

# Porównanie algorytmów ACO, PSO i zachłannego w rozwiązywaniu problemu marszrutyzacji

Karolina Drabent

Grzegorz Kacprowicz

8 kwietnia 2020

# Spis treści

<b>1</b>	<b>Wstęp</b>	<b>5</b>
<b>2</b>	<b>Opis problemu</b>	<b>6</b>
2.1	Przykładowa instancja problemu . . . . .	7
<b>3</b>	<b>Wybrane algorytmy</b>	<b>9</b>
3.1	Algorytm zachłanny . . . . .	9
3.2	Algorytm losowy . . . . .	10
3.3	Standardowy algorytm mrówkowy . . . . .	10
3.4	Wariant Elitist . . . . .	12
3.5	Wariant Rank . . . . .	12
3.6	Algorytm rojowy . . . . .	12
3.6.1	Operacja zamiany . . . . .	14
3.6.2	Sekwencja zamian . . . . .	14
3.6.3	Prędkość i kierunek . . . . .	14
3.6.4	Wyliczenie następnego położenia . . . . .	14
<b>4</b>	<b>Hipoteza badawcza</b>	<b>16</b>
4.1	Stawiane hipotezy . . . . .	16
4.2	Weryfikacja hipotezy . . . . .	17
4.2.1	Hipoteza 1 . . . . .	17

4.2.2	Hipoteza 2 . . . . .	17
4.2.3	Hipoteza 3 . . . . .	18
4.2.4	Hipoteza 4 . . . . .	18
4.2.5	Hipoteza 5 . . . . .	18
4.3	Opis eksperymentów dotyczących wyznaczenia parametrów . . . . .	18
<b>5</b>	<b>Wyniki</b>	<b>20</b>
5.1	Hipoteza 1 . . . . .	20
5.2	Hipoteza 2 . . . . .	22
5.3	Hipoteza 3 . . . . .	23
5.4	Hipoteza 4 . . . . .	25
5.5	Hipoteza 5 . . . . .	26
<b>6</b>	<b>Podział pracy</b>	<b>28</b>

# Streszczenie

W poniższym dokumencie przedstawiony jest opis przeprowadzonych badań nad różnymi metodami optymalizacji rozwiązania problemu marszrutyzacji (z ang. *Capacitated Vehicle Routing Problem*), czyli wariantu problemu komiwojażera, w którym zakłada się ograniczoną ładowność pojazdu, a co za tym idzie, najczęściej konieczność kilkukrotnego powrotu do wierzchołka początkowego. Poniższa praca skupiać się będzie na kilku wybranych przez autorów algorytmach: algorytm mrówkowy (ACO) w wariacie standardowym, w wariacie *Rank* i wariacie *Elitist*; algorytm optymalizacji rojem cząstek (PSO); algorytm zachłanny oraz algorytm losowy. Badania przeprowadzone przez autorów opierają się o pomiar skuteczności danego algorytmu głównie na podstawie liczby wykonanych iteracji, czasu potrzebnego do wykonania obliczeń, a także porównanie z wynikiem uzyskiwanym przez algorytm zachłanny w takiej samej konfiguracji grafu wejściowego. W postawionych przez autorów hipotezach znajduje się również zbadanie szybkości wykonania poszczególnych metod optymalizacji względem algorytmu zachłannego. Wykazano, że już dla niewielkiej liczby iteracji algorytm zachłanny ustępuje jakością zwracanego wyniku wszystkim wariantom algorytmu mrówkowego. Pokazano także, że dla wybranych początkowych parametrów, ze wszystkich algorytmów mrówkowych to wersja *Rank* osiągała najlepsze wyniki dla grafów o różnym rozmiarze. Pokazano też, że dla przyjętej, względnie niedużej, liczby iteracji żaden algorytm nie był w stanie osiągnąć minimum globalnego, a jedynie zbliżyć się do niego. Opisany i zbadany w dokumencie algorytm optymalizacyjny rojem cząstek dla problemu marszrutyzacji był próbą autorów zastosowania nieco odmiennego podejścia względem algorytmów mrówkowych do rozwiązania tego konkret-

nego problemu grafowego. Pokazano, że wyniki powyższej metody dla zadanej liczby iteracji sprawdziły się gorzej od algorytmów mrówkowych, lecz wciąż lepiej od algorytmu wybierającego drogę w sposób przypadkowy.

**Słowa kluczowe:** problem marszrutyzacji, CVRP, algorytm mrówkowy, optymalizacja rojem cząstek, ACO, PSO, Rank AS, Elitist AS

# Rozdział 1

## Wstęp

Poniższa praca jest raportem końcowym z wykonanych badań nad różnymi sposobami rozwiązywania problemu marszrutyzacji dla wybranych parametrów.

Dokument zawiera wprowadzenie teoretyczne pozwalające czytelnikowi zrozumieć koncepcję problemu marszrutyzacji wraz z zaprezentowanym przykładem w formie graficznej. Co więcej, przedstawiony jest również dokładny opis teoretyczny wszystkich wykorzystywanych w badaniu algorytmów. W dalszej części pracy postawione zostaje pięć hipotez, wraz z koniecznymi warunkami określającymi przebieg badań oraz wybranymi parametrami. Następnie przedstawione są dane wynikowe i wnioski płynące z każdej z hipotez, z jednoczesną informacją mówiącą o przyjęciu lub odrzuceniu hipotezy. Na końcu znajduje się także przegląd literatury.

# Rozdział 2

## Opis problemu

Problemem badań będzie **problem marszrutyzacji** (ang. *Capacitated Vehicle Routing Problem*). Jest to problem optymalizacyjny, który jest wariantem problemu komiwojażera. Polega on na wyznaczeniu optymalnych tras przewozowych dla pewnej liczby środków transportu (ciężarówek) o ograniczonej ładowności, której zadaniem jest obsłużenie zbioru klientów (sklepów) o różnym zapotrzebowaniu na produkty i różnej lokalizacji.

Możemy ten problem definiować grafem pełnym, ważonym i nieskierowanym, w którym  $V = (\text{magazyn}, \text{sklep}_1, \text{sklep}_2, \dots, \text{sklep}_n)$ . Dodatkowo każdy wierzchołek  $\text{sklep}_i \in [1, n]$  ma skorelowaną ze sobą wartość  $z_i$  oznaczającą zapotrzebowanie, gdzie  $z_i \in \mathbb{N}$ . Magazyn nie ma żadnego zapotrzebowania, natomiast po odwiedzeniu magazynu ciężarówka odzyskuje swoją całkowitą ładowność, z tego też wierzchołka ciężarówka startują oraz kończą. Dla naszych badań przyjmujemy, że graf ten jest euklidesowy, czyli jest osadzony w metryce euklidesowej (m.in. spełnia nierówność trójkąta). Wtedy waga krawędzi  $v_1 v_2$  to odległość z  $v_1$  do  $v_2$  w metryce euklidesowej.

