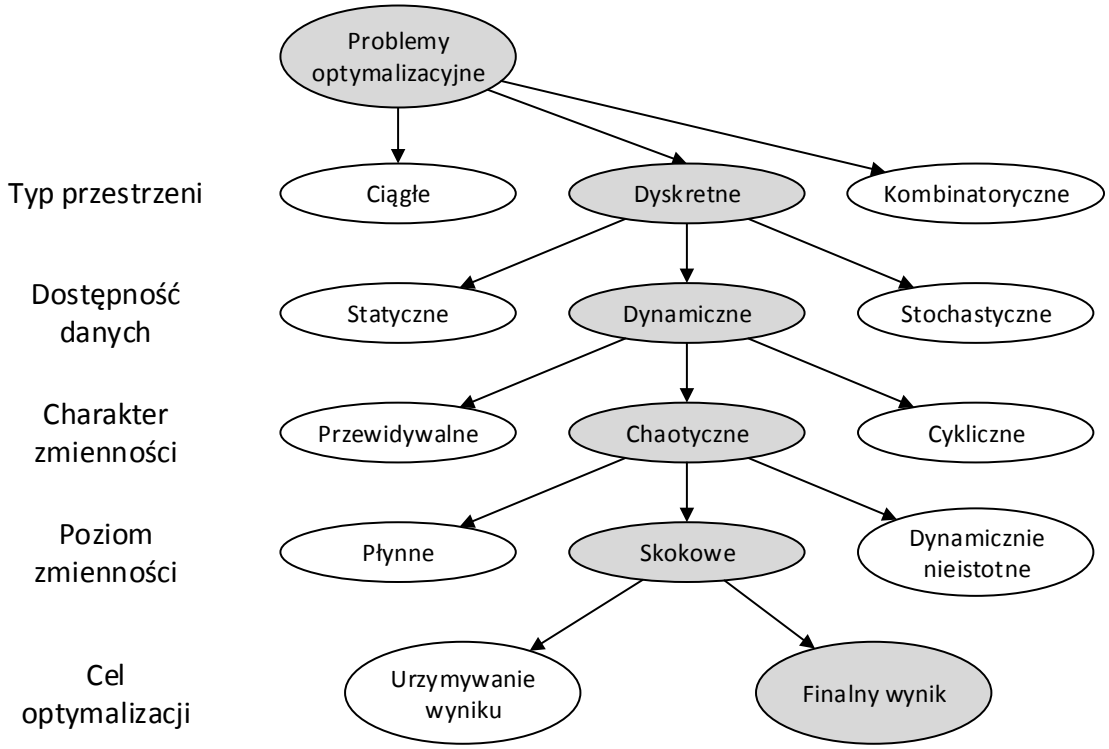
1.1 Podstawowe definicje

Na potrzeby prezentacji technik wykorzystywanych w optymalizacji problemów dynamicznych oraz sklasyfikowania problemu dynamicznej marszrutyzacji, można podzielić problemy optymalizacyjne ze względu na następujące cechy:

- typ przestrzeni rozwiązań problemu,
- poziom dostępności danych problemu,
- poziom i charakter zmienności problemu,
- cel optymalizacji.

Rysunek 1.1 prezentuje fragment klasyfikacji problemów optymalizacyjnych skoncentrowany na skategoryzowaniu problemu dynamicznej marszrutyzacji, reprezentowanego w naturalnej dyskretnej przestrzeni (zaprezentowanej w podrozdziale 5.1.1).

Pierwszą pracą porządkującą podział dynamicznych problemów optymalizacyjnych była praca De Jonga [29]. Klasyfikacja De Jonga wyróżnia 4 typy dynamicznych problemów optymalizacyjnych: z płynnie przemieszczającym się optimum, z zanikającymi i pojawiającymi się optimami, z cyklicznymi zmianami oraz z gwałtownymi zmianami. Późniejsza rozprawa doktorska Weickera [130] uściśla klasyfikację problemów w kontekście sposobu dokonywania zmian w funkcji (poprzez obroty, translacje, skalowanie) i opisuje



Rysunek 1.1: Fragment klasyfikacji problemów optymalizacyjnych skoncentrowany na skategoryzowaniu problemu dynamicznej marszrutyzacji. Szarym kolorem oznaczona jest klasyfikacja tego problemu, reprezentowanego w **dyskretnej** przestrzeni przeszukiwań.

dynamizm tych zmian również ilościowo. Obie powyższe klasyfikacje (De Jonga i Weickera) bezpośrednio dotyczyły jedynie problemów ciągłych. Odmienną klasyfikację proponują Jin i Branke [52], którzy koncentrując się na podziale ze względu na sposób obliczania wartości funkcji jakości, dzielą problemy na: zaszumione (wartość funkcji jakości pochodzi z pewnego rozkładu i jej wartość oczekiwana w danym punkcie może być szacowana przez wielokrotne próbkowanie), odporne (parametry modelu podlegają niepewnościom, ale po ustaleniu teoretycznej wartości optimum), przybliżone (funkcja jakości jest zbyt kosztowna do bezpośredniego obliczenia i musi być zastąpiona jej estymacją) i zmienne w czasie (wartość funkcji jakości jest w każdym momencie znana i dokładnie określona, ale podlega zmianom w czasie). Kolejne klasyfikacje problemów dynamicznych można odnaleźć w pracach związanych z konstrukcją generatorów problemów dynamicznych [65, 136] oraz pracach przeglądowych [26, 82].

Podstawowe definicje dotyczące optymalizacji są podane w oparciu o rozprawę doktorską Weickera [130] poświęconą wykorzystaniu algorytmów ewolucyjnych do optymalizacji problemów dynamicznych.

Definicja 1. ***Problem optymalizacyjny*** $\mathcal{P} = (\Omega, f, \succ)$ *jest zdefiniowany przez przestrzeń rozwiązań* Ω*, funkcję jakości*[1] $f : \Omega \to \mathbb{R}$ *oraz relację* $\succ \in \{<, >\}$*. Rozwiązaniem problemu jest znalezienie zbioru elementów (elementu)* $\mathcal{X} \subseteq \Omega$ *spełniającego następujące warunki:*

$$\mathcal{X} = \{x \in \Omega : \forall_{x' \in \Omega} f(x) \succeq f(x')\}$$

Należy zaznaczyć, że rozprawa dotyczy wyłącznie parametrycznych problemów optymalizacyjnych, tj. takich w których przestrzeń rozwiązań jest "iloczynem kartezjańskim zbiorów wartości zmiennych niezależnych"(strona 30 z [5]).

Definicja 2. ***Podzbiorem dopuszczalnym*** *zbioru* Ω *jest zbiór* $\mathcal{D}$ *spełniający następujące warunki:*

$$\mathcal{D} = \{x \in \Omega : f(x) \text{ jest skończona}\}$$

Definicja 3. ***Przestrzenią przeszukiwań*** $\mathcal{G}$ *problemu* $\mathcal{P}$ *jest zbiór, na którym zdefiniowana jest operacja* $dec : \mathcal{G} \to \Omega$ *transformująca element przestrzeni przeszukiwań* $g \in \mathcal{G}$ *w element przestrzeni rozwiązań* $x \in \Omega$*.*

Przykład 1. *W algorytmie genetycznym elementy przestrzeni przeszukiwań* $\mathcal{G}$ *określane są mianem gentotypów, zaś elementy przestrzeni rozwiązań* Ω *określane są mianem fenotypów.*

Definicja 4. ***Algorytm optymalizacyjny*** $\mathcal{A}$ *to zbiór operatorów przeszukiwania* $\mathcal{O}$ *pracujących na elementach przestrzeni przeszukiwań* $\mathcal{G}$ *oraz wykorzystujący złożenie* $f \circ dec : \mathcal{G} \to \mathbb{R}$ *do ewaluacji elementów tej przestrzeni i zwracający zbiór* $\mathcal{X}^*$*, będący aproksymacją poszukiwanego zbioru* $\mathcal{X}$ *o następujących cechach:*

$$\mathcal{X}^* = \{x \in \Omega_A : \forall_{x' \in \Omega_A} f(x) \succeq f(x')\}$$

Ω_A *jest podzbiorem przestrzeni rozwiązań przeliczonych przez algorytm* $\mathcal{A}$ *w trakcie jego działania. Operator* $O \in \mathcal{O}$ *jest postaci* $O : \mathcal{P} \times \mathcal{G}^{n_0} \to \mathcal{G}^{n_1}$*, gdzie* $n_0 \in \mathbb{N}_0$*,* $n_1 \in \mathbb{N}_+$*.*

Przykład 2. *Przykładem operatora (ozn.* I*) korzystającego wyłącznie z danych problemu i generującego element przestrzeni przeszukiwań* $I : P \to \mathcal{G}$ *są wszelkiego rodzaju inicjalizacje rozwiązań oraz algorytmy jednokrokowe.*

Przykład 3. *Przykładem operatora (ozn.* M*) korzystającego z pojedynczego elementu przestrzeni przeszukiwań i zwracającego jeden taki element* $M : \mathcal{G} \to \mathcal{G}$ *jest mutacja w algorytmie genetycznym.*

Przykład 4. *Przykładami operatorów wykorzystujących wiele elementów przestrzeni przeszukiwań i zwracających jeden lub wiele elementów jest krzyżowanie wymieniające (ozn.* C*) w algorytmie genetycznym* $C : \mathcal{G} \times \mathcal{G} \to \mathcal{G} \times \mathcal{G}$*, czy zmiana prędkości (ozn.* Δ_V*) w optymalizacji rojem cząstek* $\Delta_V : \mathcal{G} \times \mathcal{G} \times \mathcal{G} \to \mathcal{G}$*.*

[1] W zależności od dziedziny funkcja jakości może być też określana jako funkcja celu lub funkcja przystosowania.

1.2 Przestrzeń rozwiązań problemu

Za podręcznikiem [5] parametryczne problemy optymalizacyjne można skategoryzować jako **ciągłe**, **dyskretne** i **kombinatoryczne**.

Definicja 5. ***Problem ciągły*** *charakteryzuje się tym, że jego rozwiązanie jest reprezentowane w formie wektora liczb rzeczywistych ($\Omega = \mathbb{R}^n$).*

Definicja 6. ***Problem dyskretny*** *charakteryzuje się tym, że jego rozwiązanie jest reprezentowane w formie wektora liczb całkowitych ($\Omega = \mathbb{Z}^n$).*

Definicja 7. ***Problem kombinatoryczny*** *charakteryzuje się tym, że jego rozwiązanie jest reprezentowane w formie wektora binarnego ($\Omega = \mathbb{Z}_2^n$).*

Należy przy tym zauważyć, że typ przestrzeni przeszukiwań nie musi być zgodny z typem przestrzeni rozwiązań (co jest wykorzystywane w głównej części rozprawy). Trywialnymi przykładami takich transformacji jest przeszukiwanie w przestrzeni ciągłej, a następnie zaokrąglanie wartości argumentów do liczb całkowitych lub zdefiniowanie *a priori* wartości dla każdego z ciągłych parametrów, a następnie przeszukiwanie ich iloczynu kartezjańskiego w poszukiwaniu optimum (np. poprzez *grid search*).

1.3 Poziom dostępności danych problemu

Dla zdefiniowania kategorii dostępności danych w problemach optymalizacyjnych wykorzystane są pojęcia proponowane w pracy przeglądowej dotyczącej problemów marszrutyzacji [94]. Pomimo skoncentrowania tego artykułu na konkretnej grupie problemów, same pojęcia wykorzystane w tej pracy można uogólnić na inne rodzaje zagadnień optymalizacyjnych.

Ze względu na poziom dostępności danych można problemy podzielić na **statyczne**, **stochastyczne** i **dynamiczne**. Należy przy tym zauważyć, że chociaż standardowo w literaturze cel optymalizacji jest powiązany z typem dostępności danych problemu, jednak w celu skategoryzowania omawianego w głównej części rozprawy problemu dynamicznej marszrutyzacji należy rozdzielić te kategorie.

Definicja 8. ***Statycznym*** *problemem optymalizacyjnym nazywamy taki, w którym wszystkie dane niezbędne do wskazania wartości rozwiązania względem celu optymalizacji są dostępne, a więc funkcja jakości f pozostaje niezmienna podczas całego procesu optymalizacji.*

Definicja 9. ***Stochastycznym*** *problemem optymalizacyjnym nazywamy taki, w którym zwracane wartości funkcji jakości są niepewne zarówno ze względu na możliwe zmiany środowiska jak i możliwą niedokładną realizację optymalizowanych parametrów.*

Otrzymywane wartości funkcji jakości możemy wyrazić jako pochodzące z pewnego, potencjalnie zmiennego w czasie t, rozkładu $F(x,t)$:

$$f(x) \sim F(x,t)$$

Dla niektórych typów problemów rozkład $F(x,t)$ jest znany i można wykorzystać jako estymację wartości $\hat{f}(x)$ wartość oczekiwaną tego rozkładu:

$$\hat{f}(x) = E[F(x,t)]$$

W celu zdefiniowania problemu dynamicznego ponownie jest wykorzystana nomenklatura z rozprawy [130], uzupełniona spostrzeżeniami z prac [26, 136] dotyczących nie tylko zmienności wartości funkcji w czasie, ale również dziedziny argumentów i zbioru rozwiązań dopuszczalnych.

Definicja 10. ***Dynamicznym*** *problemem optymalizacyjnym $\mathcal{P}(t) = (\Omega(t), f, \succ)$ nazywamy taki, w którym wartość funkcji jakości $f : \Omega(t) \times \mathbb{R} \to \mathbb{R}$ zależna jest zarówno od wartości parametrów x, jak i czasu t. Zmianom może podlegać przestrzeń rozwiązań $\Omega(t)$ oraz zbiór rozwiązań dopuszczalnych $\mathcal{D}(t) \subseteq \Omega(t)$.*

W pracy [82] wprowadzony jest dodatkowy (intuicyjny) wyróżnik problemu dynamicznego. Problem jest uznawany za dynamiczny, gdy decyzje są podejmowane w trakcie zmian zachodzących w problemie i działania algorytmu (w przeciwnym przypadku jest to jedynie problem z zależnością czasową). Dokładniejsze definicje w tym zakresie oraz szersza definicja problemu zmiennego w czasie (uwzględniającego zmiany przestrzeni rozwiązań w czasie) przedstawione są w rozprawie doktorskiej Nguyena [81].

1.4 Poziom i charakter zmienności problemu

Charakterystyka typów zmienności problemu jest wykonana w oparciu o prace [16, 29, 136]. Warto też wspomnieć, że zbliżone charakterystyki, dla problemów zmiennych w czasie (stochastycznych i dynamicznych), funkcjonują również w dziedzinach innych niż optymalizacja (np. w problemie klasyfikacji [17, 18]).

Problemy zmienne w czasie, ze względu na charakter zmienności, można podzielić na **cykliczne**, **przewidywalne** oraz **chaotyczne**. Względem poziomu zmienności można takie problemy podzielić na **płynne** oraz **skokowe**.

Definicja 11. *Problem o **zmienności cyklicznej** charakteryzuje się okresowym powrotem do identycznych stanów. Funkcja jakości takiego problemu spełnia warunek:*

$$\exists_{T \in \mathbb{R}} \forall_{x \in \Omega} f(x, t+T) = f(x,t)$$

Definicja 12. *Problem o* ***przewidywalnej zmienności*** *charakteryzuje się istnieniem zależności późniejszych stanów problemu od stanów wcześniejszych. Funkcja jakości takiego problemu spełnia warunek:*

$$\exists_{T_0 \in \mathbb{R}} \forall_{x \in \Omega, t > T_0} f(x,t) = F(x, X, t_0, t_1, \ldots, t_{-1})$$

Definicja 13. *Problem o* ***zmienności chaotycznej*** *nie gwarantuje powrotu do istniejącego wcześniej stanu, ani przewidywalności kolejnych stanów co można wyrazić następującymi zależnościami:*

$$\exists_{x \in \Omega} \neg \exists_{T \in \mathbb{R}} f(x, t+T) = f(x,t)$$

$$\exists_{x \in \Omega} \neg \exists_{T_0 \in \mathbb{R}} \forall_{t > T_0} f(x,t) = F(x, X, t_0, t_1, \ldots, t_{-1})$$

Definicja 14. *Problem o* ***płynnej zmienności*** *charakteryzuje się tym, że funkcja jakości problemu dla każdego zestawu parametrów jest ciągła w czasie.*

$$\forall_{x \in \Omega} f(x,t) \in C_1$$

Definicja 15. *Problem o* ***zmienności skokowej*** *charakteryzuje się tym, że funkcja jakości dla pewnego zestawu parametrów nie jest funkcją ciągłą.*

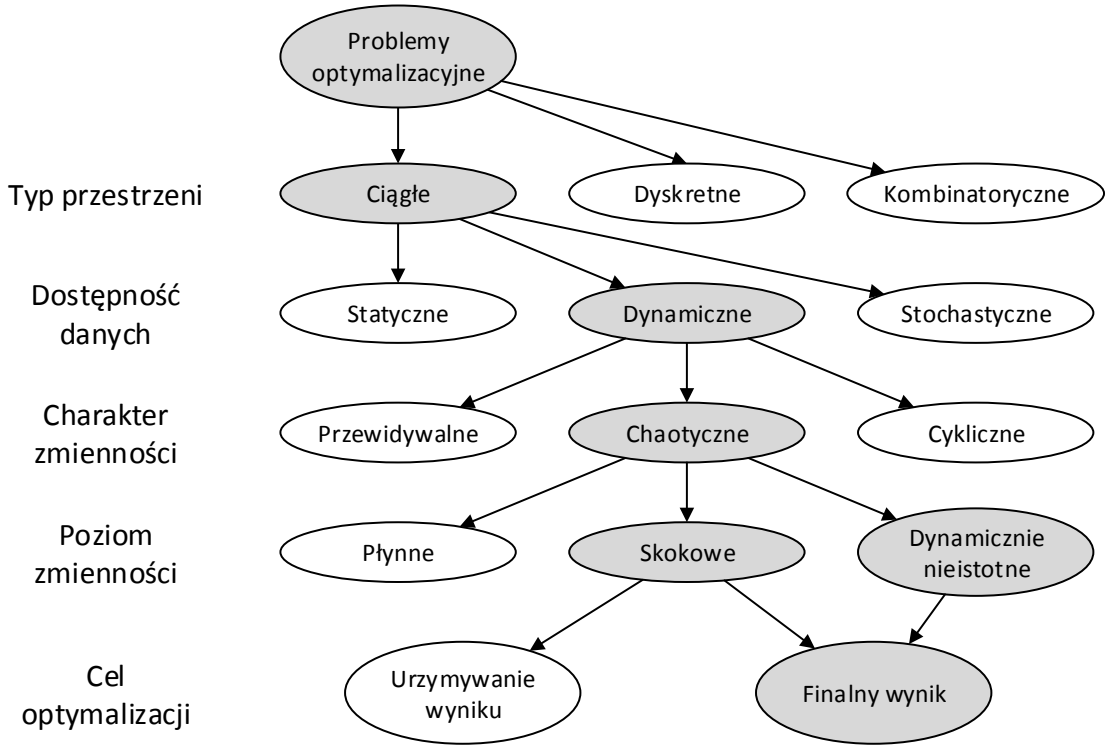
$$\exists_{x \in \Omega} f(x,t) \notin C_1$$

Praca [136] wprowadza dodatkowo pojęcie dynamicznie istotnej zmiany. Wprowadzenie tego pojęcia motywowane jest faktem, że powyższa charakterystyka dynamiki zmian jest niewystarczająca dla skutecznego opisywania problemów optymalizacyjnych z punktu widzenia algorytmów optymalizacyjnych, skoro nie uwzględnia wpływu tej dynamiki na optymalne rozwiązanie.

Definicja 16. ***Dynamicznie istotna zmiana*** *funkcji jakości to taka zmiana środowiska, która powoduje zmianę lokalizacji optimum globalnego.*

Poza jakościową charakteryzacją zmienności problemu, w literaturze proponowane są wskaźniki definiujące charakterystykę ilościową tej zmienności. Kompleksowa propozycja w tym aspekcie jest przedstawiona w pracy [16], gdzie zaprezentowano następujące mierniki zmian (przy założeniu istnienia pojedynczego optimum globalnego lub wyborze losowego spośród takich optimów):

- istotność zmiany (odległość pomiędzy optimum przed i po zmianie),
- estymowana istotność zmiany (odległość pomiędzy najlepszym znanym rozwiązaniem przed i po zmianie),
- korelacja wartości funkcji przystosowania przed i po zmianie,



Rysunek 1.2: Fragment klasyfikacji problemów optymalizacyjnych skoncentrowany na skategoryzowaniu problemu dynamicznej marszrutyzacji. Szarym kolorem oznaczona jest klasyfikacja tego problemu, reprezentowanego w **ciągłej** przestrzeni przeszukiwań.

- korelacja wartości funkcji przystosowania przed i po zmianie po zastosowaniu algorytmu *hill-climbing*,
- korelacja wartości funkcji przystosowania przed i po zmianie dla bliskich punktów przestrzeni przeszukiwań,
- różnica wartości pomiędzy wynikami uzyskanymi przez podejście *hill-climbing* rozpoczynające przeszukiwanie z optimum poprzedniej fazy a rozpoczynające obliczenia z losowo wybranego punktu przestrzeni,
- estymowana wartość poprzednich optimów (wartość osiągana przez *hill-climbing* uruchomiony z najlepszej próbki z przestrzeni).

Chociaż dwa pierwsze z tych wskaźników mają bezpośrednie zastosowanie jedynie w sytuacji, gdy rozmiar przestrzeni rozwiązań nie zmienia się w trakcie zmiany środowiska, jednak można je zmodyfikować również do sytuacji przestrzeni rozwiązań (lub przeszukiwań) o zmiennym rozmiarze. Jedna z możliwych modyfikacji polega na porównaniu wyłącznie tych zmiennych, które występują przed i po zmianie stanu. Inna modyfikacja może polegać na porównaniu położenia optimum z poprzedniego stanu zaadaptowanego szybkim algorytmem konstrukcyjnym do aktualnych warunków. Modyfikacja tego pierwszego typu jest wykorzystywana w tej rozprawie i zaprezentowana w podrozdziale 5.3.

Istotność pomiaru różnicy między kolejnymi rozwiązaniami jest też dyskutowana w pracach Rohlfshagena i Yao [108, 109]. Prace te wskazują na istotność rozpatrywania problemu dynamicznego łącznie z charakterystyką zachodzących w nim zmian, nie jedynie jako zmienną w czasie wersję problemu statycznego. Tego rodzaju całościowe podejście jest jedną z motywacji wykorzystywania ciągłego kodowania dla problemu dynamicznej marszrutyzacji. W wyniku wykorzystywania w algorytmie ContDVRP ciągłej przestrzeni przeszukiwań, opisanej w podrozdziale 5.2, zmienia się kategoryzacja problemu również w zakresie poziomu zmienności. Przy stosowaniu ciągłego kodowania przynajmniej niektóre ze zmian zachodzących w problemie, mogą mieć charakter dynamicznie nieistotnych zmian, co jest odzwierciedlone w zmienionej klasyfikacji DVRP na Rysunku 1.2.

1.5 Cel optymalizacji

Jak napisano w podrozdziale 1.3, chociaż poziom dostępności danych problemu zwykle powiązany jest również z celem optymalizacji, w tej rozprawie są one zdefiniowane osobno. W problemach statycznych i stochastycznych istotna jest **optymalizacja dla finalnego stanu systemu** natomiast w problemach dynamicznych standardowym celem jest **ciągłe utrzymywanie optymalnego wyniku**.

Definicja 17. *W **zadaniu optymalizacji finalnego rozwiązania** celem jest osiągnięcie wartości optymalnej na koniec procesu optymalizacji (ozn. T_{MAX}). Dla problemów statycznych poszukiwany zbiór $\mathcal{X} \subseteq \Omega$ ma postać identyczną jak w definicji problemu optymalizacyjnego. Dla problemów zmiennych w czasie poszukiwane jest rozwiązanie optymalne w chwili T_{MAX}:*

$$\mathcal{X}_{T_{MAX}} = \{x \in \Omega(T_{MAX}) : \forall_{x' \in \Omega(T_{MAX})} f(x, T_{MAX}) \succeq f(x', T_{MAX})\}$$

Definicja 18. *W **zadaniu optymalizacji z ciągłym utrzymaniem wyniku** celem jest pozostawanie rozwiązania w pobliżu optimum podczas całego procesu optymalizacji. Rozwiązanie optymalne takiego zadania ma postać ciągu zbiorów $(\mathcal{X}_t)$ o elementach następującej postaci:*

$$\mathcal{X}_t = \{x \in \Omega(t) : \forall_{x' \in \Omega(t)} f(x, t) \succeq f(x', t)\}$$

Należy też zauważyć, że zmienne w czasie problemy dostarczają trudności w mierzeniu jakości danego algorytmu. Ze względu na brak możliwości restartów metod (decyzje podejmowane w trakcie działania wpływają na stan problemu) istotne jest prezentowanie wartości średnich (lub median) czy rozkładu wyników w celu oszacowania działania algorytmu (pojedyncze dobre wyniki nie powinny być uznane za główną miarę jakości, a jedynie możliwości danej metody). Możliwość zmiany wartości optimum globalnego przy jednoczesnej nieznajomości jego wartości również utrudniania porównywanie uzyskiwa-

nych wyników. W zadaniach optymalizacji z utrzymywaniem wyniku (śledzeniem optimum) funkcjonuje kilka konkurencyjnych miar:

- średnia dokładność (uśredniona różnica między wartością wskazaną przez algorytm a wartością optimum w trakcie całego czasu działania algorytmu),
- stabilność wyniku (utrzymywanie zbliżonej względnej jakości wyników w trakcie całego procesu optymalizacji),
- szybkość zbieżności po zmianie (jak szybko po zmianie algorytm dochodzi w pobliże najlepszej jakości osiąganej przy danym stanie problemu).

Więcej informacji na temat mierników jakości można znaleźć w rozprawach [82, 130].

Dla omawianego w głównej części tej rozprawy problemu dynamicznej marszrutyzacji, podstawowym miernikiem są średnie wyniki uzyskiwane na koniec procesu optymalizacji. Dodatkową miarą jest zmienność rozwiązania w czasie i różnice pomiędzy rozwiązaniami dla pośrednich stanów problemu a rozwiązaniem finalnym. Miary zmienności są istotne ze względu na badanie praktycznych własności danego podejścia. Mniejsza zmienność rozwiązania oznacza mniejszą konieczność wymiany informacji pomiędzy dyspozytorem a kierowcami, czyli upraszcza i zmniejsza koszty bieżącego zarządzania flotą pojazdów.

Rozdział 2

Metodyka prac z problemami dynamicznymi

Opisana wcześniej kategoryzacja dynamicznych problemów optymalizacyjnych wiąże się ściśle z dwoma obszarami badań: konstrukcją generatorów problemów testowych oraz technikami uzupełniającymi algorytmy optymalizacyjne dla problemów statycznych, zwiększającymi skuteczność działania tych algorytmów dla problemów zmiennych w czasie. W tym rozdziale zaprezentowany jest przegląd technik stosowanych w różnego rodzaju podejściach i problemach dynamicznych. Szczególny nacisk położony jest na techniki stosowane pierwotnie w algorytmach ewolucyjnych, a później również w innych populacyjnych metodach metaheurystycznych. Natomiast sam opis populacyjnych algorytmów metaheurystycznych znajduje się w Rozdziale 6.

2.1 Instancje problemów testowych

Wymienione w tym rozdziale techniki często prezentowane są jako poprawiające skuteczność działania algorytmów optymalizacyjnych dla konkretnej grupy problemów. W pracach [108, 109] wskazana jest rozbieżność pomiędzy badaniami teoretycznymi, koncentrującymi się na funkcjach ciągłych i charakterystyce ich zmian, a dużą grupą praktycznych problemów dynamicznych, które mają charakter dyskretny. Prace te wskazują też na istotność właściwego doboru kodowania odpowiedniego dla zmian zachodzących w danym środowisku dynamicznym, zapewniającego łatwość przekazywania rozwiązań. W tym zakresie stanowią rozwinięcie artykułu [136], w którym zostało zidentyfikowane zagadnienie wpływu zmian zachodzących w problemie na rozwiązanie optymalne.

Praca [136] jest jednocześnie jedną z pierwszych, proponujących zunifikowany generator dynamicznych problemów testowych. Niezależnie od [136] generator problemów dynamicznych jest rozwijany przez grupę badawczą Yanga [65,124,126]. Generator [65] pozwala na parametryzację wielkości szumu, wielkości kroku zmiany oraz cykliczności zmian.

Prace przeglądowe, z 2012 i 2013 roku, [82,107] zbierają i prezentują wyniki teoretyczne w zakresie problemów dynamicznych. Praca [107] pokazuje nieintuicujność problemów dynamicznych (np. pozornie duże zmiany bez istotnego wpływu na wynik). Praca ta wytycza też kilka kierunków przyszłych badań, wskazując m.in. na konieczność uspójnienia klasyfikacji algorytmów oraz sposobu definiowania optimum. Praca [82] wskazuje na istniejące wciąż braki w wynikach teoretycznych w zakresie Ewolucyjnej Dynamicznej Optymalizacji (ang. *Evolutionary Dynamic Optimization*) oraz pokrewnych metod metaheurystycznych.

2.2 Rozszerzenia metod metaheurystycznych

Podstawowe techniki, wykorzystywane do wzmocnienia metod ewolucyjnych w zadaniach optymalizacji dynamicznej, są przytoczone we wspomnianej wcześniej rozprawie doktorskiej Weickera [130]. Są to:

- restart algorytmu (z wykorzystaniem informacji o wcześniejszych wynikach),
- lokalne przeszukiwanie względem wcześniej znalezionych rozwiązań,
- przechowywanie wcześniejszych rozwiązań w pamięci,
- utrzymywanie różnorodności populacji,
- adaptacja zakresu mutacji i nauka trendu zmian problemu,
- ponowna ewaluacja najstarszych osobników,
- zaprojektowanie kodowania z sąsiedztwem dobrze dopasowanym do zmian zachodzących w problemie.

Dużą pulę algorytmów populacyjnych dla problemów dynamicznych prezentuje również praca przeglądowa grupy badawczej *Models of Decision and Optimiztion*[1] [26], która grupuje wykorzystywane techniki w:

- utrzymywanie różnorodności populacji,
- pamięć z wcześniejszymi rozwiązaniami,
- detekcję zmian,
- obliczenia z wykorzystaniem wielu populacji.

Praca ta porusza również sposoby pomiaru jakości algorytmów rozwiązujących problemy dynamiczne: dokładność, stabilność i reaktywność.

Zasadność wykorzystania konkretnej z wyżej wymienionych technik wspomagających jest oczywiście silnie zależna od rodzaju dynamizmu danego problemu. Jednak do podstawowych narzędzi związanych z populacyjnymi algorytmami dynamicznymi należą: przechowywanie historycznych rozwiązań w pamięci oraz utrzymywanie różnorodności populacji [134, 135]. Można dostrzec, że stanowią one przeniesienie problemu balansu między eksploracją a eksploatacją w dziedzinę optymalizacji dynamicznej. Warto dodać, że analogiczne techniki znajdują zastosowanie również w optymalizacji wielokryterialnej.

[1] Witryna poświęcona dynamicznym problemom optymalizacyjnym utrzymywana przez grupę badawczą MODO: `http://www.dynamic-optimization.org/`

Przechowywanie zestawów rozwiązań historycznych jest szczególnie istotnym zagadnieniem dla problemów o zmienności cyklicznej [74, 133]. Natomiast utrzymywanie różnorodnej populacji jest istotne dla problemów, w których celem jest śledzenie zmieniającego się optimum [10, 77]. Porównanie algorytmów dla tego rodzaju problemów (z wykorzystaniem *Moving Peeks Benchmark* [13]) zostało opisane w pracy [30]. W tym porównaniu został wykorzystany algorytm genetyczny z samoorganizującymi się losowymi imigrantami SORIGA [125], warianty kwantowej optymalizacji rojem cząstek mQSO [11], algorytm współpracujących agentów [91] oraz strategie śledzenia trajektorii [46].

Z kolei w przypadku problemów dynamicznych, w których celem jest optymalizacja finalnego wyniku, ważne jest znajdowanie rozwiązań pośrednich odpornych na zmiany. Jednym z możliwych podejść jest wprowadzenie do funkcji oceniającej czynnika estymującego elastyczność rozwiązania [14] (sprowadzając w ten sposób aktualny stan problemu do zagadnienia optymalizacji dwukryterialnej: aktualnej jakości wyniku oraz elastyczności rozwiązania).

Oprócz wymienionych wcześniej metod, prezentowanych we wspomnianych pracach przeglądowych, warto jeszcze wspomnieć o następujących zagadnieniach:

- doborze kodowania najlepiej sprawdzającym się dla danego rodzaju dynamizmu,
- optymalnym wykorzystaniu zasobów w obliczeniach asynchronicznych (aktualny stan problemu nie jest jednocześnie propagowany do wszystkich procesorów),
- nieodwracalności decyzji podejmowanych w problemach optymalizacji dynamicznej.

Istotność doboru kodowania była omawiana na przykładzie problemu plecakowego [15], jednak nie doczekała się szerszej generalizacji czy popularności w problemach dynamicznych [110].

Optymalne wykorzystanie zasobów w obliczeniach asynchronicznych jest rozważane w pracy [58], na przykładzie problemu dynamicznej marszrutyzacji. Praca prezentuje techniki integracji w populacji rozwiązań uzyskanych dla innych stanów problemu, niż aktualnie obliczany na danym procesorze.

Ostatnim ze wspomnianych powyżej zagadnień jest nieodwracalność podejmowanych decyzji, co skutkuje tym, że nie tylko zewnętrzne czynniki środowiskowe, ale również wybór algorytmu mogą mieć wpływ na pośrednie stany optymalizowanego problemu. Odpowiedzią na ten efekt może być wybór podejścia hiperheurystycznego. Zgodnie z pracą przeglądową [112] podejścia te można dzielić na heurystykę wyboru metody [24] oraz metodykę generowania heurystyk [20]. Warto zauważyć, że w ramach tych definicji mieszczą się również warianty algorytmów kulturowych [106] i memetycznych [79]. Sposób wykorzystania hiperheurystyki w rozumieniu pracy [24], na przykładzie problemu dynamicznej marszrutyzacji, można znaleźć w jednej z prac prezentujących częściowe wyniki tej rozprawy [87]. W rozprawie jest ona przedstawiona w podrozdziale 7.3.

2.3 Inne metody optymalizacyjne

Niezależnie od technik dedykowanych algorytmom populacyjnym (takim jak algorytm ewolucyjny czy metody rojowe) rozwijane były podejścia tzw. optymalizacji odpornej. Optymalizacja odporna (ang. *robust optimization*) uwzględnia niepewność danych problemu w formułowaniu zadania optymalizacji w metodach programowania liniowego, kwadratowego itp. Prace dotyczące metod optymalizacji odpornej oraz optymalizacji najgorszego przypadku (ang. *worst case optimization*) dostarczają odmiennego wglądu w sposób traktowania problemów dynamicznych. Za podłoże teoretyczne tego podejścia można uznać pracę Walda dotyczącą wskazywania estymacji minimalizujących ryzyko "dużych" błędów dla pewnych rodzin rozkładów [129]. Dla niektórych problemów próba wskazania bezpośredniej zależności rozkładu wyników działania danego algorytmu od rozkładu danych problemu może być skazana na niepowodzenie (np. z powodu braku informacji na temat rozkładu danych problemu). Jednak można zastosować techniki biorące pod uwagę istnienie niepełnej informacji i próbę wskazania rozwiązań uwzględniających nieznaną część problemu. Związek modelu Walda z podejmowaniem decyzji w warunkach niepewności jest zaprezentowany w pracach [114, 115].

W podejściu optymalizacji odpornej główny nacisk położony jest na sposób przeformułowania zadania i jego ograniczeń, aby uzyskać problem deterministyczny, którego rozwiązanie z dużym prawdopodobieństwem zmieści się w zbiorze dopuszczalnym pierwotnego problemu niepewnego. Praca [7] prezentuje wpływ zaburzeń zmiennych modelu, uzyskanych metodami programowania matematycznego. W odpowiedzi na zaobserwowany problem praca ta proponuje sposoby uwzględniania niepewności w definicji zadania, co prowadzi do znajdowania rozwiązań suboptymalnych, ale odpornych na zaburzenia. Z kolei praca [137] koncentruje się na sposobie zmniejszenia liczby ograniczeń, o które trzeba zmodyfikować definicje problemów logistycznych na potrzeby programowania liniowego, aby możliwe było zastosowanie podejścia odpornego. Wreszcie artykuł [8] zwraca uwagę na potencjał mniejszego kosztu obliczeniowego optymalizacji odpornej względem podejść stochastycznych. Praca ta koncentruje się też na elastyczności rozwiązań i ich związku ze strukturą optymalizowanych problemów.

W nurt optymalizacji odpornej można wpisać również wspomnianą wcześniej optymalizację najgorszego przypadku. Podejście takie zostało z powodzeniem zastosowane m.in. w doborze parametrów terapii wiązką protonów o modelowanej intensywności, stosowanej w leczeniu raka. Metoda polega na szukaniu zestawu parametrów maksymalizujących wartość oczekiwaną dawki promieniowania w chorych komórkach z jednoczesną minimalizacją wartości oczekiwanej dawki promieniowania w komórkach zdrowych [92]. Praca ta nie tylko prezentuje praktyczną skuteczność tej metody, ale również omawia jej uniwersalność względem algorytmu optymalizacyjnego (prezentując jej działanie na przykładzie metody gradientowej L-BFGS-B [138]).

Próba zunifikowania spojrzenia na techniki wykorzystywane w algorytmach populacyjnych oraz w optymalizacji odpornej została podjęta przez Jina i Brankego w 2005 [52] oraz Beyera i Sendhofa w 2007 [9]. Praca [52] zwraca uwagę na podobieństwo tematyki problemów odpornych z zaszumionymi funkcjami. Natomiast praca [9] przedstawia optymalizację odporną, biorąc pod uwagę nie tylko techniki programowania matematycznego stosowane do problemów projektowania inżynierskiego, ale również wyniki związane z badaniami operacyjnymi oraz metodami ewolucyjnymi i algorytmami dokładnymi.

2.4 Wnioski

W tej części rozprawy została zaprezentowana klasyfikacja problemów optymalizacyjnych oraz przedstawione techniki wspomagające algorytmy optymalizacyjne w rozwiązywaniu problemów dynamicznych. Przeprowadzona analiza literatury wykazuje istnienie braków w badaniach nad problemami o zmiennym rozmiarze i w klasyfikacji tego typu problemów. Ponadto techniki wspomagające rozwiązywanie problemów dynamicznych i stochastycznych dedykowane algorytmom ewolucyjnym (a szerzej populacyjnym), jedynie w niewielkim stopniu były zestawiane z podejściami do tych samych problemów wykorzystującymi metody gradientowe czy programowanie liniowe.

W konsekwencji przeprowadzonej analizy algorytm prezentowany w głównej części rozprawy, rozwiązujący problem dynamicznej marszrutyzacji, wykorzystuje zarówno techniki związane z metodami metaheurystycznymi (niezależne populacje, przekazywanie rozwiązań historycznych, zwiększanie różnorodności populacji poprzez dodawanie rozwiązań stworzonych algorytmem heurystycznym), jak również podejście wzorowane na metodach odpornych, poprzez uwzględnienie estymacji finalnej sumy wielkości zamówień. Podobnie metody pomiaru dynamizmu problemu w oparciu o porównanie kolejnych wektorów rozwiązań, zostały zaadaptowane do problemu o zmiennym rozmiarze. Ostatnim wnioskiem jest konieczność wyodrębnienia kategorii poziomu dostępności danych problemu oraz celu optymalizacji, aby było możliwe poprawne opisanie omawianego w rozprawie problemu dynamicznej marszrutyzacji.

Część II

Problem Dynamicznej Marszrutyzacji

Rozdział 3

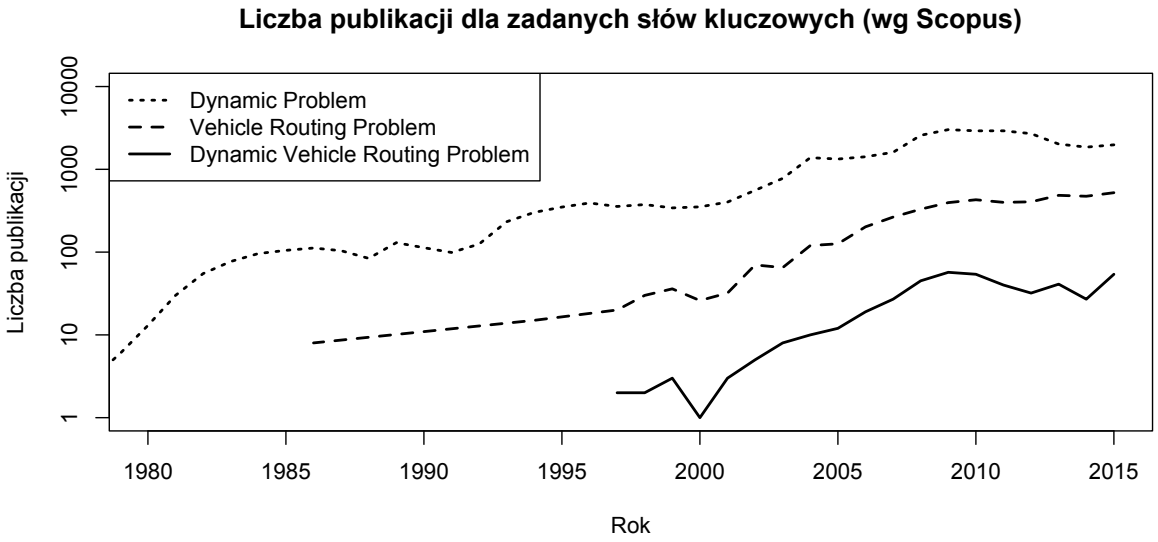
Problem Dynamicznej Marszrutyzacji

W tym rozdziale opisany jest Problem Dynamicznej Marszrutyzacji (ang. *Dynamic Vehicle Routing Problem* (DVRP)). Zdefiniowany jest jego model matematyczny oraz parametry operacyjne. Rozdział ten stanowi również wprowadzenie do całego obszaru problemów transportowych, ze szczególnym uwzględnieniem problemów marszrutyzacji.

Opisywany w tej rozprawie typ DVRP jest najpopularniejszym występującym w literaturze [94] i bywa określany jako problem marszrutyzacji z dynamicznymi zamówieniami (ang. *Vehicle Routing Problem with Dynamic Requests* (VRPwDR)) [57]. DVRP jest zagadnieniem o dużym znaczeniu praktycznym w obszarze logistyki i badań operacyjnych. DVRP jest również interesujący naukowo, zarówno ze względu na jego złożoność wynikającą z NP-trudnej natury problemu statycznego, na którą dodatkowo nałożony jest brak części danych wynikający z dynamizmu problemu, jak i możliwość stosowania różnego rodzaju podejść: grafowych, geometrycznych czy statystycznych.

Poza licznymi publikacjami (kilkadziesiąt rocznie od 2008 (Rysunek 3.1)) temat został również opracowany w innych rozprawach doktorskich: [54, 63, 93]. Rozprawy te kładą nacisk na różne aspekty zagadnienia: od matematycznej natury problemu, poprzez obliczenia równoległe, aż do badań operacyjnych.

Osiągnięta poprawa wyników dla problemów testowych wykorzystywanych w literaturze, uzyskana zarówno w tej rozprawie, jak i pracy [71], wskazuje na istotność badań w tym zakresie. Ponadto zarówno problemy dynamiczne (kilka tysięcy publikacji rocznie dla *Dynamic Problem*), jak i problemy transportowe (kilkaset publikacji rocznie i rosnący trend dla *Vehicle Routing Problem*), stanowią aktualny temat prac badawczych.



Rysunek 3.1: Liczba publikacji (wg bazy Scopus) w skali logarytmicznej dla poszczególnych słów kluczowych w podziale na lata.

Tabela 3.1: Symbole wykorzystywane w opisie DVRP

Symbol	Typ	Opis
$arv_{r_{i,j}}$	$\mathbb{R}$	Czas przyjazdu i-tego pojazdu do j-tego zamówienia
$\widehat{arv}_{r_{i,j}}$	$\mathbb{R}$	Estymowany czas przyjazdu i-tego pojazdu do j-tego zamówienia
C	Ciąg	Zamówienia
cap	$\mathbb{R}_+$	Ładowność pojazdu
l_i	$\mathbb{R}^2$	Położenie i-tego zamówienia
l_0	$\mathbb{R}^2$	Położenie bazy pojazdów
n	$\mathbb{Z}_+$	Liczba pojazdów
m	$\mathbb{Z}_+$	Finalna liczba zamówień
m_i	$\mathbb{Z}_+$	Liczba lokalizacji (zamówień i bazy) przypisanych do i-tego pojazdu
$m(t)$	$\mathbb{Z}_+$	Liczba zamówień znanych w czasie t
$mp(t)$	$\mathbb{Z}_+$	Liczba oczekujących zamówień w czasie t
$\mathbf{r_i}$	Ciąg	Trasa i-tego pojazdu (indeksy lokalizacji kolejnych zamówień na trasie)
$r_{i,j}$	$\mathbb{Z}_0$	Identyfikator j-tej lokalizacji w trasie i-tego pojazdu
$\rho(i,j)$	$\mathbb{R}$	Odległość pomiędzy l_i a l_j
s_i	$\mathbb{R}_+$	Rozmiar i-tego zamówienia
sp	$\mathbb{R}_+$	Prędkość pojazdu
t_i	$\mathbb{R}$	Czas zgłoszenia i-tego zamówienia
t_{start}	$\mathbb{R}$	Czas otwarcia bazy
t_{end}	$\mathbb{R}$	Czas zamknięcia bazy
u_i	$\mathbb{R}_+$	Czas obsługi i-tego zamówienia
V	Ciąg	Flota pojazdów

3.1 Optymalizacja problemów transportowych

Jednym z podstawowych optymalizacyjnych problemów transportowych jest problem komiwojażera (ang. *Traveling Salesman Problem* (TSP)), spopularyzowany przez Flooda [37]. Ze względu na to, że TSP przyjmuje jedynie bardzo prosty model rzeczywistości, został on przez Dantziga i Ramsera uogólniony do problemu marszrutyzacji (*Vehicle Routing Problem* (VRP)), pod pierwotną nazwą *truck dispatching problem* [28]. Problem marszrutyzacji rozszerza problem komiwojażera o znalezienie podziału wierzchołków pomiędzy wiele pojazdów (w miejsce pojedynczego komiwojażera) i zbudowanie optymalnych tras dla każdego z pojazdów. Lenstra i Kan [64] wykazali, że problem ten jest NP-trudny. Problem marszrutyzacji doczekał się licznych wariantów mających za zadanie dokładniejsze modelowanie rzeczywistości[1]. Wśród tych wariantów warto wyróżnić: uwzględnienie ograniczenia pojemności pojazdu (*Capacitated VRP* [105]), uwzględnienie niepewności wielkości zamówienia (*VRP with Stochasic Demands* [31]), modelowanie okien czasowych na obsługę zamówienia (*VRP with Time Windows* [116]), czy wykorzystanie niejednorodnej floty pojazdów (*Heterogenous Fleet VRP* [121]).

Dynamiczne warianty problemu zaczęto rozważać w związku z rozwojem systemów informacji przestrzennej (GIS) oraz systemów komunikacji (GSM) i lokalizacji (GPS) [102]. Popularyzacja DVRP nastąpiła wraz z ustandaryzowaniem zbioru przypadków testowych przez Kilby'ego i in. [59].

Szczególnie popularne jest wykorzystywanie do rozwiązywania DVRP metod metaheurystycznych:

- losowego algorytmu zachłannego (Montemanni i in. [76]),
- przeszukiwania z tabu (Hanshar i Ombuki-Berman [47]),
- metody zmiennego sąsiedztwa (Khouadjia i in. [57]),
- algorytmu mrówkowego (Montemanni i in. [75], Elhasannia i in. [33]),
- algorytmu genetycznego (Hanshar i Ombuki-Berman [47], Elhasannia i in. [34]),
- optymalizacji rojem cząstek (Khouadjia i in. [57], Okulewicz i Mańdziuk [83])
- metod adaptacyjnych (Garrido i Riff [43], Okulewicz i Mańdziuk [87], Mańdziuk i Żychowski [71]).

Inne podejścia stosowane do DVRP obejmowały: programowanie liniowe (Yang i in. [132]), czy heurystykę podziału zamówień opartą o ich niezgodność i przewidywaną liczbę potrzebnych pojazdów (Moretti-Branchini i in. [78]).

Bogaty przegląd algorytmów dokładnych, aproksymacyjnych, metaheurystycznych i hybrydowych stosowanych do problemów marszrutyzacji można znaleźć w rozprawie doktorskiej Pillaca [93].

[1]Listę wybranych wariantów VRP oraz literaturę przedmiotu można odnaleźć na stronie `http://neo.lcc.uma.es/vrp/`

3.2 Opis problemu

Zgodnie z klasyfikacją zaprezentowaną w Rozdziale 1, DVRP jest dynamicznym dyskretnym problemem optymalizacyjnym o chaotycznej skokowej zmienności, w którym celem jest optymalizacja finalnego wyniku. Przy bezpośredniej dyskretnej reprezentacji problemu oznacza to, że rozmiar przestrzeni przeszukiwań jest zmienny w czasie. Zaproponowane w rozprawie ciągłe kodowanie, omówione w Rozdziale 5, zmniejsza poziom zmienności problemu.

Należy też zauważyć, że DVRP łączy w sobie cechy problemu dynamicznego oraz stochastycznego. Z jednej strony zmienia się wymiarowość problemu poprzez pojawianie się kolejnych klientów i obsługiwanie kolejnych zamówień przez pojazdy, co sprawia, że warto stosować techniki związane w problemami dynamicznymi (np. przekazywanie poprzednich rozwiązań przy zmianie stanu problemu). Z drugiej strony celem jest optymalizacja finalnego rozwiązania w warunkach niepewności, co wskazuje na zasadność stosowania technik związanych z problemami stochastycznymi, generującymi ciąg rozwiązań cząstkowych powiązany nie tyle z aktualnym stanem, ale z przewidywanym finalnym stanem środowiska.

Rozwiązaniem optymalnym DVRP jest zbiór permutacji podzbiorów listy klientów C, składających się z niepodzielnych zamówień, generujący najkrótszą łączną trasę dla floty pojazdów V w jednym dniu roboczym. Każda permutacja z tego zbioru definiuje kolejność w jakiej jeden z pojazdów ma obsłużyć zamówienia z tej permutacji. Trasa i harmonogramy pojazdów generowane przez permutację muszą spełniać ograniczenia pojemności, czasu dostępności pojazdu oraz logiki wynikającej z czasu obsługi zamówienia i czasu pojawiania się zamówień. Wszystkie pojazdy należące do V mają identyczną pojemność *cap* i prędkość *sp*. Wszystkie pojazdy mają wspólną bazę (zajezdnię) oznaczoną indeksem 0. Tabela 3.1 przedstawia oznaczenia niezbędne do formalnego zdefiniowania celu optymalizacji oraz ograniczeń w DVRP.

W dalszej części rozdziału w sposób formalny jest przedstawiona definicja samego zadania. Omówione jest również miejsce procesu optymalizacji w działalności operacyjnej firmy transportowej, a także sposób budowania i zatwierdzania harmonogramu pojazdu na podstawie listy lokalizacji do odwiedzenia.

3.3 Cel i ograniczenia optymalizacji

Celem DVRP jest znalezienie takiego zbioru permutacji $R^* = \{\mathbf{r_1^*}, \mathbf{r_2^*}, \ \ldots, \mathbf{r_n^*}\}$, który stanowi minimum następującej funkcji kosztu:

$$Cost(R) = \sum_{i=1}^{n} \sum_{j=2}^{m_i} \rho(r_{i,j-1}, r_{i,j}) \tag{3.1}$$

Harmonogram i-tego pojazdu generowany na podstawie permutacji r_i musi posiadać następujące cechy:
Pojazd rozpoczyna i kończy swoją trasę w bazie:

$$\forall_{i\in\{1,2,\ldots n\}} \left(r_{i,1} = 0 \wedge r_{i,m_i} = 0\right) \tag{3.2}$$

Pojazd wraca do bazy przed jej zamknięciem:

$$\forall_{i\in\{1,2,\ldots n\}} arv_{r_{i,m_i}} \leq t_{end} \tag{3.3}$$

Pojazd opuszcza bazę po jej otwarciu:

$$\forall_{i\in\{1,2,\ldots n\}} arv_{r_{i,1}} \geq t_{start} \tag{3.4}$$

Pojazd może dotrzeć do kolejnej lokalizacji po obsłużeniu zamówienia z poprzedniej:

$$\forall_{i\in\{1,2,\ldots n\}}\forall_{j\in\{2,3\ldots m_i\}} arv_{r_{i,j}} \geq arv_{r_{i,j-1}} + u_{r_{i,j-1}} + \rho(r_{i,j-1}, r_{i,j})/sp \tag{3.5}$$

Pojazd wyrusza do kolejnej lokalizacji dopiero gdy jest znana:

$$\forall_{i\in\{1,2,\ldots n\}}\forall_{j\in\{2,3\ldots m_i\}} arv_{r_{i,j}} \geq t_{r_{i,j}} + \rho(r_{i,j-1}, r_{i,j})/sp \tag{3.6}$$

Każde zamówienie jest obsługiwane przez dokładnie jeden pojazd:

$$\forall_{j\in\{1,2,\ldots m\}} \exists!_{i\in\{1,2,\ldots n\}} j \in \mathbf{r_i} \tag{3.7}$$

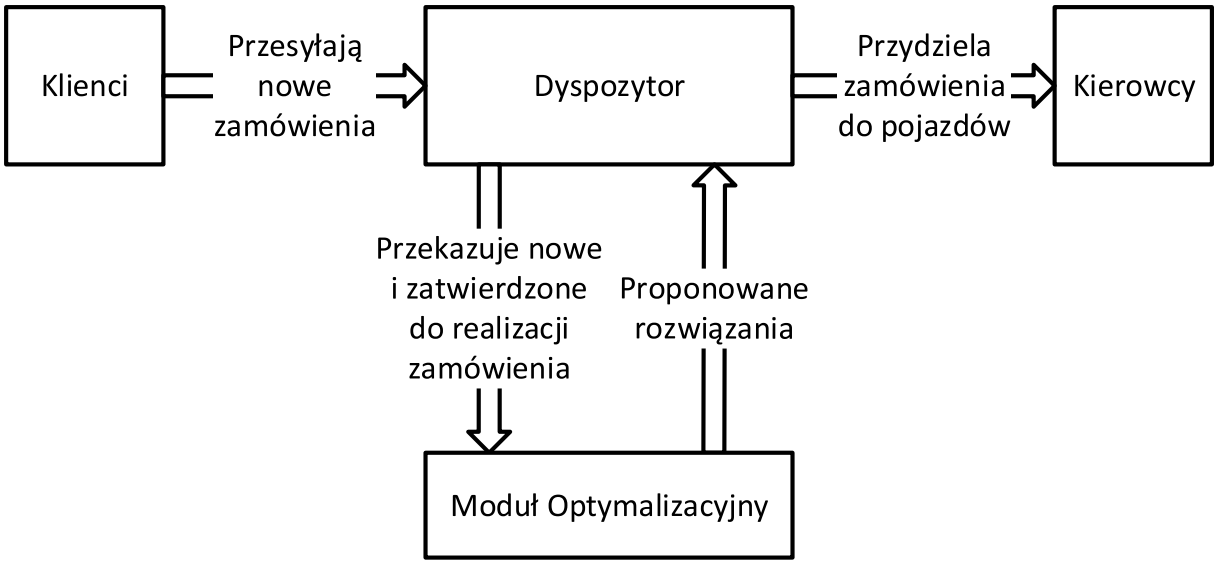
Pomiędzy każdymi dwiema wizytami w bazie, suma zamówień obsługiwanych przez pojazd nie może przekroczyć jego pojemności:

$$\forall_{i\in\{1,2,\ldots n\}}\forall_{j_1<j_2<j_3\in\{1,2\ldots m_i\}} r_{i,j_1} = 0 \wedge r_{i,j_3} = 0 \wedge r_{i,j_2} \neq 0 \Rightarrow \sum_{j=j_1+1}^{j_3-1} s_{r_{i,j}} \leq cap \tag{3.8}$$

3.4 Optymalizacja DVRP

Skokowy i chaotyczny charakter zmian wynikający z samego zadania DVRP powoduje, że proces optymalizacji może być dzielony na szereg przedziałów czasowych równej wielkości [33, 34, 43, 47, 59, 75, 83], bez dodatkowej utraty ciągłości między kolejnymi stanami.

W każdym z takich przedziałów stan problemu jest traktowany jako niezmienny (nowe zamówienia oczekują poza procesem optymalizacji aż do początku następnego przedziału czasowego). W związku z tym, w procesie optymalizacji rozwiązywany jest szereg zależnych od siebie problemów statycznych.



Rysunek 3.2: Diagram przepływu danych pomiędzy klientami, kierowcami, modułem optymalizacyjnym a dyspozytorem.

