

PEA – projekt nr 2

Temat: Implementacja i analiza efektywności algorytmu Tabu Search dla wybranego problemu optymalizacji

Należy zaimplementować oraz dokonać analizy efektywności algorytmu **Tabu Search** dla wybranego problemu z poprzednich zajęć tj: problemu komiwojażera (TSP) lub problemu jednoprocessorowego szeregowania zadań przy kryterium minimalizacji ważonej sumy opóźnień zadań (SMTWT).

Wskazane jest, aby algorytm uwzględniał mechanizm dywersyfikacji przeszukiwania przestrzeni rozwiązań (powroty, ruchy losowe, itp.) jak również różne definicje sąsiedztwa (typy ruchów). Należy porównać rozwiązanie dostarczone przez algorytm z najlepszymi znanymi rozwiązaniami dla przykładów testowych.

Podczas realizacji zadania należy przyjąć następujące założenia:

- używane struktury danych powinny być alokowane dynamicznie (w zależności od aktualnego rozmiaru problemu),
- program powinien umożliwić wczytanie danych testowych z pliku -źródła danych znajdują się:
dla TSP: <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>
(zwrócić uwagę na formaty plików dla wersji symetrycznej (euklidesowej) i asymetrycznej – program powinien wczytać poprawnie oba formaty)
dla SMTWT: <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/wtinfo.html>
- program musi umożliwiać wprowadzenia kryterium stopu jako czasu wykonania podawanego w sekundach
- program musi umożliwiać włączenie/wyłączenie mechanizmu dywersyfikacji oraz wybór typu sąsiedztwa jeśli zaimplementowano
- implementacje algorytmów należy dokonać zgodnie z obiektowym paradygmatem programowania,
- używanie „okienek” nie jest konieczne i nie wpływa na ocenę (wystarczy wersja konsolowa),
- kod źródłowy powinien być komentowany.

Menu programu powinno zawierać następujące opcje:

- 1.Wczytanie danych z pliku
- 2.Wprowadzenie kryterium stopu
- 3Włączenie/Wyłączenie dywersyfikacji
- 4.Wybór sąsiedztwa
- 5.Uruchomienie algorytmu dla wczytanych danych i ustawionych parametrów i wyświetlenie wyników

UWAGI

Razu ustawione parametry obowiązują do ich zmiany (chodzi o to, aby nie ustawiać ich ponownie przy wczytywaniu nowych danych jak i nie wczytywać ponownie danych przy zmianie parametrów)

Sprawozdanie powinno zawierać:

- a) wstęp teoretyczny zawierający opis algorytmu, omówienie jego elementów (lista tabu, dywersyfikacja, opis (definicja) sąsiedztwa itd.)
- b) opis najważniejszych klas w projekcie
- c) dane w postaci tabel i wykresów - dla wybranych 3 różnych danych testowych (o różniącym się wyraźnie rozmiarze danych) wykonać pomiary błędu względnego (wyrażonego go w %) w funkcji czasu wykonywania algorytmu
- d) jeżeli zastosowano dywersyfikację, to pokazać na w/w wykresach algorytm z dywersyfikacją i bez
- e) w przypadku zdefiniowania różnych typów sąsiedztwa dla każdego sąsiedztwa narysować wykres jak w punkcie c i d

Błąd względny to $|f_{zn} - f_{opt}|/f_{opt}$, gdzie f_{zn} – wartość obliczona przez nasz algorytm, f_{opt} – wartość optymalna – najlepsze znane rozwiązanie

Ocena:

- a) prosty algorytm bez dywersyfikacji z jedną definicją sąsiedztwa – 4.0
- b) algorytm z dywersyfikacją i jedną definicją sąsiedztwa – 4.5
- c) algorytm bez dywersyfikacji i trzema definicjami sąsiedztwa – 4.5
- d) algorytm z dywersyfikacją i trzema definicjami sąsiedztwa – 5.0

Dodatkowe materiały internetowe:

<http://cs.pwr.edu.pl/zielinski/lectures/om/localsearch.pdf>

http://www.cs.put.poznan.pl/mrdom/resources/labs/OptKomb/dziamski_4.pdf

<http://www2.imm.dtu.dk/courses/02719/tabu/4tabu2.pdf>

http://riad.pk.edu.pl/~zk/MP_W8.pdf

http://www.pi.zarz.agh.edu.pl/intObl/notes/IntObl_w2.pdf

https://file.scirp.org/pdf/AJOR20120200002_63598589.pdf