

**Akademia Górniczo-Hutnicza
im. Stanisława Staszica w Krakowie**

Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej

KATEDRA AUTOMATYKI I ROBOTYKI

AUTOREFERAT

DR INŻ. WOJCIECH CHMIEL

**WNIOSEK O WSZCZĘCIE POSTĘPOWANIA HABILITACYJNEGO
W DZIEDZINIE NAUK TECHNICZNYCH W DYSCYPLINIE
AUTOMATYKA I ROBOTYKA**

Kraków, 2019

Spis treści

1. Uzyskane tytuły i stopnie naukowe	5
2. Przebieg zatrudnienia	6
3. Wskazanie osiągnięć naukowo-badawczych	7
3.1. Tytuł i zakres osiągnięcia naukowego.....	7
3.1.1. Publikacje wchodzące w skład osiągnięcia	8
3.2. Syntetyczny opis oryginalnych wyników zawartych w cyklu publikacji.....	9
3.2.1. Opracowanie mechanizmów bazujących na zaproponowanej i udowodnionej przez Autora formule umożliwiającej określenie warunkowej wartości oczekiwanej funkcji celu dla zagadnienia kwadratowego przydziału	11
3.2.2. Opracowanie mechanizmów poprawiających wydajność algorytmów przybliżonych dla \mathcal{NP} -trudnych zagadnień optymalizacyjnych	17
3.2.3. Opracowanie dla zagadnienia kwadratowego przydziału (QAP) algorytmu przybliżonego stosującego paradygmaty mechaniki kwantowej.....	26
3.2.4. Opracowanie równoległych implementacji algorytmów populacyjnych na platformie sprzętowej GPU (<i>Graphical Processing Unit</i>) w celu poprawy ich efektywności w optymalizacji problemów \mathcal{NP} -trudnych	30
3.2.5. Opracowanie algorytmów do optymalizacji ruchu drogowego z zastosowaniem arytmetyki przedziałowej	41
3.2.6. Opracowanie algorytmu rojowego (PSO) do wyznaczania najlepszej trasy oraz sterowania ruchem drogowym poprzez dynamiczny dobór długości faz sygnalizacji świetlnej	48
3.2.7. Opracowanie specjalizowanego algorytmu ewolucyjnego wykorzystywanego w predykcji parametrów ruchu drogowego z zastosowaniem formalizmu rozmytych map kognitywnych	53
3.2.8. Opracowanie specjalizowanego algorytmu przybliżonego bazującego na algorytmie Tabu Search do oceny scenariuszy rozwoju technologii (<i>roadmapping</i>)	57
4. Wskazanie dodatkowych osiągnięć naukowych	62
4.1. Publikacje wchodzące w skład dodatkowych osiągnięć naukowych	62
4.2. Syntetyczny opis osiągnięć dodatkowych.....	63
5. Ogólny przegląd osiągnięć - statystyki	65
5.1. Publikacje z <i>Impact Factor</i> oraz punktacja na podstawie list ministerialnych.....	66
5.1.1. Cytowania prac	69
5.2. Realizacja krajowych i zagranicznych projektów badawczych	70

5.3. Udział w komitetach redakcyjnych	71
5.4. Współpraca międzynarodowa	71
5.5. Współpraca z przemysłem - wdrożenia.....	72
5.5.1. Wdrożenia zrealizowane indywidualne.....	72
5.5.2. Wdrożenia zrealizowane w ramach realizowanych projektów naukowo- badawczych	72
5.6. Nagrody	73
5.7. Działalność dydaktyczna w latach 2004-2018.....	74
5.7.1. Prowadzone zajęcia	74
5.7.2. Działalność organizacyjna	74
5.7.3. Prace inżynierskie i dyplomowe	75
5.7.4. Recenzje artykułów w czasopismach międzynarodowych	75
5.8. Inne	75

1. Uzyskane tytuły i stopnie naukowe

2004 - Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie, Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki. Uzyskanie stopnia doktora nauk technicznych w dyscyplinie Automatyka i Robotyka.

1992 - Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie, Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki. Uzyskanie stopnia magistra inżyniera w dyscyplinie Elektronika ze specjalizacją Automatyka.

2. Przebieg zatrudnienia

Okres	Miejsce zatrudnienia
1992-2003	Wydział Elektroniki, Automatyki, Informatyki i Elektrotechniki, Katedra Automatyki, Laboratorium Badań Operacyjnych na Akademii Górniczo-Hutniczej w Krakowie – stanowisko naukowo-techniczne.
2004–2019	Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej, Katedra Automatyk i Robotyki – stanowisko adiunkta.
2000-2002	BOT.NET USA – zastosowanie metod sztucznej inteligencji do przetwarzania danych w oparciu o algorytmy wnioskujące m.in. projekt oraz realizacja automatycznego systemu wnioskującego na rozproszonych repozytoriach ontologicznych – lider grupy tworzącej algorytmy AI.
2002-2004	ComputerLand S.A – zarządzanie analizą biznesową, opracowanie wymagań oraz projekt techniczny projektów dla: <i>Intranet Portal PZU GROUP</i> , <i>System zarządzania przepływem pracy UPC Broadband</i> , <i>System zarządzania przepływem pracy PFRON</i> – kierownik grupy analitycznej.

3. Wskazanie osiągnięć naukowo-badawczych

3.1. Tytuł i zakres osiągnięcia naukowego

Zgodnie z art. 16 ust. 2 ustawy z dnia 14 marca 2003 r. o stopniach naukowych i tytule naukowym oraz o stopniach i tytule w zakresie sztuki, Autor wskazuje jako osiągnięcie naukowe cykl publikacji pod wspólnym tytułem:

„Zastosowanie algorytmów przybliżonych do rozwiązywania \mathcal{NP} -trudnych zagadnień dyskretnych.”

Prezentowany cykl publikacji obejmuje 14 publikacji, w tym 5 z listy filadelfijskiej i 12 indeksowanych w *Web of Science*. Wśród osiągnięć dodatkowych, rozszerzających główny nurt badawczy, wyróżniono 11 publikacji, w tym 2 z listy filadelfijskiej. Wyniki przedstawione w poniżej zaprezentowanych pracach stanowią oryginalne osiągnięcia Autora i mogą być traktowane jako jego oryginalny wkład w rozwój nauk technicznych w dyscyplinie Automatyki i Robotyki. Na osiągnięcie naukowe składa się:

1. Opracowanie mechanizmów bazujących na zaproponowanej i udowodnionej przez Autora formule umożliwiającej wyznaczenie warunkowej wartości oczekiwanej funkcji celu dla zagadnienia kwadratowego przydziału (*Quadratic Assignment Problem - QAP*) [4].
2. Opracowanie mechanizmów poprawiających wydajność populacyjnych algorytmów przybliżonych dla \mathcal{NP} -trudnych zagadnień optymalizacyjnych ([11], [22], [10], [21]).
3. Opracowanie dla zagadnienia kwadratowego przydziału (QAP) algorytmu przybliżonego wykorzystującego paradygmaty mechaniki kwantowej. Zaproponowany mechanizm bazujący na reprezentacji z zastosowaniem q -bitów oraz przetwarzaniu z zastosowaniem bramek kwantowych pozwala na sterowanie poziomem eksploracji lub intensyfikacji podczas przeszukiwania przestrzeni rozwiązań ([15]).
4. Opracowanie równoległych implementacji algorytmów populacyjnych na platformie sprzętowej GPU (*Graphical Processing Unit*) w celu poprawy ich efektywności w poszukiwaniu rozwiązań przybliżonych dla \mathcal{NP} -trudnych zagadnień optymalizacyjnych ([27], [19]).
5. Opracowanie algorytmów umożliwiających optymalizację ruchu drogowego, w których model dyskretny został rozszerzony poprzez zastosowanie arytmetyki przedziałowej. Pozwoliło to sformułować odpowiednie twierdzenia umożliwiające optymalizację decyzji o wyborze trasy, gdy stan ruchu drogowego ulega częstym zmianom ([17]).
6. Opracowanie algorytmu rojowego (PSO) w celu wyznaczania najlepszej trasy oraz sterowania ruchem drogowym poprzez dynamiczny dobór długości faz sygnalizacji świetlnej. Zastosowanie algorytmu przybliżonego pozwoliło na optymalizację trasy ruchu w sieci

- drogowej w czasie rzeczywistym, uwzględniając częste zmiany wartości parametrów opisujących stan ruchu drogowego ([1], [6], [9]).
7. Opracowanie specjalizowanego algorytmu ewolucyjnego wykorzystywanego w predykcji parametrów opisujących ruch drogowy z zastosowaniem formalizmu rozmytych map kognitywnych. Zaproponowana metoda pozwala na efektywne oszacowanie parametrów ruchu dla całej sieci na podstawie rzadkich pomiarów ([18]).
 8. Opracowanie specjalizowanego algorytmu przybliżonego bazującego na algorytmie Tabu Search do oceny scenariuszy rozwoju technologii ([12]).

3.1.1. Publikacje wchodzące w skład osiągnięcia

1. [Chmiel, 2019] W. Chmiel. Evolutionary algorithm using conditional expectation value for quadratic assignment problem. *Swarm and Evolutionary Computation*, 46:1 – 27, 2019 (IF=3.818, MNiSW: 50).
2. [Chmiel, Kwiecień, 2018a] W. Chmiel and J. Kwiecień. Quantum-Inspired Evolutionary Approach for the Quadratic Assignment Problem. *Entropy*, 20(10), 2018 (IF=2.35, MNiSW: 30) (60%).
3. [Chmiel et al., 2018b] W. Chmiel, I. Skalna, and S. Jędrusik. Intelligent route planning system based on interval computing. *Multimedia Tools and Applications*, Oct 2018 (IF=1.154, MNiSW: 30) (75%).
4. [Chmiel et al., 2017a] W. Chmiel, P. Kadłuczka, J. Kwiecień, and B. Filipowicz. A comparison of nature inspired algorithms for the quadratic assignment problem. *Bulletin of the Polish Academy of Sciences: Technical Sciences*, 65(4):513–522, 2017 (IF=1.361, MNiSW: 25) (55%).
5. [Dziech, Chmiel et al., 2017b] Andrzej Dziech and Wojciech Chmiel and Sebastian Ernst and Piotr Kadłuczka and Zbigniew Mikrut and Piotr Szwed and Igor Wojnicki. System and the method to manage the rules of control of the traffic lights on the crossing, PL414870A1, BUP 11/2017, 11 2017 (Patent) (55%).
6. [Chmiel et al., 2016a] W. Chmiel, J. Dańda, A. Dziech, S. Ernst, P. Kadłuczka, Z. Mikrut, P. Pawlik, P. Szwed, and I. Wojnicki. INSIGMA: an intelligent transportation system for urban mobility enhancement. *Multimedia Tools and Applications*, 75(17):10529–10560, Sep 2016 (IF=1.361, MNiSW: 25) (32%).
7. [Chmiel et al., 2016b] W. Chmiel, P. Kadłuczka, J. Kwiecień, B. Filipowicz, and P. Pukocz. Strategic Planning Optimization Using Tabu Search Algorithm. In A. M. Skulimowski and J. Kacprzyk, editors, *Knowledge, Information and Creativity Support Systems: Recent Trends, Advances and Solutions*, pages 315–328, Cham, 2016. Springer International Publishing (WoS, MNiSW: 15) (60%).
8. [Chmiel et al., 2015a] W. Chmiel and P. Szwed. Learning Fuzzy Cognitive Map for Traffic Prediction Using an Evolutionary Algorithm. In A. Dziech, M. Leszczuk, and R. Baran, editors, *Multimedia Communications, Services and Security*, volume 566 of *Communications in Computer and Information Science*, pages 195–209. Springer International Publishing, 2015 (WoS, MNiSW: 15) (60%).
9. [Szwed, Chmiel et al., 2015b] P. Szwed, W. Chmiel, and P. Kadłuczka. OpenCL Implementation of PSO Algorithm for the Quadratic Assignment Problem. In L. Rutkowski,

- M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, and J. M. Zurada, editors, *Artificial Intelligence and Soft Computing*, volume 9120 of *Lecture Notes in Computer Science*, pages 223–234. Springer International Publishing, 2015 (WoS, MNI_{SW}: 15) (**45%**).
10. [**Chmiel**, Szwed, **2015c**] W. Chmiel and P. Szwed. Bees Algorithm for the Quadratic Assignment Problem on CUDA Platform. In A. Gruca, A. Brachman, S. Kozielski, and T. Czachórski, editors, *Man–Machine Interactions 4*, volume 391 of *Advances in Intelligent Systems and Computing*, pages 615–625. Springer International Publishing, 2016 (WoS, MNI_{SW}: 15) (**70%**).
 11. [Filipowicz, **Chmiel et al**, **2010**] B. Filipowicz, W. Chmiel, and P. Kadłuczka. Kontur wypukły w trójwymiarowym zagadnieniu pakowania — Convex contour for 3D bin packing problem. *Automatyka/Automatics*, 14(3):867–883, 2010 (MNI_{SW}: 6) (**60%**).
 12. [**Chmiel et al**, **2011**] W. Chmiel, P. Kadłuczka, and S. Ernst. A Multicriteria Model for Dynamic Route Planning. In A. Dziech and A. Czyżewski, editors, *Multimedia Communications, Services and Security*, pages 174–182, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg (WoS, MNI_{SW}: 10) (**60%**).
 13. [**Chmiel et al**, **2010**] W. Chmiel, P. Kadłuczka, K. Wala, and S. Jędrusik. Algorytmy heurystyczne w trójwymiarowym zagadnieniu pakowania — heuristic algorithm for three-dimensional packing problem. *Automatyka/Automatics*, 14(3/2):827–840, 2010 (MNI_{SW}: 6) (**67%**).
 14. [Filipowicz, **Chmiel et al**, **2009**] B. Filipowicz, W. Chmiel, and P. Kadłuczka. Ukierunkowane przeszukiwanie przestrzeni rozwiązań w algorytmach rojowych — guided search using particle swarm optimization algorithm. *Automatyka*, 13(2):247–255, 2009 (**70%**).

3.2. Syntetyczny opis oryginalnych wyników zawartych w cyklu publikacji

Wprowadzenie

Zasadniczym celem badań prowadzonych przez Autora było opracowanie algorytmów przybliżonych umożliwiających efektywne rozwiązywanie problemów należących do klasy dyskretnych zagadnień \mathcal{NP} -trudnych. W prowadzonych pracach badawczych skupiono się na problemie kwadratowego przydziału (*Quadratic Assignment Problem* - QAP), jednym z najtrudniejszych \mathcal{NP} -trudnych zagadnień dyskretnych.

Zagadnienie kwadratowego przydziału zostało zdefiniowane przez Koopmansa oraz Beckmana w 1957 roku, jako matematyczny model alokacji przedsięwzięć w działalności gospodarczej. Pozostaje on jak dotąd jednym z najtrudniejszych problemów optymalizacji kombinatorycznej, generalizującym wiele innych trudnych do rozwiązania zagadnień. Jak dotąd brak jest metod, które w akceptowalnym czasie, w sposób dokładny, pozwalają rozwiązać problem QAP o rozmiarze większym niż 25.

Problemy badawcze związane z zagadnieniem QAP, nie ograniczają się jedynie do uzyskania rozwiązania optymalnego. Sahni and Gonzalez [4] udowodnili, że zagadnieniem \mathcal{NP} -trudnym jest wyznaczenie dowolnej ε – *aproxymacji* problemu QAP.

Jak dotąd udowodniono istnienie algorytmów wielomianowych dla niewielu specyficznych przypadków generalizowanych przez QAP: zagadnienia gęstego liniowego uporządkowania

(*dense linear arrangement problem*) oraz problemu komiwojażera, w którym macierz współczynników spełnia nierówność trójkąta.

Problem kwadratowego przydziału znajduje zastosowanie w wielu interesujących zagadnieniach praktycznych nauki i techniki. Wśród nich można wymienić zagadnienia przydziału zadań do maszyn, planowania architektonicznego (budynków oraz ciągów komunikacyjnych), optymalizacji długości połączeń, analizy zależności ekonomicznych, wyważania łopat turbiny, określania powiązań pomiędzy informacjami zapisanymi na taśmach magnetycznych, projektowania układów VLSI oraz analizy reakcji chemicznych dla substancji organicznych.

Postać Koopmansa-Beckmana problemu QAP można sformułować następująco. Rozważmy problem przydziału zbioru zadań do zbioru maszyn, w którym całkowity koszt działania systemu zależy od odległości pomiędzy maszynami oraz od wielkości przepływów definiujących technologiczne powiązania pomiędzy zadaniami (np. transportu przetwarzanych elementów). Wprowadźmy dodatkowo koszt określający przydział zadania do określonej maszyny. Celem jest taki przydział zadań do maszyn, który minimalizuje całkowity koszt funkcjonowania systemu, związanego np. z kosztami transportu technologicznego pomiędzy zadaniami. Ścisłej mówiąc, w celu opisu tego zagadnienia definiujemy trzy $n \times n$ wymiarowe rzeczywiste macierze $F = (f_{ij})$, $D = (d_{kl})$ oraz $B = (b_{ik})$, gdzie f_{ij} jest przepływem (lub np. liczbą połączeń) pomiędzy zadaniami i oraz j , d_{kl} jest odległością pomiędzy maszyną k oraz l , natomiast b_{ik} jest kosztem związanym z przydziałem zadania i do maszyny k . Teraz możemy sformułować postać Koopmansa-Beckmana problemu QAP.

Rozwiązanie powyższego zagadnienia (zwanego również jako $\text{QAP}(F, D)$) może być zdefiniowane w postaci permutacji $\pi = (\pi(1), \dots, \pi(n))$ zbioru n elementów (zadań). Celem optymalizacji jest znalezienie takiej permutacji π^* która minimalizuje następującą funkcję celu:

$$f(\pi^*) = \min_{\pi \in S_n} \sum_{i=1}^n \sum_{j=1}^n d_{ij} f_{\pi(i), \pi(j)} + \sum_{i=1}^n b_{\pi(i), i} \quad (3.1)$$

gdzie S_n jest zbiorem permutacji liczb całkowitych $\{1, \dots, N\}$. Jeśli macierze D i F są symetryczne tj. odległość $d_{ij} = d_{ji}$ i liczba połączeń $f_{ij} = f_{ji}$ wtedy mówimy o *symetrycznym* problemie QAP

Znacząca rola problemu kwadratowego przydziału wynika również faktu, że zagadnienie to generalizuje wiele ważnych zagadnień teoretycznych, jak: problem podziału grafu (*graph partitioning problem* - GPP), problem maksymalnej klikli (*maximal clique problem* - MCP), problem komiwojażera (*travelling salesman problem* - TSP), linowy problem ułożenia (*linear arrangement problem* - LAP), problem minimalnej wagi zbioru łuków sprzężonych (*minimum weight feedback arc set* - FASP), problem pakowania grafów (*graph packing* - GPP).

W związku z tym, zakres stosowalności cech zagadnienia QAP badanych przez Autora wykracza daleko poza to zagadnienie i dotyczy wielu innych ważnych teoretycznych i praktycznych problemów generalizowanych i modelowanych przez to zagadnienie.

Prócz badania teoretycznych własności zagadnienia QAP, Autor prowadził także badania dotyczące realizacji algorytmów przybliżonych, w głównej mierze algorytmów populacyjnych inspirowanych naturą, na platformach sprzętowych opartych na GPU (*Graphical Processing Unit*). Doświadczenia zebrane w zakresie implementacji algorytmów inspirowanych naturą znalazły swoje zastosowanie także w mechanizmach wspomagających zarządzanie samolotami bezzałogowymi (*Unmanned Aerial Vehicle* - UAV)

Wiele prac badawczych Autora znalazło zastosowanie w praktycznych rozwiązaniach wdrożonych w ramach realizowanych grantów naukowych.

Opis oryginalnych wyników

3.2.1. Opracowanie mechanizmów bazujących na zaproponowanej i udowodnionej przez Autora formule umożliwiającej określenie warunkowej wartości oczekiwanej funkcji celu dla zagadnienia kwadratowego przydziału

Autor zaprojektował, zaimplementował i przeprowadził badania eksperymentalne w celu oceny możliwości zastosowania zaproponowanej formuły wyznaczającej warunkową wartość oczekiwaną funkcji celu dla rozwiązania częściowo ustalonego w konstrukcji przybliżonych algorytmów optymalizacyjnych takich jak: algorytm pszczele (*Bees Algorithm* - BA) oraz algorytm ewolucyjny (*Evolutionary Algorithm* - EA).

Przeprowadzone zostały obszernie i gruntownie badania zaproponowanych algorytmów z zastosowaniem zróżnicowanych instancji testowych dla zagadnienia QAP. Badania te obejmują zarówno ocenę wydajności, efektywności jak również testy nieparametryczne umożliwiające stwierdzenie, czy zastosowana metoda pozwoliła na uzyskanie statystycznie znaczącej różnicy w jakości uzyskanych wyników.

Poniższe dwa twierdzenia dotyczą średniej wartości funkcji oraz warunkowej wartości oczekiwanej funkcji celu dla problemu QAP (dalej dla wygody zwanej CEV od skrótu angielskiego: *Conditional Expectation Value for the QAP problem*). Dowody obydwu twierdzeń (pominięte tutaj ze względu na ich obszerność) zostały zaprezentowane w publikacji [4].

Theorem 1. Średnia wartość funkcji celu dla problemu QAP

Niech $\varphi : S_n \rightarrow \mathbb{R}$ będzie zdefiniowana następująco:

$$\varphi(\pi) = \sum_{i=1}^n \sum_{j=1}^n d_{ij} f_{\pi(i)\pi(j)} \quad (3.2)$$

gdzie $D = [d_{ij}]_{n \times n}$ and $F = [f_{ij}]_{n \times n}$ są macierzami kwadratowymi. Wtedy:

$$\bar{\varphi} = \frac{1}{n(n-1)} \sum_{1 \leq i \neq j \leq n} d_{ij} \sum_{1 \leq k \neq l \leq n} f_{kl} + \frac{1}{n} \sum_{i=1}^n d_{ii} \sum_{i=1}^n f_{ii} \quad (3.3)$$

□

Niech φ będzie pewną funkcją taką, że $\varphi: S_n \rightarrow \mathbb{R}$. Rozważmy następujący problem optymalizacyjny:

$$\varphi(\pi) \rightarrow \min, \pi \in S_n. \quad (3.4)$$

Zdefiniujmy $\pi_{i_1}^0, \dots, \pi_{i_k}^0 \in \{1, \dots, n\}$, gdzie $1 \leq i_s \neq i_t \leq n, 1 \leq s < t \leq k \leq n$. Niech $S_n(\pi_{i_1}^0, \dots, \pi_{i_k}^0) = \{(\pi(1), \pi(2), \dots, \pi(n)) \in S_n : \pi(i_1) = \pi_{i_1}^0, \dots, \pi(i_k) = \pi_{i_k}^0\}$ i dla $k = 0 : S_n(\pi_{i_1}^0, \dots, \pi_{i_k}^0) = S_n$. Problem optymalizacyjny w postaci (3.4) może być interpretowany jako zmienna losowa, poprzez przypisanie każdemu elementowi $\pi^0 = \{\pi_1^0, \dots, \pi_n^0\}$ z S_n nieujemnej wartości - prawdopodobieństwa $P(\pi^0)$, takiego że $\sum_{\pi^0 \in S_n} P(\pi^0) = 1$.

Przy takiej interpretacji φ jest zmienną losową, dzięki czemu możemy posługiwać się wartością oczekiwaną $E[\varphi]$ funkcji φ , a także prawdopodobieństwem warunkowym.

Załóżmy, że $P(\pi \in S_n) = \frac{1}{n!}$ oraz, że warunkowa wartość funkcji celu jest w postaci $E(\varphi | \pi(s_1) = c(s_1), \dots, \pi(s_i) = c(s_i), \dots, \pi(s_k) = c(s_k))$, gdzie $c(s)$ to numer obiektu przypisanego do pozycji s .

Theorem 2. Warunkowa wartość oczekiwana funkcji celu problemu QAP.

Niech $\varphi : S_n \rightarrow \mathbb{R}$ będzie zdefiniowana następująco:

$$\varphi(\pi) = \sum_{i=1}^n \sum_{j=1}^n d_{ij} f_{\pi(i)\pi(j)} \quad (3.5)$$

gdzie $D = [d_{ij}]_{n \times n}$ i $F = [f_{ij}]_{n \times n}$ będą kwadratowymi macierzami opisującymi odpowiednio odległości między pozycjami oraz liczbę połączeń pomiędzy obiektami. Określny zbiór liczb naturalnych zakresu 1 do n jako $L = \{1, 2, \dots, n\}$ i zbiór $U = \{1, \dots, n\} \setminus \{c(s_1), \dots, c(s_i), \dots, c(s_k)\}$ jako zbiór nieprzydzielonych obiektów, gdzie $c(s_1), \dots, c(s_i), \dots, c(s_k)$ są różnymi wartościami ze zbioru L i $H = \{s_1, \dots, s_i, \dots, s_k\}$, natomiast $1 \leq s_1, \dots, s_i, \dots, s_k \leq n$ są ustalonymi pozycjami. W takim przypadku warunkowa wartość oczekiwana częściowo ustalonego rozwiązania problemu QAP, przy k dowolnie ustalonych pozycjach:

$$\begin{aligned}
E(\varphi | \pi(s_1) = c(s_1), \dots, \pi(s_i) = c(s_i), \dots, \pi(s_k) = c(s_k)) = \\
\sum_{i \in H} \sum_{j \in H} d_{ij} f_{c(i)c(j)} + \frac{1}{n-k} \sum_{i \in H} \sum_{j \in L \setminus H} d_{ij} \sum_{u \in U} f_{c(i)u} + \\
\frac{1}{n-k} \sum_{i \in L \setminus H} \sum_{j \in H} d_{ij} \sum_{u \in U} f_{uc(j)} + \frac{1}{n-k} \sum_{i \in L \setminus H} d_{ii} \sum_{u \in U} f_{uu} + \\
\frac{1}{n-k} \frac{1}{n-k-1} \sum_{i, j \in L \setminus H, i \neq j} d_{ij} \sum_{o, l \in U, o \neq l} f_{ol}
\end{aligned} \tag{3.6}$$

□

W celu przebadania możliwości zastosowania zaproponowanej formuły wyznaczającej warunkową wartość oczekiwaną funkcji celu dla problemu QAP, zaprojektowano 25 operatorów pseudo-genetycznych, które korzystają z niej w swym działaniu. Bazują one na standardowych operatorach stworzonych dla problemów permutacyjnych: PMX (*Partially Matched Crossover*), OX (*Order Crossover*), POX (*Position Based Crossover*) i MUT-S (*Scramble Mutation*). Aby algorytm ewolucyjny uzyskiwał rozwiązania charakteryzujące się wysoką jakością, konieczne jest odpowiednie połączenie mechanizmów odpowiedzialnych za eksplorację (poszukiwanie nowych obszarów przestrzeni rozwiązań) z mechanizmami odpowiedzialnymi intensyfikację (dokładniejsze przeglądanie już zbadanych obszarów przestrzeni rozwiązań). W związku z tym, aby było możliwe świadomie sterowanie działaniem algorytmu przybliżonego, zaproponowano różne konstrukcje operatorów, które można podzielić na trzy zasadnicze grupy.

Pierwsza grupa to operatory odpowiedzialne za intensyfikację procesu przeszukiwania przestrzeni, zwiększające zbieżność algorytmu ewolucyjnego: $(PMX1_{CE}^I, PMX2_{CE}^I, PMX3_{CE}^I, OX_{CE}^I, POX1_{CE}^I, POX2_{CE}^I, POX3_{CE}^I, MUT1_{CE}^I, MUT2_{CE}^I, MUT3_{CE}^I)$.

Druga grupa to operatory wspomagające proces eksploracji, zmniejszające zbieżności algorytmu: $(PMX1_{CE}^E, PMX2_{CE}^E, PMX3_{CE}^E, OX_{CE}^E, POX1_{CE}^E, POX2_{CE}^E, POX3_{CE}^E, MUT1_{CE}^E, MUT2_{CE}^E)$.

Ostatni typ operatorów to operatory łączące obie cechy, intensyfikująco-różnicujące, w których oba te mechanizmy zostały połączone: $(PMX1_{CE}^{IE}, PMX2_{CE}^{IE}, POX1_{CE}^{IE}, POX2_{CE}^{IE}, MUT1_{CE}^{IE}, MUT2_{CE}^{IE})$. Odpowiednio dobierając prawdopodobieństwo wywołania w algorytmie powyższych operatorów, w zależności od cech optymalizowanej instancji, można kłaść nacisk na eksplorację bądź eksploatację przestrzeni rozwiązań. Zastosowanie operatorów opartych o wartość oczekiwaną funkcji celu wymaga doboru odpowiedniego parametru określającego liczbę losowo wybranych sekcji kojarzenia. W celu ilustracji, poniżej przedstawiono opisany powyżej proces dla jednego, wybranego operatora intensyfikującego $PMX1_{CE}^I$ opracowanego na bazie standardowego pseudo-genetycznego operatora PMX . W operatorze tym losowo wybiera się k sekcji kojarzenia. Dla każdej sekcji wymiany określa się sumę wartości CEV dla obu rodziców L_i , ($i \in 1, 2, \dots, k$). Następnie do krzyżowania wybiera się tę sekcję, która charakteryzuje się najmniejszą wartością sumy L_i . Poniżej przedstawiono przykład działania tego operatora dla dwóch wybranych rodziców, będących rozwiązaniami problemu QAP i wartości $k = 3$:

$$\begin{aligned}\pi_1 &= (1, 2, 3, 4, 5, 6, 7) \\ \pi_2 &= (5, 4, 7, 1, 2, 3, 6)\end{aligned}$$

Losowo wybrane sekcje kojarzenia (znak '*' określa sekcję kojarzenia) określone są następująco:

$$\begin{aligned}(|*, *, *|, -, -, -, -) \\ (-, -, -, |*, *, *|, -) \\ (-, |*, *|, -, -, -, -)\end{aligned}$$

Sumując wartości CEV dla częściowo ustalonych rozwiązań :

$$L_1 = E((1, 2, 3, -, -, -, -)) + E((5, 4, 7, -, -, -, -)) = 20 + 25 = 45$$

$$L_2 = E((-, -, -, 4, 5, 6, -)) + E((-, -, -, 1, 2, 3, -)) = 12 + 30 = 42$$

$$L_3 = E((-, 2, 3, -, -, -, -)) + E((-, 4, 7, -, -, -, -)) = 17 + 30 = 47$$

Ponieważ $L_2 < L_1 < L_3$, to wybrana zostaje druga sekcja kojarzenia(L_2):

$$\begin{aligned}\tilde{\pi}_1 &= (-, -, -, 4, 5, 6, -) \\ \tilde{\pi}_2 &= (-, -, -, 1, 2, 3, -)\end{aligned}$$

Sekcje kojarzenia są wymieniane pomiędzy rodzicami, ale niektóre wartości u potomków występują więcej niż jeden raz, co jest niedopuszczalne, gdyż rozwiązaniem problemu jest permutacja. Powtarzające się wartości poza sekcją kojarzenia, są określane mianem dziur, które są wypełniane na podstawie mapowania określonego na podstawie sekcji kojarzenia:

$$\begin{aligned}\tilde{\pi}_1 &= (\mathcal{B}, \mathcal{A}, 7, 4, \mathbf{5}, \mathbf{6}, \mathcal{C}) \\ \tilde{\pi}_2 &= (\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathbf{1}, \mathbf{2}, \mathbf{3}, 7)\end{aligned}$$

Mapowanie określane na podstawie sekcji kojarzenia jest zdefiniowane następująco:

$$\begin{aligned}\tilde{\pi}_1 : 4 \mapsto 1, 5 \mapsto 2, 6 \mapsto 3 \\ \tilde{\pi}_2 : 1 \mapsto 4, 2 \mapsto 5, 3 \mapsto 6\end{aligned}$$

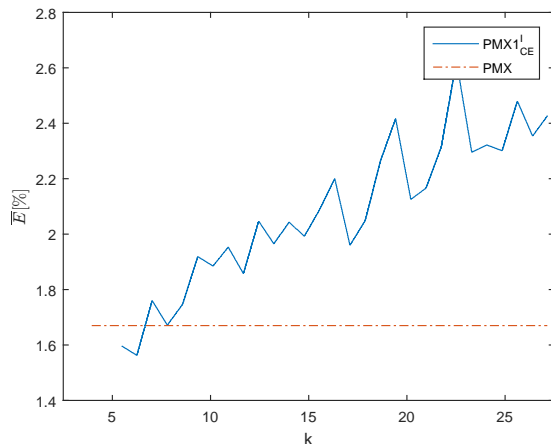
Po zaaplikowaniu mapowania na podstawie sekcji kojarzenia $\tilde{\pi}_1$ i $\tilde{\pi}_2$, otrzymujemy dwóch potomków:

$$\begin{aligned}\pi_1^\circ &= (2, 1, 7, 4, 5, 6, 3) \\ \pi_2^\circ &= (4, 5, 6, 1, 2, 3, 7)\end{aligned}$$

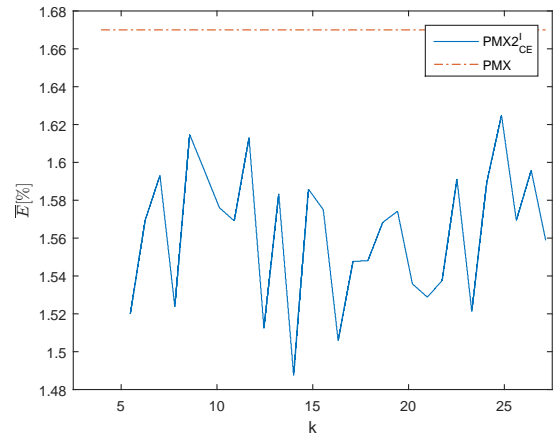
Na Rys. 3.1 oraz Rys. 3.2 przedstawiono wpływ parametru k na uzyskanie wyniku poprzez zastosowanie operatorów $PMX1_{CE}^I$ oraz operator $PMX2_{CE}^I$ (niebieska linia) i porównano je z wynikami uzyskanymi poprzez zastosowanie standardowego operatora PMX (linia czerwona przerywana). Na przedstawionych wykresach parametr $\bar{E} = (\sum_{i=1}^L \frac{\varphi_{best_i} - \varphi_{ref}}{\varphi_{ref}} \cdot 100\%) / L$ – oznacza względny średni procentowy błąd dla L uruchomień algorytmu dla wybranej instancji testowej, gdzie $L = 36$. Aby przeprowadzić testy zgodnie z metodyką testów algorytmów przybliżonych, dla każdego z sześciu uruchomień dla różnego posiewu generatora liczb losowych, algorytm był uruchamiany sześciokrotnie z różną populacją startową.

Jak widać na rysunkach, dla $k \leq 6$ operator $PMX1_{CE}^I$ daje lepsze wyniki niż standardowy operator PMX . Natomiast operator $PMX2_{CE}^I$ osiąga lepsze wyniki od operatora PMX dla każdej wartości k .

