

Projektowanie Efektywnych Algorytmów – projekt – wersja 1.0 (niekoniecznie ostateczna)

Zasady zaliczenia

Oceniane będą:

- a) Czy program się uruchamia, pozwala na wprowadzenie danych, wykonuje określone zadanie (rozwiązuje TSP), wyświetla wyniki? ocena T/N,
- b) Czy zaimplementowano wszystkie wymagane metody? ocena T/N,
- c) Czy został zaprezentowany na zajęciach? ocena T/N,
- d) Czy dostarczony został raport? ocena T/N; dotyczy grup zadań 1) i 2) oraz 3) i 4),
- e) Czy zostały zbadane instancje symetryczne i asymetryczne, czy tylko jedno z nich?

Warunki a) do d), to warunki dopuszczające do dalszej oceny. Spełnione muszą być wszystkie. Zbadanie tylko jednego rodzaju instancji – warunek e – i w konsekwencji brak odpowiedzi na pytanie, czy rodzaj instancji (sym. – asym.) ma, i jeśli tak, to jaki na czas i jakość wyniku, oznacza obniżenie oceny z zadania o 0,5 stopnia.

- f) Poprawność implementacji, czyli czy zaimplementowano metody określone w zadaniu. Poprawność i sposób prezentacji wyników (patrz: Co ma pojawić się na ekranie?). Sposób wprowadzania danych oraz sposób zmiany parametrów metody (patrz: Jak powinien wyglądać plik konfiguracyjny?) Ocena w skali 0 – 10.
- g) Znajomość kodu programu oraz implementowanych metod i treści raportu (dotyczy: grup zadań 1) i 2) oraz 3) i 4)). Odpowiedź, „nie pamiętam, robiłam/ -em program dwa tygodnie temu” przy okazji np. pytania o przebieg algorytmu czy nazwy parametrów, oznaczać będzie 0 punktów i niezaliczenie zadania. Ocena w skali 0 – 10.
- h) Treść raportu. W tym ocena formy (patrz: Jak powinien wyglądać raport?), ocena poprawności merytorycznej, ocena doboru oraz czytelności wykresów i tabel, ocena edycji dokumentu, ocena postawionych tez badawczych, opracowanej procedury badawczej, wykonanych badań, ocena analizy wyników oraz wniosków, ocena porównania wszystkich metod i odpowiedzi na pytanie... kiedy, każdą z badanych metod można uznać za efektywną?; ocena w skali 0 – 25.

Przeliczenie punktów na ocenę z grup projektów odbywa się wg przelicznika standardowego (od 50% - 3,0; od 60% - 3,5; itd. Ocena końcowa z kursu (projekt) to średnia ważona z obu grup zadań w proporcji 4/6, czyli $0,4 \cdot \text{ocena z 1) i 2)}$ oraz $0,6 \cdot \text{ocena z 3) i 4)}$. Widełki średnich: 3,0 : 3,0 – 3,3; 3,5 : 3,4 – 3,7; 4,0 : 3,8 – 4,2; 4,5 : 4,3 – 4,6; 5,0 : 4,7 – 5,0.

Zadania

1. *Brute-force, nearest-neighbour, random*
Rozwiązanie problemu komiwojażera (TSP – *Travelling Salesman Problem*) ww. metodami. Rozwiązania muszą spełniać warunki zaliczenia od a) do g), i w tym zakresie będą oceniane. To zadania ma na celu nabycie podstawowych informacji o TSP, sposobie rozwiązania tego problemu, sposobie przygotowania i przeprowadzenia eksperymentu badawczego, prezentacji oraz analizy wyników (wraz z wnioskami).
2. Zastosowanie metody podziału i ograniczeń (*Branch and Bound*) do rozwiązania TSP. Należy zastosować trzy metody przeszukiwania przestrzeni rozwiązań. Do wyboru są (na przykład) przeszukiwanie: po szerokości (*Breadth first search*), po głębokości (*Depth first search*) i przy minimum kosztów (*Lowest cost / Best first search*) bądź *IDA**. Dozwolone jest użycie innych

metod przeszukiwania przestrzeni rozwiązań, o ile będą to metody opisane w literaturze (metoda, realizacja i badania z wynikami). Żadnych własnych pomysłów i udoskonalień.

3. Rozwiązanie TSP metodami *Tabu Search* (przeszukiwanie z zakazami) i/lub symulowanego wyżarzania (*Simulated Annealing*).
4. Zastosowanie algorytmów genetycznych bądź algorytmów mrówkowych do rozwiązywania TSP.

Szczegóły zostaną podane na zajęciach i/lub jako uzupełnienie tego dokumentu.

Dane do testowania poprawności działania programów mogą pochodzić ze strony dra. J. Mierzwy, mogą być danymi wygenerowanymi losowo bądź pochodzić z TSPLIB (<http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>).

Dane do badań mogą pochodzić ze strony dra. J. Mierzwy (jaroslaw.mierzwa.staff.iiar.pwr.wroc.pl), mogą być danymi generowanymi losowo oraz z TSPLIB (zadania 1. i 2.) oraz TSPLIB (zadania 3. i 4.).