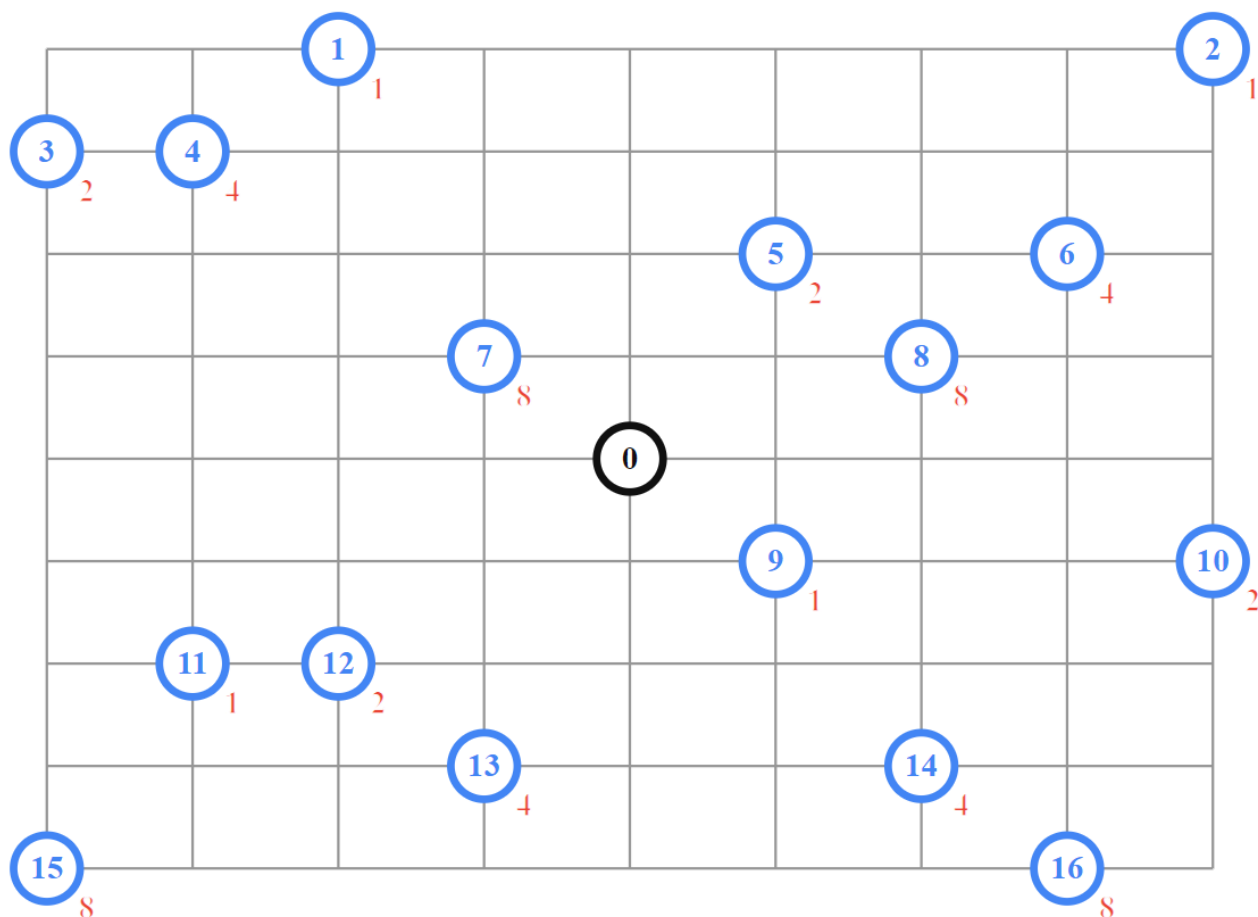
Parametr  $l$  oznacza ładowność ciężarówek. Ciężarówki jest nieograniczona liczba. W sytuacji gdybyśmy musieli zastosować ograniczoną ich liczbę, można traktować jeden cykl nie jako ciężarówkę, ale jako jeden kurs jednej ciężarówki, wtedy jedna ciężarówka wykonuje wiele

kursów.

W zadaniu optymalizowana jest łączna liczba kilometrów, które muszą przejechać wszystkie ciężarówki we wszystkich kursach.

## 2.1 Przykładowa instancja problemu

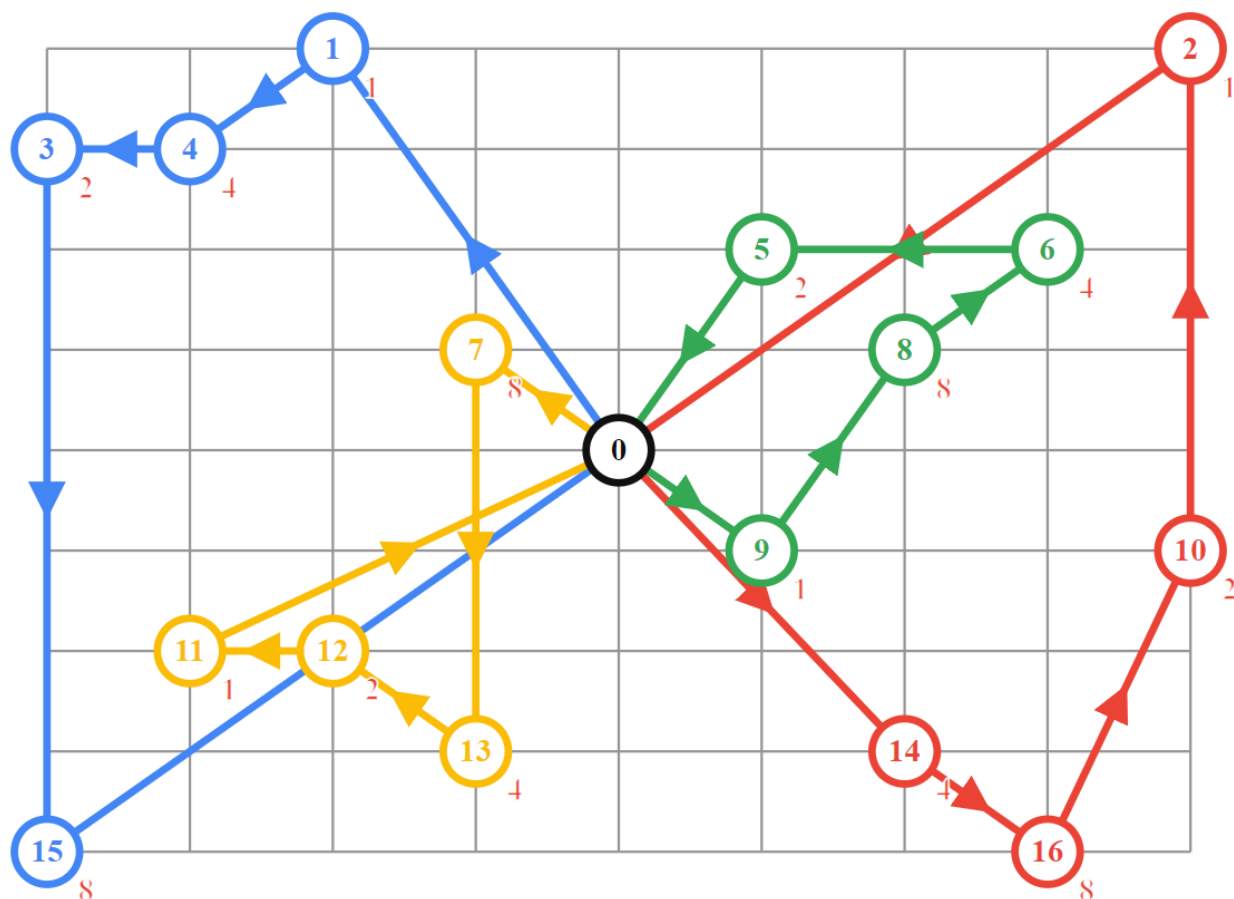
Wizualną reprezentacją przykładu jest Rysunek 5.5. W okręgach znajdują się numery wierzchołków a kolorem czerwonym przy wierzchołkach oznaczone jest zapotrzebowanie. Wierzchołek o numerze zero jest wierzchołkiem magazynu.



