Wrocław, 3 października 2019

Projektowanie **E**fektywnych **A**lgorytmów - Projekt
Założenia projektu

prowadzący: *mgr inż. Radosław Idzikowski*

1 Informacje ogólne

W ramach zajęć należy zaimplementować oraz dokonać analizy efektywności algorytmów dla problemu komiwojażera(*lub innego wcześniej ustalonego*). W celu zaliczenia kursu należy zrealizować 4 etapy:

Etap 0: *Wczytanie instancji, liczenie funkcji celu*

Etap 1: *Algorytm dokładny,*

Etap 2: *Algorytm poszukiwania lokalnego,*

Etap 3: *Algorytm populacyjny.*

Aby uzyskać końcową ocenę pozytywną:

- ze wszystkich etapów trzeba uzyskać co najmniej ocenę *3.0* (*dopuszczalna jest 1 ocena warunkowa 2.5*),
- kody programów trzeba oddać osobiście i zaprezentować najpóźniej w terminie wyznaczonym dla każdej grupy. Za każdy tydzień spóźnienia obniżana jest ocena o *0.5* (*można przyjść przed upływem terminu*),
- sprawozdanie należy wysłać tytułem "[PEA][*DDGG*] Sprawozdanie" w ciągu tygodnia od oddania kodu (*nie wysyłać sprawozdania jeśli program nie został zaakceptowany*), gdzie *DD* dzień tygodnia, *GG* godzina,
- $O_k = 0.1C_0 + 0.2C_1 + 0.1R_1 + 0.2C_2 + 0.1R_2 + 0.2C_3 + 0.1R_3$, gdzie C_i ocena za kod z etapu i-tego, R_i ocena za sprawozdanie z etapu i-tego,

2 Sprawozdanie

Sprawozdanie powinno zawierać:

- opis badanego problemu,
- opis algorytmu lub algorytmów,
- szczegóły odnośnie zaimplementowanego algorytmu (np. użyty operator krzyżowania, lista tabu, parametry algorytmu),
- wyniki z analizą wydajności algorytmu,
- wnioski.

oraz jeśli dotyczy:

- wyniki z analizą jakości dostarczanych rozwiązań,
- wpływ najważniejszych parametrów (jak liczba iteracji, liczba osobników, długość listy tabu) na jakość wyników.

UWAGA! Sprawozdanie musi być w formacie **PDF** (*preferowany* LaTeX).

3 Szczegóły etapów

3.1 Algorytm dokładny

- przegląd zupełny (***B**rute **F**orce*),
- algorytm podziału i ograniczeń (***B**ranch and **B**ound*),
- programowanie dynamiczne (***D**ynamic **P**rogramming*).

3.2 Algorytm poszukiwania lokalnego

- poszukiwanie z zakazami (***T**abu **S**earch*),
- symulowane wyżarzanie (***S**imulated **A**nnealing*),
- inny po wcześniejszej konsultacji.

3.3 Algorytm populacyjny

- algorytm genetyczny (***G**enetic **A**lgorithm*),
- algorytm mrówkowy (***A**nt **C**olony **O**ptimization*),
- inny po wcześniejszej konsultacji.

4 Aspekty techniczne

Podczas realizacji zadania należy przyjąć następujące założenia:

- preferowany język C++,
- implementacji algorytmu należy dokonać zgodnie z obiektowym paradygmatem programowania,
- program ma umożliwić weryfikację poprawności działania algorytmu. W tym celu trzeba zapewnić możliwość wczytania danych wejściowych z pliku tekstowego,
- w celu ułatwienia dalszych badań, program powinien mieć możliwość testowania automatycznego wielu instancji,
- po zaimplementowaniu i sprawdzeniu poprawności działania algorytmu należy dokonać pomiaru czasu jego działania w zależności od rozmiaru problemu dla załączonych danych wejściowych,
- używanie „okienek" nie jest konieczne i nie wpływa na ocenę.

5 Format danych wejściowych

name
n

$$\begin{array}{ccccc} -1 & d_{12} & d_{13} & \dots & d_{1n} \\ d_{21} & -1 & d_{23} & \dots & d_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & d_{n3} & \dots & -1 \end{array}$$

gdzie:

- *name* – nazwa instacji,
- n – rozmiar problemu,
- d_{ij} – odległość z miasta i do miasta j.

6 Terminy

	Terminy						
	etap 0	etap 1		etap 2		etap 3	
Grupa	C_0	C_1	R_1	C_2	R_2	C_3	R_3
CZ17	17.10.2019	14.11.2019	21.11.2019	12.12.2019	19.12.2019	23.01.2020	30.01.2020
CZ19	17.10.2019	14.11.2019	21.11.2019	12.12.2019	19.12.2019	23.01.2020	30.01.2020
PT15	18.10.2019	15.11.2019	22.11.2019	13.12.2019	20.12.2019	24.01.2020	31.01.2020
PN15	21.10.2019	13.11.2019	28.11.2019	09.12.2019	16.12.2019	20.01.2020	27.01.2020
PN17	21.10.2019	13.11.2019	28.11.2019	09.12.2019	16.12.2019	20.01.2020	27.01.2020