Co ma pojawić się na ekranie?

Program w wersji konsolowej ma po uruchomieniu czytać plik konfiguracyjny. W trakcie pracy może, choć nie musi – a służyć to ma jedynie kontroli działania, wyświetlać wskaźnik postępu realizacji działania. Ponieważ wyświetlanie może mieć wpływ na czas wykonywania zadania, opcję tę powinno się dać włączyć i wyłączyć w ustawieniach w pliku konfiguracyjnym. Program musi wyświetlać, dla każdej badanej instancji: nazwę pliku z danymi, wynik optymalny (z TSPLIB), ścieżka optymalna, jeśli jest znana (wystarczy jedna, bo może być ich wiele), parametry metody, czas realizacji, wynik dla rozwiązania, ścieżkę dla rozwiązania (dla instancji <16), błąd względny i bezwzględny (x2, bo w procentach też) (dotyczy zadań 3. i 4.). układ napisów powinien być czytelny, np. konieczne są jednostki przy wartości czasu.

Program w wersji graficznej ma pokazywać co najmniej to samo.

Każde zadanie to jeden program. Proszę nie łączyć ich ze sobą. To nie ma być BMW X6M – samochód uniwersalny, czyli do niczego. Ani terenówka, ani SUV, ani auto sportowe.

Jak powinien wyglądać plik konfiguracyjny?

Plik konfiguracyjny to plik z parametrami wejściowymi i sterującymi procesem realizacji zadania. Ma być plikiem tekstowym, aby można było bez problemów zmieniać wartości parametrów algorytmu (metody). Wynika stąd wniosek, że żadne parametry mające znaczenie na przebieg procedury badawczej (patrz: Jak powinna wyglądać procedura badawcza?), nie mogą być dostępne jedynie w kodzie programu. Do tych parametrów należą: nazwy plików wejściowych i wyjściowych, wartości parametrów metody, parametry uruchamiania programu, np. liczba powtórzeń wykonania poszukiwania rozwiązania dla pojedynczej instancji problemu, wartość optymalna (z TSPLIB), warunki zatrzymania, np. czas działania czy liczba iteracji bądź kroków algorytmu. Wszelkie inne przełączniki sterujące działaniem programu, np. włączanie i wyłączanie wyświetlania wskaźnika postępu (Co ma pojawić się na ekranie?).

Zero menu i zero wprowadzania danych z klawiatury podczas działania programu.

Liczba wymagań dotyczących zawartości pliku konfiguracyjnego może się powiększyć. Zostanie to powiedziane, kiedy okoliczność taka nastąpi.

Jak powinien wyglądać plik wyjściowy?

W pliku wyjściowym bądź plikach wyjściowych, czyli plikach z wynikami działania programu (xls), muszą znajdować się: nazwa badanej instancji (nazwa pliku z danymi), wynik optymalny (z TSPLIB bądź innego wiarygodnego źródła, czyli od dra. J. Mierzwę), ścieżka optymalna (z TSPLIB, jeśli jest podana; czasy dla każdej próby wykonania (tożsamy z liczbą powtórzeń wykonania poszukiwania rozwiązania dla pojedynczej instancji z pliku konfiguracyjnego), średni czas wyznaczenia rozwiązania, błąd względny i bezwzględny (x2, bo w procentach też).

Liczba wymagań dotyczących zawartości pliku konfiguracyjnego może się powiększyć. Zostanie to powiedziane, kiedy okoliczność taka nastąpi.

Jak powinien wyglądać plik z danymi?

Plik z danymi ma być plikiem tekstowym (.txt, .xls), czyli o swobodnym dostępie do danych. Ma zawierać instancję problemu (najczęściej w postaci macierzy), wartość optymalną rozwiązania oraz ścieżkę rozwiązania (cykl Hamiltona), jeśli jest ona dostępna. Może zawierać również rozmiar instancji.

Jak powinien wyglądać raport?

Strona tytułowa: dane autora (imię, nazwisko, numer indeksu), datę złożenia, nazwę przedmiotu, temat zadania, treść zadania.

Punkt 2: Opis problemu (tu: TSP), opis zadania, czyli co jest do zbadania, jakie wyniki chcemy uzyskać, skąd pochodzą dane i w jakiej formie występują, ile ich jest (liczba instancji), rodzaj (symetryczne / asymetryczne), uzasadnienie wyboru oraz liczby instancji, Hipoteza bądź hipotezy badawcze, czyli co autor (badacz) chce potwierdzić (sfalsyfikować) w kontekście celu nadrzędnego, którym jest badanie efektywności czasowej i pamięciowej oraz określenie warunków uznania metody za efektywną – łączenie z odniesieniem do znanych oszacowań złożoności obliczeniowej dla określonych metod, intuicji badacza (opartej o wiedzę o problemie i własne doświadczenia) oraz wyników badań uzyskanych dla innych metod.