Rysunek 2.1: Przykładowa instancja problemu[2]



Gdzie wierzchołek magazynu ma współrzędne (0,0). Przykładowe rozwiązanie jest zaprezentowane na Rysunku 2.2. Osobny kolor reprezentuje jeden kurs ciężarówki.



Rysunek 2.2: Przykładowe rozwiązanie problemu

Koszt wykonania kursów w tym przykładzie to około 52 jednostki. Przykład jest zaczerpnięty ze źródła [2].

# Rozdział 3

## Wybrane algorytmy

Do rozwiązywania zadania będą użyte przez nas następujące algorytmy:

1. algorytm mrówkowy w podstawowej wersji (ACO),
2. algorytm mrówkowy w wersji Elitist AS,
3. algorytm mrówkowy w wersji Rank AS,
4. algorytm rojowy (PSO),
5. algorytm zachłanny,
6. algorytm losowy.

Dla każdego algorytmu sprawdzony będzie również czas wykonania. Liczba mrówek lub owadów w roju to 200.

### 3.1 Algorytm zachłanny

Algorytm zachłanny wybiera ten wierzchołek, do którego jest najmniejsza odległość i do którego ciężarówka jest w stanie rozładować cały towar. W następnej kolejności rozważa dwa wierzchołki - magazyn i sklep. Jeżeli nie ma takiego sklepu, to jedzie do magazynu.

## 3.2 Algorytm losowy

Algorytm losowy, tak jak zachłanny, wybiera wierzchołek spośród możliwych do odwiedzenia w danym momencie, jednak dokonuje on wyboru w sposób losowy. Algorytm ten jest dodany jedynie w celu porównania wyników.

## 3.3 Standardowy algorytm mrówkowy

Algorytm mrówkowy, inspirowany zachowaniem prawdziwych kolonii mrówek, jest algorytmem stosowanym do szukania dobrych dróg w grafie. Główną jego cechą jest fakt, że pojedyncza mrówka przemieszcza się “losowo”, jednak kolonia jako całość przemieszcza się w sposób zorganizowany. Dzieje się tak na podstawie zostawianego przez mrówki feromonu na krawędziach grafu – wybierają one drogę na podstawie jego stężenia. Pod uwagę brana jest też odległość między poszczególnymi wierzchołkami grafu. Po pewnym czasie z największym prawdopodobieństwem wybierana jest ścieżka najkrótsza. Na Listingu 3.1 przedstawiony jest pseudokod algorytmu w wersji podstawowej.

Listing 3.1: Pseudokod algorytmu mrówkowego

Resetuj feromon na krawedziach

Dla kazdej iteracji:

Stworz/resetuj mrowki

Dla kazdej mrowki:

Znajdz rozwiazanie

Jesli jest lepsze od najlepszego, to zapamietaj

Wyparuj czesc feromonu

Dla kazdej mrowki:

Rozloz feromon proporcjonalnie do znalezionej drogi

Zwroc najkrotsza droge

Wybór wierzchołka do którego uda się mrówka, zależy w głównej mierze od dwóch czynników: wartości feromonu na krawędzi łączącej bieżący wierzchołek z rozważanym oraz odległość wierzchołków – w przypadku rozważanego przez nas problemu CVRP, najkorzystniejsza jest najkrótsza. Prawdopodobieństwo  $p_{ij}$  przejścia mrówki z wierzchołka  $i$  do  $j$  można wyrazić wzorem 3.1.

$$p_{ij} = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{k \in allowed(i)} [\tau_{ik}]^\alpha [\eta_{ik}]^\beta} \quad (3.1)$$

gdzie

- $\tau_{ij}$  – ilość feromonu na krawędzi między wierzchołkami  $i$  i  $j$ ,
- $\eta_{ij}$  – odwrotność odległości między wierzchołkami  $i$  i  $j$ ,
- $allowed(i)$  – wierzchołki możliwe do odwiedzenia przez mrówkę z wierzchołka  $i$ ,
- $\alpha$  – parametr określający jak bardzo należy wziąć pod uwagę wartość feromonu,
- $\beta$  – parametr określający jak bardzo należy wziąć pod uwagę odwrotność odległości między wierzchołkami.

### 3.4 Warianit Elitist

Jest to modyfikacja standardowego algorytmu mrówkowego, gdzie po położeniu przez wszystkie mrówki feromonu następuje dodatkowe zwiększenie feromonu na krawędziach najlepszego znalezionej rozwiązania przez mrówki, które je znalazły.

$$e \cdot Q_{best} \tag{3.2}$$

gdzie

- $e$  - liczba mrówek, które znalazły najlepsze rozwiązanie,
- $Q_{best}$  wartość feromonu na najlepszej ścieżce.

### 3.5 Warianit Rank

Jest to kolejna modyfikacja standardowego algorytmu mrówkowego, w której tylko wybrana liczba mrówek  $\sigma$  rozkłada feromon na końcu iteracji. Brane są pod uwagę te mrówki, które znalazły najlepszą drogę i rozkładają one feromon w ilości proporcjonalnej do jakości rozwiązania, czyli od swojej rangi  $\mu$ .

$$(\sigma - \mu) \cdot Q_{\mu} \tag{3.3}$$

gdzie

- $Q_{\mu}$  – wartość feromonu na ścieżce  $\mu = 0, \dots, \sigma - 1$ , gdzie  $\mu$  zależne jest od pozycji w hierarchii najlepszych rozwiązań.

### 3.6 Algorytm rojowy

W algorytmie rojowym cząstki (agenci) poszukują optymalnego rozwiązania poruszając się w jego pobliżu w przestrzeni rozwiązań [4]. Każda cząstka porusza się indywidualnie po

przestrzeni rozwiązań - posiada w iteracji  $n$  swoje własne współrzędne  $x_n$  i prędkość  $v_n$ . Jej kolejne współrzędne ( $x_{n+1}$ ) wyznaczone są ze wzoru:

$$x_{n+1} = x_n + v_{n+1} \quad (3.4)$$

$V_{n+1}$  jest prędkością w iteracji  $n+1$  i jest wyznaczane z odległości do najlepszego rozwiązania globalnego( $g_{best}$ ), najlepszego rozwiązania personalnego( $p_{best}$ ) oraz z poprzedniej prędkości( $v_n$ ), ze wzoru:

$$v_{n+1} = w \cdot v_n + c_p \cdot rand() \cdot (p_{best}x_n) + c_g \cdot rand() \cdot (g_{best}x_n) \quad (3.5)$$

gdzie  $c_g$  - współczynnik przyspieszania związany z  $g_{best}$ ,  $c_p$  - współczynnik przyspieszania związany z  $p_{best}$ ,  $g_{best}$  - jest wektorem prędkości do rozwiązania  $g_{best}$ ,  $p_{best}$  - jest wektorem prędkości do rozwiązania  $p_{best}$ ,  $w$  - współczynnik związany z prędkością  $v_n$ . Zatem algorytm optymalizacji rojem cząstek różni się od algorytmu mrówkowego głównie osobistą pamięcią cząstki.

