

NOWY !!!

dr inż. Tomasz Kapłon
mail: tomasz.kaplon@pwr.edu.pl

PEA PRJ (wersja 1: 03.10.22)

Spis treści	
<u>Komunikacja</u>	2
<u>Opis projektu</u>	2
<u>Zadanie 1</u>	3
<u>Zadanie 2</u>	4
<u>Zadanie 3</u>	5
<u>Zadanie 4</u>	6
<u>Zadanie 5</u>	7
<u>Zadanie 6</u>	8
<u>Zadanie 7</u>	10
<u>Dane testowe</u>	11
<u>Działanie programów</u>	12
<u>Schemat raportu</u>	12
<u>Oceny</u>	13
<u>Terminy realizacji</u>	13
<u>Terminy spotkań, przekazania raportów i programów</u>	13
<u>Uwagi do raportów</u>	14

Komunikacja

([spis treści](#))

1. Wykłady, projekty i konsultacje odbywają się systemie stacjonarnym.
2. Terminy konsultacji (**KNS**) podane poniżej. Proszę umawiać się mailowo na konkretną godzinę w ramach podanych terminów.

	11:15 - 13:00	13:15-15:00	15:15-16:55	17:05-18:45
pn		LUC lab	LUC lab	
wt		PEA w	PEA prj	PEA prj
śr	LUC lab	KNS	PEA prj	
cz	EUG prj	EUG prj	KNS	LUC w/ EUG w
pt	PEA prj	PEA prj	PEA prj	PEA prj

3. Zadania i informacje dla grup projektowych umieszczane będą na ePortalu.
4. Na ePortalu należy umieszczać raporty (pdf) oraz kolejne wersje realizowanych programów ([źródło, dane, plik wykonywalny] (zip)). Raporty w wersji papierowej wręczacie mi osobiście przed prezentacją programu.

Opis projektu

([spis treści](#))

Zadanie polega na stworzeniu, implementacji oraz zbadaniu efektywności (czasowej i pamięciowej) kilku algorytmów rozwiązujących problem komiwojażera. Algorytmy mają zostać stworzone w oparciu o metody: przeglądu zupełnego (brute force), programowania dynamicznego (dynamic programming), podziału i ograniczeń (branch and bound), symulowanego wyżarzania (simulated annealing), poszukiwania z zakazami (tabu search), algorytmów genetycznych (genetic algorithm), algorytmów mrówkowych (ant colony optimization). Projekt podzielony jest na zadania. Zadań jest siedem (7). Obowiązkowe do wykonania są zadania 1, zadanie 2 lub zadanie 3, zadanie 4 lub zadanie 5 oraz zadanie 6 lub zadanie 7. Oznacza to, że aby móc myśleć o zaliczeniu projektu konieczne jest wykonanie czterech (4) zadań, np. 1, 2, 5 i 6. Liczba wykonanych zadań oraz jakość opracowanego sprawozdania (składającego się z kolejnych - codwutygodniowych - raportów) ma wpływ na ocenę końcową z projektu (patrz: Oceny).

Zadanie 1

([spis treści](#))

Zadanie polega na stworzeniu algorytmu opartego o metodę przeglądu zupełnego. (W raporcie ma znaleźć się krótki opis metody oraz opis algorytmu, najlepiej w postaci czytelnego schematu blokowego z komentarzami wyjaśniającymi funkcje/role kolejnych bloków.) Program ma stanowić samodzielną całość, czyli służyć tylko i wyłącznie do rozwiązywania problemu komiwojażera opracowanym algorytm (metodą przeglądu zupełnego). Proszę nie tworzyć "uniwersalnego" programu umożliwiającego (z menu) wyboru dowolnej metody/algorytmu i konieczności wyboru z kolejnych menu różnych opcji, nazw i ustawiania warunków. Uniwersalność jest słaba, patrz BMW X6 :-)

Poprawność działania algorytmu należy sprawdzać na danych od dra Mierzwy i/lub dra Rudego.

Zadanie 2

([spis treści](#))

Zadanie polega na implementacji algorytmu Held-Karpi. Można pokusić się również o samodzielne opracowanie algorytmu. Podejrzewam jednak, że w najlepszym wypadku wyjdzie z tego algorytm... Held-Karpi :-). Należy zbadać efektywność algorytmu w rozwiązywaniu problemu komiwojażera. Na pewno należy zbadać zależność czasu uzyskiwania rozwiązania od wielkości instancji (złożoność czasową, podobnie jak w przypadku zadania 1.), oraz zależność zużycia pamięci w zależności od wielkości instancji (złożoność pamięciową). Konstrukcja raportu jak do zadania 1. Proszę uwzględnić uwagi i sugestie przekazane w trakcie zajęć, na których omawiane były raporty z zadania nr 1.