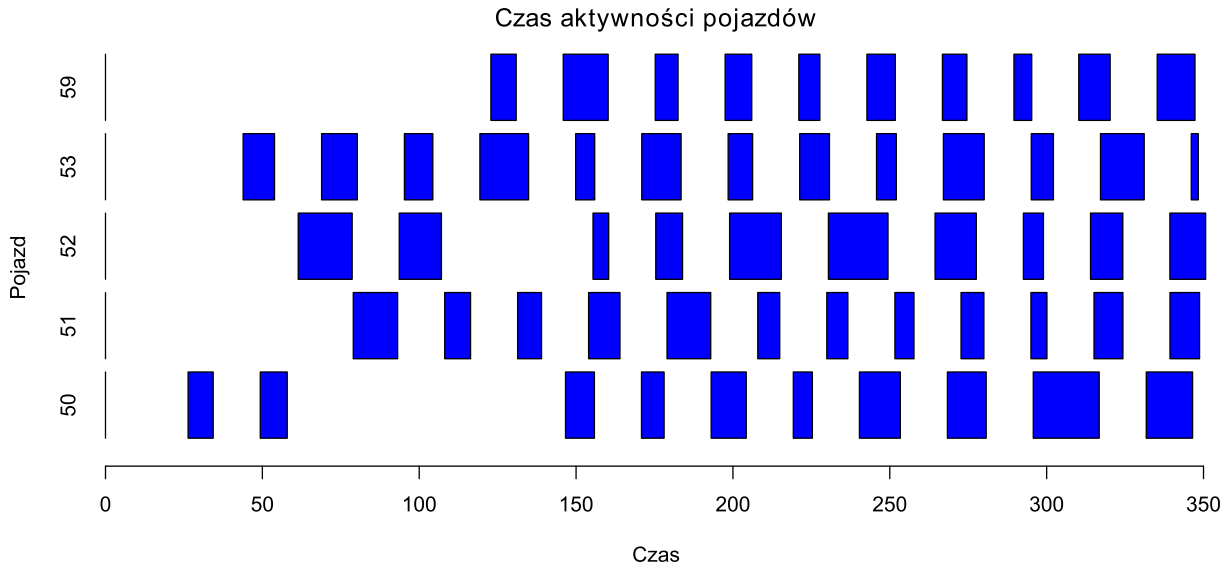
Należy zauważyć, że daje to również możliwość uruchomienia procesu optymalizacyjnego przez dyspozytora pojazdów w dowolnym momencie na podstawie bieżącego stanu problemu. Miejsce Modułu Optymalizacyjnego w praktyce pracy dyspozytora pojazdów prezentuje Rysunek 3.2. Należy wspomnieć, że istnieją też podejścia alternatywne wobec podziału na przedziały czasowe, w których optymalizacja prowadzona jest w sposób ciągły [128]. W tym kontekście algorytm MEMSO [54, 55, 58] jest podejściem hybrydowym. Każdy z wykorzystywanych w nim serwisów optymalizacyjnych prowadzi obliczenia w dyskretnych przedziałach czasowych, natomiast wymiana danych pomiędzy tymi serwisami zakłada możliwość integracji w populacji rozwiązań pochodzących z innego kroku czasowego.

Zamówienia w danej chwili t (a w konsekwencji w dyskretnym przedziale czasowym, do którego należy t) możemy podzielić na 3 grupy: nieznane, oczekujące i zatwierdzone.

Definicja 19. *Zamówieniem **nieznanym** (ang. unknown request) nazywamy zamówienie, które nie było (i być może dalej nie jest) znane dyspozytorowi w momencie uruchamiania procesu optymalizacji.*

Definicja 20. *Zamówieniem **oczekującym** (ang. pending request) nazywamy zamówienie, które poddawane jest procesowi optymalizacji. Pojazd do jakiego jest przypisane oraz miejsce w kolejności obsługi może jeszcze ulec zmianie. Może ono także pozostawać okresowo nie przypisane do żadnego pojazdu.*

Definicja 21. *Zamówieniem **zatwierdzonym** (ang. commited request) nazywamy zamówienie, które jest przypisane do konkretnego pojazdu i jego planowany czas realizacji nie podlega już zmianie.*



Rysunek 3.3: Przykładowy harmonogram przejazdów floty pomiędzy kolejnymi punktami. Przejazdy są rozdzielone czasami rozładunku i czasem oczekiwania na zatwierdzenie kolejnego zamówienia.

Należy zauważyć, że zatwierdzenie zamówienia ma istotny wpływ na możliwe do osiągnięcia wyniki. Z jednej strony wspomaga proces optymalizacji poprzez zmniejszenie przestrzeni przeszukiwań oraz wprowadzenie rozróżnialności pojazdów. Z drugiej strony na stałe ogranicza możliwość znalezienia pewnych rozwiązań problemu. Przykład realizacji harmonogramu pojazdów jest przedstawiony na Rysunku 3.3.

3.5 Parametry operacyjne DVRP

Oprócz formalnych warunków dopuszczalności rozwiązania, istotne jest również zapewnienie parametrów operacyjnych umożliwiających przeprowadzenie samego procesu optymalizacji oraz określających zasady generowania harmonogramu pojazdów.

W tym podrozdziale są opisane trzy parametry wpływające kolejno na: poziom dynamizmu problemu, aktualność danych w procesie optymalizacji, elastyczność zmian harmonogramów. Te trzy parametry to:

- czas odcięcia (ang. *cut–off time* (T_{CO})),
- liczba przedziałów czasowych (ang. *number of time slices* (N_{TS})),
- bufor czasowy (ang. *advanced commitment time* (T_{AC})).

Definicja 22. ***Czas odcięcia*** *T_{CO} określa czas, po którym nowo pojawiające się zamówienia nie są już akceptowane do realizacji danego dnia roboczego.*

W problemach testowych wykorzystywanych w tej pracy oraz w literaturze przedmiotu T_{CO} jest standardowo ustawiany na połowę dnia roboczego i jest jednakowy dla wszystkich zamówień. W praktyce operacyjnej jego wartość jest zależna od odległości regionu (w którym pojawia się zamówienie) od bazy.

T_{CO} jest czynnikiem wpływającym wprost proporcjonalnie na poziom dynamizmu problemu. Im większy T_{CO}, tym więcej danych jest nieznanych na początku procesu optymalizacji, i tym większy dynamizm problemu.

Definicja 23. ***Stopień dynamizmu problemu*** *(ang. degree of dynamism (dod)) określa liczbę zamówień nieznanych na początku dnia roboczego w stosunku do wszystkich zamówień*

$$dod = \frac{m - m(t_{start})}{m} \tag{3.9}$$

Wpływ różnych wartości T_{CO}, a tym samym różnych wartości *dod*, na działanie algorytmu optymalizacyjnego został opisany na przykładzie algorytmu DAPSO w [57].

Definicja 24. ***Liczba przedziałów czasowych*** N_{TS} *definiuje liczbę okresów, w których prowadzony jest proces optymalizacji na ustalonym stanie zadania. Jednocześnie wyznacza ona liczbę momentów, w których tworzony jest nowy harmonogram pojazdów.*

Przyjęta wartość N_{TS} definiuje czas pojedynczego okna czasowego ΔTS:

$$\Delta TS = \frac{t_{end} - t_{begin}}{N_{TS}} \tag{3.10}$$

W początkowych pracach nad DVRP [59, 75] wartość parametru N_{TS} była przedmiotem badań. Wyniki uzyskane dla stosowanych w tych pracach algorytmów sugerowały liczbę 25 lub 50 kroków czasowych. Wartość 25 została w literaturze przyjęta jako referencyjna (ze względu na wybór takiej wartości w najczęściej cytowanych pracach [47, 75, 76]). W tej rozprawie, zgodnie z zaleceniami z pierwotnych prac, parametr ten został dostrojony dla ContDVRP[2] (szczegółowe wyniki strojenia są zaprezentowane w Dodatku A.2). Należy tutaj zauważyć, że w praktyce operacyjnej wartość tego parametru może zostać ustalona z góry, gdyż definiuje on ciąg punktów decyzyjnych (TS_i) procesu optymalizacji. Liczba punktów decyzyjnych jest jednym z czynników wpływających na koszty zarządzania flotą pojazdów, co również należy wziąć pod uwagę przy wdrożeniu danego procesu optymalizacji. i-ty punkt decyzyjny, gdzie $i = \{0, 1, \ldots, N_{TS}\}$, można zdefiniować w następujący sposób:

$$TS_i = t_{begin} + i\Delta TS \tag{3.11}$$

Aby określić jaka część zamówień przypisanych do danego pojazdu powinna zostać zatwierdzona do realizacji w punkcie decyzyjnym TS_i, należy określić zasady zatwierdzania harmonogramu obsługi zamówień. W rozprawie jest stosowana metoda najpóźniejszego możliwego zatwierdzania. Oznacza to, że harmonogram pojazdów, których powrót do bazy przewidywany jest w ostatnim przedziale czasowym (zgodnie ze wzorem (3.12)), należy uznać za całkowicie wypełniony. Postać (3.12) wynika bezpośrednio z warunku (3.3)

[2]Wartość N_{TS} przyjęta dla ContDVRP wynosi 40.

o powrocie do bazy po uwzględnieniu faktu podziału procesu optymalizacji DVRP na dyskretne przedziały czasowe.

$$\forall_{i\in\{1,2,\ldots n\}} arv_{r_{i,m_i}} \in (t_{end} - \Delta TS, t_{end}] \tag{3.12}$$

W związku z tym w każdym punkcie decyzyjnym TS_i trzeba zatwierdzić do realizacji zamówienia, które według aktualnego harmonogramu pojazdu, są zaplanowane do realizacji w kolejnym przedziale czasowym (3.13). Przy czym realizacja zamówienia rozpoczyna się w momencie, kiedy pojazd wyrusza w kierunku jego lokalizacji.

$$\bigcup_{i\in\{1,2,\ldots n\}} \left\{ r_{i,j} : j \leq \min_{k\in\{1,2,\ldots,m_i\}} \left\{ k : arv_{r_{i,k}} + u_{r_{i,k}} > TS_i + 2\Delta TS \right\} \right\} \tag{3.13}$$

Ze względu na charakter dynamizmu w DVRP uzyskiwane wyniki można poprawić poprzez dodawanie do planowanego czasu przejazdu bufora czasowego, który ma za zadanie zapewnić możliwość dodania kolejnych zamówień do pojazdu w trakcie procesu optymalizacji.

Definicja 25. ***Bufor czasowy*** T_{AC} *decyduje o momencie, w którym zamówienia muszą być najpóźniej zatwierdzone do realizacji zmieniając oznaczenie pojazdów o wypełnionym harmonogramie na:*

$$\forall_{i\in\{1,2,\ldots n\}} arv_{r_{i,m_i}} \in (t_{end} - (\Delta TS + T_{AC}), t_{end}] \tag{3.14}$$

W większości pozycji literaturowych T_{AC} jest równy 0, natomiast w pracy [47] jest równy $0.01 \times (t_{end} - t_{start})$. Podobnie jak dla parametru N_{TS}, praca [59] zaleca dostrojenie go do konkretnego algorytmu. Dla ContDVRP parametr ten został dostrojony[3] na zbiorze przypadków testowych (szczegółowe wyniki strojenia są zaprezentowane w Dodatku A.1)).

Wybór wartości N_{TS}, T_{AC} oraz przebieg $\mathcal{A}_R$ użytego algorytmu optymalizacyjnego $\mathcal{A}$ wpływa na współczynnik, który w rozprawie definiowany jest jako empiryczny stopień dynamizmu problemu.

Definicja 26. ***Empiryczny stopień dynamizmu problemu*** *(em.dod) określa minimalną liczbę zamówień nieznanych i zatwierdzonych w punktach decyzyjnych TS_i procesu optymalizacji w stosunku do liczby wszystkich zamówień.*

$$em.dod = \min_{i\in\{0,1,\ldots,N_{TS}\}} \frac{m - mp(TS_i)}{m} \tag{3.15}$$

Punkt decyzyjny TS_i, w którym osiągnięta jest wartość *em.dod* jest kluczowy z punktu widzenia procesu optymalizacyjnego. Algorytm $\mathcal{A}$ ma jednocześnie dostęp do informacji o lokalizacji i wielkości wielu zamówień, a także dla wielu z nich może zmienić przypisanie do pojazdów.

[3] Wartość T_{AC} przyjęta dla ContDVRP wynosi $0.04 \times (t_{end} - t_{start})$

Rozdział 4

Algorytmy heurystyczne w problemie marszrutyzacji

W tym rozdziale omówione są algorytmy heurystyczne wykorzystywane w problemach marszrutyzacji. Krótko scharakteryzowane zostaną zarówno samodzielnie wykorzystywane metody, jak i takie, które są wykorzystywane przez podejścia metaheurystyczne jako operatory lokalnego przeszukiwania i poprawy wyników. Szczegółowy opis będzie poświęcony metodom wykorzystywanym w podejściu autorskim, czyli algorytmowi 2–OPT [25] oraz zmodyfikowanemu algorytmowi Kruskala [62, 85].

Klasyfikacja algorytmów heurystycznych stosowanych do optymalizacji VRP jest przedstawiona w pracy przeglądowej [32]. Praca ta dzieli heurystyczne algorytmy rozwiązujące VRP na następujące grupy:

Najpierw przypisujące zamówienia a następnie konstruujące trasy.
Przykładem takiego podejścia jest zmodyfikowany algorytm k-średnich [44].

Najpierw konstruujące trasy a następnie przypisujące zamówienia.
Przykładem takiego podejścia jest algorytm płatkowy [111]),

Konstruujące trasy poprzez ich scalanie ze sobą.
Przykładem takiego podejścia jest pierwotny algorytm Dantziga i Ramsera [28].

Stosujące poprawianie iteracyjne.
Do tej grupy należą algorytm Clarke'a i Wrighta [23], λ-wymian [89], czy 2–OPT [25].

W podejściach metaheurystycznych stosowanych do DVRP, omawianych w Rozdziale 6, poza algorytmem 2–OPT oraz λ-wymian jako pomocnicze algorytmy wykorzystywane są: zachłanne wstawianie [38, 60] oraz przeszukiwanie dużego sąsiedztwa (ang. *Large Neighbourhood Search* (LNS)) [96] dostosowane do VRP.

4.1 Przykładowe algorytmy heurystyczne

W tym podrozdziale krótko scharakteryzowane są wybrane algorytmy rozwiązujące VRP. Ich szczegółowy opis można znaleźć w [54]. W pierwszej kolejności opisane są algorytmy konstrukcyjne, nie wymagające początkowego poprawnego rozwiązania:

Zmodyfikowany algorytm k-średnich [44]

Algorytm rozwiązuje problem grupowania z ograniczeniem wielkości (wagi) skupiska (ang. *Capacitated Clustering Problem* (CCP)) i znajduje podział zamówień pomiędzy k pojazdów. Od standardowego algorytmu k-średnich, poza ograniczeniem na łączną wagę (lub liczbę) obiektów znajdujących się w jednej grupie, odróżnia go: (1) sposób inicjalizacji centroidów skupisk oraz (2) sposób liczenia odległości pomiędzy centroidem a obiektem. Centroidy inicjalizowane są w lokalizacji k obiektów o największej wadze. Natomiast odległość pomiędzy centroidem a obiektem jest dzielona przez wagę obiektu (czyli w przypadku VRP wielkość zamówienia). Algorytm ten musi być uzupełniany algorytmem rozwiązującym TSP w każdym ze wskazanych skupisk zamówień.

Algorytm płatkowy [111]

Ten algorytm operuje na koncepcji płatków, czyli tras utworzonych ze spójnych maksymalnych podzbiorów zamówień. Spójność podzbioru, w podstawowej wersji algorytmu, jest zdefiniowana w ten sposób, że do podzbioru zamówień należą wszystkie (i wyłącznie) takie zamówienia, których współrzędna kątowa znajduje się pomiędzy dwiema wybranymi wartościami kątów. Podzbiór jest maksymalny, kiedy dodanie do niego jakiegokolwiek zamówienia, nie naruszającego warunku spójności, powodowałoby przekroczenie ograniczenia czasowego (3.3) lub pojemnościowego (3.8).
Po skonstruowaniu wszystkich płatków, wybierany jest ich podzbiór pokrywający zbiór zamówień oraz minimalizujący sumaryczną długość tras pojazdów.

Algorytm Dantziga i Ramsera [28]

Algorytm jest iteracyjnym algorytmem konstrukcyjnym, dążącym do maksymalizacji wypełnienia poszczególnych pojazdów. W kolejnych krokach algorytmu łączone są ze podzbiory zamówień, nie przekraczające zadanego w danej fazie rozmiaru, przy czym dopuszczalny łączny rozmiar zamówień rośnie hiperbolicznie, aż do osiągnięcia pojemności pojazdu.

Algorytm Clarke'a i Wrighta [23]

Algorytm, pierwotnie wprowadzony jako metoda poprawy tras znajdowanych algorytmem Dantziga-Ramsera, w każdej iteracji wyszukuje pary krawędzi, których usunięcie a następnie zastąpienie inną parą (lub pojedynczą krawędzią), przyniesie największy zysk. Ta sama procedura wyszukiwania największego zysku jest stosowana do konstrukcji rozwiązania. Znalezienie rozwiązania rozpoczyna się od utworzenia

tras zawierających pojedyncze pojazdy. Następnie rozważane są zyski z usuwania wyłącznie par krawędzi, należących do tras różnych pojazdów, o końcach w zajezdni.

Zachłanne wstawianie [38, 60]

Algorytm konstruuje trasy sekwencyjnie. Dla aktualnie budowanej trasy wyszukuje on zamówienia, nie przypisane do żadnego z pojazdów, minimalizujące koszt wstawienia do trasy aktualnie rozpatrywanego pojazdu. W przypadku braku zamówień nie powodujących przekroczenia ograniczenia czasowego (3.3) lub pojemnościowego (3.8), rozpoczynana jest konstrukcja trasy nowego pojazdu, inicjalizowanej losowym nieprzypisanym zamówieniem.

Drugą grupę stanowią algorytmy, które rozpoczynają od rozwiązania początkowego, które jest następnie iteracyjnie poprawiane:

Algorytm λ-wymian [89]

Algorytm wyszukuje pary pojazdów, pomiędzy którymi zamiana nie więcej niż λ zamówień, przyniesie największy zysk. Dla konkretnej wartości parametru λ algorytm rozważa następujący zbiór (i,j) liczb zamówień do wymiany pomiędzy każdymi z dwóch pojazdów: (0,1), (0,2), ... (0, λ), (1, 0), (1, 1), ... (0, λ), ..., (λ, λ). Dla każdej pary pojazdów algorytm oblicza zysk z usunięcia i zamówień z pierwszego pojazdu oraz j z drugiego, a następnie zachłannego wstawienia j usuniętych zamówień do pierwszego pojazdu oraz i usuniętych zamówień do drugiego pojazdu. Zalecanym sposobem wykorzystania algorytmu jest iteracyjne zwiększanie wartości parametru λ w celu uzyskiwania wyników coraz wyższej jakości zaczynając od najmniej kosztownych obliczeniowo operacji.

Algorytm LNS dla VRP [33]

Idea algorytmów LNS polega na jednoczesnej modyfikacji wielu zmiennych w proponowanym rozwiązaniu. Omawiana wersja dla VRP dokonuje wyboru pewnej liczby zamówień, które mają zostać ponownie przypisane w oparciu o następujące zasady: (1) pierwsza połowa zamówień wybierana jest losowo (z rozkładu jednostajnego), (2) druga połowa zamówień wybierana jest zgodnie z prawdopodobieństwem związania z jednym z zamówień z pierwszej połowy. Związanie zamówień jest odwrotnie proporcjonalne do odległości i wzmacniane przynależnością do różnych pojazdów. Innymi słowy, za najsilniej związane ze sobą uważane są zamówienia znajdujące się blisko siebie, ale przypisane do różnych pojazdów.

Ponowne przypisanie zamówień odbywa się zgodnie z zasadami zachłannego wstawiania.

Dalsza część rozdziału poświęcona jest opisowi metod kluczowych dla działania algorytmu ContDVRP.

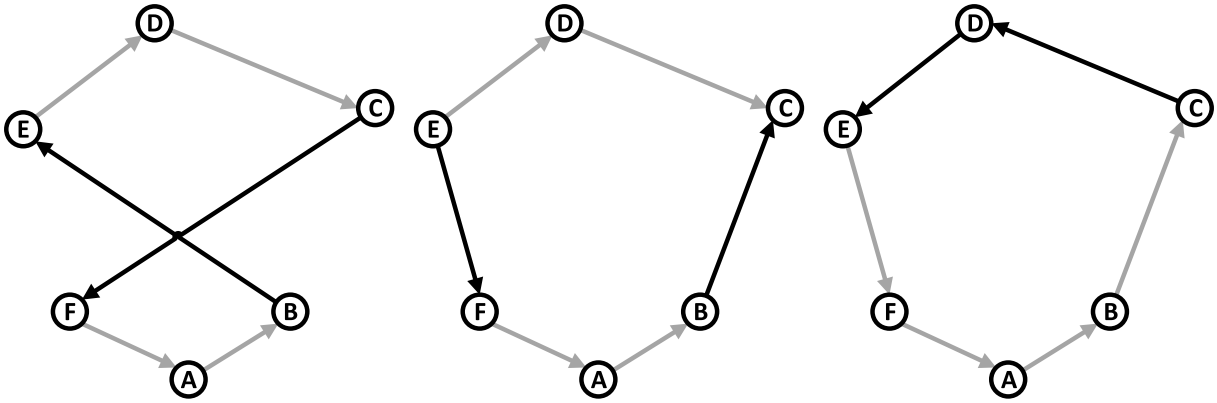
4.1.1 2–OPT

Algorytm 2–OPT [25] jest algorytmem iteracyjnym, pierwotnie stosowanym do optymalizacji rozwiązań w problemie komiwojażera. Jego główną cechą jest zdolność do skracania tras, w których występują zapętlenia.

Algorytm 4.1 Pseudokod algorytmu 2–OPT.

```
    {E zbiór ważonych krawędzi skierowanego cyklu, których wagi reprezentują odległość
    między węzłami}
 1: while improvement do
 2:   improvement ← FALSE
 3:   for all (v1, v2) ∈ E do
 4:     for all (v3, v4) ∈ E do
 5:       if w(v1, v3) + w(v2, v4) ≤ w(v1, v2) + w(v3, v4) then
 6:         v2 ↔ v3
 7:         ReversePath(v2, v3)
 8:         improvement ← TRUE
 9:         goto 1
10:       end if
11:     end for
12:   end for
13: end while
```

Algorytm 4.1 prezentuje pseudokod dla 2–OPT. Algorytm w kolejnych iteracjach głównej pętli (linia 1), poszukuje par krawędzi (v_1, v_2) i (v_3, v_4), w których zamiana v_2 z v_3 (linia 6) skróci długość cyklu. Oprócz zamiany miejsc wierzchołków w permutacji opisującej pierwotny cykl, należy odwrócić przebieg ścieżki pomiędzy nimi (linia 7). Przykład wykonania pojedynczej operacji weryfikacji i wykonania poprawy długości (linie 5–10) jest przedstawiony na Rysunku 4.1.



Rysunek 4.1: Operacja pojedynczej poprawy długości skierowanego cyklu w algorytmie 2–OPT. Najpierw następuje weryfikacja czy wymiana końców pomiędzy krawędziami CF i BE skróci długość cyklu. Następnie wierzchołki C i E są zamieniane ze sobą, co powoduje powstanie tymczasowego acyklicznego grafu. W ostatnim kroku zamieniany jest kierunek ścieżki łączącej wierzchołki C i E, aby utworzyć poprawny skierowany cykl.

Algorytm 2–OPT był wykorzystywany jako pomocnicza metoda poprawiająca wyniki działania algorytmów w pracy prezentującej GA i TS dla DVRP [47]. Natomiast w algorytmach DAPSO/MAPSO/MEMSO [54–58] oraz 2PSO/2MPSO/ContDVRP [83, 84] zadaniem algorytmu 2–OPT jest konstrukcja tras dla optymalizowanego podziału zamówień pomiędzy pojazdy.

Ma koniec warto nadmienić, że w literaturze funkcjonują również popularne uogólnienia algorytmu 2–OPT. Algorytm k–OPT dla TSP [67] w jednym kroku testuje możliwość zysku z zamiany k krawędzi w cyklu. Algorytm 2–OPT* jest dostosowany do VRP [98] i testuje możliwość zysków z zamiany krawędzi pomiędzy różnymi cyklami, w sposób zbliżony do rozważań z pracy [23].

4.1.2 Minimalne drzewo rozpinające

Algorytm Kruskala [62] znajdujący minimalne drzewo rozpinające grafu, oryginalnie został zaproponowany do rozwiązywania problemu komiwojażera. W kontekście tej rozprawy jego modyfikacja jest wykorzystana jedynie do podziału zamówień pomiędzy pojazdy. Algorytm ten jest wykorzystywany w sposób podobny do zmodyfikowanego algorytmu k-średnich, czyli rozwiązania CCP. Kolejność łączenia w minimalnym drzewie rozpinającym jest tożsama z hierarchiczną klasteryzacją w oparciu o kryterium minimalnej odległości (ang. *single-link, MIN link*) [51].

Algorytm 4.2 Pseudokod zmodyfikowanego algorytmu Kruskala do rozwiązywania CCP.

```
    {E := zbiór ważonych krawędzi grafu pełnego, których wagi reprezentują odległość
    między węzłami}
    {V := zbiór ważonych wierzchołków osadzonych na płaszczyźnie ℝ²}
    {CAPACITY := stała definiująca maksymalną sumę wierzchołków w składowej wy-
    nikowego grafu}
    {DEPOT := wyróżniony wierzchołek, definiujący położenie zajezdni}
 1: E_sort ← SortByWeightInAscendingOrder(E)
 2: T_clusters ← CreateSeparateTrees(V)
 3: for all (v1, v2) ∈ E_sort do
    {Tree(v) := drzewo do którego należy v}
 4:   if Tree(v1) ≠ Tree(v2) then
 5:     if SumNodesW(Tree(v1)) + SumNodesW(Tree(v2)) ≤ CAPACITY then
 6:       T_clusters ← T_clusters \ {Tree(v1), Tree(v2)}
 7:       T_clusters ← T_clusters ∪ {Tree(v1) ∪ Tree(v2)}
 8:     end if
 9:   end if
10: end for
```

Algorytm 4.2 prezentuje pseudokod zmodyfikowanego algorytmu Kruskala, znajdującego podział zamówień pomiędzy pojazdy dla problemu VRP. W problemie VRP do przetwarzanego zbioru wierzchołków V należą wszystkie zamówienia oczekujące oraz ostatnie

zatwierdzone zamówienie dla każdego z pojazdów. Procedura $CreateSeparate\ Trees(V)$ (linia 2) tworzy zbiór drzew składający się z grafów o jednym wierzchołku (dla zamówień oczekujących) oraz ścieżek definiujących trasy pojazdów (dla zamówień już zatwierdzonych). Główna pętla algorytmu (linia 3) oraz podstawowy warunek łączenia podgrafów zapobiegający powstawaniu cyklów (linia 4) pozostają niezmienione względem oryginalnego algorytmu. Dokonane w algorytmie zmiany służą tworzeniu skupisk zamówień (podgrafów) o łącznej sumie wielkości zamówień (wag wierzchołków) nie przekraczającej pojemności pojazdu (linia 5).

Zbiór drzew, który jest wynikiem działania zmodyfikowanego algorytmu Kruskala, może być wykorzystany do dokonania podziału zamówień pomiędzy pojazdy. Jednak należy wziąć pod uwagę, że dokonany w ten sposób podział nie daje gwarancji znalezienia tras pojazdów spełniających ograniczenie (3.3).

Rozdział 5

Kodowanie DVRP na potrzeby optymalizacji

W tym rozdziale omówione są sposoby kodowania wykorzystywane w podejściach metaheurystycznych rozwiązujących problem DVRP (prezentowane w literaturze przedmiotu oraz autorskie). W zależności od wybranego podejścia kodowanie ma charakter wektora liczb całkowitych lub rzeczywistych i obrazuje całość rozwiązania lub tylko jego fragment (podział zamówienia na pojazdy, kolejność zamówień), a kompletne rozwiązanie (tj. zbiór permutacji R opisany w podrozdziale 3.3) jest generowane zgodnie z ustalonymi regułami (np. heurystyczny algorytm znajdowania trasy pojazdu, zachłanny podział trasy pomiędzy pojazdy).

Zgodnie ze spostrzeżeniami z pracy [110] kodowanie może mieć wpływ na wielkość zmian istotnych z punktu widzenia optymalizacji, zachodzących pod wpływem zmian w problemie. Aby zaprezentować wpływ wyboru kodowania wykorzystana jest różnica w estymowanych przez algorytm położeniach optimum. Jest to jedna z miar proponowanych w pracy [16], które zostały opisane w Rozdziale 1. Pod koniec tego rozdziału zaproponowane jest sformułowanie tej miary dla problemu DVRP.

Tabela 5.1: Dodatkowe symbole do opisu DVRP w poszczególnych algorytmach

Symbol	Typ	Opis
k	$\mathbb{Z}_+$	Liczba skupień zamówień przypadających na pojazd
$\hat{n}$	$\mathbb{Z}_+$	Estymowana liczba pojazdów
$rank_i$	$\mathbb{R}$	Ranga i-tego zamówienia
$v(i)$	$\mathbb{Z}_+$	Pojazd obsługujący i-te zamówienie
$v_i.o_j$	$\mathbb{R}$	o-współrzędna (x lub y) j-tego skupienia zamówień przypisanego do i-tego pojazdu
$C_U(t)$	Ciąg	Zamówienia nieznane w chwili t
$C_P(t)$	Ciąg	Zamówienia oczekujące w chwili t
$C_C(t)$	Ciąg	Zamówienia zatwierdzone w chwili t

Następujące kodowania były wykorzystywane przy rozwiązywaniu DVRP:

- ciąg identyfikatorów lokalizacji (łącznie z lokalizacją zajezdni) [33, 34, 75],
- ciąg identyfikatorów lokalizacji oraz identyfikatorów tych pojazdów, które mają już zatwierdzone zamówienia [47],
- wektor przyporządkowań zamówień do pojazdów [54–58],
- wektor centroidów klastrów zamówień przypisywanych do jednego pojazdu wraz z osobnym wektorem rang decydującym o kolejności obsługi zamówień [1, 2, 83–85].

W dalszej części rozdziału omówione są poszczególne rodzaje kodowania, zaprezentowane w Tabelach 5.2 i 5.3, odpowiednio dla sytuacji z oczekującymi i zatwierdzonymi zamówieniami. Omawiane rozwiązania są też prezentowane graficznie na wspólnym przykładzie, przedstawionym na Rysunkach 5.1 i 5.2. Rozdział omawia też rozmiary przestrzeni przeszukiwań indukowanych przez dany rodzaj kodowania oraz sposób uzyskiwania kompletnych rozwiązań w formie zbioru tras R.

5.1 Kodowania dyskretne

W tym rozdziale zaprezentowane są dyskretne kodowania wykorzystywane w literaturowych podejściach do DVRP.

5.1.1 Ciąg identyfikatorów lokalizacji (CIL)

Wektor kodujący DVRP poprzez CIL składa się zarówno z identyfikatorów zamówień jak i identyfikatorów zajezdni. W wektorze znajdują się identyfikatory wszystkich oczekujących zamówień (czyli maksymalnie m identyfikatorów) oraz $\hat{n}$ wystąpień identyfikatora zajezdni, które przy zatwierdzaniu zamówień zostaną zastąpione przez identyfikator ostatniego zatwierdzonego zamówienia dla danego pojazdu.

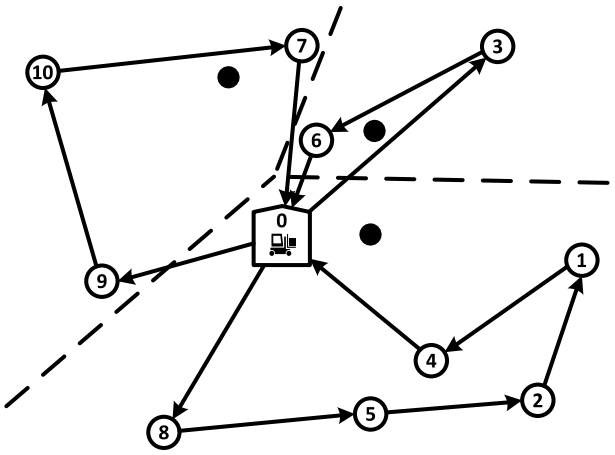
$$(r_{1,2}, r_{1,3}, \ldots, r_{1,m_1}, \ldots, r_{\hat{n},m_{\hat{n}}}) \in \mathbb{Z}^{m+\hat{n}}$$

Kodowanie ma charakter bezpośredni, gdyż opisuje pełną trasę oczekujących zamówień każdego pojazdu i dzieli zamówienia pomiędzy pojazdy poprzez oznaczanie wyjazdu kolejnego pojazdu z bazy (w przypadku gdy pojazd nie ma zatwierdzonych zamówień) lub ostatnie zatwierdzone zamówienie danego pojazdu. Dla przykładów zaprezentowanych na Rysunkach 5.1 (brak zatwierdzonych zamówień) i 5.2 (zatwierdzone zamówienia w każdym z pojazdów) są to odpowiednio ciągi: 0–3–6–0–8–5–2–1–4–0–9–10–7 oraz 3–6–5–2–1–4–10–7.

Pierwotny artykuł [75] nie precyzował w jaki sposób oznaczane są zatwierdzone zamówienia, ale bazujące na nim prace [33, 34] wskazują, że zatwierdzone fragmenty tras są skracane do ostatniego zatwierdzonego punktu dla danego pojazdu.

Tabela 5.2: Porównanie kodowania DVRP dla Rysunku 5.1

Algorytm	Kodowanie	
GA [34], ACO [75], ACOLNS [33]	0 | 3 | 6 | 0 | 8 | 5 | 2 | 1 | 4 | 0 | 9 | 10 | 7	CIL
GA [47], Tabu [47]	3 | 6 | 8 | 5 | 2 | 1 | 4 | 9 | 10 | 7	CIZ
MEMSO [58], VNS [57]	2 | 2 | 1 | 2 | 2 | 1 | 3 | 2 | 3 | 3	WPP
ContDVRP [84, 87, 88]	1.0 | 1.4 || 1.0 | 0.0 || -0.5 | 2.1	WCZ
	0.8 | 0.6 | 0.5 | 1.0 | 0.4 | 1.0 | 1.0 | 0.2 | 0.3 | 0.7	WRZ



Rysunek 5.1: Przykład DVRP z oczekującymi zamówieniami. Czarne punkty generują podział płaszczyzny (i w efekcie zamówień) pomiędzy pojazdy.

5.1.2 Ciąg identyfikatorów zamówień (CIZ)

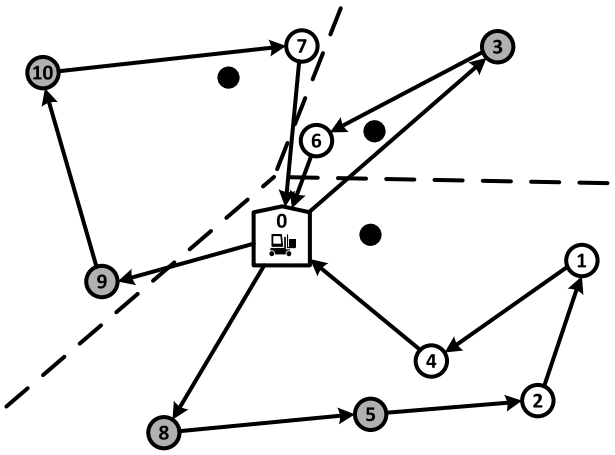
Wektor kodujący DVRP poprzez CIZ składa się wyłącznie z identyfikatorów zamówień. Jego maksymalny rozmiar jest zatem równy liczbie wszystkich zamówień m. Ten rodzaj kodowania wymaga zdefiniowania operacji dekodowania na zbiór tras R, gdyż zapisuje rozwiązanie w formacie pojedynczej trasy TSP, bez podziału na pojazdy dopóki nie ma zatwierdzonych zamówień.

$$(r_{1,2}, r_{1,3}, \ldots, r_{1,m_1-1}, \ldots, r_{n,m_n-1}) \in \mathbb{Z}^m$$

To kodowanie nie definiuje trasy bezpośrednio, dopóki każdy z wykorzystywanych pojazdów nie ma zatwierdzonych zamówień. Zamówienia są przydzielane do tego samego pojazdu dopóki jest to możliwe, a kiedy pojazd jest przepełniony rozpoczynana jest nowa trasa. Alternatywnie rozpoczęcie nowej trasy jest możliwe, jeżeli w ciągu pojawią się zatwierdzone zamówienia, reprezentowane przez ujemne identyfikatory pojazdów, do których są przypisane. Dla przykładów zaprezentowanych na Rysunkach 5.1 (brak zatwierdzonych zamówień) i 5.2 (zatwierdzone zamówienia w każdym z pojazdów) są to odpo-

Tabela 5.3: Porównanie kodowania VRP dla Rysunku 5.2

Algorytm	Kodowanie	
GA [34], ACO [75], ACOLNS [33]	3 | 6 | 0 | 5 | 2 | 1 | 4 | 10 | 7	CIL
GA [47], Tabu [47]	-1 | 6 | -2 | 2 | 1 | 4 | -3 | 7	CIZ
MEMSO [58], VNS [57]	2 | 2 | 1 | 2 | 2 | 1 | 3 | 2 | 3 | 3	WPP
ContDVRP [84, 87, 88]	1.0 | 1.4 || 1.0 | 0.0 || -0.5 | 2.1	WCZ
	0.7 | 0.3 | 1.0 | 1.0 | 1.0	WRZ



Rysunek 5.2: Przykład stanu DVRP z zatwierdzonymi zamówieniami oznaczonymi szarym kolorem

wiednio ciągi: 3–6–8–5–2–1–4–9–10–7 oraz (-1)–6–(-2)–2–1–4–(-3)–7. Należy zauważyć, że możliwość zakodowania tego konkretnego rozwiązania bez zatwierdzonych zamówień zakłada, że odpowiednie sumy wielkości zamówień $s_3 + s_6$ i $s_8 + s_5 + s_2 + s_1 + s_4$ są nie większe niż pojemność pojazdu *cap*, natomiast po dodaniu wielkości następnego zamówienia z ciągu lokalizacji spowoduje przekroczenie pojemności pojazdu: $s_3 + s_6 + s_8 > cap$ i $s_8 + s_5 + s_2 + s_1 + s_4 + s_9 > cap$.

Ten sposób kodowania jest wykorzystany w pracach [47, 71]. W literaturze dotyczącej VRP (np. pracach Prinsa [99, 100] czy Goksal i in. [45]), ten rodzaj kodowania jest określany jako *giant TSP tours*.

5.1.3 Wektor przyporządkowań do pojazdów (WPP)

Wektor kodujący DVRP poprzez WPP składa się z identyfikatorów pojazdów $\{1, 2, \ldots \hat{n}\}$. Każde z zamówień jest przypisane do dokładnie jednego pojazdu, a zatem wektor ten ma maksymalny rozmiar równy m. Ze względu na to, że kodowanie to obejmuje wyłącznie

przypisanie zamówień do pojazdów wymaga ono dodatkowo metody generowania tras dla zbiorów zamówień, czyli rozwiązywania niezależnych TSP dla każdego z pojazdów.

$$(v(1), v(2), \ldots, v(m)) \in \mathbb{Z}^m$$

Kodowanie definiuje wyłącznie identyfikatory pojazdów, które mają obsłużyć dane zamówienie. W trakcie zmiany przypisania zamówienia do pojazdu, zamówienie jest umieszczane w trasie pojazdu w sposób zachłanny, minimalizując zwiększenie długości trasy pojazdu. W dalszej kolejności trasy są optymalizowane algorytmem 2–OPT i w ten sposób uzyskiwana jest finalna postać R. Wraz z zatwierdzaniem zamówień wektor reprezentujący rozwiązanie nie jest skracany, ale wprowadzana jest maska uniemożliwiająca zmianę przypisań dla zatwierdzonych zamówień. Dla przykładów zaprezentowanych na Rysunkach 5.1 (brak zatwierdzonych zamówień) i 5.2 (zatwierdzone zamówienia w każdym z pojazdów) jest to ten sam wektor przypisań identyfikatorów pojazdów: 2–2–1–2–2–1–3–2–3–3. Wektor ten wygeneruje rozwiązania zaprezentowane na tych rysunkach pod warunkiem odpowiedniej początkowej kolejności zamówień. W przypadku innej początkowej kolejności zamówień, poza prostym odwróceniem kolejności zamówień na trasach, działanie algorytmu 2–OPT mogłoby doprowadzić do stworzenia trasy 0–8–5–4–2–1–0, w miejsce 0–8–5–2–1–4–0.

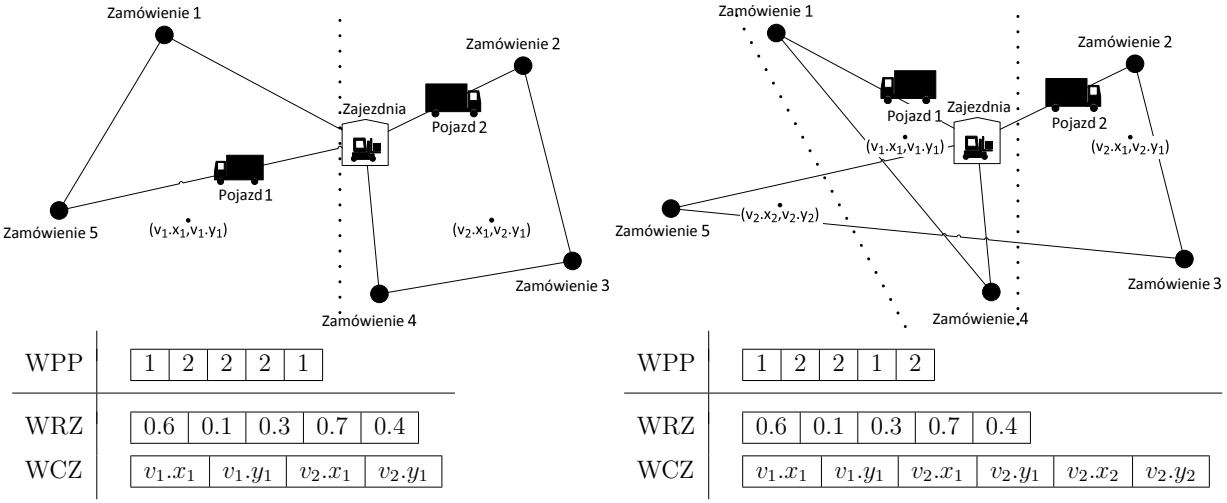
Ten sposób kodowania jest wykorzystywany w pracach [54–58].

5.2 Kodowania ciągłe

W tym podrozdziale zaprezentowane są ciągłe kodowania wykorzystywane przez algorytm ContDVRP.

5.2.1 Wektor centroidów skupisk zamówień (WCZ)

Wektor kodujący DVRP poprzez WCZ składa się z lokalizacji centroidów skupisk zamówień indukujących podział zamówień na grupy. Zamówienia znajdujące się w tej samej grupie są przypisywane do tego samego pojazdu. Aby umożliwić przypisywanie do tego samego pojazdu zamówień nie tworzących spójnych liniowo separowalnych skupisk kodowanie zakłada przypisanie k grup do jednego pojazdu. Przy liczbie pojazdów koniecznej do realizacji zbioru zamówień $\hat{n}$, wektor kodujący ma rozmiar $2k\hat{n}$. Rysunek 5.3 prezentuje możliwości i ograniczenia kodowania WCZ w zależności od wybranej liczby skupień k. Analogicznie jak dla kodowania WPP wymagane jest określenie sposobu budowania



WPP	1	2	2	2	1
WRZ	0.6	0.1	0.3	0.7	0.4
WCZ	$v_1.x_1$	$v_1.y_1$	$v_2.x_1$	$v_2.y_1$	

WPP	1	2	2	1	2	
WRZ	0.6	0.1	0.3	0.7	0.4	
WCZ	$v_1.x_1$	$v_1.y_1$	$v_2.x_1$	$v_2.y_1$	$v_2.x_2$	$v_2.y_2$

Rysunek 5.3: Przykład instancji VRP z 2 pojazdami i 5 zamówieniami. Linie ciągłe prezentują trasy. Linie przerywane prezentują podział skupisk zamówień indukowany przez kodowanie WCZ. Centroidy skupisk są oznaczone przez $(v_n.x_k, v_n.y_k)$. Prawy rysunek prezentuje propozycję rozwiązania niereprezentowalną przy użyciu kodowania WCZ z jednym skupiskiem zamówień na pojazd. Pod rysunkami zaprezentowane są kodowania WPP,WCZ i WRZ opisujące rozwiązania.

tras dla zbioru zamówień przypisanego do jednego pojazdu.

$$(v_1.x_1, v_1.y_1, \ldots, v_1.x_k, v_1.y_k, \ldots, v_{\hat{n}}.x_k, v_{\hat{n}}.y_k) \in \mathbb{R}^{2k\hat{n}}$$

Kodowanie definiuje podział zamówień pomiędzy pojazdy poprzez grupowanie lokalizacji, w których znajdują się zamówienia. Obszar, na którym zlokalizowane są zamówienia, jest dzielony pomiędzy pojazdy odpowiadające poszczególnym centroidom, tak jak zaprezentowane jest to na Rysunkach 5.1, 5.2 i 5.3. Lokalizacje centroidów oznaczone są czarnymi punktami, zaś indukowany przez nie podział obszaru przerywanymi liniami. Wszystkie zamówienia zlokalizowane w jednym podobszarze przypisywane są do tego samego pojazdu. Rysunek 5.3 prezentuje dodatkowo możliwości kodowania podziałów zamówień wynikające ze stosowania $k > 1$ grup zamówień przypisanych do jednego pojazdu.

W podstawowej wersji algorytmu ContDVRP [83, 84], podobnie jak dla kodowania WPP, algorytm 2–OPT jest wykorzystywany do znalezienia finalnej postaci zbioru permutacji R. W aktualnej wersji ContDVRP, na wzór prac [1, 2], może być ono połączone z kodowaniem ciągu zamówień, tym samym jednoznacznie definiując zbiór R.

5.2.2 Wektor rang zamówień (WRZ)

Wektor kodujący DVRP poprzez WRZ jest ciągłym odpowiednikiem wektora CIZ. Jego maksymalny rozmiar jest także równy liczbie wszystkich zamówień m. Analogicznie jak

CIZ, kodowanie WRZ wymaga zdefiniowania sposobu podziału zamówień pomiędzy pojazdy. Kolejność zamówień definiowana jest poprzez posortowanie indeksów wektora według ich wartości (co w efekcie tworzy wektor kodowania CIZ).

$$(rank_1, rank_2, \ldots, rank_m) \in \mathbb{R}^m$$

W algorytmach rozwiązujących DVRP wektor rang zamówień stanowi uzupełnienie dla kodowań opierających się na przypisywaniu pojazdów do zamówień. Oczekujące zamówienia przypisane do pojazdu są sortowane zgodnie z rangami przypisanymi w ciągu kodującym. Zamówienia zatwierdzone nie są uwzględniane w wektorze kodującym, gdyż ich kolejność jest już ustalona w rozwiązaniu i znajdują się one w harmonogramie pojazdu przed zamówieniami oczekującymi. W przykładzie zaprezentowanym na Rysunku 5.1, zamówienia (w podziale pomiędzy pojazdy) mają przypisane następujące wartości rang: $\{3 \to 0.5, 6 \to 1.0\}, \{1 \to 0.8, 2 \to 0.6, 4 \to 1.0, 8 \to 0.2\}, \{7 \to 1.0, 9 \to 0.3, 10 \to 0.7\}$. Posortowanie zamówień według wartości rang w każdym zbiorze osobno, wygeneruje trasy 0–8–5–2–1–4–0, 0–3–6–0, 0–9–10–7–0 prezentowane na tym rysunku.

Dwa powyższe ciągłe sposoby kodowania (WCZ i WRZ) zostały zaproponowane dla DVRP i opisane w pracach prezentujących częściowe wyniki tej rozprawy [83, 84, 86–88]. Należy zauważyć, że zastosowanie tego rodzaju kodowania otwiera problem DVRP na zastosowanie dowolnych populacyjnych algorytmów ciągłej optymalizacji globalnej. Praca [83] stanowiła pierwsze zastosowanie kodowania ciągłego dla DVRP, natomiast praca [84] wprowadza koncepcję wieloskupiskowego wariantu WCZ do dziedziny problemów marszrutyzacji.

Niezależnie od wymienionych prac zbliżone kodowanie (z rankingiem i centroidami pojedynczych klastrów połączonymi w jeden wektor o łącznej o długości $m + 2\hat{n}$) było stosowane do problemów z rodziny VRP przez Ai i Kachitvichyanukula [1, 2]. Również samo kodowanie WRZ, bez dodatkowego bezpośredniego kodowania podziału zamówień, było stosowane do różnych wariantów VRP przez Marinakisa i in. [72, 73].

5.3 Miara zmienności problemu

Do pomiaru różnic w dynamizmie problemu, w zależności od kodowania i sposobu optymalizacji, zaadaptowana jest miara zmienności polegająca na pomiarze różnicy pomiędzy rozwiązaniami wskazywanymi w kolejnych krokach czasowych algorytmu. Można argumentować, że najistotniejszą częścią rozwiązania (z operacyjnego punktu widzenia) jest przypisanie zamówienia do konkretnego pojazdu, ze względu na to, że ewentualna zmiana kolejności zamówień może być dokonana na poziomie pojedynczego kierowcy i może nie wymagać (w zależności od praktyki biznesowej) ani informowania koordynatora ani innych kierowców.

Aby uchwycić tę najistotniejszą część rozwiązania miara jest liczona na rozwiązaniach prezentowanych w formie wektora przyporządkowań zamówień do pojazdów. W celu uproszczenia obliczeń rozwiązania izomorficzne nie są rozróżniane. Miara ρ różnicy między rozwiązaniami x_j i x_k jest liczona jako znormalizowana liczba zamówień (znanych zarówno w czasie t_j jak i t_k) przypisanych w x_j i x_k do różnych pojazdów. Normalizacja jest wykonywana poprzez podzielenie tej liczby różnych zamówień przez liczbę zamówień znanych zarówno w czasie t_j jak i t_k.

$$\rho(x_j, x_k) = \frac{\sum_{i \notin C_U(t_j) \wedge i \notin C_U(t_k)} I\left(x_{j,i} \neq x_{k,i}\right)}{|\{i : i \notin C_U(t_j) \wedge i \notin C_U(t_k)\}|}, \text{ gdzie funkcja } I \text{ oznacza indykator} \tag{5.1}$$

Przy prezentacji wyników miara ta została zastosowana zarówno do pokazania poziomu zmian w estymowanym przez algorytm rozwiązaniu optymalnym pomiędzy dwoma kolejnymi stanami problemu, jak i rozwiązaniami z każdego kolejnego kroku czasowego a rozwiązaniem finalnym. Porównanie wyników dla różnych wariantów algorytmu Cont-DVRP (wykorzystującego ciągłe kodowanie WCZ) oraz własnej implementacji algorytmu GA [47] (wykorzystującej dyskretne kodowanie CIZ) znajduje się w dalszej części rozprawy na Rysunku 8.5.

Rozdział 6

Algorytmy metaheurystyczne i transfer rozwiązań w problemie dynamicznej marszrutyzacji

W tym rozdziale opisane są podejścia metaheurystyczne stosowane do rozwiązywania problemu DVRP oraz metody transferu rozwiązań wykorzystywane w autorskim podejściu ContDVRP. Podejścia literaturowe opierają się na adaptacji danej metaheurystyki do DVRP, poprzez odpowiedni dobór przestrzeni przeszukiwań oraz operatorów algorytmu. Zaproponowane podejście ContDVRP, wykorzystujące ciągłe kodowania, otwiera zagadnienia DVRP na stosowanie dowolnych populacyjnych metaheurystyk optymalizacji ciągłej, co jest zaprezentowane z wykorzystaniem optymalizacji rojem cząstek i ewolucji różnicowej.

6.1 Populacyjne algorytmy metaheurystyczne

Za podręcznikiem i wykładami Arabasa [5, 6], metaheurystyka jest w rozprawie rozumiana jako pewien ramowy (niezależny od rozwiązywanego problemu) sposób przeszukiwania przestrzeni rozwiązań. W sposobie tym należy określić (zależne już od problemu) kodowanie rozwiązania, operatory przeszukiwania tej przestrzeni oraz rozmiar pamięci algorytmu.

Ze względu na ograniczenia algorytmów heurystycznych co do charakteru znajdowanych rozwiązań oraz wykładniczą złożoność algorytmów dokładnych dla problemów typu VRP [64], alternatywną metodą jest zastosowanie podejścia metaheurystycznego. Podejścia te można podzielić na metody jednopunktowe (symulowane wyżarzanie, przeszukiwanie ze zmiennym sąsiedztwem) oraz populacyjne (algorytmy ewolucyjne, algorytmy rojowe). Omówione są podejścia populacyjne, ponieważ uzyskiwane dla nich wyniki przewyższają te dla podejść jednopunktowych. Obserwowana w praktyce przewaga podejść

populacyjnych, uzasadniona istotnością utrzymywania różnorodnych rozwiązań w problemach dynamicznych, poparta jest również pracami teoretycznymi w tym obszarze [27].

W dalszej części rozdziału zaprezentowane są: algorytm mrówkowy, algorytm genetyczny, optymalizacja rojem cząstek oraz ewolucja różnicowa. Te algorytmy populacyjne należą do najpowszechniej wykorzystywanych w literaturze przedmiotu lub zostały wykorzystywane w badaniach przeprowadzonych w tej rozprawie. Prezentowane dla tych algorytmów przykłady operatorów, nawiązują do wspólnego przykładu stosowanego przy prezentacji różnic w kodowaniu, przedstawionego na Rysunku 5.1 na stronie 39. Opisy algorytmów odnoszą się również do sposobu ich wykorzystania w problemach marszrutyzacji i metodach przekształcenia rozwiązań na zbiór tras (permutacji lokalizacji) R, który został zdefiniowany w podrozdziale 3.3.

6.1.1 Algorytm mrówkowy

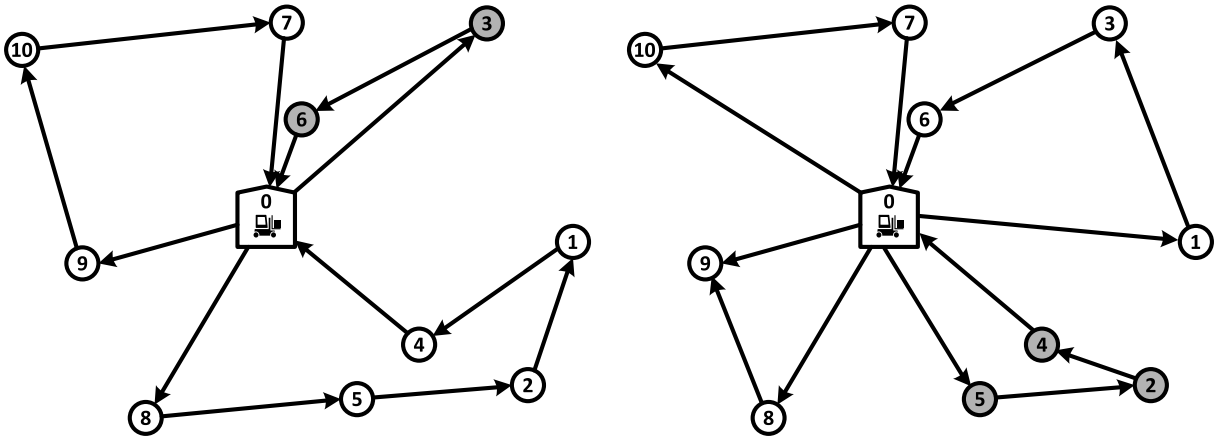
Rozwiązywanie DVRP algorytmem mrówkowym (ang. *Ant Colony System* (ACS) / *Ant Colony Optimization* (ACO)), zostało oparte na algorytmie analogicznym do oryginalnego podejścia dla TSP [39], zastosowanym do problemu marszrutyzacji z oknami czasowymi [40].

Oryginalny algorytm mrówkowy stanowi jedną z realizacji idei inteligencji rojowej, czyli prostych bytów (w tym wypadku zwanych mrówkami) znajdujących potencjalnie skomplikowane rozwiązania poprzez komunikację i proste reguły decyzyjne. Idea algorytmu mrówkowego opiera się na symultanicznym losowym przeszukiwaniu grafu wgłąb. Wybór kolejnego nieodwiedzonego wierzchołka dokonywany jest w oparciu o największe prawdopodobieństwo uzyskania dobrego finalnego wyniku. Konstrukcja reguły wyboru i estymacji tego prawdopodobieństwa inspirowana jest sposobem funkcjonowania rzeczywistych kolonii mrówek, zwłaszcza metodą przekazywania informacji o źródle pokarmu. Mrówka, która znalazła pożywienie oznacza swoją trasę feromonem, który przyciąga inne mrówki. Podobne zasady obowiązują w przypadku symulowanych mrówek. Krawędzie, które należały do krótszych znalezionych tras mają większe prawdopodobieństwo wyboru przy kolejnym cyklu przeszukiwania grafu.