Jednak tak opisany algorytm nie daje możliwości zastosowania do problemu w przestrzeni z wartościami dyskretnymi [3] lub takiej, w której trudno jest zdefiniować odległość pomiędzy dwoma punktami, co występuje w problemie marszrutyzacji. Zatem aby dostosować algorytm PSO do naszego problemu potrzebne jest odpowiednie zdefiniowanie prędkości i kierunku dla dwóch punktów w przestrzeni.

Założmy, że graf w zadaniu jest  $d$ -wymiarowy. Wtedy punkt  $p_1$  będzie określony przez  $d-1$  współrzędnych:  $w_1, w_2, \dots, w_{d-1}$ , gdzie  $w_1, \dots, w_{d-1}$  są wierzchołkami sklepów a  $w_0$  jest wierzchołkiem magazynu. W przyjętej notacji, w trasie nie jest zapisywany wierzchołek magazynu, ponieważ jest on tzw. domyślny, czyli występuje on na trasie kiedy ciężarówka musi zjechać do magazynu, ponieważ jest przepełniona, czyli ładunek następnego sklepu + ładunek na ciężarówce  $<$  ładowność ciężarówki. Domyślny wierzchołek magazynu jest konieczny do zadziałania algorytmu PSO.

### 3.6.1 Operacja zamiany

Aby zdefiniować prędkość cząstki potrzebne jest zdefiniowanie operacji zamiany [5]. Operacja zamiany  $MO(i_1, i_2)$  oznacza zamianę wierzchołków o indeksach  $i_1, i_2$  na trasie oraz  $0 \leq i_1, i_2 < d$ , np:

$$d=5, p_1 = w_2, w_1, w_4, w_3 \quad p_2 = p_1 + MO(2, 3) = w_2, w_4, w_1, w_3.$$

### 3.6.2 Sekwencja zamian

Operacje zamiany można stosować sekwencyjnie, czyli jedna po drugiej, np:

$$\begin{aligned} d=5, p_1 = w_2, w_1, w_4, w_3, S = [MO(2, 3), M(1, 2), M(0, 1)] \quad p_2 = p_1 + S \\ = p_1 + MO(2, 3) + M(1, 2) + M(0, 1) = w_1, w_2, w_4, w_3. \end{aligned}$$

Wtedy  $S$  jest sekwencją. Sekwencje można konkatelować, np:

$$\begin{aligned} S_1 = [MO(2, 3), M(1, 2), M(0, 1)], S_2 = [MO(1, 0), M(2, 3)] \\ S_1 + S_2 = [MO(2, 3), M(1, 2), M(0, 1), MO(1, 0), M(2, 3)] \end{aligned}$$

### 3.6.3 Prędkość i kierunek

Prędkością cząstki jest sekwencja  $S$ . Wektorem prędkości od punktu  $p_1$  do punktu  $p_2$  jest taka sekwencja  $S$ , że:

$$p_1 + S = p_2 \tag{3.6}$$

Wtedy umownie możemy powiedzieć, że  $p_2 - p_1 = S$ .

### 3.6.4 Wyliczenie następnego położenia

Z tak opisanymi współrzędnymi, prędkością oraz kierunkiem potrzebne jest lekkie zmniejszenie wzoru 3.5.

$$v_{n+1} = \gamma \cdot v_n + \alpha \cdot (p_{best}x_n) + \beta \cdot (g_{best}x_n) \tag{3.7}$$

gdzie:  $\gamma \in [0, 1]$  jest współczynnikiem zapominania,  $\alpha \in [0, 1]$  jest współczynnikiem odpowiadającym prędkości do  $p_{best}$ ,  $\beta \in [0, 1]$  jest współczynnikiem zapominania odpowiadającym prędkości do  $g_{best}$ . Mechanizm działania jest następujący gdy wyliczane jest  $v_{n+1}$  losowana jest wartość  $r$  z przedziału  $(0, 1)$  dla każdego współczynnika, jeśli  $r > \gamma$  to sekwencja odpowiadająca temu współczynnikowi jest dodawana do  $v_{n+1}$ , w przeciwnym wypadku jest pomijana. Stąd następne położenie cząstki ( $x_{n+1}$ ) to  $x_n + v_{n+1}$ .



# Rozdział 4

## Hipoteza badawcza

### 4.1 Stawiane hipotezy

Stawiamy następujące hipotezy badawcze:

1. Po 50 iteracjach wszystkie warianty algorytmów mrówkowych i PSO będą miały gorsze rezultaty niż algorytm zachłanny.
2. PSO będzie szybciej (w sensie liczby iteracji) znajdował optimum globalne od algorytmów mrówkowych, jeśli zostanie ono znalezione podczas 200 iteracji,
3. Po 100 iteracjach algorytmy mrówkowe i PSO będą znacząco lepsze od algorytmu zachłannego, tzn. będą miały średnio 20% lepszy wynik.
4. Po 150 iteracjach, najlepiej zoptymalizowany wynik będzie przez PSO, wariant algorytmu mrówkowego Rank AS, następny Elitist AS, po nim podstawowy wariant algorytmu mrówkowego a najgorzej zoptymalizowany wynik będzie przez algorytm zachłanny.
5. Najlepszy czas wykonania będzie miał algorytm zachłanny, który będzie miał co najmniej 50% szybszy czas wykonania od pozostałych algorytmów w chwili osiągnięcia przez nie wyniku lepszego lub równemu algorytmowi zachłannemu.

Dla wszystkich wymienionych hipotez algorytmy będą miały ustalone parametry wybrane drogą własnych eksperymentów.

## 4.2 Weryfikacja hipotezy

W ocenie sprawności algorytmu uwzględnione będzie porównanie skuteczności i czasu obliczeń poszczególnych podejść, uwzględniając rozmiar wejściowego problemu i różne warianty algorytmu mrówkowego. Hipotezy zostały przetestowane na trzech grafach z jednego zbioru danych [1]. Grafy te mają wymiary (liczbę wierzchołków): 51, 76, 101. Aby przekonać się czy wyniki są powtarzalne testy zostały wykonane 4 razy, czyli dla 4 różnych ziaren generatora: 6666, 7777, 8888, 9213. Następnie została wzięta z nich średnia arytmetyczna i ta średnia traktowana jest jako ostateczny wynik. Ustalone parametry opisane są w podrozdziale 4.3.