Algorytm należy sprawdzać na danych od dra Mierzwy i/lub dra Rudego oraz na instancjach ze stron:

<http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/index.html>

burma14.tsp

gr17.tsp

gr21.tsp

gr24.tsp

bays29.tsp

att48.tsp

eil51.tsp

berlin52.tsp

st70.tsp

gr120.tsp

<http://elib.zib.de/pub/mp-testdata/tsp/tsplib/atsp/index.html>

br17.atsp

ftv33.atsp

ftv35.atsp

ftv38.atsp

p43.atsp

ftv44.atsp

ft53.atsp

ftv70.atsp

Jeżeli do instancji nie określono wartości optymalnej, proszę pominąć jej badanie.

Nie ma konieczności zbadania algorytmu dla wszystkich ww. instancji. Niemniej, pięć to minimum. Należy również znaleźć granicę możliwości algorytmu bądź sprzętu, którym dysponujecie, czyli praktycznie bądź teoretycznie, dla jak dużej instancji można uzyskać rozwiązanie w czasie do 30 minut bądź do wyczerpania pamięci. Wybierając instancje proszę dobierać je tak, aby punkty na wykresie (w zakresie badania) były równomiernie rozłożone pomiędzy najmniejszą (6 węzłów) a największą (granica możliwości) instancją.

Zadanie 3

([spis treści](#))

Zadanie polega na opracowaniu algorytmu w oparciu o metodę podziału i ograniczeń (Branch & Bound). W metodzie B&B do przeszukiwania przestrzeni rozwiązań wykorzystuje się metody: breadth search, depth search i low cost (best search). Należy zbadać wpływ każdej z metod na czas uzyskiwania wyniku oraz na zużycie pamięci. W raporcie muszą znaleźć się krótkie acz precyzyjne opisy ww. metod (w punkcie "Algorytm").

Implementacja i badanie jednej z ww. metod pozwoli na uzyskanie maksymalnie oceny 4,0 z zadania. Implementacja dwóch i trzech, odpowiednio 4,5 i 5,0.

Zadanie 4

([spis treści](#))

Opracowanie i implementacja algorytmu w oparciu o metodę symulowanego wyżarzania

Zadanie polega na opracowaniu i implementacji algorytmu rozwiązywania problemu komiwojażera z wykorzystaniem metody symulowanego wyżarzania.

W rozwiązaniu należy zbadać wpływ:

- sposobu wyboru temperatury początkowej T_0 ,
- sposobu wyboru rozwiązania początkowego x^0 ,
- schematu chłodzenia (*Boltzmann*, *Cauchy'ego* i geometryczny),
- długości epoki L ,
- sposobu przeglądania sąsiedztwa (np. *greedy*, *steepest*, inne znane z literatury) oraz
- sposobu wyboru rozwiązania w sąsiedztwie rozwiązania bieżącego (np. 2–zamiana),

na

- jakość uzyskiwanych rozwiązań (wielkość błędu – różnicy pomiędzy rozwiązaniem uzyskanym i optymalnym),
- czas uzyskiwania rozwiązań oraz
- zużycie pamięci.

Rozwiązanie minimalne (na 3,0) to opracowanie algorytmu i jego implementacja z zastosowaniem:

- geometrycznego schematu chłodzenia,
- uzasadnienia sposobu wyboru temperatury początkowej i rozwiązania początkowego,
- przeglądu sąsiedztwa typu *greedy* oraz
- wyboru rozwiązania w sąsiedztwie typu 2–zamiany.

Rozwiązanie na 4,0 obejmuje, oprócz powyższego:

- zbadanie i porównanie dwóch schematów chłodzenia (geometryczny i *Boltzmann* lub *Cauchy'ego*) oraz
- dwóch sposobów wyboru rozwiązania w sąsiedztwie (2–zamiany i innego znanego z literatury).

Rozwiązanie na 5,0 obejmuje:

- dwa schematy chłodzenia (geometryczny i *Boltzmann* lub *Cauchy'ego*),
- przedstawienie co najmniej dwóch sposobów wyboru temperatury początkowej i rozwiązania początkowego oraz uzasadnienia wyboru. w każdym przypadku, implementowanych rozwiązań,
- zbadania wpływu sposobu przeglądania sąsiedztwa (np. *greedy*, *steepest*, inne znane z literatury) oraz
- co najmniej dwóch sposobów wyboru rozwiązania w sąsiedztwie (2–zamiany i innego znanego z literatury).