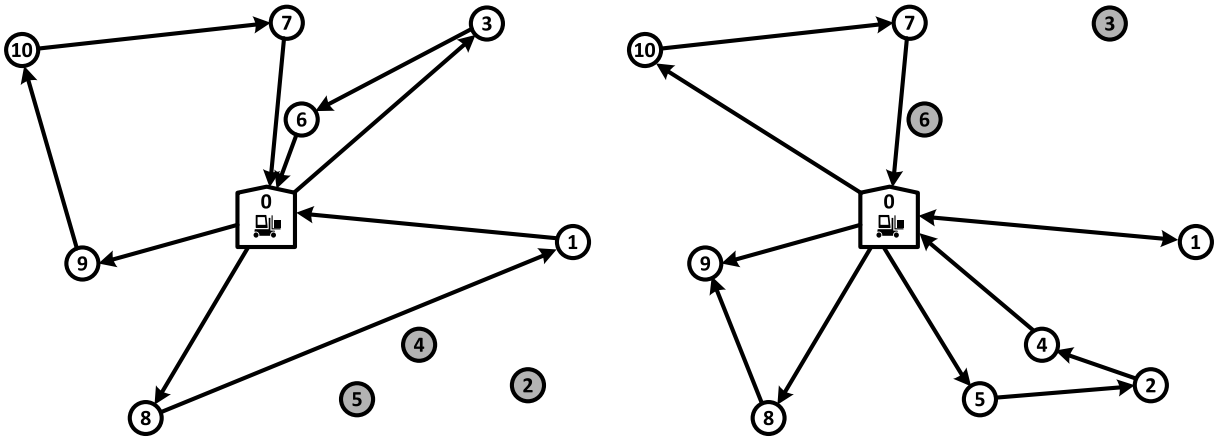
Pierwotnie algorytm został zastosowany do rozwiązania TSP [39]. Wprowadzenie dodatkowej reguły związanej z koniecznością powrotu mrówki do wierzchołka wyjściowego (tj. bazy) dostosowało go do modelu VRP. Powrót do wierzchołka wyjściowego (i tym samym rozpoczęcie nowej ścieżki) wymuszany jest, gdy odwiedzenie kolejnego wierzchołka (reprezentującego zamówienie) spowodowałoby przekroczenie przez sumę wag wierzchołków na aktualnej ścieżce zadanego parametru (tj. pojemności pojazdu).

Każdy powrót do bazy (zarówno dokonany z wyboru mrówki jak i z konieczności) rozpoczyna trasę nowego pojazdu. Takie kodowanie problemu, zdefiniowane jako CIL, w praktyce sprowadza się do bezpośredniego operowania na zbiorach permutacji R.

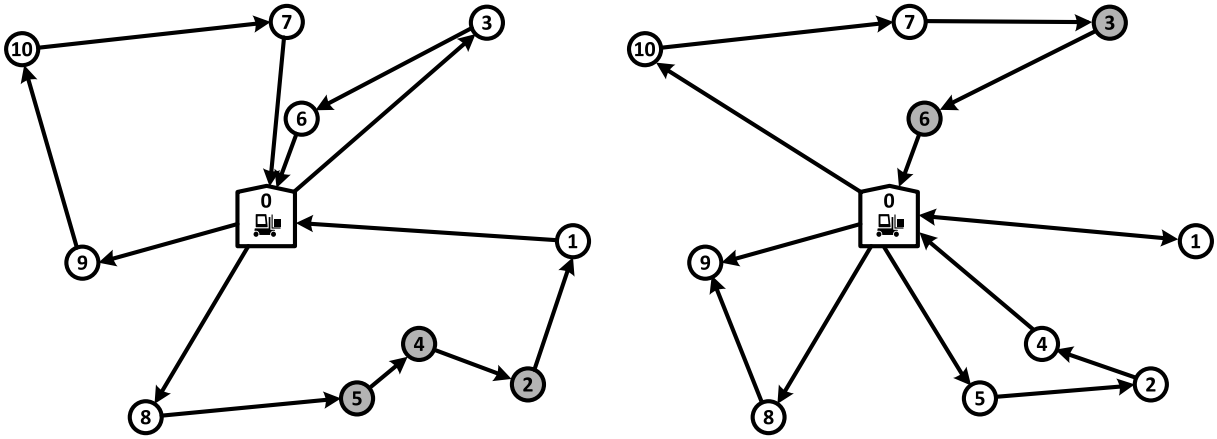
Podejścia oparte o algorytm mrówkowy zostały dwukrotnie zaprezentowane w literaturze przez Montemanniego i in. [75] oraz Elhassanię i in. [33]. Główna różnica pomiędzy tymi podejściami opierała się na zastosowaniu różnych pomocniczych heurystyk poprawiających trasy. W pracy [75] było to zachłanne wstawianie, zaś w pracy [33] LNS.



(a) Wybór pojazdów, których zamówienia będą poddane zmianom



(b) Usunięcie przypisania zamówień 2,4,5 z pierwszego osobnika oraz 3,6 z drugiego osobnika



(c) Przypisanie zamówień 2,4,5 oraz 3,6 w sposób zachłanny

Rysunek 6.1: Operacja krzyżowania w GA dla DVRP z pracy [47]. Szarym kolorem oznaczone są zamówienia, których przypisanie jest zmieniane przez operację krzyżowania.

6.1.2 Algorytm genetyczny

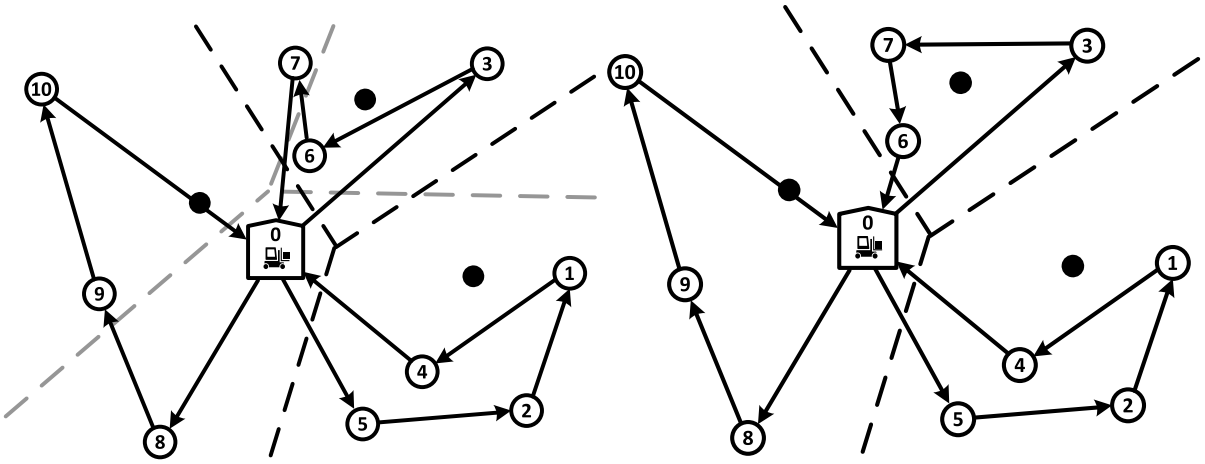
Innym możliwym podejściem do bezpośredniego przeszukiwania przestrzeni możliwych zbiorów permutacji R jest zastosowanie algorytmu genetycznego (ang. *Genetic Algorithm* (GA)). Algorytm genetyczny inspirowany jest teorią ewolucji, traktując rozwiązanie jak pewien zestaw genów, który jest testowany w środowisku problemu (optymalizowana funkcja jakości określana jest tutaj jako funkcja przystosowania). Dla generowania nowych rozwiązań wykorzystywane są operatory krzyżowania i mutacji. Operator krzyżowania ma za zadanie łączyć fragmenty rozwiązań z kilku różnych propozycji, podczas gdy operator mutacji dokonuje mniejszych zmian w pojedynczym rozwiązaniu.

Algorytm genetyczny dla DVRP, zaproponowany w pracy Hanshara i Ombuki–Berman [47], wykorzystuje dyskretne kodowanie CIZ. Operator krzyżowania w tym algorytmie generuje dwójkę osobników potomnych X'_1 i X'_2, na podstawie dwójki osobników rodzicielskich X_1 i X_2. Każdy z osobników potomnych X'_i jest tworzony z osobnika rodzicielskiego X_i poprzez ponowne zachłanne wstawianie tych podzbioru zamówień, które w drugim osobniku rodzicielskim X_{3-i} są przypisane do jednego (losowo wybieranego) pojazdu. Rysunek 6.1 przedstawia przykład realizacji tego operatora krzyżowania. Operator mutacji polega na inwersji fragmentu genotypu pomiędzy dwoma losowo wybranymi punktami.

Natomiast w pracy Elhassanii i in. [34] wykorzystywane jest dyskretne kodowanie CIL oraz odmienne operatory krzyżowania i mutacji. Krzyżowanie polega na wybraniu fragmentu trasy w każdym z osobników rodzicielskich, a następnie ponownym wstawieniu zamówień z obu tych fragmentów w obu osobnikach, tworząc również dwa osobniki potomne. Natomiast operator mutacji zamienia miejscami dwa zamówienia przypisane do tego samego pojazdu.

6.1.3 Optymalizacja rojem cząstek

Algorytm optymalizacji rojem cząstek (ang. *Particle Swarm Optimization* (PSO)), podobnie do algorytmu mrówkowego, opiera się na koncepcji inteligencji rojowej. Algorytm PSO został zaproponowany w 1995 roku przez Eberharta i Kennedy'ego [53]. W PSO proste byty, których zbiorem steruje algorytm, nazywane są cząstkami. Cząstki tworzą rój, w którym wymieniają informacje ze swoimi sąsiadami o wartościach rozwiązania w różnych punktach przestrzeni przeszukiwań. Sąsiedztwo danej cząstki należy rozumieć jako arbitralnie ustalany zbiór cząstek, którym przekazuje ona informacje o spróbkowanych wartościach rozwiązań. Każda cząstka składa się z czterech wektorów wielkości przestrzeni przeszukiwania, zawierających następujące informacje: aktualną lokalizację cząstki x_i, aktualną prędkość cząstki v_i, lokalizację najlepszego rozwiązania sprawdzonego przez cząstkę $x_i^{(BEST)}$ i lokalizację najlepszego rozwiązania sprawdzonego przez sąsiadów cząstki $x_N^{(BEST)}$.



(a) Zmiana centroidów klastrów zamówień. Kolejność zamówień na trasach jest zdeterminowana przez kolejność obowiązującą w poprzednim przypisaniu zamówień. Linie szarego koloru prezentują sposób pierwotnego podział zamówień pomiędzy pojazdy

(b) Optymalizacja trasy jednego z pojazdów wykonywana algorytmem 2–OPT przed oceną aktualnego podziału zamówień.

Rysunek 6.2: Operacja zmiany centroidów klastrów oraz późniejszej optymalizacji tras dla ciągłego kodowania WCZ.

Poza sposobem konstrukcji sąsiedztwa cząstki, dynamiką roju sterują trzy parametry: bezwładność cząstki ω, współczynnik c_1 maksymalnej wartości przyciągania do lokalizacji $x_i^{(BEST)}$ i współczynnik c_2 maksymalnej wartości przyciągania do lokalizacji $x_N^{(BEST)}$. Reguły zmiany położenia i prędkości i-tej cząstki w iteracji t opisują poniższe formuły (u_1,u_2 to zmienne z wielowymiarowego rozkładu jednostajnego na przedziale $[0; 1]$):

$$x_{i,t} = x_{i,t-1} + v_{i,t-1} \tag{6.1}$$

$$v_{i,t} = \omega v_{i,t-1} + u_1 c_1 (x_i^{(BEST)} - x_{i,t}) + u_2 c_2 (x_N^{(BEST)} - x_{i,t}) \tag{6.2}$$

Algorytm PSO jest wykorzystywany do rozwiązywania DVRP w różny sposób, w zależności od przyjętego sposobu kodowania. Pierwszy został zaproponowany przez Khouadjię i in. [54–58] i polega na dyskretyzacji lokalizacji cząstek oraz ograniczeniu operacji dodawania prędkości do przestrzeni $\mathbb{Z}_{\hat{n}}$. W ten sposób każda lokalizacja cząstki stanowi pewne przypisanie zamówień do pojazdów, zgodnie z dyskretnym kodowaniem WPP. Drugi sposób, wykorzystywany w [45], wiąże się z odmienną metodą dyskretyzacji operatorów PSO, związaną z reprezentacją rozwiązania w postaci ciągu zamówień. Sposób zaproponowany w tej rozprawie, pod nazwą ContDVRP [83, 84] (wykorzystywany również w [69]), polega na rozwiązaniu DVRP w dwóch sekwencyjnych fazach optymalizacji. W pierwszej fazie wykorzystane jest ciągłe kodowanie WCZ do optymalizacji przypisania zamówień, jako problemu grupowania z ograniczeniami (CCP). W drugiej fazie, korzystającej z ciągłego

kodowania WRZ, rozwiązywane są zadania TSP, dla każdego z pojazdów osobno. Rysunek 6.2 pokazuje przykład operacji zmiany przypisania zamówień (względem rozwiązania bazowego z Rysunku 5.1) wraz ze wstępną optymalizacją tras. Wstępne rozwiązanie TSP w ramach pierwszej fazy procesu optymalizacji ma na celu estymację jakości przypisania pod kątem rozwiązywania kompletnego problemu VRP.

6.1.4 Ewolucja różnicowa

Algorytm ewolucji różnicowej (ang. *Differential Evolution* (DE)) został zaproponowany przez Storna i Price'a [117]. Główną ideą tego algorytmu, różniącą go od standardowego algorytmu ewolucyjnego, jest operator mutacji, który jest sterowany rozkładem osobników w przestrzeni przeszukiwania.

W podejściu ContDVRP, jako alternatywny wobec PSO algorytm optymalizacyjny, wykorzystywana jest standardowa metoda DE (oznaczana w literaturze jako konfiguracja `DE/rand/1/bin`). W tej konfiguracji DE, w każdej iteracji t, każdy z osobników x_t^i jest krzyżowany z osobnikiem $y_t^{(3)}$, powstałym w wyniku mutacji różnicowej osobnika $x_t^{(3)}$. Mutacja różnicowa polega na dodaniu do $x_t^{(3)}$ przeskalowanego przez $F \in \mathbb{R}$ wektora różnicy między dwoma innymi, również losowo wybranymi, osobnikami $x^{(1)}$ $x^{(2)}$:

$$y_t^{(3)} = x_t^{(3)} + F \times (x_t^{(2)} - x_t^{(1)}) \tag{6.3}$$

Każdy z losowych osobników $x_t^{(1)}$, $x_t^{(2)}$, $x_t^{(3)}$ jest inny i różny od rozpatrywanego x_t^i.

Następnie zmutowany osobnik $y_t^{(3)}$ zostaje skrzyżowany x_t^i przez rekombinację binarną z prawdopodobieństwem p:

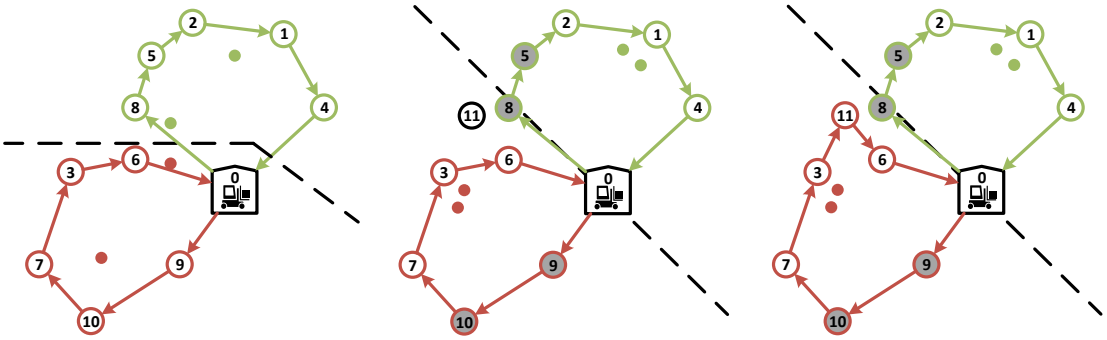
$$y_t^i = Bin_p(x_t^i, y_t^{(3)}) \tag{6.4}$$

W ostatnim kroku nowa lokalizacja y_t^i zastępuje pierwotną x_t^i wtedy i tylko wtedy, gdy zapewnia lepsze rozwiązania funkcji dopasowania f:

$$x_{t+1}^i = \begin{cases} y_t^i & f(y_t^i) < f(x_t^i) \\ x_t^i & \sim \end{cases} \tag{6.5}$$

6.2 Transfer wiedzy

Jedną z popularnych technik wspomagających algorytmy optymalizujące problemy dynamiczne jest transfer wiedzy. Szczególna uwaga poświęcona jest wiedzy o danej instancji problemu, zdobytej w trakcie optymalizacji w postaci rozwiązania dla poprzedniego kroku czasowego. Ponadto, w przypadku algorytmów pracujących w sposób równoległy, niezbędna jest metoda synchronizacji informacji pomiędzy instancjami serwisów optymalizacyjnych.



(a) Rozwiązane znalezione w przedziale czasowym i

(b) Zatwierdzenie zamówień "5", "8", "9", "10", obliczenie centroidów dla pozostałych zamówień oczekujących oraz przyjęcie nowego zamówienia "11".

(c) Wygenerowanie tras użytych do inicjalizacji w przedziale czasowym $i+1$

Rysunek 6.3: Przykład znalezienia dobrej jakości rozwiązania w wyniku zastosowania transferu informacji pomiędzy krokami czasowymi metodą odzyskania aproksymacji, dzięki podążaniu centroidów skupień za średnimi lokalizacjami zamówień oczekujących. W przypadku bezpośredniego transferu rozwiązania z poprzedniego kroku czasowego zamówienie "11" zostałoby początkowo przypisane do zielonej trasy, co dawałoby w efekcie rozwiązanie niższej jakości, ze względu na jednoczesne zatwierdzenie zamówień "8" i "5".

W tym podrozdziale zostaną omówione sposoby przekazywania rozwiązań w algorytmach rozwiązujących DVRP, ze szczególnym uwzględnieniem mechanizmów stosowanych w metodzie ContDVRP.

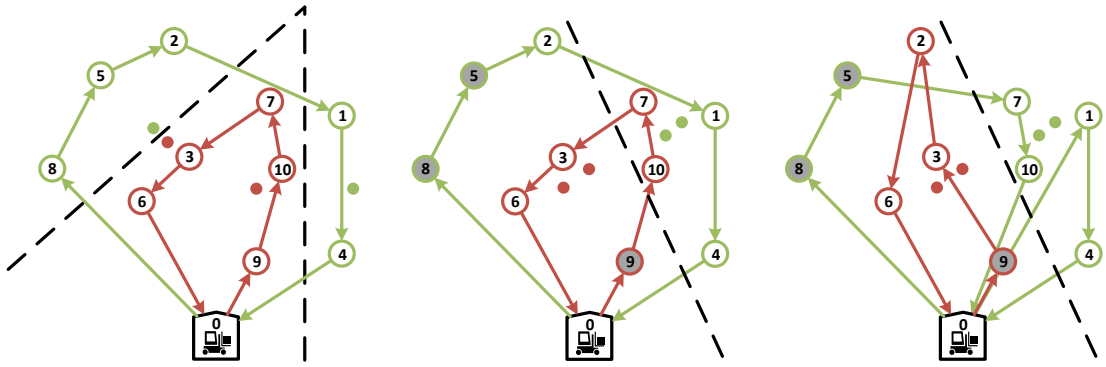
6.2.1 Transfer pomiędzy krokami czasowymi

Najprostszą, z punktu widzenia algorytmu optymalizacyjnego, strategią jest dostosowanie każdego z utrzymywanych rozwiązań (mrówek, osobników, cząstek) do aktualnego stanu problemu (nowe oczekujące zamówienia, konieczność wprowadzenia nowych pojazdów, zatwierdzone zamówienia).

Tego rodzaju podejście jest wykorzystywane w GA [47] i MEMSO [58] (gdzie nowe oczekujące zamówienia są wstawiane w sposób zachłanny do istniejącego zestawu rozwiązań). ACS [75] stosuje odmienne podejście przenosząc, adaptując i uzupełniając macierz rozkładu feromonów uzyskaną dla poprzedniego stanu grafu zamówień[1].

W rozprawie przyjęta jest strategia, w której do kolejnego kroku czasowego przenoszone (i adaptowane) są wyłącznie najlepsze rozwiązania. Zaletą takiej strategii jest jej większa niezależność od konkretnego algorytmu, co daje potencjał do wykorzystania różnych algorytmów w kolejnych krokach czasowych. Strategia przenoszenia najlepszych wy-

[1]Wartości feromonu na krawędziach istniejących we wcześniejszym stanie problemu są "spychane" o 30% w kierunku wyjściowej wartości feromonu.



(a) Rozwiązane znalezione w przedziale czasowym i

(b) Zatwierdzenie zamówień "5", "8" i "9" oraz obliczenie tych do inicjalizacji w przecentroidów dla pozostałych dziale czasowym $i+1$ zamówień oczekujących.

(c) Wygenerowanie tras użyte

Rysunek 6.4: Przykład możliwego obniżenia jakości rozwiązania w wyniku zastosowania transferu poprzez odzyskanie aproksymacji, przy założeniu dużego rozmiaru zamówień "1" i "10". W sytuacji, w której rozwiązanie wskazane w poprzednim kroku czasowym składa się z tras o nie separowalnych liniowo zamówieniach, w wyniku transferu rozwiązań poprzez odzyskiwanie aproksymacji, jakość rozwiązania może ulec znacznemu pogorszeniu w stosunku do transferu bezpośredniego.

ników ma dwie odmiany: pierwotnie stosowane odzyskiwanie aproksymacji oraz, dodane w finalnej wersji, przenoszenie bezpośrednie.

Odzyskiwanie aproksymacji polega na stworzeniu ciągłej reprezentacji o jednym centroidzie grup zamówień na pojazd, poprzez uśrednienie lokalizacji wszystkich oczekujących zamówień przypisanych do danego pojazdu. W populacji początkowej aktualnego kroku czasowego umieszczane jest rozwiązanie, którego każdy z k centroidów dla pojazdu inicjalizowane jest tą samą uśrednioną lokalizacją, zaburzoną małą zmienną losową. Jeżeli liczba pojazdów wykorzystywanych w poprzednim kroku czasowym jest niewystarczająca, rozwiązanie jest poszerzane o losowo rozmieszczone centroidy skupisk odpowiadające dodawanym pojazdom.

Przenoszenie bezpośrednie polega na umieszczeniu w populacji początkowej w aktualnym kroku czasowym wektora centroidów skupisk zamówień wskazanego jako optymalne rozwiązanie w poprzednim kroku czasowym. Zapotrzebowanie na zwiększoną liczbę pojazdów jest uwzględniane w sposób analogiczny jak w odzyskiwaniu aproksymacji.

Strategie te dotyczą wyłącznie transferu wektora kodującego przypisanie zamówień do pojazdów. Wektor rang zamówień, decydujący o ich kolejności w harmonogramach pojazdów, jest zawsze przenoszony w sposób bezpośredni.

6.2.2 Transfer pomiędzy serwisami optymalizacyjnymi

Dla algorytmów wykorzystujących obliczenia równoległe konieczne jest stworzenie sposobu wymiany danych pomiędzy instancjami serwisów optymalizacyjnych. Spośród omawianych w rozprawie algorytmów przetwarzanie równoległe wykorzystują ContDVRP i MEMSO.

W przypadku MEMSO, zakładającym możliwość obliczeń w rozproszonym heterogenicznym środowisku, wymiana informacji jest zrealizowana na poziomie samego algorytmu optymalizacyjnego, w którym cząsteczki PSO mają pewne prawdopodobieństwo migracji pomiędzy serwisami optymalizacyjnymi, tym samym przenosząc w ten sposób informację o rozwiązaniach z innych rojów. Natomiast ContDVRP, zakładający wykonywanie obliczeń w równoległym homogenicznym środowisku, korzysta z faktu, że pomiędzy uruchomieniami serwisu należy przekazać kierowcom informacje o nowo zatwierdzonych zamówieniach. W związku z tym, serwisy są synchronizowane poprzez ich jednakową reinicjalizację, z wykorzystaniem najlepszych znalezionych rozwiązań, w każdym kolejnym punkcie decyzyjnym TS_i.

W sytuacji kiedy najlepsze rozwiązania są przenoszone poprzez odzyskiwanie aproksymacji, możliwe staje się zasilenie populacji dla ContDVRP rozwiązaniami wskazanymi przez dowolny algorytm[2]. Rysunek 6.3 prezentuje dodatkowy pozytywny aspekt odzyskiwania aproksymacji w transferze informacji, polegający na podążaniu centroidów za średnią lokalizacją zamówień oczekujących, a więc tym otoczeniem trasy, w którym możliwe jest dodawanie nowych zamówień niewielkim kosztem. Natomiast bezpośredni transfer najlepszych rozwiązań pozwala na zachowanie pełnej informacji o rozwiązaniu wskazanym przez ContDVRP w poprzednim kroku czasowego. Negatywne konsekwencje utraty informacji w przypadku stosowania odzyskiwania informacji zaprezentowane są na Rysunku 6.4. W celu wykorzystania pozytywnych aspektów obu sposobów transferu, w populacji początkowej w następnym kroku czasowym są umieszczane oba rodzaje rozwiązań.

[2]Podejście to jest wykorzystywane do zasilania populacji rozwiązaniami opartymi o zmodyfikowany algorytm Kruskala

Rozdział 7

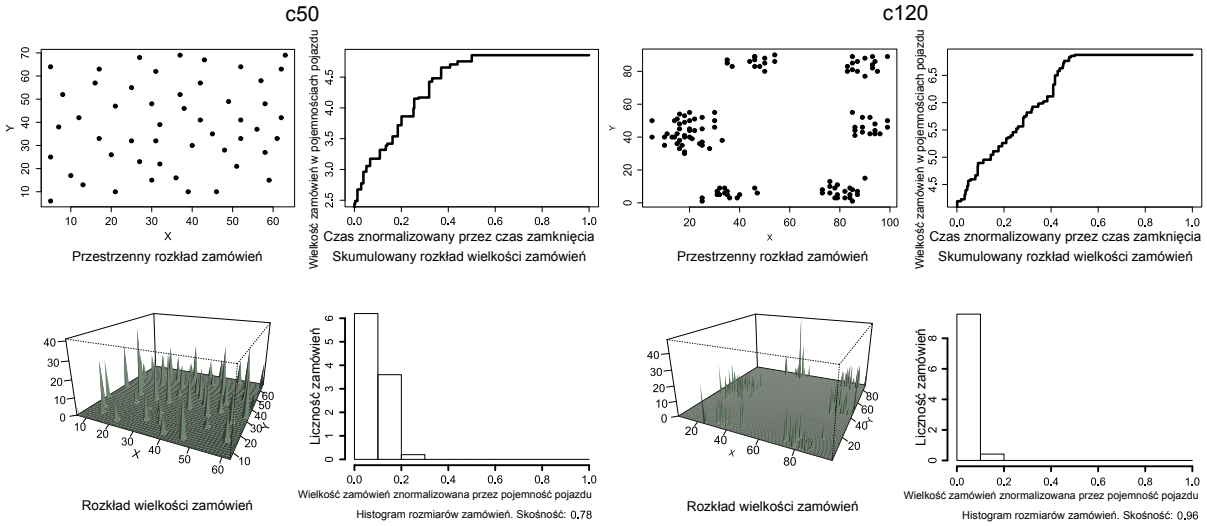
Wykorzystanie danych problemu

W tym rozdziale, po prezentacji proponowanej charakterystyki rozkładów zamówień w problemach VRP, zaprezentowane są dwie metody wykorzystujące dane problemu w celu poprawy jakości działania algorytmów optymalizacyjnych. Pierwsza z nich (metoda predykcyjna) uwzględnia dynamiczny charakter danych, druga (metoda hipeheurystyczna) – charakterystykę danych zadania.

Dotychczas zaprezentowane metody jedynie w niewielkim stopniu odnosiły się do dynamicznej natury problemu, wskazując wyłącznie na istotność transferu informacji z poprzednich kroków czasowych oraz dobór parametru T_{AC}. Zaprezentowane zostały też różnego rodzaju kodowania, które mają wpływ na otrzymywane wyniki, jednak brak było wskazań kiedy należałoby wykorzystać daną przestrzeń przeszukiwania.

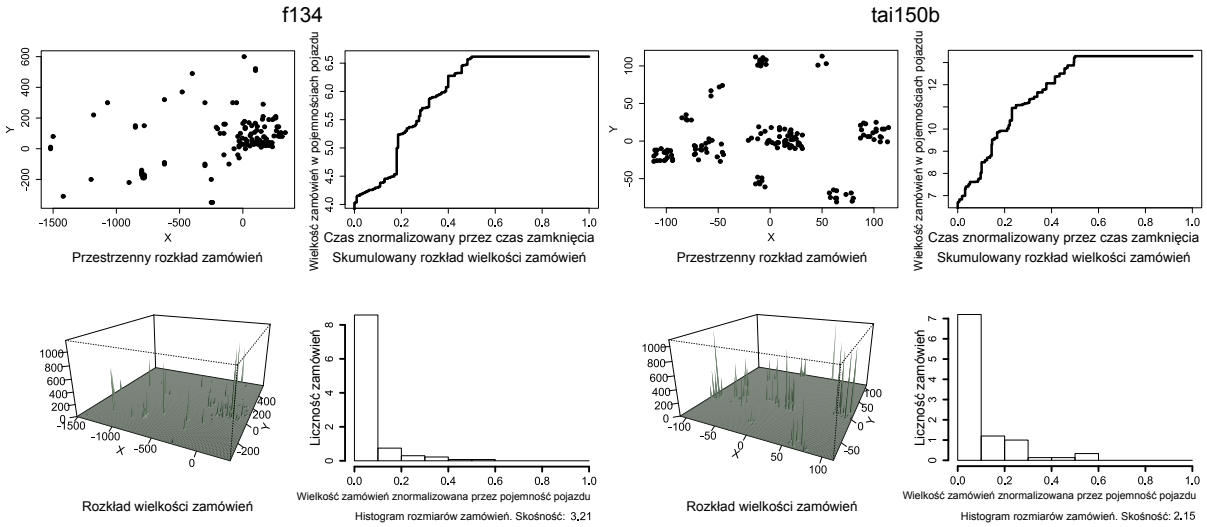
W zakresie wykorzystania danych, rozwiązania prezentowane w literaturze koncentrują się na zamodelowaniu wiedzy dyspozytorów [48,50] oraz na wysokopoziomowej identyfikacji typu problemu poprzez interakcję człowiek-maszyna [49,118]. Modelowanie wiedzy dyspozytorów miało za zadanie stworzenie zestawu reguł postępowania przy zaobserwowaniu określonego rodzaju sytuacji nietypowych. Przykładem takiej sytuacji jest przekroczenie spodziewanej wielkości zamówienia, a reguła postępowania zależna jest od położenia pojazdów, które mogłyby zrealizować resztę planowanej trasy niespodziewanie zapełnionego pojazdu. Natomiast identyfikacja typu problemu miała za zadanie dobrać właściwy model matematyczny w zależności od tego jakiego rodzaju zadanie VRP było do rozwiązania (np. jednoczesne odbiory i dostawy, wiele zajezdni).

Celem tej rozprawy jest zaproponowanie takiego wykorzystania dostępnych danych dla danej instancji DVRP, aby lepiej zaadresować jej szczególną charakterystykę. Należy zauważyć, że dynamizm DVRP wynika z dwóch czynników: bieżącego podejmowania decyzji na podstawie rozwiązań wskazywanych jako optymalne oraz pojawiania się nowych zamówień w trakcie procesu optymalizacji. Zatem metoda, która pozwala skutecznie dobrać algorytm (kodowanie) do problemu, będzie miała szczególne znaczenie w optymalizacji dynamicznej, gdzie dla danej rzeczywistej sytuacji powtórzenie obliczeń jest niemożliwe. Algorytm (kodowanie) dobierany jest w tej metodzie na podstawie niepełnych danych,



(a) Charakterystyka zamówień dla zadania **c50**. To zadanie charakteryzuje się jednostajnym rozkładem przestrzennym zamówień oraz stosunkowo zwartym rozkładem wielkości zamówień.

(b) Charakterystyka zamówień dla zadania **c120**. To zadanie charakteryzuje się rozkładem przestrzennym z wyraźnymi skupiskami zamówień oraz bardzo zwartym rozkładem wielkości zamówień.



(c) Charakterystyka zamówień dla zadania **f134**. To zadanie charakteryzuje się mieszanym rozkładem przestrzennym zamówień oraz rozkładem wielkości zamówień z licznymi małymi zamówieniami oraz kilkoma większymi pojawiającymi się w ciągu pierwszych 20% dnia roboczego.

(d) Charakterystyka zamówień dla zadania **tai150b**. Zadanie to charakteryzuje się rozkładem przestrzennym z wyraźnymi skupieniami oraz stosunkowo silnie odstającymi dużymi zamówieniami. Dodatkowo, niektóre z tych dużych zamówień stają się znane stosunkowo późno (tj. po 25% dnia roboczego).

Rysunek 7.1: Rozkład wielkości zamówień (prawe dolne histogramy) i rozkład przestrzenny zamówień (lewe wykresy) oraz zmienność wielkości sumy objętości zamówień problemu w czasie (prawe górne wykresy) dla przykładowych zadań ze zbioru testowego.

zgodnie z estymacją jego skuteczności dla problemu o danej charakterystyce. Z kolei metoda, która skutecznie antycypuje finalny kształt problemu, powinna zmniejszyć zmienność rozwiązania pomiędzy kolejnymi stanami zadania i w konsekwencji poprawić finalny

wynik. Zmniejszenie zmienności rozwiązania jest korzystne z punktu widzenia praktyki operacyjnej. Omawiane podejście jest wzorowane na metodach odpornych i optymalizacji najgorszego przypadku.

7.1 Charakterystyka zbiorów danych

Przed omówieniem metod korzystających z danych problemu, w tym podrozdziale zaprezentowane są przykładowe instancje testowe oraz przedstawione są ilościowe charakterystyki, za pomocą których można opisywać cechy instancji VRP. Ze względu na popularność zbioru testowego wprowadzonego w pracy Kilby'ego i in. [59] charakterystyka danych problemu jest omawiana na tym zestawie danych testowych. Pierwotne (statyczne) wersje problemu pochodzą z prac Christofidesa [22], Taillarda [120] i Fishera [36] nad statyczną wersją VRP. W literaturze przedmiotu dokonywano jedynie jakościowego opisu przestrzennego charakteru rozkładu zamówień, wskazując na ich jednorodny, skupiony lub mieszany układ. Informacja ta nie była jednak wykorzystywana w algorytmach, nie dokonywano też opisu tych cech w sposób ilościowy. Oprócz układu przestrzennego zamówień, istotna jest również charakterystyka rozkładu ich wielkości: skośna lub centralna, skupiona lub o długich ogonach, posiadająca duże lub drobne zamówienia względem pojemności pojazdów. Rysunek 7.1 prezentuje różnorodność rozkładów lokalizacji i wielkości zamówień na przykładzie czterech wybranych instancji DVRP ze zbioru testowego. Wszystkie instancje ze zbioru cechuje stosunkowo równomierny przyrost liczby (oraz sumarycznej wielkości) zamówień w trakcie dnia roboczego. Różnią się one natomiast między sobą rozkładem przestrzennym, rozrzutem wielkości zamówień oraz występowaniem zamówień o dużym rozmiarze względem pojemności pojazdu. Jeżeli chodzi o rozkład przestrzenny zamówień w zaprezentowanych przykładach, to `c50` cechuje ich równomierny rozkład, `c120` i `tai150b` mają wyraźne zaznaczone skupiska, natomiast `f134` ma w tym względzie charakter mieszany. W zakresie rozkładu wielkości zamówień `c50` i `c120` cechują się zamówieniami o stosunkowo niewielkich i zbliżonych do siebie rozmiarach. Natomiast `f134`, a zwłaszcza `tai150b`, charakteryzują się występowaniem pewnej liczby, nierównomiernie rozłożonych w przestrzeni dużych zamówień.

W celu ilościowego scharakteryzowania wymienionych cech jakościowych, zostały wyliczone następujące statystyki:

- średnie wartości μ_x, μ_y lokalizacji zamówień względem osi układu odniesienia,
- odchylenia standardowe sd_x, sd_y rozkładu lokalizacji zamówień względem osi układu odniesienia,
- skośność $skew_x$, $skew_y$ rozkładu lokalizacji zamówień względem osi układu odniesienia,
- optymalna liczba skupisk zamówień k_{gap} wskazana przez *gap statistic* [123],
- średnia wartość μ_s wielkości zamówienia,

Tabela 7.1: Wartości statystyk charakteryzujących początkowe dane (dla $t = 0$) ze zbioru testowego Kilby'ego. Pierwsza litera w nazwie zbioru wskazuje na autora zbioru statycznego a liczba mówi o liczności wszystkich zamówień w danej instancji problemu.

Name	μ_x	sd_x	$skew_x$	μ_y	sd_y	$skew_y$	μ_s	sd_s	$skew_s$	nc
c50	0.53	0.32	-0.18	0.51	0.33	0.08	0.09	0.04	0.58	2.00
c75	0.51	0.30	-0.08	0.41	0.29	0.27	0.12	0.06	0.34	4.00
c100	0.48	0.27	0.35	0.46	0.27	0.53	0.08	0.05	0.71	1.00
c100b	0.68	0.17	0.48	0.56	0.26	0.31	0.09	0.05	1.39	0.00
c120	0.38	0.32	0.84	0.67	0.24	0.20	0.06	0.03	1.46	0.67
c150	0.49	0.27	0.06	0.45	0.24	0.28	0.08	0.04	0.78	6.00
c199	0.52	0.26	-0.04	0.47	0.24	0.19	0.09	0.04	0.52	9.00
f71	0.44	0.23	-0.02	0.72	0.18	-0.17	0.04	0.05	1.87	0.00
f134	0.61	0.29	-0.47	0.42	0.22	0.70	0.07	0.11	2.35	1.50
tai75a	0.27	0.18	1.39	0.47	0.20	1.22	0.11	0.16	2.27	3.00
tai75b	0.69	0.25	-1.14	0.42	0.22	-0.69	0.11	0.16	1.24	1.00
tai75c	0.43	0.16	-0.93	0.49	0.25	-0.02	0.11	0.15	2.30	3.00
tai75d	0.43	0.30	0.55	0.49	0.29	-0.23	0.09	0.13	1.55	0.25
tai100a	0.39	0.30	0.61	0.57	0.25	-0.27	0.11	0.18	2.04	1.00
tai100b	0.46	0.29	0.46	0.46	0.22	0.56	0.12	0.16	1.64	1.00
tai100c	0.63	0.27	-0.91	0.50	0.20	-1.10	0.10	0.14	1.34	5.00
tai100d	0.44	0.23	-0.39	0.46	0.21	0.06	0.10	0.17	2.37	2.00
tai150a	0.47	0.29	0.17	0.55	0.31	-0.36	0.09	0.14	2.32	1.67
tai150b	0.61	0.23	0.13	0.52	0.24	0.81	0.09	0.14	2.30	6.00
tai150c	0.47	0.23	0.46	0.61	0.14	-0.06	0.09	0.13	1.80	6.00
tai150d	0.47	0.34	-0.05	0.67	0.23	-0.88	0.12	0.16	1.41	0.14

- odchylenie standardowe sd_s rozkładu wielkości zamówienia,
- skośność $skew_s$ rozkładu wielkości zamówień,
- dolne oszacowanie liczby pojazdów m_v niezbędnych do zrealizowania zamówień ($m_v = \lceil \sum_{i=1}^{m} \frac{s_i}{cap} \rceil$).

W celu normalizacji wymienionych cech, w obrębie zbioru różnych instancji problemu, lokalizacje zamówień były skalowane do obszaru $[0; 1]^2$, wielkości zamówień były dzielone przez pojemność pojazdu, a m_v i k_{gap} zostały ze sobą połączone w parametr nc:

$$nc = |1 - \frac{m_v}{k_{gap}}|$$

Wartości nc bliskie zeru, w połączeniu niewielką wartością odchylenia standardowego sd_s, wskazują na możliwość umieszczenia każdej z grup zamówień wynikających z rozkładu przestrzennego w jednym pojeździe bez przekraczania jego pojemności.

Tabela 7.1 zawiera numeryczne wartości dla proponowanych statystyk zbioru zamówień. Statystyki tego rodzaju mogą posłużyć również do opisu innych zadań z rodziny

VRP i być przydatne we wskazywaniu zależności między nimi a rezultatami osiąganymi przez dany algorytm.

7.2 Metody predykcyjne

W tym podrozdziale zaproponowane są metody predykcyjne, uwzględniające szczególny charakter dynamizmu problemu DVRP, czyli pojawianie się nowych zamówień. Finalny wariant metody predykcyjnej wykorzystuje założenie, że można przewidzieć sumę rozmiarów wszystkich zamówień, na podstawie rozmiarów zamówień dostępnych w danym momencie. W celu przedstawienia kontekstu opisany jest najpierw pierwotny wariant tej metody. Pierwotny wariant został odrzucony, ze względu na konieczność przyjęcia założeń dotyczących przestrzennego rozkładu zamówień oraz mniejszą stabilność uzyskiwanych wyników.

W obszarze problemów marszrutyzacji metody predykcyjne były wykorzystywane dla wariantu stochastycznego (ang. *Stochastic VRP* (SVRP)) oraz cyklicznego (ang. Periodic VRP (PVRP)). Dla SVRP zaproponowana została proaktywna metoda projektowania tras oparta na TS i VNS [35]. W proaktywnym projektowaniu tras wykorzystywane są zamówienia historyczne z kilku poprzednich dni. Alternatywna metoda dla tego wariantu problemu marszrutyzacji zaproponowana została w [119]. W tej alternatywnej metodzie ograniczenie na sumę wielkości zamówień w pojeździe jest uzupełnione o parametr (bufor pojemności) uwzględniający poziom niepewności w określeniu wielkości zamówień. Dla PVRP zostało zaproponowane podejście wykorzystujące rozkłady prawdopodobieństwa zamówień [4]. Do grupy metod, które uwzględniają niepewność danych, można zaliczyć również algorytm *Upper Confidence Bound applied to Trees* (UCT), zastosowany do statycznego VRP[1] [69].

7.2.1 Pierwotna wersja metody predykcyjnej: generowanie sztucznych zamówień

Na wstępie należy zwrócić uwagę na fakt, że możliwość wykorzystania danych w DVRP do predykcji przyszłych zamówień jest ograniczona. Wynika to z faktu, że problem ma dynamiczny (a nie stochastyczny) oraz chaotyczny (a nie cykliczny lub przewidywalny) charakter, w rozumieniu z definicji podanych w Rozdziale 1. Zatem, aby predykcja nieznanych zamówień była możliwa, konieczne jest przyjęcie zestawu założeń. Pierwotna wersja metody predykcyjnej przyjmuje następujące założenia:

- od początku dnia znany jest prostokąt ograniczający obszar występowania zamówień,

[1] Możliwość zastosowania UCT do wariantu SVRP ze zmiennym ruchem zaprezentowana jest na stronie `http://www.mini.pw.edu.pl/ mandziuk/dynamic/?page_id=142`

Tabela 7.2: Dodatkowe symbole do opisu DVRP w algorytmach wykorzystujących dane problemu

Symbol	Typ	Opis
C'_t	Ciąg	Sztuczne zamówienia generowane w chwili t
m'_t	$\mathbb{Z}_+$	Liczba sztucznych zamówień wygenerowanych w chwili t
$l_i.x$	$\mathbb{R}$	Współrzędna x i-tego zamówienia
$l_i.y$	$\mathbb{R}$	Współrzędna y i-tego zamówienia
L'_t	Ciąg	Położenia sztucznych zamówień wygenerowanych w chwili t
S'_t	Ciąg	Rozmiary sztucznych zamówień generowanych w chwili t
$\bar{u}_{m_t}$	$\mathbb{R}_+$	Średni czas obsługi pierwszych m_t zamówień

- średnia częstotliwość pojawiania się zamówień jest jednakowa w każdym okresie,
- rozkład rozmiarów zamówień znanych na początku dnia roboczego dobrze odzwierciedla rozkład rozmiarów zamówień pojawiających się przez cały dzień.

Przyjęcie powyższych założeń pozwoliło na zaproponowanie metody generowania sztucznych zamówień [85], mających za zadanie służyć jako bufor bezpieczeństwa przy planowaniu tras pojazdów. Metoda generowania zamówień składa się z następujących elementów: estymacji spodziewanej liczby zamówień, generatora lokalizacji zamówień oraz generatora wielkości zamówień. Dodatkowe symbole, wykorzystane do opisu estymatora i generatorów, zdefiniowane są w Tabeli 7.2.

Metoda generowania ma za zadanie w chwili t stworzyć ciąg sztucznych zamówień C'_t. Ciąg zamówień C'_t składa się z ciągu wielkości zamówień S'_t, ciągu lokalizacji zamówień L'_t, ciągu czasów dostępności (ustawionych na t) oraz ciągu czasów rozładunku (równych średniemu empirycznemu czasowi rozładunku $\bar{\mathbf{u}}_{\mathbf{m_t}}$).

$$C'_t := (S'_t, L'_t, (t, \ldots, t), (\bar{\mathbf{u}}_{\mathbf{m_t}}, \ldots \bar{\mathbf{u}}_{\mathbf{m_t}})) \tag{7.1}$$

Liczba sztucznych zamówień do wygenerowania m'_t jest estymowana w oparciu o średnią zaobserwowaną częstotliwość pojawiania się zamówień:

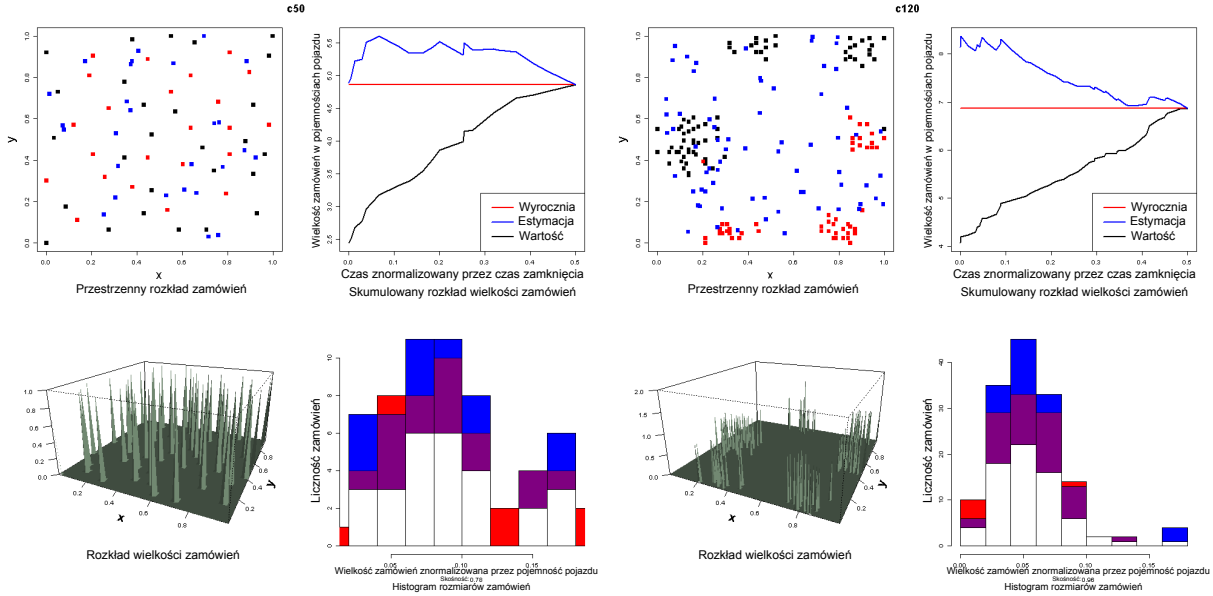
$$m'_t = m_t \frac{T_{CO}(t_{end} - t_{start}) + t_{start} - t}{T_{CO}(t_{end} - t_{start}) - t_{start} + t} \tag{7.2}$$

Rozmiary zadań są generowane poprzez próbkowanie wielkości zamówień znanych w chwili t:

$$S'_t \sim U(\{s_1, s_2, \ldots, s_{m_t}\}) \tag{7.3}$$

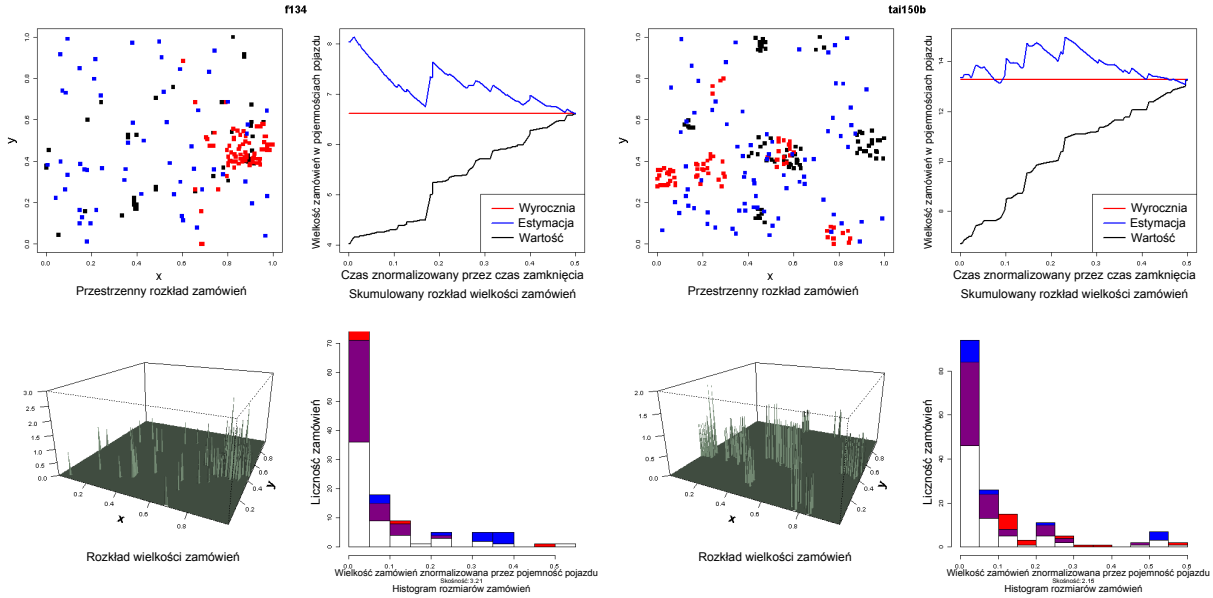
Lokalizacje zamówień są generowane poprzez próbkowane z dwuwymiarowego rozkładu jednostajnego na prostokącie ograniczającym lokalizacje zamówień:

$$L'_t \sim U([\min_{i_1 \in \{1,2,\ldots,m\}}(l_{i_1}.x), \max_{i_2 \in \{1,2,\ldots,m\}}(l_{i_2}.x)] \times [\min_{i_3 \in \{1,2,\ldots,m\}}(l_{i_3}.y), \max_{i_4 \in \{1,2,\ldots,m\}}(l_{i_4}.y)]) \tag{7.4}$$



(a) Charakterystyka estymacji zamówień dla zadania **c50**.

(b) Charakterystyka estymacji zamówień dla zadania **c120**.



(c) Charakterystyka estymacji zamówień dla zadania **f134**.

(d) Charakterystyka estymacji zamówień dla zadania **tai150b**.

Rysunek 7.2: Rozkład wielkości zamówień znanych i przyszłych (estymowanych i faktycznych) w przykładowej chwili t (prawe dolne histogramy). Rozkład przestrzenny zamówień w analogicznym układzie (lewe wykresy). Zmienność w czasie sumy rozmiarów zamówień znanych i estymowanych (prawe górne wykresy).

Jednostajny rozkład lokalizacji zamówień wynika z przyjęcia minimalnych informacji na temat ich możliwego położenia. W praktyce, zarówno czas dostępności jak i rozkład przestrzenny zamówień, mogłyby być próbkowane z historycznych rozkładów tych wielkości w danym obszarze.

Rysunek 7.2 pokazuje efekty procesu estymacji sztucznych zamówień dla przykładowych testowych instancji DVRP, prezentowanych wcześniej na Rysunku 7.1. Można zaobserwować, że estymacja rozkładu wielkości zamówień poprzez próbkowanie z wielkości znanych zamówień dobrze oddaje charakterystykę faktycznego rozkładu. Rozkład przestrzenny przyszłych zamówień, ze względu na przyjęte założenia, może istotnie odbiegać od rozkładu faktycznego. Można też zauważyć, że dla przykładowych testowych instancji, w wyniku generowania sztucznych zamówień, estymowana suma ich wielkości jedynie nieznacznie przekracza faktyczną finalną sumę.

Eksperymenty z wykorzystaniem generatora sztucznych zamówień potwierdzają możliwość przyjęcia założeń dotyczących częstotliwości oraz rozkładu wielkości zamówień dla zbioru instancji testowych. Połączenie tego wniosku z podejściami stosowanymi w optymalizacji odpornej oraz metodzie najgorszego przypadku zaowocowały zaprojektowaniem i włączeniem do ContDVRP finalnej wersji metody predykcyjnej.

7.2.2 Finalna wersja metody predykcyjnej: estymacja sumy rozmiarów zamówień

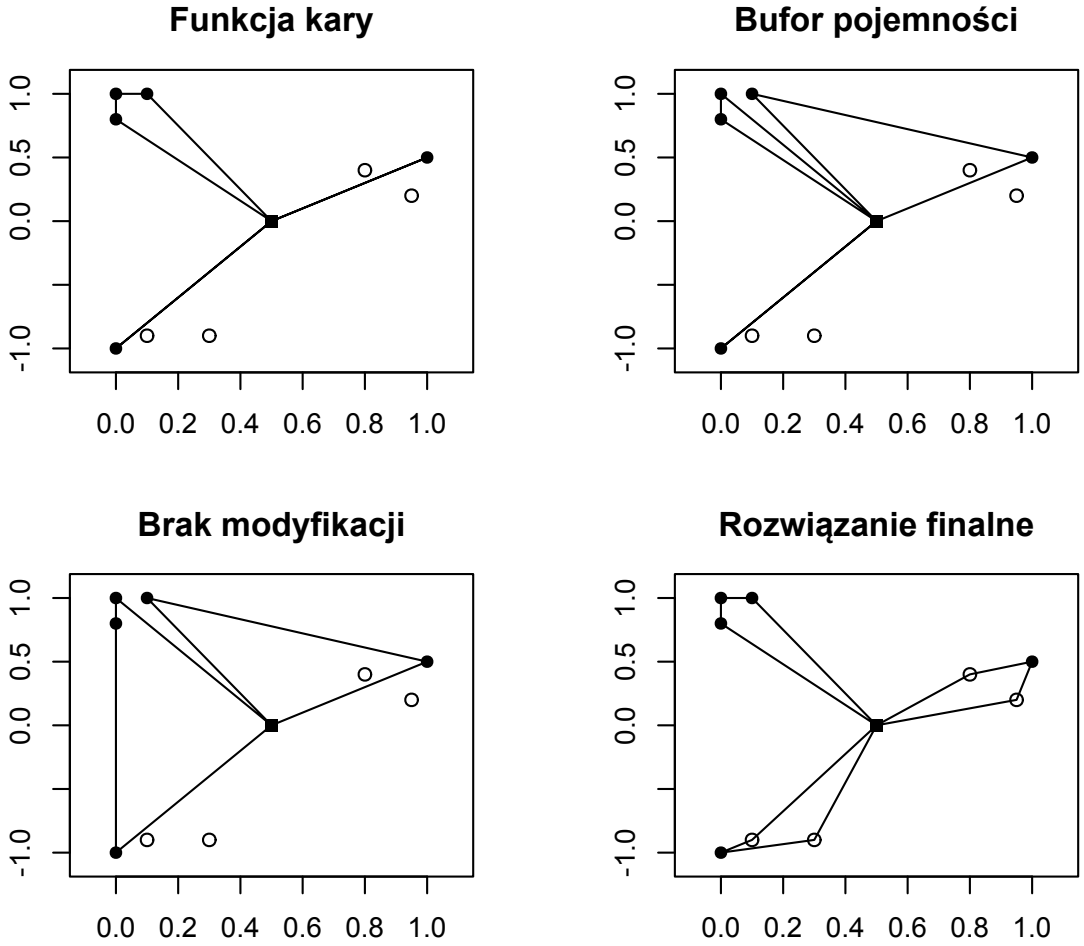
Finalna metoda predykcyjna zaproponowana w tej rozprawie, bazuje na koncepcji estymacji liczby i wielkości zamówień, ale całkowicie rezygnuje z uwzględnienia ich potencjalnych lokalizacji. Podejście to oznacza przyjęcie mniejszej liczby założeń, większą łatwość implementacji oraz bardziej stabilne predykcje pomiędzy kolejnymi krokami czasowymi. Finalna wersja metody predykcyjnej przyjmuje następujące założenia:

- średnia częstotliwość pojawiania się zamówień jest jednakowa w każdym okresie,
- suma rozmiarów zamówień znanych w trakcie dnia roboczego dobrze odzwierciedla finalną sumę rozmiarów zamówień.

W porównaniu z wersją pierwotną oznacza to całkowitą rezygnację z założenia dotyczącego prostokąta ograniczającego lokalizacje zamówień, a także osłabienie założenia o poprawności estymacji rozkładu rozmiarów zamówień do założenia o poprawności estymacji jedynie sumy rozmiarów zamówień.

Proponowana w rozprawie metoda estymuje $n_{t_{end}}$, czyli liczbę pojazdów niezbędnych do obsłużenia zamówień w finalnym rozwiązaniu. Estymacja $\hat{n}_{t_{end}}$ jest liczona w oparciu o założenia o rozmiarach zamówień i częstotliwości ich pojawiania się:

$$\hat{n}_{t_{end}}(t) = \left\lceil \frac{T_{CO}(t_{end} - t_{start}) + t_{start} - t}{T_{CO}(t_{end} - t_{start}) - t_{start} + t} * \sum_{i=1}^{m_t} \frac{s_i}{cap} \right\rceil \tag{7.5}$$



Rysunek 7.3: Porównanie kształtu pośrednich optymalnych rozwiązań w zależności od zastosowania funkcji kary, bufora pojemności dla każdego z pojazdów oraz braku modyfikacji z pożądanym rozwiązaniem finalnym. Zajezdnia jest oznaczona kwadratem, zamówienia znane w chwili t pełnymi kołami, zaś zamówienia nieznane w chwili t pustymi kołami.

Estymacja przewidywanej liczby pojazdów może być wykorzystywana w ContDVRP na dwa sposoby. Pierwszy polega na wprowadzeniu kryterium stopu w zmodyfikowanym algorytmie Kruskala. Drugi sposób polega na dodawaniu funkcji kary do funkcji jakości rozwiązania, w sytuacji gdy wykorzystywana w chwili t liczba pojazdów $\hat{n}_R(t)$ jest mniejsza od estymacji $\hat{n}_{t_{end}}(t)$.