### 4.2.1 Hipoteza 1

Do zweryfikowania tej hipotezy zostały przeprowadzone obliczenia dla algorytmu optymalizacji rojem cząstek, podstawowego algorytmu mrówkowego, algorytmu mrówkowego w wariacie Rank oraz Elitist. Liczba iteracji dla powyższych algorytmów to 50. Do porównania wykonano również obliczenia dla algorytmu losowego, wykonano go raz dla każdego ziarna.

### 4.2.2 Hipoteza 2

Przyjęto, że optimum globalne jest to wartość z zakresu  $[\text{optimum globalne}, \text{optimum globalne} + 0.05 * \text{optimum globalne}]$ . Do zweryfikowania tej hipotezy zostały przeprowadzone obliczenia dla algorytmu optymalizacji rojem cząstek, podstawowego algorytmu mrówkowego, algorytmu mrówkowego w wariacie Rank oraz Elitist. Liczba iteracji to 200 dla wszystkich algorytmów. Porównywana liczba to numer iteracji, w którym znaleziono optimum.

### 4.2.3 Hipoteza 3

Do zweryfikowania tej hipotezy zostały przeprowadzone obliczenia dla algorytmu optymalizacji rojem cząstek, podstawowego algorytmu mrówkowego, algorytmu mrówkowego w wariacie Rank oraz Elitist. Liczba iteracji to 100 dla powyższych algorytmów. Wykonano również raz obliczenia dla algorytmu zachłannego. Do porównania wykonano również obliczenia dla algorytmu losowego, wykonano go raz dla każdego ziarna.

### 4.2.4 Hipoteza 4

Do zweryfikowania tej hipotezy zostały przeprowadzone obliczenia dla algorytmu optymalizacji rojem cząstek, podstawowego algorytmu mrówkowego, algorytmu mrówkowego w wariacie Rank oraz Elitist. Liczba iteracji to 150 dla powyższych algorytmów. Wykonano również raz obliczenia dla algorytmu zachłannego. Do porównania wykonano również obliczenia dla algorytmu losowego, wykonano go raz dla każdego ziarna.

### 4.2.5 Hipoteza 5

Wybrano algorytm, który wykonał najszybciej obliczenia dla zbliżonej wartości jaką obliczył algorytm zachłanny. to znaczy, mierzono czas, dla którego uzyskał wynik przynajmniej tak samo dobry jaki uzyskał algorytm zachłanny.

## 4.3 Opis eksperymentów dotyczących wyznaczenia parametrów

Aby rzetelnie porównać wymienione algorytmy najpierw należy wyznaczyć odpowiednie parametry, przy których będą one osiągać możliwie najlepsze wyniki. Wykonano szereg testów na grafach o rozmiarach 22, 33 [1], liczbie mrówek ( $n$ ) 200 oraz ze względu na ograniczony

czas – o liczbie iteracji 50. Parametry są opisane w tablicy 4.1.

Parametr	Wartość
$\gamma$	0.5
$\alpha$	0.85
$\beta$	0.55
$\frac{\sigma}{n}$	0.2
$\rho$	0
$\alpha_{\text{aco}}$	10
$\beta_{\text{aco}}$	4

Tablica 4.1: Tablica wybranych parametrów

Gdzie  $\alpha, \beta, \gamma$  są to parametry algorytmu PSO. Natomiast  $\alpha_{\text{aco}}, \beta_{\text{aco}}$  są parametrami algorytmów mrówkowych. Parametr  $\frac{\sigma}{n}$  to ułamek mrówek, które roznoszą feromony w wariancie Rank -  $\sigma$  jest to liczba tych mrówek,  $\rho$  jest parametrem parowania feromonów. Kiedy wynosi on 0, feromony nie parują wcale.

# Rozdział 5

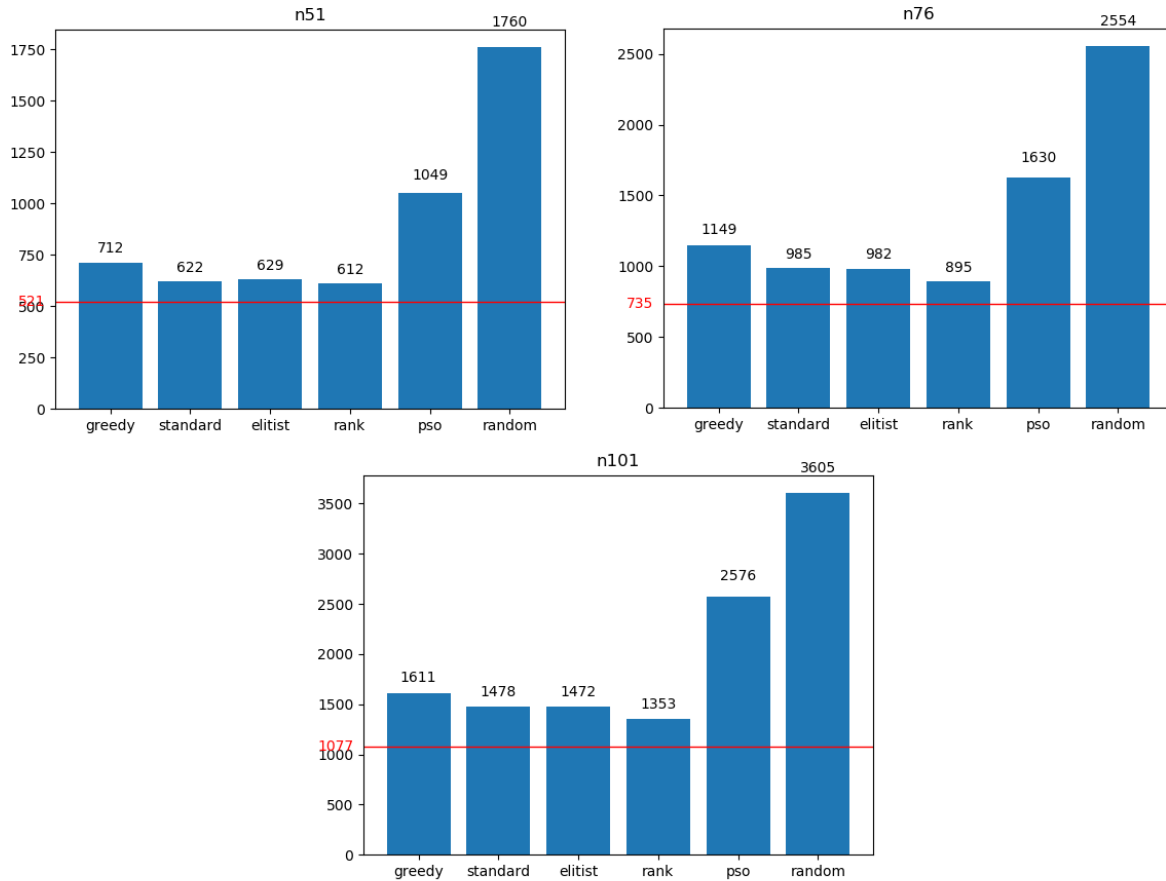
## Wyniki

### 5.1 Hipoteza 1

*Po 50 iteracjach wszystkie warianty algorytmów mrówkowych i PSO będą miały gorsze rezultaty niż algorytm zachłanny.*

Hipoteza to została zweryfikowana dla parametrów ustalonych w wyżej i została **odrzucona**.