Dodatkowo oceniane będzie:

- badanie wpływu liczby epok na czas i jakość rozwiązania oraz
- porównanie jakości uzyskiwanego rozwiązania (czas, pamięć i wielkość błędu) z dotychczas zbadanymi metodami.

Zadanie 5

([spis treści](#))

Opracowanie i implementacja algorytmu w oparciu o metodę *Tabu Search*

Zadanie polega na opracowaniu i implementacji algorytmu rozwiązywania problemu komiwojażera przy użyciu algorytmu wykorzystującego metodę *Tabu Search* (poszukiwania z zakazami).

Zbadać należy wpływ:

- rozmiaru listy ruchów zakazanych (tabu list),
- długości kadencji,
- kryterium aspiracji,
- strategii dywersyfikacji,
- sposobu wyboru rozwiązania początkowego,
- sposobu wyboru rozwiązań w sąsiedztwie rozwiązania bieżącego,
- sposobu określenia kryterium zakończenia oraz
- sposobu obsługi zdarzeń krytycznych

na jakość i czas uzyskiwanych rozwiązań oraz zużycia pamięci.

Należy również określić:

- zestaw wartości ww. parametrów umożliwiających uzyskiwanie najlepszych możliwych rozwiązań ze względu na wielkość błędu oraz
- granicę możliwości zastosowania metody *Tabu Search*, z uwzględnieniem specyfikacji dostępnego sprzętu oraz narzędzi programistycznych, czyli wielkość instancji (dla problemu symetrycznego i asymetrycznego), dla której następuje wyczerpanie zasobów (czasu i/lub pamięci).

Rozwiązanie na 3,0 obejmuje opracowanie i implementację algorytmu oraz uzyskiwanie wyników z błędem poniżej 50% dla dowolnej instancji poniżej granicy wyczerpania zasobów.

Rozwiązanie na 4,0 obejmuje dodatkowo zbadanie wpływu na jakość i czas rozwiązania:

- rozmiaru listy tabu,
- sposobu określenia i zastosowania kryterium aspiracji (co najmniej jednego) oraz
- sposobu określenia i zastosowania kryterium zakończenia.

Rozwiązanie na 5,0 obejmuje oprócz ww. również zbadanie wpływu na jakość i czas rozwiązania:

- długości kadencji
- sposobu wyboru rozwiązań w sąsiedztwie rozwiązania bieżącego oraz
- określenia i wyboru obsługi zdarzeń krytycznych.

Dodatkowo oceniane będzie porównanie jakości uzyskiwanego rozwiązania (czas, pamięć i wielkość błędu) z dotychczas zbadanymi metodami.

UWAGA (do obu zadań):

Algorytm należy sprawdzać na danych od dra Mierzwy i/lub dra Rudego oraz na instancjach ze stron... patrz strona 4 oraz

- gr96.tsp, kroa100.tsp, krob150.tsp, krob200.tsp, pr152.tsp,
- kr124.atsp, ftv170.atsp, rbg323.atsp.

Zadanie 6

([spis treści](#))

Opracowanie i implementacja algorytmu w oparciu o (metodę) *Algorytm genetyczny*

Zadanie polega na opracowaniu i implementacji algorytmu rozwiązywania problemu komiwojażera przy użyciu algorytmu wykorzystującego (metodę) *Algorytm genetyczny*.

Zbadać należy wpływ, na jakość i czas uzyskiwanych rozwiązań oraz zużycia pamięci:

- wielkości populacji N ,
- warunku stopu,
- prawdopodobieństwa krzyżowania p_k ,
- prawdopodobieństwa mutacji p_m ,
- metody selekcji,
- metody sukcesji,
- metody krzyżowania,
- metody mutacji.

Należy również określić:

- zestaw wartości ww. parametrów umożliwiających uzyskiwanie najlepszych możliwych rozwiązań ze względu na wielkość błędu oraz
- granicę możliwości zastosowania (metody) *Algorytmu genetycznego*, z uwzględnieniem specyfikacji dostępnego sprzętu oraz narzędzi programistycznych, czyli wielkość instancji (dla problemu symetrycznego i asymetrycznego), dla której następuje wyczerpanie zasobów (czasu i/lub pamięci).

Rozwiązanie na **3,0** obejmuje opracowanie i implementację algorytmu z jedną metodą:

- selekcji,
- krzyżowania,
- mutacji,
- sukcesji.

Konieczne jest zbadanie wpływu wartości prawdopodobieństw krzyżowania i mutacji oraz wielkości populacji i warunku zatrzymania - co najmniej jednego z: czas, liczba iteracji, liczba iteracji bez poprawy wyniku, wielkość błędu bezwzględnego dla instancji do: $n = 24$, równego 0%, do $n = 400$ poniżej 50%, do $n = 2500$ poniżej 150%.