Kryterium stopu w zmodyfikowanym algorytmie Kruskala polega na przerwaniu procesu scalania skupisk zamówień, jeżeli ich liczba jest mniejsza lub równa estymacji $\hat{n}_{t_{end}}(t)$. Natomiast funkcja kary zwiększa wartość funkcji jakości o tyle średnich długości tras z rozwiązania R, o ile wykorzystywana w R liczba pojazdów $\hat{n}_R$ jest mniejsza od estymacji ich finalnej liczby $n_{t_{end}}$:

$$Penalty(R,t) = \left(\hat{n}_{t_{end}(t)} - \hat{n}_R\right) * \frac{Cost(R)}{\hat{n}_R} \tag{7.6}$$

Zastosowanie funkcji kary przynosi efekt podobny do zastosowania bufora w pojemności pojazdu (str. 11 w [119]) czy bufora czasu dnia (str. 153 w [54]), gdyż generowane są rozwiązania pośrednie wykorzystujące liczbę pojazdów bliską liczbie pojazdów w rozwiązaniu finalnym. Zastosowanie funkcji kary ma przewagę nad stosowaniem takich buforów w przypadku nierównomiernej reprezentacji zamówień dla tras z pożądanego finalnego rozwiązania w pośrednich stanach problemu. Rysunek 7.3 prezentuje przykładowe rozwiązania cząstkowe dla tego rodzaju sytuacji, w której optymalne finalne rozwiązanie składa się z trzech tras pojazdów, po trzy zamówienia na każdej trasie. W zaprezentowanym pośrednim stanie problemu znane są trzy zamówienia z jednej z finalnych tras oraz po jednym zamówieniu z dwóch pozostałych tras. Metoda wykorzystująca funkcję kary oraz metoda wykorzystująca bufor pojemności wskazują rozwiązanie z trzema trasami jako rozwiązanie optymalne, w przeciwieństwie do metody rozwiązującej zadanie jako instancję statyczną, która wskaże jako optymalne rozwiązanie z wyłącznie dwiema trasami. Prezentowany przypadek ujawnia też przewagę stosowania funkcji kary nad zastosowaniem bufora pojemności pojazdu. Rozwiązanie pośrednie zwracane w podejściu z funkcją kary jest w tym przypadku podrozwiązaniem finalnego optymalnego rozwiązania.

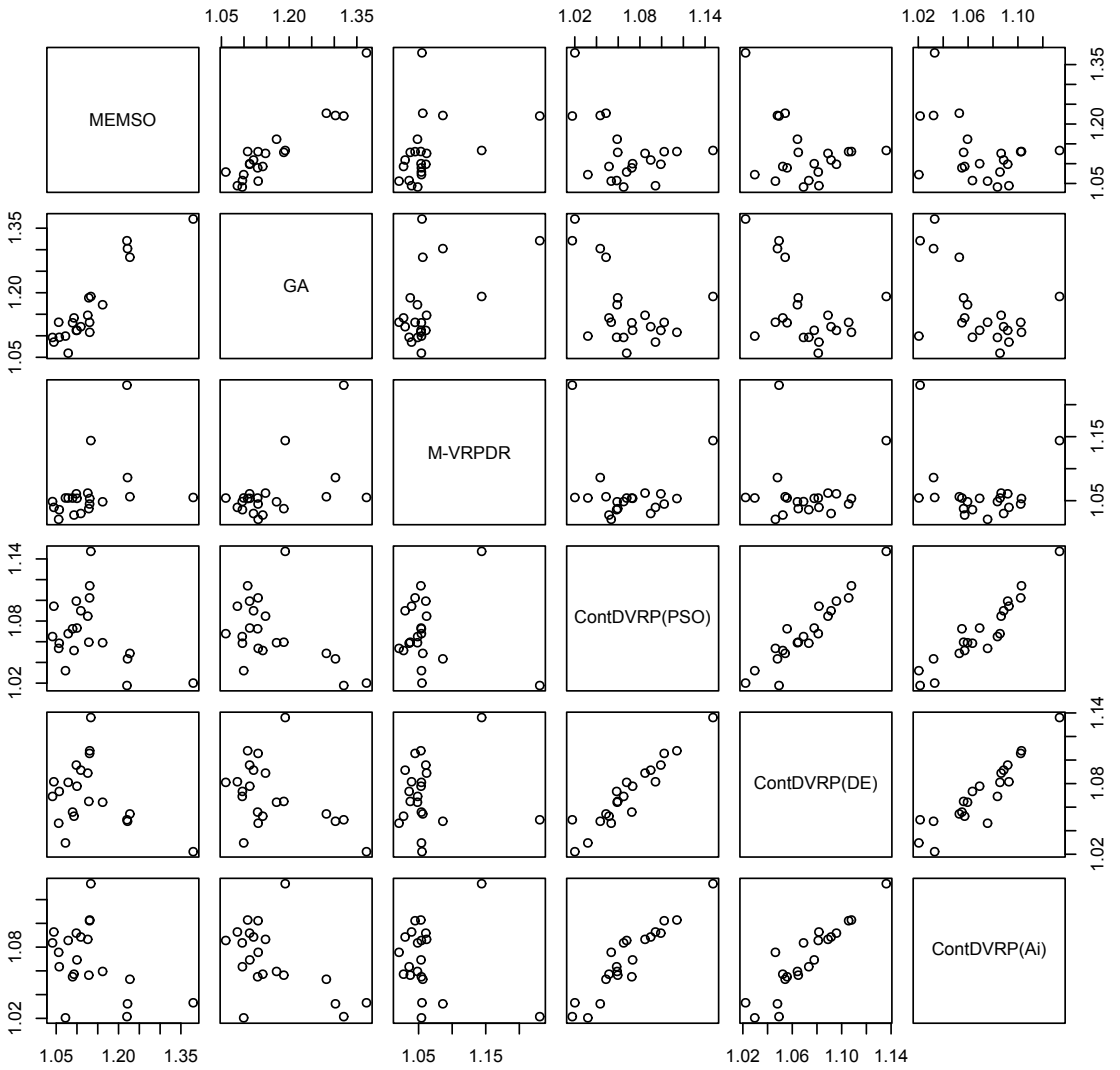
Należy jednak zauważyć, że metody korzystające z estymacji $\hat{n}_{t_{end}}(t)$ nie będą w stanie znaleźć optymalnych, z punktu widzenia finalnego wyniku, cząstkowych rozwiązań dla pośrednich stanów problemu, w których nie ma przynajmniej jednego reprezentanta zamówień dla każdej z tras z pożądanego optymalnego rozwiązania finalnego.

7.3 Algorytm hiperheurystyczny

W tym podrozdziale przedstawiony jest drugi sposób wykorzystania danych problemu, polegający na zastosowaniu podejścia hiperheurystycznego [19]. Motywacją dla omówionego w poprzednich podrozdziałach podejścia było bezpośrednie zaadresowanie charakteru dynamizmu w DVRP. Główną motywacją dla podejścia hiperheurystycznego jest wykorzystanie różnorodności już istniejących algorytmów (typów kodowania problemu i przestrzeni przeszukiwań) dla DVRP. Proponowane podejście hiperheurystyczne wykorzystuje powiązanie skuteczności danego algorytmu z charakterystyką optymalizowanej instancji problemu. Należy też zauważyć, że proponowane podejście hiperheurystyczne może być stosowane równolegle z omawianą wcześniej metodą predykcyjną.

Podejście hiperheurystyczne było stosowane do rozwiązywania DVRP [41–43], ale stosowana w tych pracach hiperheurystyka należy do podgrupy generatorów heurystyk [20]. Podejście [41–43] jest zatem znacznie bliższe podejściu memetycznemu do optymalizacji DVRP [71], zaprezentowanemu wcześniej przez Onga i in. na przykładzie SVRP [21].

Podejście proponowane w tej rozprawie ma odmienny charakter, gdyż polega na wyborze algorytmu optymalizacyjnego jeszcze przed rozpoczęciem obliczeń, na podstawie



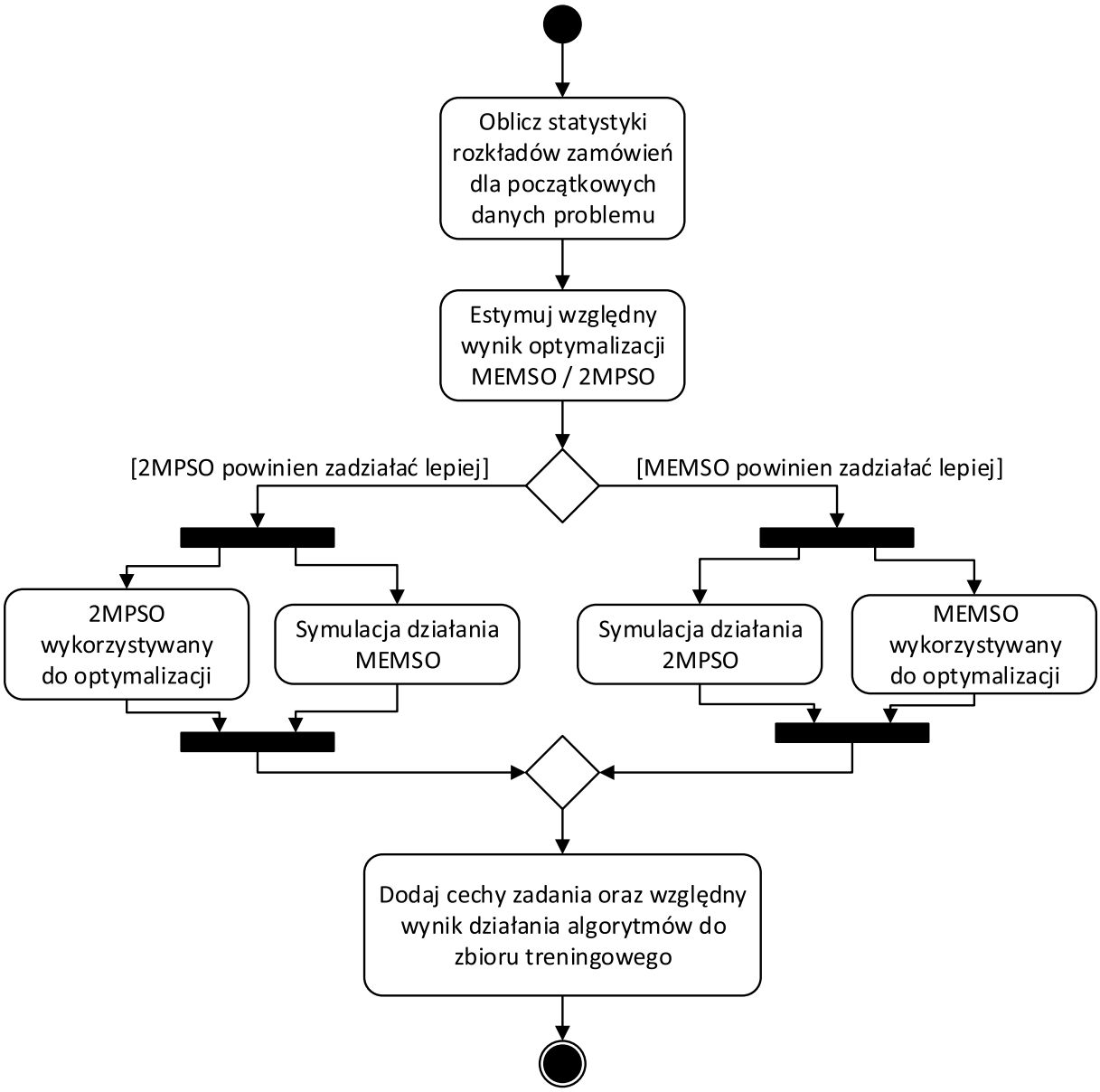
Rysunek 7.4: Porównanie względnych wyników uzyskiwanych na zbiorze problemów testowych za pomocą algorytmu MEMSO [58], GA [47], M-VRPDR [71], ContDVRP wykorzystującego PSO i kodowanie WCZ [84] (ozn. ContDVRP(PSO)), ContDVRP wykorzystującego DE i kodowanie WCZ (ozn. ContDVRP(DE)), ContDVRP wykorzystującego PSO i połączone kodowanie WRZ+WCZ [1] (ozn. ContDVRP(Ai)).

omówionych wcześniej charakterystyk rozkładu zamówień [87]. W związku z tym, idea proponowanego podejścia hiperheurystycznego opiera się na dwóch założeniach:

- charakterystyka problemu pozostaje niezmienna w ciągu dnia roboczego (zamówienia znane na początku pozwalają na dobrą estymację cech przyszłych zamówień),
- zidentyfikowana charakterystyka problemu ma związek z wynikami osiąganymi przez algorytm wykorzystujący dany sposób kodowania.

Niezmienność charakterystyki problemu została częściowo zweryfikowana poprzez skuteczność metody predykcyjnej. Ponadto obserwacja przykładowych przestrzennych rozkładów

zamówień z początku oraz z całości dnia roboczego, prezentowanych na Rysunku 7.2, potwierdza założenie o niewiele zmieniającej się charakterystyce problemu w trakcie procesu optymalizacji. Przesłanki, potwierdzające założenie o związku charakterystyki problemu z wynikami osiąganymi danym algorytmem, są zaprezentowane na Rysunku 7.4. Prezentowana na tym rysunku wzajemna zależność średnich względnych wyników uzyskanych dla poszczególnych instancji problemu sugeruje, że największe znaczenie dla wyniku na danej instancji testowej ma wybór przestrzeni przeszukiwań. Wyniki algorytmu ContDVRP, niezależnie od algorytmu optymalizacyjnego, są ze sobą silnie skorelowane. Podobną zależność, chociaż w znacznie mniejszym stopniu, można zaobserwować w grupie algorytmów MEMSO, GA i M-VRPDR, wykorzystujących dyskretne kodowanie. Natomiast pomiędzy wynikami tych dwóch grup można zaobserwować brak korelacji.



Rysunek 7.5: Diagram aktywności podejścia hiperheurystycznego opartego na algorytmach MEMSO i 2MPSO.

Wydruk 7.1: Parametry modelu liniowego przewidującego względny wynik algorytmów MEMSO i 2MPSO w oparciu o cechy zadania testowego, zaprezentowane jako rezultat dopasowania modelu liniowego z pakietu statystycznego R [104].

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.42899    0.15676   9.116 9.64e-07 ***
sd_x          1.23119    0.26551   4.637 0.000573 ***
mu_y         -1.11031    0.16881  -6.577 2.62e-05 ***
sd_y         -1.56756    0.28836  -5.436 0.000151 ***
skew_y       -0.04576    0.02481  -1.844 0.089977 .
mu_s          2.72795    1.25435   2.175 0.050362 .
sd_s         -3.39358    0.71918  -4.719 0.000498 ***
skew_s        0.21507    0.04592   4.683 0.000529 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.'
Residual standard error:
                0.03915 on 12 degrees of freedom
Multiple R-squared: 0.86, Adjusted R-squared: 0.7764
F-statistic: 10.43 on 7 and 12 DF, p-value: 0.000286
```

Najistotniejszą częścią podejścia hiperheurystycznego jest model predykcyjny, decydujący o wyborze algorytmu optymalizacyjnego. Diagram aktywności procesu decyzyjnego w podejściu hiperheurystycznym, jest przedstawiony na Rysunku 7.5. Aby wybór algorytmu przy rozwiązywaniu kolejnych instancji problemu był właściwy należy nie tylko prowadzić obliczenia zarówno algorytmem, który został wybrany, ale również przeprowadzić symulację działania drugiego algorytmu na tych samych danych. W ten sposób względny wynik obu algorytmów wraz z wyliczonymi cechami problemu poszerza zbiór treningowy modelu predykcyjnego.

Wybór algorytmu optymalizacyjnego jest dokonywany w oparciu o predykcję względnego wyniku algorytmów dostępnych w hiperheurystyce, poprzez model liniowy dopasowany na danych zaprezentowanych w Tabeli 7.1. W ogólnym przypadku, dla algorytmów $\mathcal{A}_1$ i $\mathcal{A}_2$ optymalizujących DVRP, jeżeli wartość zwróconej estymacji ich względnego wyniku $\left(\widehat{\frac{Res(\mathcal{A}_1)}{Res(\mathcal{A}_2)}}\right)(C_{t_{begin}}) > 1$, to do optymalizacji instancji DVRP o początkowych zamówieniach $C_{t_{begin}}$ wybierany jest algorytm $\mathcal{A}_1$ (natomiast $\mathcal{A}_2$ w przeciwnym przypadku). Uzasadnieniem realizacji predykcji w oparciu o model regresyjny, jest większy nacisk na poprawne wskazywanie instancji testowych, dla których algorytmy te osiągają istotnie różne wyniki. Model klasyfikacyjny powodowałby spłaszczenie tej informacji, a w konsekwencji potencjalnie gorsze wyniki.

Przykład współczynników modelu dopasowanego do stosunku średnich wyników uzyskiwanych przez MEMSO względem 2MPSO, wraz z selekcją zmiennych tego modelu przy użyciu kryterium informacyjnego Akaike [3], jest zaprezentowany na Wydruku 7.1. Dla al-

gorytmów MEMSO i 2MPSO model decyzyjny, zwracający estymację $\left(\frac{Res(\widehat{MEMSO})}{Res(2MPSO)}\right)$ względnego wyniku tych algorytmów, ma (zgodnie z Wydrukiem 7.1) następującą postać:

$$\left(\frac{Res(\widehat{MEMSO})}{Res(2MPSO)}\right)(C_{t_{begin}}) = 1.2sd_x - 1.1\mu_y - 1.6sd_y - 0.05skew_y + \\ + 2.7\mu_s - 3.4sd_s + 0.2skew_s + 1.4$$

Warto podkreślić, że model wskazał jako istotne dla wyboru algorytmu zarówno zmienne związane z położeniem, jak i rozmiarem zamówień.

Przy prezentacji wyników w podrozdziale 8.4.5 metoda hiperheurystyczna jest wykorzystana również dla algorytmów M-VRPDR [71] i ContDVRP, ale dla uzyskania dobrych rezultatów wymagała ona uwzględnienia dodatkowo liczby zamówień w cechach opisujących zadanie, względem pierwotnego zbioru cech prezentowanego w Tabeli 7.1.

Rozdział 8

System optymalizacyjny

W tym rozdziale opisane wcześniej elementy: metoda kodowania, algorytmy heurystyczne i metaheurystyczne, sposób budowania harmonogramów na podstawie zaproponowanego rozwiązania R oraz proponowana hiperheurystyka, są zaprezentowane jako elementy składowe systemu informatycznego rozwiązującego problem DVRP. Publicznie dostępna część źródeł oraz aplikacja wykonywalna jest dostępna pod adresem:
`https://sourceforge.net/p/continuous-dvrp/`.

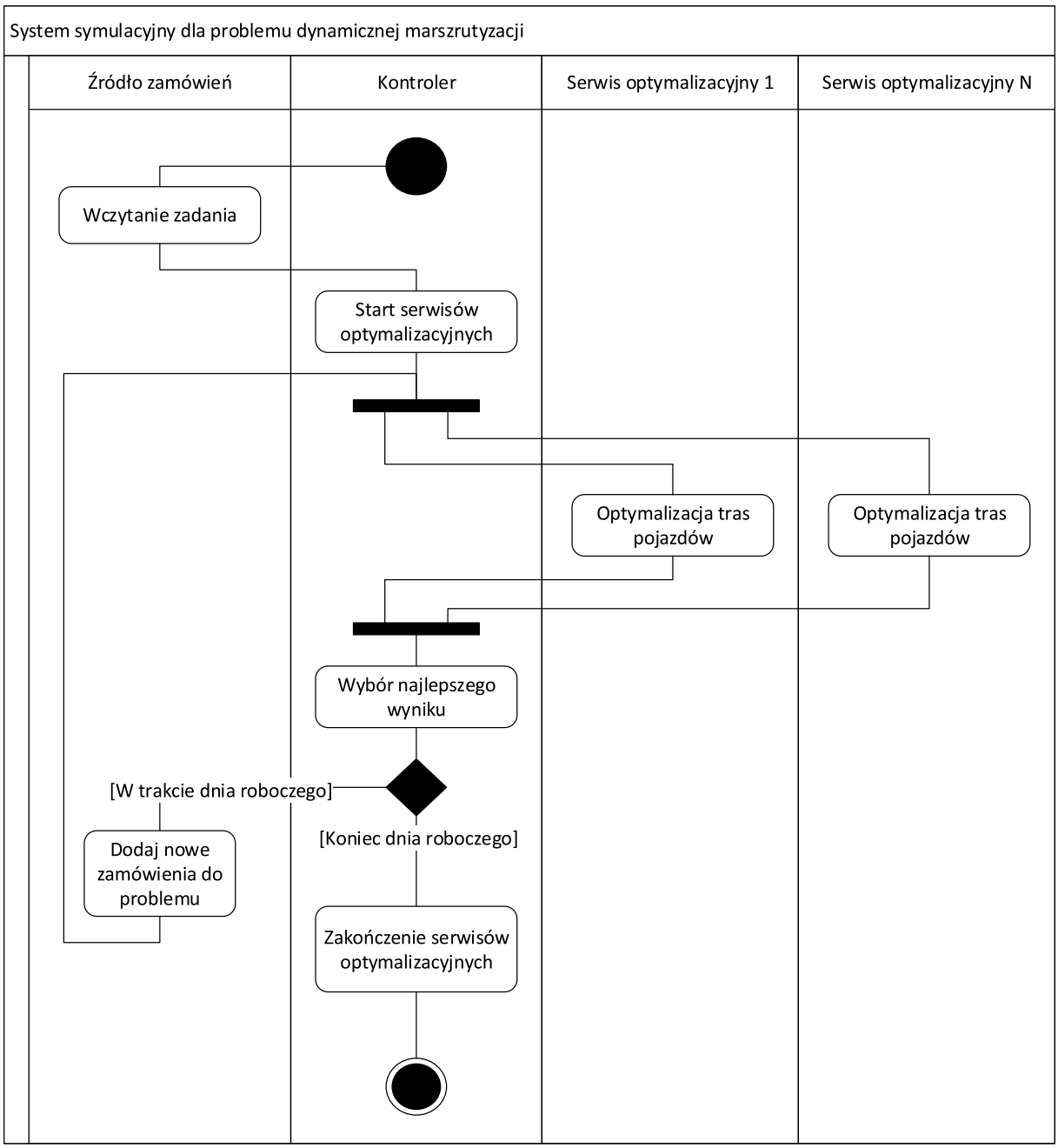
Zagadnienie budowy takiego systemu informatycznego na potrzeby wspomagania decyzji w zakresie DVRP było rozważane w pracach [66, 95, 113, 122]. Przedmiotem zainteresowania badaczy były również kwestie prawne, dotyczące ograniczeń czasu pracy kierowców, przekładające się na ograniczenia optymalizowanego problemu, które powinien uwzględniać tego rodzaju system przy stosowaniu do problemów rzeczywistych [61].

W rozprawie opisana jest wysokopoziomowa architektura takiego systemu symulacyjnego oraz zaproponowana metoda optymalizacyjna. W tym rozdziale zawarte są również wskazówki dotyczące implementacji przetwarzania równoległego wykorzystywanego w zaimplementowanych metodach.

8.1 Środowisko testowe i symulacyjne

W celu przetestowania proponowanych metod została stworzona aplikacja pełniąca rolę systemu symulacyjnego oraz serwisu optymalizacyjnego. Większa część systemu została stworzona w technologii `.NET` [80, 97, 101], z wyjątkiem części związanej z predykcją algorytmu w metodzie hiperheurystycznej zaimplementowanej jako skrypty w języku `R` [104]. Serwisy optymalizacyjne są wywoływane poprzez protokół `TCP` w oparciu o technologię zdalnego wywoływania metod w ramach `Windows Communication Foundation` [68]. Zaletą tego rodzaju implementacji jest uruchamianie serwisów optymalizacyjnych jako niezależnych procesów oraz możliwość potencjalnego rozproszenia obliczeń na wielu maszynach, bez konieczności dokonywania większych zmian w kodzie aplikacji. Uruchamianie serwisów jako niezależnych procesów, w przeciwieństwie do wątków, chroni przed zjawi-

skiem *false sharing* [12,127], skutkującym niepełnym wykorzystaniem czasu procesora, ze względu na nietrafione odczyty z jego pamięci podręcznej (ang. *cache*).



Rysunek 8.1: Wysokopoziomowy diagram aktywności systemu przetwarzającego zadania DVRP w oparciu o równolegle działające serwisy optymalizacyjne. Działanie pojedynczego serwisu optymalizacyjnego realizującego metodę ContDVRP jest zaprezentowane na Rysunku 8.2.

Ponadto potraktowanie serwisu optymalizacyjnego jako rodzaju czarnej skrzynki pozwala na hybrydyzację algorytmów, co zostało wykorzystane przy łączeniu podejścia opartego na generowaniu sztucznych zamówień i przypisywaniu ich przy użyciu zmodyfikowanego algorytmu Kruskala do czasu T_{CO}, a w dalszej części dnia roboczego płynne przejście do optymalizacji algorytmem 2MPSO (ContDVRP) [85].

8.2 Architektura rozwiązania

Całość zaimplementowanego systemu można podzielić na: moduł źródła danych, moduł kontrolera oraz moduł serwisu optymalizacyjnego. Wysokopoziomowy diagram aktywności całego systemu jest przedstawiony na Rysunku 8.1.

Źródłem danych, w zaimplementowanym podejściu, są pliki, których postać pierwotnie została zdefiniowana w [90][1]. Zadaniem kontrolera jest uruchamianie serwisów optymalizacyjnych oraz wybór i propagacja najlepszego rozwiązania. Pseudokod kontrolera jest przedstawiony w Algorytmie 8.3. Zadaniem serwisu optymalizacyjnego jest znalezienie estymacji optymalnego zbioru tras R dla zadanego stanu problemu DVRP. Serwis optymalizacyjny stanowi zatem właściwą implementację metody ContDVRP. Diagram aktywności pojedynczego serwisu optymalizacyjnego jest przedstawiony na Rysunku 8.2. Pseudokod pierwszej i drugiej fazy optymalizacji algorytmu ContDVRP przedstawiony jest odpowiednio w Algorytmach 8.4 i 8.5. Algorytm ContDVRP, stanowiący jeden z głównych wyników rozprawy, jest szczegółowo opisany w podrozdziale 8.2.2.

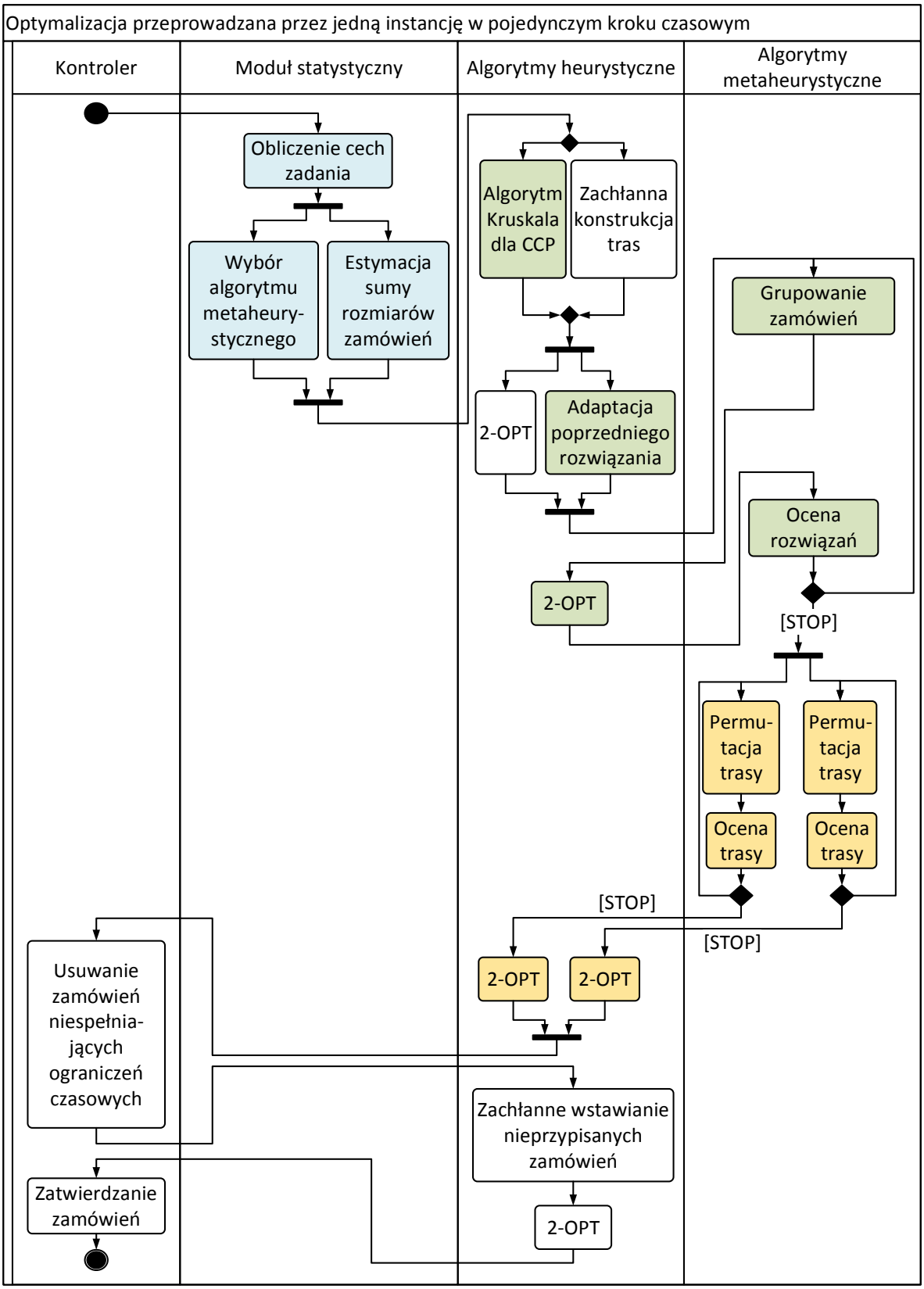
8.2.1 Kontroler

Algorytm 8.3 Psuedokod kontrolera serwisów optymalizacyjnych.