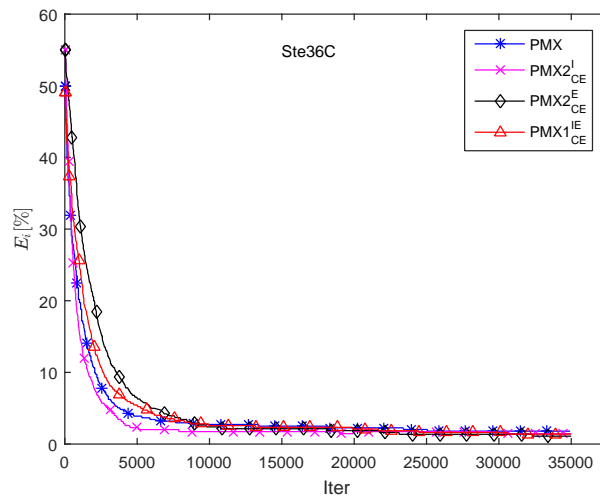
Na Rys. 3.3 zaprezentowano przebieg zmian wartości E_i ($E_i = \frac{\varphi_{best_i} - \varphi_{ref}}{\varphi_{ref}} \cdot 100\%$ – procentowy błąd względny) dla czterech różnych operatorów PMX , $PMX2_{CE}^I$ (operator intensyfikujący), $PMX2_{CE}^E$ (operator eksplorujący), $PMX1_{CE}^E$ (operator intensyfikująco-eksplorujący) na przykładzie instancji Ste36C problemu QAP. Można zauważyć, że E_i maleje najszybciej dla operatora intensyfikującego $PMX2_{CE}^I$, natomiast w przypadku operatora



Rysunek 3.1: Zależność pomiędzy względnym procentowym błędem \bar{E} dla operatora $PMX1_{CE}^I$ i wartości parametru k dla instancją Lipa30a.



Rysunek 3.2: Zależność pomiędzy względnym procentowym błędem \bar{E} dla operatora $PMX2_{CE}^I$ i wartości parametru k dla instancją Lipa30a.



Rysunek 3.3: Przebieg wartości E_i dla algorytmu QAPEV stosującego operatory PMX , $PMX2_{CE}^I$, $PMX2_{CE}^E$, $PMX1_{CE}^I$ podczas poszukiwania najlepszego rozwiązania dla instancji Ste36c.

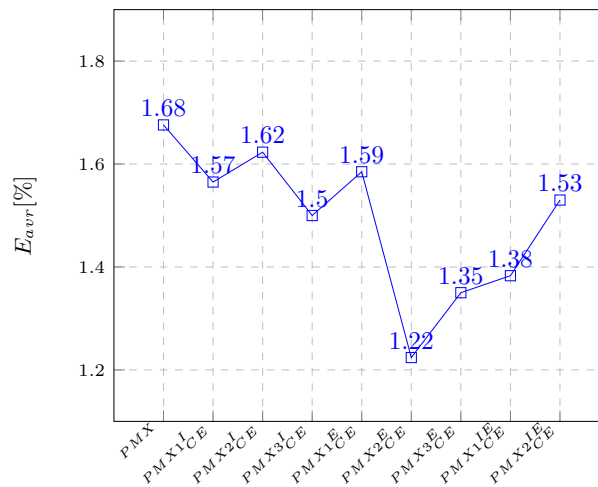
$PMX2_{CE}^E$ wartość błędem spada najwolniej, ale osiąga on najlepszy wynik końcowy (najmniejszy błąd końcowy, $E_i = 1.1\%$). Można przypuszczać, że wynik ten uzyskano dzięki cechom eksploracyjnym tego operatora.

Tabela 3.1 przedstawia E_{avr} , E_{min} , E_{max} oraz wartość standardowego odchylenia σ wartości E_{avr} . Rys. 3.4 przedstawia względny błąd procentowy E_{avr} .

Wyniki w tabeli uzyskano za pomocą algorytmu QAPEV, który wykorzystuje tylko jeden typ operatora krzyżowania spośród operatorów: PMX , $PMX1_{CE}^I$, $PMX2_{CE}^I$, $PMX3_{CE}^I$, $PMX1_{CE}^E$, $PMX2_{CE}^E$, $PMX3_{CE}^E$, $PMX1_{CE}^{IE}$ and $PMX2_{CE}^{IE}$, szczegółowo opisanych w [4]. Analiza tabeli pozwala zauważyć, że wartość E_{avr} dla operatora PMX jest równa 1.68% i w każdym przypadku jest ona większa niż w przypadku operatorów stosujących formułę pozwalającą na wyliczenie warunkowej wartości funkcji celu. Najlepszy wynik osiągnął operator genetyczny $PMX2_{CE}^E$ (1,22 %), który wzmacnia cechy eksploracyjne standardowego

Tablica 3.1: Uzyskane wartości E_{avr} , E_{best} , E_{worst} i odchylenia standardowego σ dla operatorów bazujących na operatorze PMX dla 38 instancji problemu QAP.

Operator	E_{avr} [%]	E_{best} [%]	E_{worst} [%]	σ
PMX	1.680	1.477	1.805	0.091
$PMX1_{CE}^I$	1.565	1.387	1.674	0.078
$PMX2_{CE}^I$	1.623	1.411	1.736	0.083
$PMX3_{CE}^I$	1.500	1.322	1.605	0.077
$PMX1_{CE}^E$	1.585	1.408	1.699	0.078
$PMX2_{CE}^E$	1.224	1.069	1.314	0.065
$PMX3_{CE}^E$	1.350	1.189	1.443	0.070
$PMX1_{CE}^{IE}$	1.383	1.225	1.479	0.069
$PMX2_{CE}^{IE}$	1.530	1.345	1.645	0.084



Rysunek 3.4: Średnia procentowa wartość błędu E_{avr} dla operatorów pseudo-genetycznych bazujących na operatorze genetycznym PMX dla 38 instancji problemu QAP.

operatora pseudo-genetycznego PMX .

Analogiczne badania zostały przeprowadzone dla pozostałych 24 zaproponowanych operatorów pseudo-genetycznych. Badania te zaprezentowane są w [4].

Tabela 3.2 przedstawia porównanie wyników uzyskanych przez algorytm QAPEV przy użyciu standardowych operatorów pseudo-genetycznych z wynikami uzyskanymi przez najbardziej skuteczne operatory pseudo-genetyczne wykorzystujące CEV. Tabela ta zawiera wartości parametru k , które zapewniają najlepszy wynik optymalizacji dla konkretnej instancji QAP i operatora pseudo-genetycznego. Wartości k przedstawione w Tabeli 3.2 zostały uzyskane na podstawie podobnych testów, jak te zaprezentowane na rysunkach 3.1 i 3.2. Przykładowo, jeśli algorytm QAPEV używa operatora $PMX2_{CE}^E$ do optymalizacji instancji Bur26a, wówczas parametr k powinien przyjąć wartość 12. Rys. 3.5 przedstawia E_{avr} , E_{best} , E_{worst} , σ w formie, która jest lepsza dla analizy wyniku. Analiza rysunku pozwala zauważyć, że operatory $PMX2_{CE}^E$ i $POX2_{CE}^E$ osiągnęły najmniejszą wartość E_{avr} i najmniejszą wartość odchylenia standardowego.

Aby potwierdzić przewagę operatorów bazujących na CEV nad ich wersjami standardowymi, przeprowadzono nieparametryczny jednostronny test Wilcozona. Wyniki przedstawione w tabeli 3.3 potwierdzają, że operator $PMX2_{CE}^E$ statystycznie osiąga znacznie lepsze wyniki niż operator PMX , ponieważ p -value = 0.00149 dla $\alpha = 0.05$. Podobnie w przypadku OX_{CE}^I hipotezę zerową H_0 można odrzucić, przyjmując alternatywną hipotezę, że istnieją istotne róż-

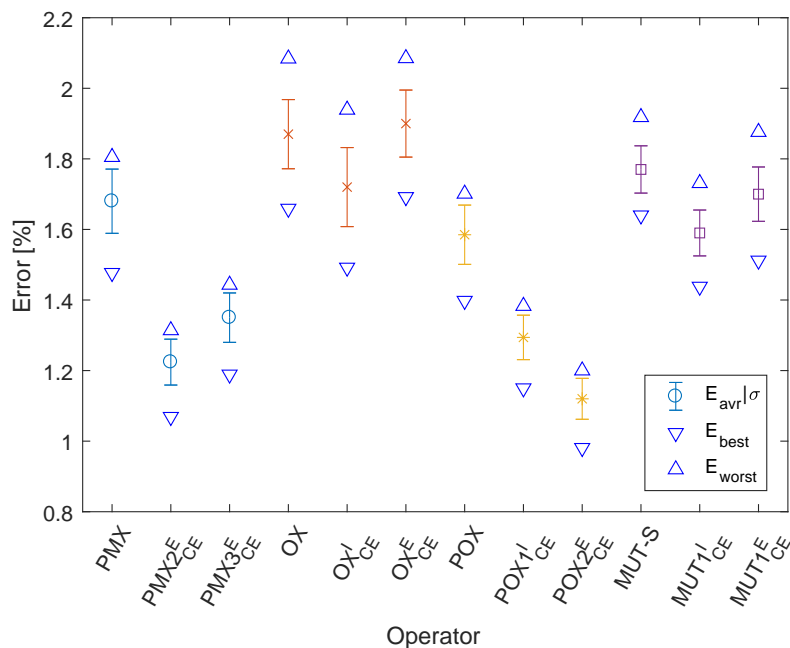
Tablica 3.2: Wybrane najlepsze rezultaty algorytmu @APEV stosującego standardowe operatory pseudo-genetyczne i operatory bazujące na wartości CEV oraz wartości k dla każdego operatora pseudo-genetycznego zapewniające najlepsze rezultaty.

	PMX	PMX 2_{CE}^E	PMX 3_{CE}^E	OX	OX $_{CE}^E$	OX $_{CE}^I$	POX	POX $_{CE}^I$	POX $_{CE}^E$	MUTS	MUT 1_{CE}^I	MUT 1_{CE}^E
E_{avr}	1.680	1.224	1.350	1.870	1.720	1.900	1.585	1.294	1.120	1.770	1.590	1.700
E_{min}	1.477	1.069	1.189	1.659	1.492	1.692	1.398	1.150	0.981	1.640	1.438	1.512
E_{max}	1.805	1.314	1.443	2.084	1.939	2.085	1.701	1.383	1.200	1.918	1.731	1.876
σ	0.091	0.065	0.070	0.098	0.112	0.095	0.084	0.063	0.058	0.067	0.065	0.077
						k						
Bur26a	10	7	5	11	8	10	15	8	10	4	9	9
Bur26b	11	8	6	9	10	15	8	10	15	5	8	8
Bur26c	10	9	8	8	8	11	11	8	11	6	10	10
Bur26d	9	9	6	10	9	18	6	9	18	6	11	11
Bur26e	12	8	7	8	7	15	7	7	15	7	12	12
Bur26f	10	12	5	8	9	10	6	9	10	6	11	11
Bur26g	8	11	4	7	8	10	6	8	10	6	9	9
Bur26h	8	12	9	9	7	12	5	7	12	5	12	12
Chr22a	7	6	7	6	10	6	4	10	6	4	7	7
Chr22b	10	7	6	7	10	5	5	10	5	5	9	9
Esc32a	9	12	9	7	7	20	11	7	20	11	14	14
Esc32b	12	9	10	9	6	17	11	6	17	11	11	11
Esc32c	11	10	7	9	8	19	10	7	19	10	7	7
Esc32d	10	9	6	7	6	9	9	6	9	9	9	9
Esc32e	11	8	9	8	9	18	11	8	18	11	9	9
Esc32f	9	7	11	10	10	16	12	9	16	12	13	13
Esc32g	7	8	12	7	12	7	10	10	17	10	12	12
Esc32h	10	7	10	11	7	21	13	7	21	13	11	11
Esc64a	35	6	19	24	18	29	10	18	29	10	28	28
Kra30a	11	10	16	12	10	11	12	10	11	12	13	13
Kra30b	9	12	11	15	11	11	10	11	11	10	16	16
Lipa30a	14	8	12	12	7	22	9	7	22	9	9	9
Lipa30b	13	11	16	11	10	20	7	10	20	7	7	7
Lipa40a	21	23	17	23	12	15	19	12	15	19	20	20
Lipa40b	19	5	15	20	10	18	21	10	18	21	22	22
Lipa50a	14	21	16	21	13	28	24	13	28	24	18	18
Lipa50b	16	20	19	18	10	19	22	10	19	22	22	22
Lipa60a	33	25	21	24	7	32	19	7	32	19	34	34
Lipa60b	27	29	24	26	10	34	17	10	34	17	31	31
Sko42	5	21	24	14	23	26	19	23	26	19	26	26
Sko49	17	7	21	15	9	18	28	9	18	28	22	22
Sko56	15	25	19	24	22	17	16	22	17	16	28	28
Ste36a	4	14	7	11	6	24	11	6	24	11	12	12
Ste36b	5	15	8	11	7	27	7	7	27	7	9	9
Ste36c	4	14	7	8	7	30	6	7	30	6	10	10
Tho30	11	12	10	14	13	18	16	13	18	16	18	18
Tho40	20	19	13	15	14	32	19	14	32	19	23	23
Wli50	29	15	18	19	5	10	21	5	10	21	28	28

Tablica 3.3: Rezultaty nieparametrycznego testu Wilcoxon'a dla opracowanych operatorów pseudo-genetycznych.

Operators	p -value
PMX vs PMX 2_{CE}^E	0.00149
OX vs OX $_{CE}^I$	0.00587
POX vs POX 2_{CE}^E	0.03673
MUTS vs MUT 1_{CE}^I	0.13786

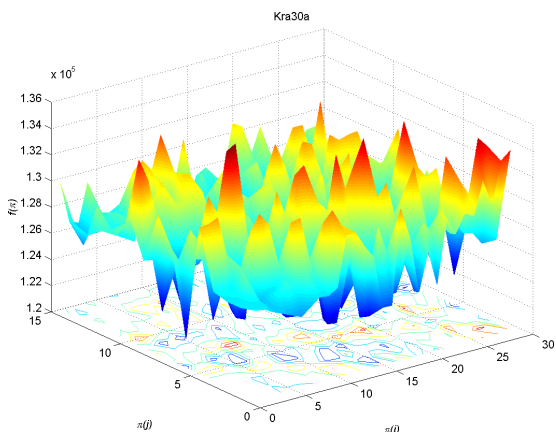
nice pomiędzy uzyskanymi wynikami, ponieważ p -value = 0.00587 dla $\alpha = 0.05$. Co więcej, operator POX 2_{CE}^E daje statystycznie lepsze wyniki niż operator POX, ponieważ p -value = 0.0367 przy $\alpha = 0.05$. Nie ma jednak statystycznych różnic między algorytmem, który używa operatora MUT 1_{CE}^I lub operatora MUT-S, ponieważ p -value = 0.13786 dla $\alpha = 0.05$ (hipoteza H_0 nie może zostać odrzucona). W ujęciu statystycznym nie ma różnic między wynikami uzyskanymi z pomocą obu tych operatorów mutacji.



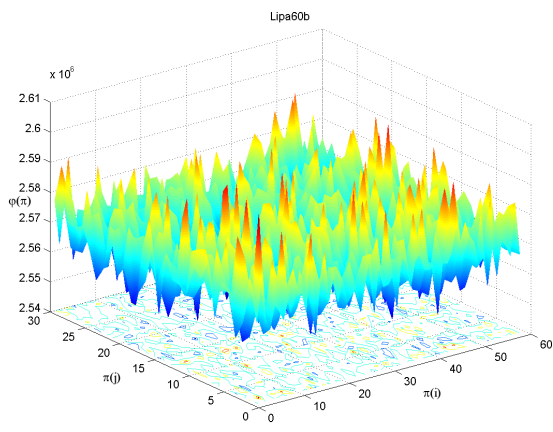
Rysunek 3.5: Najlepsze uzyskane wyniki ze względu na wartość błędu E_{avr} , E_{best} , E_{worst} i odchylenia standardowego σ dla operatorów bazujących na CEV i standardowych operatorach PMX , OX , POX i $MUT-S$ dla 38 instancji problemu QAP.

3.2.2. Opracowanie mechanizmów poprawiających wydajność algorytmów przybliżonych dla \mathcal{NP} -trudnych zagadnień optymalizacyjnych

Poszukiwanie dobrej jakości rozwiązań problemu QAP, już nawet dla instancji niewielkich rozmiarów sprawia poważne trudności. Wynikają one m.in. z wielomodalności, a więc dużej liczby minimów lokalnych. Cechę tę ilustrują krajobrazy przedstawione na Rys. 3.6 i Rys. 3.7 dla dwóch instancji problemu QAP, $Lipa60b$ i $Kra30$. W związku z tym pokłada się duże nadzieje w zastosowaniu algorytmów populacyjnych pozwalających jednocześnie przetwarzać wiele rozwiązań.



Rysunek 3.6: Przykład krajobrazu dla instancji $Kra30a$ problemu QAP i $2-opt$ struktury sąsiedztwa.



Rysunek 3.7: Przykład krajobrazu dla instancji $Lipa60b$ problemu QAP i $2-opt$ struktury sąsiedztwa.

W ramach badań Autor przeprowadził analizę wyników uzyskiwanych przez metodę dokładną oraz dwa inspirowane naturą algorytmy optymalizacji dyskretnej: algorytm pszczeleli (*Bees Algorithm* - BA), algorytm mrówkowy (*Ant Colony Algorithm* - ACO). Szczegółowo opisano to w [11]. Określono wpływ na uzyskane wyniki stosowanego przez oba algorytmy typu otoczenia. Zbadano następujące typy otoczeń:

- $2-opt$ powstałe poprzez wymianę par, zawierające $n(n-1)/2 + 1$ permutacji:

$$2 - OPT = \{\pi \in S_n \mid \pi \circ (i, j) : 1 \leq i, j \leq n; i \neq j\}$$

- otrzymane poprzez ustalenie wszystkich elementów rozwiązania na pozycjach nieparzystych, usunięcie elementów z pozycji parzystych, a następnie wstawienie usuniętych elementów na puste pozycje w dowolnej kolejności, zawierające $(n/2)!$ permutacji:

$$ASSIGN_n = \left\{ \pi \in S_n \mid \pi(2i-1) = 2i-1, i = 1, 2, \dots, \frac{n}{2} \right\},$$

- zdefiniowane analogicznie jak $ASSIGN_n$ z ustalonymi pozycjami parzystymi n :

$$TWIN_n = \left\{ \pi \in S_n \mid \pi(2i-1) = 2i-1 \wedge \pi(2i) = 2i, i = 1, 2, \dots, \frac{n}{2} \right\}$$

- otrzymane poprzez odpowiedni przydział kolejnych elementów w permutacji rosnąco, a następnie malejąco:

$$PYRAMID_n = \left\{ \pi \in S_n \mid i_1, i_2, \dots, i_k, s, j_1, j_2, \dots, j_{n-k-1}; k \geq 0, \right. \\ \left. i_1 < i_2 < \dots < i_k, j_1 >, j_2 > \dots > j_{n-k-1} \right\}$$

- otrzymane poprzez złożenie dwóch permutacji typu $PYRAMID_n$ i α (permutacja powstała poprzez rotację elementów od 1 do n):

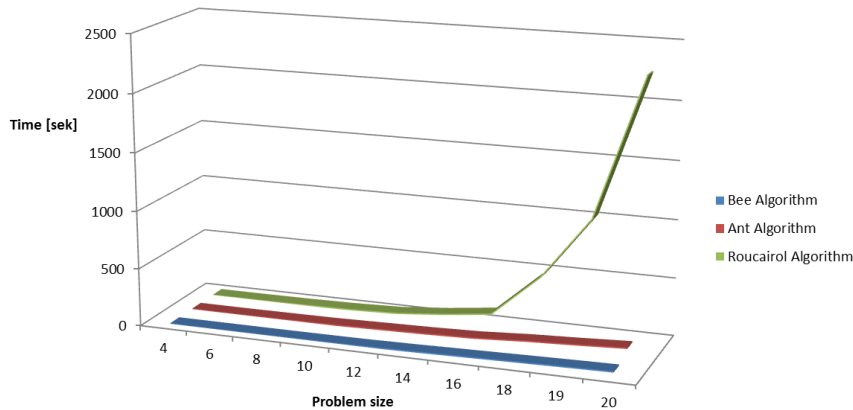
$$PYRAMID - CV_n = \left\{ \pi \in S_n \mid \pi_{Pyramid} \circ \alpha; \alpha = \{ \alpha \in S_n \mid (k, k+1, \dots, n, \dots, \right. \\ \left. 1, \dots, k-1) \}, k = 1, \dots, n \right\}$$

- otoczenie $TWISTED_n$, gdzie nowe permutacje są tworzone poprzez podział permutacji na sekcje. Następnie każda sekcja jest odwracana w kolejności losowej. Tego typu otoczenie to może być reprezentowane przez drzewo permutacji i zawiera co najwyżej $O(6^n)$ permutacji:

$$TWISTED_n = \left\{ \pi \in S_n \mid \pi \in TREE(T) \wedge id \in TREE(T) \wedge \right. \\ \left. \forall v_i \mid posterity(v_i) = d : \Psi(v_i) = \{id_d, id_d^-\} \right\}.$$

Badania przeprowadzone na reprezentatywnym zestawie instancji testowych zaczerpniętych z biblioteki testowej QAPLIB wykazały, że otoczenie $2-opt$ pozwala uzyskać najlepsze wyniki optymalizacji dla obu algorytmów. Wiarygodne wyniki uzyskano poprzez zapewnienie porównywalnych warunków pracy dla obu algorytmów oraz poprzez zastosowanie takiej samej metodyki testów i identycznego zbioru instancji testowych. Należy podkreślić jednorodny sposób liczenia iteracji w obu algorytmach, co było trudne do osiągnięcia w przypadku porównania wyników uzyskanych przez różnych autorów. Porównano zależności czasu realizacji i jakości uzyskiwanych rozwiązań przez algorytm pszczeleli, algorytm mrówkowy oraz algorytm bazujący na metodzie podziału i ograniczeń (*Branch&Bound*), pozwalający na uzyskanie rozwiązań optymalnych, zaproponowany dla problemu QAP przez C. Roucairol. Przeprowadzone badania eksperymentalne wykazały przewagę algorytmu pszczelego nad algorytmem mrówkowym. Ze względu na złożoność problemu i ograniczenia czasowe ograniczono się do instancji

o maksymalnym rozmiarze $n = 20$. Rys. 3.8 przedstawia wyniki uzyskiwane przez trzy testowane algorytmy. Na podstawie wyników przedstawionych na rysunku można sformułować wniosek natury ogólnej, iż obecnie algorytmy przybliżone inspirowane naturą pozwalają na uzyskanie rozwiązania wysokiej jakości, często optymalnego, w dużo krótszym czasie niż metody dokładne. Dla instancji o rozmiarze $n = 20$, algorytm pszczelel uzyskuje rozwiązanie optymalne 100 razy szybciej, niż algorytm dokładny bazujący na metodzie B&B.



Rysunek 3.8: Zależność między czasem obliczeń a wielkością problemu dla algorytmu pszczelego, algorytmu mrówkowego i algorytmu B&B C. Roucairol.

W ramach badań związanych z zastosowaniem wartości oczekiwanej do poszukiwania rozwiązań zagadnienia QAP, zaproponowano i poddano badaniom eksperymentalnym stosującą ją algorytm pszczelel (Algorytm 1) [21].

Algorithm 1: Algorytm BA.

Dane wejściowe: $\lambda, I_{max}, l_e, l_b, n_e, n_b, \varphi(\cdot), LT$.

Krok 1. Utwórz populację zawierającą λ losowych rozwiązań:

1. Utwórz losowo λ rozwiązań.
2. Oceń jakość rozwiązań i posortuj członków roju od najlepszych do najgorszych.
3. Zapamiętaj aktualnie najlepsze rozwiązanie w roju, jeśli jest ono lepsze od najlepszego dotąd znalezionego przez algorytm $\pi_{best} = \arg \min\{\varphi(\pi^i)\}, i = 1, \dots, \lambda$.

Krok 2. Określ elitę i zbiór najlepszych rozwiązań : $L_e : |L_e| = l_e$ i $L_b : |L_b| = l_b$:

1. Utwórz otoczenie stosując procedury *swap* lub *shift*: $\forall \pi \in L_e : N(\pi)$ gdzie $|N(\pi)| = n_e$ i $\forall \pi \in L_b : N(\pi)$, gdzie $|N(\pi)| = n_b$ i wylicz wartość funkcji celu dla wszystkich rozwiązań.
2. Dla każdego z rozwiązań ze zbioru $\{L_e \cup L_b\}$ wybierz najlepsze rozwiązania z utworzonego otoczenia: $\forall \pi \in L_e : \pi^* = \arg \min_{\pi' \in \{N(\pi) \cup \pi\}} \varphi(\pi')$ i $\forall \pi \in L_b : \pi^* = \arg \min_{\pi' \in \{N(\pi) \cup \pi\}} \varphi(\pi')$.

Krok 3. Utwórz nowy rój:

1. Zastąp rozwiązania ze zbioru $\{L_e \cup L_b\}$ przez nowe utworzone w kroku **Krok 2.2**.
2. Usuń μ rozwiązań, które istnieją w roju dłużej niż LT (czas życia) iteracji.
3. Utwórz losowo μ nowych rozwiązań, aby zachować rozmiar populacji.
4. Posortuj członków roju od najlepszych do najgorszych.

Krok 4. Zapamiętaj najlepsze rozwiązanie : $\pi_{best} = \arg \min\{\varphi(\pi^i)\}, i = 1, \dots, \lambda$.

Krok 5. Test warunku stopu:

1. Jeśli wykonano I_{max} iteracji, to zwróć π_{best} i $\varphi(\pi_{best})$.
 2. W p.p. przejdź do kroku **Krok 2**.
-

Algorytm BA posiada następujące parametry: I_{max} - maksymalna liczba iteracji, π_{best} - najlepsze dotąd znalezione rozwiązanie, λ - rozmiar roju, $\varphi(\cdot)$ - wartość funkcji celu, l_e - liczba elitarnych lokalizacji (liczba rozwiązań w elicie), l_b - liczba dobrych lokalizacji (liczba

dobrych rozwiązań), n_e - rozmiar sąsiedztwa dla elitarniej lokalizacji, n_b - rozmiar sąsiedztwa dla dobrej lokalizacji, LT - długość życia rozwiązania, μ - liczba rozwiązań istniejących w roju dłużej niż długość życia. Wartość I_{max} jest definiowana jako maksymalna liczba utworzonych rozwiązań podczas wykonywania algorytmu (lub liczba obliczeń wartości funkcji celu).

Zaproponowana w [21] wersja algorytmu pszczelego (BAEX) do sortowania rozwiązań w roju, stosuje nie wartość funkcji celu problemu QAP, ale wartość oczekiwaną rozwiązań częściowo ustalonych - $E(\varphi(\pi)|i \in D)$, gdzie D określa zbiór ustalonych pozycji. Od klasycznej wersji algorytmu BA różni się on krokiem, w którym realizowane jest sortowanie populacji z zastosowaniem procedury SORTCEV() bazującej na wartości oczekiwanej.

Algorithm 2: Procedura SORTCEV().

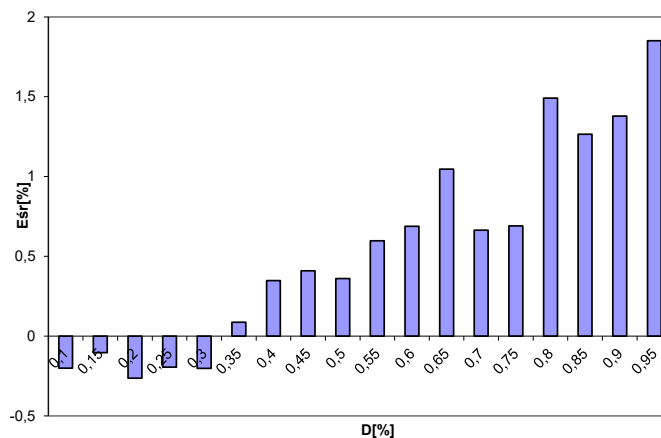
Dane wejściowe: $\varphi(\cdot)$, D .

Krok 1. Generuj losowo maskę d ustalonych pozycji rozwiązania (zbiór D).

Krok 2. Dla maski D oblicz $E(\varphi(\pi)|i \in D)$ dla wszystkich rozwiązań w populacji.

Krok 3. Sortuj rozwiązania w populacji wg $E(\varphi(\pi)|i \in D)$.

Dla zaproponowanego algorytmu przeprowadzono badania eksperymentalne dla rozbudowanego zbioru instancji testowych z biblioteki QAPLIB. Uzyskane wyniki dla dwóch rozmiarów maski D , równej $d = 20\%$ oraz $d = 80\%$ przedstawiono w Tab. 3.4 oraz Rys. 3.9. Wartość $E = 100\% * (\varphi_{BAEX} - \varphi_{BA}) / \varphi_{BA}$ to względna procentowa różnica wartości funkcji celu najlepszego rozwiązania uzyskanego przez algorytm BA w stosunku do wyniku uzyskanego przez algorytm BA-EX.



Rysunek 3.9: Średnia wartość błędów E_r (względna procentowa różnica wartości funkcji celu najlepszego rozwiązania uzyskanego przez algorytm BA w stosunku do wyniku uzyskanego przez algorytm BA-EX) dla dziesięciu przebiegów algorytmu.

Wyniki zamieszczone w Tab. 3.4 oraz na Rys. 3.9 pozwalają zauważyć istotny wpływ liczby ustalonych pozycji rozwiązania (wielkość losowej maski) na jakość uzyskiwanych rozwiązań przez algorytm BA-EX. Najlepsze wyniki uzyskiwano dla maski obejmującej niewielką liczbę pozycji – ok. 10-20%. Ustalenie rozmiaru maski powyżej 50% rozmiaru rozwiązania, przy losowym wyborze pozycji, daje dużo mniejsze prawdopodobieństwo poprawy funkcji celu. Porównanie uzyskanych wyników z wersją podstawową algorytmu BA, wskazuje na istotną poprawę efektywności algorytmu przez wprowadzenie ukierunkowania przeszukiwania wartością oczekiwaną. Jest ona uzyskana pomimo, że algorytm z częściowo ustalonym rozwiązaniem przegląda sąsiedztwo o mniejszej liczności. Wyznaczenie wartości oczekiwanych wymaga na-

tomiast dodatkowego nakładu obliczeniowego. Uzyskane wyniki wskazują, że wprowadzenie maski wyraźnie zmniejsza zbieżność algorytmu, poprawiając końcowy wynik optymalizacji.

Tablica 3.4: Wyniki algorytmów BA oraz BA-EX dla zadań testowych zaczerpniętych z biblioteki QAPLIB.

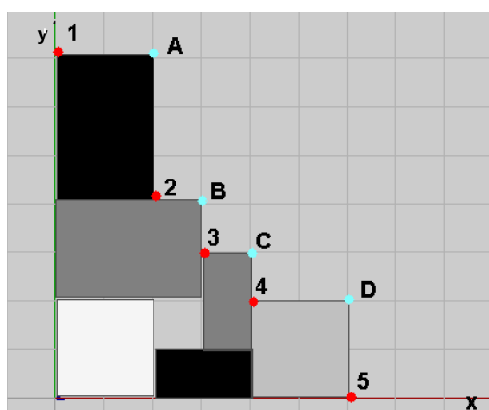
Nazwa	φ_{BA}	I_{best}	D=20%			D=80%		
			φ_{BAEX}	I_{best}	$E[\%]$	φ_{BAEX}	I_{best}	$E[\%]$
BUR26A	5436918	17	5435723	34	-0.02	5434234	361	-0.05
BUR26B	3827840	17	3825228	34	-0.07	3834281	490	0.17
BUR26C	5443738	23	5427308	160	-0.30	5427227	223	-0.30
BUR26D	3828852	21	3832582	28	0.10	3823195	449	-0.15
BUR26E	5389340	23	5387837	249	-0.03	5396997	224	0.14
BUR26F	3782051	16	3782044	137	0.00	3782747	178	0.02
BUR26G	10124306	17	10118634	104	-0.06	10138385	184	0.14
BUR26H	7098658	19	7099875	31	0.02	7102958	348	0.06
CHR22A	6832	14	6474	168	-5.24	7070	45	3.48
CHR22B	6858	10	6498	50	-5.25	6956	450	1.43
CHR25A	5218	14	4900	357	-6.09	6090	351	16.71
ESC32A	152	14	136	39	-10.53	158	492	3.95
ESC32B	200	12	192	168	-4.00	208	54	4.00
ESC32C	642	8	642	8	0.00	642	22	0.00
ESC32D	200	12	200	8	0.00	208	155	4.00
ESC32E	2	0	2	0	0.00	2	0	0.00
ESC32F	2	1	2	1	0.00	2	1	0.00
ESC32G	6	0	6	1	0.00	6	2	0.00
ESC32H	442	8	440	89	-0.45	442	104	0.00
ESC64A	116	9	116	11	0.00	116	66	0.00
KRA30A	93360	18	93540	22	0.19	97460	451	4.39
KRA30B	97490	15	96850	28	-0.66	100480	298	3.07
LIPA30A	13430	16	13406	15	-0.18	13386	252	-0.33
LIPA30B	176610	16	174829	36	-1.01	175041	341	-0.89
LIPA40A	32003	21	31969	210	-0.11	31953	488	-0.16
LIPA40B	564495	28	562076	301	-0.43	561346	389	-0.56
LIPA50A	62897	27	62804	38	-0.15	62803	455	-0.15
LIPA50B	1434253	27	1424316	497	-0.69	1427682	494	-0.46
LIPA60A	108306	43	108308	455	0.00	108336	452	0.03
LIPA60B	3001326	40	3014080	184	0.42	2993828	496	-0.25
SKO42	16178	35	16114	410	-0.40	16636	184	2.83
SKO49	23662	43	24050	243	1.64	24568	358	3.83
SKO56	34842	51	35556	87	2.05	35804	334	2.76
STE36A	10102	24	10086	153	-0.16	11462	378	13.46
STE36B	18024	27	19024	349	5.55	20016	390	11.05
STE36C	8468504	26	8687268	467	2.58	9537950	178	12.63
THO30	159112	16	153742	52	-3.37	157214	461	-1.19
THO40	247282	38	249698	342	0.98	248974	484	0.68
WIL50	49890	45	49286	464	-1.21	49590	464	-0.60
$E_{sr}[\%]$					-0.69			2.15

Algorytmy przybliżone i heurystyki konstrukcyjne dla zagadnienia pakowania trójwymiarowego

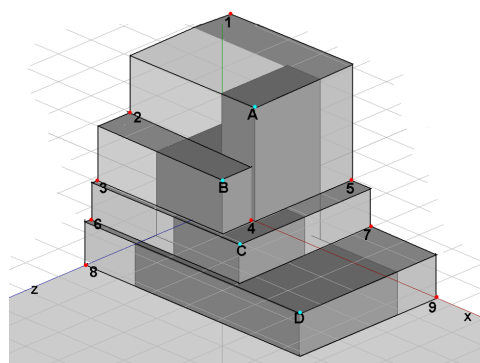
Zagadnienie trójwymiarowego pakowania (*3D Bin Packing Problem* - 3D-BPP), któremu poświęcono publikacje [10], [22], należy do niezwykle trudnych zagadnień optymalizacji dyskretnej, określanymi mianem *mocno NP-trudnych*, gdyż już samo zagadnienie pakowania jednowymiarowego należy do klasy zagadnień *NP-trudnych*.