Wyniki uzyskane przez algorytmy przedstawione są na Rysunku 5.1.



Rysunek 5.1: Wyniki po 50 iteracjach algorytmu

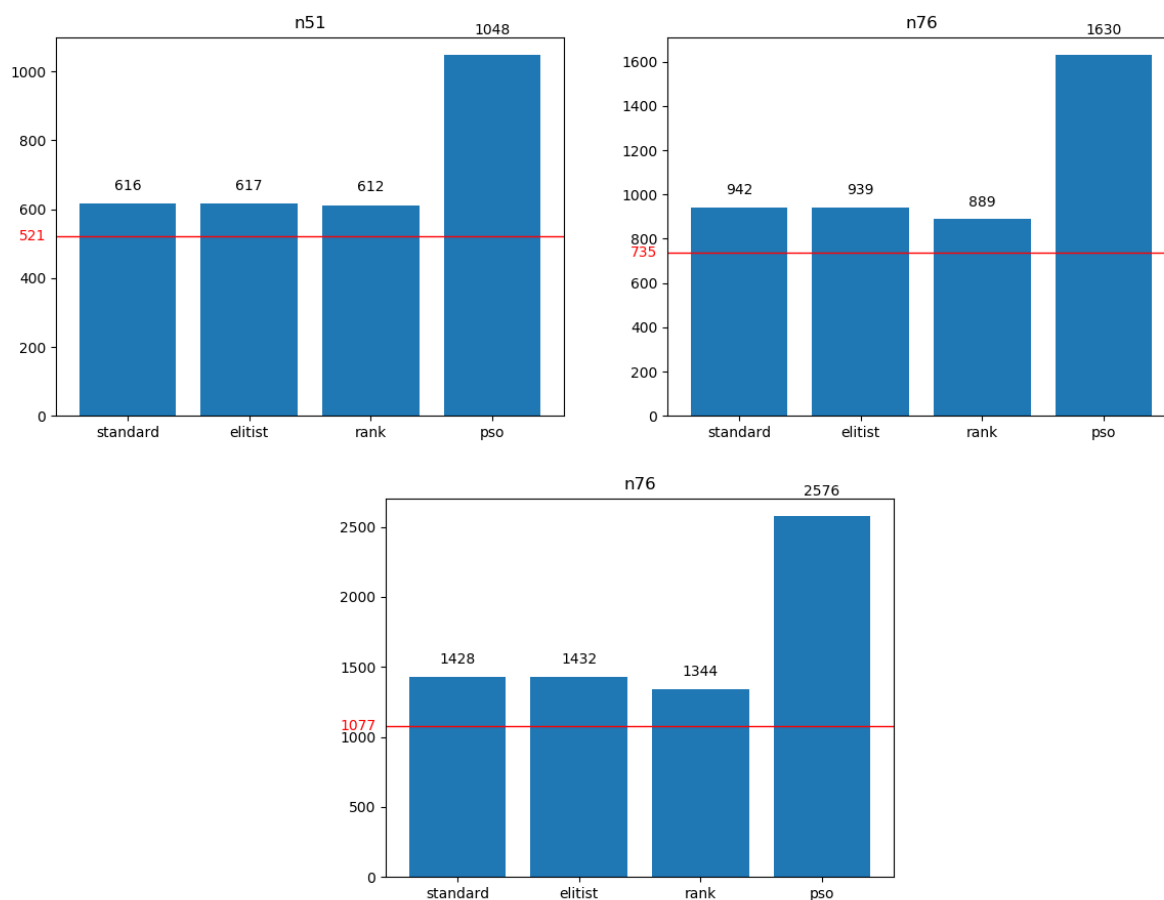
Eksperymenty przeprowadzone były dla czterech różnych ziaren, a następnie zostały uśrednione. Jak widać, dla wszystkich trzech grafów algorytmy mrówkowe osiągają nieco lepsze wyniki już po 50 iteracjach niż algorytm zachłanny. Najlepiej spisał się algorytm mrówkowy w wersji *rank*, który dla każdego grafu po 50 iteracjach znalazł ścieżkę najkrótszą względem pozostałych. Algorytmy mrówkowe w wersji standardowej oraz w wersji *elitist* dla wszystkich przypadków grafów wypadły podobnie po 50 iteracjach, również pokonując algorytm zachłanny. Im większy był rozmiar grafu, tym bliżej były one wyniku wersji *rank*. Najgorszy wynik dla wszystkich przypadków osiągnął algorytm rojowy PSO, który wypadł zdecydowanie gorzej niż algorytm zachłanny. Aby lepiej zobrazować uzyskane wyniki, dla porównania z

prawej strony wykresu został zamieszczony wynik algorytmu, który wybiera ścieżki całkowicie przypadkowo.

## 5.2 Hipoteza 2

*PSO będzie szybciej (w sensie liczby iteracji) znajdował optimum globalne od algorytmów mrówkowych, jeśli zostanie ono znalezione podczas 200 iteracji.*

Hipoteza **odrzucona**, jak widać na Rysunku 5.2.



Rysunek 5.2: Wyniki po 200 iteracjach algorytmu

Żaden z algorytmów opisanych w hipotezie nie osiągnął optimum globalnego, ani też nie

zbliżył się do rozwiązania gorszego o 5% od rozwiązania optymalnego. Co więcej, nawet 200 iteracji nie starczyło, aby algorytm PSO zbliżył się jakością wyniku do algorytmów mrówkowych i wypada on na ich tle zdecydowanie gorzej.