Punkt 3: Lista instancji wraz z wynikami optymalnymi oraz ścieżkami (jeśli są znane), specyfikacja sprzętu: procesor, zegar, wielkość pamięci, czy badania prowadzone były na sprzęcie, na którym są prezentowane, opis warunków badania, czyli czy komputer pracował na zasilaniu sieciowym czy na baterii, ile i jakie inne programy działały na komputerze w trakcie badania. Tu mała uwaga. Nie powinny działać żadne inne w trakcie wykonywania badań.

Punkt 4: Opis procedury badawczej: ile instancji było badanych, ile razy każda i dlaczego? Czy badania wykonane zostały na instancjach symetrycznych i asymetrycznych? Jeżeli na obu, to wyniki muszą być prezentowane oddzielnie i powinien się pojawić wykres z zestawieniem wyników – jeden bądź wiele, w zależności od liczby użytych parametrów metody. Jak były traktowane anomalie, błędy grube, systematyczne i przypadkowe? Z czego wynikały? Czy średnie czasów i zajętości pamięci liczone były dla wszystkich prób, czy może część z nich (błędy, anomalie), były pomijane. Jeśli tak, to dlaczego? Każde twierdzenie i założenie należy uzasadnić. Dlaczego, z punktu widzenia hipotez, właśnie w ten sposób zaprojektowano procedurę badawczą? Tu znowu uzasadnienie.

Punkt 5. Wyniki, czyli tabele i wykresy. Tabele mają numery i opisy nad (np. Tabela 1. Średnie czasy uzyskiwania wyników dla instancji bla_bla.sym.), a wykresy pod sobą (np. Rysunek 23. Wykres

zależności czasu rozwiązania TSP metodą brute-force dla wszystkich badanych instancji..). Numery i opisy w tabelach wyrównywane są do lewej, pod wykresami do środka. We wnioskach należy odwoływać się do tabel i rysunków podając ich numery. Fragmenty programów (listingi) należy traktować jak rysunki. Liczba tabel i rysunków jest ograniczona jedynie przez Waszą wyobraźnię, a tak naprawdę, przez potrzebę wynikającą z tego, co chcecie pokazać, czyli co mieliście (cel nadrzędny badań) i chcieliście udowodnić (Wasze hipotezy badawcze).

O tym, jak mają wyglądać i co zawierać tabele i wyniki opowiem więcej na zajęciach.

Punkt 6. Analiza wyników oraz odniesienie do postawionych hipotez badawczych.

Każdy punkt należy rozpoczynać od nowej strony.

Pierwsze sprawozdanie ma być wydrukowane. Może być dwustronnie.

Mają powstać dwa raporty. Jeden do zadań 1. i 2., drugi do pozostałych.

Terminy spotkań

środa	czwartek	piątek	
02/10	03/10	04/10	zajęcia wstępne – obowiązkowe
09/10	10/10	11/10	zajęcia konsultacyjne – nieobowiązkowe
16/10	17/10	18/10	oddanie zadania 1) – obowiązkowe
23/10	24/10	25/10	zajęcia konsultacyjne – nieobowiązkowe
30/10	–	–	zajęcia konsultacyjne – nieobowiązkowe
06/11	07/11	08/11	zajęcia konsultacyjne – obowiązkowe
13/11	14/11	–	zajęcia konsultacyjne – nieobowiązkowe
20/11	21/11	22/11	oddanie zadania 2) – obowiązkowe
27/11	28/11	29/11	zajęcia konsultacyjne – nieobowiązkowe
04/12	05/12	06/12	zajęcia konsultacyjne – obowiązkowe
–	–	11/12	zajęcia konsultacyjne – nieobowiązkowe
–	12/12	13/12	zajęcia konsultacyjne – nieobowiązkowe
18/12	19/12	20/12	oddanie zadania 3) – obowiązkowe
08/01	09/01	10/01	zajęcia konsultacyjne – obowiązkowe
15/01	16/01	17/01	zajęcia konsultacyjne – nieobowiązkowe
22/01	23/01	24/01	oddanie zadania 4) – obowiązkowe
29/01	30/01	03/02	szansa na oddanie zaległości

Zadania można oddawać wcześniej niż wg powyższego terminarza. Studenci siódmego semestru chcący zakończyć semestr w 10 tygodniu muszą oddać wszystkie zadania najpóźniej do 13/12/24.

Zadania można oddawać tylko w trakcie swoich zajęć. Opóźnienie przesłania zadania na ePortal skutkuje, za każdy rozpoczęty dzień opóźnienia, obniżeniem oceny o 0,5 stopnia za zadanie. Dotyczy to również studentów siódmego semestru.

Rozwiązania zadań należy umieszczać na ePortalu. Mają się tam znaleźć co najwyżej dwa pliki: pdf z raportem oraz zip z kodem programu, danymi, plikiem konfiguracyjnym, programem w wersji wykonywalnej (.exe), ewentualnie niezbędnymi do uruchomienia exe, dodatkami. Program (.exe) ma działać w środowisku Windows bez konieczności posiadania, a tym bardziej instalowania, czegokolwiek.

W przypadku wątpliwości proszę pytać.