Rzeczywiste zagadnienie pakowania paczek, w znamienitej większości prowadzane jest do trójwymiarowego problemu plecaka (3D-BPP). Zagadnienie to można sformułować następująco. Dany jest zbiór n prostopadłościennych paczek $j = 1, \dots, n$ o wymiarach: w_j, h_j, d_j oraz zysk związany z zapakowaniem paczki - c_j , do kontenera o wymiarach W, H, D . Sposób pakowania ma wykluczyć wzajemne „nachodzenie” na siebie paczek i wystawanie paczek poza obrys kontenera. Najczęściej poszukujemy podzbioru zbioru paczek, który spełniając ograniczenia przestrzeni ładunkowej maksymalizuje sumaryczny zysk załadowanych paczek, co jest związane z maksymalizacją wykorzystania przestrzeni kontenera.

Zaproponowane przez Autora heurystyki bazują na pojęciu konturu (obwiedni), który jest określony przez zrealizowane rozwiązanie częściowe i definiuje zbiór charakterystycznych punktów, określający przestrzeń (powierzchnię - $2D$) zajętą przez zapakowane paczki (określa aktualny stan rozwiązania). Kontur (lista punktów - *CPL*) jest pomocniczą strukturą danych, tworzoną na podstawie aktualnego rozwiązania. Założeniem towarzyszącym tworzeniu takiego opisu jest możliwe największa prostota. Stąd dążność do zmniejszenia liczby punktów definiujących kontur, a co za tym idzie, nakładu obliczeniowego związanego z jego obsługą (procedury obliczenia wartości funkcji celu, dodania - usunięcia punktów konturu, dodania - usunięcia punktów wstawienia), a w konsekwencji złożoności obliczeniowej algorytmu. Pomocniczą strukturą danych związaną z konturem jest zbiór punktów wstawienia, definiujący miejsca potencjalnego dołożenia nowych paczek do aktualnego rozwiązania. Dla zbioru punktów wstawienia, zakładającego dosunięcie paczek do konturu, wymagane jest dodatkowe sprawdzenie warunków ograniczeń zewnętrznych przestrzeni ładunkowej. Liczność zbioru punktów wstawienia dla konturu $2D$ (o liczności n) wynosi $n+1$ (Rys. 3.10 - zbiór 1, 2, ..., 5), natomiast dla konturu $3D$ maksymalnie $2n + 1$ (Rys. 3.11 - zbiór 1, 2, ..., 9).



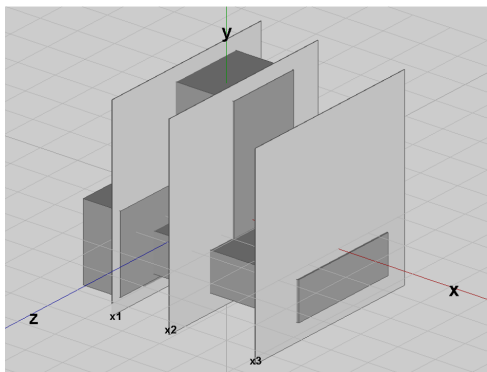
Rysunek 3.10: Kontur wypukły - zbiór $\{A, B, C, D\}$ oraz zbiór punktów wstawienia $\{1, 2, \dots, 5\}$ dla dwuwymiarowego zagadnienia pakowania.



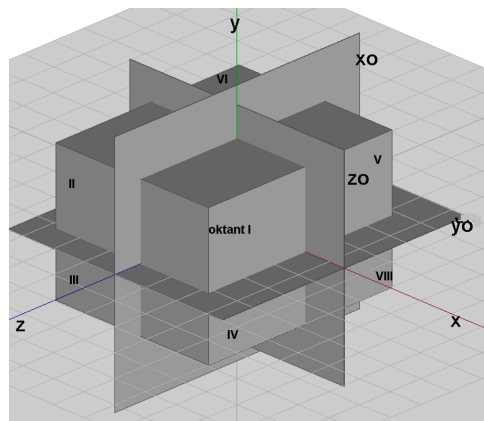
Rysunek 3.11: Kontur wypukły - zbiór $\{A, B, C, D\}$ oraz zbiór punktów wstawienia $\{1, 2, \dots, 9\}$ dla trójwymiarowego zagadnienia pakowania.

Zaproponowane procedury tworzenia konturów dwu- i trójwymiarowych oraz obliczana ich powierzchni i objętości, stosują założenie o wypukłości figury. Procedury te bazują na podziale

przestrzeni konturu płaszczyznami na warstwy wyznaczone przez punkty wyznaczające kontur wypukły (Rys. 3.12), na podstawie kolejnych punktów definiujących kontur wypukły dla kolejnych oktantów przestrzeni 3D (Rys. 3.13). Opracowane procedury podczas wstawiania nowego obiektu trójwymiarowego umożliwiają usunięcie od jednego do kilku punktów wcześniejszej definiujących kontur wypukły oraz uaktualnienie listy możliwych punktów wstawienia, bez konieczności przeliczania całego konturu.



Rysunek 3.12: Podział przestrzeni konturu na warstwy płaszczyznami równoległymi do yz .



Rysunek 3.13: Podział przestrzeni ładunkowej kontenera na oktanty.

Zaproponowana procedura określania konturu wypukłego, punktów wstawienia oraz sposobu wyliczania objętości konturu pozwoliła na stworzenie wydajnego algorytmu konstrukcyjnego dla zagadnienia pakowania trójwymiarowego.

W przypadku trójwymiarowego zagadnienia reprezentacja rozwiązania może stosować potrójną sekwencję liczb (permutacji), zawierającej relatywne położenie paczek w stosunku do pozostałych. Reprezentacja ta nie opisuje wszystkich możliwych sposobów upakowania, jednak pozwala na reprezentację ich dużej części, w szczególności rozwiązań realizowanych przez robota (*ang. robot packing*). Jest to sposób pakowania od lewego, dolnego rogu kontenera, z dostawianiem kolejnych paczek naprzeciw, powyżej lub po prawej, względem wszystkich poprzednio wstawionych. Metoda ta modeluje rzeczywiste zagadnienie pakowania kontenera występujące w większości produkcyjnych systemów załadowniczych. Aby zapobiec kolizji pomiędzy ramieniem robota oraz paczkami już wstawionymi, zakłada się, że żadna z paczek nie może blokować ruchu ramienia. Można wykazać, że to ograniczenie nie ma dużego wpływu na jakość pakowania. Można zaproponować abstrakcyjną reprezentację sekwencji pakowania w oparciu o trzy permutacje A, B i C. Dla każdej z tych permutacji, możemy zdefiniować pewną relację ω_{ij} , która oznacza, że w permutacji liczba i występuje przed j . Załóżmy również, że $\omega_{ij} \Leftrightarrow \omega_{ji}$. Niech (x_i, y_i, z_i) i (w_i, d_i, h_i) będą odpowiednio współrzędnymi dolnego, lewego, tylnego wierzchołka i -tej paczki oraz jej szerokością, głębokością i wysokością. Definiuje się trzy relacje porządku częściowego $\alpha_{ij}, \beta_{ij}, \chi_{ij}$:

1. $\alpha_{ij} \Leftrightarrow (x_i + w_i \leq x_j \vee y_i \geq y_j + h_j \vee z_i \geq z_j + d_j)$ - paczka i jest na lewo, ponad lub z przodu w stosunku do paczki j ,
2. $\beta_{ij} \Leftrightarrow (x_i \leq x_j + w_j \vee y_i \leq y_j + h_j \vee z_i \leq z_j + d_j)$ - paczka i jest na lewo, pod lub z tyłu w stosunku do paczki j ,
3. $\chi_{ij} \Leftrightarrow (x_i \geq x_j + w_j \vee y_i + h_i \leq y_j \vee z_i \geq z_j + d_j)$ - paczka i jest na prawo, pod lub z przodu w stosunku do paczki j .

Zgodnie z zasadą pakowania przez robota, każda nowa paczka, w stosunku do poprzednich jest umieszczana najbardziej na lewo, nad i z tyłu. Można też udowodnić, że każdej z sekwencji A, B i C można przypisać odpowiednie ułożenie paczek. Aby określić wzajemne położenie paczek i oraz j, opierając się na (1), (2) oraz (3), można zastosować następujące relacje:

$$\begin{aligned}\alpha_{ij} \wedge \beta_{ij} \wedge !\chi_{ij} &\Rightarrow i \text{ jest na lewo od } j, \\ !\alpha_{ij} \wedge \beta_{ij} \wedge \chi_{ij} &\Rightarrow i \text{ jest poniżej } j, \\ !\alpha_{ij} \wedge \beta_{ij} \wedge !\chi_{ij} &\Rightarrow i \text{ jest z tyłu } j, \\ \alpha_{ij} \wedge \beta_{ij} \wedge \chi_{ij} &\Rightarrow i \text{ jest z tyłu } j.\end{aligned}$$

Dwa ostatnie wyrażenia powodują preferowanie jednego kierunku pakowania, mając negatywny wpływ na uzyskaną jakość rozwiązania. Gwarantują one jednak to, że każdy zestaw permutacji A, B, C definiuje pewną sekwencję pakowania. W oparciu o powyższe równania można zdefiniować procedurę określania pozycji nowej paczki, na podstawie paczek już wstawionych. Paczki wstawiamy do kontenera w kolejności określonej przez sekwencję, gdzie pierwsza paczka ma współrzędne (0, 0, 0). Jeśli zbiór P zawiera paczki już załadowane do kontenera, to pozycję, punkt wstawienia (x_j, y_j, z_j) , kolejnej paczki $j \in P$, wstawianej do kontenera, wyznaczamy na podstawie załadowanych paczek $i \notin P$. Niech dalej:

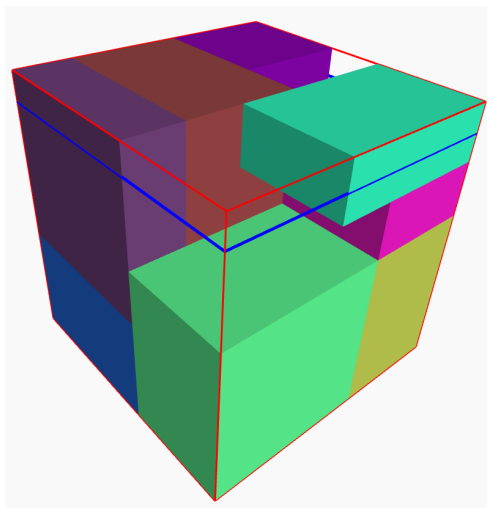
$$\begin{aligned}P_x &\subseteq P \text{ jest podzbiorem paczek dla których spełniona jest zależność } A_{ij} \wedge B_{ij} \wedge \sim C_{ij}, \\ P_y &\subseteq P \text{ jest podzbiorem paczek dla których spełniona jest zależność } \sim A_{ij} \wedge B_{ij} \wedge C_{ij}, \\ P_z &\subseteq P \text{ jest podzbiorem paczek dla których spełniona jest zależność } (\sim A_{ij} \wedge B_{ij} \wedge \sim C_{ij}) \vee (A_{ij} \wedge B_{ij} \wedge C_{ij}).\end{aligned}$$

Współrzędne paczki j są wyznaczone za pomocą następujących reguł:

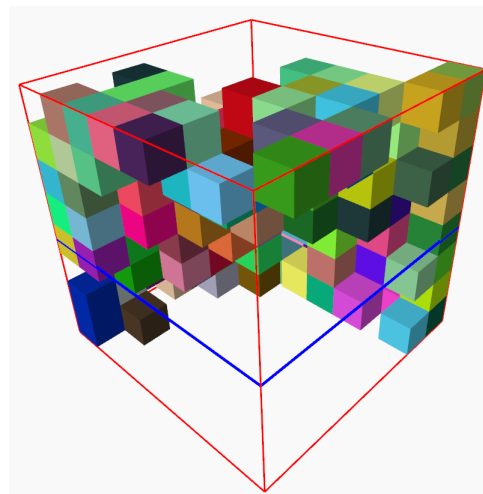
$$\begin{aligned}x_i &= \max \{0; \max \{x_i + w_i\}, i \in P_x\}, \\ y_i &= \max \{0; \max \{y_i + d_i\}, i \in P_y\}, \\ z_i &= \max \{0; \max \{z_i + h_i\}, i \in P_z\},\end{aligned}$$

gdzie (w_i, d_i, z_i) są wymiarami krawędzi paczki j po uwzględnieniu jej rotacji. Jeżeli po wstawieniu paczki j okaże się, że wykracza ona poza obrys kontenera, paczka nie jest ładowana. W przypadku spełnienia ograniczenia wymiaru kontenera, j -ta paczka jest ładowana i wstawiana do zbioru P . Jeśli w tablicy zapisujemy pozycję każdej załadowanej paczki i w trzech sekwencjach A, B, C, to dla dwóch paczek i, j można w stałym czasie sprawdzić relacje A_{ij}, B_{ij}, C_{ij} . Stąd wstawienie jednej paczki za pomocą porównań z wcześniej wstawionymi za pomocą wymienionych reguł jest wykonalne w czasie $O(|P|)$, a wszystkich paczek w czasie $O(n^2)$. Działanie procedury wstawiania można przyspieszyć usuwając ze zbioru P paczki całkowicie zakryte przez nowo wstawioną paczkę j . Paczka i jest zakryta przez paczkę j jeżeli $x_i + w_i \leq x_j + w_j, y_i + h_i \leq y_j + h_j, z_i + d_i \leq z_j + d_j$ i wtedy nie wpływa ona na proces wstawiania następnych paczek. Usuwając taką paczkę z P unikamy niepotrzebnych obliczeń podczas wstawiania kolejnych paczek do kontenera. Każde rozwiązanie zagadnienia pakowania postać $p = \{A, B, C, R\}$, gdzie: A, B, C są permutacjami n -elementowego zbioru określającymi porządek pakowania, gdzie n – oznacza liczbę paczek, natomiast wektor R , określa rotację paczki w stosunku do położenia początkowego $R = r_1, r_2, \dots, r_n$, gdzie $i \in 0, \dots, 5$.

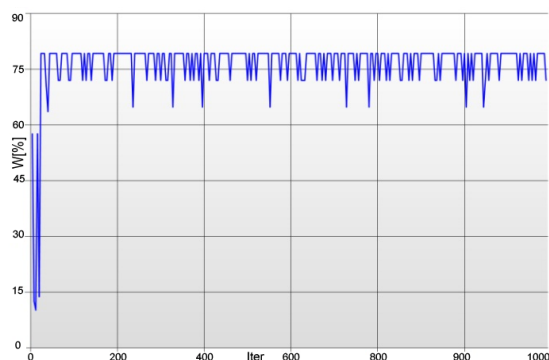
Do rozwiązywania problemu zaproponowano algorytm ewolucyjny (EA) w którym zastosowano selekcję typu *Steady-State*. Dodatkowo wprowadzono w nim uproszczenia polegające na losowym wyborze rodziców z populacji w oparciu o rozkład jednostajny oraz na współzawodnictwie o miejsce w populacji nowo utworzonych rozwiązań potomnych, na podstawie wartości funkcji celu. W celu przebadania możliwości zastosowania dla problemu 3D-BPP innych algorytmów przybliżonych, opracowano algorytm symulowanego wyżarzania, w którym modyfikację otoczenia aktualnego rozwiązania dokonuje się poprzez zastosowanie procedur:



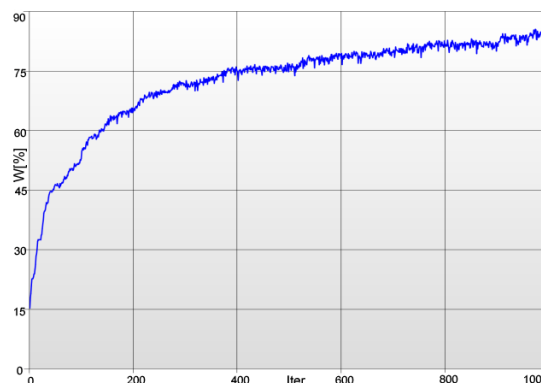
Rysunek 3.14: Przykład pakowania dla $n = 20$ paczek uzyskanych poprzez cięcie gilotynowe (współczynnik wypełnienia 90%).



Rysunek 3.15: Przykład niskiej jakości upakowania uzyskanego za pomocą algorytmu symulowanego wyżarzania kontenera 8x8x8 dla $n = 512$ paczek uzyskanych poprzez cięcie gilotynowe.



Rysunek 3.16: Przebieg procesu optymalizacji pomocą algorytmu ewolucyjnego kontenera 8x8x8 dla $n = 512$ paczek uzyskanych poprzez cięcie gilotynowe).



Rysunek 3.17: Przebieg procesu optymalizacji za pomocą algorytmu symulowanego wyżarzania kontenera 8x8x8 dla $n = 512$ paczek uzyskanych poprzez cięcie gilotynowe.

SWAP(\cdot) – dokonującej losowej zmiany położenia dwóch paczek w losowo wybranej permutacji (A, B lub C) z rozkładem jednostajnym oraz RCHAN(\cdot) – dokonującej losowej modyfikacji, zgodnie z rozkładem jednostajnym, jednej ze współrzędnych w wektorze R ([10]).

Na Rys. 3.14 i Rys. 3.15 przedstawiono przykładowe rozwiązania uzyskane dla zadania testowego wygenerowanego w wyniku cięcia gilotynowego.

W Tab. 3.5 zamieszczono wyniki porównawcze algorytmu ewolucyjnego (EA), symulowanego wyżarzania (SA) oraz trzech algorytmów konstrukcyjnych, realizujących trzy heurystyki konstrukcyjne *ATD*, *BTD*, *CTD*, których dokładny opis zamieszczono w publikacji [10]. Algo-

rytmy te działają na rozwiązaniach kodowanych w postaci sekwencji 3 permutacji wprowadzających częściowy porządek (*semi-order*), startując z wygenerowanych losowo rozwiązań początkowych. Konstrukcja operatorów i sąsiedztwa w obu przypadkach korzystała z typowych rozwiązań dla zagadnień permutacyjnych. Wyniki badań - duża rozpiętość jakości uzyskiwanych rozwiązań, dowodzą, że typowe operatory, nie uwzględniające specyfiki rozwiązywanego problemu nie są efektywne w trudnych zagadnieniach optymalizacyjnych, charakteryzujących się dużą liczbą ograniczeń. Znaczna poprawa wyników została uzyskana przez Autora, w przypadku zastosowania algorytmów zachłanych jako procedur inauguracyjnych oraz stosując metody zachłanne w konstrukcji operatorów. Wprowadzenie inteligentnych reguł ukierunkowuje algorytm w stronę rozwiązań dopuszczalnych, zwiększając szanse na uzyskanie rozwiązań dobrej jakości. Algorytmy złożone (heurystyki + algorytmy popraw), w przypadku uwzględnienia specyfiki zagadnienia, gwarantują uzyskanie powtarzalnych, lepszych wyników ([10]).

Tablica 3.5: Wyniki testów algorytmu ewolucyjnego, symulowanego wyżarzania i trzech heurystyk konstrukcyjnych dla instancji testowych uzyskanych poprzez cięcie gilotynowe oraz losową generację paczek z rozkładem normalnym, w którym $\mu = 2$ oraz $\sigma_2 = 2$.

Typ instancji	Rozmiar kontenera WxDxS	Liczba paczek	Współczynnik wypełnienia(W[%])					
			EA	SA	ATD	BTD	CTD	LTD
Cięcie gilotynowe	4x4x4	48	48 (75%)	52 (81%)				
	5x5x5	125	89 (71%)	119 (95%)	91(73%)	105(84%)	110(88%)	60(48%)
	8x8x8	512	264 (51%)	148 (29%)	317(62%)	276(54%)	282(55%)	124(24%)
	10x10x10	1000	126 (12%)	189 (19%)				
Generacja losowa	4x4x4	50	32 (50%)	64 (100%)				
	5x5x5	125	70(56%)	104 (83%)	50(40%)	32(25%)	42(34%)	26(20%)
	8x8x8	512	160 (31%)	242 (47%)	169(33%)	115(23%)	65(13%)	94(18%)
	10x10x10	1000	240 (24%)	340 (34%)				

3.2.3. Opracowanie dla zagadnienia kwadratowego przydziału (QAP) algorytmu przybliżonego stosującego paradygmaty mechaniki kwantowej

Zaproponowany mechanizm bazujący na reprezentacji rozwiązań z zastosowaniem *Q-bitów* oraz przetwarzaniu z zastosowaniem bramek kwantowych pozwala na sterowanie poziomem eksploracji lub intensyfikacji podczas przeszukiwania przestrzeni rozwiązań. W ramach badań nad implementacją algorytmów przybliżonych dla zagadnienia QAP zaproponowano, dotąd nie stosowaną do jego rozwiązania, implementację algorytmu przybliżonego wykorzystującego paradygmaty mechaniki kwantowej ([15]). W opracowanym algorytmie do reprezentacji rozwiązania zastosowano ciąg Q-bitów. Rozwiązania były poddawane modyfikacjom, zależnym od stanu całej populacji, a w szczególności od aktualnie najlepszego uzyskanego rozwiązania. Efekt ten został osiągnięty dzięki zastosowaniu tzw. tabeli *look-up* pozwalającej na sterownię zbieżnością algorytmu oraz *dualnej* (permutacyjnej i Q-bitowej) reprezentacji rozwiązania. Odpowiedni dobór wartości zawartych w tablicy *look-up* może przyczynić się zarówno do szerszej eksploracji przestrzeni rozwiązań, jak też do intensyfikacji przeszukiwania otoczenia najlepszych znalezionych rozwiązań.

Opracowany algorytm optymalizacyjny bazujący na reprezentacji kwantowej został przebadany poprzez przeprowadzenie rozbudowanych eksperymentów obliczeniowych z zastosowaniem dużej liczby reprezentatywnych instancji testowych dla zagadnienia QAP.

Q-bit to najmniejsza i niepodzielna jednostka informacji kwantowej, definiowana przez liniową superpozycję stanów podstawowych. Inaczej mówiąc, jest on reprezentowany przez

wektor w przestrzeni Hilberta ze stanami podstawowymi $|0\rangle$ i $|1\rangle$. Kwantowe algorytmy ewolucyjne stosują Q-bity do reprezentacji rozwiązań, a bramki kwantowe (bramki: rotacyjna, NOT, AND, OR, NAND i Hadamard) do tworzenia na ich podstawie rozwiązań potomnych. Przejście z reprezentacji Q-bitowej do postaci obserwowalnej, a więc zmiennych opisujących zmienne decyzyjne optymalizowanego problemu, następuje poprzez pomiar („obserwację”). Stan $|\Psi\rangle$ Q-bitu jest reprezentowany w następujący sposób:

$$|\Psi\rangle = \alpha |0\rangle + \beta |1\rangle,$$

gdzie α i β są liczbami zespolonymi, określającymi amplitudy prawdopodobieństwa odpowiednich stanów i spełniają warunek normalizacji $|\alpha|^2 + |\beta|^2 = 1$.

W związku z tym wartości $|\alpha|^2$ i $|\beta|^2$ określają prawdopodobieństwo przyjęcia przez Q-bit stanów odpowiednio „0” lub „1”. Mówiąc ogólnie, Q-bit jest najmniejszą jednostką informacji i jest reprezentowany przez parę liczb $[\alpha\beta]^T$. Jeśli wyrażenie q jest zdefiniowane jako łańcuch n Q-bitów, to za jego pomocą można opisać 2^n stanów:

$$q = \left[\begin{array}{c|c|c|c|c} \alpha_1 & \alpha_2 & \alpha_3 & \cdots & \alpha_n \\ \beta_1 & \beta_2 & \beta_3 & \cdots & \beta_n \end{array} \right].$$

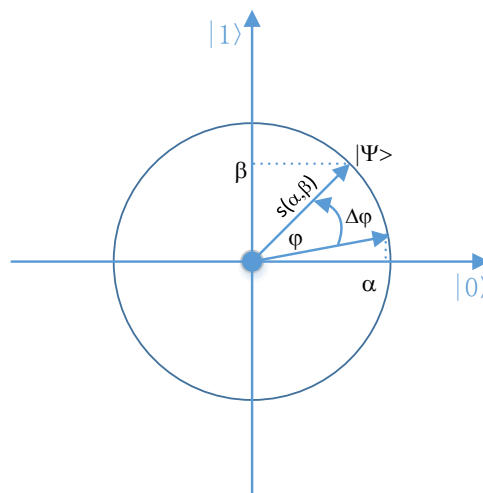
i -ty Q-bit jest aktualizowany poprzez zastosowanie następującej bramki kwantowej:

$$G(\varphi) = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix}, \quad (3.7)$$

gdzie kąt φ w kwantowej bramce rotacyjnej jest zdefiniowany następująco:

$$\varphi = s(\alpha, \beta)\Delta\varphi, \quad (3.8)$$

gdzie $s(\alpha, \beta)$ i $\Delta\varphi$ są odpowiednio kierunkiem rotacji bramki (znak φ) i wielkością kąta rotacji φ . Przykład kwantowej bramki rotacyjnej przedstawiono na Rys. 3.18.



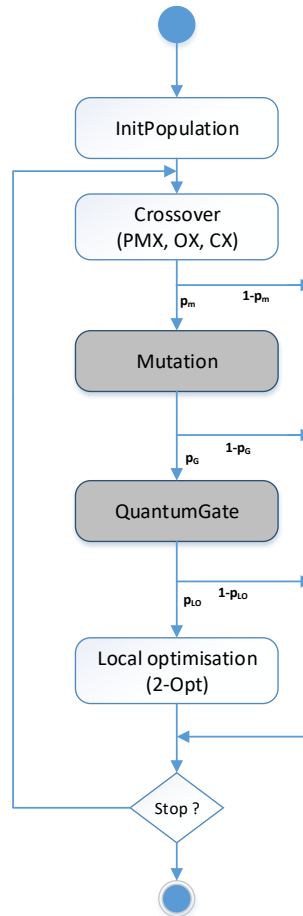
Rysunek 3.18: Geometryczna interpretacja bramki rotacyjnej.

Kierunek i kąt rotacji jest określany na podstawie odpowiednio zdefiniowanej tablicy *lookup* (Tab. 3.6), gdzie $f(x)$ i $f(b)$ reprezentują odpowiednio wartość funkcji kryterialnej problemu QAP dla aktualnego rozwiązania i najlepszego rozwiązania (w danej iteracji lub globalnie).

Zaproponowany algorytm kwantowy algorytm dla problemu QAP (Q²APA) wykorzystuje selekcję rankingową oraz kilka typów operatorów pseudo-genetycznych, działających na permutacyjnej postaci rozwiązania - PMX (*Partially Matched Crossover*), OX (*Order Crossover*)

i CX (*Cycle Crossover*). Aby było możliwe ich zastosowanie, to łańcuch Q-bitów kodujący rozwiązanie musi zostać zdekodowany do postaci „obserwowalnej”, tj. permutacji.

Ogólną strukturę algorytmu Q²APA przedstawiono na Rys. 3.19. Na schemacie procedury wykorzystujące reprezentację Q-bitową permutacji są oznaczone kolorem szarym.



Rysunek 3.19: Algorytm Q²APA.

Każde rozwiązanie w populacji jest reprezentowane przez krotkę $S_{ij} = \{P_{ij}, Q_{ij}\}$, gdzie Q_{ij} jest Q-bitową reprezentacją rozwiązania, podczas gdy P_{ij} to obserwowalny stan Q-bitu (permutacja). Indeksy i i j definiują odpowiednio indeks populacji (iteracja) i indeks rozwiązania w populacji. Przed zastosowaniem operatora krzyżowania lub mutacji, rozwiązanie musi zostać przeniesione ze stanu superpozycji (Q_{ij}) do obserwowalnego stanu (P_{ij}) za pomocą opracowanej procedury `ObservableState()`.

Tablica 3.6: Tablica *look-up* dla problemu QAP.

x	b	$f(x) < f(b)$	$\Delta\varphi$	$s(\alpha, \beta)$			
				$\alpha \cdot \beta > 0$	$\alpha \cdot \beta < 0$	$\alpha = 0$	$\beta = 0$
0	0	False	0.2π	0	0	0	0
0	0	True	0	0	0	0	0
0	1	False	0.5π	0	0	0	0
0	1	True	0	-1	+1	± 1	0
1	0	False	0.5π	-1	+1	± 1	0
1	0	True	0	+1	-1	0	± 1
1	1	False	0.2π	+1	-1	0	± 1
1	1	True	0	+1	-1	0	± 1

Operator mutacji generuje losową permutację w odległości Hamminga równej 2. Pozwala to zapobiec uwięzieniu algorytmu Q^2 APA w lokalnym optimum.

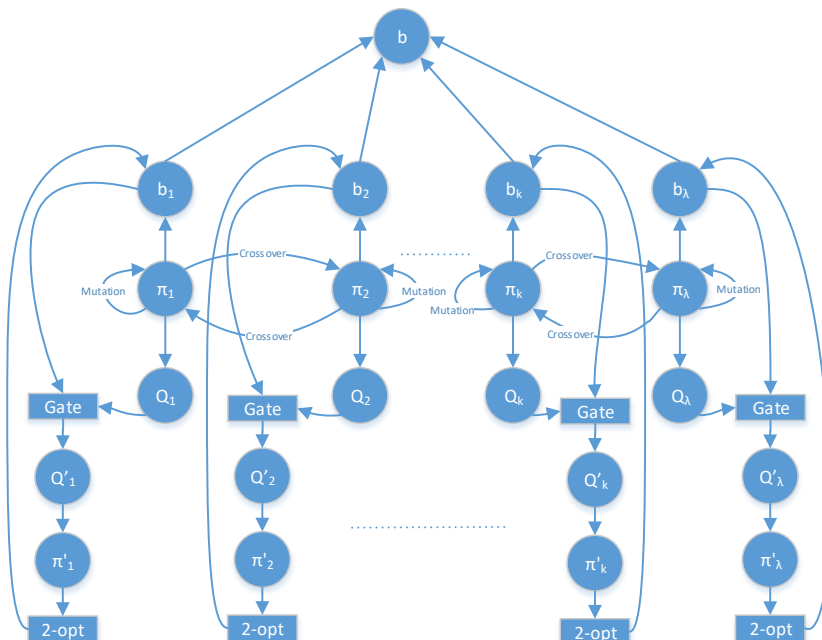
Następnie, aby kontrolować właściwości eksploracyjne algorytmu, z pewnym z góry określonym prawdopodobieństwem, stosuje się kwantową bramkę rotacyjną.

W opracowanym algorytmie zaproponowano zastosowanie zmiennej wielkość kąta obrotu w rotacyjnej bramce kwantowej określanego na podstawie tabeli *look-up*. Ostateczna wartość kąta φ jest związana z numerem generacji i (Rys. 3.19). W algorytmie zastosowano geometryczną redukcję kąta bramki rotacyjnej wyrażoną jako $\varphi \cdot \delta^i$, gdzie $\delta < 1$. W zależności od ustawień algorytmu, używane są zmodyfikowane lub oryginalne wartości z tabeli *look-up* (Tab. 3.6).

Wszystkie rozwiązania generowane za pomocą bramki rotacyjnej, po przejściu na postać permutacyjną z zastosowaniem procedury *ObservableState*, są poprawiane z zastosowaniem lokalnej procedury optymalizacji *2-opt*, która efektywnie przegląda sąsiedztwo o licznosci $\frac{n(n-1)}{2} + 1$.

Przebieg algorytmu jest przedstawiony na Rys. 3.20, gdzie b oznacza najlepsze rozwiązanie dotąd znalezione przez algorytm Q^2 APA, a b_1, \dots, b_λ reprezentują rozwiązania w postaci dualnej (permutacji i kwantowej), więc $b_i = \{\pi_i, Q_i\}$. Należy zauważyć, że permutacyjne postacie rozwiązań są przetwarzane z zastosowaniem operatorów mutacji i krzyżowania. Operatory te jednak nie wpływają na stan kwantowy potomków, na podstawie których tworzona jest kwantowa reprezentacja rozwiązań.

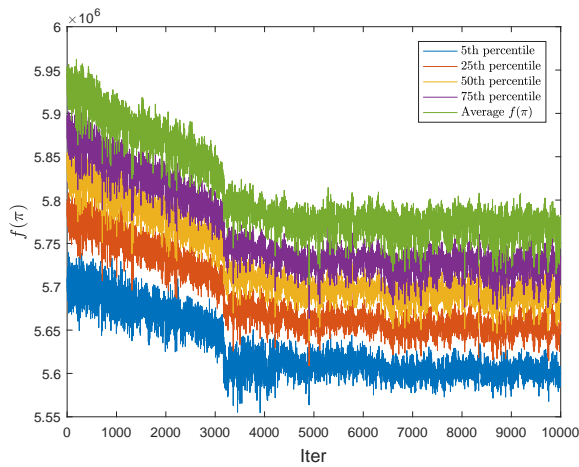
Q-bitowa postać rozwiązania jest modyfikowana przez zastosowanie kwantowej bramki rotacyjnej z prawdopodobieństwem p_m . Następnie stan każdego Q-bitu rozwiązania (Q') jest porównywany z najlepszym rozwiązaniem dotychczas otrzymanym przez algorytm (lub najlepszym w aktualnej iteracji) i rozwiązanie to jest konwertowane z zastosowaniem zaproponowanego algorytmu konwersji na permutację (zbiór Π'). Tak uzyskane rozwiązania są ulepszone z zastosowaniem procedury *2-opt* z prawdopodobieństwem p_{LO} (Rys. 3.19).



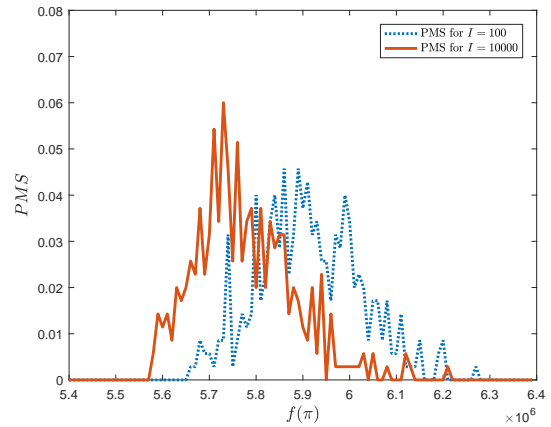
Rysunek 3.20: Detaliczny Q^2 APA przebieg algorytmu.