```
   {V_t := zbiór pojazdów dostępnych w czasie t}
   {C_t := zbiór zamówień oczekujących w czasie t}
1: while time ≤ end do
2:   for all optimizer ∈ optimizers do
3:     OptimizeRequestsAssignment(V_t, C_t) {(szczegóły przedstawione w Algorytmie 8.4)}
4:     OptimizeVehiclesRoutes(V_t, C_t) {(szczegóły przedstawione w Algorytmie 8.5)}
5:   end for
6:   bestSolution = ChooseBestSolution(optimizers)
7: end while
```

W każdym przedziale czasowym ContDVRP przetwarza zbiór zamówień oczekujących C_t. W każdym przedziale czasowym, aż do końca dnia (linia 1 w Algorytmie 8.3) moduł kontrolera uruchamia niezależne procesy optymalizacyjne (ozn. *optimizers*) (linia 2 Algorytmu 8.3). Procesy te są synchronizowane na koniec każdego przedziału czasowego, kiedy wybierane jest najlepsze znalezione rozwiązanie (ozn. *bestSolution*) (linia 6 Algorytmu 8.3).

[1] Opis tych plików oraz zbiory instancji testowych wykorzystywane w rozprawie są dostępne na stronie: `http://www.mini.pw.edu.pl/~mandziuk/dynamic/?page_id=67`



Rysunek 8.2: Diagram aktywności prezentujący kolejność przetwarzania przez poszczególne moduły w zaimplementowanym procesie optymalizacji. Moduły oznaczone białym kolorem są wykonywane niezależnie od konfiguracji metody optymalizacyjnej. Bazowa wersja algorytmu ContDVRP składa się z modułów białych i zielonych. Niebieskim kolorem oznaczone są moduły związane z wykorzystaniem danych problemu. Żółtym kolorem moduły związane z ciągłą optymalizacją kolejności zamówień.

8.2.2 Serwis optymalizacyjny realizujący algorytm ContDVRP

Diagram aktywności serwisu optymalizacyjnego, przedstawiony na Rysunku 8.2, obejmuje pełną możliwą ścieżkę przetwarzania zadanego stanu problemu VRP. Na tym diagramie aktywności oznaczone niebieskim kolorem wiążą się z wykorzystaniem danych problemu w metodach hiperheurystycznej i predykcyjnej. Aktywności oznaczone zielonym kolorem tworzą główny silnik optymalizacyjny ContDVRP, oparty o metaheurystyczne algorytmy optymalizacji globalnej oraz zmodyfikowany algorytm Kruskala rozwiązujący CCP. Aktywności oznaczone żółtym kolorem stanowią drugą (opcjonalną) fazę optymalizacji, w której trasa każdego z pojazdów poddana jest dodatkowej optymalizacji algorytmem metaheurystycznym.

Algorytm 8.4 Pseudokod pierwszej fazy optymalizacji (przypisań zamówień do pojazdów).

1: $radius \Leftarrow 2 \max_{l_1,l_2=k+1,\ldots,k+m} \rho(l_1, l_2)$
{rozwiązanie *heuristicSolution* jest konstruowane aby zwiększyć różnorodność puli rozwiązań oraz utrzymać wyższą jakość rozwiązania}
2: **if** USE_TREE_HEURISTIC **then**
3: $heuristicSolution \Leftarrow CapacitatedMinimalSpanningForest(C_t)$ {szczegóły w Algorytmie 4.2}
4: **else**
5: $heuristicSolution \Leftarrow CreateRandomGreedySolution(C_t)$
6: **end if**
7: **if** $t = 0$ **then**
8: $bestSolution \Leftarrow heuristicSolution$
9: **else**
10: $bestSolution \Leftarrow Adapt(bestSolution, heuristicSolution,$ DISCRETIZE_BEST$)$
11: **end if**
{część rozwiązań początkowych jest opartych na *heuristicSolution*, co zapewnia ograniczenie dolne na jakość rozwiązania}
{część rozwiązań początkowych opartych jest na *bestSolution*, co pozwala wykorzystać wiedzę o problemie z poprzedniego kroku czasowego}
{pozostałe rozwiązania początkowe są generowane w promieniu *radius* od *bestSolution*}
12: $optimizer \Leftarrow InitializeOptimizer(heuristicSolution, bestSolution, radius)$
13: **for** $i = 1, 2, \ldots, maxAssignmentIterations$ **do**
14: $PerformOptimizationStep(optimizer)$
15: **end for**
{znalezione rozwiązanie (ozn. *optimizerBestSolution*) jest traktowane jako podział zamówień pomiędzy pojazdy, wraz początkowymi trasami}
16: $optimizerBestSolution = GetBestSolution(optimizer)$

Każdy proces optymalizacyjny wywołuje najpierw procedurę $OptimizeRequestsAssignment(V_t, C_t)$ (przedstawioną w Algorytmie 8.4), która zwraca podział zamówień pomiędzy pojazdy, wraz z trasami przetworzonymi algorytmem 2–OPT. Następnie wywoływana jest procedura $OptimizeVehiclesRoutes(V_t, C_t)$ (przedstawiona w Algorytmie 8.5), któ-

rej zadaniem jest poprawienie trasy dla każdego z pojazdów i zapewnienie dopuszczalność zwracanego rozwiązania.

W optymalizacji podziału zamówień populacja początkowa w algorytmie optymalizacji ciągłej, składa się z: rozwiązania opartego na dyskretnym rozwiązaniu *heuristicSolution*, ciągłym rozwiązanu *bestSolution* zaadaptowanym z poprzedniego przedziału czasowego (linia 10 Algorytmu 8.4) oraz losowych rozwiązań wygenerowanych w promieniu *radius* od *bestSolution* (linia 11 Algorytmu 8.4). Dyskretne rozwiązanie *heuristicSolution* jest transformowane do przybliżonego rozwiązania w przestrzeni ciągłej poprzez proces odzyskiwania aproksymacji, opisany w podrozdziale 6.2.1. Proces transformacji polega na uśrednianiu lokalizacji zamówień przypisanych do jednego pojazdu, a następnie lekkim zaburzaniu tej lokalizacji. Dzięki temu każde z k skupień zamówień przypadających na pojazd ma nieco inną wartość. Najlepsze rozwiązanie z poprzedniego przedziału czasowego jest adaptowane do aktualnego stanu problemu na podstawie liczby pojazdów wykorzystanej w *heuristicSolution*. Jeżeli liczba pojazdów w *heuristicSolution* jest większa od liczby pojazdów w *bestSolution*, to *bestSolution* jest rozszerzone o losowo położone centroidy klastrów. Parametr DISCRETIZE_BEST decyduje o tym, czy centroidy klastrów są zachowywane (gdy FAŁSZ), czy też odzyskiwane poprzez uśrednianie lokalizacji zamówień oczekujących (gdy PRAWDA).

Algorytm 8.5 Pseudokod drugiej fazy optymalizacji (kolejności obsługi zamówień) i procedury naprawczej rozwiązań.

1: **for all** $vehicle \in optimizerBestSolution$ **do**
2: $vehicle.route \Leftarrow EnhanceWith2OPT(vehicle.route)$
3: $vehicleOptimizer \Leftarrow InitializeVehicleOptimizer(vehicle.route)$
4: **for** $i = 1, 2, \ldots, maxRouteIteration$ **do**
5: $PerformOptimizationStep(vehicleOptimizer)$
6: **end for**
7: $vehicle.route \Leftarrow GetBestRoute(vehicleOptimizer)$
8: $vehicle.route \Leftarrow EnhanceWith2OPT(vehicle.route)$
9: **end for**
{rozmieszczenie w sposób zachłanny zamówień łamiących ograniczenie powrotu do zajezdni (warunek (3.3))}
10: $optimizerBestSolution = RepairBestSolution(optimizerBestSolution)$
{zachowanie rozwiązania z kroku $t-1$, jeżeli w kroku t nie znaleziono lepszego, a stan problemu się nie zmieniał}
11: **if** $C_t \subseteq C_{t-1}$ AND $optimizerBestSolution > bestSolution$ **then**
12: $optimizerBestSolution \Leftarrow bestSolution$
13: **end if**

Po zainicjalizowaniu populacji początkowej, serwis optymalizacyjny iteracyjnie poszukuje coraz lepszych rozwiązań za pomocą algorytmu stanowiącego parametr metody ContDVRP[2]. Następnie najlepsze rozwiązanie *optimizerBestSolution* jest zwracane jako

[2]W rozprawie rozważane są algorytmy PSO i DE.

wynik tej fazy optymalizacji (linia 16 Algorytmu 8.4). W drugiej fazie optymalizacji trasy każdego pojazdów są poprawiane niezależnie od siebie (linie 1-9 Algorytmu 8.5). Z uwagi na fakt, że w ramach optymalizacji przypisania zamówień do pojazdów trasy były już optymalizowane algorytmem 2–OPT, dalsza optymalizacja w tym kroku jest opcjonalna. Obowiązkową procedurą jest natomiast weryfikacja spełnialności reguły powrotu do zajezdni na czas (warunek (3.3)). Ostatnie zamówienia z danego pojazdu, które bezpośrednio powodują złamanie tej reguły, są usuwane i w sposób zachłanny przypisywanego do innego (w szczególności nowego) pojazdu, tak aby ograniczenia problemu zostały zachowane (linia 10 Algorytmu 8.5). Jeżeli w wyniku procedury optymalizacyjnej znalezione rozwiązanie uległoby pogorszeniu w stosunku do poprzedniego rozwiązania, a stan problemu nie uległ zmianie, przywracane jest poprzednie najlepsze rozwiązanie (linie 11-12 Algorytmu 8.5).

Po przeprowadzeniu procesu optymalizacyjnego, sterowanie powraca do kontrolera, który zatwierdza do realizacji zamówienia, zgodnie z zasadami przedstawionymi w podrozdziale 3.5.

8.3 Dobór parametrów procesu optymalizacji

Wykorzystanie metody ContDVRP wymagało dostrojenia szeregu parametrów, obejmujących nie tylko parametry właściwe dla samej metody (liczba klastrów na pojazd, liczba równoległych serwisów optymalizacyjnych czy algorytm optymalizacyjny), ale również parametry algorytmów metaheurystycznych i parametry operacyjne samego problemu DVRP. W celu uzyskania jak najlepszych wyników na zbiorze testowych instancji problemu strojeniu podlegały parametry operacyjne zadania oraz parametry związane z kodowaniem i sposobem realizacji obliczeń. Wartości parametrów wykorzystane w bazowych konfiguracjach algorytmu ContDVRP są zaprezentowane w Tabeli 8.1.

Szczegółowemu strojeniu podlegały następujące parametry operacyjne:

- bufor czasowy T_{AC},
- liczba okien czasowych N_{TS}.

Parametry te mają bezpośredni wpływ na sposób budowania harmonogramów oraz częstotliwość uruchamiania serwisów optymalizacyjnych.

Szczegółowemu strojeniu podlegały również następujące parametry kodowania i sposobu realizacji obliczeń:

- liczba skupień zamówień k przypisana do jednego z pojazdu,
- stosunek liczby iteracji do liczności populacji w algorytmie optymalizacyjnym,
- liczba równoległych procesów optymalizacyjnych.

Parametry te mają wpływ na sposób wykorzystywania dostępnych zasobów, postać przestrzeni przeszukiwania i w konsekwencji na zbiór możliwych proponowanych rozwiązań.

Tabela 8.1: Wartości parametrów w bazowych konfiguracjach algorytmu ContDVRP.

Parametr	Wartość	
	Liczba ewaluacji	Limit czasu
ContDVRP		
k	2	1
liczba serwisów optymalizacyjnych	8	8
algorytm optymalizacyjny	PSO	DE
T_{co}	0.5	0.5
N_{ts}	40	40
T_{ac}	4%	4%
PSO		
g	0.60	0.60
l	2.20	2.20
a	0.63	0.63
$P(X$ jest sąsiadem $Y)$	0.50	0.50
liczba iteracji w 1./2. fazie	140/0	1.875/0 sek.
liczba cząstek	22	22
DE		
c	0.9	0.9
F	0.5	0.5
liczba iteracji w 1./2. fazie	195/0	1.875/0 sek.
liczba osobników	16	16

Większość eksperymentów dostrajających była wykonana przy użyciu algorytmu PSO, jako głównego algorytmu optymalizacyjnego. Wyjątkiem był dobór liczby iteracji w stosunku do liczności populacji, gdzie przeprowadzone zostały testy zarówno dla PSO, jak i DE, gdyż algorytmy te mają w zakresie tych parametrów istotnie różną charakterystykę działania. Szczegółowe wyniki eksperymentów wykonanych dla doboru wyżej wymienionych parametrów znajdują się w Dodatku A.

Należy jeszcze dodać, że wartość parametrów sterujących PSO i DE została ustalona na podstawie wskazań literatury oraz małej liczby eksperymentów początkowych. Liczba iteracji i wielkość populacji oraz kryterium stopu w przypadku limitu czasowego, wynikają z limitów wykorzystywanych w konkurencyjnych podejściach literaturowych do rozwiązywania DVRP. Wartość czasu odcięcia $T_{CO} = 0.5$, jako definiująca faktyczną instancję problemu, musi być zachowana w celu zapewnienia porównywalności wyników ContDVRP z wynikami literaturowymi. Warto w tym miejscu wspomnieć, że adaptacja statycznych instancji VRP wykonana przez Kliby'ego polegała na wylosowaniu czasów pojawiania się zamówień z rozkładu jednostajnego na przedziale $[t_{begin}, t_{end}]$. Za zamówienia znane w chwili t_{begin} w tych instancjach, przyjmuje się te zamówienia, których czas zgłoszenia zamówienia t_i został ustalony na późniejszy, niż czas progu odcięcia $t_{begin} + T_{CO}(t_{end} - t_{begin})$.

8.4 Testy algorytmu ContDVRP

Proponowane w rozprawie podejście ContDVRP polegające na wykorzystaniu ciągłej reprezentacji WCZ problemu marszrutyzacji, przypisującej wiele skupisk zamówień do jednego pojazdu, jest osadzone w środowisku równoległych niezależnych serwisów optymalizacyjnych. W celu zaprezentowania własności podejścia ContDVRP został wykonany szereg testów porównawczych, zarówno pomiędzy poszczególnymi konfiguracjami, jak i z wynikami literaturowymi innych algorytmów metaheurystycznych. Wszystkie testy były prowadzone na 21 instancjach zaadaptowanych z instancji statycznych [59]. Dla każdej z instancji testowych zostało przeprowadzonych 30 eksperymentów (dla każdego z zestawów parametrów).

Niezależne eksperymenty, bazujące wyłącznie na średnich wynikach, zostały wykonane dla dokonywania wyboru algorytmu optymalizacyjnego w metodzie hiperheurystycznej.

W dalszej części rozdziału zaprezentowane są:

- miary jakości algorytmów,
- analiza poszczególnych komponentów metody,
- analiza dynamizmu problemu z punktu widzenia różnych kodowań i funkcji jakości,
- porównanie z wynikami literaturowymi w 3 układach:
  - porównanie z budżetem 10^6 ewaluacji funkcji jakości w całym procesie optymalizacji (wyniki literaturowe MAPSO i MEMSO),
  - porównanie z budżetem 10^7 ewaluacji funkcji jakości w całym procesie optymalizacji (wyniki literaturowe M-VRPDR),
  - porównanie z kryterium stopu 750 sekund (dla Intel Pentium IV@2.8GHz) na cały proces optymalizacji (wyniki literaturowe ACS, GA i EH),
- wyniki zastosowania podejścia hiperheurystycznego dla ContDVRP i algorytmów literaturowych, osiągających porównywalnej jakości wyniki względem odpowiadającej im konfiguracji ContDVRP.

8.4.1 Miary jakości algorytmów

W celu porównania wyników osiąganych przez poszczególne konfiguracje algorytmu ContDVRP oraz algorytmy literaturowe wykorzystywane są przede wszystkim średnie wyniki algorytmu $\mathcal{A}_i$ dla instancji $DVRP_j$, oznaczone jako $Res(\mathcal{A}_i, DVRP_j)$. Średnie wyniki wydają się być najbardziej zasadną miarę porównania w problemach dynamicznych, gdzie w sytuacji praktycznej nie ma możliwości powtarzania obliczeń. W tym kontekście, najlepsze rezultaty osiągnięte przez ContDVRP (zaprezentowane w Dodatku B) służą jedynie prezentacji możliwości metody i przyjętego kodowania.

W celu umożliwienia agregacji wyników algorytmu $\mathcal{A}_i$ uzyskanych dla różnych testowych instancji problemu $DVRP_j$, wprowadzone zostają pojęcia najlepszego znanego wyniku $BestRes(DVRP_j)$ oraz względnego średniego wyniku $RelRes(\mathcal{A}_i, DVRP_j)$. War-

tości te są zdefiniowane następująco:

$$BestRes(DVRP_j) := \min_i Res(\mathcal{A}_i, DVRP_j) \tag{8.1}$$

$$RelRes(\mathcal{A}_i, DVRP_j) := \frac{Res(\mathcal{A}_i, DVRP_j)}{BestRes(DVRP_j)} \tag{8.2}$$

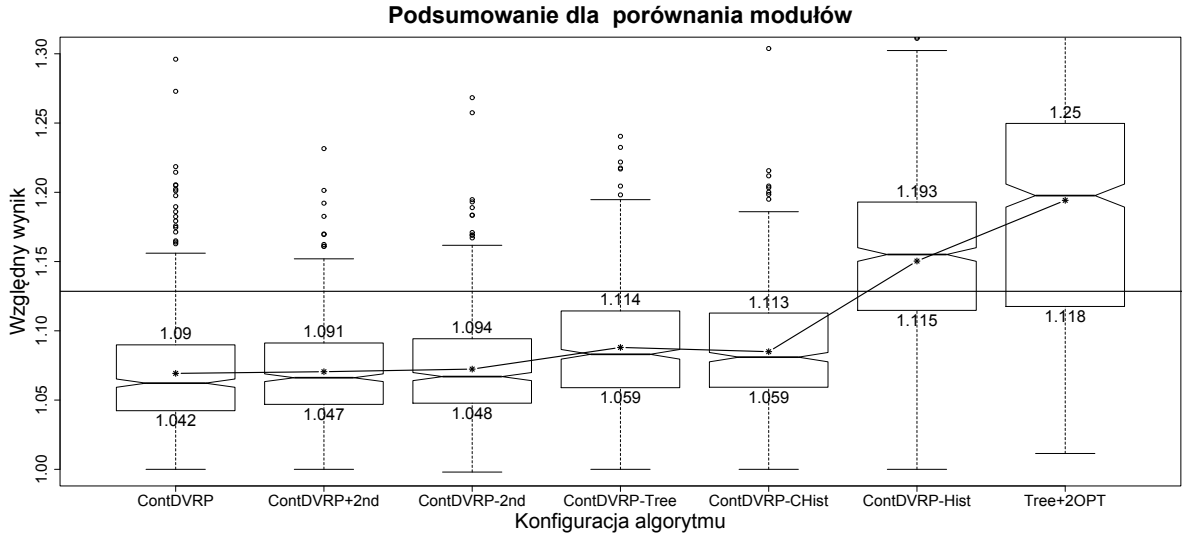
Względne średnie wyniki algorytmu $\mathcal{A}_i$ mogą zostać zagregowane do wartości $Perf(\mathcal{A}_i)$ reprezentującej sumaryczną skuteczność algorytmu na zbiorze instancji testowych:

$$Perf(\mathcal{A}_i) := \sum_j RelRes(\mathcal{A}_i, DVRP_j) \tag{8.3}$$

Rozprawa porównuje poszczególne algorytmy w oparciu o liczbę instancji, dla których dana metoda osiągnęła statystycznie lepszy średni wynik oraz ich sumaryczną skuteczność na zbiorze instancji testowych.

8.4.2 Wpływ komponentów algorytmu na wyniki optymalizacji

Jednym z podstawowych wyników rozprawy jest wskazanie istotności transferu wiedzy pomiędzy krokami czasowymi, wykorzystywanie w populacji rozwiązań wskazanych przez algorytm heurystyczny oraz ciągłej optymalizacji metaheurystycznej w pierwszej fazie algorytmu (przypisywaniu zamówień do pojazdów). Wyniki eksperymentów z tym związanych zaprezentowane są w Tabeli 8.2 oraz na Rysunku 8.3.



Rysunek 8.3: Rozkład względnych wyników dla różnych konfiguracji algorytmu ContDVRP.

Tabela 8.2: Zestawienie średnich wyników z wykorzystaniem różnych modułów przedstawionych na Rysunku 8.2. Szare tło oznacza wynik statystycznie istotnie gorszy od najlepszego średniego wyniku (w oparciu o test t-Studenta).

	$ContDVRP$	$ContDVRP$ $+2nd$	$ContDVRP$ $-2nd$	$ContDVRP$ $-Tree$	$ContDVRP$ $-CHist$	$ContDVRP$ $-Hist$	$Tree$ $+2OPT$
	Śr.	Śr.	Śr.	Śr.	Śr.	Śr.	Śr.
c50	578.31	579.65	581.62	584.25	**576.48**	605.64	736.66
c75	**903.72**	905.72	907.64	927.1	906.5	999.11	1121.94
c100	933.46	**926.56**	931.43	939.31	930.17	1014.72	1102.27
c100b	845.8	843.52	841.15	843.89	838.74	838.36	**835.05**
c120	**1071.38**	1081.9	1090.51	1119.76	1085.82	1087.68	1100.35
c150	**1134.2**	1146.07	1143.32	1193.8	1145.67	1243.95	1278.69
c199	1408.7	1415.27	1407.89	1441.17	**1403.86**	1608.39	1640.37
f71	298.5	**291.04**	291.24	291.23	294.57	301.94	332.05
f134	11892	11904.48	**11880.04**	11974.4	11992.36	12188.79	12928.45
tai75a	**1805.03**	1818.46	1844.97	1823.18	1828.85	1916.67	2011.77
tai75b	1422.6	1419.48	**1408.43**	1423.05	1485.56	1535.72	1605.2
tai75c	1510	**1504.51**	1513.15	1541.31	1520.98	1594.42	1665.86
tai75d	**1433.25**	1438.07	1441.79	1442.08	1454.6	1496.25	1480.29
tai100a	**2216.23**	2235.43	2259.82	2308.59	2277.29	2515.87	2582.84
tai100b	**2136.8**	2162.21	2142.82	2194.7	2165.55	2356.51	2356.64
tai100c	**1494.72**	1497.99	1499.23	1517.49	1536.52	1630.28	1570.9
tai100d	**1727.95**	1736.32	1746.72	1769.5	1757.42	1953.39	2063.46
tai150a	**3530.82**	3558.19	3558.34	3540.63	3718.11	3956.46	3735.37
tai150b	3026.89	3046.21	**3019.01**	3135.83	3096.83	3274.73	3472.06
tai150c	2603.53	**2583.87**	2611.98	2675.97	2698.73	2833.65	2678.9
tai150d	3009.01	**2985.12**	2995.35	3061.67	3090.26	3251.66	3354.67
sum	**44982.9**	45080.07	45116.45	45748.91	45804.87	48204.19	49653.79

Przeprowadzony eksperyment porównuje 7 konfiguracji metody ContDVRP, z wykorzystaniem algorytmu PSO i budżetem na obliczenia wynoszącym 10^6 ewaluacji funkcji jakości w eksperymencie bazowym. Eksperyment weryfikuje na ile istotne są:

- metaheurystyczna optymalizacja przypisania zamówień do pojazdów, stanowiąca istotę algorytmu ContDVRP,
- metaheurystyczna optymalizacja kolejności zamówień w pojazdach (ozn. $2nd$),
- generowanie rozwiązania heurystycznego dla aktualnego stanu problemu w każdym kroku czasowym i inicjalizowanie jednego z osobników populacji początkowych w oparciu o to rozwiązanie (ozn. $Tree$),
- bezpośredni transfer rozwiązania z poprzedniego kroku czasowego (ozn. $CHist$),
- pośredni transfer rozwiązania z poprzedniego kroku czasowego, z wykorzystaniem odzyskiwania aproksymacji opisanej w podrozdziale 6.2.1 (ozn. $Hist$),

Konfiguracja oznaczona w tabeli i na wykresie jako $ContDVRP$, jest bazową konfiguracją ContDVRP z wartościami parametrów zdefiniowanymi w Tabeli 8.1. Konfiguracja $PSO+2nd$ różni się od bazowej wykorzystaniem 1/6 budżetu ewaluacji funkcji jakości na aktywności związane z optymalizacją tras (wartości 120/20 w pozycji "liczba iteracji w 1./2. fazie" w tabeli parametrów). Konfiguracja $PSO-2nd$ różni się od $PSO+2nd$ brakiem optymalizacji ciągłej w drugiej fazie (wartości 120/0 w pozycji "liczba iteracji w 1./2.

fazie" w tabeli parametrów). W konfiguracji $PSO - Tree$ nie są wykonywane obliczenia zmodyfikowanym algorytmem Kruskala (liczba potrzebnych pojazdów jest estymowana z przykładowego rozwiązania zachłannego). W konfiguracji $PSO - Hist$ nie jest przekazywane rozwiązanie z poprzedniego kroku czasowego, natomiast w $PSO - CHist$ to rozwiązanie jest przekazywane jedynie poprzez odzyskiwanie aproksymacji. W $Tree + 2OPT$ nie jest wykonywana optymalizacja metaheurystyczna, ani nie są przekazywane rozwiązania z poprzedniego kroku czasowego.

Zarówno z Tabeli 8.2 prezentującej średnie wyniki, jak i z Rysunku 8.3 prezentującego wykres rozkładów względnych wyników można zaobserwować, że druga faza optymalizacji pozostaje bez większego wpływu na rezultaty, a przeznaczenie całości budżetu na optymalizację w pierwszej fazie pozwala na osiągnięcie większej liczby lepszych średnich wyników (chociaż o dłuższym prawym ogonie rozkładu).

Natomiast usunięcie któregokolwiek z modułów: konstruowania rozwiązań zmodyfikowanym algorytmem Kruskala, przekazywania rozwiązań z poprzedniego kroku czasowego (zarówno ciągłego jak i poprzez fazę dyskretyzacji), czy optymalizacji metaheurystycznej w pierwszej fazie, powoduje znaczne pogorszenie uzyskiwanych wyników.

Tabela 8.3: Porównanie z algorytmami MAPSO i MEMSO oparte na całkowitym budżecie 10^6 ewaluacji funkcji jakości. Liczby w nawiasach oznaczają odpowiednio: liczbę przedziałów czasowych, liczbę serwisów optymalizacyjnych oraz liczbę ewaluacji funkcji jakości w pojedynczym kroku czasowym w jednym serwisie optymalizacyjnym.

	$MAPSO$ $(25 * 8 * (0.5 * 10^4))$		$MEMSO$ $(25 * 8 * (0.5 * 10^4))$		$ContDVRP_{PSO}^{k=2}$ $(40 * 8 * (0.31 * 10^4))$		$ContDVRP_{DE}^{k=2}$ $(40 * 8 * (0.31 * 10^4))$	
	Min	Śr.	Min	Śr.	Min	Śr.	Min	Śr.
c50	571.34	610.67	577.60	592.95	**544.11**	**578.31**	563.09	580.02
c75	931.59	965.53	928.53	962.54	884.43	**903.72**	**878.32**	908.28
c100b	866.42	882.39	864.19	878.81	**819.56**	845.80	**819.56**	**843.83**
c100	953.79	973.01	949.83	968.92	902.00	**933.46**	**893.24**	937.07
c120	1223.49	1295.79	1164.63	1284.62	**1053.18**	**1071.38**	1067.64	1104.61
c150	1300.43	1357.71	1274.33	1327.24	**1098.03**	**1134.20**	1103.46	1140.14
c199	1595.97	1646.37	1600.57	1649.17	**1362.65**	**1408.70**	1379.34	1414.85
f71	287.51	296.76	283.43	**294.85**	274.16	298.50	**273.00**	295.58
f134	15150.50	16193.00	14814.10	16083.82	11746.40	**11892.00**	**11743.05**	11916.70
tai75a	1794.38	1849.37	1785.11	1837.00	**1685.23**	**1805.03**	1725.11	1807.49
tai75b	1396.42	1426.67	1398.68	1425.80	**1365.36**	1422.60	1372.39	**1412.85**
tai75c	1483.10	1518.65	1490.32	1532.45	1439.02	1510.00	**1423.21**	**1501.64**
tai75d	1391.99	**1413.83**	**1342.26**	1448.19	1408.79	1433.25	1416.89	1451.04
tai100a	2178.86	2214.61	2170.54	**2213.75**	**2137.30**	2216.23	2193.61	2247.21
tai100b	2140.57	2218.58	2093.54	2190.01	2060.65	**2136.80**	**2041.46**	2145.85
tai100c	1490.40	1550.63	1491.13	1553.55	1458.81	**1494.72**	**1442.63**	1495.87
tai100d	1838.75	1928.69	1732.38	1895.42	**1663.87**	**1727.95**	1694.79	1736.33
tai150a	3273.24	3389.97	**3253.77**	**3369.48**	3338.71	3530.82	3346.17	3489.65
tai150b	**2861.91**	**2956.84**	2865.17	2959.15	2910.06	3026.89	2914.79	3038.63
tai150c	2512.01	2671.35	2510.13	2644.69	**2497.65**	2603.53	2505.70	**2563.01**
tai150d	**2861.46**	**2989.24**	2872.80	3006.88	2869.79	3009.01	2890.02	2999.43

Tabela 8.4: Porównanie z M-VRPDR oparte na całkowitym budżecie 10^7 ewaluacji funkcji jakości. Liczby w nawiasach oznaczają odpowiednio: liczbę przedziałów czasowych, liczbę serwisów optymalizacyjnych oraz liczbę ewaluacji funkcji jakości w pojedynczym kroku czasowym w jednym serwisie optymalizacyjnym.

	$M-VRPDR$ [71] (10^7)		$ContDVRP_{PSO}^{k=2}$ $(40*8*(0.31*10^5))$		$ContDVRP_{DE}^{k=2}$ $(40*8*(0.31*10^5))$	
	Min	Śr.	Min	Śr.	Min	Śr.
c50	**524.61**	**548.10**	562.76	573.81	550.68	573.55
c75	**852.95**	**885.00**	866.34	894.71	879.20	906.41
c100b	820.92	864.03	**819.56**	836.23	**819.56**	**834.98**
c100	**860.56**	**913.81**	886.52	925.47	900.81	920.70
c120	1189.06	1295.31	**1059.40**	**1083.01**	1064.86	1127.46
c150	**1083.79**	1142.24	1083.86	**1114.09**	1100.50	1126.47
c199	1377.26	1466.30	**1341.47**	**1375.77**	1382.29	1458.59
f71	**260.17**	297.61	272.86	**292.88**	279.03	294.42
f134	11815.95	12299.59	**11735.98**	**11768.98**	11751.51	12015.79
tai75a	**1656.12**	**1705.76**	1680.55	1757.12	1686.80	1761.55
tai75b	**1350.28**	**1378.47**	1357.32	1382.52	1379.85	1411.84
tai75c	**1355.41**	**1427.74**	1415.18	1477.71	1416.05	1474.95
tai75d	**1376.42**	**1414.93**	1398.80	1423.15	1404.56	1434.20
tai100a	**2093.63**	**2168.87**	2118.38	2200.03	2132.30	2209.93
tai100b	**1990.99**	2097.86	2013.74	**2088.83**	2032.49	2103.91
tai100c	**1421.50**	**1460.72**	1428.99	1466.16	1460.76	1495.67
tai100d	**1631.63**	**1710.44**	1633.32	1710.59	1659.79	1726.70
tai150a	3226.51	**3354.09**	**3212.62**	3370.91	3307.59	3429.35
tai150b	2847.08	2980.33	**2842.36**	**2954.84**	2939.20	3056.80
tai150c	**2427.53**	2558.99	2451.53	2522.03	2456.71	**2508.01**
tai150d	**2737.37**	2903.68	2795.46	**2884.79**	2858.18	2981.39

8.4.3 Porównanie z wynikami literaturowymi

Aby zaprezentować wyniki działania algorytmu na tle wyników literaturowych zostały wykonane 3 rodzaje eksperymentów. Podstawowy rodzaj eksperymentu wykorzystuje ograniczenie 10^6 na liczbę ewaluacji funkcji jakości z równym podziałem liczby ewaluacji pomiędzy przedziały czasowe i serwisy optymalizacyjne. Kolejny eksperyment wykorzystuje 10^7 ewaluacji funkcji jakości. Trzeci rodzaj eksperymentu wykorzystuje czas jako kryterium stopu procesu optymalizacji.

Wielkość ograniczenia czasowego została dobrana w oparciu o wyniki szybkości obliczeniowej poszczególnych modeli procesorów zmierzone benchmarkiem Geekbench[3]. Jako punkt odniesienia został wybrany jednoprocesorowy Intel Pentium IV taktowany z częstotliwością 2.8GHz, wykorzystywany do obliczeń przez algorytm GA [47]. Do badań prowadzonych w tej rozprawie były wykorzystywane komputery wyposażone w czterordzeniowe procesory Intel Core i7 drugiej generacji z technologią *hyper-threading*, taktowane z częstotliwością 3.4GHz. Ze względu na to, że ten model procesora, wykorzystując obliczenia równoległe, uzyskiwał dziesięciokrotnie lepsze wyniki w testach wydajności niż referencyjny Intel Pentium IV, ograniczenie czasowe zostało ustawione na dziesięciokrotnie

[3] `https://browser.primatelabs.com/`

Tabela 8.5: Porównanie z algorytmami GA, TS i EH uruchomionymi z limitem czasu na obliczenia. Limit czasu został dobrany zgodnie ze względną mocą obliczeniową poszczególnych procesorów.

	GA [47] 750 sekund Intel Pentium IV @2.8GHz		TS [47] 750 sekund Intel Pentium IV @2.8GHz		EH [43] 250 sekund Athlon 64 @2.2 GHz		$ContDVRP^{k=1}_{PSO}$ 75 sekund Intel Core i7(2nd) @3.4GHz		$ContDVRP^{k=1}_{DE}$ 75 sekund Intel Core i7(2nd) @3.4GHz	
	Min	Śr.	Min	Śr.	Min	Śr.	Min	Śr.	Min	Śr.
c50	570.89	593.42	603.57	627.90	597.72	632.71	**553.15**	579.65	555.05	**578.58**
c75	981.57	1013.45	981.51	1013.82	979.29	1019.05	**878.41**	908.39	890.08	**907.47**
c100b	881.92	900.94	891.42	932.14	956.67	1020.02	**819.60**	848.13	820.62	**839.50**
c100	961.10	987.59	997.15	1047.60	975.20	1003.95	899.12	**933.77**	**894.89**	939.61
c120	1303.59	1390.58	1331.80	1468.12	1245.94	1372.45	**1054.26**	**1090.80**	1060.76	1116.12
c150	1348.88	1386.93	1318.22	1401.06	1342.91	1413.05	**1112.76**	**1150.60**	1119.32	1154.64
c199	1654.51	1758.51	1750.09	1783.43	1689.55	1747.02	1393.09	1452.65	**1385.30**	**1449.77**
f71	301.79	309.94	280.23	306.33	287.99	299.58	**272.44**	**292.01**	272.56	296.03
f134	15528.81	15986.84	15717.90	16582.04	14801.60	14952.66	**11827.12**	**12061.88**	11860.33	12069.21
tai75a	1782.91	1856.66	1778.52	1883.47	1769.75	1859.25	**1737.55**	**1811.95**	1742.63	1816.30
tai75b	1464.56	1527.77	1461.37	1587.72	1450.45	1502.09	**1405.41**	1457.89	1410.29	**1441.39**
tai75c	1440.54	1501.91	**1406.27**	1527.80	1685.10	1779.08	1418.65	1504.28	1436.31	**1492.03**
tai75d	**1399.83**	**1422.27**	1430.83	1453.50	1432.92	1445.89	1411.52	1451.73	1419.51	1453.04
tai100a	2232.71	2295.61	2208.85	2310.37	2227.43	2309.90	2174.40	**2231.23**	**2149.22**	2234.19
tai100b	2147.70	2215.39	2219.28	2330.52	2183.38	2221.40	**2047.04**	**2137.25**	2081.83	2154.17
tai100c	1541.28	1622.66	1515.10	1604.18	1656.97	1756.25	**1456.09**	**1511.10**	1463.07	1518.16
tai100d	1834.60	1912.43	1881.91	2026.76	1834.47	2029.45	**1694.10**	1741.20	1697.33	**1732.13**
tai150a	**3328.85**	3501.83	3488.02	3598.69	3346.02	**3487.78**	3450.58	3618.90	3381.65	3528.20
tai150b	2933.40	3115.39	3109.23	3215.32	**2874.72**	**3068.64**	2946.05	3094.30	2937.64	3087.32
tai150c	2612.68	2743.55	2666.28	2913.67	2583.13	2731.14	**2480.72**	2607.10	2480.86	**2564.34**
tai150d	2950.61	**3045.16**	2950.83	3111.43	3084.58	3252.03	2928.05	3096.37	**2891.29**	3067.26

mniejsze, niż w eksperymentach literaturowych przeprowadzonych na tym referencyjnym procesorze.

Dla wszystkich eksperymentach szare tło w ich średnich wynikach oznacza statystycznie istotnie lepszy wynik dla danej konfiguracji ContDVRP niż dla każdego z podejść literaturowych lub literaturowy wynik statystycznie istotnie lepszy niż dla każdej z konfiguracji ContDVRP. Statystyczna istotność była testowana za pomocą t-testu z poprawką Bonferroniego. Konfiguracje ContDVRP są prezentowane z podaniem limitu ewaluacji funkcji jakości (lub limitem czasu obliczeń), liczby k skupisk zamówień przypadających na pojazd oraz wykorzystywanego algorytmu optymalizacyjnego (PSO lub DE).

Podstawowe porównanie dotyczy algorytmów MAPSO [55] i MEMSO [58] i jest zaprezentowane w Tabeli 8.3. Porównanie ze zwiększonym budżetem na liczbę ewaluacji funkcji jakości dotyczy algorytmu M-VRPDR [71] i jest zaprezentowane w Tabeli 8.4. Natomiast porównanie z limitem czasu na obliczenia dotyczy algorytmów GA i TS [47] oraz EH [43] i jest przedstawione w Tabeli 8.5.

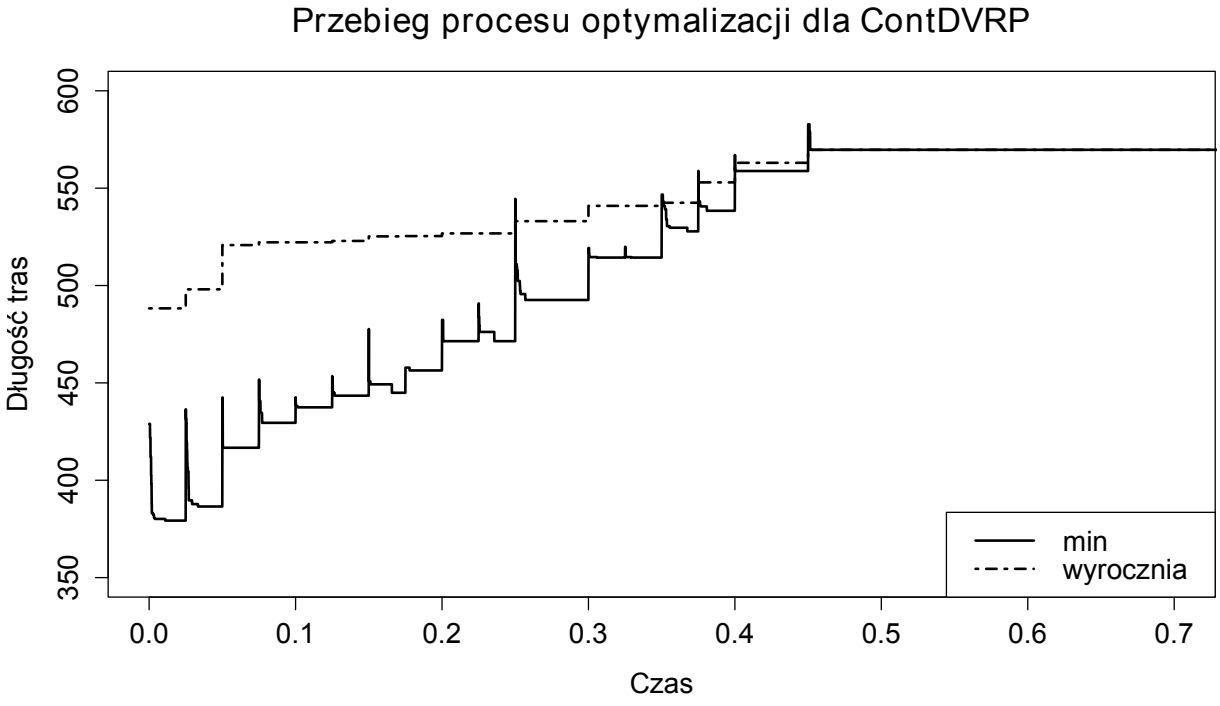
Porównując średnie wyniki w tych trzech zestawieniach można zaobserwować, że ContDVRP, zarówno wykorzystujący PSO, jak i DE, osiąga więcej statystycznie istotnie lepszych średnich wyników niż pozostałe metody (z wyjątkiem porównania pomiędzy M-VRPDR a ContDVRP$_{DE}$). Jednak porównując wyniki algorytmu M-VRPDR względem

wyników algorytmu ContDVRP, warto zwrócić uwagę na to, że ContDVRP osiąga tym lepsze wyniki, względem wyników M-VRPDR, im więcej jest zamówień w danej instancji problemu. Podejście ContDVRP ma też przewagę nad pozostałymi metodami w sumarycznej skuteczności, oraz w osiąganych wartościach minimalnych (ustępując jedynie metodzie M-VRPDR). Przy czym duża liczba najlepszych wyników wskazanych przez M-VRPDR, wynika prawdopodobnie z wykorzystania w tym algorytmie mieszaniny różnych heurystyk lokalnego przeszukiwania.

Szczegółowe zestawienie liczby najlepszych wyników, liczby statystycznie istotnie lepszych rezultatów oraz sumarycznej skuteczności zaprezentowane jest w Tabeli 9.1 w Rozdziale 9.

8.4.4 Dynamizm problemu

Dynamizm problemu DVRP jest zaprezentowany w oparciu o obserwowane zmiany rozwiązań wskazywanych jako optymalne dla pośrednich stanów problemu oraz wartości funkcji jakości dla tych estymacji położenia optimum. Autor rozprawy zaimplementował również serwis optymalizacyjny realizujący algorytm GA dla DVRP [47], aby mógł stanowić dodatkowy punkt odniesienia dla wyników uzyskiwanych przez ContDVRP.



Rysunek 8.4: Przykładowy przebieg procesu optymalizacji dla instancji c50 z limitem 10^7 ewaluacji funkcji jakości. *min* to wynik wybranego przez algorytm optimum, *wyrocznia* to długość tras z rozwiązania finalnego wskazanego przez algorytm, bez zamówień jeszcze nieznanych w danym przedziale czasowym.

W celu przeprowadzenia analizy dynamizmu DVRP został zarejestrowany przebieg procesu optymalizacji z wykorzystaniem ContDVRP oraz GA, uruchomionych w środo-

wisku równoległych procesu optymalizacyjnych. Zapisane zostały rozwiązania w postaci kodowania przypisania zamówień do pojazdów wskazywane jako optymalne na koniec każdego z przedziałów czasowych oraz wartości optimum w kolejnych iteracjach w każdym z serwisów optymalizacyjnych. W dalszej części podrozdziału, poziom dynamizmu problemu jest zaprezentowany w oparciu o zmienność liczby zamówień oczekujących, liczbę wykorzystywanych pojazdów oraz różnice pomiędzy rozwiązaniami dla poszczególnych stanów DVRP z kolejnych kroków czasowych. Odległość pomiędzy tymi estymacjami położenia optimum dla pośrednich stanów problemu została obliczona przy użyciu miary zaprezentowanej we wzorze 5.1.

Dynamizm problemu a liczba ewaluacji funkcji jakości

Porównując wyniki ContDVRP z Tabeli 8.3 (limit 10^6 ewaluacji funkcji jakości) z wynikami z Tabeli 8.4 (limit 10^7 ewaluacji funkcji jakości) można zauważyć, że zwiększenie liczby ewaluacji poprawia jakość finalnego wyniku (szczegółowe porównanie znajduje się w Dodatku A.6). Z drugiej strony, z przykładowego przebiegu działania algorytmu dla większego budżetu ewaluacji funkcji jakości (Rysunek 8.4), można zaobserwować, że pośrednie rozwiązania są nadmiernie dopasowane do aktualnego stanu problemu (silnie odbiegając od wartości jaką miałoby rozwiązanie pośrednie, gdyby było częścią rozwiązania finalnego). Zestawienie tych dwóch faktów prowadzi do wniosku (sprzecznego z intuicją), że dokładniejsza optymalizacja lokalnych stanów prowadzi do lepszej jakości finalnego rozwiązania pomimo nadmiernego dopasowania pośrednich rozwiązań do stanów problemu, w których nie są znane wszystkie zamówienia (przykład takiego pośredniego rozwiązania zaprezentowany jest na Rysunku 7.3).

Częściowe wyjaśnienie tej obserwacji można dostrzec na prezentowanym w początkowej części pracy harmonogramie floty pojazdów (Rysunek 3.3). Przedstawiony tam harmonogram rozwiązania o dobrej finalnej jakości zawiera duże przerwy w pracy pojazdów, co sugeruje możliwość przeprowadzenia kilku tur procesu optymalizacji przed ostatecznym zatwierdzeniem przypisania zamówienia do danego pojazdu. Dalsze wyjaśnienie tego zjawiska prezentuje Rysunek 8.5c, przedstawiający uśrednioną względną wielkość zadania obserwowaną w procesie optymalizacji. Na wykresie można zaobserwować, że istnieje przedział czasowy (tuż przed T_{CO}), w którym algorytm optymalizacyjny może zmieniać przypisanie 70-80% zamówień (w zależności od przyjętej konfiguracji serwisu optymalizacyjnego), które są już znane, ale jeszcze nie zatwierdzone. Wykres ten tłumaczy też istotność przekazywania rozwiązań z poprzedniego kroku czasowego. Proces optymalizacyjny mając możliwość modyfikacji przypisania do pojazdu i kolejności obsługi przynajmniej 50% zamówień przez prawie 75% dnia roboczego, może efektywnie wykorzystać informację z poprzedniego kroku czasowego. Te obserwacje potwierdzają słuszność zdefiniowania empirycznego stopnia dynamizmu problemu *em.dod* (Definicja 26), gdyż jego średnia war-

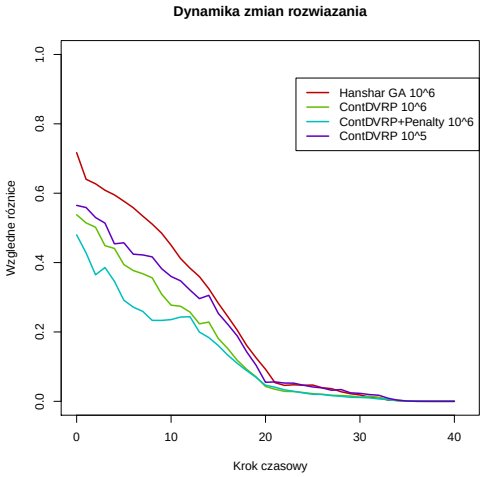
Tabela 8.6: Porównanie wyników ContDVRP z dodaną funkcją kary i bez niej z wynikami MAPSO i MEMSO oparte na całkowitym budżecie 10^6 ewaluacji funkcji jakości. Liczby w nawiasach oznaczają odpowiednio: liczbę przedziałów czasowych, liczbę serwisów optymalizacyjnych oraz liczbę ewaluacji funkcji jakości w pojedynczym kroku czasowym w jednym serwisie optymalizacyjnym.

	$MAPSO$ $(25 * 8 * (0.5 * 10^4))$		$MEMSO$ $(25 * 8 * (0.5 * 10^4))$		$ContDVRP_{PSO}^{k=2}$ $(40 * 8 * (0.31 * 10^4))$		$ContDVRP_{PSO}^{k=2} + P$ $(40 * 8 * (0.31 * 10^4))$	
	Min	Śr.	Min	Śr.	Min	Śr.	Min	Śr.
c50	571.34	610.67	577.60	592.95	**544.11**	**578.31**	551.34	580.56
c75	931.59	965.53	928.53	962.54	**884.43**	903.72	886.42	**901.05**
c100b	866.42	882.39	864.19	878.81	**819.56**	845.80	**819.56**	**844.56**
c100	953.79	973.01	949.83	968.92	902.00	933.46	**873.77**	**920.69**
c120	1223.49	1295.79	1164.63	1284.62	**1053.18**	**1071.38**	1056.70	1116.08
c150	1300.43	1357.71	1274.33	1327.24	1098.03	1134.20	**1097.27**	**1127.04**
c199	1595.97	1646.37	1600.57	1649.17	**1362.65**	**1408.70**	1374.47	1418.71
f71	287.51	296.76	283.43	294.85	274.16	298.50	**270.20**	**275.27**
f134	15150.50	16193.00	14814.10	16083.82	11746.40	11892.00	**11713.20**	**11810.04**
tai75a	1794.38	1849.37	1785.11	1837.00	**1685.23**	1805.03	1691.95	**1782.79**
tai75b	1396.42	1426.67	1398.68	1425.80	1365.36	1422.60	**1356.50**	**1401.38**
tai75c	1483.10	1518.65	1490.32	1532.45	1439.02	1510.00	**1424.91**	**1486.78**
tai75d	1391.99	**1413.83**	**1342.26**	1448.19	1408.79	1433.25	1403.85	1428.91
tai100a	2178.86	2214.61	2170.54	**2213.75**	**2137.30**	2216.23	2147.07	2226.48
tai100b	2140.57	2218.58	2093.54	2190.01	2060.65	2136.80	**2041.96**	**2125.19**
tai100c	1490.40	1550.63	1491.13	1553.55	1458.81	1494.72	**1446.98**	**1485.47**
tai100d	1838.75	1928.69	1732.38	1895.42	1663.87	1727.95	**1658.48**	**1718.18**
tai150a	3273.24	3389.97	**3253.77**	**3369.48**	3338.71	3530.82	3396.49	3526.10
tai150b	**2861.91**	**2956.84**	2865.17	2959.15	2910.06	3026.89	2931.16	3034.65
tai150c	2512.01	2671.35	2510.13	2644.69	**2497.65**	**2603.53**	2523.53	2642.82
tai150d	**2861.46**	**2989.24**	2872.80	3006.88	2869.79	3009.01	2929.91	3023.48

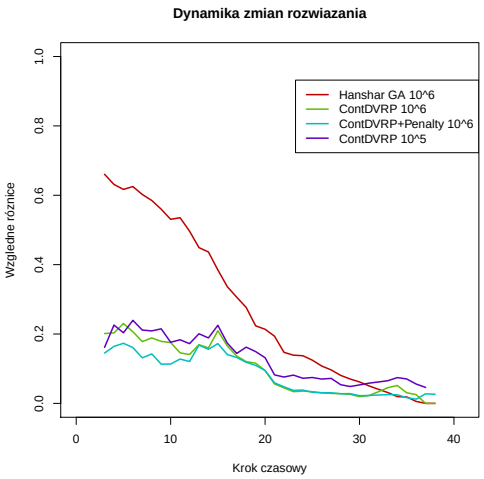
tość (w zakresie 0.2-0.3) jest istotnie mniejsza od średniej wartości stopnia dynamizmu *dod* (Definicja 23), wynoszącej 0.5 dla zbioru instancji testowych.

Dynamizm problemu a zastosowanie funkcji kary

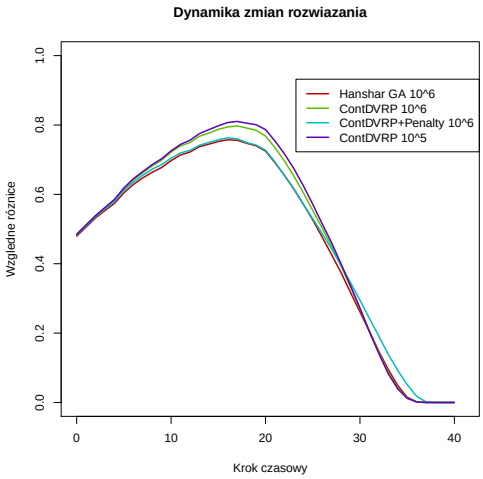
W celu weryfikacji wpływu dodania funkcji kary (wzór (7.6)) do funkcji jakości, na uzyskiwane wyniki oraz poziom ich zmienności w czasie, zostały przeprowadzone obliczenia metodą ContDVRP w konfiguracji z algorytmem PSO, z limitem 10^6 ewaluacji funkcji jakości oraz dodaną funkcją kary. W Tabeli 8.6, porównującej wyniki tego eksperymentu z eksperymentem bazowym oraz MAPSO i MEMSO, można zaobserwować, że został osiągnięty nieznaczny wzrost poprawy średnich rezultatów (o około 0.5%, względem ContDVRP bez funkcji kary). Bardziej istotnych obserwacji można dokonać na Rysunku 8.5e, prezentującym rozkłady względnych wyników. Dla eksperymentu z funkcją kary (ozn. ContDVRP+P 10^6) rozkład ten ma krótszy ogon od rozkładu względnych wyników dla bazowej konfiguracji ContDVRP (ozn. ContDVRP 10^6). Na Rysunku 8.5a można dodatkowo zaobserwować, że algorytm z funkcją kary zwraca bardziej stabilne rozwiązania dla pośrednich stanów problemu niż algorytm bazowy.



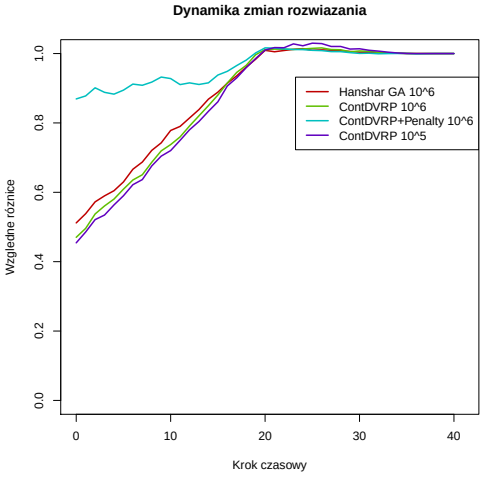
(a) Różnica względem finalnego rozwiązania



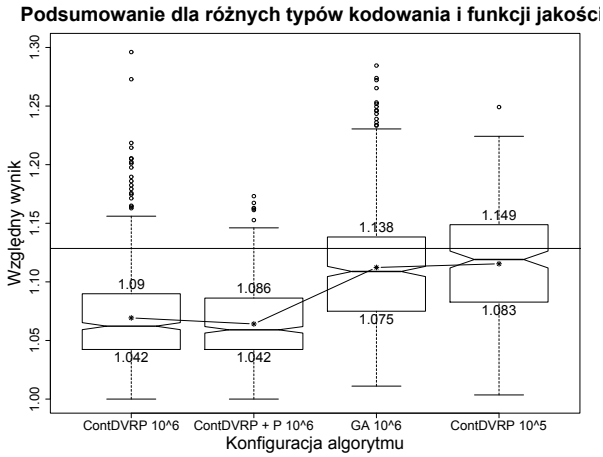
(b) Zmienność rozwiązania



(c) Rozmiar zadania



(d) Liczba wykorzystywanych pojazdów



(e) Jakość uzyskiwanych wyników

Rysunek 8.5: Prezentacja dynamiki problemu dla uśrednionych wyników cząstkowych w każdym z kroków czasowych dla podstawowej wersji algorytmu, wersji z dodaną estymacją finalnie potrzebnej liczby pojazdów, wersji ze zmniejszonym budżetem na obliczenia oraz wersji wykorzystującej kodowanie CIZ i operatory genetyczne z pracy [47].

Ostatnia obserwacja dotyczy skuteczności estymacji liczby pojazdów potrzebnych do obsłużenia finalnego zbioru zamówień (Rysunek 8.5d). Przy zastosowaniu funkcji kary liczba pojazdów wykorzystywanych w rozwiązaniu od początku oscyluje w pobliżu 85-90% finalnego zapotrzebowania. Z operacyjnego punktu widzenia jest to niezwykle istotne, gdyż potencjalnie pozwala na zmniejszenie liczby pojazdów i kierowców utrzymywanych w rezerwie danego dnia, co powinno przełożyć się na znaczny spadek kosztów.

Dynamizm problemu a wybór przestrzeni przeszukiwań

W celu weryfikacji wpływu wyboru przestrzeni przeszukiwań na dynamizm problemu, został zaimplementowany serwis optymalizacyjny realizujący algorytm GA dla DVRP [47]. Na Rysunku 8.5a można zaobserwować różnicę między rozwiązaniem znalezionym przez algorytm w danym przedziale czasowym a finalnym rozwiązaniem optymalnym, natomiast na Rysunku 8.5b różnice między rozwiązaniami wskazywanymi w kolejnych przedziałach czasowych. W obu tych porównaniach zastosowanie algorytmu ContDVRP skutkuje większą stabilnością rozwiązań pośrednich, niż stabilność uzyskana przez algorytm GA wykorzystujący dyskretną przestrzeń przeszukiwań. Zestawiając wyniki konfiguracji ContDVRP 10^6, ContDVRP 10^5 oraz GA 10^6, można zaobserwować, że stabilność pośrednich wyników jest silniej powiązana z wyborem przestrzeni przeszukiwań, niż jakością finalnego wyniku.

8.4.5 Podejście hiperheurystyczne

Pomimo zastosowania szeregu metod poprawiających działanie algorytmu ContDVRP względem pierwotnej pracy [83], takich jak: obliczenia równoległe, bezpośrednie przekazywanie rozwiązań z poprzednich kroków czasowych, estymacja liczby pojazdów na podstawie rozwiązania heurystycznego czy funkcja kary, w zbiorze instancji testowych znajdują się zadania (np. tai75d, tai150a, tai150b), dla których alternatywne podejścia osiągają lepsze średnie rezultaty.

W odpowiedzi na tą obserwację została zaproponowana metoda hiperheurystyczna [87]. Metoda opiera swoje działanie na predykcji względnego wyniku dwóch wybranych algorytmów optymalizacyjnych na instancji problemu o zadanej charakterystyce danych początkowych.

W tym podrozdziale przedstawione są wyniki zastosowania tej metody hiperheurystycznej dla algorytmów M-VRPDR i ContDVRP$_{PSO}$ z budżetem 10^7 ewaluacji funkcji jakości oraz algorytmów MEMSO i ContDVRP$_{PSO}$ z budżetem 10^6 ewaluacji funkcji jakości. Statystyki charakteryzujące instancję DVRP zostały poddane analizie istotności, poprzez wizualną analizę wyników z zastosowaniem *elastic net* oraz ich istotności statystycznej względem jakości predykcji modelu liniowego. W wyniku tej analizy do modelu predykcyjnego zostały wybrane następujące statystyki: μ_y, sd_x, sd_y, sd_s, $skew_s$, $m_{t_{begin}}$.