Na wykresach należy pokazać wpływ N , p_k i p_m na szybkość osiągnięcia wartości funkcji dopasowania:

- optymalnej dla instancji do $n = 70$,
- nie gorszej niż 50% dla instancji do $n = 350$,
- nie gorszej niż 150% dla instancji do $n = 1500$.

Po ustaleniu najlepszych wartości parametrów (N , p_k , p_m , warunek stopu), należy przeprowadzić badania na wybranych instancjach: burma14.tsp, gr17.tsp, gr21.tsp, gr24.tsp, bays29.tsp, ftv33.atsp, ftv44.atsp, ft53.atsp, ftv70.atsp, ch150.tsp, ftv170.atsp, gr202.tsp, gr666.tsp, rgb323.atsp, pcb442.tsp, rgb443.atsp, pr1002.tsp, pr2392.tsp.

Badanie wykonać dziesięciokrotnie dla każdej instancji. Wyniki uśrednić. Czas badania pojedynczej instancji ograniczyć do 10 minut.

Rozwiązanie na **4,0** obejmuje dodatkowo zastosowanie:

- co najmniej dwóch metod selekcji,

- co najmniej dwóch metod krzyżowania.

W przypadku metod selekcji i krzyżowania należy pokazać różnice oraz wyjaśnić powody ich występowania.

Rozwiązanie na **5,0** obejmuje oprócz ww., zastosowanie i zbadanie:

- wpływu co najmniej dwóch metod mutacji,
- co najmniej dwóch metod sukcesji.

Dodatkowo oceniane będzie porównanie jakości uzyskiwanego rozwiązania (czas, pamięć i wielkość błędu) z dotychczas zbadanymi metodami.

Zadanie 7

([spis treści](#))

Opracowanie i implementacja algorytmu w oparciu o (metodę) *Algorytm mrówkowy*

Zadanie polega na opracowaniu i implementacji algorytmu rozwiązywania problemu komiwojażera przy użyciu algorytmu wykorzystującego (metodę) *Algorytmy mrówkowe*.

Rozwiązanie na **3,0** obejmuje opracowanie i implementację algorytmu ze stałymi podanymi na wykładzie (patrz: str. 45) wartościami parametrów α , β , ρ , m , τ_0 i heurystykę *visibility* oraz jednym schematem rozkładu feromonu (DAS, QAS, bądź CAS). Wielkość błędu bezwzględnego, w zależności od wielkości instancji, nie może przekraczać:

- dla $n < 25$, 0%,
- dla $24 < n < 350$, 50%,
- dla $75 < n < 2500$, 150%.

Rozwiązanie na **4,0** obejmuje zbadanie jakości rozwiązania (wartość funkcji celu oraz czas) w zależności od rodzaju dwóch schematów rozkładu feromonu oraz udowodnienie prawdziwości tez dotyczących wpływu parametrów α i β na zachowanie się algorytmu (patrz: wykład str. 29). Wielkość błędu jak wyżej.

Rozwiązanie na **5,0** (i więcej) obejmuje zbadanie wpływu parametrów: α , β , ρ , każdego schematu rozkładu feromonu oraz wymyślenie heurystyki wyboru i zbadanie jej wpływu na jakość rozwiązania, włącznie z porównaniem do *visibility*.

Zestaw instancji jak w zadaniu 6.

Dane testowe

([spis treści](#))

Dane do testów dostępne są na stronie dra Jarosława Mierzwę

<http://jaroslaw.mierzwa.staff.iiar.pwr.wroc.pl/pea-stud/tsp/>

oraz

Dla **asymetrycznego** problemu komiwojażera z większą liczbą miast na stronie:

<http://elib.zib.de/pub/mp-testdata/tsp/tsplib/atsp/index.html>

Te same przykłady w postaci spakowanej (należy pobrać plik sourcesAsymmetricTSP.zip) na stronie:

<http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/XML-TSPLIB/instances/>

Najlepsze znane rozwiązania dla ww. przykładów na stronach:

<http://elib.zib.de/pub/mp-testdata/tsp/tsplib/atsp-sol.html>

<http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/ATSP.html>

Dla **symetrycznego** problemu komiwojażera z większą liczbą miast na stronie:

<http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/index.html>

Opis formatu danych znajduje się w pliku tsp95.pdf:

<http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/tsp95.pdf>

Pliki testowe zawierają specjalne nagłówki, w których zawarta jest między innymi liczba miast. Dodatkowo w niektórych plikach wiersze macierzy odległości rozdzielone są na kilka linii. Można wykonać ręczną edycję plików, zmodyfikować procedurę wczytywania lub skorzystać z pomocniczego programu do konwersji:

[Program do konwersji danych dla problemu TSP](#)

Działanie programów

[\(spis treści\)](#)

Program uruchamiany jest w oparciu o informacje zawarte w pliku inicjującym (.INI) zawierającym parametry (częściowo różne dla każdego z zadań), sterujące działaniem programu. Parametrami wspólnymi dla wszystkich zadań są: nazwy instancji, które będą rozwiązywane, liczba powtórzeń wykonania programu dla każdej instancji, wartości optymalne oraz ścieżki optymalne dla każdej instancji, nazwa pliku wyjściowego.

przykładowa zawartość pliku .INI

```
br17.atsp 50 39 [kolejne węzły ścieżki]
ftv170.atsp 70 2755 [kolejne węzły ścieżki]
p43.atsp 50 5620 [kolejne węzły ścieżki]
test_atsp_out.csv
```

Plik wynikowy (tu: test_atsp_out.csv) ma zawierać powyższe informacje oraz wszystkie zmierzone czasy wykonania, koszt ścieżki oraz kolejne węzły ścieżki dla każdej instancji. [Zapamiętywanie kosztu oraz ścieżki jest szczególnie istotne w zadaniach 4, 5, 6 i 7] Plik ma być w formacie .csv. Umożliwi to łatwe przeglądanie wyników oraz obróbkę i analizę wyników.

przykładowa zawartość pliku wynikowego .csv

```
br17.atsp 50 39 [kolejne węzły ścieżki]
czas_1
czas_2
...
ftv170.atsp 70 2755 [kolejne węzły ścieżki]
czas_1
czas_2
...
p43.atsp 50 5620 [kolejne węzły ścieżki]
czas_1
czas_2
...
test_atsp_out.csv
```

Schemat raportu

[\(spis treści\)](#)

[dodany w oddzielnym pliku na dysku Google]

Oceny

[\(spis treści\)](#)

Zadania: 1, 2 lub 3, 4 lub 5, 6 lub 7 + sprawozdanie : 3,5
(możliwe 4,0 w przypadku wyjątkowości programów bądź sprawozdania)

Zadania: 1, 2, 3, 4 lub 5, 6 lub 7 + sprawozdanie : 4,0
(możliwe 4,5 w przypadku wyjątkowości programów bądź sprawozdania)

Zadania: 1, 2 lub 3, 4, 5, 6 lub 7 + sprawozdanie : 4,5
(możliwe 5,0 w przypadku wyjątkowości programów bądź sprawozdania)

Zadania: 1, 2, 3, 4, 5, 6 lub 7 + sprawozdanie : 5,0
(możliwe 5,5 w przypadku wyjątkowości programów bądź sprawozdania)

Zadania: 1, 2, 3, 4, 5, 6, 7 + sprawozdanie : 5,5
(ale sprawozdanie musi być naprawdę dobre)

Terminy realizacji[\(spis treści\)](#)

	grupy wtorkowe	grupa środowa	grupy piątkowe
zadanie(-a)	data	data	data
1	18.10.2022	19.10.2022	21.10.2022
2, 3	15.11.2022	16.11.2022	18.11.2022
4, 5	20.12.2022	21.12.2022	23.12.2022
6, 7	17.01.2023	18.01.2023	20.01.2023

Opóźnienie przekazania zadania skutkować będzie obniżeniem oceny z zadania o pół (0,5) stopnia od oceny uzyskanej z zadania, za każdy dzień opóźnienia.

Terminy spotkań obowiązkowych, przekazania raportów i programów[\(spis treści\)](#)

grupy wtorkowe	grupa środowa	grupy piątkowe
data	data	data
18.10.2022	19.10.2022	21.10.2022
08.11.2022	02.11.2022	04.11.2022
15.11.2022	16.11.2022	18.11.2022
29.11.2022	30.11.2022	02.12.2022
20.12.2022	21.12.2022	16.12.2022
10.01.2023	11.01.2023	13.01.2023
17.01.2023	18.01.2023	20.01.2023

Uwagi do raportów

([spis treści](#))

W punkcie "Procedura badawcza" proszę umieszczać:

- specyfikację sprzętu, na którym przeprowadzane są badania,
- opis metody pomiaru czasu uzyskiwania rozwiązania instancji; ma pojawić się opis metody pomiaru czasu oraz odpowiedni fragment kodu programu.

Wszelkie uwagi i wątpliwości proszę zgłaszać drogą mailową bądź w trakcie spotkań