Rys. 3.21 prezentuje wartość średnią funkcji celu oraz przebieg wartości kwantyli rzędu 5, 25 i 75 w kolejnych iteracjach. Widać wyraźny spadek wszystkich wartości w kolejnych

iteracjach. Należy zauważyć, że zmienność uzyskanych wyników jest nieunikniona, ponieważ Q^2 APA jest algorytmem stochastycznym.



Rysunek 3.21: Wartości percentyli dla instancji BUR26A ($p_G = 0.7$).



Rysunek 3.22: Funkcja masy prawdopodobieństwa (PMF) BUR26A, dla $I = 100$ i $I = 10000$ ($p_G = 0.7$).

Rys. 3.22 przedstawia funkcję masy prawdopodobieństwa (*probability mass function* - PMF) dla dwóch różnych długości przebiegu algorytmu ($I = 100, I = 10000$). Na początku procesu optymalizacji rozkład wartości funkcji celu w populacji ma wartość średnią około $5,9 \times 10^6$. W kolejnych iteracjach maksimum PMF rośnie i przesuwa się w kierunku mniejszych wartości funkcji celu, utrzymując jednocześnie relatywnie zróżnicowaną populację. Dowodzi to, że zastosowanie reprezentacji kwantowej i bramki kwantowej zapobiega zjawisku ujednoczenia populacji, które jest dużym problemem podczas stosowania algorytmów ewolucyjnych w optymalizacji problemów dyskretnych, a szczególnie permutacyjnych.

3.2.4. Opracowanie równoległych implementacji algorytmów populacyjnych na platformie sprzętowej GPU (*Graphical Processing Unit*) w celu poprawy ich efektywności w optymalizacji problemów \mathcal{NP} -trudnych

W ramach badań nad poprawą efektywności algorytmów populacyjnych w rozwiązywaniu \mathcal{NP} -trudnego problemu QAP opracowano implementacje algorytmu rojowego i pszczelego na procesorach graficznych GPU (platformy CUDA i OpenCL), uwzględniając specyfikę tej platformy obliczeniowej ([19], [27]).

Uzyskane wyniki są wskazówką dotyczącą sposobu implementacji algorytmów populacyjnych (PSO i BA), który uwzględnia wymagania i ograniczenia platformy sprzętowej GPU. Zaproponowane podejście, oprócz przyspieszenia obliczeń, pozwala uzyskać lepsze rozwiązania w porównaniu z algorytmami sekwencyjnymi wykonywanymi na platformie CPU. Opracowane algorytmy stosowały dwie metodyki przetwarzania danych dostępne na platformie sprzętowej GPU.

Pierwsza metoda opiera się na dekompozycji danych przetwarzanych przez algorytm do postaci dużych macierzy (zawierających np. kilka tysięcy elementów), które można niezwykle szybko (w sposób równoległy) przetwarzać na procesorach graficznych.

Druga metoda zasadniczo różni się do pierwszej, ponieważ każde rozwiązanie jest przetwarzane w osobnym wątku. W tym przypadku, każdy wątek realizuje cały algorytm, a

populacja (zbiór rozwiązań) składa się z kilku- lub kilkudziesięciu tysięcy rozwiązań przetwarzanych osobno przez taką samą liczbę wątków. Dzięki temu unika się konieczności kopiowania danych pomiędzy GPU i CPU, bowiem wszystkie obliczenia i synchronizacja rozwiązań jest realizowana na GPU. Każdy wątek to osobna pszczoła poszukująca rozwiązania wymieniając informacje z sąsiadami. Można więc powiedzieć, że algorytm ten jest najbliższy swemu biologicznemu wzorcowi.

Przeprowadzone zostały rozbudowane eksperymenty określające wpływ parametrów algorytmów na uzyskane rozwiązania. W ramach badań uwzględniono zarówno specyfikę sprzętową GPU, jak również cechy stosowanych algorytmów populacyjnych.

Wyniki eksperymentów wykazały, że w przypadku przeszukiwania przestrzeni rozwiązań z zastosowaniem wielu rojów należy położyć duży nacisk na implementację odpowiednich procedur odpowiedzialnych za optymalizację lokalną. Ponadto, zastosowanie wielu mniejszych niezależnych rojów sprzyja uzyskaniu lepszych rozwiązań niż zastosowanie mniejszej liczby rojów o większym rozmiarze. Uzyskane rezultaty stanowią ważną przesłankę, co do konstrukcji i doboru parametrów algorytmów populacyjnych, których odmianą są algorytmy pszczeli i rojowy.

Implementacja algorytmu rojowego (PSO) na platformie OpenCL

W ramach prac związanych z wydajną optymalizacją problemów \mathcal{NP} -trudnych opracowano implementację algorytmu PSO dla problemu QAP na platformie OpenCL [27]. OpenCL to rozwiązanie pozwalające programistom na akcelerację algorytmów z wykorzystaniem mocy obliczeniowej wielordzeniowych kart graficznych i procesorów. Urządzenia obsługujące standard OpenCL są obecnie szeroko dostępne. Jednym z celów badań, było sprawdzenie, czy możliwości oferowane przez popularne procesory graficzne można wykorzystać do przyspieszenia algorytmów optymalizacji wymagających znacznej mocy obliczeniowej.

Klasyczny algorytm rojowy (*Particle Swarm Optimisation* - PSO) to metoda optymalizacji opracowana do rozwiązywania problemów ciągłych. W tym celu pewna liczba cząstek przemieszcza się w przestrzeni rozwiązań, aktualizując swój stan w dyskretnych krokach czasowych $t = 1, 2, 3, \dots$. Każda cząstka opisana jest przez pozycję $x(t)$ i prędkość $v(t)$. Pamięta ona także swoją najlepszą pozycję osiągniętą w dotychczasowym przebiegu algorytmu (najlepsze rozwiązanie) $p^L(t)$, a także najlepsze rozwiązanie znalezione przez cały rój $p^G(t)$.

Stan cząstki jest określony za pomocą formuły (3.9). Współczynniki $c_1, c_2, c_3 \in [0, 1]$ to kolejno: *inercja*, *czynnik lokalny* (lub *samoświadomość*) i *czynnik społeczny*, natomiast r_1, r_2 są liczbami losowymi o rozkładzie jednostajnym na przedziale $[0, 1]$.

$$\begin{aligned} v(t+1) &= c_1 \cdot v(t) + c_2 \cdot r_2(t) \cdot (p^L(t) - x(t)) \\ &\quad + c_3 \cdot r_3(t) \cdot (p^G(t) - x(t)) \\ x(t+1) &= x(t) + v(t) \end{aligned} \tag{3.9}$$

Adaptacja algorytmu PSO dla zagadnień permutacyjnych wymaga odpowiedniej interpretacji pojęcia prędkości, a także zdefiniowanie równoważników skalarnego mnożenia, odejmowania i dodawania dla argumentów będących permutacjami.

Stan cząstki to para macierzy (X, V) . W zaproponowanym podejściu obie macierze są rozmiaru $n \times n$, gdzie n jest wielkością problemu. Macierz permutacji $X = [x_{ij}]$ koduje przydział obiektów do lokalizacji i jej elementy x_{ij} są równe 1, jeśli j -ty obiekt jest przypisany do i -tej lokalizacji lub 0 w przeciwnym przypadku.

Cząstka porusza się w przestrzeni rozwiązań zgodnie z kierunkiem zdefiniowanym przez prędkość V . Elementy v_{ij} mają następującą interpretację: jeśli v_{ij} ma dużą wartość, to procedura określająca następne rozwiązanie powinna faworyzować przydział $x_{ij} = 1$. Natomiast, jeśli $v_{ij} \leq 0$, wówczas preferowany jest przydział $x_{ij} = 0$.

Stan cząstki osiągnięty w i -tej iteracji jest określony przez $(X(t), V(t))$. W każdej iteracji jest ona aktualizowana zgodnie z formułami (3.10) i (3.11).

$$V(t+1) = S_v (c_1 \cdot V(t) + c_2 \cdot r_2(t) \cdot (P^L(t) - X(t)) + c_3 \cdot r_3(t) \cdot (P^G(t) - X(t))) \quad (3.10)$$

$$X(t+1) = S_x(X(t) + V(t)) \quad (3.11)$$

Współczynniki r_2 i r_3 to losowe liczby z przedziału $[0, 1]$, określane w każdej iteracji dla każdej cząstki osobno. Wprowadzają one element losowości podczas wyboru kolejnego kierunku wynikającego z inercji (c_1), najlepszego rozwiązania lokalnego (samoświadomość - c_2) lub najlepszego rozwiązania dotąd znalezionej (czynnik społeczny - c_3).

Wszystkie operatory pojawiające się w (3.10) i (3.11) są standardowymi operatorami z algebry liniowej. Zamiast przedefiniować je dla konkretnego problemu, jak to czynią inni autorzy, proponuje się użycie funkcji agregujących S_v i S_x , które pozwalają dostosować algorytm do rozważanego problemu.

Funkcja S_v zapewnienia, że wartości prędkości mają ograniczoną wartość. W początkowych eksperymentach, zaproponowano funkcję, która ogranicza wartości V do przedziału $[-v_{max}, v_{max}]$. Jednak przeprowadzone doświadczenia wykazały, że w przypadku małego współczynnika bezwładności, np. $c_1 = 0,5$, po kilku iteracjach prędkości wszystkich cząstek spadają do 0 i wszystkie cząstki zbiegają się do najlepszego rozwiązania napotkanego wcześniej przez rój. Aby tego uniknąć, zaproponowano funkcję, która dodatkowo wykonuje normalizację kolumny macierzy. Dla każdej j -tej kolumny obliczana jest suma wartości bezwzględnych elementów $n_j = \sum_{i=1}^n |v_{ij}|$, a następnie wykonywane jest przypisanie: $v_{ij} \leftarrow v_{ij}/n_j$.

Zgodnie z formułą (3.11) nowa pozycja cząstek $X(t+1)$ jest uzyskiwana przez agregację poprzednich składników stanu: $X(t)$ i $V(t)$. Ponieważ elementy macierzy $X(t) + V(t)$ mogą przyjmować wartości z przedziału $[-v_{max}, v_{max} + 1]$, funkcja S_x jest odpowiedzialna za przekształcenie jej w poprawną macierz permutacji, tj. posiadającą dokładnie jedną 1 w każdym wierszu i kolumnie. W rzeczywistości S_v jest procedurą, a nie funkcją, ponieważ zawiera elementy losowe.

Zaimplementowano trzy warianty procedur S_x :

1. *GlobalMax*(X) - iteracyjnie wyszukuje x_{rc} , maksymalny element w macierzy X , ustawia go na 1 i zeruje wszystkie elementy w wierszu r i c .
2. *PickColumn*(X) - wybiera kolumnę c z X , wybiera maksymalny element x_{rc} , zastępuje go 1 i zeruje inne elementy w r i c .
3. *SecondTarget*(X, Z, d) - podobny do *GlobalMax*(X), w pierwszych d iteracjach ignoruje elementy x_{ij} , takie, że $z_{ij} = 1$ (jako parametr Z jest używane rozwiązanie X z ostatniej iteracji).

W eksperymentach, w których zastosowano procedurę agregacji *GlobalMax*, algorytm zatrzymywał się w minimum lokalnym, nawet jeśli prędkości cząstek były dalekie od zera. Przyczyny takiego zachowania przedstawiono na poniższym przykładzie o rozmiarze 3×3 :

$$X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad V = \begin{bmatrix} 7 & 1 & 3 \\ 0 & 4 & 5 \\ 2 & 3 & 2 \end{bmatrix}$$

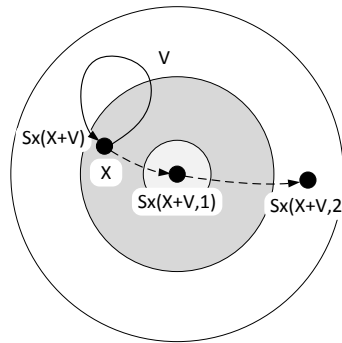
$$X + V = \begin{bmatrix} 8 & 1 & 3 \\ 0 & 4 & 6 \\ 2 & 4 & 2 \end{bmatrix} \quad S_x(X + V) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Dla opisanego przypadku w kolejnych iteracjach $X(t+1) = X(t)$, póki inna cząstka nie będzie w stanie zmienić wartości wyrażenia $(P^G(t) - X(t))$ w formule wyliczającej prędkość (3.10).

Rozwiązaniem tego problemu może być zmiana kierunku ruchu cząsteczki, ignorując $k < n$ elementów, które w rozwiązaniu $X(t)$ są ustawione na 1. Odpowiedni dobór wartości parametru k umożliwi dotarcie do innych, być może lepszych rozwiązań. Jeśli zostaną wybrane elementy maksymalne w pozostałej części macierzy, oznaczone tutaj jako $X \circledast_k V$, to nadal niosą one szansę poprawy wartości funkcji celu.

$$X \circledast_{k=3} V = \begin{bmatrix} 0 & 1 & \textcircled{3} \\ 0 & \textcircled{4} & 0 \\ \textcircled{2} & 0 & 2 \end{bmatrix} \quad (3.12)$$

Można zaobserwować, że dla $k = 1$ postać macierzy permutacji jest dokładnie taka sama, jak wynikałoby to z procedury *GlobalMax*, jednak ustawienie $k = 3$ pozwala na znalezienie innego rozwiązania.



Rysunek 3.23: Idea działania procedury *SecondTarget*.

OpenCL to standard zapewniający wspólny język, programowanie interfejsów i abstrakcję sprzętu dla heterogenicznych platform, w tym GPU, procesorów wielordzeniowych, procesorów DSP i FPGA. Pozwala to na przyspieszenie obliczeń poprzez dekomponowanie ich w zestaw równoległych zadań (*work items*) działających na osobnych danych.

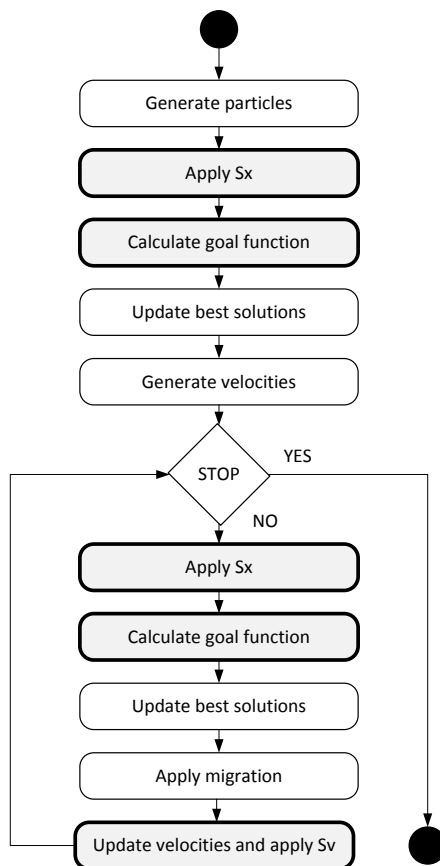
Program na platformie OpenCL jest dekomponowany na dwie części: sekwencyjną wykonywaną przez CPU *host* i równoległą wykonywaną przez *jednostkę wielordzeniową*. Funkcje wykonywane na urządzeniach noszą miano się *jądra* (lub *kernels*). Są napisane w języku będącym odmianą C z pewnymi ograniczeniami związanymi ze słowami kluczowymi i typami danych. Po załadowaniu po raz pierwszy, jądra są automatycznie tłumaczone na zestaw instrukcji urządzenia docelowego.

OpenCL obsługuje organizację danych w tablice (1D), macierze (2D) i woluminy (3D). Każdy element danych jest identyfikowany za pomocą od 1 do 3 indeksów, np. $d[i][j]$ dla tablic dwuwymiarowych. *Work item* to zaplanowana instancja jądra, która ma dostęp do danych za pomocą indeksów z pewnego zakresu. Przykładowo, tablica 2D danych $n \times m$ powinna być przetworzona przez $n \cdot m$ instancji jądra (*kernels*), które są przypisane za pomocą pary indeksów (i, j) , $0 \leq i < n$ i $0 \leq j < m$. Indeksy te służą do identyfikacji pozycji danych przypisanych do jądra.

Dodatkowo jądra mogą być zorganizowane w bloki, np. odpowiadającym częściom macierzy i synchronizowane przy użyciu mechanizmu lokalnej bariery. Jednak pojawiają się tu ograniczenia sprzętowe platformy związane z liczbą *work-items* i ilością dostępnej pamięci.

OpenCL wykorzystuje trzy typy pamięci: globalną (wymienianą między hostem a urządzeniem), lokalną dla grupy roboczej i prywatną dla elementu pracy. Wszystkie te ograniczenia

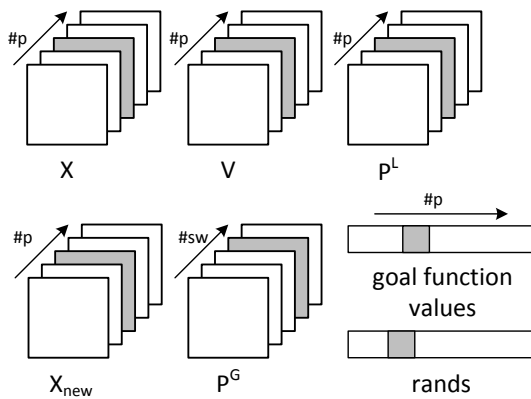
powodują, że implementacja algorytmów optymalizacyjnych korzystających z dobrodziejstw GPU jest wymagającym zadaniem.



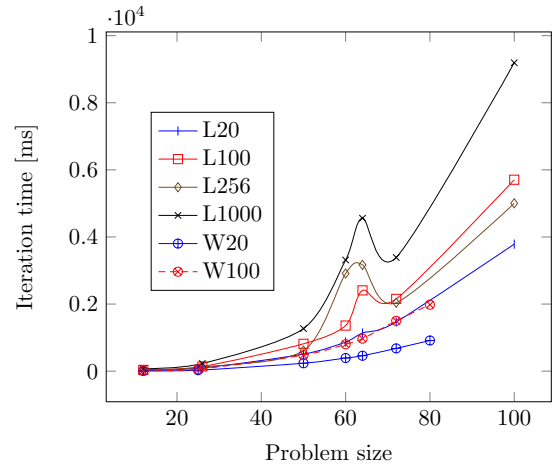
Rysunek 3.24: Bloki funkcjonalne algorytmu implementowanego w OpenCL.

Dane opisujące cząstki zawarte są w pewnej liczbie macierzy (Rys. 3.25): X i X_{new} - rozwiązania, P^L - lokalne najlepsze cząstki i V - prędkość. Wszystkie są przechowywane w dużych tablicach współdzielonych przez wszystkie cząstki. Odpowiednią część tabeli należąca do cząstki można zidentyfikować na podstawie *particle id* przydzielonej do *kernela*. Ponadto, podczas aktualizacji prędkości, cząstki odwołują się do tablicy P^G indeksowanej przez *identyfikator roju*.

Zaprezentowane w [27] wyniki wskazują, że zastosowanie GPU w optymalizacji zagadnień permutacyjnych, nawet w wersji dostępnej w przeciętnym laptopie, przy zastosowaniu odpowiedniego modelu obliczeniowego, pozwala osiągnąć znaczne przyspieszenie obliczeń.



Rysunek 3.25: Zmienne globalne zastosowane w algorytmie PSO na platformie OpenCL.



Rysunek 3.26: Czas jednej iteracji dla różnych rozmiarów problemu. L20, L100, L256 i L1000: 20, 100, 256 i 1000 cząstek (laptop), W20 i W100: 20 i 100 cząstek (stacja robocza).

Implementacja algorytmu pszczelego (Bees Algorithm) na platformie CUDA

Badania związane z przyspieszeniem algorytmów optymalizacyjnych poprzez ich implementację na platformie sprzętowej GPU (GUGPU), związane były również z implementacją algorytmu pszczelego (BA) na platformie CUDA (*Compute Unified Device Architecture*) opracowanej przez firmę NVIDIA ([19]).

CUDA to platforma programowo-sprzętowa, która podobnie jak OpenCL, umożliwia równoległe wykonywanie programów na procesorach graficznych. Programy te (zwane *kernelami*) mogą być wykonywane przez dużą liczbę równoległych wątków, przypisanych do setek lub tysięcy rdzeni.

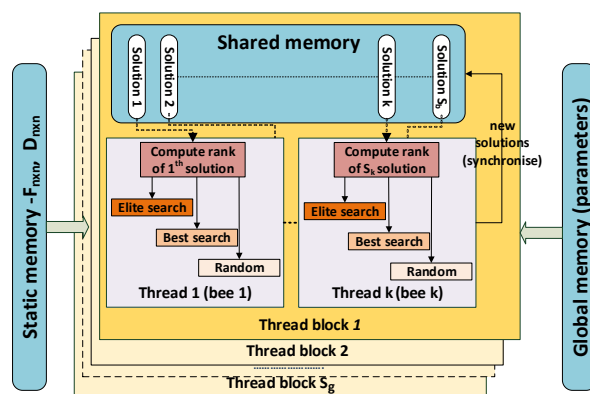
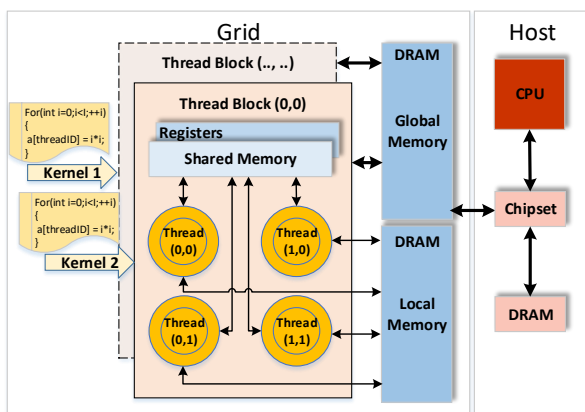
W przypadku platformy sprzętowej GPU nakłady obliczeniowe związane z obsługą wątków są niewielkie, co zapewnia niewielkie nakłady na ich tworzenie oraz bardzo szybkie przełączanie pomiędzy nimi. Ważną cechą CUDA jest współpraca wątków opartych na bardzo szybkiej pamięci współdzielonej (Rys. 3.27). Za pośrednictwem tej pamięci, wątki podzielone na bloki o maksymalnym rozmiarze równym 1024, mogą być synchronizowane i komunikować się między sobą (Rys 3.27). To rozwiązanie sprawia, że współpraca wątków jest skalowalna. Wątki w różnych blokach nie mogą się komunikować bez pośrednictwa procesora CPU. Innymi typami pamięci wykorzystywanej przez wątki są: rejestry (on-chip), pamięć lokalna (off-chip, uncached) i pamięć globalna (off-chip, duża, uncached i dostępna dla wszystkich wątków). W ramach jednego bloku CUDA zapewnia podstawowe mechanizmy wspierające synchronizację np. lokalne bariery.

W celu implementacji algorytmu pszczelego na platformie CUDA wprowadzono w nim modyfikacje umożliwiające przyspieszenie obliczeń poprzez jego równoległą realizację, w której to przestrzeń rozwiązań jest przeszukiwana przez wiele niezależnych rojów.

Zasadniczą różnicą w stosunku do przedstawionego wcześniej algorytmu PSO (Rys 3.24), jest przetwarzanie pojedynczego rozwiązania w osobnym wątku. Dzięki temu wszystkie obliczenia są realizowane na GPU, unikając konieczności czasochłonnego kopiowania danych pomiędzy GPU i CPU.

Przed startem (Algorytm 3), wszystkie niezbędne dane są ładowane do pamięci GPU. Dane te zawierają:

- Macierz przepływu $F_{n \times n}$ i macierz odległości $D_{n \times n}$ są ładowane do pamięci statycznej.
- Zestaw parametrów algorytmu MSBA: S_g - liczba bloków zawierających wątki, S_b - liczba wątków w bloku (innymi słowy liczba pszczoł w pod-roju, więc cały rój ma rozmiar $\lambda = S_g \cdot S_b$), $l_{e_{MSBA}}$ - liczba rozwiązań w elicie, $l_{b_{MSBA}}$ - liczba dobrych rozwiązań, n_e - wielkość sąsiedztwa dla elitarniej lokalizacji, n_b - wielkość sąsiedztwa dla dobrej lokalizacji, I_{max} - maksymalna liczba iteracji, LT - długość życia. Dane te są ładowane do pamięci globalnej.
- Każdy rój jest tworzony zgodnie z jednostajnym rozkładem losowym z różnym posiewem, dzięki czemu roje są zróżnicowane i każdy rój eksploruje nieco inną część przestrzeni rozwiązań. Rozwiązania są ładowane do pamięci dzielonej bloku.



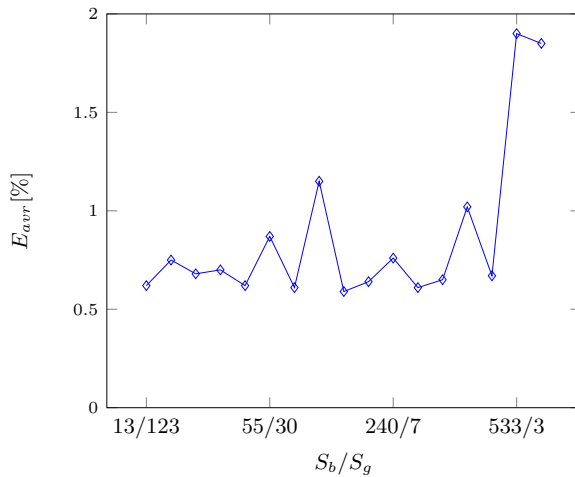
Rysunek 3.27: CUDA - model architektury.

Rysunek 3.28: Schemat działania algorytmu MSBA.

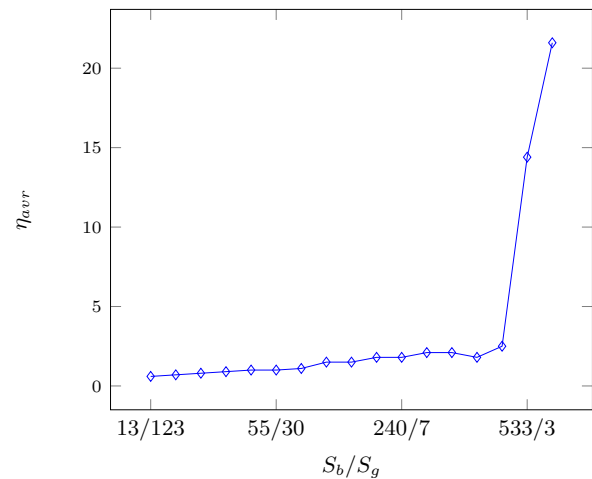
Dwie macierze $F_{n \times n}$ i $D_{n \times n}$ są ładowane do pamięci statycznej (tylko do odczytu), co zapewnia szybki dostęp do danych dla wszystkich wątków. Ilość dostępnej pamięci statycznej jest ograniczona do 64 kilobajtów, co może to stanowić problem przy ładowaniu dużych macierzy. W związku z tym, w uzasadnionych przypadkach, gdy rozmiar problemu jest duży, pamięć statyczna GPU może zostać zastąpiona globalną pamięcią GPU. Zwiększa to jednak 100-krotnie czas dostępu do danych.

Wyniki przedstawione w Tab. 3.7 wskazują na średnio 9-krotną redukcję czasu wykonywania algorytmu MSBA w porównaniu z algorytmem BA, podczas gdy uzyskane wyniki są podobne. Kolumna E zawiera $E = \frac{\varphi_{best} - \varphi_{ref}}{\varphi_{ref}} \cdot 100\%$ - procentową względną różnicę między najlepszą wartością funkcji celu z biblioteki QAPLIB i najlepszą wartością funkcji celu rozwiązania znalezionej przez algorytm MSBA lub algorytm BA. Kolumna η przedstawia stosunek czasu obliczeń - $\eta = \frac{T_A(BA)}{T_A(MSBA)}$. Ostatni wiersz w tabeli zawiera średnią wartość ($E_{avr}, \bar{E}_{MSBA/BA}, Avr$) wszystkich wartości z kolumny powyżej.

Analiza wyników z Tab. 3.8 prowadzi do wniosku, że mniejszy średni procentowy błąd algorytm uzyskuje dla dużej liczby mniejszych rojów (patrz Rys. 3.29). Dla $S_g=123$ względny średni błąd uzyskany przez algorytm MSBA i algorytm BA równa się odpowiednio 0.62% i 1.12%. Dla $S_b = 630$ i $S_g = 2$ przyspieszenie algorytmu jest największe: $\eta = 21.2$. Niestety, w tym przypadku wartość E_{avr} dla algorytmu MSBA jest równa 1.85%, która jest mniejsza niż E_{avr} algorytmu BA, jednak trzy razy większa niż najlepsze wyniki uzyskane przez algorytm MSBA dla większej liczby rojów. Można zaobserwować (Rys. 3.30), że wzrost wielkości bloku powoduje większe przyspieszenie obliczeń.



Rysunek 3.29: Średni błąd E_{avr} dla różnych liczb bloków i rozmiarów roju S_b/S_g .



Rysunek 3.30: Wartość przyspieszenia uzyskana przez algorytm MSBA dla różnej liczby bloków i rozmiaru roju S_b/S_g .

Rys. 3.31 przedstawia zmianę wartości funkcji celu dla 1600 pszczoł przetwarzanych w osobnych wątkach w algorytmie MSBA. Można zauważyć, że przebiegi wartości funkcji celu są podzielone na dwie grupy. Pierwszy charakteryzuje się znacznie gorszymi wartościami funkcji celu (górną wiązką przebiegów). Natomiast druga (niższa wiązka) przebiegów charakteryzuje

Algorithm 3: MSBA thread algorithm.

Dane wejściowe: Rozwiązanie powiązane z wątkiem π , $\varphi(\cdot)$ - funkcja kryterialna, I_{max} - maksymalna liczba iteracji, LT - czas życia rozwiązania.

Krok 1. Określ ranking r rozwiązania π , zliczając rozwiązania z pamięci współdzielonej, których wartość funkcji celu jest mniejsza niż $\varphi(\pi)$.

Krok 2. Synchronizuj wszystkie rozwiązania w bloku (bariera lokalna).

Krok 3. Jeśli $r \leq |L_e|$, wtedy losowo wybierz n_e rozwiązań z sąsiedztwa π używając procedury *2-opt*. Wybierz najlepsze rozwiązanie w sąsiedztwie

$\pi^* = \arg \min_{\pi' \in \{N(\pi) \cup \pi\}} \varphi(\pi')$ i zamień π na π^* . Przejdź do **Krok 6**.

Krok 4. Jeśli $|L_e| < r \leq |L_e| + |L_b|$, to losowo wybierz n_b rozwiązań z sąsiedztwa π , używając procedury *swap*. Wybierz najlepsze rozwiązanie w sąsiedztwie

$\pi^* = \arg \min_{\pi' \in \{N(\pi) \cup \pi\}} \varphi(\pi')$ i zamień π przez π^* . Przejdź do **Krok 6**.

Krok 5. Jeśli $|L_e| + |L_b| < r$, to zamień rozwiązanie π na nowe, generowane losowo. Przejdź do **Krok 7**.

Krok 6. Jeśli w krokach 3 lub 4 nie znaleziono lepszego rozwiązania, a π istnieje w roju dłużej niż liczba iteracji LT , to zamień to rozwiązanie na nowe, wygenerowane losowo.

Krok 7. Wstaw rozwiązanie π do pamięci współdzielonej.

Krok 8. Sprawdź warunek stopu. Jeśli jest spełniony, to STOP, w przeciwnym razie przejdź do **Krok 1**.

Tablica 3.7: Rezultaty algorytmu MSBA i BA dla 38 instancji testowych ($LT = 200$, $S_b/S_g = 400/4$, $n_e = 25$, $n_b = 18$).

Name	φ_{ref}	MSBA(GPU)			BA(CPU)			η
		φ_{best}	T_A	E	φ_{best}	T_A	E	
Bur26a	5426670	5427110	10.4	0.01	5426670	122.3	0.00	11.8
Bur26b	3817852	3817852	10.7	0.00	3817852	122.6	0.00	11.4
Bur26c	5426795	5426795	10.0	0.00	5426795	122.3	0.00	12.2
Bur26d	3821225	3821225	10.4	0.00	3821232	122.6	0.00	11.8
Bur26e	5386879	5386879	10.2	0.00	5386879	122.8	0.00	12.0
Bur26f	3782044	3782044	10.4	0.00	3782044	122.4	0.00	11.8
Bur26g	10117172	10117172	9.7	0.00	10117172	122.5	0.00	12.6
Bur26h	7098658	7098658	9.7	0.00	7098658	123.1	0.00	12.7
Chr22a	6156	6236	7.4	1.30	6266	87.3	1.79	11.8
Chr22b	6194	6286	7.5	1.49	6320	87.3	2.03	11.7
Esc32a	130	134	19.4	3.08	134	185.0	3.08	9.5
Esc32b	160	168	19.1	5.00	168	186.9	5.00	9.8
Esc32c	642	642	20.1	0.00	642	185.9	0.00	9.2
Esc32d	200	200	18.9	0.00	200	185.0	0.00	9.8
Esc32e	2	2	20.7	0.00	2	185.0	0.00	8.9
Esc32f	2	2	20.7	0.00	2	185.0	0.00	8.9
Esc32g	6	6	20.4	0.00	6	185.1	0.00	9.1
Esc32h	438	438	19.3	0.00	438	185.2	0.00	9.6
Esc64a	116	116	116.8	0.00	116	775.5	0.00	6.6
Kra30a	88900	89100	15.5	0.22	90100	163.4	1.35	10.5
Kra30b	91420	91990	15.5	0.62	91850	163.4	0.47	10.5
Lipa30a	13178	13196	16.2	0.14	13222	163.7	0.33	10.1
Lipa30b	151426	151426	15.8	0.00	151426	163.3	0.00	10.3
Lipa40a	31538	31940	34.7	1.27	31945	296.4	1.29	8.5
Lipa40b	476581	498634	34.8	4.63	487526	296.2	2.30	8.5
Lipa50a	62093	62862	61.7	1.24	62872	467.8	1.25	7.6
Lipa50b	1210244	1347609	62.5	11.35	1342719	467.7	10.95	7.5
Lipa60a	107218	108422	95.6	1.12	108419	680.2	1.12	7.1
Lipa60b	2520135	3031375	96.9	20.29	3036389	680.2	20.49	7.0
Sko42	15812	16080	38.4	1.69	16044	327.4	1.47	8.5
Sko49	23386	23892	56.6	2.16	23800	449.4	1.77	7.9
Sko56	34458	35292	78.6	2.42	35000	590.2	1.57	7.5
Ste36a	9526	9796	24.4	2.83	9854	239.4	3.44	9.8
Ste36b	15852	16438	23.7	3.70	16378	239.7	3.32	10.1
Ste36c	8239110	8393064	24.2	1.87	8408548	239.4	2.06	9.9
Tho30	149936	150910	15.4	0.65	151172	163.5	0.82	10.6
Tho40	240516	245946	33.3	2.26	245230	296.2	1.96	8.9
Wil50	49320	49244	59.7	0.88	49342	467.9	1.08	7.8
E_{avr}			30.9	1.80		262.3	1.77	8.5

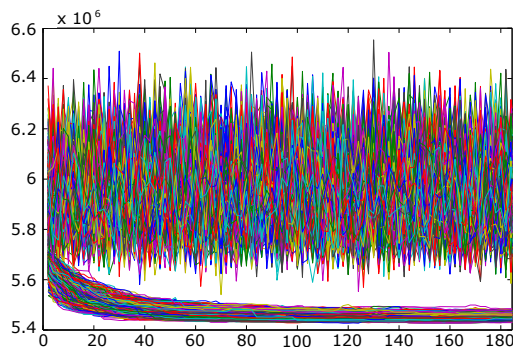
Tablica 3.8: Średnia wartość E_{avr} , T_{Aavr} i η dla 38 instancji testowych i różnych wartości S_b/S_g ($n_e = 177$, $n_b = 80$).