Tabela 8.7: Wyniki przewidywania algorytmu optymalizacyjnego w walidacji krzyżowej w wariancie *leave-one-out*. W tabeli zaprezentowane są minimalne i średnie wyniki osiągnięte przez M-VRPDR i ContDVRP dla porównywalnych budżetów ewaluacji funkcji jakości. Lepsze wyniki są wytłuszczone. Kolumna *Hiperheurystyka* objaśnia który z algorytmów został wybrany przez model liniowy, czy był to dobry wybór i ile na długości trasy zyskano (stracono) w ten sposób. Zyski i starty dla przypadków o statystycznie istotnych różnicach w wynikach średnich zostały oznaczone szarym kolorem.

	ContDVRP		M-VRPDR [71]		Hiperheurystyka		
Nazwa	Min.	Śr.	Min.	Śr.	Wybór	P/F	Zysk
c50	562.76	573.81	**524.61**	**548.10**	ContDVRP	F	-4.69%
c75	866.34	894.71	**852.95**	**885.00**	M-VRPDR	P	1.10%
c100	886.52	925.47	**860.56**	**913.81**	ContDVRP	F	-1.28%
c100b	**819.56**	**836.57**	820.92	864.03	M-VRPDR	F	-3.28%
c120	**1059.40**	**1083.01**	1189.06	1295.31	ContDVRP	P	19.60%
c150	1083.86	**1113.95**	**1083.79**	1142.24	ContDVRP	P	2.54%
c199	**1341.47**	**1375.49**	1377.26	1466.30	ContDVRP	P	6.60%
f71	272.86	**292.88**	**260.17**	297.61	ContDVRP	P	1.61%
f134	**11735.98**	**11768.98**	11815.95	12299.59	ContDVRP	P	4.51%
tai75a	1680.55	1757.12	**1656.12**	**1705.76**	M-VRPDR	P	3.01%
tai75b	1357.32	1382.52	**1350.28**	**1378.47**	M-VRPDR	P	0.29%
tai75c	1415.18	1477.71	**1355.41**	**1427.74**	M-VRPDR	P	3.50%
tai75d	1398.80	1423.15	**1376.42**	**1414.93**	M-VRPDR	P	0.58%
tai100a	2118.38	2200.03	**2093.63**	**2168.87**	M-VRPDR	P	1.44%
tai100b	2013.74	**2088.83**	**1990.99**	2097.86	M-VRPDR	F	-0.43%
tai100c	1428.99	1466.30	**1421.50**	**1460.72**	ContDVRP	F	-0.38%
tai100d	1633.32	1710.59	**1631.63**	**1710.44**	M-VRPDR	P	0.01%
tai150a	**3212.62**	3370.91	3226.51	**3354.09**	ContDVRP	F	-0.50%
tai150b	**2842.36**	**2954.30**	2847.08	2980.33	ContDVRP	P	0.88%
tai150c	2451.53	**2520.83**	**2427.53**	2558.99	ContDVRP	P	1.51%
tai150d	2795.46	**2882.21**	**2737.37**	2903.68	ContDVRP	P	0.74%

W celu zweryfikowania skuteczności metody hiperheurystycznej została wykonana walidacja krzyżowa w wariancie *leave-one-out*, której wyniki są przedstawione w Tabelach 8.7 i 8.8. W eksperymencie wykorzystującym M-VRPDR i ContDVRP zastosowanie metoda hiperheurystyczna osiągnęła sumaryczną skuteczność o 0.3% większą niż ContDVRP oraz o 1.4% większą niż M-VRPDR, a dla 81% instancji (13 z 16) ze statystycznie różnymi wynikami został wskazany właściwy algorytm. W drugim z eksperymentów, wykorzystującym MEMSO i ContDVRP, hiperheurystyka osiągnęła o 0.2% gorszą sumaryczną skuteczność niż ContDVRP, jednak o 5.5% większą niż MEMSO i wskazała poprawnie 78% instancji (14 z 18) ze statystycznie istotnie różnymi wynikami. Osiągnięte w ten sposób wyniki można podsumować jako zadowalające, zwłaszcza że w eksperymencie wykorzystującym ContDVRP i M-VRPDR występuje równowaga przypadków z dobrymi wynikami obu algorytmów, co uzasadnia stosowanie podejścia hiperheurystycznego.

Tabela 8.8: Wyniki przewidywania algorytmu optymalizacyjnego w walidacji krzyżowej w wariancie *leave-one-out*. W tabeli zaprezentowane są minimalne i średnie wyniki osiągnięte przez MEMSO i ContDVRP dla porównywalnych budżetów ewaluacji funkcji jakości. Lepsze wyniki są wytłuszczone. Kolumna *Hiperheurystyka* objaśnia który z algorytmów został wybrany przez model liniowy, czy był to dobry wybór i ile na długości trasy zyskano (stracono) w ten sposób. Zyski i starty dla przypadków o statystycznie istotnych różnicach w wynikach średnich zostały oznaczone szarym kolorem.

	ContDVRP		MEMSO [58]		Hiperheurystyka		
Nazwa	Min.	Śr.	Min.	Śr.	Wybór	P/F	Zysk
c50D	**551.34**	**580.56**	577.60	592.95	ContDVRP	P	2.13%
c75D	**886.42**	**901.05**	928.53	962.54	ContDVRP	P	6.82%
c100D	**873.77**	**920.69**	949.83	968.92	ContDVRP	P	5.24%
c100bD	**819.56**	**844.56**	864.19	878.81	MEMSO	F	-4.06%
c120D	**1056.70**	**1116.08**	1164.63	1284.62	ContDVRP	P	15.10%
c150D	**1097.27**	**1127.04**	1274.33	1327.24	ContDVRP	P	17.76%
c199D	**1374.47**	**1418.71**	1600.57	1649.17	ContDVRP	P	16.24%
f71D	**270.20**	**275.27**	283.43	294.85	ContDVRP	P	7.11%
f134D	**11713.20**	**11808.53**	14814.10	16083.82	ContDVRP	P	36.21%
tai75aD	**1691.95**	**1782.79**	1785.11	1837.00	ContDVRP	P	3.04%
tai75bD	**1356.50**	**1401.38**	1398.68	1425.80	ContDVRP	P	1.74%
tai75cD	**1424.91**	**1486.78**	1490.32	1532.45	MEMSO	F	-3.07%
tai75dD	1403.85	**1428.91**	**1342.26**	1448.19	ContDVRP	P	1.35%
tai100aD	**2147.07**	2226.48	2170.54	**2213.75**	MEMSO	P	0.57%
tai100bD	**2041.96**	**2125.19**	2093.54	2190.01	ContDVRP	P	3.05%
tai100cD	**1446.98**	**1485.47**	1491.13	1553.55	ContDVRP	P	4.58%
tai100dD	**1658.48**	**1718.18**	1732.38	1895.42	ContDVRP	P	10.32%
tai150aD	3396.49	3526.10	**3253.77**	**3369.48**	ContDVRP	F	-4.65%
tai150bD	2931.16	3034.65	**2865.17**	**2959.15**	ContDVRP	F	-2.55%
tai150cD	2523.53	**2642.63**	**2510.13**	2644.69	ContDVRP	P	0.08%
tai150dD	2929.91	3023.41	**2872.80**	**3006.88**	MEMSO	P	0.55%

Natomiast w eksperymencie wykorzystującym ContDVRP i MEMSO jest prawdopodobnie zbyt mała liczba instancji, dla których MEMSO osiąga istotnie lepsze wyniki, co jest widoczne w Tabeli 8.3.

Rozdział 9

Podsumowanie

W rozprawie zaprezentowano algorytm ContDVRP[1] osadzony w środowisku równoległych serwisów optymalizacyjnych. W przeciwieństwie do większości prac w obszarze dynamicznej marszrutyzacji, ContDVRP wykorzystuje ciągłe kodowanie w celu bezpośredniego użycia algorytmów metaheurystycznych, takich jak PSO czy DE. Algorytm ContDVRP, początkowo rozwijany niezależnie od ciągłych podejść obecnych w statycznych problemach marszrutyzacji, w finalnej wersji przedstawionej w rozprawie, pozwala na łączenie autorskiego kodowania ciągłego [84] ze zbliżonym podejściem prezentowanym w literaturze problemów statycznych [1]. Podejście zastosowane w ContDVRP nie tylko pozwala zastosować populacyjne algorytmy metaheurystyczne, co jest zaprezentowane w rozprawie na przykładzie PSO i DE, ale również elastycznie dobrać kodowanie w zależności od tego, czy ważniejsza jest jakość wyników czy czas optymalizacji.

Tabela 9.1: Podsumowanie zagregowanych rezultatów uzyskanych przez ContDVRP dla każdego z eksperymentów. Porównanie prezentuje: średni procentowy zysk w długości trasy względem najlepszego podejścia literaturowego, liczbę statystycznie istotnie lepszych średnich wyników, liczbę najlepszych wyników w danej konfiguracji kryterium stopu.

Eksperyment	Algorytm	ContDVRP_{PSO}			ContDVRP_{DE}		
	literaturowy	Zysk	Średnie	Najlepsze	Zysk	Średnie	Najlepsze
10^6 ewaluacji	MEMSO	5.7%	13	10	5.5%	15	8
10^7 ewaluacji	M-VRPDR	1.2%	8	6	-0.2%	5	1
Czas	GA	7.8%	16	13	7.9%	15	4

Finalna wersja algorytmu ContDVRP zaprezentowana w rozprawie zdecydowanie poprawia wyniki literaturowe względem stanu wiedzy przy rozpoczęciu badań. Pozostaje również metodą konkurencyjną wobec najnowszego podejścia wykorzystującego algorytm memetyczny [71]. W oparciu o podsumowanie wyników numerycznych zawarte w Tabeli 9.1, można stwierdzić, że ContDVRP jest skuteczniejszy niż najlepsze podejścia literaturowe w każdym z 3 układów eksperymentów: ograniczonym czasowo, z całkowitym

[1] Kod źródłowy i aplikacja dostępne w repozytorium: `https://sourceforge.net/p/continuous-dvrp/`

budżetem 10^6 ewaluacji funkcji jakości oraz z całkowitym budżetem 10^7 ewaluacji funkcji jakości. Średnie trasy uzyskane przez ContDVRP są krótsze względem najlepszych algorytmów w danym eksperymencie odpowiednio o 7.8% (GA [47]), 5.7% (MEMSO [58]) i 1.2% (M–VRPDR [71]). Można stąd wyciągnąć wniosek, że **potwierdzona została prawdziwość pierwszej hipotezy badawczej, dotyczącej możliwości skutecznego zastosowanie ciągłych algorytmów optymalizacji globalnej do problemu dynamicznej marszrutyzacji.**

Kodowanie przypisania zamówień do pojazdów przyjęte w rozprawie pozwala na kontrolę liczby pojazdów wykorzystywanych w rozwiązaniu. Przyjęcie założenia o proporcjonalnym przyroście sumy rozmiarów zamówień w trakcie pierwszej połowy dnia (kiedy akceptowane są nowe zamówienia), pozwoliło na znajdowanie rozwiązań, które od początku procesu optymalizacji uwzględniały średnio 90% liczby pojazdów, które okazywały się konieczne w finalnych rozwiązaniach, co można zaobserwować na Rysunku 8.5d. Przewidywanie liczby finalnie potrzebnych pojazdów, jest wykorzystane w konstrukcji funkcji kary dodawanej do funkcji jakości (wzór 7.6). Poprzez wymuszanie, przy pomocy tej funkcji kary, wykorzystania większej liczby pojazdów, osiągnięto dodatkowe zwiększenie stabilności rozwiązań cząstkowych względem rozwiązania finalnego, co można zaobserwować na Rysunkach 8.5b i 8.5a. Takie zwiększenie stabilności ma szczególne znaczenie praktyczne, gdyż oznacza możliwość przekazywania kierowcom mniejszej liczby informacji przez dyspozytora.

Zastosowanie funkcji kary w algorytmie ContDVRP (w konfiguracji z budżetem 10^6 ewaluacji funkcji jakości, optymalizacją PSO i kodowaniem WCZ z 2 skupiskami zamówień na pojazd) przyniosło poprawę sumarycznej skuteczności o 0.5% względem identycznej konfiguracji bazowej. Natomiast liczba instancji testowych, dla których ContDVRP osiągnął statystycznie lepszy wynik niż MEMSO wzrosła o dwie. Na rozkładzie względnych wyników towarzyszącym Tabeli 8.6 można zaobserwować, że poprawa wyników była spowodowana zmniejszeniem prawego ogona tego rozkładu. Przed podsumowaniem warto jeszcze zauważyć na Rysunku 8.5c, że pomimo początkowej wiedzy o jedynie około 50% zamówień, tuż przed czasem odcięcia w puli zamówień oczekujących znajduje się średnio aż 70%–80% wszystkich zamówień problemu. Wynika stąd, że dynamizm problemu jest w praktyce mniejszy niż wynikałoby to z samego faktu przyjęcia czasu odcięcia w połowie dnia roboczego. Jednocześnie nawet przy takim stopniu dynamizmu problemu można zaobserwować zysk w jakości i stabilności rozwiązań. Można zatem skonkludować, że **dodanie predykcji finalnego wyniku, nawet przy stosunkowo małym dynamizmie problemu, pozwala poprawić wyniki oraz możliwość zastosowania ich w praktyce, co potwierdza drugą hipotezę badawczą.**

Zgodnie z wnioskami z twierdzenia *No Free Lunch* [131], niezależnie od wszelkich technik poprawiających działanie konkretnego algorytmu, istnieją instancje problemu, które będą dla tego algorytmu trudne w optymalizacji. Identyfikując charakterystykę instancji

można próbować dopasować właściwy dla niej algorytm. W rozprawie zaproponowano zestaw statystyk opisujących charakterystykę problemu marszrutyzacji. Na podstawie tego zestawu statystyk, obliczanych dla danych znanych na początku problemu, budowane są liniowe modele predykcyjne. Modele te estymują względną jakość rozwiązań, które zostaną osiągnięte przez zadane algorytmy. Osadzenie predykcji i trenowania tych modeli w procesie optymalizacyjnym, wykorzystującym pulę dwóch algorytmów rozwiązujących problem dynamicznej marszrutyzacji tworzy propozycję podejścia hiperheurystycznego do DVRP. Przykład tego podejścia, wykorzystującego algorytmy MEMSO i 2MPSO[2], jest zobrazowany na Rysunku 7.5.

Podejście hiperheurystyczne pozwala na połączenie różnych algorytmów i korzystanie z silnych stron każdego z nich. Działanie hiperheurystyki jest zależne od zrównoważenia wyników algorytmów optymalizacyjnych na zbiorze instancji testowych. 2MPSO i MEMSO osiągnęły odpowiednio 9 i 8 statystycznie lepszych średnich wyników [87]. Wykorzystująca je hiperheurystyka poprawia uzyskane średnie wyniki o 0.6% względem 2MPSO, poprawnie wskazując algorytm dla 14 spośród 17 badanych instancji. Po udoskonaleniu algorytmu 2MPSO do ContDVRP, stosunek liczby najlepszych średnich wyników zwiększył się istotnie na korzyść ContDVRP. ContDVRP i MEMSO osiągnęły odpowiednio 16 i 2 statystycznie lepszych średnich wyników. Nie dziwi więc, że chociaż metoda wskazała poprawnie 14 spośród tych 18 instancji, to wyniki uległy pogorszeniu o 0.2% względem ContDVRP. W celu dodatkowej weryfikacji wyników eksperyment został powtórzony dla ContDVRP i M-VRPDR, które uzyskały odpowiednio 9 i 7 statystycznie lepszych średnich wyników. Zastosowanie hiperheurystyki poprawiło średnie wyniki o 0.3% względem ContDVRP, a predykcyjny model liniowy poprawnie wskazuje 13 spośród 16 statystycznie lepszych instancji. Można postawić tezę, że metoda hiperheurystyczna ma szansę działać lepiej dla problemów statycznych, gdzie wyeliminowane jest dodatkowe ryzyko niespójności charakterystyk początkowych i finalnych danych problemu. Rozważania na temat metody hieprheurystycznej można podsumować stwierdzeniem, że **hipoteza badawcza o możliwości poprawy wyników przy wykorzystaniu puli algorytmów o różnej jakości działania zależnej od cech instancji jest prawdziwa, przy przyjęciu dodatkowego założenia o zrównoważonych wynikach tych algorytmów.**

9.1 Dyskusja w dziedzinie problemów marszrutyzacji

Oprócz eksperymentalnego potwierdzenia hipotez badawczych, z badań przeprowadzonych w rozprawie można wyciągnąć szereg wniosków dotyczących optymalizacji problemów marszrutyzacji, zarówno dynamicznych jak i statycznych.

[2]Wcześniejsza wersja ContDVRP, wykorzystująca transfer wyników poprzez fazę dyskretyzacji oraz algorytm PSO, zarówno w pierwszej, jak i drugiej fazie optymalizacji.

9.1.1 Wnioski dla dowolnych problemów marszrutyzacji

Do wniosków, które można uogólnić na całą grupę problemów marszrutyzacji należą:

- możliwość wykorzystania ciągłego kodowania, wraz z możliwością jego modyfikacji względem stosowanych niezależnie podejść [2, 84] poprzez jednoczesne zastosowanie kodowania kolejności i kodowania wielu skupisk na pojazd w jednym wektorze,
- możliwość poprawy jakości wyników uzyskiwanych algorytmami metaheurystycznymi przy wykorzystaniu zmodyfikowanego algorytmu Kruskala do inicjalizacji części rozwiązań w populacji początkowej,
- przeznaczanie budżetu obliczeniowego w pierwszej kolejności na optymalizację podziału zamówień pomiędzy pojazdy, a w drugiej kolejności na optymalizację poszczególnych tras,
- możliwość wyboru algorytmu optymalizacyjnego w oparciu o cechy zadania do zoptymalizowania, co dla zadań statycznych może przynieść zysk w wyniku zmniejszenia kosztów obliczeniowych,
- możliwość osiągnięcia jednoczesnego zysku jakości wyników i czasu obliczeń, przy wykorzystaniu przetwarzania przez niezależne procesy optymalizacyjne (stosunek jakości do czasu obliczeń w zależności od liczby procesów optymalizacyjnych obrazują wyniki z Tabeli A.2).

Uproszczona wersja algorytmu ContDVRP została z powodzeniem zastosowana do optymalizacji statycznego problemu marszrutyzacji [69]. Należy jednocześnie zauważyć, że skuteczne zastosowanie w innych odmianach problemu marszrutyzacji może wymagać stosownych modyfikacji. Przykładowo dla problemu marszrutyzacji z oknami czasowymi konieczne może być: dodanie czasu jako trzeciego wymiaru do analizy skupień, czy zapewnienie wzajemnego wykluczania się zamówień, które nie mogą być obsłużone przez ten sam pojazd (ze względu na wąskie i pokrywające się okna czasowe), już na etapie łączenia zamówień w skupiska.

Wnioski z analizy zależności jakości wyniku od zastosowanego kodowania, zaprezentowanej na Rysunku A.4, nie powinny być bezpośrednio przenoszone na problemy statyczne. Dla problemu dynamicznego nawet kodowanie z jednym tylko skupiskiem zamówień na pojazd może skutkować przecinającymi się trasami w finalnym rozwiązaniu, ze względu na zatwierdzanie kolejnych zamówień w trakcie procesu optymalizacji. W związku z tym dla problemów statycznych należy zweryfikować zależność skuteczności algorytmu od liczby skupisk zamówień.

Na koniec warto zauważyć dwa aspekty proponowanego ciągłego kodowania przypisania zamówień do pojazdów. Pierwszy z nich ma wymiar praktyczny i wiąże się z geometryczną interpretacją centroidów skupisk zamówień, jako punktów definiujących podział terenu (płaszczyzny) na obszary, których obsługa jest przypisywana konkretnemu pojazdowi. Podział terenu na predefiniowane obszary jest też standardową praktyką dyspozy-

tora, wykorzystywaną przy planowaniu transportu. Zatem algorytm, oparty na kodowaniu centroidów skupisk zamówień, zwraca rozwiązania, które mogą być wyrażone w sposób przyjazny dla dyspozytora. Drugi aspekt jest powiązany z pierwszym, niewielkie zmiany wektora centroidów skutkują małymi zmianami rozwiązania, z punktu widzenia dyspozytora i kierowców. Wynika to z faktu, że niewielkie zmiany wektora centroidów mogą zmienić przypisanie zamówień jedynie do pojazdów obsługujących sąsiadujące obszary.

9.1.2 Wnioski dla dynamicznego problemu marszrutyzacji

Poza wnioskami dotyczącymi wielu wariantów problemów marszrutyzacji, z analizy wyników algorytmu można wyciągnąć dodatkowe wnioski mające zastosowanie jedynie dla problemu dynamicznej marszrutyzacji:

- przyjęty w początkowych pracach czas odcięcia $T_{CO} = 0.5$ prowadzi do mniejszego faktycznego średniego dynamizmu problemu (niedostępność 20%-30% zamówień przy podejmowaniu decyzji), niż spodziewana intuicyjnie średnia wartość 50% zamówień,
- proponowane w rozprawie kodowanie zwiększa stabilność rozwiązań cząstkowych wskazywanych w trakcie procesu optymalizacji.

Dynamika DVRP była poddawana wcześniej jedynie analizie typowej dla standardowych problemów dynamicznych w pracy [57]. Analiza ta koncentrowała się na szybkości zbieżności algorytmu po każdej ze zmian oraz jakości znajdowanego optimum względem optimum dla pośredniego stanu problemu. Tymczasem, z punktu widzenia celu optymalizacji (zgodnego z Definicją 17), jakość optimum dla problemów pośrednich wydaje się mieć istotne znaczenie tylko dlatego, że dla $T_{CO} = 0.5$ istnieje taki przedział czasowy, w którym jest dostępna przeważająca część zamówień (co można zaobserwować na Rysunku 8.5c). W przeciwnym przypadku powiększanie budżetu ewaluacji funkcji jakości nie powinno prowadzić do poprawy wyników, obserwowanego na Rysunku A.5. Powiększanie budżetu liczby ewaluacji, bez uwzględniania przyszłych zamówień, będzie skutkowało zwiększeniem różnic między rozwiązaniami optymalnymi dla stanów pośrednich, a podrozwiązaniem będącym fragmentem rozwiązania finalnego, jak jest to zaprezentowane na Rysunku 8.4. Pomimo różnych sposobów obserwacji dynamizmu wnioski z rozprawy i przytoczonej pracy [57] są zbieżne, wskazując na konieczność utworzenia nowych instancji testowych. Przytoczona praca dostarcza wyników dla problemów o większych wartościach T_{CO}, czyniąc je bardziej dynamicznymi. Niestety te wyniki referencyjne są zaprezentowane na zbiorze instancji, dla którego jest dostarczony jedynie ich niedeterministyczny generator[3], co ogranicza możliwość bezpośredniego porównania.

Oprócz wymienionej we wnioskach dla problemów marszrutyzacji cechy kodowania ciągłego zaprezentowanego w rozprawie, zbliżającej to kodowanie do sposobu pracy dys-

[3] `http://paradiseo.gforge.inria.fr/index.php?n=Benchmarks.VRPgenerator`

pozytorów, zwiększa ono również stabilność rozwiązania, nawet bez uwzględniania przyszłych zamówień. Jak wynika z zestawienia stabilności (Rysunki 8.5a i 8.5b) z rozkładem względnym wyników (Rysunek 8.5e), stabilność rozwiązania powiązana jest z trzema czynnikami: jakością wyników, uwzględnieniem przyszłych zamówień oraz wyborem kodowania. Najbardziej stabilne wyniki zostały uzyskane przez bazową konfigurację ContDVRP z dodaną funkcją kary, a w drugiej kolejności dla identycznej konfiguracji ContDVRP bez funkcji kary. Następne pod względem stabilności były wyniki eksperymentu z ContDVRP ze zmniejszonym budżetem obliczeniowym, które są zestawione ze zbliżonymi jakościowo, ale mniej stabilnymi wynikami uzyskanymi algorytmem GA (zgodnym z propozycją z pracy [47]) uruchomionym w środowisku równoległych procesów optymalizacyjnych. Należy też zwrócić uwagę, że przy użyciu ciągłego kodowania przypisania zamówień do pojazdów znalezienie finalnego rozwiązania, które pozostanie niezmienione przez cały proces optymalizacji możliwe jest już w pierwszym kroku czasowym. Warunkami takiej możliwości są: prawidłowa estymacja liczby pojazdów oraz istnienie zamówień reprezentujących każde ze skupień z rozwiązania finalnego. Z tej możliwości wynika zmiana w klasyfikacji problemu dynamicznej marszrutyzacji: z problemu o skokowej zmienności w przestrzeni dyskretnej (Rysunek 1.1) do połączenia zmienności skokowej z występowaniem nieistotnych dynamicznie zmian w przestrzeni ciągłej (Rysunek 1.2).

9.2 Dyskusja w dziedzinie optymalizacji dynamicznej

Wnioski płynące z wyników rozprawy można uogólnić na dziedzinę optymalizacji problemów dynamicznych. Traktując problem dynamicznej marszrutyzacji jako studium przypadku dla zagadnienia optymalizacji problemów dynamicznych, można uzupełnić listę technik wspomagających populacyjne metody optymalizacyjne w problemach dynamicznych o:

- włączanie do populacji, po każdej zmianie stanu problemu, rozwiązań wyszukiwanych lub budowanych stosunkowo niewielkim kosztem obliczeniowym zapewniających dolne ograniczenie jakości rozwiązania,
- wykorzystanie hiperheurystyki, bazującej na identyfikacji charakterystyki problemu, do wyboru przestrzeni przeszukiwań i operatorów algorytmu o największej skuteczności dla danego typu problemów,
- włączenie do oceny rozwiązań funkcji kary opartej o charakterystykę zmian zachodzących w problemie.

W problemie dynamicznej marszrutyzacji pomocne okazały się też następujące techniki spośród wymienionych w Rozdziale 2:

- zastosowanie przestrzeni zmniejszającej poziom dynamizmu,
- przekazywanie historycznych rozwiązań,
- restartowanie algorytmu po każdej zmianie stanu problemu,

- obliczenia wykorzystujące wiele niezależnych populacji.

9.3 Dalsze kierunki badań

Wprowadzenie ciągłego algorytmu ContDVRP wraz z funkcją kary oraz metodą hiperheurystyczną otwiera nowe pytania dotyczące optymalizacji problemu dynamicznej marszrutyzacji. Dalsze badania nad algorytmem ContDVRP mogą obejmować następujące obszary:

- analizę dynamizmu problemu i jakości wyników uzyskiwanych z zastosowaniem i bez zastosowania funkcji kary dla większych wartości czasu odcięcia T_{CO},
- uwzględnienie w funkcji kary, wymuszającej wykorzystanie estymowanej liczby pojazdów, aktualnego rozkładu zamówień w przestrzeni,
- wykorzystanie w modelu predykcyjnym metody hipeheurystycznej innych statystyk opisujących charakterystykę rozkładu zamówień (z uwzględnieniem odporności na obroty układu odniesienia, czy położenie zajezdni względem zamówień),
- wprowadzenie możliwości wyboru algorytmu przez hiperheurystykę w każdym kroku czasowym,
- wprowadzenie efektywnych metod lokalnej optymalizacji stosowanych w procesie optymalizacji po czasie odcięcia T_{CO}.

Dodatek A

Dobór parametrów

W tym dodatku zaprezentowane są wyniki eksperymentów, w których dobierane były wielkości parametrów i weryfikowana była wrażliwość wyników na ich zmianę.

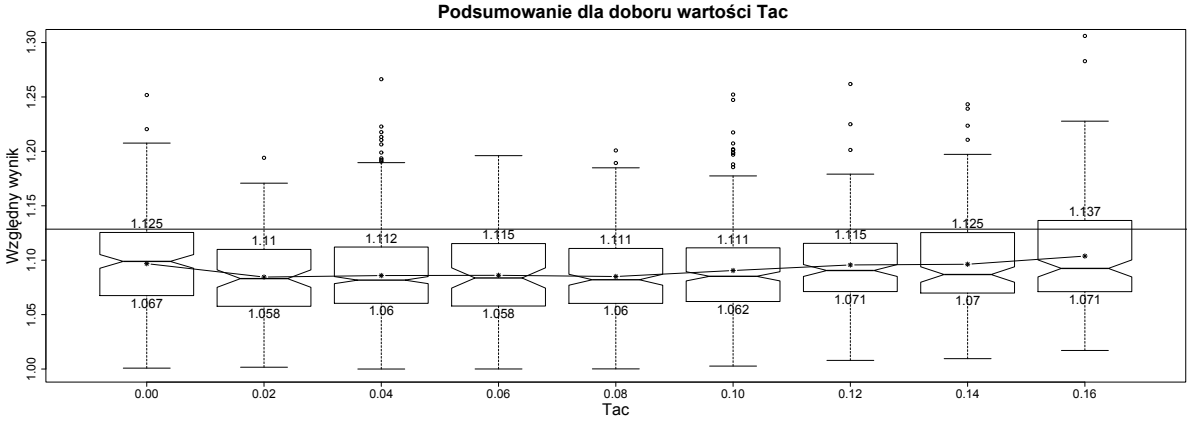
Wyjściową konfiguracją algorytmu dla dostrojenia parametrów była wersja algorytmu zaprezentowana w artykule [84]. Wyjściowym eksperymentem był algorytm ContDVRP wykorzystujący PSO z 8 równoległymi serwisami optymalizacyjnymi, 25 krokami czasowymi, 40 cząstkami wykonującymi 125 iteracji, 3 klastrami zamówień na pojazd oraz przekazywaniem informacji pomiędzy krokami czasowymi poprzez fazę dyskretyzacji.

We wstępnym eksperymencie liczba kroków czasowych została zmieniona z 25 na 50. Później parametry były strojone sekwencyjnie: w pierwszej kolejności stosunek liczby cząstek do liczby iteracji w algorytmie PSO, w następnej kolejności wielkość buforu T_{AC}, następnie liczba kroków czasowych N_{TS} i na koniec stosunek liczby osobników do liczby iteracji w algorytmie DE.

Następnie dla tak dobranych parametrów badana była zależność wyników od sposobu transferu informacji między kolejnymi krokami czasowymi (omawianego w Rozdziale 6.2.1), wybranego kodowania, liczby skupisk zamówień przypisanych do pojazdu, wybranego algorytmu optymalizacyjnego i liczby równoległych procesów optymalizacyjnych.

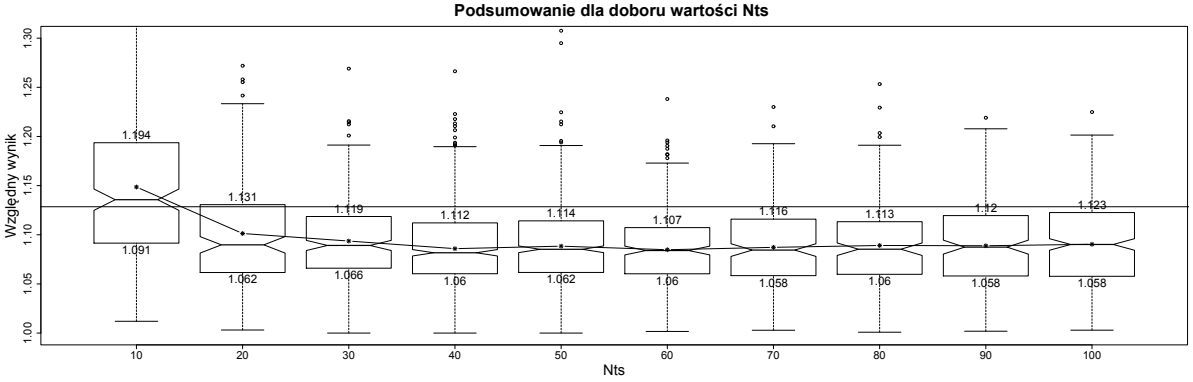
A.1 Dobór bufora czasowego

Dobór bufora czasowego T_{AC} został wykonany spośród zbioru następujących wartości $\{0\%,\ 2\%, \ldots, 16\%\}$ względem długości dnia roboczego $t_{end} - t_{start}$. Dla każdej z tych wartości zostało przeprowadzonych kilkadziesiąt testów dla zadań ze zbioru testowego. Jako finalna wartość został wybrany $T_{AC} = 4\%$ jako najmniejsza wartość dająca średnio najlepsze wyniki. Rozkład wyników dla poszczególnych wartości jest zaprezentowany na Rysunku A.1.



Rysunek A.1: Względne wyniki $ContDVRP_{PSO}$ dla różnych wartości procentowego T_{AC}.

A.2 Dobór liczby kroków czasowych

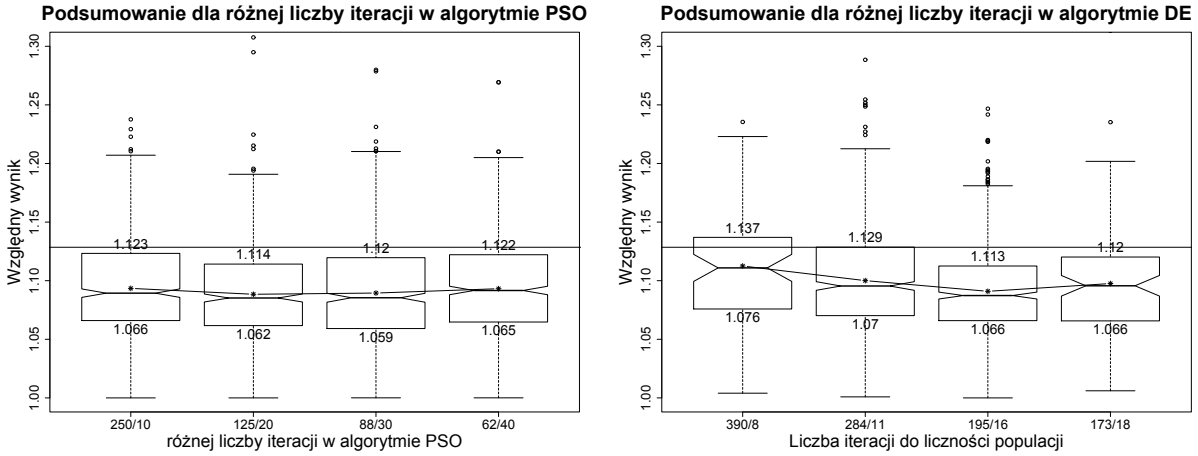


Rysunek A.2: Względne wyniki $ContDVRP_{PSO}$ dla różnych wartości N_{TS} i odpowiednio zmienianego budżetu na ewaluacje funkcji jakości w pojedynczym kroku czasowym.

Dobór liczby kroków czasowych N_{TS} został wykonany spośród zbioru następujących wartości $\{10, 20, \ldots, 100\}$. Dla każdej z tych wartości zostało przeprowadzonych kilkadziesiąt testów dla zadań ze zbioru testowego. Jako finalna wartość został wybrany $N_{TS} = 40$ jako najmniejsza wartość dająca średnio najlepsze wyniki. Poszukiwanie najmniejszej dobrej wartości wynikało z faktu, że oznacza to więcej czasu na optymalizację w poszczególnych oknach czasowych przy stałym budżecie na całość dnia oraz rzadsze aktualizacje tras pojazdów (co jest pożądane z punktu wiedzenia operacyjnego, jako łatwiejsze do wykonania). Rozkład wyników dla poszczególnych wartości jest zaprezentowany na Rysunku A.2.

A.3 Dobór liczby iteracji w stosunku do liczności populacji

Eksperymenty związane z doborem stosunku liczby iteracji do liczności populacji, przy ustalonym stałym budżecie na obliczenia, zostały przedstawione na Rysunku A.3.



(a) Wyniki dla PSO

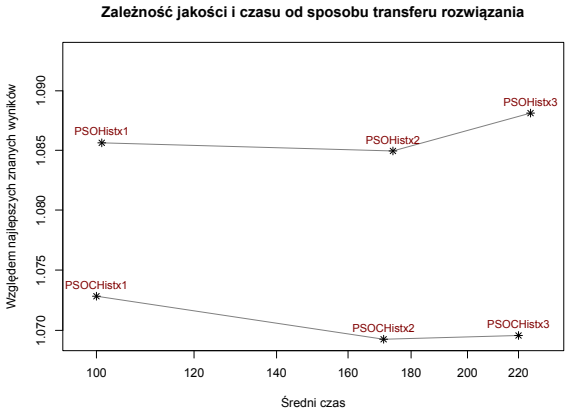
(b) Wyniki dla DE

Rysunek A.3: Prezentacja rozkładów względnych wyników dla różnych algorytmów optymalizacyjnych (PSO i DE) oraz różnego stosunku liczby iteracji do liczności populacji.

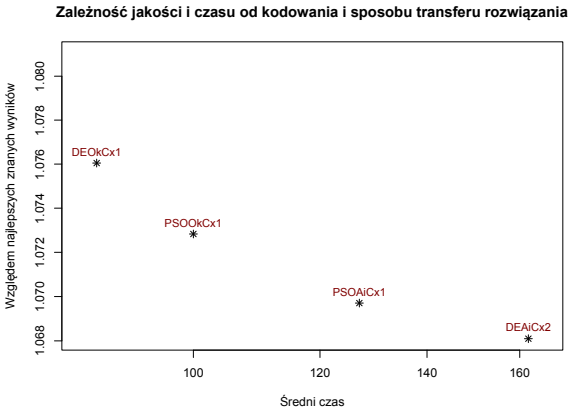
Algorytm PSO okazał się znacznie mniej wrażliwy na wartość tego parametru od algorytmu DE, w związku z czym we wszystkich eksperymentach stosunek liczby iteracji do liczności populacji jest ustawiony na około 125 : 20. Dla algorytmu DE dobór tego parametru przebiegał dwuetapowo, w drugim etapie za wartość optymalną z punktu widzenia zbioru instancji testowych został przyjęty stosunek 195 : 16. Przy eksperymentach ze zwiększonym budżetem ewaluacji funkcji jakości zachowany był stosunek liczby roju do wielkości populacji.

A.4 Dobór sposobu transferu informacji i liczby klastrów zamówień przypadających na pojazd

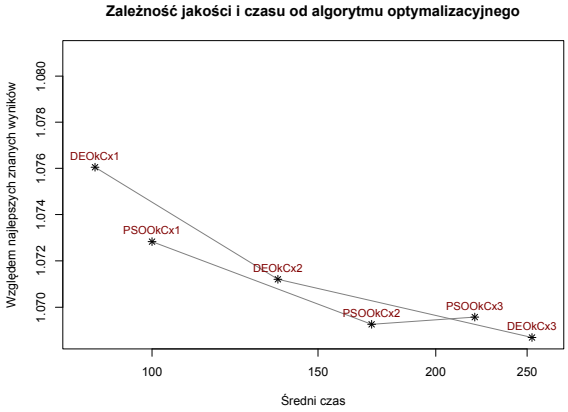
W celu doboru kodowania dającego najlepsze wyniki na danym zbiorze testowym zostało przetestowanych 15 konfiguracji kodowania z różną liczbą k klastrów przypadających na pojazd, różnymi ciągłymi kodowaniami (z uwzględnieniem kolejności zamówień [1,2] i bez niej [83, 84]) oraz różnym sposobem transferu informacji pomiędzy krokami czasowymi. Konfiguracje są przedstawione w formacie [*Algorytm*][*Kodowanie*][*Transfer*]x[*Liczba klastrów*], gdzie algorytm może przyjmować wartości *PSO* lub *DE*, kodowanie wartości *Ok* lub *Ai*, transfer *D* lub *C*, a liczba klastrów od 1 do 3. Kodowanie *Ai* oznacza kodowanie rankingu zamówień i centroidów skupisk zamówień w jednym wektorze [1, 2], zaś *Ok* wyłącznie kodowanie centroidów zamówień [83, 84]. Transfer *D* oznacza przetwarzanie rozwiązań poprzez fazę dyskretyzacji a następnie uciąglenie, natomiast *C* bezpośrednie przenoszenie (z ewentualną adaptacją do nowego stanu problemu).



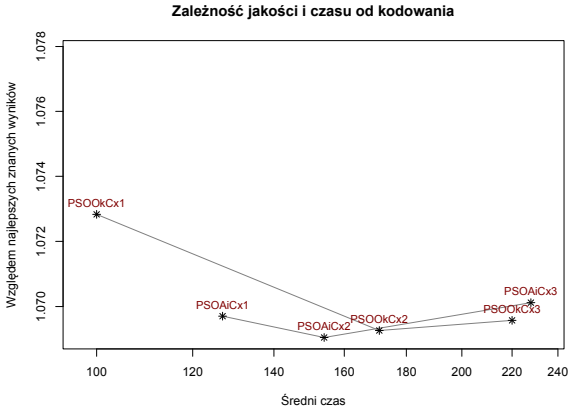
(a) Porównanie średnich wyników i średniego czasu obliczeń dla różnych konfiguracji liczby klastrów i sposobu transferu informacji.



(b) Porównanie średnich wyników i średniego czasu obliczeń dla konfiguracji należących do frontu Pareto.



(c) Porównanie średnich wyników i średniego czasu obliczeń dla różnych konfiguracji liczby klastrów i algorytmu optymalizacyjnego.



(d) Porównanie średnich wyników i średniego czasu obliczeń dla różnych konfiguracji liczby klastrów i sposobu kodowania rozwiązania.

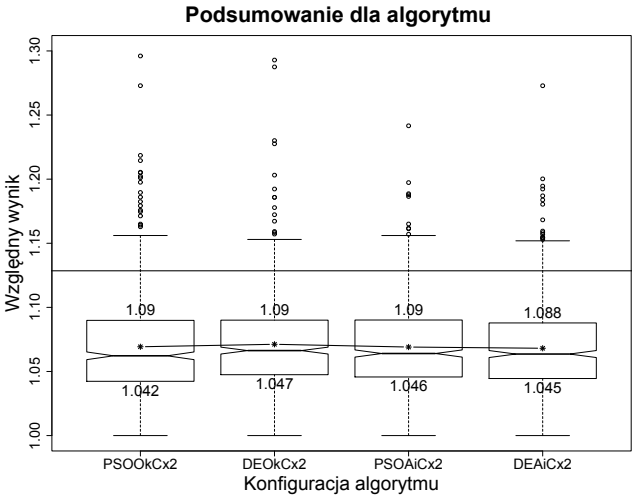
Rysunek A.4: Porównanie zależności średniego czasu obliczeń oraz średniej jakości uzyskanej na zbiorze instancji testowych dla wybranych konfiguracji algorytmu optymalizacyjnego, kodowania i sposobu transferu informacji.

Najistotniejszą poprawę wyników przyniosła zamiana sposobu transferu informacji na bezpośredni, czyli przenoszenie centroidów skupisk zamówień z poprzedniego kroku czasowego, uzupełnione o losowy centroid, jeżeli konieczne było wprowadzenie nowego pojazdu. Porównanie dla algorytmu PSO pomiędzy tymi dwoma sposobami transferu informacji jest zaprezentowane na Rysunku A.4a.

Do frontu Pareto konfiguracji (Rysunek A.4b), dla kryteriów średniej jakości wyników i średniego czasu obliczeń, należą konfiguracje z bezpośrednim przenoszeniem informacji z poprzedniego kroku czasowego oraz z jednym skupiskiem zamówień na pojazd. Ponadto do frontu Pareto należy konfiguracja ze zmodyfikowanym kodowaniem Ai i Kachitvichyanu-

Tabela A.1: Porównanie średnich wyników uzyskanych przez podejścia z 2. skupiskami zamówień na pojazd, bezpośrednim transferem informacji pomiędzy krokami czasowymi i różnymi algorytmami optymalizacyjnymi i kodowaniem.

	$ContDVRP$ (PSO, Ok)	$ContDVRP$ (DE, Ok)	$ContDVRP$ (PSO, Ai)	$ContDVRP$ (DE, Ai)
	Śr.	Śr.	Śr.	Śr.
c50	**578.31**	580.02	583.49	581.24
c75	903.72	908.28	**901.63**	904.11
c100	933.46	937.07	**932.76**	936.92
c100b	845.8	843.83	842.85	**826.82**
c120	**1071.38**	1104.61	1082.02	1079.75
c150	**1134.2**	1140.14	1145.89	1147.87
c199	1408.7	1414.85	1412.28	**1407.8**
f71	298.5	295.58	292.92	**289.48**
f134	**11892**	11916.7	11969.68	12006
tai75a	**1805.03**	1807.49	1817.41	1812.93
tai75b	1422.6	1412.85	**1411.7**	1419.86
tai75c	1510	**1501.64**	1501.91	1523.06
tai75d	**1433.25**	1451.04	1442.31	1444.14
tai100a	**2216.23**	2247.21	2251.18	2229.71
tai100b	2136.8	2145.85	**2126.98**	2157.35
tai100c	1494.72	1495.87	**1494.31**	1502.06
tai100d	**1727.95**	1736.33	1742.27	1729.63
tai150a	3530.82	3489.65	3528.1	**3453.73**
tai150b	3026.89	3038.63	**3014.45**	3023.5
tai150c	2603.53	2563.01	2574.42	**2562.45**
tai150d	3009.01	2999.43	**2955.65**	2987.38
sum	**44982.9**	45030.08	45024.21	45025.79



kula [2], poszerzonym o proponowane w rozprawie kodowanie przypisujące wiele skupisk zamówień do jednego pojazdu.

Należy jednak zauważyć, że pomiędzy konfiguracjami wykorzystującymi PSO, a odpowiadającymi im konfiguracjami wykorzystującymi DE, oraz tymi wykorzystującymi kodowanie Ok, a tymi wykorzystującymi kodowanie Ai brak jest istotnych różnic (odpowiednio Rysunki A.4c i A.4d). Potwierdza to porównanie pomiędzy wszystkimi przeprowadzonymi eksperymentami z dwoma skupiskami zamówień na pojazd i bezpośrednim transferem informacji zaprezentowane w Tabeli A.1, gdzie można zaobserwować, że brak jest w tym porównaniu konfiguracji istotnie lepszej od pozostałych.

Z uzyskanych wyników można wnioskować, że w przypadku niewielkiego budżetu (wyrażonego czasem na obliczenia lub liczbą ewaluacji) najlepiej zastosować kodowanie Ok z pojedynczym skupiskiem zamówień na pojazd i algorytmem DE. W przypadku większego budżetu warto zwiększyć liczbę skupisk zamówień do dwóch na pojazd, natomiast dalsze zwiększanie liczby skupisk prowadzi do pogorszenia wyników, prawdopodobnie ze względu na wzrost wymiaru przestrzeni przeszukiwań. Jednocześnie połączenie autorskiej koncepcji wielu skupisk zamówień na pojazd z reprezentacją w jednym wektorze rozwiązania zarówno kolejności jak i centroidów skupisk zamówień (kodowanie Ai) wydaje się być odpowiednie do najszerszej klasy problemów (z ewentualnym uzupełnieniem centroidów o wymiar czasowy), a jednocześnie bardziej elastyczne w stosunku do pojedynczych skupisk.

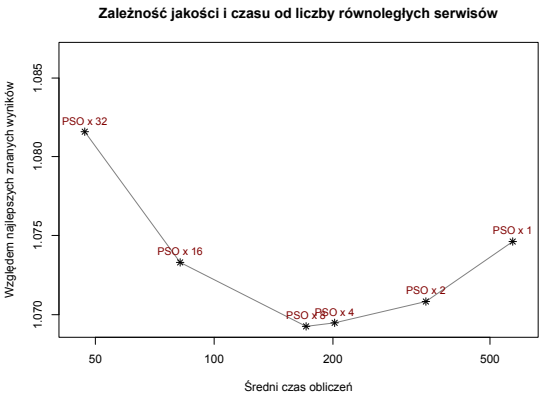
A.5 Dobór liczby równoległych serwisów optymalizacyjnych

Kolejnym istotnym zagadnieniem do przetestowania była weryfikacja, czy zrównoleglenie obliczeń nie prowadzi do spadku jakości rozwiązań. Przy dużym zrównolegleniu, ze względu na coraz mniejszą liczbę osobników w pojedynczym procesie i brak wymiany informacji pomiędzy serwisami optymalizacyjnymi w obrębie jednego kroku czasowego, spodziewany był spadek jakości optymalizacji.

W eksperymentach modyfikowane było prawdopodobieństwo komunikacji pomiędzy cząstkami roju w ten sposób, aby uzyskać podobny nakład obliczeniowy co w bazowym eksperymencie z ośmioma serwisami optymalizacyjnymi (rojami).

Tabela A.2: Porównanie średnich wyników uzyskanych przez podejścia z 1., 2., 4., 8., 16. i 32. rojami z całkowitym budżetem 10^6 ewaluacji funkcji jakości.

	ContDVRP (1)	*ContDVRP* (2)	*ContDVRP* (4)	*ContDVRP* (8)	*ContDVRP* (16)	*ContDVRP* (32)
	Śr.	Śr.	Śr.	Śr.	Śr.	Śr.
c50	**577.84**	584.22	584.22	578.31	585.92	579.29
c75	907.69	909.25	904.38	**903.72**	909.79	911.59
c100	929.97	**926.21**	926.99	933.46	926.69	936.07
c100b	844.15	845.27	**836.77**	845.8	847.99	844.4
c120	1118.33	1099.3	1102.65	**1071.38**	1081.55	1084.14
c150	1146.93	1139.07	1143.26	**1134.2**	1157.29	1158.52
c199	**1402.26**	1403.91	1405.13	1408.7	1435.32	1430.9
f71	293.32	294.29	292.74	298.5	**289.02**	292.37
f134	11969.52	11980.93	11897.61	**11892**	11920.35	11995.15
tai75a	1823.35	1802.13	**1802.01**	1805.03	1819.72	1824.8
tai75b	1426.32	1423.41	**1414.32**	1422.6	1419.68	1451.29
tai75c	1507.11	1510.7	**1501.65**	1510	1508.19	1516.12
tai75d	1447.09	1435.03	1442.73	**1433.25**	1444.7	1439.29
tai100a	2262.96	2240.08	2231.74	**2216.23**	2239.65	2279.16
tai100b	2153.12	2139.38	2147.33	**2136.8**	2150.43	2176.92
tai100c	1497.03	1498.18	1500.57	1494.72	**1490.65**	1519.67
tai100d	1732.49	1740.9	**1719.67**	1727.95	1729.94	1765.87
tai150a	3571.96	3503.57	**3499.02**	3530.82	3599.7	3594.81
tai150b	3036.43	**3023.1**	3048.28	3026.89	3031.7	3076.36
tai150c	2585.7	2599.32	2600.65	2603.53	**2582.56**	2649.51
tai150d	3038.09	**2989.85**	2998.66	3009.01	3027.67	3065.02
sum	45271.66	45088.1	45000.38	**44982.9**	45198.51	45591.25

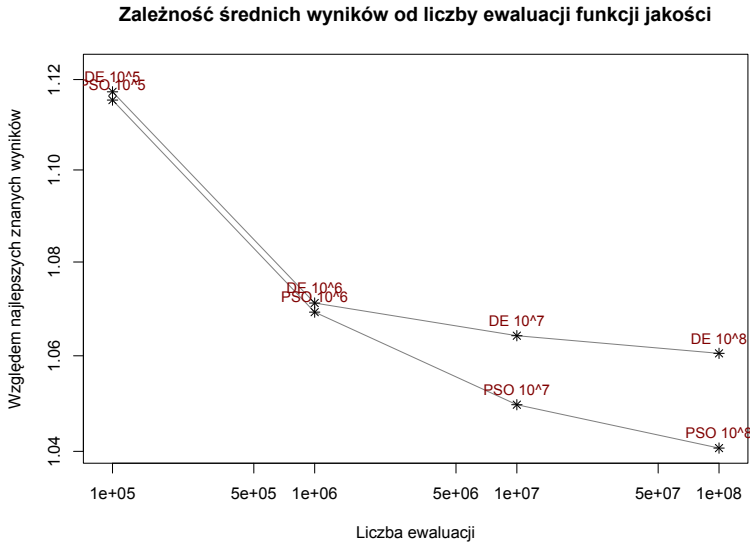


Wyniki eksperymentu (zaprezentowane w Tabeli A.2) potwierdziły, że brak jest istotnych różnic pomiędzy uzyskanymi w ten sposób wynikami, a zrównoleglenie na poziomie czterech i ośmiu serwisów prowadzi nie tylko do skrócenia czasu obliczeń, ale również do zysku na jakości, zwłaszcza względem pojedynczego serwisu.

A.6 Wpływ wielkości budżetu na liczbę ewaluacji funkcji jakości

W celu zbadania zależności między wynikami a liczbą ewaluacji funkcji jakości, przeprowadzono dla PSO i DE eksperymenty polegające na dziesięciokrotnym zmniejszeniu, dziesięciokrotnym oraz stukrotnym zwiększeniu budżetu na ewaluację funkcji jakości względem eksperymentów bazowych. Wyniki tych eksperymentów są zaprezentowane w Tabelach A.3 i A.4. Dodatkowo Rysunek A.5 przedstawia wyniki tych eksperymentów w ukła-

dzie zależności pomiędzy czasem przeznaczonym na obliczenia a uzyskanym rezultatem. Z uzyskanych wyników można wywnioskować, że równomierny sposób skalowania liczby obliczeń pomiędzy liczbę cząstek a liczbę iteracji przy zwiększaniu budżetu sprawdził się w przypadku PSO, natomiast DE prawdopodobnie mógłby osiągnąć lepsze wyniki przy innym stosunku dla większego budżetu ewaluacji funkcji jakości.



Rysunek A.5: Średni względny wynik (po 30 przebiegach każdego z 21 problemów testowych) dla dwóch algorytmów optymalizacyjnych operujących na ciągłym kodowaniu oraz różnym budżecie na ewaluację funkcji jakości.

Uzyskane wyniki pokazują także, że ograniczenia w reprezentowalności rozwiązań wynikające z proponowanego kodowania nie stanowią przeszkody w uzyskiwaniu dobrej jakości wyników na zbiorze instancji testowych. Można natomiast próbować zaproponować operatory w tej przestrzeni prowadzące do szybszej zbieżności algorytmów w stosunku do ogólnych operatorów wykorzystywanych w PSO i DE.

Tabela A.3: Porównanie wyników dla ciągłego kodowania problemu optymalizowanego algorytmem PSO dla różnej wielkości budżetu na liczbę ewaluacji funkcji jakości. Podane wartości oznaczają odpowiednio: PSO_2 - zastosowanie algorytmu PSO z 2 klastrami zamówień na pojazd, 40 - liczba kroków czasowych, 8 - liczba równoległych instancji serwisów optymalizacyjnych, $0.31 * 10^l$ - liczba ewaluacji funkcji jakości w ramach jednego kroku czasowego. Wykres obok tabeli prezentuje zagregowane względne wyniki dla budżetu danej wielkości.

	PSO_2 $(40*8*(0.31*10^3))$		PSO_2 $(40*8*(0.31*10^4))$		PSO_2 $(40*8*(0.31*10^5))$		PSO_2 $(40*8*(0.31*10^6))$	
	Min	Śr.	Min	Śr.	Min	Śr.	Min	Śr.
c50	562.57	589.11	**544.11**	578.31	562.76	573.81	558.04	**572.69**
c75	886.74	936.65	884.43	903.72	**866.34**	894.71	866.92	**886.03**
c100	949.38	969.56	902	933.46	886.52	925.47	**879.04**	**914.13**
c100b	822.46	855.86	**819.56**	845.8	**819.56**	**836.23**	**819.56**	842.54
c120	1074.37	1083.85	**1053.18**	**1071.38**	1059.4	1083.01	1059.48	1095.38
c150	1176.48	1207.39	1098.03	1134.2	1083.86	1114.09	**1075.92**	**1109.19**
c199	1412.77	1491.53	1362.65	1408.7	1341.47	1375.77	**1339.51**	**1362.09**
f71	279.27	295.95	274.16	298.5	**272.86**	292.88	273.5	**290.7**
f134	11978.77	12202.41	11746.4	11892	11735.98	11768.98	**11673.54**	**11758.56**
tai75a	1836.89	1906.89	1685.23	1805.03	**1680.55**	1757.12	1691.32	**1739.73**
tai75b	1460.59	1510.59	1365.36	1422.6	1357.32	1382.52	**1351.99**	**1368.74**
tai75c	1519.95	1559.24	1439.02	1510	**1415.18**	1477.71	1419.29	**1469.05**
tai75d	1433.62	1463.53	1408.79	1433.25	**1398.8**	**1423.15**	1399.8	1425.55
tai100a	2258.92	2368.8	2137.3	2216.23	2118.38	2200.03	**2110.86**	**2172.41**
tai100b	2134.31	2236.01	2060.65	2136.8	**2013.74**	2088.83	2023.65	**2058.99**
tai100c	1526.36	1590.31	1458.81	1494.72	1428.99	1466.16	**1427.65**	**1450.84**
tai100d	1786.45	1840.23	1663.87	1727.95	**1633.32**	1710.59	1648.5	**1691.96**
tai150a	3521.64	3732.19	3338.71	3530.82	3212.62	3370.91	**3190.33**	**3274.96**
tai150b	3082.06	3201.58	2910.06	3026.89	**2842.36**	2954.84	2862.04	**2930.77**
tai150c	2626.84	2740.67	2497.65	2603.53	2451.53	2522.03	**2407.61**	**2454.36**
tai150d	3036.42	3210.58	2869.79	3009.01	2795.46	2884.79	**2773.98**	**2835.5**
sum	45366.86	46992.93	43519.76	44982.9	42977	44103.63	**42852.53**	**43704.17**

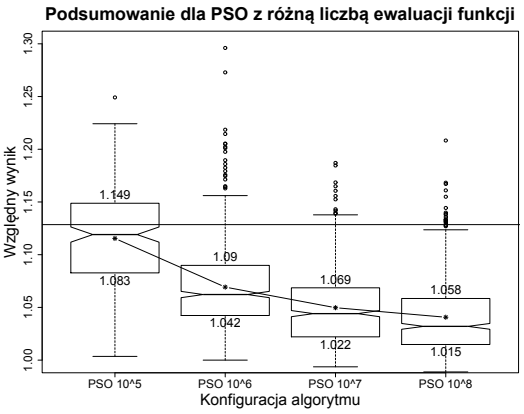
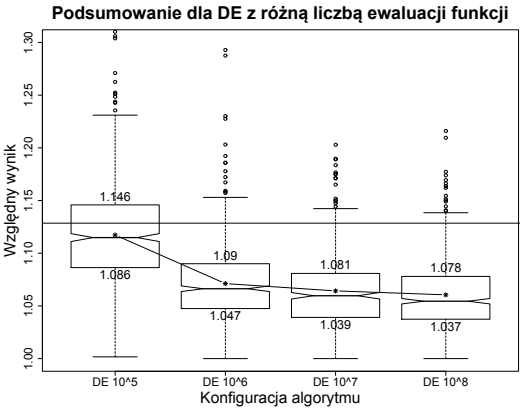


Tabela A.4: Porównanie wyników dla ciągłego kodowania problemu optymalizowanego algorytmem DE dla różnej wielkości budżetu na liczbę ewaluacji funkcji jakości. Podane wartości oznaczają odpowiednio: DE_2 - zastosowanie algorytmu DE z 2 klastrami zamówień na pojazd, 40 - liczba kroków czasowych, 8 - liczba równoległych instancji serwisów optymalizacyjnych, $0.31 * 10^l$ - liczba ewaluacji funkcji jakości w ramach jednego kroku czasowego. Wykres obok tabeli prezentuje zagregowane względne wyniki dla budżetu danej wielkości.

	DE_2 $(40*8*(0.31*10^3))$		DE_2 $(40*8*(0.31*10^4))$		DE_2 $(40*8*(0.31*10^5))$		DE_2 $(40*8*(0.31*10^6))$	
	Min	Śr.	Min	Śr.	Min	Śr.	Min	Śr.
c50	566.23	587.57	563.09	580.02	**550.68**	573.55	553.3	**572.9**
c75	891.2	931.49	878.32	908.28	879.2	906.41	**872.88**	**893.24**
c100	899.24	952.39	893.24	937.07	900.81	920.7	**880.47**	**914.69**
c100b	820.92	841.03	**819.56**	843.83	**819.56**	**834.98**	**819.56**	846.56
c120	1067.62	**1095.84**	1067.64	1104.61	1064.86	1127.46	**1059.87**	1137.43
c150	1153.39	1197.18	1103.46	1140.14	1100.5	1126.47	**1087.5**	**1113.54**
c199	1427.56	1489.25	1379.34	**1414.85**	1382.29	1458.59	**1363.93**	1437
f71	278.52	295.7	**273**	295.58	279.03	**294.42**	273.82	294.82
f134	12031.82	12389.26	**11743.05**	**11916.7**	11751.51	12015.79	11744.04	12011.44