## 5.3 Hipoteza 3

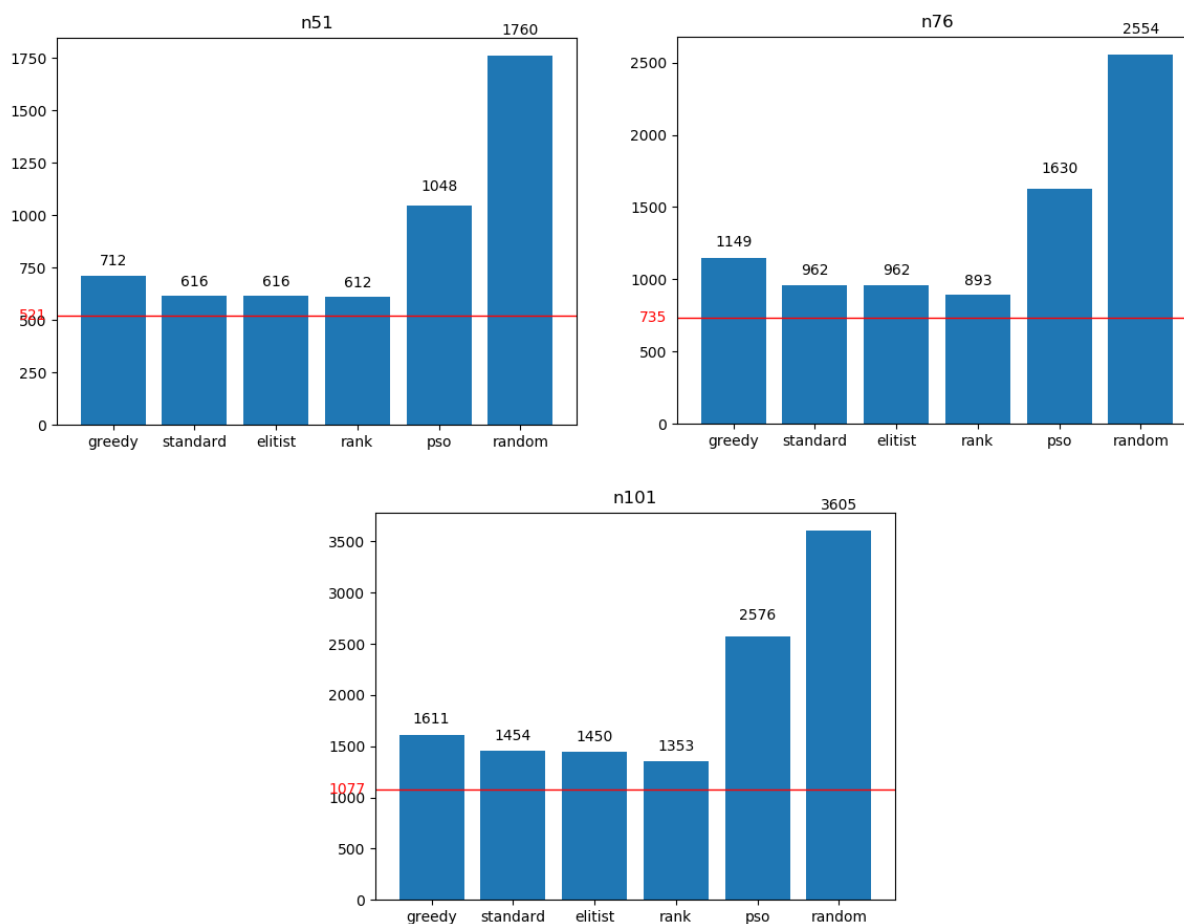
*Po 100 iteracjach algorytmy mrówkowe i PSO będą znacząco lepsze od algorytmu zachłanego, tzn. będą miały średnio 20% lepszy wynik.*

Hipoteza **odrzucona**. Wyniki dla trzech rozmiarów grafów zaprezentowane są na Rysunku 5.3.



Tablica 5.1: Wartości dla algorytmu zachłannego

Rozmiar grafu	Wynik algorytmu zachłannego	Wynik o 20% lepszy
51	712	570
76	1149	919
101	1611	1289



Rysunek 5.3: Wyniki po 100 iteracjach algorytmu

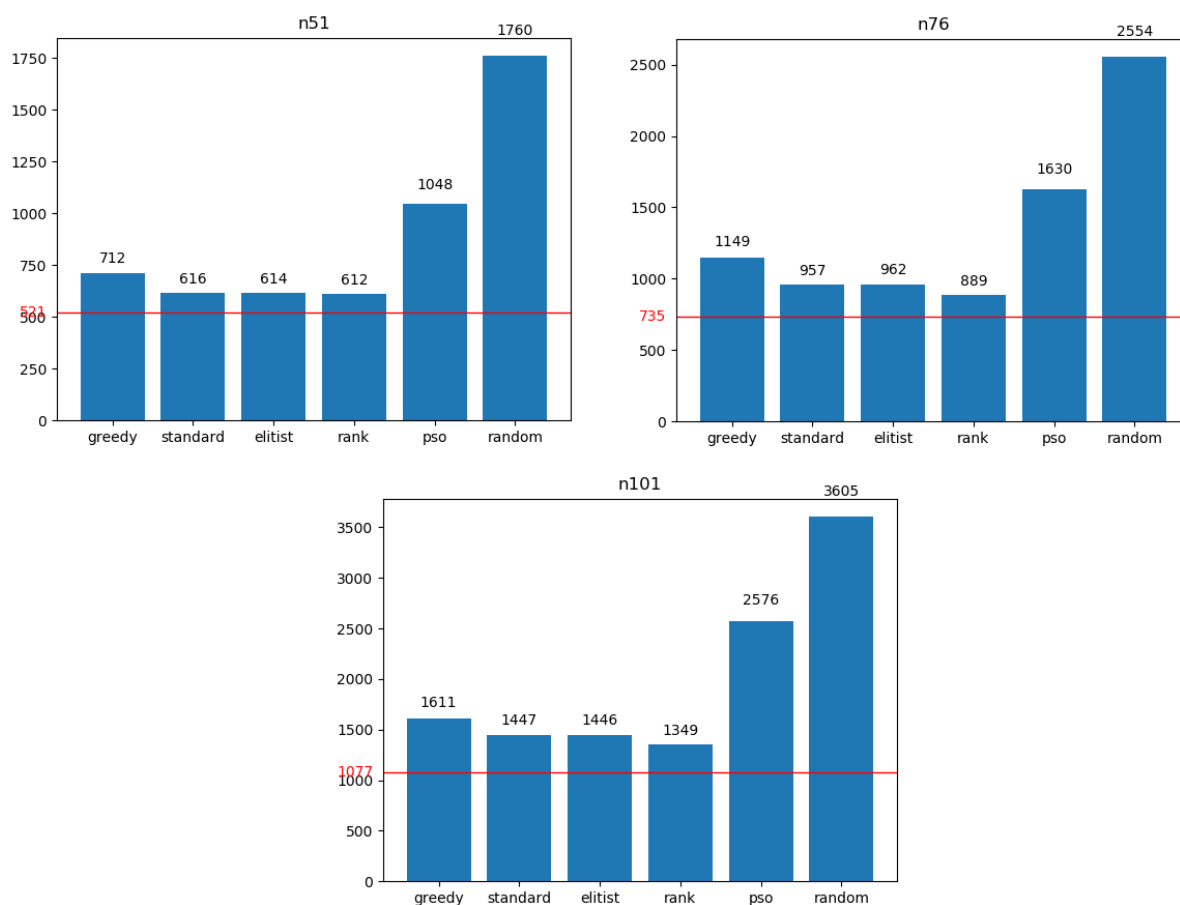
Po zapoznaniu się z Tabelą 5.3 można zauważyć, że dla grafów o rozmiarze 51 oraz 101 żaden algorytm nie był o 20% lepszy niż algorytm zachłanny. Taka sama konkluzja dotyczy grafu o rozmiarze 76 z wyjątkiem algorytmu mrówkowego w wersji *rank*, który okazał się o

22,6% lepszy niż algorytm zachłanny.

## 5.4 Hipoteza 4

*Po 150 iteracjach, najlepiej zoptymalizowany wynik będzie przez PSO, wariant algorytmu mrówkowego Rank AS, następny Elitist AS, po nim podstawowy wariant algorytmu mrówkowego a najgorzej zoptymalizowany wynik będzie przez algorytm zachłanny.*

Ponownie, hipoteza **odrzucona** – bazując na wynikach przedstawionych na Rysunku 5.4.