S_b/S_g	MSBA		BA		η_{avr}
	E_{avr}	T_{Aavr}	E_{avr}	T_{Aavr}	
13/123	0.62	205.0	1.12	120.6	0.6
16/100	0.75	186.1	1.13	124.9	0.7
20/80	0.68	166.7	1.06	125.7	0.8
27/60	0.70	148.1	1.08	137.6	0.9
40/40	0.62	135.6	1.11	131.0	1.0
55/30	0.87	133.0	1.08	134.6	1.0
80/20	0.61	106.9	1.18	118.0	1.1
160/10	1.15	57.6	1.14	88.7	1.5
178/9	0.59	57.3	1.51	85.6	1.5
200/8	0.64	46.7	1.71	82.4	1.8
240/7	0.76	46.1	1.46	82.1	1.8
260/6	0.61	36.4	1.38	74.9	2.1
280/6	0.65	38.0	1.53	79.1	2.1
320/5	1.02	41.0	1.76	73.4	1.8
400/4	0.67	28.7	1.80	70.6	2.5
533/3	1.90	4.7	2.00	67.7	14.6
630/2	1.85	2.4	2.60	52.4	21.2
Avr	0.86	84.7	1.45	97.0	3.6

się stałą poprawą wartości funkcji celu w kolejnych iteracjach. Ta grupa zawiera rozwiązania, które pochodzą od lepszych rozwiązań losowo generowanych na początku algorytmu (mówiąc inaczej, już na początku wygenerowanych w obiecujących obszarach przestrzeni rozwiązań). Rozwiązania te nigdy nie opuszczają elitarnego lub najlepszego zestawu rozwiązań, a zatem liczba kursów w grupie niższych przebiegów jest równa $(l_e + l_b)S_g$. Wysoka wartość parametru *czas życia*, która wynosi $LT = 200$, jest powodem, tego że pierwsza grupa rozwiązań, pochodzących z gorszej części rozwiązań początkowych, nie będzie miała szansy na przejście do elity lub rozwiązań najlepszych.

Analiza wyników przedstawionych na Rys. 3.29 prowadzi do wniosku, że mniejszą średnią procentową względną wartość błędu można uzyskać dla dużej liczby rojów. Dla $S_g = 123$ względna wartość błędu uzyskana przez algorytm MSBA i algorytm BA równa się odpowiednio 0,62% i 1,12%. Dla $S_b = 630$ i $S_g = 2$ przyspieszenie algorytmu jest największe: $\eta = 21,2$. Niestety, w tym przypadku wyniki uzyskane przez algorytm MSBA pogarszają się. Można zaobserwować (Rys. 3.30), że wzrost wielkości bloku powoduje większe przyspieszenie obliczeń.

Przeprowadzono również badania dotyczące własności rojów eksplorujących przestrzeń rozwiązań. Szczególnie interesująca była odpowiedź na pytanie, czy niezależne roje nie przeszukują tych samych obszarów przestrzeni rozwiązań, prowadząc do spadku efektywności algo-



Rysunek 3.31: Zmiana wartości funkcji celu dla 1600 pszczoł przetwarzanych za pomocą algorytmu MSBA dla zagadnienia `Lipa50B` ($\lambda = 1600$, $LT = 200$, $S_b/S_g = 55/30$, $n_e = 25$, $n_b = 18$).

rytmu. Aby na nie odpowiedzieć, prowadzono badania zmian entropii roju podczas eksploracji przestrzeni rozwiązań.

Entropia Shannona (*Shannon's information entropy* - SIE) może być stosowana do oceny miary różnorodności roju. Dla populacji S , gdzie można rozróżnić k klas, SIE definiuje się w następujący sposób:

$$SIE(S) = - \sum_k p_k \log p_k, \quad (3.13)$$

gdzie p_k to odsetek S zajmujący partycję populacji k w danej chwili czasu. SIE opisuje poziom chaosu. Duże wartości entropii odpowiadają małym wartościom p_k (w każdym obszarze przestrzeni rozwiązań znajduje się niewielka liczba rozwiązań), natomiast małe wartości entropii odpowiadają większym wartościom p_k (znaczną liczbą rozwiązań jest skoncentrowana w kilku obszarach przestrzeni). Dla populacji S o rozmiarze λ w iteracji t , wartość SIE dla populacji (określaną teraz skrótem SEP) można obliczyć w następujący sposób:

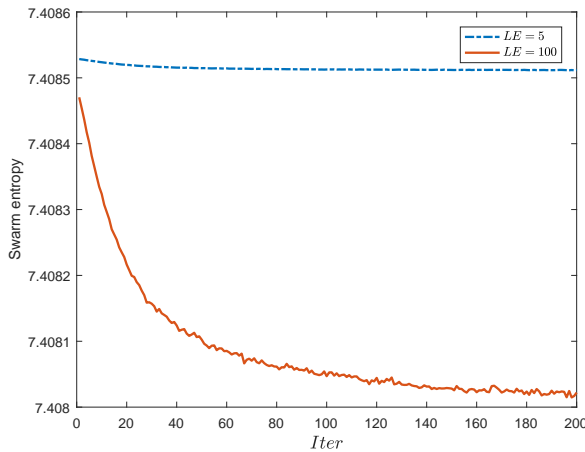
$$SEP_t(S) = - \sum_{i=1}^{\lambda} P_i(t) \log P_i(t), \quad (3.14)$$

gdzie $P_i(t) = \frac{f(\pi_i(t))}{\sum_{j=0}^{\lambda} f(\pi_j(t))}$, można interpretować jako proporcję wartości funkcji celu cząstki π_i , w iteracji t , w stosunku do pozostałych. SEP dostarcza informacje dotyczące rozkładu wartości funkcji celu. Wysokie wartości SEP odpowiadają dużej zmienności funkcji celu. Mniejsze natomiast, wskazują na podobne wartości funkcji celu. W celu oceny zachowania się roju przeprowadzono szereg eksperymentów.

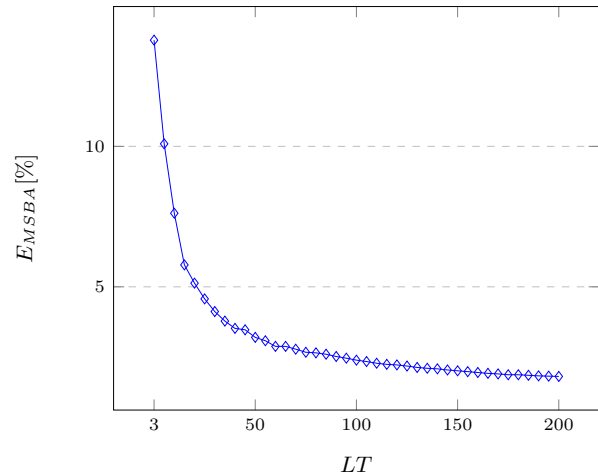
Rys. 3.32 przedstawia zmianę wartości $SEP_t(S)$ w kolejnych iteracjach dla roju przetwarzanego przez algorytm MSBA dla małej i dużej wartości czasu życia (LT) rozwiązania. Można zauważyć, że entropia praktycznie nie zmienia się w przypadku małej wartości LT , podczas gdy dla dużej wartości LT maleje, co wskazuje na utratę różnorodności populacji. Obserwacja ta dowodzi, że niska wartość LT pozwala na lepszą eksplorację obiecujących regionów przestrzeni rozwiązań.

Rys. 3.33 przedstawia wpływ wartości LT na wyniki uzyskane przez algorytm MSBA. Przedstawione wartości są średnią wartością względnego średniego procentowego błędu E_{avr} uzyskanej przez algorytm dla 38 testowych instancji problemu QAP.

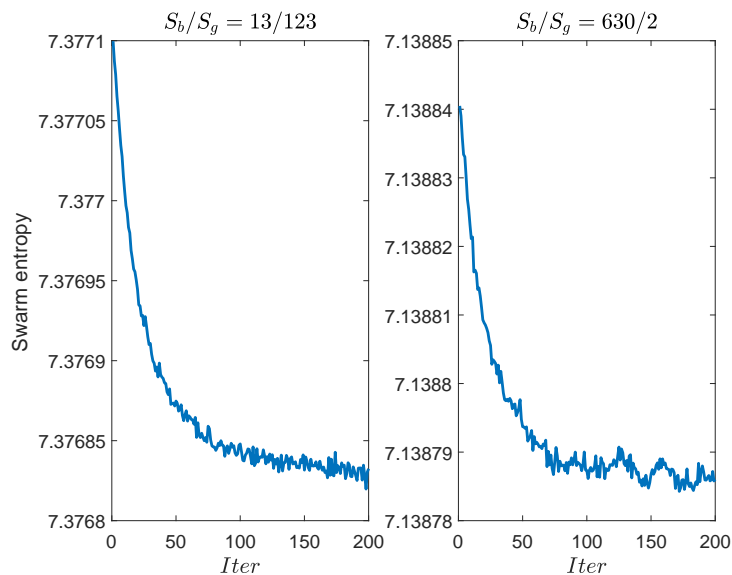
Dla $S_b/S_g = 630/2$ przestrzeń rozwiązań jest przeszukiwana tylko przez dwa niezależne roje, co sprzyja szybkiej zbieżności, przyczyniając się jednocześnie do niewielkiej poprawy wartości funkcji celu. Potwierdza to analiza Rys. 3.34 wskazująca, że entropia na początku



Rysunek 3.32: Wartości entropii dla roju przetwarzanego przez algorytm MSBA dla instancji problemu QAP Lipa50B dla $LT = 5$ i $LT = 100$ ($\lambda = 1600$, $n_e = 177$, $n_b = 80$).



Rysunek 3.33: Wpływ czasu życia (LT) na rezultaty optymalizacji (procentowy błąd względny) prowadzonej przez algorytm MSBA.



Rysunek 3.34: Wartości entropii dla roju przetwarzanego przez algorytm MSBA dla instancji problemu QAP Lipa50B dla różnej liczby pszczoł w bloku oraz różnej liczby bloków : $S_b/S_g = 13/123$ i $S_b/S_g = 630/2$ ($\lambda = 1600$, $n_e = 177$, $n_b = 80$).

jest stosunkowo niska (inaczej mówiąc, istnieje niewiele różnych wartości) i w zasadzie nie ulega zmianie już do końca działania algorytmu (zamiana wynosi zaledwie około $5 \cdot 10^{-5}$).

Odwrotną sytuację można zaobserwować dla $S_b/S_g = 13/123$, gdy przestrzeń rozwiązań jest przeszukiwana przez 123 niewielkie i niezależne roje, co poprawia eksplorację przestrzeni rozwiązań i umożliwia uzyskanie lepszych wyników. W tym przypadku na początku entropia jest stosunkowo wysoka (tj. wartości funkcji celu przyjmują szeroki zakres różnych wartości) i podczas działania algorytmu zmienia się ok. 100-krotnie więcej w porównaniu z poprzednim przykładem, ponieważ roje eksplorują i koncentrują się na przeszukiwaniu różnych obiecujących obszarów przestrzeni rozwiązań problemu QAP.

Wyniki badań wskazują, że jeśli przestrzeń rozwiązań jest eksplorowana z zastosowaniem

wielu rojów to należy położyć szczególny nacisk na implementację odpowiednich procedur odpowiedzialnych za przeszukiwanie lokalne. Dostarczają one również wiedzy na temat konstrukcji i doboru parametrów algorytmów populacyjnych, których odmianą jest algorytm pszczeleli.

3.2.5. Opracowanie algorytmów do optymalizacji ruchu drogowego z zastosowaniem arytmetyki przedziałowej

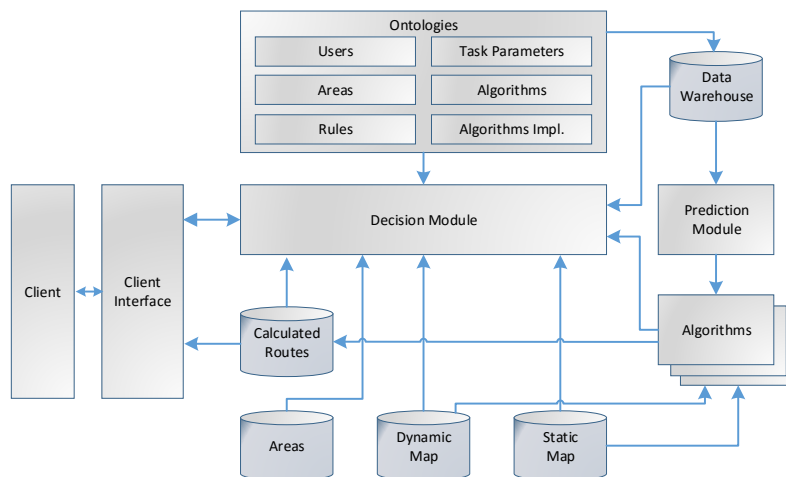
W ramach badań nad optymalizacją trudnych zagadnień dyskretnych Autor zaproponował algorytmy pozwalające na optymalizację ruchu drogowego, który z zasady charakteryzuje się wysoką niepewnością ([17]). Aby poradzić sobie z dużą złożonością obliczeniową modeli stochastycznych, do opisu parametrów ruchu drogowego zastosowano arytmetykę przedziałową. Pozwoliło to sformułować odpowiednie twierdzenia umożliwiające optymalizację decyzji o wyborze nowej trasy, gdy charakterystyka sieci drogowej, a co za tym idzie opisujących ją parametrów, ulega zmianie wskutek fluktuacji w ruchu pojazdów ([17]).

Zaproponowane rozwiązanie zostało poddane testom z zastosowaniem specjalnie opracowanego środowiska symulacyjnego wykorzystującego symulator ruchu drogowego SUMO (*Simulation of Urban MObility*). Z jego pomocą modelowano rzeczywistą sieć drogową i rzeczywiste parametry ruchu drogowego dla miasta Krakowa. Rozbudowane testy udowodniły przydatność zastosowanego podejścia i co jest z tym związane, wymierne korzyści z użycia zaproponowanego rozwiązania. Opracowany algorytm korzystający z przedziałowej opisu parametrów ruchu został zastosowany jako element większego systemu odpowiedzialnego za optymalizację ruchu drogowego w projekcie INSIGMA.

Planowanie tras pojazdów w środowisku charakteryzującym się dynamiką może być formułowane w postaci modeli optymalizacyjnych należących do klasy zagadnień \mathcal{NP} -trudnych. Aby lepiej odzwierciedlić rzeczywiste warunki zakłada się, że czasy podróży nie są dokładnie znane, ale znane są jedynie ich ograniczenia od dołu i od góry, tj. definiowane są one jako wielkości przedziałowe (interwały). Opracowane algorytmy są zintegrowane z większym systemem zarządzania ruchem, realizowanym w ramach dużego projektu. Pozwoliło to zbadać skuteczność proponowanych algorytmów i ich zdolność do wspierania uczestników ruchu drogowego w warunkach rzeczywistego ruchu drogowego. Weryfikację zaproponowanych algorytmów prowadzono także z zastosowaniem zaawansowanych symulatorów mikroskopowych ruchu drogowego korzystających z danych pozyskiwanych z rzeczywistej sieci drogowej.

Jednym z ważnych celów projektu INSIGMA, w ramach którego Autor zrealizował prezentowane rozwiązanie, jest wyznaczenie najlepszej trasy na podstawie informacji o parametrach ruchu drogowego pochodzących z różnego typu detektorów: kamer, macierzy akustycznych, lidarów, urządzeń śledzących położenie pojazdów na podstawie sygnału GPS i wielu innych. System RPS (*Route Planning System*) zapewnia wyznaczenie najlepszej trasy od punktu początkowego do docelowego. W ramach systemu opracowano różnego typu algorytmy optymalizacyjne uwzględniające zarówno preferencje użytkownika, jak i aktualny oraz historyczny stan ruchu drogowego.

Rysunek 3.35 przedstawia podstawowe elementy Systemu Planowania Trasy (RPS). Najważniejszym elementem systemu RPS jest *Moduł Decyzyjny*, który podejmuje decyzję, czy trasa zawarta w module *Calculated Routes* powinna zostać ponownie obliczona lub/i wysłana do kierowców za pomocą *Interfejsu Klienta*. Jeśli czasy przejazdu po drogach ulegną zmianie, trasy w module *Wyznaczone Trasy* są aktualizowane przy użyciu, algorytmu DHN. Moduł *Algorytmy* zawiera zarówno algorytmy dokładne i przybliżone do optymalizacji ścieżki



Rysunek 3.35: Najważniejsze elementy systemu RPS (Route Planning System).

(np. DHN), jak również algorytmy do prognozowania parametrów ruchu (np. sieć neuronowa, drzewo decyzyjne).

Gdy mamy do czynienia z rzeczywistymi problemami, rzadko kiedy można uniknąć niepewności. Na poziomie empirycznym niepewność jest nieodłącznym towarzyszem jakiegokolwiek pomiaru. Różnicę między zmierzoną a rzeczywistą wartością określa się *błędem pomiaru*. Wartość bezwzględna błędu pomiaru jest zwykle ograniczona. Jest zatem pewne, że faktyczna (nieznana) poszukiwana wartość należy do przedziału, w którym punkt środkowy jest wartością zmierzoną, a promień jest górną granicą dla (wartość bezwzględna) błędu pomiaru. Arytmetyka przedziałowa jest również bardzo przydatna w przypadku błędów zaokrąglania, które są nieodłączne w obliczeniach zmiennoprzecinkowych. Dlatego niedokładność, przybliżenie lub niepewność w zakresie znajomości dokładnych wartości parametrów fizycznych i technicznych można wygodnie modelować za pomocą interwałów. Rola interwałów stale rośnie w analizie numerycznej.

W zaproponowanym podejściu interwałowym do planowania trasy, niepewny czas podróży \tilde{t} jest reprezentowany przez zamknięty interwał $\mathbf{t} = [\underline{t}, \bar{t}]$. Natomiast brak jest dalszych informacji, które wartości z tego przedziału są bardziej prawdopodobne.

Podstawowe operatory $\circ \in \{+, -, *, /\}$ są zdefiniowane na zbiorze przedziałów w następujący sposób:

$$\begin{aligned}
 \mathbf{x} + \mathbf{y} &= [\underline{x} + \underline{y}, \bar{x} + \bar{y}], \\
 \mathbf{x} - \mathbf{y} &= [\underline{x} - \bar{y}, \bar{x} - \underline{y}], \\
 \mathbf{x} \cdot \mathbf{y} &= [\min\{\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\underline{y}, \bar{x}\bar{y}\}, \max\{\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\underline{y}, \bar{x}\bar{y}\}], \\
 \mathbf{x}/\mathbf{y} &= \mathbf{x} \cdot [1/\bar{y}, 1/\underline{y}], 0 \notin \mathbf{y}.
 \end{aligned} \tag{3.15}$$

Aby wykorzystać wielkości przedziałowe w systemie planowania trasy, wymagane jest zdefiniowanie relacji pomiędzy interwałami. W zaproponowanych algorytmach zdecydowano się zastosować probabilistyczne podejście wprowadzone przez Sevastjanova i Roga.

Niech $\mathbf{a} = [a, \bar{a}]$ i $\mathbf{b} = [b, \bar{b}]$ będą przedziałami i niech $a \in \mathbf{a}$ i $b \in \mathbf{b}$ będą zmiennymi losowymi o rozkładzie jednostajnym. Jeśli \mathbf{a} i \mathbf{b} nakładają się, można rozróżnić pewne rozłączne podprzedziały. Wylosowanie zmiennych losowych a i b w podprzedziałach $[a_1, b_1]$, $[b_1, a_2]$, $[a_2, b_2]$ może być traktowany jako zbiór niezależnych zdarzeń losowych. Niech $H_k : a \in \mathbf{a}_i, b \in \mathbf{b}_j$, dla $k = 1$ do n , będą losowymi zdarzeniami, gdzie \mathbf{a}_i i \mathbf{b}_j są pewnymi podprzedziałami przedziałów \mathbf{a} i \mathbf{b} zgodnie z $\mathbf{a} = \bigcup_i \mathbf{a}_i$ i $\mathbf{b} = \bigcup_i \mathbf{b}_i$. Niech $P(H_k)$ będzie prawdopodobieństwem zdarzenia H_k , a $P(\mathbf{b} > \mathbf{a} | H_k)$ będzie prawdopodobieństwem warun-

kowym $\mathbf{b} > \mathbf{a}$, pod warunkiem zajścia H_k . W związku z tym prawdopodobieństwo złożone można wyrazić w następujący sposób:

$$P(\mathbf{b} > \mathbf{a}) = \sum_{k=1}^n P(H_k)P(\mathbf{b} > \mathbf{a}|H_k).$$

Wynikowa formuła dla przypadku nakładania się interwałów jest następująca:

$$P(\mathbf{b} > \mathbf{a}) = 1 - \frac{1}{2} \frac{(a_2 - b_1)^2}{(a_2 - a_1)(b_2 - b_1)}.$$

Wyniki uzyskane przez Sevastjanova i Roga dla wszystkich możliwych nakładających się przedziałów są zawarte w Tab. 3.9, gdzie oczywiste przypadki (bez nakładania się) są pomijane.

Tablica 3.9: Typowe przypadki porównania interwałów (Sevastjanov i Rog).

Case	$P(\mathbf{a} > \mathbf{b})$	$P(\mathbf{a} = \mathbf{b})$
$b_1 < a_1 = a_2 < b_2$	$\frac{b_1 - a_1}{a_2 - a_1}$	0
$a_1 < b_1 = b_2 < a_2$ and $b_1 < a_2$ and $b_1 = b_2$	$\frac{b_2 - a_1}{b_2 - b_1}$	0
$a_1 \leq b_1 \wedge b_2 \leq a_2$	$\frac{b_1 - a_1}{a_2 - a_1} + \frac{1}{2} \frac{a_2 - a_1}{b_2 - b_1}$	$\frac{b_2 - b_1}{a_2 - a_1}$
$b_1 \leq a_1 \wedge a_2 \leq b_2$	$\frac{b_2 - a_2}{b_2 - b_1} + \frac{1}{2} \frac{a_2 - a_1}{b_2 - b_1}$	$\frac{a_2 - a_1}{b_2 - b_1}$
$a_1 \leq b_1 \wedge a_2 \leq b_2 \wedge b_1 \leq a_2$	$1 - \frac{1}{2} \frac{(a_2 - b_1)^2}{(a_2 - a_1)(b_2 - b_1)}$	$\frac{(a_2 - b_1)^2}{(a_2 - a_1)(b_2 - b_1)}$
$b_1 \leq a_1 \wedge b_2 \leq a_2 \wedge a_1 \leq b_2$	$1 - \frac{1}{2} \frac{(b_2 - a_1)^2}{(a_2 - a_1)(b_2 - b_1)}$	$\frac{(b_2 - a_1)^2}{(a_2 - a_1)(b_2 - b_1)}$

Aby było możliwe jest znalezienie trasy, na którą nie wpływają zmiany czasu podróży na poszczególnych drogach, należy wprowadzić specjalne założenia dotyczące szukanej trasy. Montemanni definiuje dwa typy problemów optymalizacyjnych: całkowicie odporny problem najkrótszej ścieżki (*absolute robust shortest path problem*) i całkowicie odporny na odchylenia problem najkrótszej ścieżki (*robust deviation shortest path problem*).

Załóżmy, że jakiś scenariusz r jest jednym z wielu możliwych realizacji kosztów krawędzi. *Najbardziej odporna trasa* (*absolute robust shortest path* - ABS) to ścieżka, która dla wszystkich możliwych scenariuszy r ma najmniejszy (spośród wszystkich ścieżek od miejsca początkowego do miejsca docelowego) maksymalny koszt:

$$(ABS) \quad \min_{y \in \{0,1\}^{|E|}} \max_{c \in F} c^T y, \\ s.t. \quad Ny = e_s - e_u.$$

Różnica między kosztem ścieżki od punktu początkowego do docelowego, a kosztem najkrótszej ścieżki od punktu początkowego do punktu docelowego w scenariuszu r nazywana jest *odpornym odchyleniem*.

Odporne odchylenie najkrótszej ścieżki (*robust deviation of the shortest path* - RD) to ścieżka, która dla wszystkich możliwych scenariuszy r ma najmniejsze (spośród wszystkich ścieżek od miejsca początkowego do miejsca docelowego) *maksimum*:

$$(RD) \quad \min_{y \in \{0,1\}^{|E|}} \max_{c \in F} (c^T y - sp(c)), \\ s.t. \quad Ny = e_s - e_u.$$

Zarówno problem ABS, jak i RD to problemy \mathcal{NP} -trudne, a zatem znalezienie rozwiązania optymalnego dla większych problemów jest praktycznie niemożliwe. W związku z tym do poszukiwania najkrótszej ścieżki zaproponowano algorytm PSO (*Particle Swarm Optimisation*) stosujący opis interwałowy. Algorytm ten był zastosowany we wczesnej wersji systemu RPS, jednak dostarcza on zbyt pesymistyczny wynik (wybiera drogę nienarażoną na zmiany, ale zbyt długą), stąd jego praktyczne zastosowanie jest ograniczone.

W związku z tym, stosując interwały do opisu niepewności, zaproponowano model, który umożliwi ocenę wpływu zmiany czasów przejazdu w całej sieci drogowej na wybrane trasy.

Niech model sieci drogowej zostanie zdefiniowany przez graf ważony $G = (V, E)$ z wagami $w : E \rightarrow \mathbb{R}$, przypisującymi wartość rzeczywistą (wagę) do każdej krawędzi $(u, v) \in V \times V$ grafu G . Założymy, że wagi krawędzi, które określają czas podróży, mogą zmieniać się w czasie (tj. są funkcjami czasu).

Rozważmy typową sytuację występującą w ruchu miejskim. Pojazd porusza się od punktu początkowego s (wierzchołek na wykresie) do punktu docelowego u , używając najkrótszej trasy obliczonej na podstawie chwilowych wartości czasów przejazdu. Jeśli podczas podróży zmieni się czas pokonania pewnej drogi (odpowiadającej krawędzi grafu), to może się również zmienić najkrótsza ścieżka. Zaproponowane twierdzenia pozwalają nam stwierdzić, czy nowa najkrótsza ścieżka musi zostać ponownie przeliczona za pomocą algorytmu optymalizacyjnego i wysłana do kierowcy.

Theorem 3. *Wydłużenie krawędzi*

Dla grafu $G = (V, E)$, niech p będzie najkrótszą ścieżką od wierzchołka początkowego s do wierzchołka docelowego u w G takiego, że p jest długości $L(s; u; p; G)$ i p zawiera krawędź $a = \{v_1, v_2\}$, to znaczy ma postać $p = (s, \dots, v_1, v_2, \dots, u)$. Niech teraz p' będzie długością najkrótszej ścieżki $L(s; u; p'; G')$ z s do u w grafie $G' = (V, E \setminus \{a\})$. Załóżmy również, że krawędź a została wydłużona o $\Delta a > 0$. Wtedy, ścieżka p pozostanie najkrótszą ścieżką w G jeśli

$$\Delta a < L(s, u, p', G') - L(s, u, p, G).$$

Dowód (Theorem 3). Można określić dwa następujące przypadki:

- i. Krawędź a nie jest częścią ścieżki p . Wtedy ścieżka p pozostaje najkrótszą ścieżką od s do u w G , niezależnie od długości krawędzi a .
- ii. Krawędź a to część ścieżki p . Wtedy, wydłużenie krawędzi a o Δa powoduje odpowiednie wydłużenie ścieżki p , a zatem ścieżka p może nie być już najkrótszą ścieżką w G . Będzie jednak tak w przypadku, gdy istnieje ścieżka p' , która nie zawiera krawędzi a i jest krótsza niż p , tj. $L(s, u, p, G) + \Delta a > L(s, u, p', G)$. Aby ustalić czy taka ścieżka istnieje wystarczy obliczyć najkrótszą ścieżkę w grafie $G' = (V, E \setminus \{a\})$ nie zawierającym krawędzi a .

□

Theorem 4. *(Skrócenie krawędzi)*

W grafie $G = (V, E)$, niech p będzie najkrótszą ścieżką od wierzchołka początkowego s do wierzchołka docelowego u w G takiego, że p jest długości $L(s; u; p; G)$ i niech $G' = (V, E \setminus \{a\})$, w którym $a = \{v_1, v_2\}$. Załóżmy, że krawędź a została skrócona o $\Delta a > 0$. W takim przypadku, prawdziwe są stwierdzenia:

- (i) *Jeśli a jest częścią p , to p pozostaje najkrótszą ścieżką w G , niezależnie od wielkości Δa .*

(ii) Jeśli a nie jest częścią p , to p pozostaje najkrótszą ścieżką, jeśli zostaną spełnione następujące warunki:

$$L(s, v_1, G') + l(a) - \Delta a + L(v_2, u, G') > L(s, u, G), \quad (3.16)$$

$$L(u, v_1, G') + l(a) - \Delta a + L(v_2, s, G') > L(s, u, G), \quad (3.17)$$

gdzie $l(a)$ jest długością krawędzi a przed skróceniem, $L(s, v_1, G')$ ($L(u, v_1, G')$) jest długością najkrótszej ścieżki pomiędzy s i v_1 (u i v_1), i $L(v_2, u, G')$ ($L(v_2, s, G')$) jest długością najkrótszej ścieżki pomiędzy v_2 i u (v_2 i s) w G' .

Dowód. (Theorem 4)

(i) Niech krawędź a będzie częścią ścieżki p . Jeśli p jest najkrótszą ścieżką z s do u , to $L(s, u, p, G) < L(s, u, p', G)$ dla każdej ścieżki p' z s do u w G . Skracając ścieżkę a możemy skrócić p i inne ścieżki, które zawierają krawędź a (ścieżki, które nie zawierają krawędzi a możemy pominąć). Nowa długość ścieżki p i innych rozważanych ścieżek, wynosi odpowiednio, $L(s, u, p, G) - \Delta a$ i $L(s, u, p', G) - \Delta a$. Stąd jeśli

$$L(s, u, p, G) < L(s, u, p', G),$$

to

$$L(s, u, p, G) - \Delta a < L(s, u, p', G) - \Delta a,$$

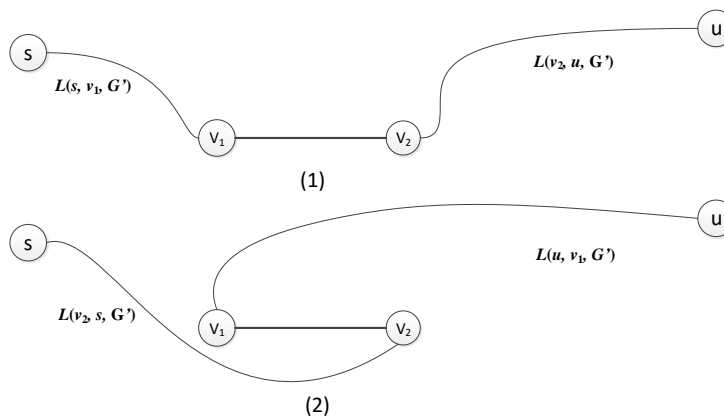
co oznacza, że p jest wciąż najkrótszą ścieżką z s do u w G .

(ii) Niech krawędź a będzie częścią ścieżki p . Wtedy, p może nie być już najkrótszą ścieżką od s do u w G , natomiast inna ścieżka zawierająca a może nią się stać. Zatem konieczne jest sprawdzenie, czy dwie ścieżki przedstawione na Rys. 3.36 nie są krótsze niż p . Ponieważ każda pod-ścieżka najkrótszej ścieżki jest najkrótszą ścieżką między dwoma dowolnymi wierzchołkami na tej ścieżce (zgodnie z zasadą optymalności Bellmana), wystarczające jest wyznaczenie najkrótszej ścieżki w grafie $G' = (V, E \setminus \{a\})$ z początkowego wierzchołka s do v_1 , najkrótszej ścieżki z v_2 do u (Rys. 3.36(1)) i odpowiednio najkrótszej ścieżki z s do v_2 oraz najkrótszej ścieżki z v_1 do u (Rys. 3.36(2)). Przez odpowiednie połączenie obu par ścieżek z krawędzią a , jest wyznaczana najkrótsza ścieżka z s i u , która zawiera krawędź a . Jeśli jedna z tych dwóch ścieżek jest krótsza niż p , wtedy ścieżka ta jest najkrótszą ścieżką pomiędzy s i u po skróceniu krawędzi a .

□

Opracowany system planowania tras (RPS) otrzymuje informacje o aktualnym stanie ruchu z czujników wideo. Na podstawie wartości parametrów z czujników RPS, czas podróży dla każdej trasy (krawędź (i, j)) jest obliczany i zapisywany w *Mapie Dynamicznej*. Topologia dróg jest zawarta w *Mapie Statycznej*. Dla każdej trasy w *Mapie Dynamicznej* zapisywana jest wartość interwałowa opisująca minimalny i maksymalny zmierzony czas podróży w ostatnich $(n - 1)\delta$ sekundach:

$$\mathbf{T}_{ij} = [\min(T_{ij}(t_s - i\delta)), \max(T_{ij}(t_s - i\delta))], i = 0, 1, \dots, n - 1. \quad (3.18)$$



Rysunek 3.36: Dwa przypadki ścieżek zawierających $a = \{v_1, v_2\}$: (1) jeśli droga przebiega najpierw przez v_1 lub (2) gdy droga najpierw przebiega przez v_2 .

Ponadto są również przechowywane średnie ruchome zmierzone czasy podróży w ostatnich $(n - 1)\delta$ sekundach $\bar{T}_{ij}(t_s)$:

$$\bar{T}_{ij}(t_s) = \frac{1}{n} \sum_{i=0}^{n-1} T_{ij}(t_s - i\delta), \quad (3.19)$$

gdzie $t_s - (n - 1)\delta, \dots, t_s - \delta, t_s$ są chwilami ostatnich n pomiarów czasu przejazdu na odcinkach. Zwykle przyjmuje się, że okres pomiaru δ wynosi od 20 do 30 sekund, a n od 10 do 15.