tai75a	1751.39	1896.46	1725.11	1807.49	**1686.8**	**1761.55**	1706.84	1773.64
tai75b	1421.32	1507.57	**1372.39**	1412.85	1379.85	**1411.84**	1379.1	1421.97
tai75c	1472.74	1564.62	1423.21	1501.64	1416.05	1474.95	**1410.78**	**1471.32**
tai75d	1427.95	1491.75	1416.89	1451.04	**1404.56**	1434.2	1405.43	**1434.07**
tai100a	2260.28	2404.77	2193.61	2247.21	**2132.3**	2209.93	2137.26	**2187.91**
tai100b	2148.57	2238.41	2041.46	2145.85	**2032.49**	**2103.91**	2035.01	2107.63
tai100c	1505.83	1573.91	**1442.63**	1495.87	1460.76	1495.67	1444.48	**1483.61**
tai100d	1758.87	1842.85	1694.79	1736.33	1659.79	1726.7	**1650.53**	**1712.01**
tai150a	3610.76	3853.73	3346.17	3489.65	3307.59	3429.35	**3281.94**	**3416.75**
tai150b	3052.38	3166.54	2914.79	3038.63	2939.2	3056.8	**2867.02**	**2995.34**
tai150c	2588.75	2772.57	2505.7	2563.01	2456.71	2508.01	**2429.98**	**2504.91**
tai150d	3022.5	3229.34	2890.02	2999.43	2858.18	2981.39	**2812.44**	**2952.18**
sum	45157.84	47322.23	43687.47	45030.08	43462.72	44852.67	**43216.18**	**44682.96**



Dodatek B

Uzyskane wyniki

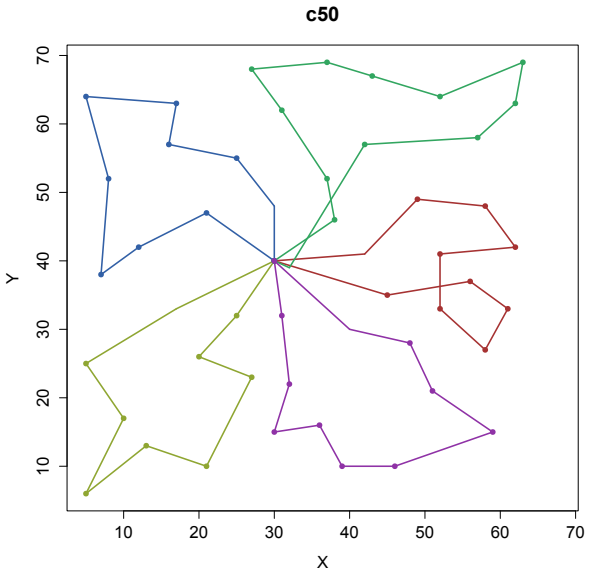
W tym dodatku zaprezentowane są wartości najlepszych znanych wyników dla instancji testowych oraz najlepsze wyniki znalezione przez ContDVRP. Wartości najlepszych wyników (wraz z algorytmem oraz łącznym budżetem liczby ewaluacji funkcji jakości) przedstawione są w Tabeli B.1. Najskuteczniejszym w znajdowaniu najlepszych wyników algorytmem jest M-VRPDR, który był w stanie znaleźć 14 najlepszych wyników, ContDVRP znalazł 6, a MEMSO 1.

Tabele B.2 – B.22 przedstawiają najlepsze wyniki znalezione przez ContDVRP w formie harmonogramów dla floty pojazdów. Harmonogramy zawierają: identyfikatory pojazdów (kolumna *Pojazd*), identyfikatory zamówień (kolumna *Zamówienie*), ich lokalizację (kolumny X i Y), czas zgłoszenia zamówienia (kolumna *Otrzymane*), czas planowanego przyjazdu do zamówienia (kolumna *Obsłużone*). Zamówienia są pogrupowane pojazdami i uszeregowane w kolejności odwiedzin, poziome linie oznaczają powroty do zajezdni. Wizualizacje tras z tych harmonogramów zaprezentowane są na Rysunkach B.1 – B.6.

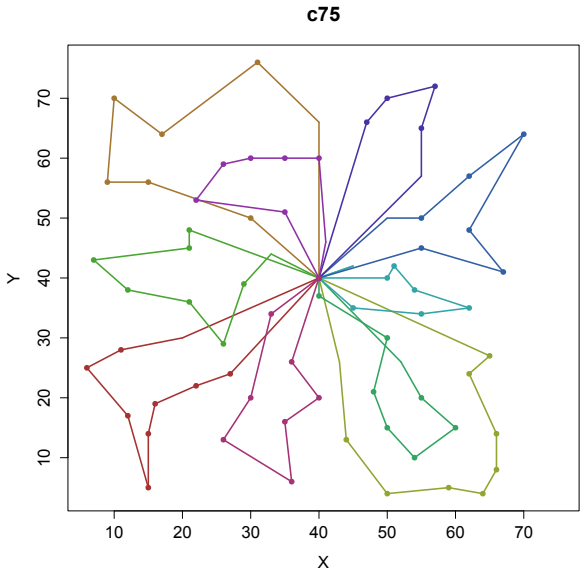
Wizualizacje najlepszych wyników osiągniętych przez ContDVRP są również prezentowane na stronie projektu poświęconego metaheurystykom w grach i problemach transportowych [70].

Tabela B.1: Sumaryczne długości tras dla najlepszych rozwiązań zbioru instancji testowych z $T_{CO} = 0.5$. Prawa kolumna (*Liczba ewaluacji*) prezentuje całościowy budżet na liczbę ewaluacji funkcji jakości przy jakim osiągnięto dany wynik.

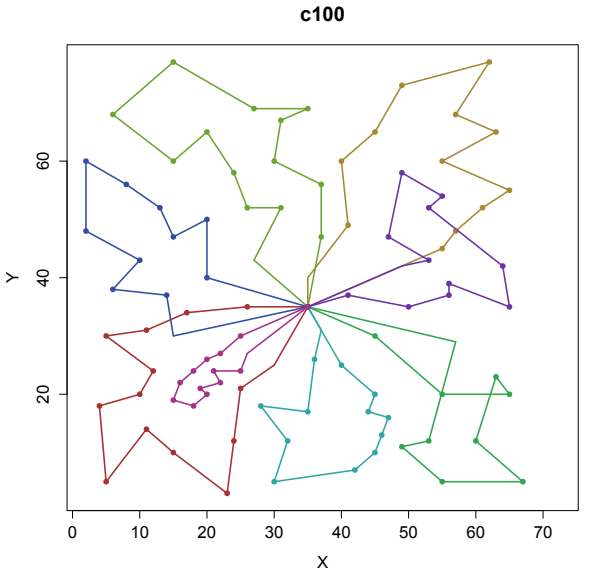
Nazwa	**Najlepszy wynik**	**Algorytm**	**Liczba ewaluacji**
c50	524.61	$M-VRPDR$	10^7
c75	852.95	$M-VRPDR$	10^7
c100	860.56	$M-VRPDR$	10^7
c100b	819.56	$ContDVRP_{PSO}$	$8*40*0.31*10^4$
c120	1052.75	$ContDVRP_{PSO}$	$8*40*0.31*10^6$
c150	1081.48	$ContDVRP_{PSO}$	$8*40*0.31*10^6$
c199	1350.04	$ContDVRP_{PSO}$	$8*40*0.31*10^6$
f71	260.17	$M-VRPDR$	10^7
f134	11658.14	$ContDVRP_{PSO}$	$8*40*0.31*10^6$
tai75a	1656.12	$M-VRPDR$	10^7
tai75b	1350.28	$M-VRPDR$	10^7
tai75c	1355.41	$M-VRPDR$	10^7
tai75d	1342.26	$MEMSO$	$8*25*0.5*10^4$
tai100a	2093.63	$M-VRPDR$	10^7
tai100b	1990.99	$M-VRPDR$	10^7
tai100c	1421.50	$M-VRPDR$	10^7
tai100d	1631.63	$M-VRPDR$	10^7
tai150a	3226.51	$M-VRPDR$	10^7
tai150b	2842.20	$ContDVRP_{PSO}$	$8*40*0.31*10^6$
tai150c	2427.53	$M-VRPDR$	10^7
tai150d	2737.37	$M-VRPDR$	10^7



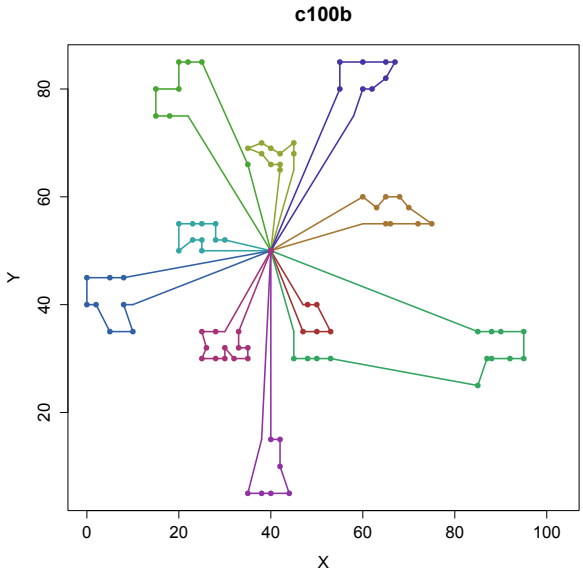
(a) Wizualizacja wyniku dla c50



(b) Wizualizacja wyniku dla c75



(c) Wizualizacja wyniku dla c100



(d) Wizualizacja wyniku dla c100b

Rysunek B.1: Wizualizacje wyników

Tabela B.2: Najlepsze rozwiązanie uzyskane przez ContDVRP dla **c50**

Pojazd	Zamówienie	X	Y	Otrzymane	Obsłużone
	Depot	30	40	0	351
1	27	30	48	0	34.33
1	48	25	55	0	57.93
1	8	31	62	34	155.84
1	26	27	68	0	178.06
1	7	17	63	21	204.24
1	23	16	57	157	225.32
1	43	5	64	0	253.36
1	24	8	52	0	280.73
1	18	17	33	100	316.75
2	5	40	30	12	93.12
2	49	48	28	0	116.36
2	10	51	21	44	138.98
2	39	59	15	0	163.98
2	33	46	10	0	192.91
2	45	39	10	0	214.91
2	15	36	16	80	236.62
2	44	30	15	0	257.70
2	37	32	22	0	279.98
2	17	27	23	86	300.08
2	47	25	32	0	324.30
3	4	20	26	12	78.63
3	41	10	17	0	107.08
3	19	13	13	104	160.40
3	42	21	10	0	183.94
3	40	5	6	0	215.44
3	13	5	25	63	249.44
3	25	7	38	0	277.59
3	14	12	42	79	298.99
3	6	21	47	16	324.29
4	32	38	46	0	53.88
4	2	49	49	4	80.28
4	29	58	48	0	104.33
4	35	62	63	0	134.86
4	36	63	69	0	155.94
4	20	57	58	116	183.47
4	3	52	64	9	206.28
4	28	43	67	0	230.77
4	31	37	69	0	252.09
4	22	42	57	138	280.09
4	1	37	52	1	302.16
4	46	32	39	0	331.09
5	12	31	32	58	130.91
5	38	45	35	0	160.23
5	9	52	33	42	182.51
5	30	58	27	0	206.00
5	34	61	33	0	227.70
5	21	62	42	128	251.76
5	50	56	37	0	274.57
5	16	52	41	80	295.23
5	11	42	41	51	320.23

Tabela B.3: Najlepsze rozwiązanie uzyskane przez ContDVRP dla **c75**

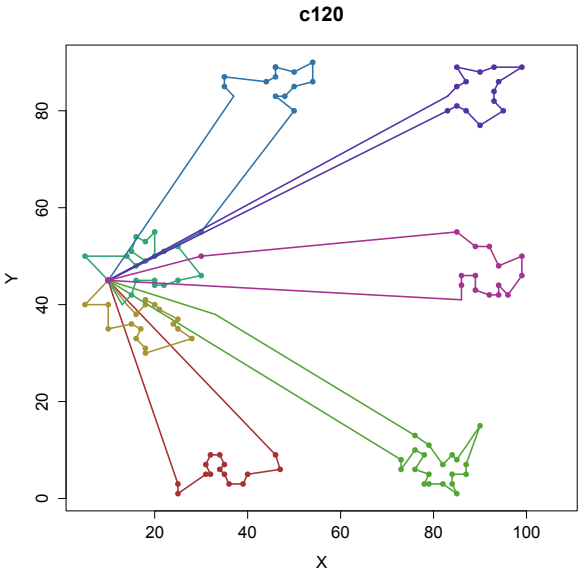
Pojazd	Zamówienie	X	Y	Otrzymane	Obsłużone
	Depot	40	40	0	346
1	73	27	24	0	115.77
1	1	22	22	3	137.15
1	43	16	19	0	159.86
1	42	15	14	0	180.96
1	64	15	5	0	205.96
1	41	12	17	0	234.33
1	56	6	25	0	260.33
1	23	11	28	85	282.16
1	63	20	30	0	307.38
2	40	30	50	0	126.59
2	50	15	56	0	158.75
2	18	9	56	66	180.75
2	55	10	70	0	210.78
2	25	17	64	101	236.00
2	31	31	76	121	270.44
2	10	40	66	24	299.90
3	57	65	27	0	114.68
3	15	62	24	42	134.92
3	20	66	14	79	161.69
3	70	66	8	0	183.69
3	60	64	4	0	204.16
3	71	59	5	0	225.26
3	69	50	4	0	250.32
3	21	44	13	84	277.13
3	30	43	26	119	306.17
4	44	21	48	0	98.47
4	3	21	45	8	117.47
4	24	7	43	87	151.24
4	49	12	38	0	174.31
4	16	21	36	42	232.82
4	33	26	29	128	257.42
4	51	29	39	0	283.86
4	17	33	44	47	306.27
5	75	40	37	0	3.00
5	45	50	30	0	31.21
5	48	48	21	0	56.43
5	47	50	15	0	203.97
5	36	54	10	138	226.38
5	37	60	15	150	250.19
5	5	55	20	17	273.26
5	29	52	26	112	295.97
6	4	45	35	13	171.42
6	27	55	34	108	197.47
6	13	62	35	32	220.54
6	52	54	38	0	245.09
6	46	51	42	0	266.09
6	34	50	40	131	284.32
6	67	45	42	0	315.71
7	8	55	45	20	136.91
7	54	67	41	0	165.56
7	19	62	48	71	190.16
7	59	70	64	0	224.05
7	14	62	57	34	250.68
7	35	55	50	135	276.58
7	7	50	50	20	297.58
8	38	47	66	152	199.93
8	65	50	70	0	220.93
8	66	57	72	0	244.21
8	11	55	65	25	267.49
8	53	55	57	0	291.49
9	12	35	51	30	167.78
9	32	22	53	122	196.94
9	9	26	59	22	220.15
9	39	30	60	152	240.27
9	72	35	60	0	261.27
9	58	40	60	0	282.27
9	26	41	46	107	312.31
10	6	33	34	20	138.97
10	62	30	20	0	169.29
10	22	26	13	84	193.35
10	61	36	6	0	221.56
10	28	35	16	109	247.61
10	74	40	20	0	270.01
10	2	36	26	5	293.22
10	68	38	33	0	316.50

Tabela B.4: Najlepsze rozwiązanie uzyskane przez ContDVRP dla **c100b**

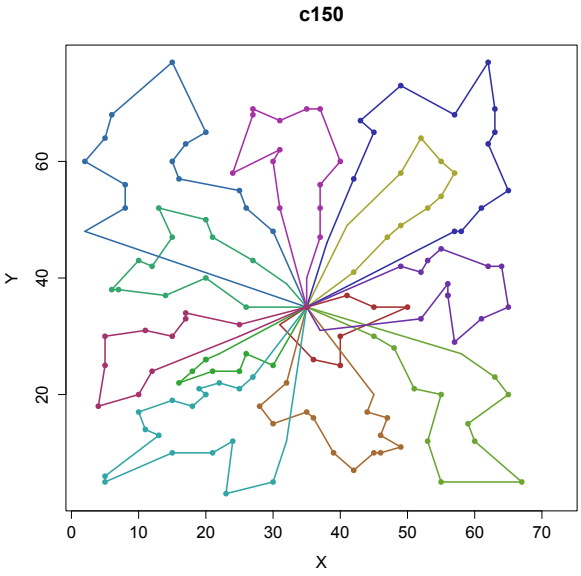
Pojazd	Zamówienie	X	Y	Otrzymane	Obsłużone
	Depot	40	50	0	468
1	66	47	35	0	222.47
1	62	50	35	0	237.47
1	74	53	35	0	380.04
1	63	50	40	0	397.87
1	65	48	40	0	411.87
1	67	47	40	0	424.87
2	91	60	60	0	87.88
2	89	63	58	0	103.49
2	88	65	60	0	304.99
2	85	68	60	0	319.99
2	84	70	58	0	334.82
2	82	75	55	0	352.65
2	83	72	55	0	367.65
2	86	66	55	0	385.65
2	87	65	55	0	398.65
2	90	60	55	0	415.65
3	5	42	65	20	267.85
3	3	42	66	11	280.85
3	7	40	66	33	294.85
3	8	38	68	42	309.68
3	11	35	69	56	324.84
3	9	38	70	48	340.01
3	6	40	69	25	354.24
3	4	42	68	18	368.48
3	2	45	70	11	384.08
3	1	45	68	2	398.08
3	75	45	65	0	413.08
4	10	35	66	56	260.12
4	12	25	85	58	293.59
4	14	22	85	67	308.59
4	16	20	85	69	322.59
4	15	20	80	69	339.59
4	19	15	80	75	356.59
4	18	15	75	75	373.59
4	17	18	75	70	388.59
4	13	22	75	67	404.59
5	81	85	35	0	197.19
5	78	88	35	0	212.19
5	76	90	35	0	226.19
5	71	95	35	0	243.19
5	70	95	30	0	269.72
5	73	92	30	0	284.72
5	77	88	30	0	300.72
5	79	87	30	0	313.72
5	80	85	25	0	331.11
5	72	53	30	0	375.49
5	61	50	30	0	390.49
5	64	48	30	0	404.49
5	68	45	30	0	419.49
5	69	45	35	0	436.49
6	21	30	52	88	272.28
6	22	28	52	88	286.28
6	23	28	55	97	301.28
6	26	25	55	100	316.28
6	28	23	55	110	330.28
6	30	20	55	114	345.28
6	29	20	50	110	362.28
6	27	23	52	105	377.88
6	25	25	52	98	391.88
6	24	25	50	97	405.88
6	20	30	50	86	422.88
7	34	8	45	127	275.75
7	36	5	45	134	290.75
7	39	0	45	148	307.75
7	38	0	40	145	324.75
7	37	2	40	140	338.75
7	35	5	35	130	356.58
7	31	10	35	120	373.58
7	33	8	40	124	390.96
7	32	10	40	124	404.96
8	99	55	80	0	276.90
8	100	55	85	0	293.90
8	97	60	85	0	310.90
8	93	65	85	0	327.90
8	92	67	85	0	341.90
8	94	65	82	0	357.51
8	95	62	80	0	373.11
8	96	60	80	0	387.11
8	98	58	75	0	404.50
9	57	40	15	206	278.36
9	55	42	15	205	292.36
9	54	42	10	201	309.36
9	53	44	5	199	326.75
9	56	40	5	205	342.75
9	58	38	5	207	356.75
9	60	35	5	0	371.75
9	59	38	15	217	394.19
10	43	33	35	174	194.39
10	42	33	32	173	209.39
10	41	35	32	172	223.39
10	40	35	30	165	237.39
10	44	32	30	174	295.80
10	46	30	32	187	310.63
10	45	30	30	180	324.63
10	48	28	30	189	338.63
10	51	25	30	196	353.63
10	50	26	32	196	367.86
10	52	25	35	196	383.03
10	49	28	35	191	398.03
10	47	30	35	188	412.03

Tabela B.5: Najlepsze rozwiązanie uzyskane przez ContDVRP dla **c100**

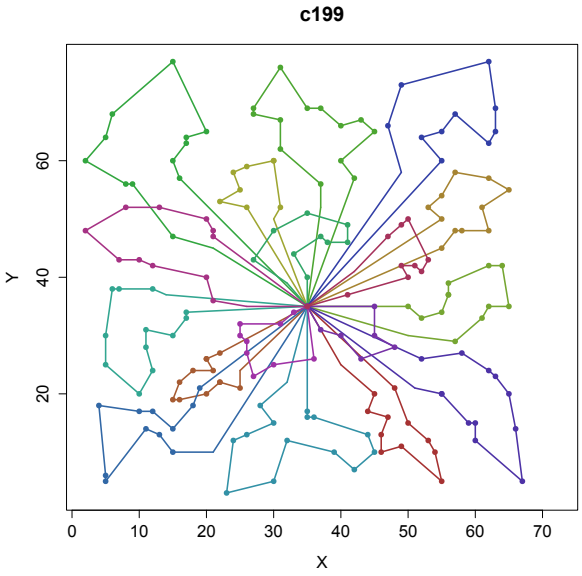
Pojazd	Zamówienie	X	Y	Otrzymane	Obsłużone
	Depot	35	35	0	399
1	89	26	35	0	78.83
1	60	17	34	0	99.88
1	84	11	31	0	148.38
1	17	5	30	60	166.47
1	61	12	24	0	187.69
1	16	10	20	59	204.16
1	86	4	18	0	222.48
1	38	5	5	124	247.52
1	44	11	14	149	270.34
1	14	15	10	57	287.99
1	43	23	3	149	310.62
1	42	24	12	147	331.68
1	97	25	21	0	352.74
1	13	30	25	57	371.14
2	3	55	45	9	42.31
2	79	57	48	0	57.92
2	78	61	52	0	75.57
2	34	65	55	108	176.60
2	9	55	60	41	199.78
2	35	63	65	111	221.21
2	71	57	68	0	239.92
2	65	62	77	0	262.22
2	66	49	73	0	287.82
2	20	45	65	73	308.76
2	30	40	60	97	327.83
2	1	41	49	2	350.88
2	27	35	40	89	373.70
3	69	37	47	0	91.97
3	70	37	56	0	112.97
3	10	30	60	47	133.03
3	90	31	67	0	152.10
3	32	35	69	105	168.57
3	63	27	69	0	188.57
3	64	15	77	0	214.99
3	49	6	68	163	239.72
3	19	15	60	64	263.76
3	11	20	65	48	282.83
3	62	24	58	0	302.90
3	88	26	52	0	321.22
3	31	31	52	102	338.22
3	52	27	43	167	360.07
4	26	45	30	86	130.88
4	4	55	20	16	157.02
4	25	65	20	83	179.02
4	55	63	23	174	205.13
4	39	60	12	126	228.53
4	67	67	5	0	250.43
4	23	55	5	82	274.43
4	75	49	11	0	294.92
4	56	53	12	175	311.04
4	54	57	29	171	340.50
5	40	40	25	141	160.81
5	21	45	20	75	179.88
5	73	44	17	0	195.04
5	72	47	16	0	210.20
5	74	46	13	0	225.36
5	22	45	10	75	240.53
5	41	42	7	146	256.77
5	15	30	5	59	280.93
5	57	32	12	176	300.21
5	87	28	18	0	319.42
5	2	35	17	9	338.50
5	58	36	26	177	359.55
5	53	37	31	169	376.65
6	18	20	40	64	165.44
6	7	20	50	28	187.44
6	82	15	47	0	205.27
6	48	13	52	161	222.65
6	47	8	56	160	241.06
6	36	2	60	114	260.27
6	46	2	48	159	284.27
6	8	10	43	35	305.70
6	45	6	38	153	324.10
6	83	14	37	0	344.17
6	5	15	30	17	363.24
7	28	41	37	94	136.00
7	12	50	35	49	157.22
7	80	56	37	0	175.54
7	68	56	39	0	189.54
7	24	65	35	83	211.39
7	29	64	42	94	230.46
7	33	53	52	106	257.33
7	81	55	54	0	272.16
7	51	49	58	167	291.37
7	50	47	47	167	314.55
7	77	53	43	0	333.76
7	76	49	42	0	349.88
8	6	25	30	22	170.78
8	96	22	27	0	187.02
8	99	20	26	0	201.26
8	93	18	24	0	216.09
8	85	16	22	0	230.92
8	91	15	19	0	246.08
8	100	18	18	0	261.24
8	37	20	20	119	276.07
8	98	19	21	0	289.48
8	92	22	22	0	304.65
8	59	21	24	185	318.88
8	95	25	24	0	334.88
8	94	26	27	0	350.04



(a) Wizualizacja wyniku dla c120



(b) Wizualizacja wyniku dla c150



(c) Wizualizacja wyniku dla c199

Rysunek B.2: Wizualizacje wyników

Tabela B.6: Najlepsze rozwiązanie uzyskane przez ContDVRP dla **c120**

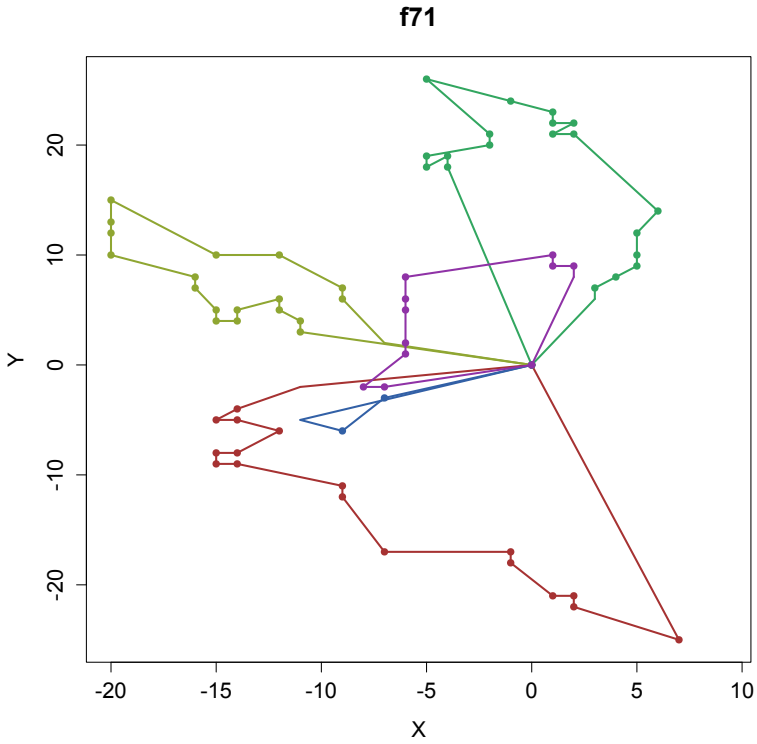
Pojazd	Zamówienie	X	Y	Otrzymane	Obsłużone
	Depot	10	45	0	794
1	8	46	9	38	467.76
1	12	47	6	70	483.92
1	13	40	5	71	504.00
1	14	39	3	89	519.23
1	15	36	3	114	535.23
1	11	35	5	70	550.47
1	10	34	6	62	564.88
1	9	35	7	46	579.30
1	7	34	9	35	594.53
1	6	32	9	34	609.53
1	5	31	7	29	624.77
1	4	32	5	27	640.00
1	3	31	5	26	654.00
1	1	25	1	1	674.21
1	2	25	3	14	689.21
1	88	11	42	0	743.65
2	119	5	40	0	463.62
2	82	10	40	0	481.62
2	81	10	35	0	499.62
2	112	15	36	0	517.72
2	84	17	35	0	532.96
2	117	16	33	0	548.19
2	113	18	31	0	564.02
2	83	18	30	0	578.02
2	108	28	33	0	601.46
2	118	25	35	0	618.07
2	18	24	36	152	632.48
2	114	25	37	0	646.89
2	90	21	39	0	664.37
2	91	20	40	0	678.78
2	92	18	41	0	694.02
2	89	18	40	0	708.02
2	85	16	38	0	723.85
2	86	14	40	0	739.67
3	17	73	8	140	350.96
3	16	73	6	122	365.96
3	20	76	10	172	383.96
3	23	78	9	200	399.20
3	19	76	6	166	415.80
3	25	79	5	223	431.97
3	22	78	3	188	447.20
3	24	79	3	206	461.20
3	27	82	3	226	477.20
3	33	85	1	274	493.81
3	30	84	3	251	509.04
3	31	84	5	254	524.04
3	34	87	5	276	540.04
3	36	87	7	292	555.04
3	29	90	15	235	576.59
3	35	85	8	289	598.19
3	32	84	9	273	612.60
3	28	82	7	234	628.43
3	26	79	11	223	646.43
3	21	76	13	181	663.04
3	109	33	38	0	725.78
4	120	5	50	0	423.92
4	105	14	50	0	445.92
4	102	16	48	0	461.75
4	101	18	49	0	476.99
4	106	15	51	0	493.59
4	107	16	54	0	509.75
4	104	18	53	0	524.99
4	103	20	55	0	551.78
4	99	20	50	0	569.78
4	100	22	51	0	585.01
4	116	25	52	0	601.18
4	115	30	46	0	621.99
4	97	25	45	0	640.09
4	94	22	44	0	656.25
4	93	20	44	0	671.25
4	96	20	45	0	685.25
4	95	16	45	0	702.25
4	87	15	42	0	718.41
4	111	13	40	0	734.24
5	98	30	55	0	439.21
5	68	50	80	0	484.23
5	73	46	83	0	502.23
5	76	48	83	0	517.23
5	77	50	85	0	533.05
5	79	54	86	0	550.18
5	80	54	90	0	567.18
5	78	50	88	0	584.65
5	75	46	89	0	601.77
5	72	46	89	0	614.77
5	74	46	87	0	629.77
5	71	44	86	0	645.01
5	70	35	87	0	667.06
5	69	35	85	0	682.06
5	67	37	83	0	697.89
6	52	83	80	0	378.71
6	54	85	81	0	432.09
6	57	87	80	0	447.32
6	59	90	77	0	464.56
6	65	95	80	0	483.40
6	61	93	82	0	499.22
6	62	93	84	0	514.22
6	64	94	86	0	529.46
6	66	99	89	0	548.29
6	63	93	89	0	567.29
6	60	90	88	0	583.45
6	56	85	89	0	601.55
6	58	87	86	0	618.16
6	55	85	85	0	633.39
6	53	83	83	0	649.22
7	110	30	50	0	377.92
7	40	85	55	324	446.14
7	43	89	52	339	464.14
7	45	92	52	356	480.14
7	48	94	48	377	497.61
7	51	99	50	0	516.00
7	50	99	46	396	533.00
7	49	96	42	383	551.00
7	47	94	44	361	566.83
7	46	94	42	358	581.83
7	44	92	42	346	596.83
7	41	89	43	330	612.99
7	42	89	46	330	628.99
7	39	86	46	324	644.99
7	38	86	44	324	659.99
7	37	86	41	304	675.99

Tabela B.7: Najlepsze rozwiązanie uzyskane przez ContDVRP dla **c150**

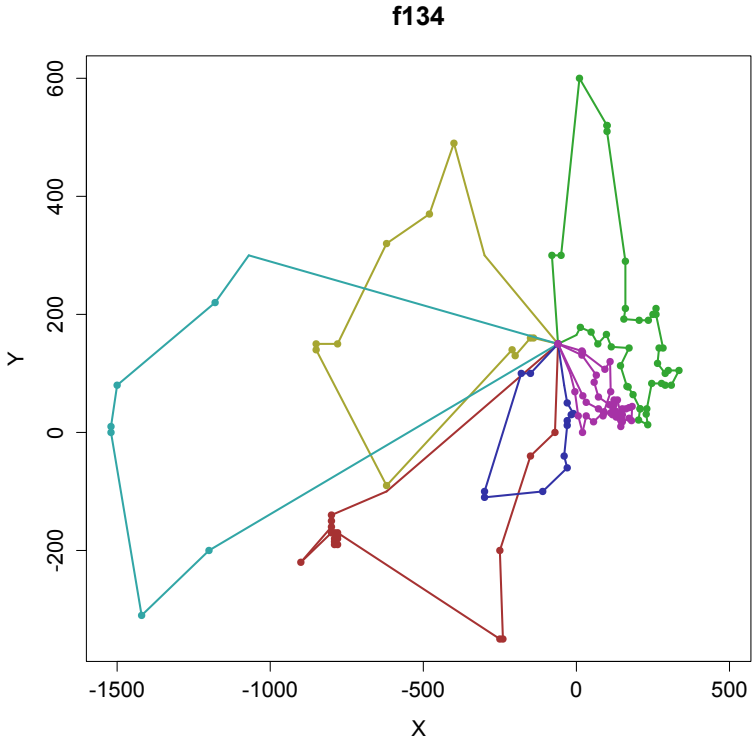
Pojazd	Zamówienie	X	Y	Otrzymane	Obsłużone
	Depot	35	35	0	399
1	28	41	37	71	255.70
1	138	45	35	0	270.17
1	12	50	35	25	285.17
1	105	40	30	0	306.35
1	40	40	25	99	334.20
1	58	36	26	150	348.32
1	112	31	32	0	366.13
2	137	32	22	0	162.97
2	87	28	18	0	178.62
2	144	30	15	0	192.23
2	2	35	17	4	207.61
2	115	36	16	0	219.03
2	145	39	10	0	246.13
2	41	42	7	100	260.38
2	22	45	10	61	274.62
2	133	46	10	0	285.62
2	75	49	11	0	298.78
2	74	46	13	0	312.39
2	72	47	16	0	325.55
2	73	44	17	0	338.71
2	21	45	20	59	351.87
3	111	42	41	0	198.74
3	50	47	47	122	216.55
3	102	49	49	0	229.38
3	33	53	52	83	244.38
3	81	55	54	0	257.21
3	120	57	58	0	271.68
3	9	55	60	20	302.10
3	103	52	64	0	317.10
3	51	49	58	124	333.81
3	1	41	49	3	355.85
4	26	45	30	67	170.78
4	149	48	28	0	184.39
4	110	51	21	0	202.00
4	4	55	20	7	216.12
4	56	53	12	141	234.37
4	23	55	5	62	251.65
4	67	67	5	190	273.65
4	39	60	12	97	293.55
4	139	59	15	0	306.71
4	25	65	20	65	324.52
4	55	63	23	141	338.13
4	130	58	27	0	354.53
5	13	30	25	30	240.61
5	94	26	27	0	255.08
5	95	25	24	0	268.24
5	59	21	24	156	282.24
5	85	16	22	0	304.66
5	93	18	24	0	317.49
5	104	20	26	0	330.32
5	99	20	26	0	340.32
5	96	22	27	0	352.55
6	89	26	35	0	178.57
6	18	20	40	45	196.39
6	83	14	37	0	213.09
6	125	7	38	0	230.16
6	45	6	38	112	241.16
6	8	10	43	20	257.57
6	114	12	42	0	269.80
6	82	15	47	0	285.63
6	48	13	52	118	301.02
6	7	20	50	15	318.30
6	106	21	47	0	331.46
6	52	27	43	127	348.67
6	146	32	39	0	365.08
7	117	27	23	0	14.42
7	97	25	21	0	27.25
7	92	22	22	0	40.41
7	98	19	21	0	142.84
7	37	20	20	92	154.25
7	100	18	18	0	167.08
7	91	15	19	0	180.24
7	141	10	17	0	195.63
7	44	11	14	109	208.79
7	119	13	13	0	221.03
7	140	5	6	0	241.66
7	38	5	5	95	252.66
7	14	15	10	33	273.84
7	142	21	10	0	289.84
7	42	24	12	103	303.44
7	43	23	3	108	322.50
7	15	30	5	36	339.78
7	57	32	12	143	357.06
8	127	30	48	0	123.65
8	88	26	52	0	139.31
8	148	25	55	0	152.47
8	123	16	57	0	188.79
8	19	15	60	52	201.96
8	107	17	63	0	215.56
8	11	20	65	22	229.17
8	64	15	77	180	252.17
8	49	6	68	119	274.90
8	143	5	64	0	289.02
8	36	2	60	89	304.02
8	47	8	56	118	321.23
8	124	8	52	0	335.23
8	46	2	48	117	352.44
9	79	57	48	0	155.23
9	129	58	48	0	166.23
9	78	61	52	0	181.23
9	34	65	55	83	196.23
9	135	62	63	0	214.77
9	35	63	65	86	227.01
9	136	63	69	0	241.01
9	65	62	77	181	259.07
9	71	57	68	0	279.37
9	66	49	73	185	298.80
9	128	43	67	0	317.29
9	20	45	65	53	330.11
9	122	42	57	0	348.66
9	132	38	46	0	370.36
10	76	49	42	0	155.30
10	116	52	41	0	168.46
10	77	53	43	0	180.70
10	3	55	45	5	193.53
10	121	62	42	0	211.15
10	29	64	42	71	251.40
10	24	65	35	63	268.47
10	134	61	33	0	282.94
10	54	57	29	130	298.60
10	150	56	37	0	316.66
10	80	56	37	0	326.66
10	68	56	39	192	338.66
10	109	52	33	0	355.87
10	53	37	31	128	381.01
11	31	31	52	76	177.06
11	10	30	60	22	195.13
11	108	31	62	0	207.36
11	62	24	58	174	225.42
11	126	27	68	0	245.87
11	63	27	69	180	256.87
11	90	31	67	0	271.34
11	32	35	69	80	285.81
11	131	37	69	0	297.81
11	30	40	60	75	317.30
11	70	37	56	197	332.30
11	101	37	52	0	346.30
11	69	37	47	193	361.30
11	27	35	40	70	378.58
12	147	25	32	0	199.97
12	60	17	34	169	218.21
12	118	17	33	0	229.21
12	5	15	30	9	253.01
12	84	11	31	0	267.13
12	17	5	30	45	283.21
12	113	5	25	0	298.21
12	86	4	18	0	315.28
12	16	10	20	41	331.61
12	61	12	24	170	346.08
12	6	25	30	11	370.40

Tabela B.8: Najlepsze rozwiązanie uzyskane przez ContDVRP dla **c199**

Pojazd	Zamówienie	X	Y	Otrzymane	Obsłużone
	Depot	35	35	0	399
1	198	48	21	0	218.60
1	197	50	15	0	233.93
1	56	53	12	111	247.17
1	186	54	10	0	258.41
1	23	55	5	39	272.51
1	75	49	11	146	289.99
1	133	46	10	0	302.15
1	74	46	13	144	314.15
1	72	47	16	143	326.32
1	73	44	17	144	338.48
1	21	45	20	33	350.64
1	40	40	25	66	366.71
2	96	22	27	0	214.76
2	99	20	26	0	226.00
2	104	20	26	0	235.00
2	59	21	24	118	246.24
2	93	18	24	194	258.24
2	85	16	22	166	270.06
2	91	15	19	191	282.23
2	193	16	19	0	292.23
2	37	20	20	61	305.35
2	92	22	22	194	317.18
2	151	22	22	0	326.18
2	97	25	21	0	338.34
2	95	25	24	0	350.34
3	185	55	50	0	214.53
3	33	53	52	53	226.35
3	81	55	54	160	238.18
3	120	57	58	0	251.65
3	164	62	57	0	265.75
3	34	65	55	55	278.36
3	78	61	52	152	292.36
3	169	62	48	0	305.48
3	129	58	48	0	318.48
3	79	57	48	157	328.48
3	158	55	45	0	341.09
3	3	55	45	3	350.09
4	88	26	52	178	258.64
4	182	22	53	0	271.76
4	148	25	55	0	284.36
4	62	24	58	122	296.53
4	159	26	59	0	307.76
4	189	30	60	0	320.89
4	10	30	60	17	329.89
4	31	31	52	52	346.95
4	190	30	50	0	358.18
5	12	50	35	19	204.53
5	109	52	33	0	216.35
5	177	55	34	0	228.52
5	80	56	37	158	240.68
5	150	56	37	0	249.68
5	68	56	39	132	260.68
5	121	62	42	0	276.39
5	29	64	42	48	287.39
5	24	65	35	39	303.46
5	163	62	35	0	315.46
5	134	61	33	0	326.69
5	54	57	29	100	341.35
5	195	50	30	0	357.42
6	122	42	57	0	182.69
6	30	40	60	50	195.29
6	20	45	65	33	211.36
6	128	43	67	0	223.19
6	160	40	66	0	235.35
6	131	37	69	0	248.60
6	32	35	69	52	259.60
6	181	31	76	0	276.66
6	63	27	69	124	293.72
6	126	27	68	0	303.72
6	90	31	67	190	316.84
6	108	31	62	0	330.84
6	70	37	56	136	348.33
6	101	37	52	0	361.33
7	123	16	57	0	188.67
7	19	15	60	32	200.83
7	107	17	63	0	213.44
7	175	17	64	0	223.44
7	11	20	65	18	235.60
7	64	15	77	124	257.60
7	49	6	68	85	279.33
7	143	5	64	0	292.45
7	36	2	60	58	306.45
7	47	8	56	76	322.66
7	168	9	56	0	332.66
7	82	15	47	161	352.48
7	153	21	45	0	367.80
8	27	35	40	46	244.40
8	167	33	44	0	257.87
8	69	37	47	133	271.87
8	132	38	46	0	282.29
8	176	41	46	0	294.29
8	1	41	49	0	306.29
8	162	35	51	0	321.61
8	127	30	48	0	336.44
8	52	27	43	91	351.27
8	146	32	39	0	366.68
9	60	17	34	118	187.60
9	118	17	33	0	197.60
9	5	15	30	5	210.21
9	84	11	31	166	223.33
9	173	11	28	0	235.33
9	61	12	24	122	248.45
9	16	10	20	25	261.93
9	113	5	25	0	278.00
9	17	5	30	25	292.00
9	45	6	38	72	309.06
9	125	7	38	0	319.06
9	199	12	38	0	333.06
9	83	14	37	164	344.30
10	2	35	17	1	157.65
10	178	35	16	0	167.65
10	115	36	16	0	177.65
10	171	44	13	0	195.19
10	22	45	10	38	207.36
10	41	42	7	66	220.60
10	145	39	10	0	233.84
10	57	32	12	113	250.12
10	15	30	5	25	266.40
10	43	23	3	68	282.68
10	42	24	12	66	300.74
10	172	26	13	0	311.97
10	144	30	15	0	325.45
10	87	28	18	177	338.05
10	137	32	22	0	352.71
11	98	19	21	0	180.86
11	100	18	18	0	193.02
11	192	15	14	0	207.02
11	191	12	17	0	220.27
11	141	10	17	0	231.27
11	86	4	18	169	246.35
11	140	5	6	0	267.39
11	38	5	5	62	277.39
11	44	11	14	69	297.21
11	119	13	13	0	308.44
11	14	15	10	24	321.05
11	142	21	10	0	336.05
12	9	55	60	13	191.62
12	103	52	64	0	205.62
12	161	55	65	0	217.78
12	71	57	68	138	230.38
12	135	62	63	0	246.45
12	35	63	65	56	257.69
12	136	63	69	0	270.69
12	65	62	77	129	287.75
12	66	49	73	130	310.35
12	188	47	66	0	326.63
12	51	49	58	91	343.88
13	179	52	26	0	188.81
13	130	58	27	0	203.89
13	165	62	24	0	217.89
13	55	63	23	104	228.31
13	25	65	20	43	240.91
13	170	66	14	0	256.00
13	67	67	5	130	274.05
13	39	60	12	65	292.95
13	187	60	15	0	304.95
13	139	59	15	0	314.95
13	155	55	20	0	330.35
13	4	55	20	5	339.35
13	110	51	21	0	352.48
14	53	37	31	98	283.77
14	105	40	30	0	295.93
14	180	43	26	0	309.93
14	149	48	28	0	324.32
14	26	45	30	44	336.93
14	138	45	35	0	350.93
14	154	45	35	0	359.93
15	156	33	34	0	251.61
15	112	31	32	0	263.44
15	147	25	32	0	278.44
15	6	25	30	6	289.44
15	183	26	29	0	299.85
15	94	26	27	196	310.85
15	117	27	23	0	323.98
15	13	30	25	19	336.58
15	58	36	26	116	351.67
15	152	36	26	0	360.67
16	106	21	47	0	188.01
16	194	21	48	0	198.01
16	7	20	50	10	209.25
16	48	13	52	78	225.53
16	124	8	52	0	263.38
16	46	2	48	72	279.59
16	174	7	43	0	295.66
16	8	10	43	13	307.66
16	114	12	42	0	318.89
16	18	20	40	30	336.14
16	166	21	36	0	349.26
16	89	26	35	189	363.36
17	28	41	37	47	245.72
17	184	50	40	0	264.21
17	76	49	42	147	275.45
17	196	51	42	0	286.45
17	116	52	41	0	296.86
17	77	53	43	148	308.10
17	157	50	50	0	324.71
17	102	49	49	0	335.13
17	50	47	47	90	346.96
17	111	42	41	0	363.77



(a) Wizualizacja wyniku dla f71



(b) Wizualizacja wyniku dla f134

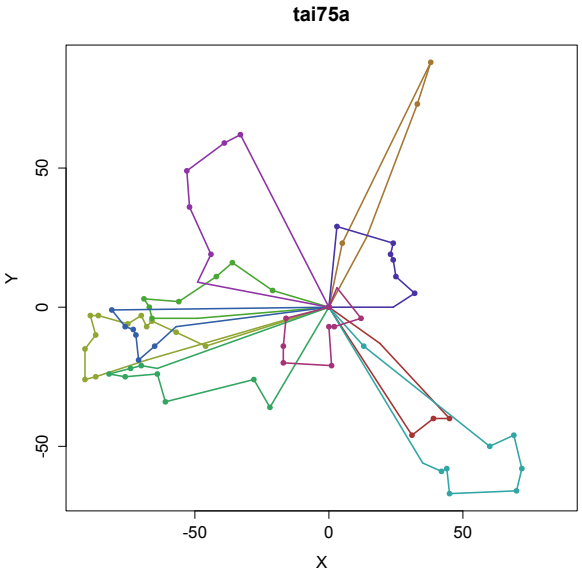
Rysunek B.3: Wizualizacje wyników

Tabela B.9: Najlepsze rozwiązanie uzyskane przez ContDVRP dla **f71**

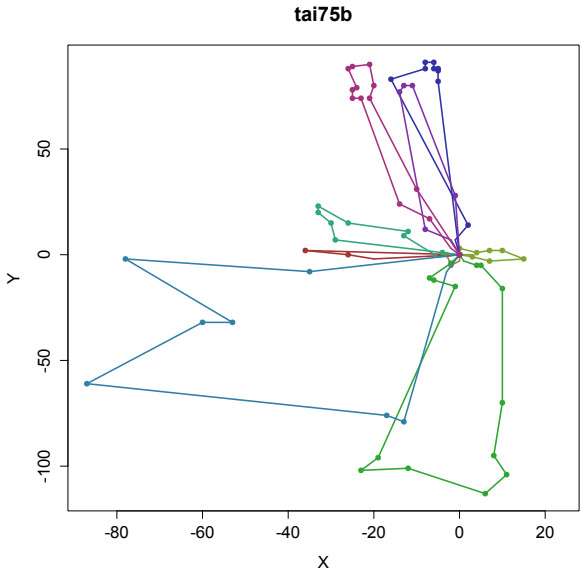
Pojazd	Zamówienie	X	Y	Otrzymane	Obsłużone
	Depot	0	0	0	211
1	9	7	-25	17	52.34
1	7	2	-22	12	63.17
1	4	2	-21	6	69.17
1	8	1	-21	16	75.17
1	3	-1	-18	5	83.77
1	5	-1	-17	7	90.40
1	10	-7	-17	18	101.40
1	6	-9	-12	8	111.79
1	71	-9	-11	0	117.79
1	12	-14	-9	26	128.17
1	16	-15	-9	37	134.17
1	17	-15	-8	38	140.17
1	13	-14	-8	29	146.17
1	1	-12	-6	2	154.00
1	15	-14	-5	35	161.23
1	2	-15	-5	4	167.23
1	19	-14	-4	43	173.65
1	14	-11	-2	35	182.25
2	60	-11	3	0	27.23
2	61	-11	4	0	33.23
2	58	-12	5	0	39.64
2	59	-12	6	0	45.64
2	63	-14	5	0	75.81
2	62	-14	4	0	81.81
2	64	-15	4	0	87.81
2	65	-15	5	0	93.81
2	66	-16	7	0	101.05
2	67	-16	8	0	107.05
2	69	-20	10	0	116.52
2	37	-20	12	83	123.52
2	38	-20	13	83	137.88
2	40	-20	15	102	144.88
2	68	-15	10	0	156.95
2	39	-12	10	95	164.95
2	57	-9	7	0	174.19
2	56	-9	6	0	180.19
2	34	-7	2	76	189.66
3	49	-4	18	0	39.54
3	50	-4	19	0	45.54
3	51	-5	18	0	64.44
3	70	-5	19	0	70.44
3	47	-2	20	0	78.60
3	48	-2	21	0	84.60
3	52	-5	26	0	95.43
3	45	-1	24	0	104.90
3	53	1	23	0	112.14
3	46	1	22	0	118.14
3	43	2	22	0	124.14
3	44	1	21	0	130.55
3	42	2	21	104	136.55
3	27	6	14	63	149.62
3	28	5	12	65	156.85
3	22	5	10	48	163.85
3	21	5	9	48	169.85
3	30	4	8	67	176.27
3	29	3	7	66	182.68
3	20	3	6	45	188.68
4	35	-7	-3	77	113.12
4	18	-9	-6	38	121.72
4	11	-11	-5	22	176.04
5	36	-7	-2	80	118.06
5	33	-8	-2	75	124.06
5	31	-6	1	72	132.66
5	32	-6	2	73	138.66
5	54	-6	5	0	146.66
5	55	-6	6	0	152.66
5	41	-6	8	103	159.66
5	25	1	10	54	171.94
5	24	1	9	54	177.94
5	26	2	9	58	183.94
5	23	2	8	52	189.94

Tabela B.10: Najlepsze rozwiązanie uzyskane przez ContDVRP dla **f134**

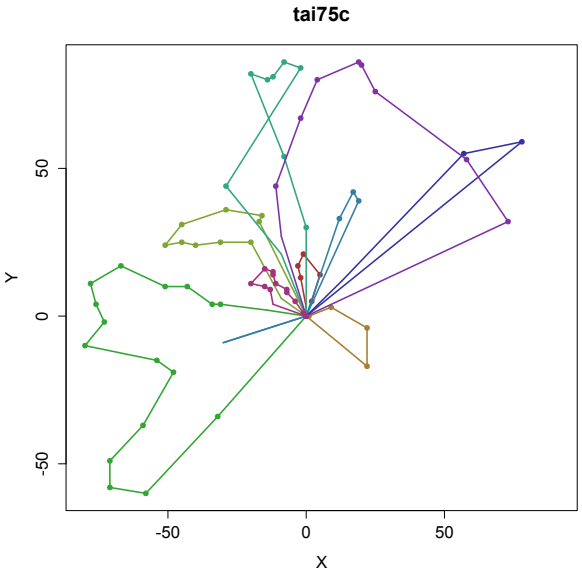
Pojazd	Zamówienie	X	Y	Otrzymane	Obsłużone
	Depot	-60	150	0	11741
1	78	-70	0	0	7488.46
1	133	-150	-40	0	7590.90
1	68	-250	-200	4650	7792.58
1	70	-240	-350	5063	7955.91
1	69	-250	-350	4975	7978.91
1	112	-780	-170	0	8551.65
1	125	-780	-175	0	9997.85
1	111	-780	-180	0	10015.85
1	110	-780	-190	0	10038.85
1	122	-790	-190	0	10061.85
1	123	-790	-185	0	10079.85
1	124	-790	-180	0	10097.85
1	126	-790	-170	0	10120.85
1	127	-800	-170	0	10143.85
1	121	-900	-220	0	10268.65
1	128	-800	-160	0	10398.27
1	129	-800	-150	0	10421.27
1	113	-800	-140	0	10444.27
1	81	-620	-100	0	10641.66
2	46	-140	160	3073	8886.37
2	118	-150	160	0	8909.37
2	17	-200	130	1171	8980.68
2	18	-210	140	1234	9007.82
2	132	-620	-90	0	9490.93
2	116	-850	140	0	9829.20
2	131	-850	150	0	9852.20
2	117	-780	150	0	9935.20
2	119	-620	320	0	10181.65
2	130	-480	370	0	10343.31
2	65	-400	490	4479	10500.53
2	19	-300	300	1278	10728.24
3	82	-80	300	0	8957.08
3	20	-50	300	1278	9000.08
3	83	10	600	0	9319.02
3	85	100	520	0	9452.43
3	84	100	520	0	9465.43
3	86	100	510	0	9488.43
3	87	160	290	0	9729.47
3	89	160	210	0	9822.47
3	90	155	192	0	9854.15
3	16	205	190	1040	9917.19
3	13	235	190	909	9960.19
3	15	250	200	1039	9991.22
3	88	260	210	0	10018.36
3	14	260	200	953	10041.36
3	11	283	143	672	10115.83
3	12	270	143	716	10141.83
3	10	265	117	587	10181.30
3	9	290	100	541	10224.54
3	8	300	105	453	10248.72
3	7	335	105	389	10296.72
3	6	310	80	345	10345.07
3	5	290	80	282	10378.07
3	4	278	83	259	10403.44
3	2	246	83	111	10448.44
3	42	230	40	2898	10507.32
3	41	228	31	2702	10529.54
3	3	233	13	130	10561.22
3	40	203	21	2702	10605.27
3	44	208	40	3030	10637.92
3	43	208	40	3006	10650.92
3	45	185	64	3049	10697.16
3	94	169	77	0	10730.77
3	93	165	78	0	10747.90
3	29	144	113	1969	10801.71
3	92	172	143	0	10855.75
3	28	114	145	1969	10926.78
3	27	98	166	1967	10966.19
3	26	70	150	1819	11011.43
3	25	48	170	1754	11054.17
3	21	13	178	1430	11103.07
3	91	0	165	0	11134.45
4	120	-1200	-200	0	8824.17
4	109	-1420	-310	0	9083.14
4	108	-1520	0	0	9421.87
4	107	-1520	10	0	9444.87
4	106	-1500	80	0	9530.67
4	114	-1180	220	0	9892.95
4	115	-1070	300	0	10041.97
5	66	-150	100	4543	10082.81
5	71	-180	100	5319	10125.81
5	33	-300	-100	2164	10372.04
5	80	-300	-110	0	10395.04
5	67	-110	-100	4648	10598.31
5	79	-30	-60	0	10700.75
5	63	-40	-40	4326	10736.11
5	64	-30	12	4327	10802.06
5	77	-30	20	5817	10823.06
5	76	-17	30	5751	10852.46
5	134	-10	32	0	10872.74
5	74	-30	50	5603	10912.65
5	73	-40	80	5538	10957.27
6	75	21	62	5729	9512.40
6	1	32	51	25	9540.96
6	62	72	40	4281	9595.44
6	50	87	28	3547	9627.65
6	51	90	33	3632	9646.49
6	53	112	33	3719	9681.49
6	102	118	30	0	9701.19
6	103	120	40	0	9724.39
6	104	128	36	0	9746.34
6	101	130	26	0	9769.53
6	35	145	10	2399	9804.47
6	36	150	18	2488	9826.90
6	37	172	24	2509	9862.70
6	95	180	20	0	9884.65
6	39	182	44	2598	9921.73
6	38	172	42	2550	9944.93
6	96	162	40	0	9968.13
6	97	150	40	0	9993.13
6	98	150	30	0	10016.13
6	99	148	24	0	10035.45
6	100	145	30	0	10055.16
6	105	134	55	0	10095.47
6	57	123	55	4043	10119.47
6	56	123	47	3874	10140.47
6	55	115	46	3828	10161.53
6	54	108	47	3762	10181.61
6	61	72	60	4194	10232.88
6	60	58	85	4155	10274.53
6	59	65	97	4066	10301.43
6	23	18	131	1495	10372.43
6	22	18	138	1492	10544.63
6	24	20	136	1666	10560.46
6	31	93	107	2098	10652.01
6	30	110	120	2096	10686.41
6	58	112	69	4066	10750.45
6	52	90	35	3700	10803.95
6	49	56	18	3354	10854.96
6	48	32	28	3265	10893.96
6	34	20	0	2273	10937.42
6	32	6	28	2142	10981.73
6	47	-5	69	3203	11037.18
6	72	-20	100	5386	11084.62



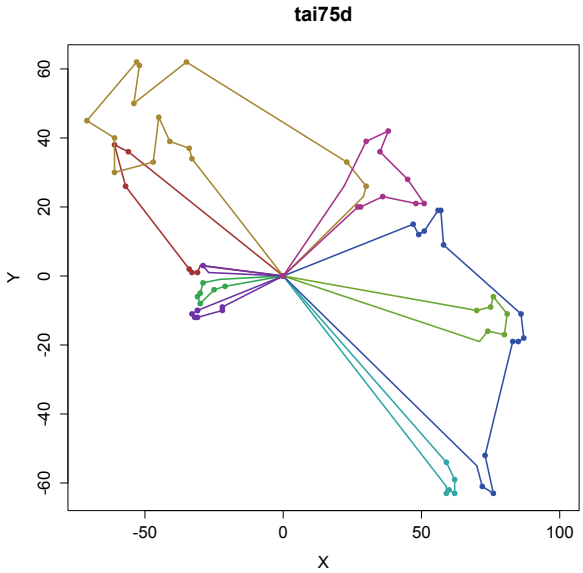
(a) Wizualizacja wyniku dla tai75a



(b) Wizualizacja wyniku dla tai75b



(c) Wizualizacja wyniku dla tai75c



(d) Wizualizacja wyniku dla tai75d

Rysunek B.4: Wizualizacje wyników

Tabela B.11: Najlepsze rozwiązanie uzyskane przez ContDVRP dla **tai75a**

Pojazd	Zamówienie	X	Y	Otrzymane	Obsłużone
	Depot	0	0	0	769
1	8	31	-46	82	516.87
1	5	39	-40	55	558.87
1	4	45	-40	52	596.87
1	16	19	-13	131	666.35
2	23	5	23	196	427.26
2	75	38	88	0	532.16
2	74	33	73	0	579.97
2	25	14	25	198	663.59
3	69	-46	-14	0	105.76
3	50	-57	-9	0	149.84
3	46	-66	-5	0	191.69
3	55	-68	-7	0	226.52
3	44	-70	-3	0	262.99
3	43	-75	-6	0	300.82
3	49	-86	-3	0	344.22
3	48	-89	-3	0	379.22
3	37	-87	-10	318	418.50
3	32	-91	-15	260	456.91
3	34	-91	-26	271	542.85
3	41	-87	-25	375	578.97
3	59	-68	-19	0	630.90
4	53	-21	6	0	21.84
4	62	-36	16	0	71.87
4	56	-42	11	0	111.68
4	63	-56	2	0	490.82
4	57	-69	3	0	535.86
4	64	-67	0	0	571.46
4	36	-66	-4	303	607.59
4	54	-49	-4	0	656.59
5	51	-22	-36	0	311.34
5	67	-28	-26	0	355.00
5	61	-61	-34	0	420.96
5	30	-64	-24	253	463.40
5	42	-76	-25	380	507.44
5	40	-82	-24	355	545.52
5	29	-74	-22	245	585.77
5	68	-70	-21	0	621.89
5	45	-64	-22	0	659.97
6	17	13	-14	133	269.03
6	6	60	-50	58	360.23
6	11	69	-46	91	402.08
6	2	72	-58	33	446.45
6	3	70	-66	50	486.70
6	10	45	-67	89	543.72
6	9	44	-58	83	584.77
6	7	42	-59	60	619.01
6	1	35	-56	6	658.62
7	39	-81	-1	352	446.28
7	35	-76	-7	274	486.09
7	38	-73	-8	349	521.25
7	47	-72	-10	0	555.49
7	31	-71	-19	259	596.55
7	33	-65	-14	266	636.36
7	60	-57	-7	0	678.99
8	15	3	29	130	182.95
8	19	24	23	148	236.80
8	21	23	19	185	516.75
8	27	24	17	236	550.98
8	18	25	11	140	589.07
8	24	32	5	196	630.29
8	12	24	0	94	671.72
9	71	-33	62	0	416.29
9	73	-39	59	0	454.99
9	72	-53	49	0	504.20
9	70	-52	36	0	549.24
9	52	-44	19	0	600.02
9	58	-49	9	0	643.21
10	26	-16	-4	229	381.77
10	65	-17	-14	0	423.82
10	66	-17	-20	0	461.82
10	14	1	-21	107	511.85
10	28	0	-7	243	557.88
10	22	2	-7	195	591.88
10	13	12	-4	104	634.32
10	20	3	7	174	680.53

Tabela B.12: Najlepsze rozwiązanie uzyskane przez ContDVRP dla **tai75b**

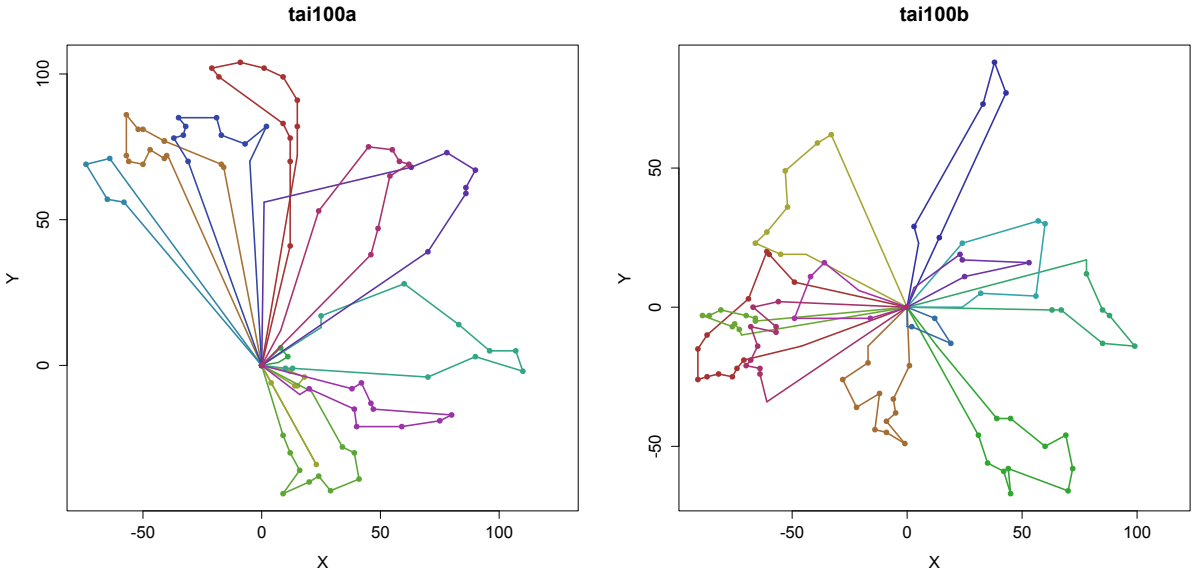
Pojazd	Zamówienie	X	Y	Otrzymane	Obsłużone
	Depot	0	0	0	905
1	43	-36	2	0	737.43
1	50	-26	0	0	773.63
1	48	-20	-2	0	805.95
2	7	-3	0	71	772.25
2	1	-2	-5	8	803.35
2	8	0	-3	96	832.18
3	71	0	3	0	613.88
3	9	4	1	98	644.35
3	15	7	2	153	673.51
3	72	10	2	0	702.51
3	75	15	-2	0	734.91
3	5	7	-3	64	768.97
3	2	3	-1	39	799.45
3	4	3	0	61	831.61
4	11	-2	-4	108	140.22
4	70	-7	-11	0	174.82
4	68	-6	-12	0	202.24
4	73	-1	-15	0	234.07
4	62	-19	-96	0	343.05
4	60	-23	-102	0	440.46
4	57	-12	-101	0	477.51
4	58	6	-113	0	525.14
4	61	11	-104	0	561.44
4	56	8	-95	0	596.92
4	55	10	-70	0	648.00
4	69	10	-16	0	728.00
4	12	5	-5	111	766.09
4	10	4	-5	105	793.09
4	74	1	-3	0	822.69
5	13	-4	1	123	547.12
5	44	-29	7	0	598.83
5	37	-30	15	374	632.90
5	49	-33	20	0	664.73
5	54	-33	23	0	693.73
5	42	-26	15	448	730.36
5	39	-12	11	415	770.92
5	51	-13	9	0	799.15
5	6	-5	-1	68	837.96
6	53	-35	-8	0	397.90
6	65	-78	-2	0	467.32
6	64	-53	-32	0	532.37
6	67	-60	-32	0	565.37
6	66	-87	-61	0	630.99
6	59	-17	-76	0	728.58
6	63	-13	-79	0	759.58
6	36	-3	-8	356	857.28
7	26	-5	82	269	398.90
7	24	-5	88	231	430.90
7	34	-5	87	320	457.90
7	17	-6	88	156	485.32
7	28	-6	91	286	514.32
7	35	-8	91	323	542.32
7	31	-8	88	305	662.50
7	18	-16	83	164	697.93
7	41	2	14	441	795.24
7	40	-1	7	418	828.86
8	45	-1	28	0	322.14
8	23	-11	80	231	401.10
8	16	-13	80	154	638.88
8	19	-14	77	174	668.04
8	52	-8	12	0	759.31
8	3	-2	7	59	825.69
9	47	-10	31	0	371.95
9	32	-21	74	306	442.33
9	27	-20	80	278	474.42
9	25	-21	90	233	510.47
9	29	-25	89	288	540.59
9	20	-26	88	205	593.04
9	21	-24	79	218	628.26
9	22	-25	78	229	655.67
9	30	-25	74	298	685.67
9	33	-23	74	313	713.67
9	38	-14	24	411	790.48
9	46	-7	17	0	826.38
9	14	-2	3	126	867.24

Tabela B.13: Najlepsze rozwiązanie uzyskane przez ContDVRP dla **tai75c**

Pojazd	Zamówienie	X	Y	Otrzymane	Obsłużone
	Depot	0	0	0	782
1	7	2	5	60	552.79
1	60	5	14	0	587.27
1	56	-1	21	0	621.49
1	65	-3	17	0	650.96
1	4	-2	13	34	680.09
1	5	0	3	43	715.28
2	11	1	0	87	548.40
2	59	9	3	0	581.94
2	72	22	-4	0	621.71
2	74	22	-17	0	659.71
2	73	13	-10	0	696.11
3	47	-17	32	0	407.69
3	45	-16	34	0	434.92
3	18	-29	36	162	473.07
3	17	-45	31	151	514.84
3	25	-51	24	229	549.06
3	28	-45	25	286	580.14
3	12	-40	24	102	610.24
3	20	-31	25	182	644.29
3	52	-20	25	0	680.29
3	6	-9	6	53	727.25
4	68	-32	-34	0	46.69
4	66	-58	-60	0	108.46
4	67	-71	-58	0	214.10
4	71	-71	-49	0	248.10
4	69	-59	-37	0	290.07
4	70	-48	-19	0	336.17
4	13	-54	-15	113	368.38
4	26	-80	-10	240	419.86
4	15	-73	-2	145	455.49
4	16	-76	4	149	487.19
4	19	-78	11	164	519.47
4	21	-67	17	191	557.00
4	23	-51	10	214	599.47
4	14	-43	10	121	632.47
4	27	-34	4	269	668.29
4	22	-31	4	210	696.29
4	10	-14	2	86	738.40
5	63	0	30	0	362.35
5	44	-8	54	0	412.65
5	49	-20	82	0	468.11
5	46	-14	80	0	499.44
5	34	-12	81	318	526.67
5	35	-8	86	326	558.08
5	43	-2	84	0	589.40
5	41	-29	44	378	662.66
5	58	-9	21	0	718.14
6	48	19	39	0	551.68
6	50	17	42	0	580.29
6	36	12	33	332	615.58
6	24	-30	-9	223	707.02
7	30	78	59	295	567.00
7	32	57	55	305	613.38
7	75	13	12	0	699.90
8	29	73	32	286	372.96
8	31	58	53	296	423.76
8	40	25	76	361	488.99
8	33	20	85	312	524.28
8	37	19	86	340	550.70
8	42	4	80	387	591.85
8	39	-2	67	352	631.17
8	38	-11	44	344	680.87
8	61	-9	27	0	722.99
9	57	-1	1	0	275.11
9	51	-4	5	0	305.11
9	2	-7	8	1	334.36
9	1	-7	9	1	360.36
9	54	-11	11	0	420.47
9	53	-12	14	0	448.63
9	64	-12	15	0	474.63
9	3	-15	16	28	556.01
9	62	-20	11	0	588.08
9	9	-15	10	74	618.18
9	8	-13	9	67	672.39
9	55	-12	4	0	702.49

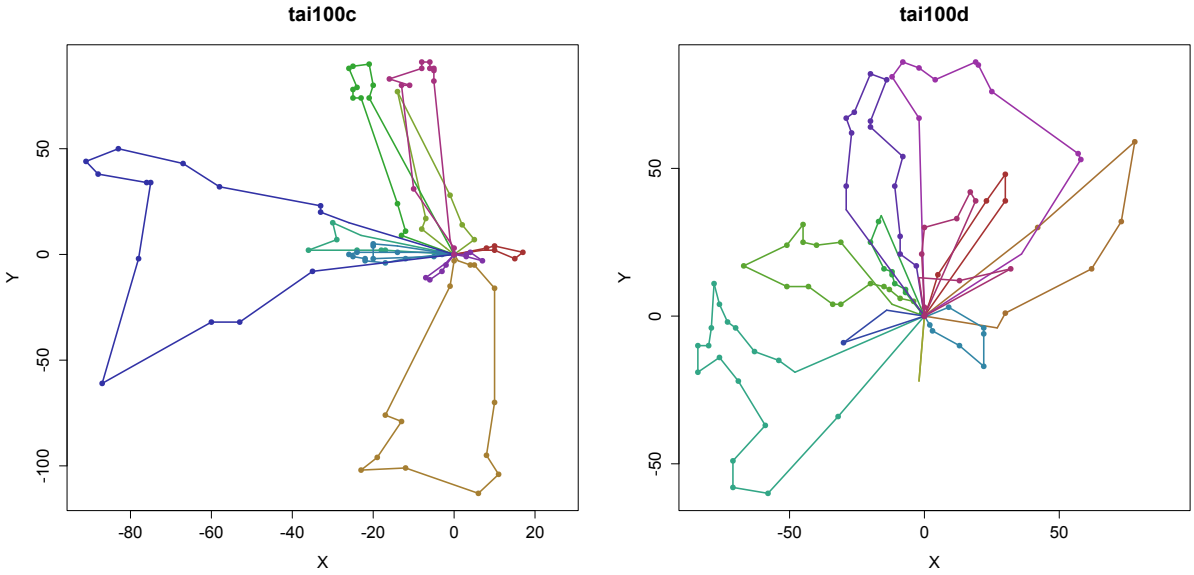
Tabela B.14: Najlepsze rozwiązanie uzyskane przez ContDVRP dla **tai75d**

Pojazd	Zamówienie	X	Y	Otrzymane	Obsłużone
	Depot	0	0	0	789
1	27	-56	36	373	480.80
1	25	-61	38	341	513.18
1	28	-57	26	374	552.83
1	51	-34	2	0	613.07
1	59	-33	1	0	641.49
1	45	-31	1	0	670.49
1	43	-30	3	0	699.72
2	37	-33	34	0	126.28
2	34	-34	37	0	156.44
2	30	-41	39	0	190.72
2	32	-45	46	0	225.79
2	35	-47	33	0	276.85
2	31	-61	30	0	318.17
2	29	-61	40	0	355.17
2	39	-71	45	0	393.35
2	33	-53	62	0	445.11
2	36	-52	61	0	473.52
2	26	-54	50	371	511.70
2	38	-35	62	0	561.18
2	13	23	33	190	653.02
2	19	30	26	274	689.92
2	20	29	23	294	720.08
3	11	70	-10	156	445.49
3	6	75	-9	37	477.58
3	7	76	-6	63	507.75
3	3	81	-11	20	541.82
3	4	80	-17	21	574.90
3	5	74	-16	23	607.98
3	2	71	-19	14	639.23
4	58	-21	-3	0	277.64
4	48	-25	-4	0	308.76
4	55	-30	-8	0	342.16
4	56	-31	-6	0	371.40
4	50	-30	-5	0	399.81
4	41	-29	-2	0	429.98
4	57	-23	-1	0	703.73
5	66	59	-54	0	494.21
5	68	62	-59	0	527.04
5	65	62	-63	0	558.04
5	69	60	-62	0	587.27
5	64	59	-63	0	615.69
5	60	59	-61	0	644.69
6	71	47	15	0	187.41
6	72	49	12	0	218.02
6	74	51	13	0	247.25
6	70	56	19	0	282.06
6	75	57	19	0	310.06
6	73	58	9	0	347.11
6	8	86	-11	92	408.52
6	1	87	-18	5	442.59
6	9	85	-19	101	471.83
6	10	83	-19	142	500.83
6	61	73	-52	0	562.31
6	63	76	-63	0	600.71
6	62	72	-61	0	632.18
6	67	70	-55	0	665.51
7	47	-22	-9	0	181.57
7	40	-22	-10	0	209.57
7	52	-31	-12	0	245.79
7	49	-32	-12	0	501.40
7	44	-33	-11	0	529.81
7	53	-31	-10	0	559.05
7	46	-31	-10	0	586.05
7	42	-29	3	0	674.78
7	54	-27	1	0	704.61
8	21	27	20	299	368.93
8	23	28	20	312	396.93
8	22	36	23	306	432.47
8	12	48	21	177	471.64
8	17	51	21	247	501.64
8	24	45	28	322	537.85
8	14	35	36	239	577.66
8	15	38	42	245	611.37
8	18	30	39	254	646.91
8	16	22	26	247	689.18



(a) Wizualizacja wyniku dla tai100a

(b) Wizualizacja wyniku dla tai100b



(c) Wizualizacja wyniku dla tai100c

(d) Wizualizacja wyniku dla tai100d

Rysunek B.5: Wizualizacje wyników

Tabela B.15: Najlepsze rozwiązanie uzyskane przez ContDVRP dla **tai100a**

Pojazd	Zamówienie	X	Y	Otrzymane	Obsłużone
	Depot	0	0	0	897
1	99	12	41	0	199.70
1	16	12	70	125	258.70
1	9	12	78	94	329.52
1	12	9	83	110	365.36
1	17	-18	99	128	426.74
1	6	-21	102	89	505.17
1	18	-9	104	141	547.33
1	10	1	102	100	587.53
1	4	9	99	56	626.08
1	11	15	91	105	666.08
1	5	15	82	70	705.08
1	8	15	72	92	745.08
2	7	-16	68	90	294.11
2	2	-17	69	21	325.52
2	79	-41	77	0	380.82
2	84	-41	77	0	410.82
2	83	-50	81	0	450.67
2	82	-52	81	0	482.67
2	92	-57	86	0	519.74
2	87	-57	72	0	563.74
2	89	-56	70	0	595.98
2	91	-50	69	0	632.06
2	88	-47	74	0	667.89
2	86	-41	71	0	704.60
2	78	-40	72	0	736.01
2	77	-39	71	0	767.43
3	55	12	-2	437	527.94
3	63	18	-4	0	564.27
3	61	15	-7	0	598.51
3	56	14	-7	447	629.51
3	62	4	-6	0	682.37
3	34	23	-34	275	746.21
3	20	23	-34	155	776.21
4	19	9	-24	152	362.01
4	22	12	-30	170	398.72
4	23	16	-36	203	435.93
4	21	9	-44	156	511.56
4	36	20	-40	298	553.26
4	29	24	-38	242	587.73
4	27	29	-43	229	624.80
4	32	41	-39	256	667.45
4	33	39	-30	267	706.67
4	31	34	-28	249	742.06
4	37	21	-9	300	795.08
5	54	8	6	422	750.02
5	60	11	3	0	784.27
5	53	7	1	406	818.74
6	64	10	-1	0	211.87
6	57	13	-1	0	244.87
6	70	70	-4	0	331.95
6	72	90	3	0	387.56
6	71	110	-2	0	438.18
6	69	107	5	0	508.54
6	66	96	5	0	549.54
6	75	83	14	0	595.35
6	67	60	28	0	684.83
6	52	25	17	394	751.51
6	59	25	13	0	785.51
7	95	-58	56	0	103.05
7	97	-65	57	0	620.12
7	96	-74	69	0	665.12
7	94	-64	71	0	705.32
7	76	-56	62	0	747.36
8	81	-31	70	0	323.23
8	80	-37	78	0	363.23
8	93	-33	79	0	482.62
8	90	-32	82	0	515.79
8	85	-35	85	0	550.03
8	3	-19	85	21	596.03
8	15	-17	79	123	632.35
8	14	-7	76	110	672.79
8	13	2	82	110	713.61
8	1	-5	70	1	757.50
9	40	70	39	317	438.93
9	45	86	59	353	504.11
9	44	86	61	351	536.11
9	39	90	67	317	573.32
9	41	78	73	322	616.74
9	48	63	68	358	662.55
9	98	1	56	0	755.70
10	35	38	-8	293	420.06
10	26	42	-6	218	454.53
10	38	46	-13	300	492.59
10	30	47	-15	247	524.83
10	73	80	-17	0	587.89
10	68	75	-19	0	623.27
10	74	59	-21	0	669.40
10	24	40	-21	208	718.40
10	25	39	-15	209	754.48
10	28	20	-8	236	804.73
10	65	16	-10	0	839.20
11	47	46	38	358	440.89
11	42	49	47	341	480.38
11	46	54	65	353	542.03
11	50	62	69	378	580.98
11	51	58	70	379	615.10
11	49	55	74	374	650.10
11	43	45	75	342	690.15
11	100	24	53	0	750.56
11	58	8	12	0	824.57

Tabela B.16: Najlepsze rozwiązanie uzyskane przez ContDVRP dla **tai100b**

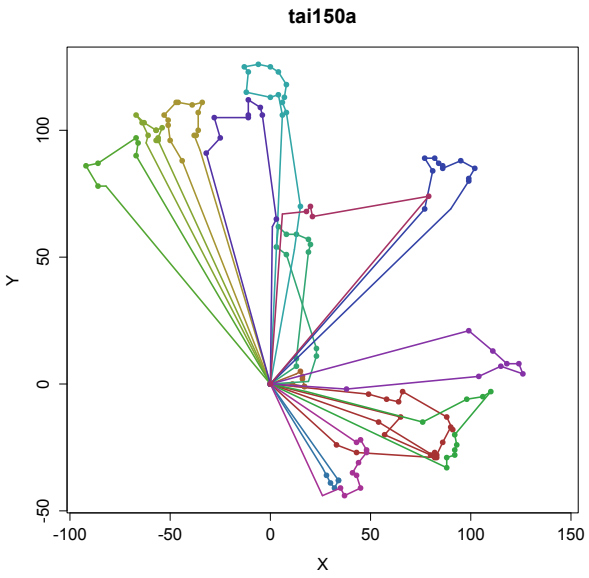
Pojazd	Zamówienie	X	Y	Otrzymane	Obsłużone
	Depot	0	0	0	799
1	58	-49	9	0	229.59
1	77	-60	19	0	273.46
1	80	-61	20	0	303.87
1	57	-69	3	0	351.66
1	37	-87	-10	267	402.87
1	32	-91	-15	228	438.27
1	34	-91	-26	245	478.27
1	41	-87	-25	286	511.39
1	40	-82	-24	282	545.49
1	42	-76	-25	304	580.57
1	29	-74	-22	216	613.18
1	31	-71	-19	222	646.42
1	69	-46	-14	0	700.92
2	14	1	-21	98	320.65
2	86	-6	-33	0	363.54
2	82	-5	-38	0	397.64
2	83	-9	-41	0	431.64
2	87	-1	-49	0	471.95
2	85	-9	-45	0	509.90
2	88	-14	-44	0	544.00
2	84	-12	-31	0	586.15
2	51	-22	-36	338	626.33
2	67	-28	-26	0	666.99
2	66	-17	-20	0	708.52
2	65	-17	-14	0	743.52
3	71	-33	62	0	429.79
3	73	-39	59	0	465.49
3	72	-53	49	0	511.70
3	70	-52	36	0	553.74
3	78	-61	27	0	595.46
3	79	-66	23	0	630.87
3	81	-55	19	0	671.57
3	52	-44	19	351	711.57
4	46	-66	-5	314	385.79
4	36	-66	-4	266	415.79
4	44	-70	-3	313	448.91
4	39	-81	-1	282	489.09
4	49	-86	-3	333	523.48
4	48	-89	-3	319	555.48
4	35	-76	-7	261	598.08
4	43	-75	-6	304	628.49
4	38	-73	-8	267	660.32
4	47	-72	-10	319	691.56
5	8	31	-46	82	275.20
5	1	35	-56	1	314.97
5	7	42	-59	80	351.58
5	10	45	-67	89	389.13
5	9	44	-58	83	427.18
5	3	70	-66	19	483.38
5	2	72	-58	19	520.63
5	11	69	-46	94	562.00
5	6	60	-50	79	600.85
5	4	45	-40	50	647.88
5	5	39	-40	63	682.88
5	17	13	-14	114	748.65
6	94	63	-1	0	362.63
6	90	67	-1	0	395.63
6	96	85	-13	0	446.27
6	99	99	-14	0	489.30
6	95	88	-3	0	533.86
6	89	85	-1	0	566.46
6	92	78	12	0	610.23
6	97	78	17	0	644.23
7	19	24	23	135	452.72
7	93	57	31	0	515.67
7	91	60	30	0	547.83
7	98	56	4	0	603.14
7	24	32	5	185	656.16
7	12	24	0	98	694.60
8	13	12	-4	98	611.90
8	16	19	-13	111	652.30
8	22	2	-7	151	699.33
8	28	0	-7	210	730.33
9	25	14	25	186	408.18
9	76	43	77	0	496.72
9	75	38	88	0	537.80
9	74	33	73	0	582.61
9	15	3	29	110	664.87
9	23	5	23	180	700.19
10	18	25	11	126	526.69
10	100	53	16	0	584.13
10	27	24	17	204	642.15
10	21	23	19	139	673.38
10	20	3	7	138	725.71
11	26	-16	-4	194	515.87
11	54	-49	-4	376	577.87
11	56	-42	11	398	623.42
11	62	-36	16	0	660.23
11	53	-21	6	361	707.26
12	63	-56	2	0	335.69
12	64	-67	0	0	375.87
12	60	-57	-7	0	417.07
12	50	-57	-9	336	448.07
12	55	-68	-7	389	488.25
12	33	-65	-14	238	524.87
12	59	-68	-19	0	559.70
12	68	-70	-21	0	591.53
12	45	-64	-22	314	626.61
12	30	-64	-24	220	657.61
12	61	-61	-34	0	697.05

Tabela B.17: Najlepsze rozwiązanie uzyskane przez ContDVRP dla **tai100c**

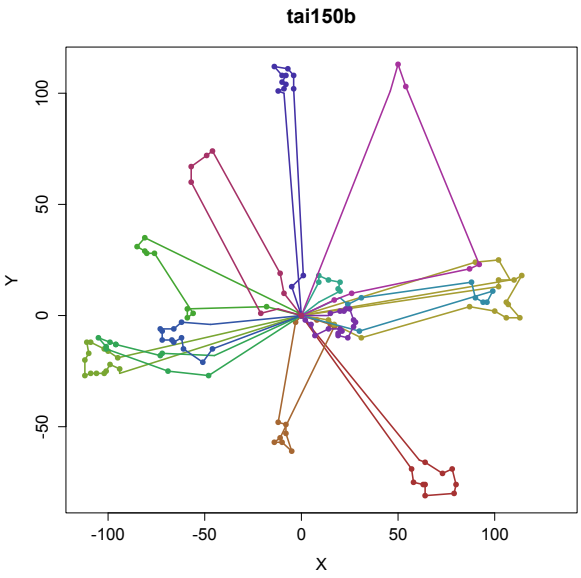
Pojazd	Zamówienie	X	Y	Otrzymane	Obsłużone
	Depot	0	0	0	905
1	77	8	3	0	709.92
1	76	10	4	0	733.16
1	99	17	1	0	761.77
1	75	15	-2	0	786.38
1	72	10	2	0	813.78
1	15	7	2	151	837.78
2	8	0	-3	61	251.88
2	73	-1	-15	0	284.92
2	59	-17	-76	0	368.98
2	63	-13	-79	0	433.25
2	62	-19	-96	0	472.28
2	60	-23	-102	0	500.49
2	57	-12	-101	0	532.53
2	58	6	-113	0	575.17
2	61	11	-104	0	606.46
2	56	8	-95	0	636.95
2	55	10	-70	0	683.03
2	69	10	-16	0	758.03
2	12	5	-5	123	791.11
2	10	4	-5	96	813.11
2	74	1	-3	0	837.72
3	100	5	7	0	506.35
3	41	2	14	420	534.97
3	45	-1	28	443	570.29
3	19	-14	77	188	641.98
3	46	-7	17	450	723.39
3	52	-8	12	0	795.35
3	14	-2	3	140	827.17
4	51	-13	9	0	445.69
4	39	-12	11	410	468.92
4	38	-14	24	391	503.08
4	33	-23	74	355	574.88
4	30	-25	74	326	597.88
4	22	-25	78	198	622.88