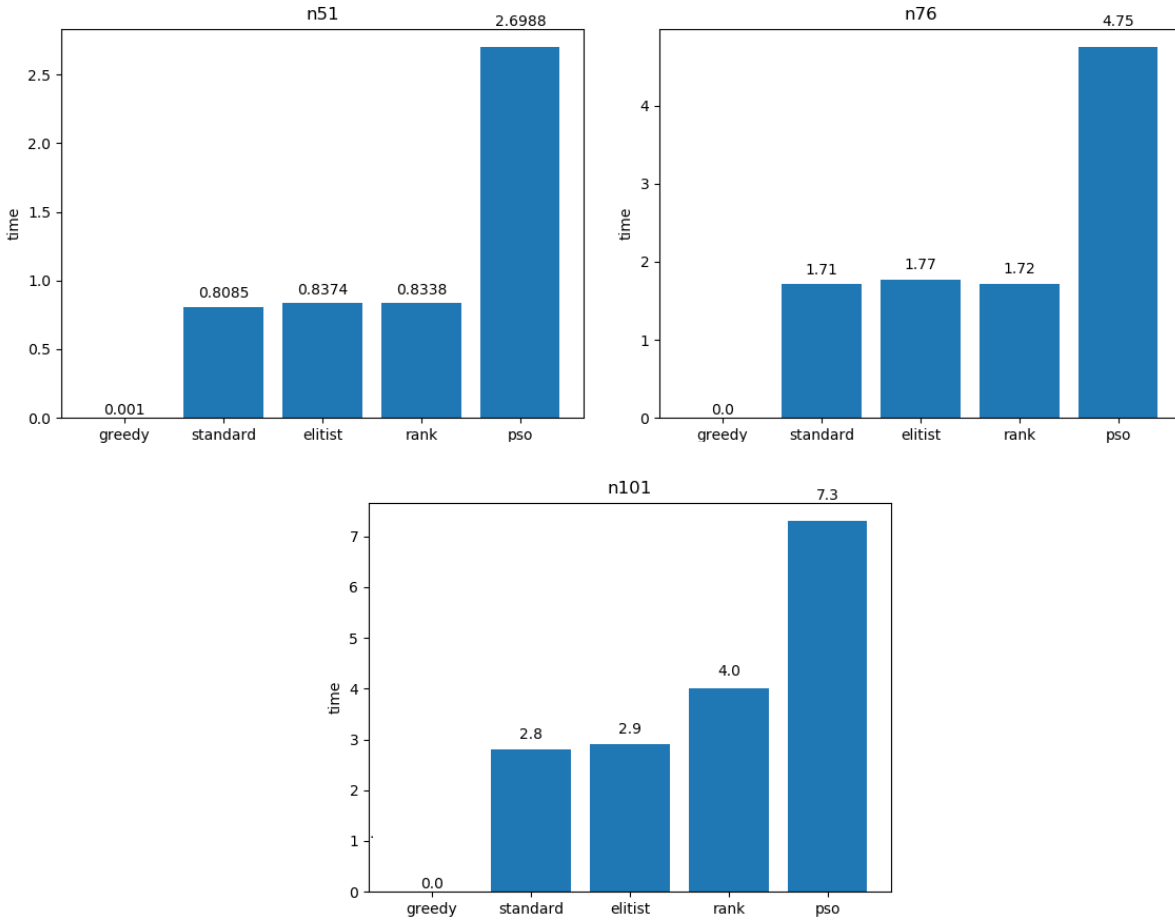
Rysunek 5.4: Wyniki po 150 iteracjach algorytmu

Spośród wymienionych w hipotezie algorytmów, paradoksalnie to PSO okazał się najgor-

szy, mimo oczekiwanego najlepszego wyniku. Najlepszy zaś okazał się algorytm mrówkowy w wariancie *rank*, po nim następnie były algorytmy mrówkowe standardowy i *elitist* osiągające bardzo zbliżone wyniki dla wszystkich grafów. Mimo przewidywanego najgorszego wyniku, algorytm zachłanny osiągnął wynik drugi najgorszy po PSO.

## 5.5 Hipoteza 5

*Najlepszy czas wykonania będzie miał algorytm zachłanny, który będzie miał co najmniej 50% szybszy czas wykonania od pozostałych algorytmów w chwili osiągnięcia przez nie wyniku lepszego lub równemu algorytmowi zachłannemu.* Hipoteza **potwierdzona**. Wyniki czasowe osiągnięcia przez inne algorytmy wartości takiej, którą osiąga algorytm zachłanny, są przedstawione na rysunku poniżej a czas mierzony był w sekundach dla 50 iteracji.



Rysunek 5.5: Szybkość dorównania algorytmowi zachłannemu

Algorytm zachłanny wykonywał się zdecydowanie najszybciej i trwało to dla każdego rodzaju grafu ułamek sekundy. Inne zaś algorytmy miały czas wykonania sięgający od niecałej sekundy dla grafu o rozmiarze 51 po kilka sekund dla grafu o rozmiarze 101. Dla każdego z przypadków czasy wykonania wszystkich algorytmów mrówkowych były do siebie zbliżone, natomiast dla największego grafu wykonanie wariantu *rank* zajęło już o sekundę dłużej względem innych algorytmów mrówkowych. We wszystkich przypadkach najdłużej wykonywał się algorytm PSO.

# Rozdział 6

## Podział pracy

Część pracy	Osoba wykonująca
Wczytywanie danych	Grzegorz Kacprowicz
Algorytm zachłanny	Karolina Drabent
Algorytm podstawowy	Karolina Drabent, Grzegorz Kacprowicz
Algorytm Elitist AS	Karolina Drabent
Algorytm Rank AS	Grzegorz Kacprowicz
Algorytm PSO	Karolina Drabent, Grzegorz Kacprowicz
Dobór parametrów	Karolina Drabent
Wykonywanie testów	Grzegorz Kacprowicz
Raport końcowy	Karolina Drabent, Grzegorz Kacprowicz

# Bibliografia

- [1] Zbiór danych. [online]. URL <http://www.dca.fee.unicamp.br/projects/infobiosys/vrp/>. Dostęp: 08-04-2020.
- [2] Strona internetowa developers google. [online]. URL <https://developers.google.com/optimization/routing/cvrp>. Dostęp: 08-04-2020.
- [3] WU Zhi-ming CHEN Ai-ling<sup>†</sup>, YANG Gen-ke. Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem. [online], 2012. URL [http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.513.8837&rep=rep1&type=pdf&fbclid=IwAR2Zf\\_30ltP5hWrMCZsH00-D0gZoQJnaNWijAupdtOM9EiACgVhH270F530](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.513.8837&rep=rep1&type=pdf&fbclid=IwAR2Zf_30ltP5hWrMCZsH00-D0gZoQJnaNWijAupdtOM9EiACgVhH270F530). Dostęp: 08-04-2020.
- [4] Prof. Jacek Mańdziuk. Wykłady o algorytmach rojowych i mrówkowych. [online]. URL [http://www.mini.pw.edu.pl/~mandziuk/#page\\_content-07\\_teaching-msi2](http://www.mini.pw.edu.pl/~mandziuk/#page_content-07_teaching-msi2). Dostęp: 08-04-2020.
- [5] Chaitalee K. Patel Yogesh P. Kosta Sarman K. Hadia, Arjun H. Joshi. Solving city routing issue with particle swarm optimization. [online], 2012. URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.258.7026&rep=rep1&type=pdf>. Dostęp: 08-04-2020.