W obecnej wersji systemu RPS do obliczania najkrótszej trasy wykorzystywany jest wersja algorytmu *Dynamic Highway-Node Routing* (DHN), zaproponowany przez Schuttersa i Sandersa w wersji serwerowej. Algorytm ten ma możliwość szybkiego i łatwego aktualizowania czasu podróży dla najkrótszej ścieżki, gdy zmieni się czas pokonania jednej lub kilku krawędzi w sieci drogowej.

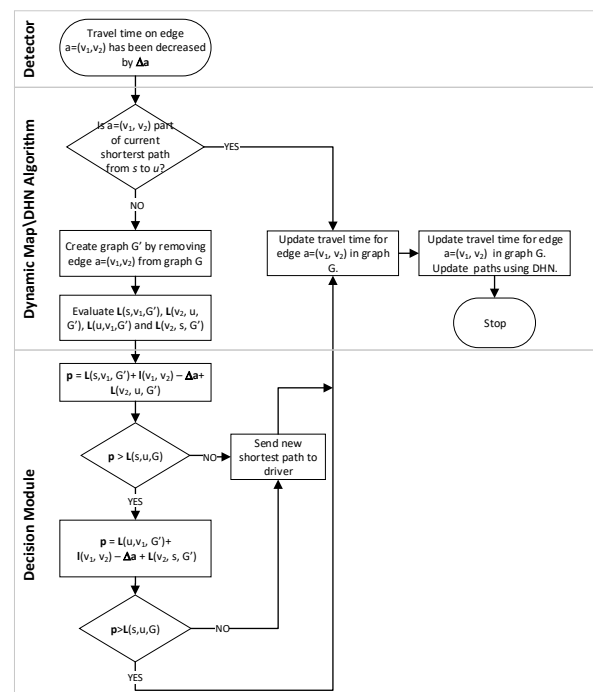
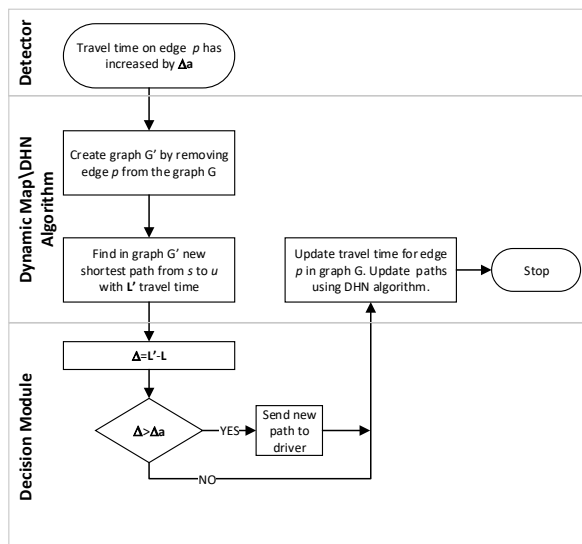
Średnia wartość ruchoma (3.19) jest używana w algorytmie wyznaczania najkrótszej ścieżki. Aby system RPS był bardziej odporny na niewielkie fluktuacje czasu podróży, decyzja o zmianie trasy na nową jest podejmowana na podstawie *interwałowych czasów przejazdu* (odróżnione pogrubioną czcionką) oraz Twierdzeń 3 i 4. Wartości zdefiniowane w tych twierdzeniach są definiowane za pomocą wartości przedziałowych w następujący sposób:

- (i) Niech $p = (a_1, a_2, \dots, a_k)$ jest najkrótszą ścieżką (listą krawędzi a_i) uzyskaną przy użyciu algorytmu DHN, gdzie każdy czas przejazdu krawędzi przyjmuje wartość prostej średniej kroczącej (równanie (3.19)).
- (ii) W przypadku wydłużenia krawędzi (Twierdzenie 3), wartości w równaniu $\Delta = \mathbf{L}(s, u, p', G') - \mathbf{L}(s, u, p, G)$ są traktowane jako wartości przedziałowe (równanie 3.18). W szczególności, $\mathbf{L}(s, u, p', G')$ i $\mathbf{L}(s, u, p, G)$ są interpretowane jako suma wartości interwałowych wag krawędzi w najkrótszej ścieżce uzyskanej za pomocą algorytmu DHN. Teoria algebry interwałowej służy do określania, czy warunki zdefiniowane w Twierdzeniu 3 są spełnione, czy też nie.
- (iii) W przypadku skrócenia (Twierdzenie 4) wszystkie wartości są interpretowane jako wartości przedziałowe (równanie 3.18)). W szczególności $\mathbf{L}(s, v_1, G')$, $\mathbf{l}(v_1, v_2)$, $\Delta \mathbf{a}$, i $\mathbf{L}(v_2, u, G')$, $\mathbf{L}(s, u, G)$, $\mathbf{L}(u, v_1, G')$, $\mathbf{l}(v_1, v_2)$, $\Delta \mathbf{a}$, $\mathbf{L}(v_2, s, G')$, $\mathbf{L}(s, u, G)$, $\mathbf{L}(s, u, p', G')$ i $\mathbf{L}(s, u, p, G)$ są interpretowane jako interwały. Teoria algebry interwa-

lowej służy do określania, czy warunki zdefiniowane w Twierdzeniu 4 są spełnione, czy nie.

W celu porównania wartości interwałowych zastosowano metodę przedstawioną w Tab. 3.9. W związku z tym, prawdopodobieństwo użyte podczas weryfikacji warunków określonych w Twierdzeniach 3 i 4 jest parametrem, który należy określić przed uruchomieniem algorytmu, na przykład na podstawie badań symulacyjnych.

Algorytmy przedstawione na Rys. 3.37 i 3.38 opisują przepływ decyzji między modułami RPS (Rys. 3.35), w celu podjęcia decyzji o wysłaniu (lub nie) nowej trasy do kierowców. Badania testowe przeprowadzono m.in. bazując na grafie odpowiadającym ulicom miasta Krakowa, stosując dane pozyskane z detektorów nadzorujących ruch drogowy w tym mieście (graf posiada 4000 krawędzi).

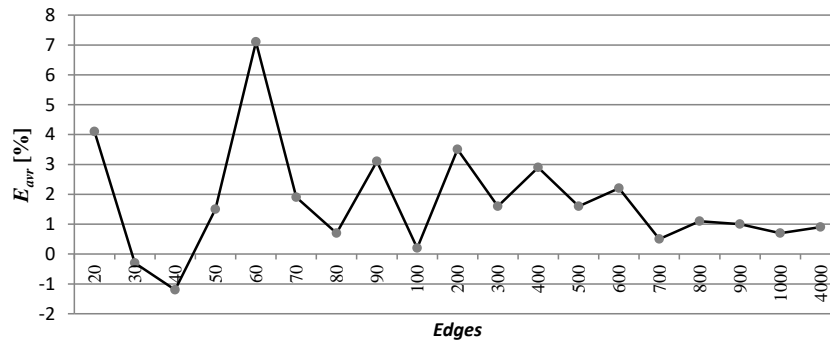


Rysunek 3.37: Przepływ decyzji w systemie RPS w przypadku wydłużenia czasu przejazdu.

Rysunek 3.38: Przepływ decyzji w systemie RPS w przypadku skrócenia czasu przejazdu.

Na Rys. 3.39 wartość $E_{avr} = \frac{\bar{t}_{avr} - \bar{t}_{opt_avr}}{\bar{t}_{avr}} \cdot 100\%$ określa względną różnicę procentową pomiędzy średnimi czasami podróży z optymalizacją stosującą zaproponowane podejście i bez niej.

Wyniki testów eksperymentalnych pokazują, że wdrożenie proponowanej „inteligentnej” metody w sterowania ruchem pozwalają skrócić średni czas podróży o ponad 1%. Informacje pochodzące z dużej liczby czujników zainstalowanych na ulicach umożliwiają optymalizację ruchu samochodowego, co w konsekwencji przyczynia się do poprawy jakości powietrza.



Rysunek 3.39: Względna różnica procentowa między średnimi czasami podróży z optymalizacją i bez optymalizacji dla grafów o liczbie krawędzi od 20 do 4000 (Kraków).

3.2.6. Opracowanie algorytmu rojowego (PSO) do wyznaczania najlepszej trasy oraz sterowania ruchem drogowym poprzez dynamiczny dobór długości faz sygnalizacji świetlnej

Zastosowanie algorytmu przybliżonego pozwoliło na optymalizację trasy w sieci drogowej w czasie rzeczywistym, uwzględniając częste zmiany parametrów opisujących ruch drogowy (natężenie i gęstość ruchu, czas przejazdu). ([1], [6], [9]). Opracowane rozwiązanie zostało przebadane dla rzeczywistej sieci drogowej i rzeczywistych parametrów ruchu drogowego z zastosowaniem środowiska symulacyjnego SUMO (*Simulation of Urban MObility*). Po etapie badań symulacyjnych, opracowane rozwiązanie zostało wdrożone i poddane testom na rzeczywistym skrzyżowaniu.

Zagadnienie optymalizacji trasy w środowisku miejskim

Zaproponowany dla systemu planowania trasy RPS model sieci transportowej jest ukierunkowanym multigrafem z wagami (etykietami) zdefiniowanymi dla łuków i wierzchołków.

Takie podejście pozwala na uwzględnienie alternatywnych (wielowariantowych) tras, optymalizowanych pod kątem różnych typów użytkowników o różnych preferencjach i kryteriach optymalności. Podstawą tego podejścia jest wymóg dostarczania indywidualnych rozwiązań (tras) dla klientów modułu. W zaproponowanym modelu grafowym dowolne dwa wierzchołki można połączyć więcej niż jedną krawędzią o tym samym zwrocie przypisując im różne wagi (etykiety).

Przykładowo multigraf jest budowany przez zdefiniowanie zbioru A jako uogólnionego zestawu łuków w postaci wielozbioru lub przez wprowadzenie wektora wag i etykiet na krawędziach.

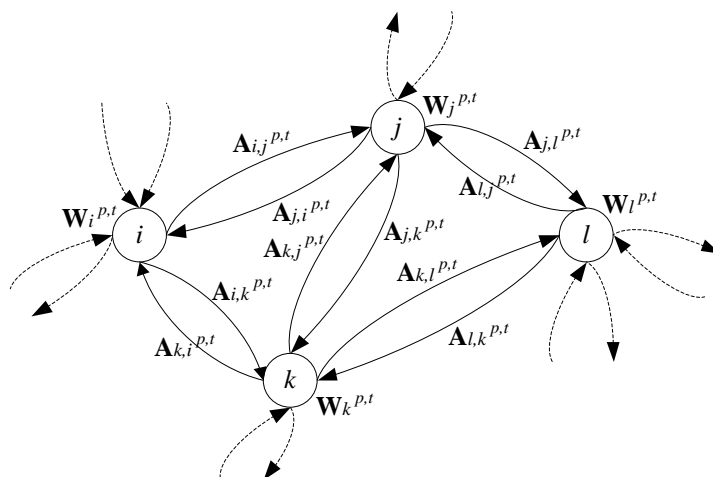
Niech:

$$G_{A,W} = \{G, A, W\} \quad (3.20)$$

definiuje sieć, gdzie $\{A_{i,j}^{p,t}\}$ jest rodziną macierzy, zawierająca opis krawędzi multigrafu G , a $\{W_i^{p,t}\}$ jest rodziną macierzy opisującą właściwości wierzchołków, gdzie:

- i, j są indeksami wierzchołków multigrafu,
- p jest etykietą krawędzi,
- t jest indeksem interwału czasu.

W rozważanym modelu macierz $A_{i,j}^{p,t}$ interpretowana jest jako parametry krawędzi (i, j) , określone dla etykiety (multigraf z etykietowanymi łukami) p w przedziale czasowym t . W



Rysunek 3.40: Model sieci transportowej w RPS.

najprostszym przypadku wartość tę można interpretować jako czas przebycia krawędzi (i, j) przez użytkownika drogi typu p . Parametr ten jest zmienny w czasie (dyskretny) i definiowany dla określonego horyzontu czasowego $t = t_0, \dots, t_{MAX}$ oraz określany na nowo okresowo (np. co 10 min.). Zawiera on zarówno historyczne, bieżące, jak i przewidywane wartości danego parametru. Analogicznie możemy interpretować macierz $\mathbf{W}_i^{p,t}$, jako zbiór parametrów opisujących czas przejścia pomiędzy łukami multigrafu. Tablice te są przypisane do wierzchołków i , etykiety kategorii p i czasu t .

Innymi słowy, dla każdego wierzchołka grafu opisującego sieć transportową definiujemy macierz opisującą jego właściwości z perspektywy sieci.

Macierz $\mathbf{W}_i^{p,t}$ jest w postaci:

$$\mathbf{W}_i^{p,t} = \begin{bmatrix} w_{in_1,i,out_1}^{p,t} & w_{in_1,i,out_2}^{p,t} & \cdots & w_{in_1,i,out_m}^{p,t} \\ w_{in_2,i,out_1}^{p,t} & \ddots & \cdots & w_{in_2,i,out_m}^{p,t} \\ \vdots & \vdots & \ddots & \vdots \\ w_{in_n,i,out_1}^{p,t} & w_{in_n,i,out_2}^{p,t} & \cdots & w_{in_n,i,out_m}^{p,t} \end{bmatrix}, \quad (3.21)$$

gdzie:

- n to liczba łuków łączących się z wierzchołkiem i i rozpoczynających się w wierzchołkach in_1, in_2, \dots, in_n ,
- m jest liczbą *wychodzących* krawędzi z wierzchołka i , kończących się w wierzchołku $out_1, out_2, \dots, out_m$.

Elementy macierzy $w_{i,j,k}^{p,t} \in \mathbf{R}$ są ciągłymi (lub dyskretnymi) funkcjami czasu. Jeśli są interpretowane jako czasy przejścia to i, j, k definiuje czas przejścia z wierzchołka i przez wierzchołek j do wierzchołka k .

Macierz $\mathbf{W}_i^{p,t}$ opisuje wszystkie możliwe przejścia przez wierzchołek (np. odpowiadające różnym manewrom na skrzyżowaniu, takim jak jazda na wprost lub skręt w lewo lub prawo) z każdego możliwego kierunku. Parametry zawarte w wektorze $\mathbf{A}_{i,j}^{p,t}$ mogą opisywać właściwości dróg, takie jak:

- czas przejazdu,
- gęstość ruchu,
- rozkład typów pojazdów,

- ruch pieszy,
- długość kolejki pojazdów,
- geometria drogi,
- prawdopodobieństwo przepełnienia drogi.

Klasyczny problem najkrótszej ścieżki należy do klasy złożoności P , jednak dodatkowe wymagania i ograniczenia w rozpatrywanym modelu sieci transportowej (np. zmienny czas przejazdu) są przyczyną, dla której problem ten należy do wyższej klasy złożoności.

Określenie globalne optymalnego rozwiązania przy użyciu dokładnych metod, ze względu na wymagany czas obliczeń, jest możliwe tylko w przypadku małych rozmiarów sieci transportowej. Aby rozwiązać większe problemy, należy zastosować skuteczne metody przybliżone.

Ogólny schemat dwuetapowego wielokryterialnego algorytmu do określania sub-optymalnej ścieżki w sieci transportowej zaprezentowano w [9].

W pierwszej fazie obliczeń jako metoda początkowa stosowana jest kolekcja algorytmów zachłannych. Znalezione rozwiązania tworzą zestaw rozwiązań początkowych dla następnej fazy optymalizacji. Różne kryteria stosowane w algorytmach konstrukcyjnych zapewniają dywersyfikację początkowych rozwiązań.

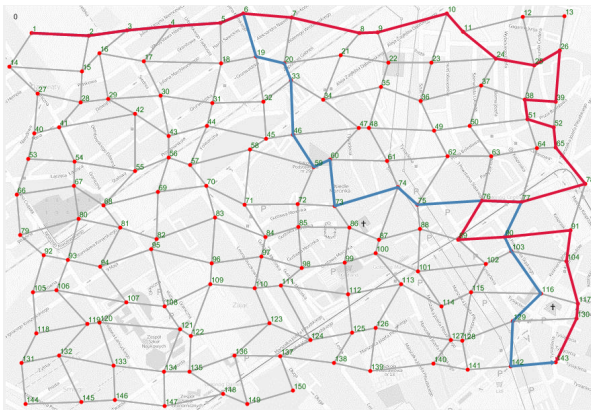
Ogólny przebieg algorytmu przedstawiono poniżej:

1. Określ początkowe i końcowe wierzchołki ścieżki oraz wektor kryteriów (wagi) powiązany z preferencjami użytkownika.
2. Zbuduj rozwiązanie - ścieżkę w grafie. Rozpocznij od początkowego wierzchołka i kontynuuj, aż do osiągnięcia końcowego wierzchołka.
3. W każdej iteracji określ nowy element rozwiązania, dodając jeden z wierzchołków osiągalnych od końca bieżącej ścieżki. Nowy wierzchołek jest określany przy użyciu jednego lub kilku kryteriów wybranych przez użytkownika, przykładowo:
 - *Kryterium 1*: najmniejsze odchylenie od azymutu do wierzchołka docelowego.
 - *Kryterium 2*: minimalizowanie odległości od celu.
 - *Kryterium 3*: odpowiednia szerokość drogi (lub inne ograniczenia określone dla krawędzi).
 - ...
 - *Kryterium n*: maksymalna pojemność drogi (lub klasy drogi).
4. Wierzchołek spełniający kryterium otrzymuje wagę zgodną z wektorem wag kryteriów. Wybierany jest wierzchołek o największej sumarycznej wadze. Możliwe jest też losowanie wierzchołka z prawdopodobieństwem proporcjonalnym do wagi.
5. W przypadku powtórzeń wierzchołków w ścieżce (cykl) następuje skrócenie rozwiązania i usunięcie wierzchołka ze zbioru wierzchołków sąsiednich. Pozwala to wykryć niespójność grafu (brak przejezdności).
6. W tworzeniu rozwiązań początkowych wykorzystuje się predefiniowane ścieżki bazowe grafu, które są w części lub całości użyte w konstrukcji poszukiwanej trasy. Podobną rolę pełnią wcześniej wyznaczone i zapisane w bazie rozwiązania.

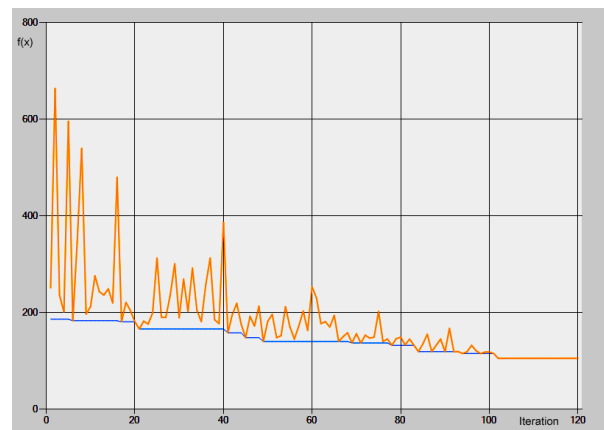
W drugiej fazie algorytmu stosuje się metody przybliżone, aby poprawić początkowe rozwiązania. Operując na zbiorze rozwiązań zachowuje się możliwość znalezienia kilku alternatywnych dróg, co jest ważne z punktu widzenia kierowcy oraz całego systemu sterowania ruchem, umożliwia bowiem wyrównanie obciążenia pojazdami dróg. W tym celu zastosowano

algorytm ewolucyjny klasy *Steady-State* oraz algorytm rojowy (PSO). W konstrukcji operatorów pseudogenetycznych krzyżowania i mutacji wykorzystano metody zachłanne poprawiając ich efektywność. W algorytmie cząsteczkowym dostosowano sposób tworzenia nowych rozwiązań do specyfiki zagadnienia komiwojżera. Ukierunkowanie cząsteczek w przestrzeni rozwiązań następuje w oparciu o pamięć częstotliwościową, wpływającą na prawdopodobieństwo pojawienia się pewnych elementów trasy w rozwiązaniu.

Algorytm operuje na danych aktualnych, które powstają przez modyfikowanie danych historycznych (zakładając istnienie pewnej cykliczności). Dzięki sensorom (kamery, czujniki) można uwzględnić zmiany parametrów dynamicznego modelu sieci transportowej. Uwzględniając sytuacje wyjątkowe (wypadek, roboty drogowe) i szybkość zmian liczby pojazdów na drodze, możliwe jest prognozowanie ruchu. Dane związane z profilem odbiorcy to: rodzaj pojazdu (motocykl, samochód osobowy, ciężarowy, autobus, pojazd przewożący substancje niebezpieczne, pojazd uprzywilejowany lub specjalny), jego gabaryty, masa, itp. oraz kryterium wyszukiwania trasy (najszybsza, najkrótsza, specjalna, terenowa, przełajowa, piesza) determinują (ograniczają) wybór etykietowanych łuków multigrafu. Profil określa także wektor wag kryteriów stosowany w algorytmach zachłannych, ograniczenia i funkcję oceny rozwiązania.



Rysunek 3.41: Przykład rozwiązania początkowego i najlepszego rozwiązania końcowego problemu najkrótszej ścieżki uzyskanej przez algorytm PSO.

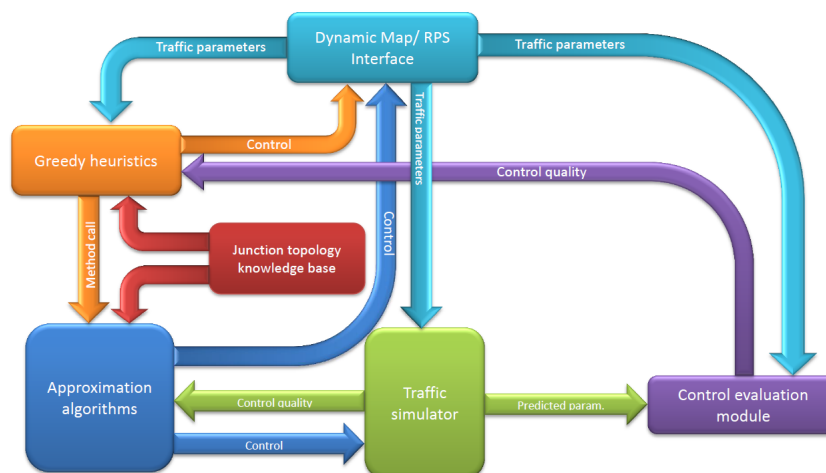


Rysunek 3.42: Przykład przebiegu algorytmu PSO (najlepsza cząstka w roju).

Zagadnienie optymalizacji sterowania ruchem drogowym z zastosowaniem sygnalizacji świetlnej

Jako wynik badań nad zagadnieniem sterowania ruchem drogowym zgłoszono patent dotyczący optymalizacji ruchu na skrzyżowaniach, stosujący zaproponowany model matematyczny sieci drogowej i korzystający prócz algorytmów, także z systemu regułowego. Opracowane rozwiązanie umożliwia dynamiczne sterowanie ruchem na jednym lub wielu sprzężonych skrzyżowaniach. W [6] zaprezentowano szczegóły systemu. Kontrola ruchu za pomocą sygnalizacji świetlnej wymaga uwzględnienia kilku czynników. Obejmują one:

- liczba skrzyżowań w kontrolowanym obszarze,
- liczba i typ strumieni pojazdów,
- geometria skrzyżowania,
- właściwości sterowników,
- właściwości pojazdu (przyspieszenie, hamowanie),
- zasady bezpieczeństwa,



Rysunek 3.43: System sterowania sygnalizacją świetlną.

- aspekty prawne.

Aby możliwe było spełnienie wszystkich wymienionych powyżej wymagań, opracowano i przetestowano zestaw algorytmów aproksymacyjnych i konstrukcyjnych do sterowania strumieniem pojazdów z zastosowaniem sygnalizacji świetlnej. Zbadano kilka metod sterowania sygnalizacją świetlną, w tym: cykliczne, acykliczne i wyzwalane, a także metodę stosującą jako kryterium sumaryczną długość kolejek zatrzymanych pojazdów. Ponadto w opracowanych algorytmach rozważono kilka scenariuszy koordynacji sąsiednich węzłów:

- koordynacja izolowanego skrzyżowania,
- koordynacja skrzyżowań połączonych,
- koordynacja skrzyżowań na jednej drodze lub w sieci,
- koordynacja scentralizowana.

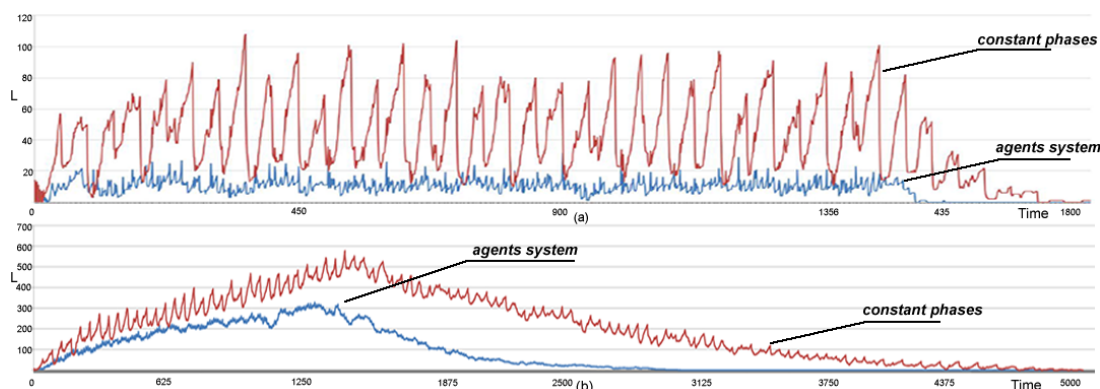
Opracowane algorytmy optymalizacyjne mogą wykorzystywać jeden z wymienionych powyżej scenariuszy, a przeprowadzone analizy i testy zaowocowały stworzeniem systemu z możliwością dostosowania do zmiennych warunków ruchu. System reaguje na bieżącą sytuację na drodze poprzez podejmowanie decyzji, np. o zmianie czasu trwania fazy sygnalizacji świetlnej lub też poprzez zmianę algorytmu optymalizacji na inny. Decyzję o ponownym obliczeniu parametrów lub zmianie algorytmu podejmuje się, gdy obniża się jakość sterowania (różnica pomiędzy przewidywanym a obserwowanymi parametrami ruchu). W przypadku znacznych zmian parametrów ruchu algorytm optymalizacyjny może zostać zmieniony. Taki schemat działania systemu jest możliwy dzięki stałemu dostępowi do aktualnych informacji o stanie ruchu drogowego oraz zastosowaniu wydajnego symulatora ruchu drogowego. Architektura systemu sterowania sygnalizacją świetlną przedstawiona jest na Rys. 3.43.

Wdrożone rozwiązanie ma cechy algorytmów hierarchicznych, w których górna warstwa (heurystyka) odpowiada za kontrolę ruchu i wywołuje metody z niższej warstwy (algorytmy przybliżone). Zaproponowana architektura umożliwi pracę systemu i jego dopasowanie się do warunków, w których występuje duża zmienność ruchu drogowego.

Poniżej przedstawiono główne moduły systemu sterowania sygnalizacją świetlną:

- *Mapa Dynamiczna/interfejs do systemu RPS* - dostarcza informacji o bieżącym stanie ruchu i umożliwia przekazanie wyznaczonych parametrów sterowania fazami świateł.
- *Heurystyki zachłanne* - zbiór heurystyk zachłannych o niskiej złożoności, która umożliwi dynamiczną i lokalną kontrolę sygnalizacji świetlnej. Są one stosowane w przypadku niskiego natężenia ruchu.

- *Algorytmy przybliżone* - zestaw algorytmów zaprojektowanych i opracowanych w projekcie, w tym:
 - *Algorytm ewolucyjny (EA)* - algorytm optymalizacji przetwarzający jednocześnie całą populację rozwiązań;
 - *Algorytm symulowanego wyżarzania (SA)* - szybki algorytm, przetwarzający w danej chwili tylko jedno rozwiązanie,
 - *System agentowy* - metoda używana do kontroli sygnalizacji świetlnej, wykorzystująca mechanizm samoadaptacji bazujący na algorytmie rojowym (PSO). Celem agentów jest znalezienie schematu synchronizacji faz świetlnych, który zapewnia maksymalną przepustowość.
- *Symulator ruchu* - umożliwia ocenę rozwiązania znalezionego przez algorytmy aproksymacyjne. Wykorzystuje szczegółowy opis sieci dróg (uzyskany z *Mapy Statycznej*) oraz informacje o aktualnym stanie ruchu (uzyskane z *Mapy Dynamicznej*).
- *Moduł ewaluacji sterowania* - moduł ten odpowiedzialny jest za ocenę jakości sterowania ruchem. Obliczona ocena jest dostarczana dla heurystyk odpowiedzialnych za wybór algorytmów sterowania.
- *Baza wiedzy topologii skrzyżowań świetlnych* - zawiera informacje o topologii skrzyżowań, wymagane do wyznaczenia ograniczeń nałożonych na długość faz sygnalizacji świetlnej.



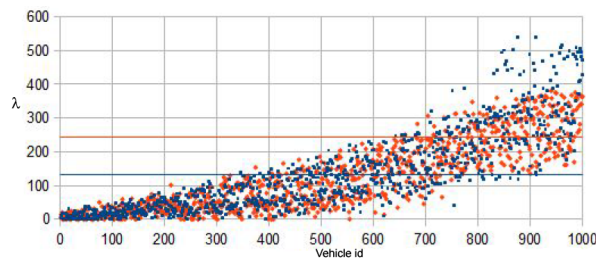
Rysunek 3.44: Sumaryczna długość kolejek pojazdów oczekujących na przejazd przez skrzyżowanie (L) w kolejnych przedziałach czasu dla stałych faz (czerwona linia) i systemu agentowego (niebieska linia); (a) - średni ruch, (b) - intensywny ruch.

Rysunki 3.44 i 3.45 pokazują różnicę między wydajnością sterowania stałofazowego i bardziej wyrafinowanych algorytmów sterowania fazami sygnalizacji świetlnej (system agentowy i algorytm symulowanego wyżarzania - SA) używanych do sterowania 30 skrzyżowaniami.

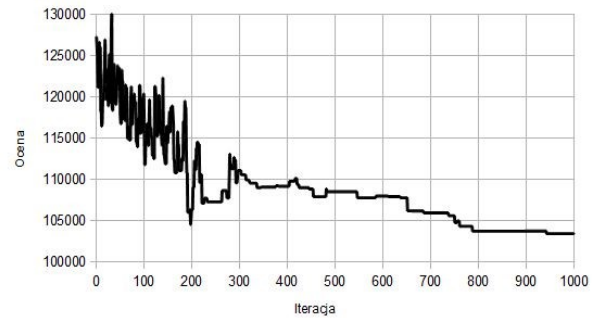
Rys. 3.46 przedstawia wartość sumarycznego czasu postoju pojazdów w kolejnych iteracjach algorytmu PSO. Ocena poszczególnych rozwiązań (zastosowanego schematu sterowania sygnalizacją świetlną) jest realizowana przez wydajny symulator ruchu drogowego.

3.2.7. Opracowanie specjalizowanego algorytmu ewolucyjnego wykorzystywanego w predykcji parametrów ruchu drogowego z zastosowaniem formalizmu rozmytych map kognitywnych

W ramach prac Autor zaproponował zastosowanie metod maszynowego uczenia, a w szczególności formalizmu rozmytych sieci kognitywnych, do predykcji parametrów opisujących ce-



Rysunek 3.45: Czas postoju na światłach dla 1000 pojazdów (λ) dla stałych faz sygnalizacji świetlnej (niebieskie punkty) i faz obliczonych przez algorytm SA (pomarańczowe punkty).



Rysunek 3.46: Proces optymalizacji sumarycznego czasu postoju pojazdów w sieci drogowej zawierającej 30 skrzyżowań (algorytm PSO).

chy sieci drogowych z zastosowaniem specjalizowanego algorytmu ewolucyjnego. Zaproponowana metoda pozwala na efektywne oszacowanie parametrów ruchu dla całej sieci, stosując rzadkie pomiary pochodzące z niewielkiej liczby detektorów zainstalowanych w dużej miejskiej sieci drogowej ([18]). Opracowane rozwiązanie przetestowano dla sieci drogowej miasta Krakowa, wykorzystując rzeczywiste dane z pomiarów ruchu drogowego.

Przeprowadzone badania wykazały dużą przydatność zastosowanego podejścia do predykcji parametrów opisujących ruch pojazdów na podstawie danych pozyskanych z niewielkiej ilości detektorów. Opracowany algorytm został zastosowany jako element większego systemu odpowiedzialnego za zarządzanie ruchem drogowym w ramach projektu INSIGMA.

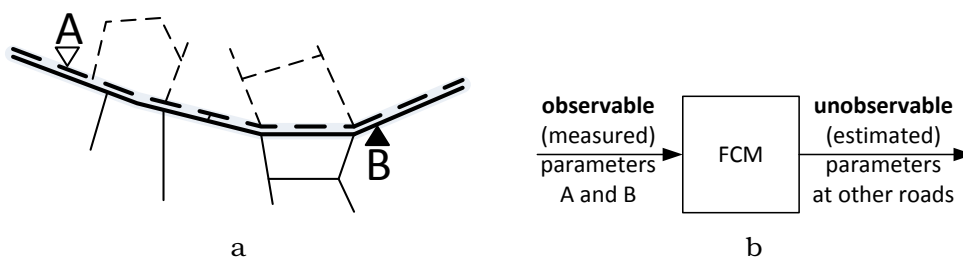
Motywacją przeprowadzonych badań naukowych było opracowanie metody pozwalającej na sprawną ocenę i predykcję parametrów ruchu drogowego w obrębie dużej miejskiej sieci drogowej, w oparciu o rzadkie pomiary obejmujące niewielką część segmentów sieci. Rozwiązanie zostało wdrożone w ramach systemu INSIGMA, umożliwiającego planowanie trasy i sterowanie ruchem drogowym. W tym celu zaproponowano model szacowania parametrów bazujący na formalizmie rozmytych map kognitywnych (*Fuzzy Cognitive Maps* - FCM), w którym do określenia parametrów modelu, zastosowano algorytm ewolucyjny (EA).

Zaproponowany algorytm ewolucyjny FCMEVOL wykazał wysoką efektywność podczas estymacji parametrów modelu opartego na FCM. W przeprowadzonych eksperymentach zastosowano symulator SUMO do generowania danych uczących dla sieci transportowych o różnych rozmiarach, a także do realizacji testów dla różnych scenariuszy ruchu drogowego.

