

Zadanie projektowe nr 2

Implementacja i analiza efektywności algorytmu Tabu Search i symulowanego wyżarzania dla wybranego problemu optymalizacji

Należy zaimplementować oraz dokonać analizy efektywności algorytmów Tabu Search (TS) i symulowanego wyżarzania (SA) dla jednego z następujących problemów (powinien to być ten sam problem, który został wybrany w zadaniu projektowym 1):

- a) problemu komiwojażera (TSP),
- b) jednoprosesorowego problemu szeregowania zadań przy kryterium minimalizacji ważonej sumy opóźnień zadań.

Zestaw algorytmów które powinny być zaimplementowane oraz ewentualne dodatkowe wymagania podane są na końcu, przy omówieniu zasad oceniania projektu.

Wskazane jest, aby algorytm Tabu Search 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.

Należy przyjąć następujące założenia:

- używane struktury danych powinny być alokowane dynamicznie (w zależności od aktualnego rozmiaru problemu), można używać STL,
- program powinien umożliwić weryfikację poprawności działania algorytmu. W tym celu powinna istnieć możliwość wczytania danych wejściowych z pliku tekstowego. Sposób uzyskania większych przykładów testowych opisany jest w dodatkowym dokumencie: http://staff.iar.pwr.wroc.pl/antoni.sterna/pea/PEA_testy.htm
- testy efektywności należy przeprowadzić dla trzech przykładów wyraźnie różniących się rozmiarem (na przykład: br17, ftv55 i ftv170),
- 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 powinny być zgodne 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,
3. Włączenie/wyłączenie dywersyfikacji,
4. Wybór sąsiedztwa,

5. Uruchomianie algorytmu dla wczytanych danych i ustawionych parametrów oraz wyświetlenie wyników i czasu wykonania.

UWAGA

Raz ustawione parametry powinny obowiązywać do ich zmiany (chodzi o to, aby nie ustawiać ich ponownie przy wczytywaniu nowych danych jak też nie wczytywać ponownie danych przy zmianie parametrów).

Sprawozdanie powinno zawierać:

- wstęp teoretyczny zawierający opis algorytmu, omówienie jego elementów (lista tabu, dywersyfikacja, definicja sąsiedztwa itd.),
- opis najważniejszych klas w projekcie,
- dane w postaci tabel i wykresów (patrz komentarz poniżej) dla trzech przykładów testowych (o wyrażnie różnej liczbie miast lub zadań),
- jeżeli zastosowano dywersyfikację, to na wykresach pokazać wyniki dla wariantu z dywersyfikacją i bez,
- w przypadku zdefiniowania różnych typów sąsiedztwa pokazać osobne wykresy dla każdego typu sąsiedztwa,
- wnioski dotyczące otrzymanych wyników,
- kod źródłowy w formie elektronicznej wraz z wersją wykonywalną programu.

Podstawowy wykres powinien pokazywać zależność błędu względnego (podanego w %) w funkcji czasu wykonywania algorytmu. Błąd względny wyrażony jest następująco:

$$|f - f_{opt}| / f_{opt}$$

gdzie:

f – wartość obliczona przez testowany algorytm

f_{opt} – wartość optymalna – najlepsze znane rozwiązanie

Ocena projektu:

- 3.0 – algorytm symulowanego wyżarzania (SA) w najprostszej wersji
- 3.5 – algorytm Tabu Search (TS) w najprostszej wersji
- 4.0 – oba algorytmy (TS i SA) w najprostszej wersji
- 5.0 – oba algorytmy, przynajmniej dwie definicje sąsiedztwa, TS z dywersyfikacją

Materiały internetowe:

http://www.zio.iar.pwr.wroc.pl/pea/w5_ts.pdf

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

http://www.cs.put.poznan.pl/mradow/teaching/laboratories/OptKomb/dziamski_4.pdf

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

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

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