4	21	-24	79	193	645.29
4	20	-26	88	188	675.51
4	29	-25	89	315	697.93
4	25	-21	90	233	723.05
4	27	-20	80	290	754.10
4	32	-21	74	351	781.18
4	3	-2	7	28	871.82
5	89	-17	2	0	650.62
5	93	-18	2	0	672.62
5	90	-24	2	0	699.62
5	43	-36	2	435	732.62
5	44	-29	7	440	762.22
5	37	-30	15	370	791.28
5	96	-23	9	0	821.50
6	7	-3	0	56	432.88
6	91	-12	-2	0	463.09
6	87	-17	-4	0	489.48
6	97	-22	-3	0	515.58
6	98	-22	-2	0	537.58
6	88	-25	-1	0	561.74
6	50	-26	0	0	584.16
6	85	-24	1	0	607.39
6	92	-14	1	0	638.39
6	13	-4	1	131	669.39
6	6	-5	-1	40	699.61
6	48	-20	-2	0	735.65
6	86	-20	4	0	762.65
6	94	-20	5	0	784.65
6	95	-14	4	0	811.73
7	53	-35	-8	0	194.28
7	64	-53	-32	0	245.28
7	67	-60	-32	0	273.28
7	66	-87	-61	0	333.90
7	65	-78	-2	0	414.58
7	83	-75	34	0	509.62
7	80	-76	34	0	531.62
7	79	-88	38	0	565.27
7	84	-91	44	0	592.98
7	81	-83	50	0	623.98
7	82	-67	43	0	662.45
7	78	-58	32	0	697.66
7	54	-33	23	0	745.23
7	49	-33	20	0	769.23
7	42	-26	15	425	798.83
8	2	3	-1	21	636.66
8	5	7	-3	33	662.13
8	9	4	1	65	688.13
8	4	3	0	33	710.55
8	1	-2	-5	16	739.93
8	36	-3	-8	368	764.10
8	68	-6	-12	0	790.10
8	70	-7	-11	0	812.51
8	11	-2	-4	123	842.11
9	71	0	3	0	206.62
9	47	-10	31	0	257.36
9	16	-13	80	151	327.45
9	23	-11	80	218	566.00
9	18	-16	83	170	592.83
9	31	-8	88	330	623.26
9	35	-8	91	361	647.26
9	28	-6	91	303	670.26
9	17	-6	88	166	694.26
9	24	-5	88	233	716.26
9	34	-5	87	360	738.26
9	26	-5	82	285	764.26
9	40	-1	7	413	860.37

Tabela B.18: Najlepsze rozwiązanie uzyskane przez ContDVRP dla **tai100d**

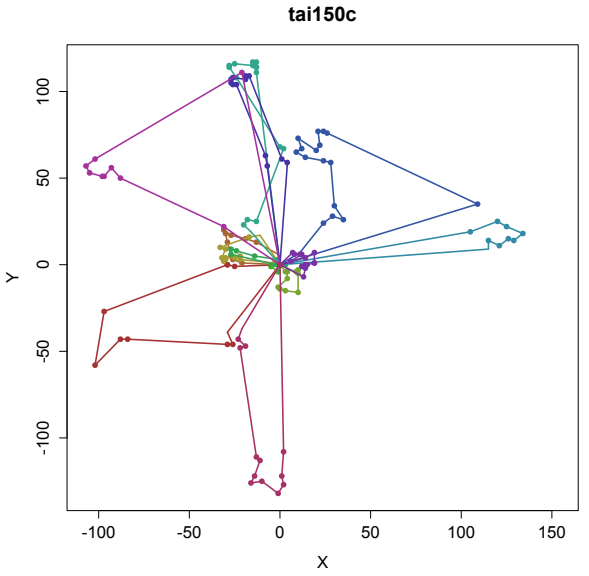
Pojazd	Zamówienie	X	Y	Otrzymane	Obsłużone
	Depot	0	0	0	782
1	94	30	39	0	557.50
1	96	30	48	0	589.50
1	95	23	39	0	623.91
1	60	5	14	0	677.71
1	7	2	5	39	710.20
2	99	42	30	0	442.61
2	30	78	59	175	511.84
2	29	73	32	175	562.30
2	100	62	16	0	604.72
2	80	30	1	0	663.06
2	77	27	-4	0	691.89
3	79	-2	-22	0	686.79
4	51	-4	5	0	201.90
4	6	-9	6	36	230.00
4	8	-13	9	40	258.00
4	9	-15	10	50	283.24
4	62	-20	11	0	311.34
4	22	-31	4	132	347.38
4	27	-34	4	174	373.38
4	14	-43	10	87	424.82
4	23	-51	10	138	455.82
4	21	-67	17	128	496.28
4	25	-51	24	156	536.75
4	17	-45	31	111	568.96
4	28	-45	25	174	597.96
4	12	-40	24	82	626.06
4	20	-31	25	128	658.12
4	55	-12	4	0	709.44
5	57	-1	1	0	372.86
5	2	-7	8	14	405.08
5	1	-7	9	7	429.08
5	54	-11	11	0	456.56
5	53	-12	14	0	482.72
5	64	-12	15	0	506.72
5	3	-15	16	23	593.11
5	52	-20	25	0	626.41
5	47	-17	32	354	657.02
5	45	-16	34	327	682.26
6	68	-32	-34	0	85.79
6	66	-58	-60	0	145.56
6	67	-71	-58	0	181.71
6	71	-71	-49	0	213.71
6	69	-59	-37	0	253.68
6	88	-69	-22	0	294.71
6	92	-76	-14	0	328.34
6	90	-84	-19	0	360.78
6	87	-84	-10	0	392.78
6	26	-80	-10	160	419.78
6	89	-79	-4	0	448.86
6	19	-78	11	128	486.89
6	16	-76	4	109	517.17
6	15	-73	-2	89	546.88
6	93	-70	-4	0	573.49
6	91	-63	-12	0	607.12
6	13	-54	-15	82	639.60
6	70	-48	-19	0	669.81
7	78	2	-3	0	492.36
7	76	3	-5	0	517.59
7	73	13	-10	0	551.77
7	74	22	-17	0	586.17
7	81	22	-6	0	620.17
7	72	22	-4	0	645.17
7	59	9	3	0	682.94
7	11	1	0	73	714.48
8	24	-30	-9	145	656.92
8	10	-14	2	63	699.34
9	65	-3	17	0	134.56
9	58	-9	21	0	164.77
9	61	-9	27	0	193.77
9	38	-11	44	217	372.47
9	44	-8	54	326	405.91
9	83	-20	64	0	444.53
9	85	-20	66	0	469.53
9	46	-14	80	347	507.76
9	49	-20	82	378	537.08
9	84	-26	69	0	574.40
9	86	-29	67	0	601.01
9	82	-27	62	0	629.39
9	41	-29	44	255	670.50
9	18	-29	36	115	701.50
10	39	-2	67	233	340.73
10	34	-12	81	206	380.93
10	35	-8	86	210	410.34
10	43	-2	84	295	439.66
10	42	4	80	289	469.87
10	37	19	86	213	509.03
10	33	20	85	204	533.44
10	40	25	76	253	566.74
10	32	57	55	188	628.01
10	31	58	53	183	653.25
10	97	36	21	0	715.08
11	48	19	39	364	414.83
11	50	17	42	0	441.44
11	36	12	33	213	474.73
11	63	0	30	0	510.10
11	56	-1	21	0	542.16
11	5	0	3	27	583.19
11	98	32	16	0	644.96
11	75	13	12	0	687.38
11	4	-2	13	25	725.41



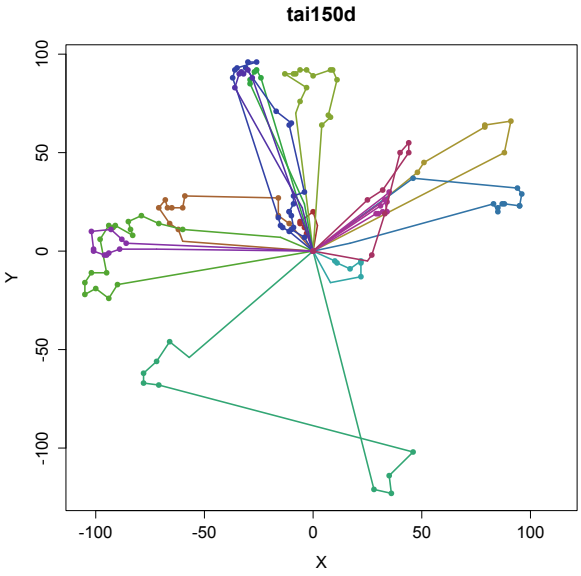
(a) Wizualizacja wyniku dla tai150a



(b) Wizualizacja wyniku dla tai150b



(c) Wizualizacja wyniku dla tai150c



(d) Wizualizacja wyniku dla tai150d

Rysunek B.6: Wizualizacje wyników

Tabela B.19: Najlepsze rozwiązanie uzyskane przez ContDVRP dla **tai150a**

Pojazd	Zamówienie	X	Y	Otrzymane	Obsłużone
	Depot	0	0	0	1062
1	130	49	-4	0	128.81
1	131	58	-6	0	168.03
1	132	64	-7	0	204.12
1	134	66	-3	0	238.59
1	127	88	-13	0	292.75
1	117	91	-18	0	328.58
1	129	90	-17	0	360.00
1	122	86	-23	0	397.21
1	121	83	-28	0	433.04
1	126	82	-27	0	464.45
1	116	80	-28	0	496.69
1	136	57	-20	0	551.04
1	135	65	-13	0	591.67
1	133	54	-15	0	744.00
1	128	83	-29	0	806.21
1	124	82	-29	0	837.21
1	25	43	-27	152	906.26
1	37	33	-24	240	946.70
1	28	32	-23	177	978.11
2	62	11	0	471	727.85
2	56	17	-1	402	763.93
2	61	17	-1	467	793.93
2	55	16	2	381	827.10
2	57	16	3	426	907.15
2	60	15	5	445	939.39
2	53	13	4	369	971.62
3	138	-44	88	0	98.39
3	81	-50	96	0	138.39
3	93	-51	102	0	487.43
3	90	-51	104	0	519.43
3	85	-53	106	0	552.26
3	143	-47	111	0	590.07
3	148	-46	111	0	621.07
3	141	-39	110	0	658.14
3	142	-34	111	0	693.24
3	149	-36	107	0	727.71
3	139	-36	100	0	764.71
3	140	-37	98	0	796.95
3	145	-38	98	0	827.95
3	150	-35	92	0	864.66
4	77	-56	96	0	535.94
4	86	-57	96	0	566.94
4	78	-56	97	0	598.35
4	80	-54	101	0	632.83
4	84	-57	100	0	665.99
4	79	-57	100	0	695.99
4	83	-63	103	0	732.70
4	92	-67	106	0	767.70
4	82	-64	103	0	801.94
4	88	-61	98	0	837.77
4	91	-62	95	0	870.93
5	76	-67	90	0	643.20
5	89	-66	95	0	678.30
5	87	-67	97	0	710.54
5	94	-86	87	0	762.01
5	96	-92	86	0	798.09
5	97	-86	78	0	838.09
5	95	-82	78	0	872.09
6	120	88	-33	0	93.98
6	125	88	-29	0	127.98
6	114	92	-28	0	512.02
6	115	92	-26	0	544.02
6	118	93	-24	0	576.26
6	123	92	-20	0	610.38
6	73	110	-3	0	745.06
6	68	106	-5	498	779.53
6	74	98	-6	0	817.59
6	119	76	-15	0	871.36
6	65	18	-3	484	960.59
7	54	13	7	375	466.11
7	58	13	10	432	499.11
7	113	19	52	0	571.54
7	106	20	55	0	604.70
7	104	19	57	0	636.94
7	99	13	59	0	673.26
7	101	8	59	0	708.26
7	112	4	62	0	743.26
7	108	3	54	0	781.33
7	107	8	51	0	817.16
7	52	23	14	369	887.08
7	59	23	11	438	920.08
7	63	19	1	471	960.85
8	16	6	106	101	424.77
8	12	4	114	74	463.02
8	13	0	113	76	497.14
8	3	-12	115	11	539.30
8	17	-11	123	105	577.37
8	6	-13	125	40	610.20
8	18	-6	126	115	647.27
8	10	0	125	68	683.35
8	4	4	123	15	717.82
8	11	8	118	70	754.22
8	5	7	113	33	789.32
8	9	6	111	62	821.56
8	8	8	107	49	856.03
8	105	15	70	0	923.69
8	110	12	53	0	970.95
9	34	34	-38	207	767.84
9	20	34	-38	137	797.84
9	36	32	-41	234	831.45
9	23	30	-39	142	864.27
9	22	28	-36	140	897.88
9	19	26	-33	119	931.49
10	47	77	69	347	475.09
10	46	81	84	344	520.62
10	43	77	89	287	557.02
10	49	82	89	365	592.02
10	51	84	87	367	624.85
10	50	86	86	365	657.08
10	48	86	85	355	688.08
10	41	95	88	275	727.57
10	39	102	85	265	765.19
10	44	99	81	297	800.19
10	45	99	80	322	831.19
10	40	90	69	275	875.40
11	146	-32	91	0	335.41
11	147	-25	97	0	374.63
11	144	-28	105	0	622.64
11	7	-11	105	42	669.64
11	2	-11	106	9	700.64
11	15	-11	112	91	736.64
11	14	-5	109	86	773.35
11	1	-4	106	4	806.51
11	111	3	65	0	878.11
11	103	1	62	0	911.71
12	137	38	-2	0	542.50
12	70	104	3	0	638.69
12	72	115	7	0	680.40
12	71	126	4	0	721.80
12	69	124	8	508	756.27
12	66	118	8	496	792.27
12	75	111	13	0	830.87
12	67	99	21	498	875.29
12	64	14	3	476	992.18
13	35	43	-23	222	393.91
13	26	45	-22	154	426.15
13	38	48	-26	242	645.65
13	30	48	-27	191	676.65
13	24	44	-31	152	712.31
13	31	41	-35	203	747.31
13	33	43	-36	205	779.54
13	32	45	-41	203	814.93
13	27	37	-44	175	853.47
13	29	35	-41	185	887.08
13	21	26	-44	138	926.56
14	42	79	74	277	453.40
14	100	21	66	0	778.85
14	102	20	70	0	812.97
14	109	18	68	0	855.88
14	98	6	67	0	897.92

Tabela B.20: Najlepsze rozwiązanie uzyskane przez ContDVRP dla **tai150b**

Pojazd	Zamówienie	X	Y	Otrzymane	Obsłużone
	Depot	0	0	0	988
1	8	57	-69	45	484.70
1	1	58	-75	3	517.78
1	7	63	-76	34	549.88
1	9	64	-76	51	577.88
1	10	64	-81	74	609.88
1	3	79	-80	7	651.91
1	2	80	-76	5	683.04
1	11	78	-69	74	717.32
1	6	73	-71	31	749.70
1	4	64	-66	18	787.00
1	5	61	-65	23	817.16
2	148	-3	-3	0	275.94
2	84	-12	-48	0	348.83
2	86	-8	-49	0	379.96
2	82	-8	-53	0	500.30
2	87	-5	-61	0	535.84
2	85	-10	-57	0	569.25
2	88	-14	-57	0	697.90
2	83	-11	-55	0	728.51
2	16	17	-5	100	812.81
2	120	15	-4	0	842.05
2	22	8	-2	131	876.33
2	28	7	-2	149	904.33
3	127	14	-2	0	63.54
3	136	17	-4	0	94.15
3	137	31	-10	0	136.38
3	106	87	4	0	221.10
3	102	100	2	0	261.26
3	96	106	-1	0	294.96
3	99	113	-1	0	328.96
3	95	107	5	0	364.45
3	89	106	6	0	392.86
3	104	114	18	0	434.29
3	107	110	16	0	465.76
3	97	102	16	0	500.76
3	92	102	13	0	530.76
3	93	90	24	0	753.73
3	108	102	25	0	792.77
3	103	108	16	0	830.59
4	33	-95	-19	207	417.98
4	47	-100	-16	298	450.81
4	35	-102	-15	211	480.05
4	49	-109	-12	300	514.66
4	48	-111	-12	300	543.66
4	37	-110	-17	218	575.76
4	32	-112	-20	180	606.37
4	34	-112	-27	207	640.37
4	41	-109	-26	247	670.53
4	40	-106	-26	244	700.53
4	42	-102	-26	258	731.53
4	29	-101	-25	154	759.95
4	31	-99	-22	174	790.55
4	45	-94	-24	282	822.94
4	30	-94	-26	165	851.94
5	149	-18	4	0	364.24
5	52	-59	3	320	432.25
5	56	-59	-1	338	463.25
5	62	-56	1	371	574.01
5	81	-76	28	0	634.61
5	77	-80	28	0	697.90
5	80	-81	29	0	726.31
5	79	-85	31	0	757.79
5	78	-81	35	0	790.44
5	145	-1	1	0	904.37
5	26	-2	0	142	934.78
6	51	-48	-27	311	425.57
6	61	-69	-25	366	473.67
6	38	-101	-15	222	579.23
6	43	-101	-14	271	607.23
6	39	-105	-10	227	639.88
6	44	-99	-12	280	673.21
6	46	-96	-13	293	703.37
6	36	-96	-13	213	730.37
6	68	-73	-18	453	780.91
6	59	-72	-17	362	809.32
6	66	-45	-18	426	863.34
7	23	9	15	133	709.09
7	15	9	18	96	739.09
7	25	14	16	142	771.48
7	19	20	15	109	804.56
7	21	19	12	123	834.72
7	27	20	11	144	863.14
7	20	9	6	114	902.22
8	121	30	-7	0	277.81
8	105	99	11	0	376.12
8	90	96	6	0	650.33
8	94	94	6	0	679.33
8	98	90	8	0	710.80
8	100	88	15	0	745.08
8	122	31	8	0	829.51
8	24	24	5	136	864.13
8	18	20	8	102	896.13
9	65	-46	-15	415	542.38
9	67	-51	-21	447	577.19
9	69	-61	-15	455	615.86
9	54	-62	-10	320	647.96
9	50	-66	-12	302	679.43
9	60	-67	-11	365	707.84
9	55	-72	-11	333	739.84
9	64	-72	-7	407	770.84
9	57	-73	-6	353	799.26
9	63	-66	-6	402	833.26
9	58	-62	-3	355	865.26
9	53	-47	-4	320	907.29
10	140	-5	13	0	260.93
10	141	1	18	0	295.74
10	116	-4	102	0	406.89
10	112	-4	108	0	477.60
10	115	-7	111	0	508.84
10	118	-14	112	0	542.91
10	111	-10	108	0	575.57
10	109	-8	108	0	671.20
10	119	-10	105	0	701.81
10	113	-8	104	0	731.04
10	110	-9	102	0	760.28
10	117	-12	101	0	790.44
10	114	-9	100	0	820.60
11	142	2	-2	0	225.13
11	144	5	-4	0	255.73
11	14	7	-9	96	288.12
11	17	14	-6	102	322.73
11	131	20	-6	0	477.60
11	128	21	-7	0	506.01
11	125	19	-8	0	535.25
11	126	19	-9	0	563.25
11	129	24	-10	0	595.35
11	133	27	-5	0	650.33
11	139	28	-3	0	679.57
11	135	27	-2	0	707.98
11	134	25	3	0	740.37
11	132	23	3	0	769.37
11	138	22	2	0	797.78
11	12	20	2	85	826.78
11	123	15	1	0	858.88
11	13	14	0	87	887.29
12	124	17	7	0	92.48
12	130	26	10	0	128.97
12	101	87	21	0	217.96
12	91	92	23	0	250.34
12	76	54	103	0	733.07
12	75	50	113	0	770.84
12	74	46	101	0	810.49
13	143	-9	10	0	556.85
13	146	-11	19	0	593.07
13	71	-46	74	486	685.27
13	73	-49	72	491	715.87
13	72	-57	67	487	752.30
13	70	-57	60	460	786.30
13	150	-21	1	0	882.42
13	147	-12	3	0	918.64

Tabela B.21: Najlepsze rozwiązanie uzyskane przez ContDVRP dla **tai150c**

Pojazd	Zamówienie	X	Y	Otrzymane	Obsłużone
	Depot	0	0	0	1081
1	87	-25	-1	0	511.47
1	98	-29	0	0	538.59
1	97	-29	0	0	561.59
1	65	-97	-27	454	657.76
1	66	-102	-58	466	712.16
1	67	-88	-43	490	755.68
1	64	-84	-43	446	782.68
1	150	-29	-46	0	860.76
1	147	-26	-46	0	917.83
1	144	-29	-39	0	948.44
2	91	-21	1	0	588.55
2	93	-26	3	0	616.93
2	86	-27	5	0	642.17
2	44	-29	13	306	673.42
2	37	-30	18	239	701.52
2	49	-31	20	328	726.75
2	42	-27	17	282	892.80
2	39	-19	15	249	924.05
2	40	-13	13	267	953.37
2	3	-1	6	8	990.26
3	13	-3	1	95	246.39
3	92	-22	3	0	288.49
3	95	-22	5	0	313.49
3	89	-25	4	0	339.65
3	90	-30	4	0	406.35
3	85	-30	3	0	430.35
3	88	-31	2	0	454.76
3	53	-32	4	350	480.00
3	96	-30	9	0	508.39
3	43	-33	10	296	534.55
3	52	-17	16	350	574.64
3	41	-11	17	270	974.96
4	10	3	-4	81	653.60
4	71	4	-4	531	677.60
4	12	4	-4	93	700.60
4	74	4	-8	0	727.60
4	70	-1	-13	503	757.67
4	68	0	-14	496	782.09
4	73	3	-15	537	808.25
4	69	10	-16	497	838.32
4	72	10	-5	533	872.32
4	76	10	-3	0	897.32
4	77	9	-4	0	921.73
4	5	6	-3	26	947.90
5	6	-5	-1	34	923.95
5	1	-1	-4	4	951.95
5	11	-1	-4	81	974.95
5	8	0	-2	49	1000.19
6	14	-1	3	105	786.89
6	36	-14	5	233	823.04
6	48	-24	8	328	856.48
6	50	-27	9	330	882.64
6	94	-27	6	0	917.83
6	7	-2	0	37	966.53
6	4	3	0	20	994.53
6	2	3	0	6	1017.53
7	26	-13	111	155	300.93
7	34	-13	114	227	326.93
7	24	-13	114	149	349.93
7	17	-14	115	111	374.35
7	28	-13	117	163	565.74
7	35	-15	117	231	590.74
7	31	-15	115	191	615.74
7	25	-25	116	155	648.79
7	29	-28	115	169	674.95
7	20	-28	114	125	698.95
7	27	-24	108	157	729.16
7	140	0	68	0	798.81
7	139	2	67	0	824.04
7	45	-13	25	308	891.64
7	47	-18	26	326	919.74
7	38	-20	23	243	946.35
7	46	-16	19	320	975.00
8	116	105	19	0	647.21
8	125	120	25	0	686.36
8	118	125	22	0	715.19
8	124	134	18	0	748.04
8	120	129	14	0	777.44
8	123	126	15	0	803.61
8	121	121	11	0	833.01
8	122	115	14	0	862.72
8	119	115	9	0	890.72
9	117	109	35	0	357.71
9	129	26	76	0	547.97
9	127	24	77	0	573.21
9	133	21	77	0	599.21
9	128	22	69	0	630.27
9	134	20	66	0	656.88
9	135	10	73	0	692.08
9	131	12	67	0	721.41
9	136	9	65	0	748.01
9	126	14	62	0	776.85
9	132	24	60	0	810.04
9	130	28	59	0	837.17
9	112	30	34	0	885.25
9	114	35	26	0	917.68
9	113	29	28	0	947.01
9	115	24	24	0	976.41
9	109	11	7	0	1020.81
10	142	-7	57	0	516.85
10	141	-8	63	0	545.94
10	32	-24	104	197	612.95
10	33	-26	104	227	673.60
10	30	-27	105	181	698.01
10	22	-27	107	143	723.01
10	21	-26	108	135	747.43
10	19	-19	107	119	777.50
10	16	-19	109	109	802.50
10	23	-17	109	145	827.50
10	138	1	61	0	901.76
10	143	4	59	0	928.37
10	137	4	57	0	953.37
11	75	13	-7	0	636.34
11	108	14	-2	0	664.44
11	103	12	-1	0	689.67
11	105	14	0	0	714.91
11	101	15	0	0	738.91
11	99	19	1	0	766.03
11	102	19	1	0	789.03
11	111	19	7	0	818.03
11	110	14	4	0	846.87
11	107	7	7	0	894.32
11	100	8	6	0	918.74
11	106	11	6	0	944.74
11	104	12	6	0	968.74
11	15	6	2	105	998.95
11	9	4	1	56	1024.19
12	18	-21	111	111	356.19
12	81	-102	61	0	604.64
12	84	-107	57	0	759.08
12	79	-105	53	0	786.55
12	80	-98	51	0	816.83
12	83	-97	51	0	840.83
12	82	-93	56	0	870.23
12	78	-88	50	0	901.04
12	54	-31	22	350	987.55
12	51	-20	14	340	1024.15
13	55	2	-108	364	540.42
13	56	1	-122	370	577.45
13	61	2	-127	400	605.55
13	58	-1	-132	388	634.38
13	57	-10	-125	386	710.03
13	60	-16	-126	400	739.11
13	62	-14	-122	406	766.58
13	63	-11	-113	440	799.07
13	59	-13	-111	396	824.90
13	148	-22	-48	0	911.54
13	146	-19	-47	0	937.70
13	145	-23	-43	0	966.36
13	149	-21	-37	0	995.68

Tabela B.22: Najlepsze rozwiązanie uzyskane przez ContDVRP dla **tai150d**

Pojazd	Zamówienie	X	Y	Otrzymane	Obsłużone
	Depot	0	0	0	1025
1	7	-3	9	41	778.24
1	57	-4	12	349	807.40
1	51	-6	14	328	836.23
1	4	-6	15	13	863.23
1	60	0	20	360	897.04
1	59	2	13	354	930.32
2	55	-11	14	345	479.05
2	62	-16	18	381	511.46
2	52	-16	27	332	546.46
2	18	-59	28	124	615.47
2	20	-60	22	128	647.55
2	12	-65	22	70	678.55
2	28	-67	22	183	719.88
2	17	-68	26	115	750.00
2	25	-71	22	160	781.00
2	14	-66	14	83	816.43
2	27	-62	11	175	847.43
2	24	-60	5	153	879.76
3	29	88	50	192	690.59
3	30	91	66	198	732.87
3	32	79	64	217	771.03
3	31	79	63	207	798.03
3	99	51	45	0	857.32
3	97	48	40	0	889.15
3	150	29	23	0	940.64
4	36	4	64	253	422.87
4	48	8	68	321	454.53
4	50	7	69	326	481.95
4	40	11	87	272	526.39
4	33	9	92	237	557.77
4	37	8	92	256	584.77
4	42	0	89	300	619.31
4	43	-3	92	311	649.56
4	35	-6	92	249	678.56
4	34	-8	90	238	707.39
4	46	-9	90	319	734.39
4	49	-13	90	323	764.39
4	39	-3	83	266	802.59
4	44	-6	76	317	836.21
4	38	-8	70	264	868.53
5	91	-90	-17	0	91.59
5	88	-94	-24	0	125.65
5	92	-100	-19	0	159.46
5	90	-105	-22	0	390.58
5	87	-105	-16	0	422.58
5	89	-102	-11	0	454.41
5	93	-95	-11	0	487.41
5	117	-98	6	0	607.01
5	114	-94	13	0	641.07
5	116	-92	12	0	669.31
5	111	-91	13	0	696.73
5	15	-83	8	90	732.16
5	16	-84	11	103	761.32
5	19	-85	15	126	791.44
5	21	-79	18	138	824.15
5	23	-71	14	151	859.10
5	22	-60	11	147	896.50
5	10	-15	7	62	967.68
6	122	-24	88	0	501.21
6	129	-26	92	0	531.69
6	126	-27	91	0	559.10
6	85	-29	87	0	589.57
6	83	-29	85	506	822.38
6	56	-4	24	349	914.30
7	136	28	-121	0	354.82
7	137	36	-123	0	389.07
7	134	35	-114	0	424.12
7	135	46	-102	0	466.40
7	66	-71	-68	392	685.97
7	67	-78	-67	398	719.04
7	71	-78	-62	417	750.04
7	69	-72	-56	409	784.52
7	70	-66	-46	415	822.18
7	68	-57	-54	409	860.22
8	78	10	-5	479	728.68
8	76	11	-6	451	756.09
8	73	17	-9	432	788.80
8	72	22	-5	430	821.21
8	81	22	-6	496	848.21
8	74	22	-13	445	881.21
8	79	8	-16	481	921.52
9	98	46	37	0	545.91
9	104	94	32	0	620.17
9	108	96	29	0	649.77
9	106	95	23	0	681.86
9	105	95	23	0	707.86
9	100	88	24	0	740.93
9	101	87	24	0	767.93
9	102	85	22	0	796.76
9	103	85	20	0	824.76
9	107	83	24	0	855.23
9	75	17	4	447	950.19
10	6	-11	10	36	117.37
10	8	-14	12	45	146.97
10	9	-15	13	62	174.39
10	3	-16	17	11	204.51
10	86	-37	88	0	304.55
10	125	-36	92	0	334.67
10	123	-35	93	0	362.09
10	127	-26	96	0	397.57
10	132	-30	96	0	427.57
10	133	-31	93	0	456.74
10	41	-17	71	279	508.81
10	45	-10	65	319	544.03
10	47	-11	64	321	616.79
10	63	-4	30	383	677.50
10	61	-9	28	362	708.89
10	58	-9	24	351	738.89
10	53	-11	20	336	769.36
10	64	-11	20	389	795.36
10	54	-10	18	338	823.60
10	1	-9	12	5	855.68
10	2	-10	11	7	883.09
10	5	-4	7	28	916.30
10	11	-3	5	68	944.54
11	131	-28	88	0	707.35
11	130	-30	92	0	737.82
11	128	-32	90	0	766.65
11	124	-33	91	0	794.06
11	84	-34	90	509	821.48
11	82	-36	83	498	854.76
11	65	-5	22	392	949.18
12	26	-86	4	162	547.34
12	118	-88	6	0	576.17
12	115	-93	11	0	609.24
12	119	-102	10	0	644.30
12	112	-101	1	0	679.35
12	120	-101	0	0	706.35
12	109	-96	-2	0	737.74
12	110	-95	-2	0	764.74
12	121	-94	-1	0	792.15
12	113	-89	1	0	823.54
12	13	-72	1	75	866.54
13	149	29	19	0	598.42
13	146	30	19	0	625.42
13	145	32	20	0	653.66
13	148	34	20	0	681.66
13	139	33	26	0	713.74
13	141	31	23	0	820.35
13	138	34	25	0	849.96
13	144	35	30	0	881.06
13	147	33	28	0	909.88
14	140	25	26	0	394.82
14	142	32	31	0	429.42
14	94	44	50	0	477.89
14	96	44	55	0	594.75
14	95	40	50	0	627.15
14	143	33	19	0	684.93
14	80	27	-2	481	893.47
14	77	25	-5	468	923.07

Bibliografia

[1] T. J. Ai and V. Kachitvichyanukul. A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. *Computers and Operations Research*, 36(5):1693–1702, 2009.

[2] T. J. Ai and V. Kachitvichyanukul. Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem. *Computers & Industrial Engineering*, 56(1):380–387, 2009.

[3] H. Akaike. A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723, 1974.

[4] M. Albareda-Sambola, E. Fernández, and G. Laporte. The dynamic multiperiod vehicle routing problem with probabilistic information. *Computers & Operations Research*, 48:31–39, 2014.

[5] J. Arabas. *Wykłady z algorytmów ewolucyjnych*. Wydawnictwa Naukowo-Techniczne, 2004.

[6] J. Arabas. Metaheuristics – a modern approach. Wykłady w ramach III Szkoły Letniej organizowanej przez IPI PAN, jun 2014.

[7] A. Ben-Tal and A. Nemirovski. Robust optimization - methodology and applications. *Mathematical Programming*, 92(3):453–480, 2002.

[8] D. Bertsimas, D. B. Brown, and C. Caramanis. Theory and Applications of Robust Optimization. *SIAM Review*, 53(3):438–464, 2011.

[9] H. G. Beyer and B. Sendhoff. Robust optimization - A comprehensive survey. *Computer Methods in Applied Mechanics and Engineering*, 196(33-34):3190–3218, 2007.

[10] T. Blackwell. Particle swarm optimization in dynamic environments. *Evolutionary Computation in Dynamic and Uncertain Environments*, 51:29–49, 2007.

[11] T. Blackwell and J. Branke. Multiswarms, exclusion, and anti-convergence in dynamic environments. *IEEE Transactions on Evolutionary Computation*, 10(4):459–472, 08 2006.

[12] W. J. Bolosky and M. L. Scott. False Sharing and Its Effect on Shared Memory Performance. W *USENIX Systems on USENIX Experiences with Distributed and Multiprocessor Systems - Volume 4*, volume 1801 of *Sedms'93*, strona 3, Berkeley, CA, USA, 1993. USENIX Association.

[13] J. Branke. Memory enhanced evolutionary algorithms for changing optimization problems. W *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, volume 3, strony 1875–1882. Citeseer, IEEE, 1999.

[14] J. Branke and D. C. Mattfeld. Anticipation and flexibility in dynamic scheduling. *International Journal of Production Research*, 43(15):3103–3129, 2005.

[15] J. Branke, M. Orbayi, and S. Uyar. The role of representations in dynamic knapsack problems. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3907 LNCS:764–775, 2006.

[16] J. Branke, E. Salihoğlu, and S. Uyar. Towards an analysis of dynamic environments. W *Proceedings of the 2005 conference on Genetic and evolutionary computation - GECCO '05*, strona 1433, New York, New York, USA, 2005. ACM Press.

[17] D. Brzezinski and J. Stefanowski. Classifiers for Concept-drifting Data Streams: Evaluating Things That Really Matter.

[18] D. Brzezinski and J. Stefanowski. Reacting to different types of concept drift: The accuracy updated ensemble algorithm. *Neural Networks and Learning Systems, IEEE Transactions on*, 25(1):81–94, 2014.

[19] E. K. Burke, G. Kendall, J. Newall, E. Hart, P. Ross, and S. Schulenburg. Hyper-heuristics: An emerging direction in modern search technology. *International series in operations research and management science*, strony 457–474, 2003.

[20] K. Chakhlevitch and P. Cowling. Hyperheuristics : Recent Developments. strony 3–29, 2008.

[21] X. Chen, L. Feng, and Y.-S. Ong. A self-adaptive memeplexes robust search scheme for solving stochastic demands vehicle routing problem. *Int. J. Systems Science*, 43(7):1347–1366, 2012.

[22] N. Christofides and J. E. Beasley. The period routing problem. *Networks*, 14(2):237–256, jan 1984.

[23] G. U. Clarke and J. W. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, 12(4):568–581, 1964.

[24] P. Cowling, G. Kendall, and E. Soubeiga. A hyperheuristic approach to scheduling a sales summit. *Practice and Theory of Automated Timetabling III*, strony 176–190, 2001.

[25] G. A. Croes. A method for solving traveling salesman problems. *Operations Research 6*, strony 791–812, 1958.

[26] C. Cruz, J. R. González, and D. A. Pelta. Optimization in dynamic environments: a survey on problems, methods and measures. *Soft Computing*, 15(7):1427–1448, 2011.

[27] D.-c. Dang and T. Jansen. Populations can be Essential in Dynamic Optimisation. strony 1407–1414.

[28] G. B. Dantzig and J. H. Ramser. The Truck Dispatching Problem. *Management Science*, 6(1):80–91, 1959.

[29] K. De Jong. Evolving in a changing world. W *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 1609, strony 512–519. 1999.

[30] I. G. Del Amo, D. A. Pelta, J. R. González, and A. D. Masegosa. An algorithm comparison for dynamic optimization problems. *Applied Soft Computing*, 12(10):3176–3192, 2012.

[31] M. Dror, G. Laporte, and P. Trudeau. Vehicle routing with stochastic demands: Properties and solution frameworks. *Transportation science*, 23(3):166–176, 1989.

[32] T. C. Du, E. Y. Li, and D. Chou. Dynamic vehicle routing for online B2C delivery. *Omega*, 33(1):33–45, 2005.

[33] M. Elhassania, B. Jaouad, and E. A. Ahmed. A new hybrid algorithm to solve the vehicle routing problem in the dynamic environment. *International Journal of Soft Computing*, 8(5):327–334, 2013.

[34] M. Elhassania, B. Jaouad, and E. A. Ahmed. Solving the dynamic Vehicle Routing Problem using genetic algorithms. W *Logistics and Operations Management (GOL), 2014 International Conference on*, strony 62–69. IEEE, 2014.

[35] F. Ferrucci, S. Bock, and M. Gendreau. A pro-active real-time control approach for dynamic vehicle routing problems dealing with the delivery of urgent goods. *European Journal of Operational Research*, 225(1):130–141, 2013.

[36] M. L. Fisher and R. Jaikumar. A generalized assignment heuristic for vehicle routing. *Networks*, 11(2):109–124, 1981.

[37] M. M. Flood. The traveling-salesman problem. *Operations Research*, 4(1):61–75, 1956.

[38] A. M. Frieze, G. Galbiati, and F. Maffioli. On the worst-case performance of some algorithms for the asymmetric traveling salesman problem. *Networks*, 12(1):23–39, 1982.

[39] L. M. Gambardella and M. Dorigo. Solving symmetric and asymmetric TSPs by ant colonies. W *IEEE Conference on Evolutionary Computation*, strony 622–627, 1996.

[40] L. M. Gambardella, É. D. Taillard, and G. Agazzi. MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows. *New Ideas in Optimization*, (IDSIA-06-99):1–17, 1999.

[41] P. Garrido and C. Castro. Stable solving of CVRPs using hyperheuristics. *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, strony 255–262, 2009.

[42] P. Garrido and C. Castro. A Flexible and Adaptive Hyper-heuristic Approach for (Dynamic) Capacitated Vehicle Routing Problems. *Fundamenta Informaticae*, 119(1):29–60, 2012.

[43] P. Garrido and M. C. Riff. DVRP: a hard dynamic combinatorial optimisation problem tackled by an evolutionary hyper-heuristic. *Journal of Heuristics*, 16(6):795–834, 2010.

[44] S. Geetha, G. Poonthalir, and P. T. Vanathi. Improved k-means algorithm for capacitated clustering problem. *INFOCOMP Journal of Computer Science*, 8(4):52–59, 2009.

[45] F. P. Goksal, I. Karaoglan, and F. Altiparmak. A hybrid discrete particle swarm optimization for vehicle routing problem with simultaneous pickup and delivery. *Computers & Industrial Engineering*, 65(1):39–53, 2013.

[46] J. R. González, A. D. Masegosa, and I. J. Garcíia. A cooperative strategy for solving dynamic optimization problems. *Memetic Computing*, 3(1):3–14, mar 2011.

[47] F. T. Hanshar and B. M. Ombuki-Berman. Dynamic vehicle routing using genetic algorithms. *Applied Intelligence*, 27(1):89–99, 2007.

[48] X. Hu and L. Sun. Knowledge-based modeling for disruption management in urban distribution. *Expert Systems with Applications*, 39(1):906–916, 2012.

[49] X. Hu, L. Sun, and T. Hu. An Approach to Knowledge Representation for Vehicle Routing Problems. *2006 International Conference on Management Science and Engineering*, (I):487–492, 2006.

[50] X. Hu, L. Sun, and L. Liu. A PAM approach to handling disruptions in real-time vehicle routing problems. *Decision Support Systems*, 54(3):1380–1393, 2013.

[51] G. J. S. R. J. C. Gower. Minimum Spanning Trees and Single Linkage Cluster Analysis. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 18(1):54–64, 1969.

[52] Y. Jin and J. Branke. Evolutionary Optimization in Uncertain Environments—A Survey. *IEEE Transactions on Evolutionary Computation*, 9(3):1–15, 2005.

[53] J. Kennedy and R. C. Eberhart. Particle Swarm Optimization. *Proceedings of IEEE International Conference on Neural Networks. IV*, strony 1942–1948, 1995.

[54] M. R. Khouadjia. *Solving Dynamic Vehicle Routing Problems: From Single-Solution Based Metaheuristics to Parallel Population Based Metaheuristics*. rozprawa doktorska, Lille1 University, France, 2011.

[55] M. R. Khouadjia, E. Alba, L. Jourdan, and E.-G. Talbi. Multi-Swarm Optimization for Dynamic Combinatorial Problems: A Case Study on Dynamic Vehicle Routing Problem. W *Swarm Intelligence*, volume 6234 of *Lecture Notes in Computer Science*, strony 227–238. Springer, Berlin / Heidelberg, 2010.

[56] M. R. Khouadjia, L. Jourdan, and E.-G. Talbi. Adaptive particle swarm for solving the dynamic vehicle routing problem. W *2010 IEEE/ACS International Conference on Computer Systems and Applications (AICCSA)*, strony 1–8. IEEE, 2010.

[57] M. R. Khouadjia, B. Sarasola, E. Alba, L. Jourdan, and E.-G. Talbi. A comparative study between dynamic adapted PSO and VNS for the vehicle routing problem with dynamic requests. *Applied Soft Computing*, 12(4):1426–1439, 2012.

[58] M. R. Khouadjia, E.-G. Talbi, L. Jourdan, B. Sarasola, and E. Alba. Multi-environmental cooperative parallel metaheuristics for solving dynamic optimization problems. *Journal of Supercomputing*, 63(3):836–853, 2013.

[59] P. Kilby, P. Prosser, and P. Shaw. Dynamic VRPs: A Study of Scenarios. Technical report, 1998.

[60] G. A. P. Kindervater, J. K. Lenstra, and D. B. Shmoys. The parallel complexity of TSP heuristics. *Journal of Algorithms*, 10(2):249–270, 1989.

[61] A. L. Kok, C. M. Meyer, H. Kopfer, and J. M. J. Schutten. A dynamic programming heuristic for the vehicle routing problem with time windows and European Community social legislation. *Transportation Science*, 44(4):442–454, 2010.

[62] J. B. Kruskal. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. *Proceedings of the American Mathematical Society*, strony 48–50, 1959.

[63] A. Larsen. *The Dynamic Vehicle Routing Problem.* rozprawa doktorska, Technical University of Denmark, 2000.

[64] J. K. Lenstra and A. H. G. R. Kan. Complexity of vehicle routing and scheduling problems. *Networks*, 11:221–227, 1981.

[65] C. Li and S. Yang. A generalized approach to construct benchmark problems for dynamic optimization. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5361 LNAI:391–400, 2008.

[66] C. Lin, K. L. Choy, G. T. S. Ho, H. Y. Lam, G. K. H. Pang, and K. S. Chin. A decision support system for optimizing dynamic courier routing operations. *Expert Systems with Applications*, 41(15):6917–6933, 2014.

[67] S. Lin and B. W. Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations research*, 21(2):498–516, 1973.

[68] A. Mackey. Windows communication foundation. W *Introducing .NET 4.0*, strony 159–173. Springer, 2010.

[69] J. Mańdziuk and C. Nejman. UCT-Based Approach to Capacitated Vehicle Routing Problem. W *International Conference on Artificial Intelligence and Soft Computing*, strony 679–690. Springer, 2015.

[70] J. Mańdziuk, S. Zadrożny, K. Walędzik, M. Okulewicz, and M. Świechowski. Adaptive metaheuristic methods in dynamically changing environments, 2015.

[71] J. Mańdziuk and A. Żychowski. A memetic approach to vehicle routing problem with dynamic requests. *Applied Soft Computing*, 48:522–534, 2016.

[72] Y. Marinakis, G.-R. Iordanidou, and M. Marinaki. Particle Swarm Optimization for the Vehicle Routing Problem with Stochastic Demands. *Applied Soft Computing Journal*, 13(4):1693–1704, 2013.

[73] Y. Marinakis, M. Marinaki, and G. Dounias. A hybrid particle swarm optimization algorithm for the vehicle routing problem. *Engineering Applications of Artificial Intelligence*, 23(4):463–472, 2010.

[74] M. Mavrovouniotis and S. Yang. Ant colony optimization with immigrants schemes for the dynamic travelling salesman problem with traffic factors. *Applied Soft Computing*, 13(10):4023–4037, 2013.

[75] R. Montemanni, L. M. Gambardella, A. E. Rizzoli, and A. V. Donati. A new algorithm for a dynamic vehicle routing problem based on ant colony system. *Journal of Combinatorial Optimization*, 10:327–343, 2005.

[76] R. Montemanni, L. M. Gambardella, A. E. Rizzoli, and A. V. Donati. Ant colony system for a dynamic vehicle routing problem. *Journal of Combinatorial Optimization*, 10(4):327–343, 2005.

[77] S. Morales-Enciso and J. Branke. Tracking global optima in dynamic environments with efficient global optimization. *European Journal of Operational Research*, 242(3):744–755, may 2015.

[78] R. Moretti Branchini, V. Amaral Armentano, and A. Løkketangen. Adaptive granular local search heuristic for a dynamic vehicle routing problem. *Computers and Operations Research*, 36(11):2955–2968, 2009.

[79] P. Moscato. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech Concurrent Computation Program, C3P Report*, 826:1989, 1989.

[80] Msdn.microsoft.com. Biblioteka klas programu .NET Framework, 05 2016.

[81] T. T. Nguyen. *Continuous Dynamic Optimisation Using Evolutionary Algorithms*. Rozprawa doktorska, 2010.

[82] T. T. Nguyen, S. Yang, and J. Branke. Evolutionary dynamic optimization: A survey of the state of the art. *Swarm and Evolutionary Computation*, 6:1–24, 08 2012.

[83] M. Okulewicz and J. Mańdziuk. Application of Particle Swarm Optimization Algorithm to Dynamic Vehicle Routing Problem. W *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 7895 LNAI, strony 547–558. 2013.

[84] M. Okulewicz and J. Mańdziuk. Two-Phase Multi-Swarm PSO and the Dynamic Vehicle Routing Problem. W *2nd IEEE Symposium on Computational Intelligence for Human-like Intelligence*, strony 86–93, Orlando, Fl, USA, 12 2014. IEEE.

[85] M. Okulewicz and J. Mańdziuk. Dynamic Vehicle Routing Problem: A Monte Carlo approach. W *2nd ITRIA Conference*, strony 119–138, 2015.

[86] M. Okulewicz and J. Mańdziuk. Dynamic Vehicle Routing Problem: optimization modules analysis for the Particle Swarm Optimization based algorithm. 2016.

[87] M. Okulewicz and J. Mańdziuk. Particle Swarm Optimization hyper-heuristic for the Dynamic Vehicle Routing Problem. W *7th BIOMA Conference*, strony 215–227, 2016.

[88] M. Okulewicz and J. Mańdziuk. Solving Dynamic Vehicle Routing Problem in a continuous search space. (w recenzji), 2016.

[89] I. H. Osman and N. Christofides. Capacitated Clustering Problems by Hybrid Simulated Annealing and Tabu Search. *International Transactions in Operational Research*, 1(3):317–336, 1994.

[90] G. Pankratz and V. Krypczyk. Benchmark data sets for dynamic vehicle routing problems, 2009.

[91] D. Pelta, C. Cruz, and J. R. González. A study on diversity and cooperation in a multiagent strategy for dynamic optimization problems. *International Journal of Intelligent Systems*, 24(7):844–861, 2009.

[92] D. Pflugfelder, J. J. Wilkens, and U. Oelfke. Worst case optimization: a method to account for uncertainties in the optimization of intensity modulated proton therapy. *Physics in medicine and biology*, 53(6):1689–700, 2008.

[93] V. Pillac. *Dynamic vehicle routing: solution methods and computational tools*. rozprawa doktorska, École des Mines de Nantes, 2012.

[94] V. Pillac, M. Gendreau, C. Guéret, and A. L. Medaglia. A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1):1–11, 2013.

[95] V. Pillac, C. Guéret, and A. L. Medaglia. An event-driven optimization framework for dynamic vehicle routing. *Decision Support Systems*, 54(1):414–423, 2012.

[96] D. Pisinger and S. Ropke. A general heuristic for vehicle routing problems. *Computers and Operations Research*, 34(8):2403–2435, 2007.

[97] D. S. Platt. *Introducing Microsoft .Net*. Microsoft press, 2002.

[98] J.-Y. Potvin and J.-M. Rousseau. An Exchange Heuristic for Routeing Problems with Time Windows. *Journal of the Operational Research Society*, 46(12):1433–1446, 1995.

[99] C. Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, 31(12):1985–2002, 2004.

[100] C. Prins. Two memetic algorithms for heterogeneous fleet vehicle routing problems. *Engineering Applications of Artificial Intelligence*, 22(6):916–928, 2009.

[101] J. Prosise. *Programming Microsoft .NET*. Microsoft Press, 2002.

[102] H. N. Psaraftis. Dynamic Vehicle Routing Problems. W *Vehicle Routing: Methods and Studies*. Elsevier, 1988.

[103] H. N. Psaraftis. Dynamic vehicle routing: Status and prospects. *Annals of Operations Research*, 61(1):143–164, 1995.

[104] R Core Team. R: A Language and Environment for Statistical Computing, 2012.

[105] T. K. Ralphs, L. Kopman, W. R. Pulleyblank, and L. E. Trotter. On the capacitated vehicle routing problem. *Mathematical programming*, 94(2-3):343–359, 2003.

[106] R. Reynolds. Knowledge-based self-adaptation in evolutionary programming using cultural algorithms. *Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC '97)*, (d):71–76, 1997.

[107] P. Rohlfshagen, P. K. Lehre, and X. Yao. Evolutionary Computation for Dynamic Optimization Problems. *Evolutionary Computation for DOPs*, 490:221–240, 2013.

[108] P. Rohlfshagen and X. Yao. Attributes of Dynamic Combinatorial Dynamic Optimisation : A Brief Overview. strony 442–451, 2008.

[109] P. Rohlfshagen and X. Yao. Dynamic combinatorial optimisation problems: An analysis of the subset sum problem. *Soft Computing*, 15(9):1723–1734, 2011.

[110] P. Rohlfshagen and X. Yao. Dynamic combinatorial optimization problems: A fitness landscape analysis. W *Metaheuristics for Dynamic Optimization*, strony 79–97. Springer, 2013.

[111] D. M. Ryan, C. Hjorring, and F. Glover. Extensions of the Petal Method for Vehicle Routeing. *The Journal of the Operational Research Society*, 44(3):289–296, 1993.

[112] P. Ryser-Welch and J. F. Miller. A Review of Hyper-Heuristic Frameworks.

[113] A. Slater. Specification for a dynamic vehicle routing and scheduling system. *International Journal of Transport Management*, 1(1):29–40, 2002.

[114] M. Sniedovich. Wald's maximin model: A treasure in disguise! *The Journal of Risk Finance*, 9(3):287–291, 2008.

[115] M. Sniedovich. A bird's view of info-gap decision theory. *Journal of Risk Finance (Emerald Group Publishing Limited)*, 11(3):268–283, 2010.

[116] M. M. Solomon. Algorithms for and Scheduling Problems the Vehicle Routing With Time Window Constraints. *Operations Research*, 35(2):254–265, 1987.

[117] R. Storn and K. Price. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11(4):341–359, 1997.

[118] L. Sun, X. Hu, Z. Wang, and M. Huang. A Knowledge-Based Model Representation and On-Line Solution Method for Dynamic Vehicle Routing Problem. *Computational Science–ICCS 2007*, strony 218–226, 2007.

[119] I. Sungur, F. Ordónez, and M. Dessouky. A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty. *IIE Transactions*, 40(5):509–523, 2008.

[120] É. D. Taillard. Parallel iterative search methods for vehicle routing problems. *Networks*, 23(8):661–673, 1993.

[121] É. D. Taillard. A heuristic column generation method for the heterogeneous fleet VRP. *RAIRO - Operations Research*, 33(1):1–14, 1999.

[122] E. Taniguchi and H. Shimamoto. Intelligent transportation system based dynamic vehicle routing and scheduling with variable travel times. *Transportation Research Part C: Emerging Technologies*, 12(3):235–250, 2004.

[123] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[124] R. Tinós. Artificially inducing environmental changes in evolutionary dynamic optimization. W *International Conference on Parallel Problem Solving from Nature*, strony 225–236. Springer International Publishing, 2016.

[125] R. Tinós and S. Yang. A self-organizing random immigrants genetic algorithm for dynamic optimization problems. *Genetic Programming and Evolvable Machines*, 8(3):255–286, 08 2007.

[126] R. Tinós and S. Yang. Analysis of fitness landscape modifications in evolutionary dynamic optimization. *Information Sciences*, 282(October):214–236, 2014.

[127] J. Torrellas, H. Lam, and J. Hennessy. False sharing and spatial locality in multiprocessor caches. *IEEE Transactions on Computers*, 43(6):651–663, jun 1994.

[128] J. I. van Hemert and J. A. La Poutré. *Parallel Problem Solving from Nature - PPSN VIII*, volume 3242 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[129] A. Wald. Statistical Decision Functions Which Minimize the Maximum Risk. *The Annals of Mathematics*, 46(2):265–280, 1945.

[130] K. Weicker. *Evolutionary Algorithms and Dynamic Optimization Problems*. Rozprawa doktorska, Universitat Stuttgart, 2003.

[131] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82, 1997.

[132] J. Yang, P. Jaillet, and H. Mahmassani. Real-Time Multivehicle Truckload Pickup and Delivery Problems. *Transportation Science*, 38(2):135–148, 2004.

[133] S. Yang. Explicit memory schemes for evolutionary algorithms in dynamic environments. W *Evolutionary computation in dynamic and uncertain environments*, strony 3–28. Springer, 2007.

[134] S. Yang, Y. Jiang, and T. T. Nguyen. Metaheuristics for dynamic combinatorial optimization problems. *IMA Journal of Management Mathematics*, 24(4):451–480, 2013.

[135] S. Yang and C. Li. A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments. *IEEE Transactions on Evolutionary Computation*, 14(6):959–974, 2010.

[136] A. Younes, P. Calamai, and O. Basir. Generalized Benchmark Generation for Dynamic Combinatorial Problems. strony 25–31, 2005.

[137] C.-S. Yu and H.-L. Li. A robust optimization model for stochastic logistic problems. *International Journal of Production Economics*, 64(1-3):385–397, 2000.

[138] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software*, 23(4):550–560, 1997.