Ze względu na złożoność problemu i względną prostotę FCM uzyskane wyniki są kompromisem pomiędzy dokładnością a wydajnością, ponieważ system pracuje w reżimie bliskim systemom czasu rzeczywistego. W tym świetle, FCM oferuje duże korzyści, gdyż po fazie nauki oszacowanie parametrów opisujących ruch może być wykonane bardzo szybko i z akceptowanym poziomem dokładności.

Zaproponowaną koncepcję przedstawiono na Rys. 3.47, który przedstawia fragment mapy złożonej z głównej drogi z wydzielonymi pasami ruchu i licznymi dochodzącymi do niej drogami. Zakładając, że dwa czujniki są zainstalowane w miejscach oznaczonych jako A i B, spodziewamy się, że istnieją zależności między pomiarami w punkcie A i częścią sieci drogowej oznaczoną linią przerywaną, a także pomiarami w punkcie B i na odcinkach dróg oznaczonych linią ciągłą. W związku z tym, abstrahując od ich dokładnej semantyki, parametry ruchu przypisywane sieci drogowej można podzielić na dwa zestawy: obserwowalne

(mierzone) i nieobserwowalne. W przypadku zastosowania FCM w dziedzinie dynamicznego planowania trasy, nieznane parametry można oszacować na podstawie znanych parametrów, stosując model wychwytyjący zależności przyczynowo-skutkowe pomiędzy interesującymi nas informacjami (Rys. 3.47).



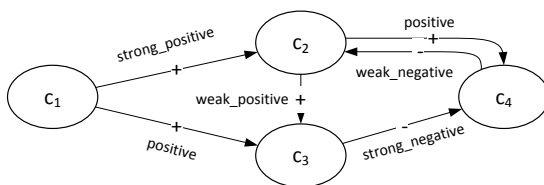
Rysunek 3.47: Fragment mapy z czujnikami A i B oraz koncepcja systemu szacowania parametrów ruchu.

FCM są skierowanymi grafami, których wierzchołki reprezentują pojęcia, podczas gdy krawędzie są używane do wyrażenia związków przyczynowych między nimi. Zestaw pojęć $C = \{c_1, \dots, c_n\}$ pojawiających się w modelu obejmuje zdarzenia, warunki lub inne istotne czynniki. Stan systemu jest wektorem zawierającym poziom aktywacji koncepcji ($n = |C|$), będącymi wartościami rzeczywistymi zdefiniowanymi na przedziałach $[0, 1]$ lub $[-1, 1]$.

Relacje między pojęciami (konceptami) w FCM mogą być reprezentowane przez wartości lingwistyczne, np. \therefore *strong_negative*, *negative*, *medium_negative*, *neutral*, *medium_positive*, *positive*, *strong_positive*, które w modelu obliczeniowym są mapowane na wartości równomiernie rozłożone w przedziale $[-1, 1]$.

Rys. 3.48 przedstawia przykład grafu FCM, którego wierzchołki są powiązane z pojęciami c_1 , c_2 , c_3 i c_4 , podczas gdy krawędzie opisujące wzajemne zależności pomiędzy pojęciami zostały opisane za pomocą zależności lingwistycznych. Odpowiednia macierz E jest zdefiniowana przez (3.22). Wybór wartości odpowiadających wartościom lingwistycznym jest arbitralny; w przykładzie zastosowano wartości: -1 , -0.66 , -0.33 , 0 , 0.33 , 0.66 i 1 .

$$E = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -0.33 \\ 0.66 & 0.33 & 0 & 0 \\ 0 & 0.66 & -1 & 0 \end{bmatrix} \quad (3.22)$$



Rysunek 3.48: Przykład grafu FCM. Wierzchołki są opisane pojęciami, a skierowane łuki z wagami lingwistycznymi określają wzajemny wpływ pomiędzy pojęciami.

Wnioskowanie za pomocą FCM polega na budowaniu sekwencji kolejnych stanów:

$$\alpha = A(0), A(1), \dots, A(k), \dots$$

począwszy od początkowego wektora poziomów aktywacji pojęć, kolejne elementy są wyliczane na podstawie formuły (3.23). W iteracji $k + 1$ wektor $A(k)$ jest mnożony przez macierz

wpływu E , a następnie wynikowe poziomy aktywacji są odwzorowywane na zakładany zakres za pomocą funkcji *aktywacji*:

$$A_i(k+1) = S_i\left(\sum_{j=1}^n e_{ij} A_j(k)\right). \quad (3.23)$$

Wybór funkcji aktywacji zależy od założeń dotyczących modelu obliczeniowego, a w szczególności od wyboru zakresu i typu wartości (dyskretne lub ciągłe). Często stosowaną funkcją aktywacji jest funkcja sigmoidalna:

$$S(x) = \frac{1}{1 + \exp(-\beta x)}.$$

Zasadniczo sekwencja kolejnych stanów $\alpha = A(0), A(1), \dots, A(k), \dots$ jest nieskończona. Można jednak pokazać, że po k iteracjach, gdzie k jest liczbą zbliżoną do rzędu macierzy E , osiąga się stan ustalony lub następuje cykl. Dlatego kryterium stopu dla algorytmu wnioskowania w kroku k jest najczęściej zdefiniowane następująco:

$$\exists j < k: d(A(k), A(j)) < \epsilon,$$

gdzie ϵ jest odpowiednio małą wartością.

Typowym celem implementacji FCM jest uczenie. W większości przypadków zestaw pojęć C ustalany jest przez ekspertów, a celem procesu nauki jest upodobnienie odpowiedzi modelowanego systemu i FCM przez dobór wartości w macierzy E .

Niech A_1^S, \dots, A_m^S , gdzie $m \leq n$ oznaczają wartości wyjściowe konceptów (lub decyzje) określonych w procesie wnioskowania przeprowadzonym zgodnie z formułą (3.23). Definiowanie funkcji oceny zwykle opiera się na założeniu, że dla danego wektora wejściowego możliwe jest wyznaczenie ścisłych zakresów $[A_i^{min}, A_i^{max}]$, w którym powinny znajdować się otrzymane wartości A_i^S . Zakresy te mogą być określone przez ekspertów uczestniczących w procesie uczenia się lub mogą pochodzić z obserwacji.

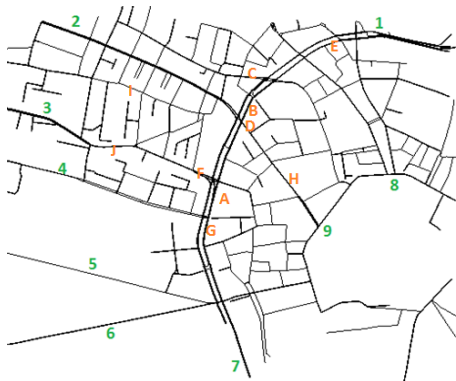
Przy takim założeniu, funkcja oceny wyraża karę za naruszenie wyznaczonych zakresów (3.24), gdzie symbol H oznacza funkcję Heaviside (3.25).

$$\begin{aligned} \varphi(E) = & \sum_{i=1}^m H(A_i^{min} - A_i^S) |A_i^{min} - A_i^S| + \\ & \sum_{i=1}^m H(A_i^S - A_i^{max}) |A_i^S - A_i^{max}| \end{aligned} \quad (3.24)$$

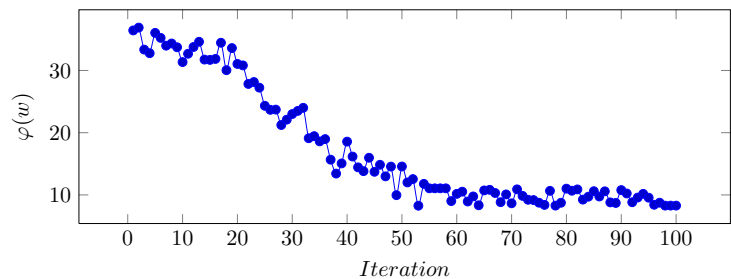
$$H(x) = \begin{cases} 0, & \text{if } x < 0 \\ 1, & \text{if } x \geq 0, \end{cases} \quad (3.25)$$

Do minimalizacji powyższej funkcji celu zaproponowano algorytm ewolucyjny (EA), z reprezentacją rozwiązań w postaci wektora liczb rzeczywistych. W celu przetwarzania populacji zaproponowano zastosowanie dwóch specjalizowanych operatorów krzyżowania *BlendAlphaCrossover* (BAC) i *BlendAlphaBetaCrossover* (BABC) z odpowiednio dobranymi parametrami α i β , operator mutacji (RM) oraz operator optymalizacji lokalnej (LO). W celu zmniejszenia zbieżności algorytmu, ze względu na rodzaj reprezentacji rozwiązania oraz w celu zabezpieczenia się przed wpadnięciem algorytmu w zamknięty cykl obliczeniowy, zastosowano mechanizm zabronień oparty na *liście tabu* - rozwiązaniu analogicznym do stosowanego w algorytmie przeszukiwania z zabronieniami (*Tabu Search*). Ponadto, w celu poprawy eksploracji przestrzeni rozwiązań, przebieg algorytmu podzielono na dwie fazy

Breadth i *Depth*, różniące się metodą wyboru kandydatów na rodziców (np. wykluczeniem najlepszych rozwiązań) oraz rodzajem stosowanych operatorów (np. zastąpieniem operatora RM operatorem LO). Testy przeprowadzono m.in. stosując dane pochodzące z miasta Krakowa,



Rysunek 3.49: Fragment dużej sieci drogowej Krakowa z zaznaczonymi punktami pomiarów.



Rysunek 3.50: Proces nauki parametrów FCM z zastosowaniem algorytmu ewolucyjnego *Breath-Depth*.

gdzie graf sieci drogowej składał się 682 wierzchołków i 464 442 krawędzi z 1500 detektorami obserwującymi aktualny stan ruchu wykazały, że zaproponowane rozwiązanie uzyskuje dobre wyniki predykcji parametrów ruchu drogowego.

3.2.8. Opracowanie specjalizowanego algorytmu przybliżonego bazującego na algorytmie Tabu Search do oceny scenariuszy rozwoju technologii (*roadmapping*)

Planowanie zadań strategicznych obejmuje tzw. *roadmapping*, określający sposób realizacji procesów typu NPD (*New Product Development*) opisywanych z zastosowaniem modeli wielokryterialnych. Adaptacja jednostki danej organizacji do realizacji przydzielonych zadań jest ważnym aspektem planowania strategicznego [12]. Jeżeli sposób ich realizacji zależy od wdrożonych procedur (np. ustawodawstwa) to w znacznym stopniu ogranicza to swobodę wyboru dopuszczalnych rozwiązań opisujących realizację założonego planu. W przypadku złożonych jednostek tworzone są wyspecjalizowane działy, które są odpowiedzialne za jednorodne podzadania, często określane mianem operacji lub czynności. Rozważając kolejność realizacji zadań w tak zdefiniowanej strukturze, możemy zaadaptować różnego typu modele zdefiniowane w teorii harmonogramowania zadań. Zapewniają one szeroki zakres możliwości wyboru funkcji kryterialnej, typów ograniczeń zasobów, priorytetów, definiowania relacji kolejności pomiędzy operacjami, terminów realizacji oraz oceny wydajności działów/zespołów. Ze względu na ograniczenia, realizowane zadania mogą być przerywane lub nie, a ich przetwarzanie może się odbywać szeregowo lub równoległe. W rozważanym przypadku przedsiębiorstwa realizują wybrany plan strategicznego rozwoju, na który składa się zdefiniowany zestaw celów osiągniętych poprzez realizację procesów strategicznych. Odziały przedsiębiorstw, po przydzieleniu określonych zasobów, specjalizują się w wykonywaniu przydzielonych im podzadań.

Realizacja planu strategicznego zakłada, że:

- podany jest czas przetwarzania podzadań w działach firmy,
- podzadania nie mogą być przerywane i są wykonywane tylko jeden raz,
- w tym samym czasie, każdy dział firmy może przetwarzać tylko jedno podzadanie,
- podzadanie jest wykonywane natychmiast, gdy dział jest wolny, w przeciwnym razie podzadanie oczekuje na zakończenie poprzedniego podzadań,

- plan strategiczny kończy się po wykonaniu ostatniego podzadania,
- określa się kolejność, w której podzadania mają być przetwarzane (jest taka sama dla każdego planu strategicznego).

Budowa planu strategicznego w dynamicznym i wysoce konkurencyjnym otoczeniu wymaga sprawnego, elastycznego i dynamicznego procesu decyzyjnego. Planowanie obejmuje opracowanie zasad podejmowania decyzji, które optymalizują kryteria dotyczące procesu wdrażania i produkcji. W zależności od sposobu sformułowania problemu decyzyjnego, zasady te mogą przyjąć formę planu strategicznego, listy priorytetów lub biznesplanu związanego z określoną technologią lub produktem.

Diagram *roadmapy* jest strukturą warstw odpowiadającą zewnętrznym obszarom działalności lub wpływom (rynk, technologie, prawodawstwo, produkty i zadania publiczne). Schemat tworzenia *roadmapy* rozwoju produktu obejmuje wybór obiektów dla poszczególnych warstw w oparciu o istotne kryteria, z warstwy (lub warstw) opisującej rozważaną organizację. Relacje między obiektami są definiowane i kodowane podczas wyboru obiektów. Relacje między warstwami są zwykle relacjami czasowymi. Opisują ewolucję obiektów każdej warstwy, wykorzystując informacje o trendach rozwojowych i możliwych scenariuszach.

Stworzenie strategicznego planu pozyskiwania i wdrażania technologii dla procesów produkcyjnych jest przykładem, kiedy organizacja powinna stosować *roadmapping*. W przypadku rozważania alternatywnych produktów i dopuszczalnych technologii lub konkretnych scenariuszy środowiskowych, optymalizowana jest decyzja dotycząca wydatków na wdrażanie technologii. Prowadzi to do rozwiązania dyskretnych problemów optymalizacyjnych, jakie zdefiniowano poniżej, gdzie parametry opisują cechy technologii i produktów (cena zakupu licencji, jednostkowe koszty wdrożenia, czas życia technologii, zakres produktów i możliwość odsprzedaży własnej technologii opisanej przez parametry rzeczywistych możliwości). Ponadto, *roadmapping* pozwala uwzględnić zależność technologii i produktów od warunków prawnych (w formie ograniczeń), w szczególności związanych z ochroną środowiska. Poniższy problem optymalizacji, szczególnie przypadek przepływowego problemu harmonogramowania (*flow-shop scheduling problem* - FSP), można sformułować na podstawie analizy schematu planowania.

Niech zbiór $\bar{Z} = \{Z_1, Z_2, Z_3, \dots, Z_r\}$ będzie zbiorem planów strategicznych r , gdzie każdy plan Z_i jest podzbiorem zbioru \bar{J} (zadania) $Z_i = \{J_i\}_{J_i \in \bar{J}}$:

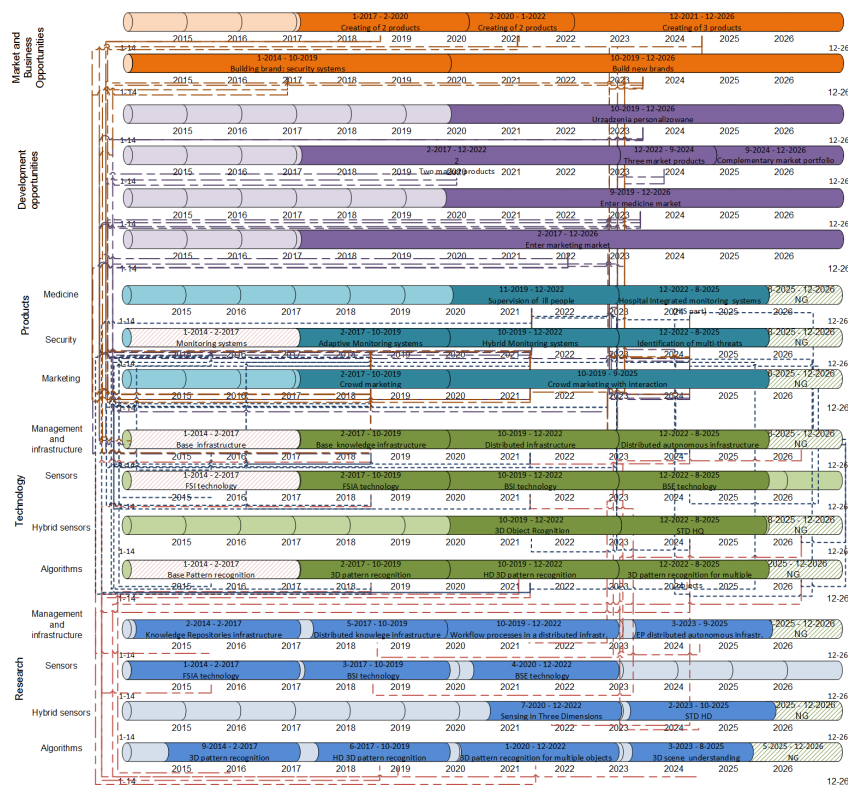
- $\bar{J} = \{J_1, J_2, \dots, J_k\}$ to zestaw zadań zdefiniowanych przez eksperta. Dla każdego zadania określona jest funkcja zysku. Wartość zysku zależy od kolejności wykonania zlecenia w planie strategicznym.
- Plan strategiczny Z_i to podzbiór zleceń zdefiniowanych przez eksperta na podstawie polityki firmy. W przypadku uogólnionym można rozważyć wszystkie możliwe podzbiory zadań z \bar{J} .
- Każde zadanie $J_j = \{o_{j1}, o_{j2}, \dots, o_{jm}\}$ zawiera m operacji odpowiadających wykonaniu zadania na zbiorze m maszyn (oddziały firmy). Zadania są wykonywane we wcześniej zdefiniowanej kolejności (badania, technologie, produkcje itp.).
- Każda maszyna (filie firmy) może wykonywać maksymalnie jedną operację (badania, technologię itd.) na w danej chwili.
- Czas wykonania operacji $t_{jk} \geq 0$ zależy od kolejności wykonania zadań. Niektóre operacje z różnych zadań mogą nakładać się lub mogą być zależne.
- Dla każdego z planów strategicznych Z_i , gdzie $i = 1, \dots, r$, przeszukiwana jest optymalna sekwencja zadań, które minimalizują C_{max} (maksymalny *makespan*). Dodatkowo sekwencja zadań spełnia ograniczenie czasowe $C_{max} \leq \bar{T}$.
- Celem jest znalezienie planu strategicznego Z^* (i jego sekwencji zadań), który maksymalizuje zysk opisany przez równanie (3.26) i spełnia ograniczenia (predefiniowane zamówienie oraz limit czasu wykonania \bar{T}).

Określmy $S(Z)$ jako zbiór wszystkich możliwych permutacji zadań w danym strategicznym planie Z . Zdefiniujemy następujący problem maksymalizacji:

$$f(Z^*) = \max_{Z \in \bar{Z}} \max_{s \in S(Z)} \{ \alpha f_i(\bar{T} - C_{max}) + \gamma f_{pref}(s) - \beta f_p(s) \}, \quad (3.26)$$

gdzie:

- α, β, γ — skalary określające preferencje zdefiniowane przez eksperta,
- $f_i(\bar{T} - C_{max})$ — określa zysk ze skrócenia czasu realizacji planu strategicznego,
- $f_p(s)$ — funkcja kary za naruszenie ograniczeń (sekwencja Z musi spełniać określoną wcześniejszą kolejność realizacji zadań i czas wykonania sekwencji zadań spełnia ograniczenie czasowe $C_{max} \leq \bar{T}$),
- $f_{pref}(s)$ — pewna funkcja określająca preferencje wobec technologii i strategii, zdefiniowana przez ekspertów.



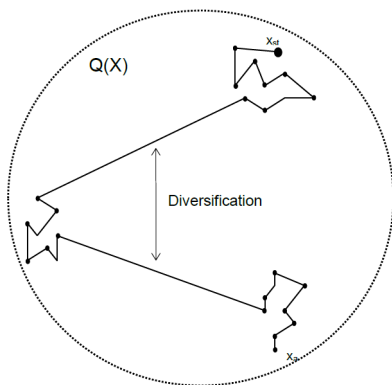
Rysunek 3.51: Przykład diagramu roadmappingu.

W celu poszukiwania rozwiązania problemu optymalizacyjnego w postaci (3.26) zaprojektowano zmodyfikowany algorytm przeszukiwania z zabronieniami (*Tabu Search* - TS). W zaproponowanym algorytmie prócz standardowych mechanizmów związanych z pamięcią krótko-, średnio- i długoterminową, wprowadzono nowe mechanizmy zwiększające jego wydajność podczas rozwiązywania zagadnienia *roadmappingu*. Modyfikacje te polegają na wprowadzeniu dodatkowego mechanizmu dywersyfikacji i intensyfikacji.

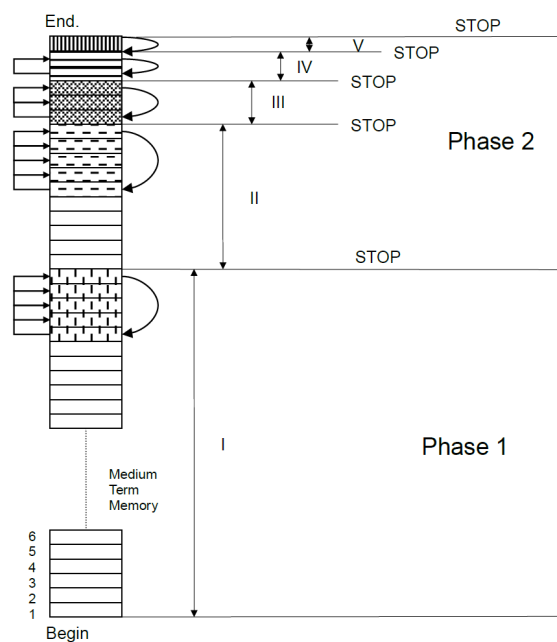
W dywersyfikacji (Rys. 3.52) preferowane są rozwiązania znajdujące się w takich obszarach przestrzeni rozwiązań, które były rzadko badane przez algorytm. Pamięć o częstotliwościowa (typ pamięci długoterminowej - LTM) to narzędzie, które rejestruje informacje statystyczne o wcześniejszych wynikach algorytmu (np. zliczanie ruchów o określonych atrybutach), a następnie restartuje procedurę poszukiwania tworząc rozwiązania „z dala”, w obszarach dotąd

nie odwiedzanych. Intensyfikacja polega na wykorzystaniu obiecujących obszarów przestrzeni rozwiązania i jest realizowana poprzez powrót do rozwiązań zapisanych w pamięci średnioterminowej (MTM) podczas pierwszej fazy działania algorytmu (Rys. 3.53).

Na podstawie historii wyszukiwania tworzona jest nowa lista tabu. Najlepsze znalezione rozwiązania, tj. te, których wartość funkcji celu mieści się w określonym zakresie, uzyskane z bieżących rozwiązań suboptymalnych, są umieszczane w MTM. Odległość kolejnych rozwiązań, mierzona liczbą iteracji, musi być większa niż z góry określona wartość (w przypadku rozwiązań bliskich wybierana jest lepsze rozwiązanie). Po zapamiętaniu rozwiązania w MTM, atrybuty kolejnych ruchów są zapisywane. Na ich podstawie tworzona jest nowa lista tabu, gdy następuje powrót do rozwiązania zapamiętanego w MTM (Rys. 3.54).

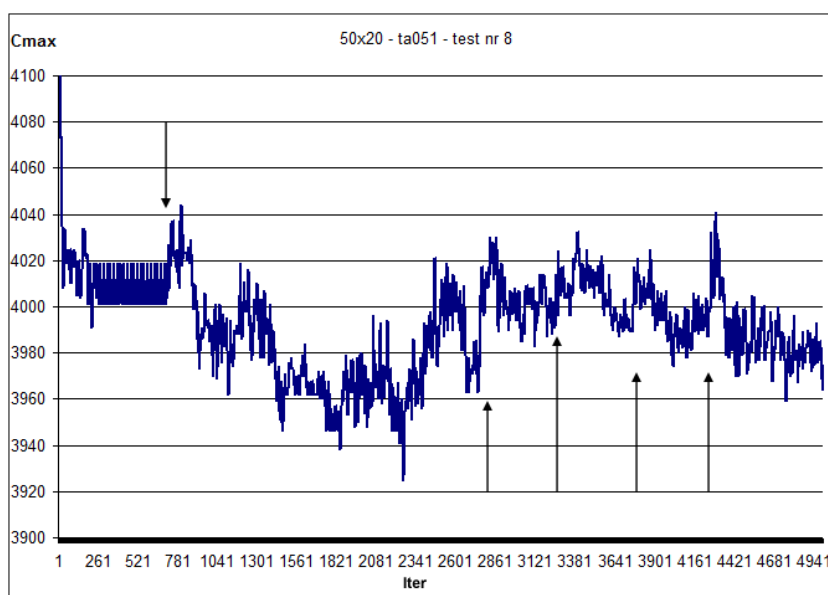


Rysunek 3.52: Mechanizm dywersyfikacji podczas eksploracji przestrzeni rozwiązań.



Rysunek 3.53: Mechanizm intensyfikacji podczas eksploracji przestrzeni rozwiązań - powrót do wcześniejszego rozwiązania.

Opracowany algorytm przeszukiwania z zabronieniami przetestowano na dużym zbiorze instancji testowych i porównano z wynikami uzyskanymi przez standardowy algorytm przeszukiwania z zabronieniami. Uzyskanie wyniki przedstawienie w [12] wskazują, że w każdym przypadku zmodyfikowany algorytm uzyskuje lepsze wyniki. Co więcej, w niektórych przypadkach udało się poprawić najlepsze dotąd znalezione rozwiązania dla instancji testowych rozpatrywanego zagadnienia, uzyskane przez inne algorytmy przybliżone.



Rysunek 3.54: Zależność funkcji celu bieżącego rozwiązania od numeru iteracji algorytmu TS^* dla zadania testowego z 51 zadaniami i 20 oddziałami. Strzałki oznaczają powrót do rozwiązania zapamiętanego w MTM.

4. Wskazanie dodatkowych osiągnięć naukowych

Jako dodatkowe osiągnięcia Autora, rozszerzające główny nurt badań związanych z zagadnieniami optymalizacji i zastosowaniem metod maszynowego uczenia, można wymienić:

1. Opracowanie metody umożliwiającej zastosowanie rozmytej mapy kognitywnej do oceny wzajemnego oddziaływania pomiędzy elementami systemu informatycznego. Zaproponowane podejście pozwoliło na ocenę wpływu ryzyka pochodzącego z obiektów niskopoziomowych (sprzętu tj. kamer, czujników, oraz oprogramowania) na ryzyko obiektów wysokopoziomowych jak usługi i procesy realizowane w informatycznym systemie nadzoru [29].
2. Zaproponowanie oraz przebadanie modelu systemu sterowania rojem obiektów bezzałogowych stosujący mechanizmy zapożyczone z algorytmu PSO (*Particle Swarm Optimisation*), stosowanego w optymalizacji przybliżonej. W rozważanym problemie zagadnienie sterowania rojem bezzałogowych obiektów latających (UAV) rozpatrywano jako zagadnienie decyzyjne i optymalizacyjne [25].
3. Opracowanie ontologii dla systemów zarządzania bezpieczeństwem i ruchem drogowym w ramach projektów INDECT, INSIGMA, SIMPOZ oraz INPREDO ([30],[6],[31], [28], [23], [26]).
4. Opracowanie systemu regułowego dla systemu wyznaczania bezpiecznych prędkości w ramach projektu INPREDO([7]).

4.1. Publikacje wchodzące w skład dodatkowych osiągnięć naukowych

1. (Rotter, **Chmiel et all, 2018a**) P. Rotter and W. Chmiel. A framework for a hierarchical model of cooperation between unmanned airplanes. *European Journal of Remote Sensing*, 51(1):276–284, 2018 (IF=1.122, punktacja MNiSW: 20).
2. [**Chmiel et all, 2018b**] W. Chmiel, A. Dziech, S. Jędrusik, P. Kadłuczka, J. Kwiecień, P. Szwed, Z. Mikrut, and G. Rogus. Rule system for speed limit determination on national roads in Poland. *MATEC Web Conf.*, 231:02001, 2018 (Scopus)
3. [Szwed, **Chmiel et all, 2016**] P. Szwed, P. Skrzyński, and W. Chmiel. Risk assessment for a video surveillance system based on Fuzzy Cognitive Maps. *Multimedia Tools and Applications*, 75(17):10667–10690, Sep 2016 (IF=1.361, punktacja MNiSW: 25).
4. [**Chmiel et all, 2014**] W. Chmiel, J. Dańda, A. Dziech, S. Ernst, A. Głowacz, P. Kadłuczka, Z. Mikrut, P. Pawlik, P. Szwed, and I. Wojnicki. Contribution of the INSIGMA Project to the Field of Intelligent Transportation Systems. In A. Dziech and A. Czyżewski, editors, *Multimedia Communications, Services and Security*, pages 58–72, Cham, 2014. Springer International Publishing (WoS)

5. [Tadeusiewicz, **Chmiel et all, 2014**] R. Tadeusiewicz, W. Chmiel, P. Kadłuczka, and Z. Mikrut. Sposób i system do zarządzania regułami w rozproszonym systemie nadzoru. Number PL 403161 A1, 09 2014 (Patent).
6. [Wojnicki, Szwed, **Chmiel et all, 2012**] I. Wojnicki, P. Szwed, W. Chmiel, and S. Ernst. Ontology Oriented Storage, Retrieval and Interpretation for a Dynamic Map System. In A. Dziech and A. Czyżewski, editors, *Multimedia Communications, Services and Security*, pages 380–391, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg (WoS).
7. [Szwed, Kadłuczka, **Chmiel et all, 2012**] P. Szwed, P. Kadłuczka, W. Chmiel, A. Głowacz, and J. Śliwa. Ontology based integration and decision support in the INSIGMA route planning subsystem. In *2012 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 141–148. IEEE Computer Society Press, Sept 2012 (WoS).
8. [Gleba, Śliwa, **Chmiel et all, 2011**] K. Gleba, J. Śliwa, W. Chmiel, P. Szwed, and A. Głowacz. Ontology engineering methodology for intelligent system for global monitoring, detection and identification of threats. *Military communications and information technology: a comprehensive approach enabler*, pages 133–144, 2011 (WoS).
9. [Śliwa, Gleba, **Chmiel et all, 2011**] J. Śliwa, K. Gleba, W. Chmiel, P. Szwed, and A. Głowacz. IOEM - Ontology Engineering Methodology for Large Systems. In P. Jędrzejowicz, N. T. Nguyen, and K. Hoang, editors, *Computational Collective Intelligence. Technologies and Applications*, pages 602–611, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg (MNiSW:10, WoS).

4.2. Syntetyczny opis osiągnięć dodatkowych

W rozdziale tym zawarto syntetyczny opis dodatkowych osiągnięć naukowo badawczych.

1. Zebrane doświadczenia związane z zastosowaniem metod przybliżonych inspirowanych naturą w optymalizacji \mathcal{NP} -trudnych zagadnień, zaowocowały ideą przeniesienia stosowanych tam mechanizmów do dziedzin wykraczających poza zakres optymalizacji. Zaproponowano oraz przebadano model systemu sterowania rojem samolotów bezzałogowych (UAV), stosującym mechanizmy zaczerpnięte z algorytmu rojowego (*Particle Swarm Optimisation* - PSO). Przeprowadzona analiza pozwoliła na zdefiniowanie zbioru scenariuszy uwzględniających zarówno różnorakie aspekty konstrukcji bezzałogowych obiektów latających, jak i zarządzania ich rojem. W opracowanych scenariuszach uwzględniono zarówno cel realizowany przez rój, jak również ograniczenia techniczne i ekonomiczne ([25]). Opracowane podejście oceniono na podstawie analizy reakcji na zdarzenia związane z awarią lub utratą samolotu dla kilku modeli sterowania.
2. Opracowanie szeregu modeli ontologicznych stosowanych do opisu przetwarzanej wiedzy oraz wnioskowania dla rozwiązań tworzonych w ramach polskich i międzynarodowych projektów badawczych:
 - Projekt INDECT - opracowanie ontologii dla systemów pozyskiwania danych i oceny bezpieczeństwa.
 - Projekt INPREDO - opracowanie ontologii w celu przetwarzania danych w systemach nadzoru.

- Projekt SIMPOZ - opracowanie ontologii do składowania danych w systemach przetwarzania strumieni wideo.
3. Opracowanie systemu regulowego do wyznaczania bezpiecznych prędkości w ramach projektu RID (Rozwój Innowacji Drogowych). Opracowany system zostanie zastosowany przez Generalną Dyрекcję Dróg Krajowych i Autostrad do określania bezpiecznych limitów prędkości oraz do składowania wszystkich danych pozyskiwanych z czujników umieszczonych na drogach (parametry prędkości, wypadki i zdarzenia drogowe).

5. Ogólny przegląd osiągnięć - statystyki

Tab. 5.1 przedstawia zestawienie statystyczne osiągnięć Autora w zakresie naukowo-badawczym, dydaktycznym oraz osiągnięć w kształceniu kadry. Zestawienie to zawiera publikacje wyróżnione przez *Journal Citation Reports* (Lista Filadelfijska), publikacje w recenzowanych czasopismach krajowych lub zagranicznych (w tym indeksowane w *Web of Science*) oraz rozdziały w monografiach. Zamieszczono również liczbę wykonanych recenzji, udział w projektach badawczych oraz opiekę nad pracami inżynierskimi i magisterskimi. Szerszy wykaz prac opublikowanych po uzyskaniu stopnia doktora znajduje się w kolejnych rozdziałach.

Tablica 5.1: Wykaz osiągnięć naukowo-badawczych, dydaktycznych oraz w kształceniu kadry

L.p	Rodzaj publikacji	Liczba prac
1	Publikacje w czasopismach wyróżnionych przez <i>Journal Citation Reports</i> (Lista Filadelfijska)	7
2	Publikacje w innym recenzowanym czasopiśmie krajowym lub zagranicznym	42
3	Autorstwo rozdziału w monografii lub podręczniku akademickim	5
4	Referaty na międzynarodowych i krajowych konferencjach naukowych	10
5	Publikacje indeksowane w <i>Web of Science</i>	12
6	Seminaria naukowe	8
7	Udział w projektach badawczych polskich i zagranicznych	9
8	Recenzje w czasopismach wyróżnionych przez <i>Journal Citation Reports</i> (Lista Filadelfijska)	12
9	Zgłoszenia patentowe i patenty	2
10	Publikacje punktowane MNiSW	30

5.1. Publikacje z *Impact Factor* oraz punktacja na podstawie list ministerialnych

W Tab. 5.2 przedstawiono wartości parametru *Impact Factor* (IF) dla poszczególnych publikacji. Natomiast w Tab. 5.3 zaprezentowano punktacje na podstawie list ministerialnych zgodnie z rokiem opublikowania wszystkich publikacji w czasopiśmie naukowych powstałych po uzyskaniu stopnia doktora (wydane w roku 2018 i 2019, wg roku 2017).

Tablica 5.2: *Impact Factor* oraz punktacja dla publikacji z listy MNiSW.

L.p	Publikacja	IF	Punktacja MNiSW
1	W. Chmiel. Evolutionary algorithm using conditional expectation value for quadratic assignment problem. <i>Swarm and Evolutionary Computation</i> , 46:1 – 27, 2019	3.818	50
2	W. Chmiel and J. Kwiecień. Quantum-Inspired Evolutionary Approach for the Quadratic Assignment Problem. <i>Entropy</i> , 20(10), 2018	2.35	30
3	W. Chmiel, I. Skalna, and S. Jędrusik. Intelligent route planning system based on interval computing. <i>Multimedia Tools and Applications</i> , Oct 2018	1.154	25
4	W. Chmiel, P. Kadłuczka, J. Kwiecień, and B. Filipowicz. A comparison of nature inspired algorithms for the quadratic assignment problem. <i>Bulletin of the Polish Academy of Sciences: Technical Sciences</i> , 65(4):513–522, 2017	1.361	25
5	W. Chmiel, J. Dańda, A. Dziech, S. Ernst, P. Kadłuczka, Z. Mikrut, P. Pawlik, P. Szwed, and I. Wojnicki. INSIGMA: an intelligent transportation system for urban mobility enhancement. <i>Multimedia Tools and Applications</i> , 75(17):10529–10560, Sep 2016	1.361	25
6	P. Rotter and W. Chmiel. A framework for a hierarchical model of cooperation between unmanned airplanes. <i>European Journal of Remote Sensing</i> , 51(1):276–284, 2018	1.122	20
7	P. Szwed, P. Skrzyński, and W. Chmiel. Risk assessment for a video surveillance system based on Fuzzy Cognitive Maps. <i>Multimedia Tools and Applications</i> , 75(17):10667–10690, Sep 2016	1.361	25
	<i>Suma</i>	12.625	200

Tablica 5.3: Punktacja pozostałych artykułów w czasopismach krajowych i zagranicznych

L.p	Publikacja	Punktacja MNiSW
1	W. Chmiel, A. Dziech, S. Jędrusik, P. Kadłuczka, J. Kwiecień, P. Szwed, Z. Mikrut, and G. Rogus. Rule system for speed limit determination on national roads in Poland. <i>MATEC Web Conf.</i> , 231:02001, 2018	5 (Scopus)
2	W. Chmiel, P. Kadłuczka, J. Kwiecień, B. Filipowicz, and P. Pukocz. Strategic Planning Optimization Using Tabu Search Algorithm. In A. M. Skulimowski and J. Kacprzyk, editors, <i>Knowledge, Information and Creativity Support Systems: Recent Trends, Advances and Solutions</i> , pages 315–328, Cham, 2016. Springer International Publishing	15 (WoS)
3	W. Chmiel and P. Szwed. Learning Fuzzy Cognitive Map for Traffic Prediction Using an Evolutionary Algorithm. In A. Dziech, M. Leszczuk, and R. Baran, editors, <i>Multimedia Communications, Services and Security</i> , volume 566 of <i>Communications in Computer and Information Science</i> , pages 195–209. Springer International Publishing, 2015	15(WoS)
4	Andrzej Dziech and Wojciech Chmiel and Sebastian Ernst and Piotr Kadłuczka and Zbigniew Mikrut and Piotr Szwed and Igor Wojnicki. System and the method to manage the rules of control of the traffic lights on the crossing, PL414870A1, BUP 11/2017, 11 2017	2 (Patent)
5	P. Szwed, W. Chmiel, and P. Kadłuczka. OpenCL Implementation of PSO Algorithm for the Quadratic Assignment Problem. In L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, and J. M. Zurada, editors, <i>Artificial Intelligence and Soft Computing</i> , volume 9120 of <i>Lecture Notes in Computer Science</i> , pages 223–234. Springer International Publishing, 2015	15 (WoS)
6	B. Filipowicz, W. Chmiel, P. Kadłuczka, and P. Wicijowski. Greedy algorithms for three dimensional packing problem. <i>Automatyzacja procesów dyskretnych : teoria i zastosowania</i> , 1:49–60, 2012	4
7	W. Chmiel, P. Kadłuczka, and S. Ernst. A Multicriteria Model for Dynamic Route Planning. In A. Dziech and A. Czyżewski, editors, <i>Multimedia Communications, Services and Security</i> , pages 174–182, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg	10 (WoS)
8	W. Chmiel, P. Kadłuczka, K.Wala, and S.Jędrusik. Algorytmy heurystyczne w trójwymiarowym zagadnieniu pakowania — heuristic algorithm for three-dimensional packing problem. <i>Automatyka/Automatics</i> , 14(3/2):827–840, 2010	6
9	P. Kadłuczka, W. Chmiel, and J. Piwowarczyk. Strategia przetwarzania wiedzy w wielopopulacyjnym algorytmie ewolucyjnym — knowledge processing strategy in the multi-population evolutionary algorithm. <i>Automatyka/Automatics</i> , 13(2):325–331, 2009	4
10	W. Chmiel, P. Kadłuczka, and G. Packanik. Performance of Swarm Algorithms for Permutation Problems . <i>Automatyka</i> , 15(2):117–126, 2009	4

11	W. Chmiel and P. Kadłuczka. Wielofazowa metoda różnicowania populacji w algorytmie ewolucyjnym — a multi-phase diversification method of population in the evolutionary algorithm. <i>Automatyka/Automatics</i> , 1797(151):195–202, 2008	4
12	W. Chmiel and P. Szwed. Bees Algorithm for the Quadratic Assignment Problem on CUDA Platform. In A. Gruca, A. Brachman, S. Kozielski, and T. Czachórski, editors, <i>Man–Machine Interactions 4</i> , volume 391 of <i>Advances in Intelligent Systems and Computing</i> , pages 615–625. Springer International Publishing, 2016	15 (WoS)
13	W. Chmiel, A. Dziech, S. Jędrusik, P. Kadłuczka, J. Kwiecień, P. Szwed, Z. Mikrut, and G. Rogus. Rule system for speed limit determination on national roads in Poland. <i>MATEC Web Conf.</i> , 231:02001, 2018	15 (Scopus&WoS)
14	W. Chmiel, J. Dańda, A. Dziech, S. Ernst, A. Głowacz, P. Kadłuczka, Z. Mikrut, P. Pawlik, P. Szwed, and I. Wojnicki. Contribution of the INSIGMA Project to the Field of Intelligent Transportation Systems. In A. Dziech and A. Czyżewski, editors, <i>Multimedia Communications, Services and Security</i> , pages 58–72, Cham, 2014. Springer International Publishing	10 (WoS)
15	R. Tadeusiewicz, W. Chmiel, P. Kadłuczka, and Z. Mikrut. Sposób i system do zarządzania regułami w rozproszonym systemie nadzoru. Number PL 403161 A1, 09 2014	2 (Patent)
16	I. Wojnicki, P. Szwed, W. Chmiel, and S. Ernst. Ontology Oriented Storage, Retrieval and Interpretation for a Dynamic Map System. In A. Dziech and A. Czyżewski, editors, <i>Multimedia Communications, Services and Security</i> , pages 380–391, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg	10 (WoS)
17	P. Szwed, P. Kadłuczka, W. Chmiel, A. Głowacz, and J. Śliwa. Ontology based integration and decision support in the INSIGMA route planning subsystem. In <i>2012 Federated Conference on Computer Science and Information Systems (FedCSIS)</i> , pages 141–148. IEEE Computer Society Press, Sept 2012	10 (WoS)
18	K. Gleba, J. Śliwa, W. Chmiel, P. Szwed, and A. Głowacz. Ontology engineering methodology for intelligent system for global monitoring, detection and identification of threats. <i>Military communications and information technology: a comprehensive approach enabler</i> , pages 133–144, 2011	5 (WoS)
19	B. Filipowicz, W. Chmiel, M. Dudek, and P. Kadłuczka. Efektywność wielopopulacyjnego algorytmu ewolucyjnego dla zagadnień permutacyjnych - performance of multi-population evolutionary algorithms for permutation problems. <i>Automatyka</i> , 15(2):147–158, 2011	4
20	W. Chmiel, J. Kwiecień, and Z. Mikrut. Realization of Scenarios for Video Surveillance. <i>Image Processing & Communications</i> , 17(4):231–240, 2012	7
21	Z. Bubleński, W. Chmiel, M. Jabłoński, P. Kadłuczka, T. Kryjak, Z. Mikrut, P. Pawlik, and R. Tadeusiewicz. System inteligentnego monitoringu przestrzeni i obiektów szczególnego znaczenia SIMPOZ. <i>PAR Pomiar Automatyka Robotyka</i> , 15(12):69–76, 2011	5

22	W. Chmiel, P. Kadłuczka, and S. Jędrusik. Nadzorowana kategoryzacja tekstów angielskojęzycznych — supervised english text classification. <i>Automatyka/Automatics</i> , 14(3/2):811–826, 2010	6
23	B. Filipowicz, W. Chmiel, and P. Kadłuczka. Kontur wypukły w trójwymiarowym zagadnieniu pakowania — Convex contour for 3D bin packing problem. <i>Automatyka/Automatics</i> , 14(3):867–883, 2010	6
<i>Suma</i>		189

5.1.1. Cytowania prac

Liczba cytowań (Tab. 5.4) została określona na podstawie danych zawartych bazach *Web of Science*, *Scopus* oraz *Google Scholar*. Uzyskane wyniki zawierają informacje o liczbie indeksowanych prac, cytowań oraz indeksie Hirscha.

Tablica 5.4: Dane bibliometryczne według różnych baz danych.

Baza	Liczba publikacji	Liczba cytowań		Indeks Hirscha
		wszystkie	bez autocytowań	
Web of Science	19	46	30	4
Scopus	16	51	32	5
Google Scholar	48	149	-	7

5.2. Realizacja krajowych i zagranicznych projektów badawczych

Prace naukowo-badawcze obejmują 9 grantów, w tym jeden międzynarodowy i osiem krajowych. W pięciu z nich Autor referatu był współautorem i pełnił rolę zastępcy kierownika projektu.

-
- *System Inteligentnego Monitoringu Przestrzeni i Obiektów Szczególnego Znaczenia – SIMPOZ* (0128/R/T00/2010/12). Projekt rozwojowy na rzecz obronności i bezpieczeństwa państwa. Kierownik projektu: prof. dr hab. inż. Ryszard Tadeusiewicz. Okres realizacji: 01.01.2010 – 30.03.2013 r. Budżet 2 500 000 zł.

Rola w projekcie: zastępca kierownika projektu.

- *Inteligentny system efektywnej analizy prac diagnostycznych i remontowych urządzeń przemysłowych z zastosowaniem jednostek mobilnych i zaawansowanej analizy obrazów – INRED* (POIR.01.01.01-00-0170/17). Projekt finansowany przez NCBIR w ramach Projektu Operacyjnego Inteligentny Rozwój 2014-2020 r. Oś priorytetowa: Wsparcie prowadzenia prac B+R przez przedsiębiorstwa. Działanie: Projekty B+R przedsiębiorstw. Poddziałanie: Badania przemysłowe i prace rozwojowe realizowane przez przedsiębiorstwa. Kierownik: prof. dr hab. inż. Andrzej Dziech. Projekt realizowany we współpracy z Doosan Babcock Energy Polska. Okres realizacji: 01.01.2018 – 31.12.2022r. Budżet 27 810 516 zł.¹

Rola w projekcie: zastępca kierownika projektu i kierownik projektu na Wydziale EAIiB Akademii Górniczo-Hutniczej w Krakowie.

- *Inteligentny System Informacyjny dla Globalnego Monitoringu, Detekcji i Identyfikacji Zagrożeń - INSIGMA* (POIG.01.01.02-00-062/09). Projekt finansowany przez Europejski Fundusz Rozwoju Regionalnego w ramach Programu Operacyjnego Innowacyjna Gospodarka. Kierownik projektu: prof. dr hab. inż. Andrzej Dziech. Członkowie konsorcjum: Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie (lider), Wojskowa Akademia Techniczna w Warszawie, Wojskowy Instytut Łączności w Zegrzu, Wyższa Szkoła Technik Komputerowych i Telekomunikacji w Kielcach. Kierownik: prof. dr hab. inż. Andrzej Dziech. Okres realizacji: 01.03.2010 - 31.07.2015 r.² Budżet 17 000 000 zł.

Rola w projekcie: zastępca kierownika projektu.

- *Inteligentne znaki drogowe do adaptacyjnego sterowania ruchem pojazdów, komunikujące się w technologii V2X - INZNAK* (POIR.04.01.04-00-0089/16). Projekt finansowany przez NCBIR w ramach Projektu Operacyjnego Inteligentny Rozwój 2014-2020 r., działanie 4.1.4. Kierownik: prof. dr hab. inż. Andrej Czyżewski (lider), kierownik projektu na AGH: prof. dr hab. inż. Andrzej Dziech. Projekt realizowany przez konsorcjum Politechnika Gdańska (lider), Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie, Siled Sp. z o.o., Microsystem Sp. z o.o. Okres realizacji: 01.07.2017 – 30.06.2020 r. Budżet 6 902 082 zł.

Rola w projekcie: zastępca kierownika projektu na AGH i kierownik projektu na wydziale EAIiB AGH.

¹ <http://www.doosanbabcock.com/pl/intro/projekt-inred/>

² <http://insigma.kt.agh.edu.pl/index.php>

-
- *Inteligentny system predykcji dopuszczalnych prędkości ruchu drogowego - RID-INPREDO* (DZP/RID-I-68/14/NCBR/2016). Projekt realizowany w ramach Wspólnego Przedsięwzięcia NCBiR i GDDKiA pt. Rozwój Innowacji Drogowych. Kierownik: prof. dr hab. inż. Andrzej Dziech. Projekt realizowany przez konsorcjum Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie(lider), Politechnika Gdańska, Wyższa Szkoła Technik Komputerowych i Telekomunikacji w Kielcach. Okres realizacji: 01.01.2016 – 30.03.2019 r. Budżet 2 419 500 zł.

Rola w projekcie: zastępca kierownika projektu.

- *Intelligent information system supporting observation, searching and detection for security of citizens in urban environment - INDECT*. Największy projekt europejski koordynowany przez uczelnię w ramach 7 Programu Ramowego (7PR), POIG 4.1.4. Realizowany przez 17 europejskich partnerów pod przewodnictwem Akademii Górniczo-Hutniczej w Krakowie. Kierownik: prof. dr hab. inż. Andrzej Dziech. Okres realizacji: 01.01.2009-31.12.2013 r. Budżet 4 mln 828 tys. euro.

Rola w projekcie: wykonawca.

- *Prototyp systemu zarządzania regułami biznesowymi i technologicznymi - REBIT* (UDA-POIG.01.03.01-12-163/08-01). Projekt realizowany w ramach Priorytetu 1., Działanie 1.3.PO IG, Poddziałanie 1.3.1. Okres realizacji: 2009 – 2012 r.

Rola w projekcie: wykonawca.

- *Inteligentny system wykrywania zagrożeń i ochrony informacji - INTOS*. Kierownik: prof. dr hab. inż. Andrzej Dziech. Okres realizacji: 2011 – 2013 r.

Rola w projekcie: wykonawca

- *Funkcjonalny model automatu z systemem wizyjnym do skaryfikacji oraz oceny żywotności żołądki na podstawie automatycznego rozpoznawania topografii zmian mumifikacyjnych - DĄB* (PBS3/A8/34/2015). Kierownik projektu AGH: prof. dr hab. Ryszard Tadeusiewicz. Okres realizacji: 2016 – 2018 r.

Rola w projekcie: wykonawca

5.3. Udział w komitetach redakcyjnych

- Członek komitetu redakcyjnego międzynarodowej konferencji *International Conference on Multimedia Communications, Services and Security* (<https://link.springer.com/conference/mcss-mult>).

5.4. Współpraca międzynarodowa

- Uczestnictwo w realizacji międzynarodowego projektu - INDECT (*Intelligent information system supporting observation, searching and detection for security of citizens in urban environment*). W jego ramach opracowano m.in. ontologie oraz bazujące na nich metody automatycznego wnioskowania o niebezpiecznych zdarzeniach.

- Współpraca naukowa z *University of Applied Sciences in Wien*. W jej ramach m.in. wygłoszenie cyklu wykładów pod wspólnym tytułem: *An approximation algorithms for \mathcal{NP} -hard combinatorial problems*.

5.5. Współpraca z przemysłem - wdrożenia

5.5.1. Wdrożenia zrealizowane indywidualnie

Prace teoretyczne związane z badaniami i implementacją algorytmów optymalizacyjnych dla zagadnień dyskretnych, ze szczególnym naciskiem na zagadnienia permutacyjne, pozwoliły na realizację szeregu wdrożeń rynkowych.

Cześć rozwiązań opracowano i wdrożono w ramach programu *Innowacje dla przemysłu*. Wdrożenia były finansowane w ramach szerszego projektu p.t. *Wiedza, praktyka, kadry – klucz do sukcesu w biznesie*, Małopolskiej Agencji Rozwoju Regionalnego (MARR), ze środków unijnych z Programu Operacyjnego Kapitał Ludzki.

- „*Opracowanie i wdrożenie zbioru metod optymalizacji dla systemu rozpoznawania obiektów trójwymiarowych dla przedsiębiorstwa Eltcrack System sp. z o.o.*”. Opracowane metody umożliwiły stworzenie rozwiązania pozwalającego na automatyczną identyfikację jednostek pływających. Wdrożenie zrealizowano w ramach POKL, priorytet VIII Regionalne Kadry Gospodarki, D.8.2 Transfer Wiedzy, Wsparcie dla współpracy sfery nauki i przedsiębiorstw, 2012.
- „*Optymalizacja komutacji połączeń w systemie łączności dla służb bezpieczeństwa i ratownictwa dla przedsiębiorstwa Eltcrack System Sp. z o.o.*”. Opracowanie zbioru algorytmów optymalizacyjnych, poprawiających przepustowość i jakość połączeń w standardzie cyfrowej radiotelefonii dyspozytorskiej przeznaczonej dla służb bezpieczeństwa publicznego i ratownictwa. Wdrożenie zrealizowano w ramach POKL, priorytet VIII Regionalne Kadry Gospodarki, D.8.2 Transfer Wiedzy, Wsparcie dla współpracy sfery nauki i przedsiębiorstw, 2013.
- „*Opracowanie systemu zarządzania adaptacyjnego oraz wymiany wiedzy pomiędzy autonomicznymi obiektami w roju w przedsiębiorstwie PROJECTBOX sp. z o.o.*”. Opracowano podstawy teoretyczne, specyfikacja wymagań, specyfikację techniczną, przeprowadzono implementację oraz wdrożenie systemu umożliwiającego współpracę urządzeń autonomicznych. Wdrożenie zrealizowano w ramach POKL, priorytet VIII Regionalne Kadry Gospodarki, D.8.2 Transfer Wiedzy, Wsparcie dla współpracy sfery nauki i przedsiębiorstw, 2015.

5.5.2. Wdrożenia zrealizowane w ramach realizowanych projektów naukowo-badawczych

- Opracowanie otwartego, rozproszonego systemu nadzoru wykrywającego podejrzanе obiekty, naruszenie strefy i podejrzanе zachowanie. Wykryte zdarzenia przez kamery spełniające rolę detektorów, wraz z obrazem opisanym za pomocą metadanych, są transmitowane do centrum zarządzania i składowane w repozytorium. Rozwiązanie opracowano i wdrożono w ramach projektu projektu rozwojowego na rzecz obronności i bezpieczeństwa państwa SIMPOZ.

- System optymalizacji ruchu drogowego poprzez sterowanie strumieniami podjazdów na skrzyżowaniach z zastosowaniem analizy obrazu. Rozwiązanie opracowane i wdrożone pilotażowo w ramach Europejskiego Funduszu Rozwoju Regionalnego, Programu Operacyjnego Innowacyjna Gospodarka, projekt INSIGMA nr POIG.01.01.02-00-062/09. Rozwiązanie uzyskało Platynową Nagrodę na Międzynarodowych Targach Wynalazków i Innowacji - INTARG 2015.
- System zarządzania ruchem drogowym z zastosowaniem algebry przedziałowej. Opracowane rozwiązanie umożliwia interakcyjne, w czasie rzeczywistym, sterowanie pojazdami w sieciach drogowych charakteryzujących się dużą zmiennością czasów przejazdów. Rozwiązanie opracowane i wdrożone pilotażowo w ramach Europejskiego Funduszu Rozwoju Regionalnego, Programu Operacyjnego Innowacyjna Gospodarka, projekt INSIGMA nr POIG.01.01.02-00-062/09 oraz POIR.04.01.04-00-0089/16 w ramach Projektu Operacyjnego Inteligentny Rozwój 2014-2020 r., działanie 4.1.4.
- Opracowanie inteligentnego systemu doboru dopuszczalnych prędkości w ruchu drogowym. Stworzony system ingeruje rozproszone statycznie (opis otoczenia drogi) i dynamiczne dane (parametry opisujące strumienie pojazdów) drogowe GDDKiA dla całej Polski, a regułowy system wnioskujący wspomagający wspomaga wyznaczanie ograniczeń prędkości na drogach krajowych na podstawie analizy danych, także w warunkach dynamicznego sterowania ruchem. Rozwiązanie opracowane i wdrożone w ramach Wspólnego Przedsięwzięcia NCBiR i GDDKiA pt. Rozwój Innowacji Drogowych GDDKiA (RID-INPREDO).
- Opracowanie systemu zarządzania rozproszonym systemem znaków drogowych (IN-ZNAK).

5.6. Nagrody

- **Platinum Award** - *Międzynarodowe Targi Wynalazków i Innowacji INTARG 2015*. Wyróżnienie uzyskane za:
„System wspomagania zarządzania ruchem drogowym oparty o analizę obrazu z kamer”³.
- **Złoty Medal** - *Międzynarodowe Targi Wynalazków i Innowacji INTARG 2015*. Wyróżnienie zyskane za wdrożenie systemu automatycznej identyfikacji pojazdów na podstawie analizy obrazu:
„IdentCAR”.
- **Diamantowy Lider Bezpieczeństwa Państwa** - konkurs *Lider Bezpieczeństwa Państwa 2015* pod patronatem Biura Bezpieczeństwa Narodowego. Wyróżnienie uzyskane za:
„Inteligentny system Wykrywania zagrożeń i Ochrony Informacji - INTOS”.
- **Dyplom Wiceprezesa Rady Ministrów i Ministra Nauki i Szkolnictwa Wyższego w roku 2016**, za:
„Inteligentny system zarządzania ruchem drogowym ”.

³ <http://www.agh.edu.pl/osiagniecia/info/article/piec-zlotych-medali-dla-naukowcow-z-agh-na-intarg/>

5.7. Działalność dydaktyczna w latach 2004-2018

5.7.1. Prowadzone zajęcia

- *Badania Operacyjne* – ćwiczenia III rok Informatyki, st. I.
- *Badania Operacyjne* – ćwiczenia III rok Automatyki i Robotyki, st. I.
- *Badania Operacyjne* – laboratorium III rok Automatyki i Robotyki, st. I.
- *Matematyczne metody wspomagania decyzji* - wykład III r. Automatyki i Robotyki, st. I
- *Matematyczne metody wspomagania decyzji* - laboratorium III r. Automatyki i Robotyki, st.I.
- *Metody Optymalizacji* – Elektronika i Telekomunikacja st. II, r. I.
- *Język C# oraz platforma .NET* – wykład obieralny V r. Informatyki Stosowanej.
- *Język C# oraz platforma .NET* – wykład obieralny V r. Automatyki i Robotyki.
- *Prezentacja Danych i Multimedia* – wykład, Socjologia st. I, r. II Wydział Humanistyczny.
- *Prezentacja Danych i Multimedia* – ćwiczenia, Socjologia st. I, r. II Wydział Humanistyczny.
- *Integracja przemysłowych systemów informatycznych* – wykład, laboratorium, projekt st. II, sem. II, Informatyka Stosowana.
- *Systemy informatyczne zarządzania przedsiębiorstwem* – laboratorium, st. II. sem. I, Automatyka i Robotyka.
- *Zagadnienia zarządzania przedsiębiorstwem* – laboratorium, st. II. sem. I, Automatyka i Robotyka.
- *Inteligencja obliczeniowa* - wykład, laboratorium st. II, sem. II, Automatyka i Robotyka.
- *Workflow - Zarządzanie informacją i dokumentami* - wykład, laboratorium st. II, sem. II, Informatyka Stosowana.

5.7.2. Działalność organizacyjna

- Udział w pracach Wydziałowej Komisji Rekrutacyjnej EAIiB AGH (do 2010 r.).
- Udział w jury Konferencji Studenckich Kół Naukowych Pionu Hutniczego AGH, 2005 r.
- Coroczna praca w Komisji ds. Egzaminów Dyplomowych dla kierunku Automatyka i Robotyka (od 2013 r. do chwili obecnej) – uczestnictwo w obronach prac inżynierskich i egzaminach kierunkowych.
- Współpraca z *University of Applied Sciences in Wien*.
- Pobyt w dniach 31.10 - 02.11.2018 w *University of Applied Sciences in Wien* i wygłoszenie cyklu prezentacji.

5.7.3. Prace inżynierskie i dyplomowe

Od 2004 roku promotor 44 prac magisterskich, 65 inżynierskich. Recenzent 37 prac magisterskich i 40 inżynierskich.

5.7.4. Recenzje artykułów w czasopismach międzynarodowych

1. *International Journal of Applied Mathematics and Computer Science* (IF: 1.694) (1 recenzja - 2009, 1 recenzja - 2018).
2. *Design and Architectures for Signal and Image Processing* (1 recenzja - 2013).
3. *Journal of Systems Architecture* (IF:0.913) (1 recenzja - 2017).
4. *Information Sciences*(IF: 4.305) (1 recenzja - 2017).
5. *Multimedia Tools and Applications* (IF:1.541) (1 recenzja - 2015, 1 recenzja - 2016, 1-2017, 2 recenzje - 2018).
6. *Metrology and Measurement Systems* (IF:1.523) (1 recenzja - 2017).
7. *Image Processing & Communication* (1 recenzja - 2015).

Recenzje materiałów konfederacyjnych:

- Service Oriented Computing and Application (SOECA) (1 recenzja - 2018).

5.8. Inne

- Promotor pomocniczy w pracy doktorskiej pt. „*Metody optymalizacji procesów zarządzania pracą*”. Promotorem przewodu jest prof. dr. hab. inż. Ryszard Tadeusiewicz. Praca realizowania od 2016 roku.

.....
Miejsce, data

.....
Wojciech Chmiel

Bibliografia

- [1] Andrzej Dziech and Wojciech Chmiel and Sebastian Ernst and Piotr Kadłuczka and Zbigniew Mikrut and Piotr Szwed and Igor Wojnicki. System and the method to manage the rules of control of the traffic lights on the crossing, PL414870A1, BUP 11/2017, 11 2017.
- [2] B.Filipowicz, W. Chmiel, and P. Kadłuczka. Kontur wypukły w trójwymiarowym zagadnieniu pakowania —Convex contour for 3D bin packing problem. *Automatyka/Automatics*, 14(3):867–883, 2010.
- [3] Z. Bublński, W. Chmiel, M. Jabłoński, P. Kadłuczka, T. Kryjak, Z. Mikrut, P. Pawlik, and R. Tadeusiewicz. System inteligentnego monitoringu przestrzeni i obiektów szczególnego znaczenia SIMPOZ. *PAR Pomiary Automatyka Robotyka*, 15(12):69–76, 2011.
- [4] W. Chmiel. Evolutionary algorithm using conditional expectation value for quadratic assignment problem. *Swarm and Evolutionary Computation*, 46:1 – 27, 2019.
- [5] W. Chmiel, J. Dańda, A. Dziech, S. Ernst, A. Głowacz, P. Kadłuczka, Z. Mikrut, P. Pawlik, P. Szwed, and I. Wojnicki. Contribution of the INSIGMA Project to the Field of Intelligent Transportation Systems. In A. Dziech and A. Czyżewski, editors, *Multimedia Communications, Services and Security*, pages 58–72, Cham, 2014. Springer International Publishing.
- [6] W. Chmiel, J. Dańda, A. Dziech, S. Ernst, P. Kadłuczka, Z. Mikrut, P. Pawlik, P. Szwed, and I. Wojnicki. INSIGMA: an intelligent transportation system for urban mobility enhancement. *Multimedia Tools and Applications*, 75(17):10529–10560, Sep 2016.
- [7] W. Chmiel, A. Dziech, S. Jędrusik, P. Kadłuczka, J. Kwiecień, P. Szwed, Z. Mikrut, and G. Rogus. Rule system for speed limit determination on national roads in Poland. *MATEC Web Conf.*, 231:02001, 2018.
- [8] W. Chmiel and P. Kadłuczka. Wielofazowa metoda różnicowania populacji w algorytmie ewolucyjnym —a multi-phase diversification method of population in the evolutionary algorithm. *Automatyka/Automatics*, 1797(151):195–202, 2008.
- [9] W. Chmiel, P. Kadłuczka, and S. Ernst. A Multicriteria Model for Dynamic Route Planning. In A. Dziech and A. Czyżewski, editors, *Multimedia Communications, Services and Security*, pages 174–182, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.
- [10] W. Chmiel, P. Kadłuczka, K.Wala, and S.Jędrusik. Algorytmy heurystyczne w trójwymiarowym zagadnieniu pakowania — heuristic algorithm for three-dimensional packing problem. *Automatyka/Automatics*, 14(3/2):827–840, 2010.

- [11] W. Chmiel, P. Kadłuczka, J. Kwiecień, and B. Filipowicz. A comparison of nature inspired algorithms for the quadratic assignment problem. *Bulletin of the Polish Academy of Sciences: Technical Sciences*, 65(4):513–522, 2017.
- [12] W. Chmiel, P. Kadłuczka, J. Kwiecień, B. Filipowicz, and P. Pukocz. Strategic Planning Optimization Using Tabu Search Algorithm. In A. M. Skulimowski and J. Kacprzyk, editors, *Knowledge, Information and Creativity Support Systems: Recent Trends, Advances and Solutions*, pages 315–328, Cham, 2016. Springer International Publishing.
- [13] W. Chmiel, P. Kadłuczka, and G. Packanik. Performance of Swarm Algorithms for Permutation Problems. *Automatyka*, 15(2):117–126, 2009.
- [14] W. Chmiel, P. Kadłuczka, and S. Jędrusik. Nadzorowana kategoryzacja tekstów angielskojęzycznych — supervised english text classification. *Automatyka/Automatics*, 14(3/2):811–826, 2010.
- [15] W. Chmiel and J. Kwiecień. Quantum-Inspired Evolutionary Approach for the Quadratic Assignment Problem. *Entropy*, 20(10), 2018.
- [16] W. Chmiel, J. Kwiecień, and Z. Mikrut. Realization of Scenarios for Video Surveillance. *Image Processing & Communications*, 17(4):231 – 240, 2012.
- [17] W. Chmiel, I. Skalna, and S. Jędrusik. Intelligent route planning system based on interval computing. *Multimedia Tools and Applications*, Oct 2018.
- [18] W. Chmiel and P. Szwed. Learning Fuzzy Cognitive Map for Traffic Prediction Using an Evolutionary Algorithm. In A. Dziech, M. Leszczuk, and R. Baran, editors, *Multimedia Communications, Services and Security*, volume 566 of *Communications in Computer and Information Science*, pages 195–209. Springer International Publishing, 2015.
- [19] W. Chmiel and P. Szwed. Bees Algorithm for the Quadratic Assignment Problem on CUDA Platform. In A. Gruca, A. Brachman, S. Kozielski, and T. Czachórski, editors, *Man–Machine Interactions 4*, volume 391 of *Advances in Intelligent Systems and Computing*, pages 615–625. Springer International Publishing, 2016.
- [20] B. Filipowicz, W. Chmiel, M. Dudek, and P. Kadłuczka. Efektywność wielopopulacyjnego algorytmu ewolucyjnego dla zagadnień permutacyjnych - performance of multi-population evolutionary algorithms for permutation problems. *Automatyka*, 15(2):147–158, 2011.
- [21] B. Filipowicz, W. Chmiel, and P. Kadłuczka. Ukierunkowane przeszukiwanie przestrzeni rozwiązań w algorytmach rojowych — guided search using particle swarm optimization algorithm. *Automatyka*, 13(2):247–255, 2009.
- [22] B. Filipowicz, W. Chmiel, P. Kadłuczka, and P. Wicijowski. Greedy algorithms for three dimensional packing problem. *Automatyzacja procesów dyskretnych : teoria i zastosowania*, 1:49–60, 2012.
- [23] K. Gleba, J. Śliwa, W. Chmiel, P. Szwed, and A. Głowacz. Ontology engineering methodology for intelligent system for global monitoring, detection and identification of threats. *Military communications and information technology: a comprehensive approach enabler*, pages 133–144, 2011.
- [24] P. Kadłuczka, W. Chmiel, and J. Piwowarczyk. Strategia przetwarzania wiedzy w wielopopulacyjnym algorytmie ewolucyjnym — knowledge processing strategy in the multi-population evolutionary algorithm. *Automatyka/Automatics*, 13(2):325–331, 2009.

- [25] P. Rotter and W. Chmiel. A framework for a hierarchical model of cooperation between unmanned airplanes. *European Journal of Remote Sensing*, 51(1):276–284, 2018.
- [26] J. Śliwa, K. Gleba, W. Chmiel, P. Szwed, and A. Głowacz. IOEM - Ontology Engineering Methodology for Large Systems. In P. Jędrzejowicz, N. T. Nguyen, and K. Hoang, editors, *Computational Collective Intelligence. Technologies and Applications*, pages 602–611, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.
- [27] P. Szwed, W. Chmiel, and P. Kadłuczka. OpenCL Implementation of PSO Algorithm for the Quadratic Assignment Problem. In L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, and J. M. Zurada, editors, *Artificial Intelligence and Soft Computing*, volume 9120 of *Lecture Notes in Computer Science*, pages 223–234. Springer International Publishing, 2015.
- [28] P. Szwed, P. Kadłuczka, W. Chmiel, A. Głowacz, and J. Śliwa. Ontology based integration and decision support in the INSIGMA route planning subsystem. In *2012 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 141–148. IEEE Computer Society Press, Sept 2012.
- [29] P. Szwed, P. Skrzyński, and W. Chmiel. Risk assessment for a video surveillance system based on Fuzzy Cognitive Maps. *Multimedia Tools and Applications*, 75(17):10667–10690, Sep 2016.
- [30] R. Tadeusiewicz, W. Chmiel, P. Kadłuczka, and Z. Mikrut. Sposób i system do zarządzania regułami w rozproszonym systemie nadzoru. Number PL 403161 A1, 09 2014.
- [31] I. Wojnicki, P. Szwed, W. Chmiel, and S. Ernst. Ontology Oriented Storage, Retrieval and Interpretation for a Dynamic Map System. In A. Dziech and A. Czyżewski, editors, *Multimedia Communications, Services and Security*, pages 380–391, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.