Techniki optymalizacji

Łukasz Wojnarowski (80164)
Tomasz Kujawa (75909)

8 listopada 2010

Spis treści

1 Opis problemu

Rozwiązywany problem jest rozwinięciem *problemu komiwojażera* (TSP - ang. traveling salesman problem), który polega na znalezieniu 4 cykli hamiltona w pełnym grafie ważonym o minimalnej sumie wag.

Dane wejściowe składają się z grafu pełnego $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, gdzie $\mathcal{V}$ to zbiór wierzchołków (można go interpretować jako zbiór punktów na płaszczyźnie), a $\mathcal{E}$ to zbiór krawędzi. Dla każdej z krawędzi $\{v_i, v_j\}\colon v_i, v_j \in \mathcal{V}$ znana jest waga, będąca odległością pomiędzy wierzchołkami v_i, v_j. Rozwiązaniem problemu są cztery cykle proste

$$w_1, w_2, w_3, \ldots, w_n, w_1 \ ; \ x_1, x_2, x_3, \ldots, x_n, x_1 \ ; \ y_1, y_2, y_3, \ldots, y_n, y_1 \text{ oraz } z_1, z_2, z_3, \ldots, z_n, z_1, \tag{1}$$

które spełniają następujące ograniczenia:

- $w_i \in \mathcal{V}'$,
- $x_j \in \mathcal{V}''$,
- $y_k \in \mathcal{V}'''$,
- $z_l \in \mathcal{V}''''$,
- $\mathcal{V}' \cup \mathcal{V}'' \cup \mathcal{V}''' \cup \mathcal{V}'''' = \mathcal{V}$,
- $\mathcal{V}' \cap \mathcal{V}'' \cap \mathcal{V}''' \cap \mathcal{V}'''' = \varnothing$,
- $|\mathcal{V}'| = |\mathcal{V}''| = |\mathcal{V}'''| = |\mathcal{V}''''| = n$, przy założeniu, że $\mathcal{V} = 4n$.

Niech $|v_i, v_j|$ oznacza wagę (koszt) krawędzi pomiędzy wierzchoękami v_i, v_j. Dla tak zdefiniowanego modelu można określić funkcję celu w następujący sposób:

$$minC = \sum_{i<n}^{i=1} |w_i, w_{i+1}| + |w_n, w_1| + \sum_{i<n}^{i=1} |x_i, x_{i+1}| + |x_m, x_1| + \sum_{i<n}^{i=1} |y_i, y_{i+1}| + |y_m, y_1| + \sum_{i<n}^{i=1} |z_i, z_{i+1}| + |z_m, z_1| \tag{2}$$

gdzie$|\mathcal{V}| = 4n$.

2 Generowanie rozwiązania początkowego (*RP*)

2.1 Opis metody

Analizowana metoda generowania rozwiązania początkowego to *grupowanie* i następnie *poszukiwanie najbliższego sąsiada.*

2.1.1 Słowny

Metoda rozpoczyna się od losowego wybrania wierzchołka początkowego, na którego podstawie stworzone zostaną grupy. Grupy mają najpierw przydzielane z puli dostępnych wierzchołków elementy początkowe - takie, że środek ciężkości od punktów już wcześniej przydzielonych jest największy. Następnie na podstawie wybranych "liderów" budowane są grupy - tak, że każdy kolejny element dodawany do grupy będzie miał najmniejszą odległość od środka ciężkości grupy. Należy zaznaczyć, że przydział po grupach odbywa się iteracyjnie - tzn. najpierw przydzielamy jeden element do grupy pierwszej, potem jeden element do grupy drugiej i iteracyjnie aż do wyczerpania się elementów nieprzydzielonych do żadnej grupy. Przydział ten jest iteracyjnie powtarzany, aż stworzone zostaną 4 grupy o równych licznościach.

Następnie w każdej grupie następuje budowanie ścieżki tak, że przy każdym kroku wybierany jest taki wierzchołek, że jego odległość od środka ciężkości dotychczas wybranych wierzchołków jest najmniejsza. Algorytm zatrzymuje się, jeśli w grupie nie będzie już nieodwiedzonych wierzchołków. Należy pamiętać, by rozwiązanie uzupełnić o krawędź pomiędzy ostatnim a pierwszym wierzchołkiem.

2.1.2 Pseudokod

Poniżej zaprezentowano pseudokod algorytmu opisanego w części 2.1.1.

```
GENERUJ ROZWIĄZANIE POCZĄTKOWE(𝒱)
v1 ← POBIERZ LOSOWO(𝒱)
v2 ← POBIERZ NAJDALSZY(𝒱 \ {v1})
v3 ← POBIERZ NAJDALSZY(𝒱 \ {v1, v2})
v4 ← POBIERZ NAJDALSZY(𝒱 \ {v1, v2, v3})
UMIEŚĆ W PIERWSZEJ GRUPIE(v1)
UMIEŚĆ W DRUGIEJ GRUPIE(v2)
UMIEŚĆ W TRZECIEJ GRUPIE(v3)
UMIEŚĆ W CZWARTEJ GRUPIE(v4)
i ← 1
while ∃ 𝒰 = wierzchołki nieumieszczone w żadnej grupie
      do
          v ← POBIERZ NAJDALSZY(𝒰)
          UMIEŚĆ W I-TEJ GRUPIE(v)
          i = (i + 1)%5

i ← 1
while ∃ grupa z nieprzydzielonymi wierzchołkami
      do
          next ← NAJBLIŻSZY NIEODWIEDZONY WIERZCHOŁEK DLA GRUPY(i)
          DODAJ WIERZCHOŁEK GO I-TEGO CYKLU(i)
          i = (i + 1)%5
```

2.2 Wyniki

W tabeli 1 zostały przedstawione uśrednione wyniki dla opracowywanej metody.

instancja	metoda	śr. wart. rozwiązania z 20 pomiarów	śr. czas [ms]	najlepsza wartość
kroA100.txt	Grupowianie oraz wybór najbliższego sąsiada.	29878	0,25	24249

Tabela 1: Uśrednione wyniki pomiarów.

3 Local search (LS)

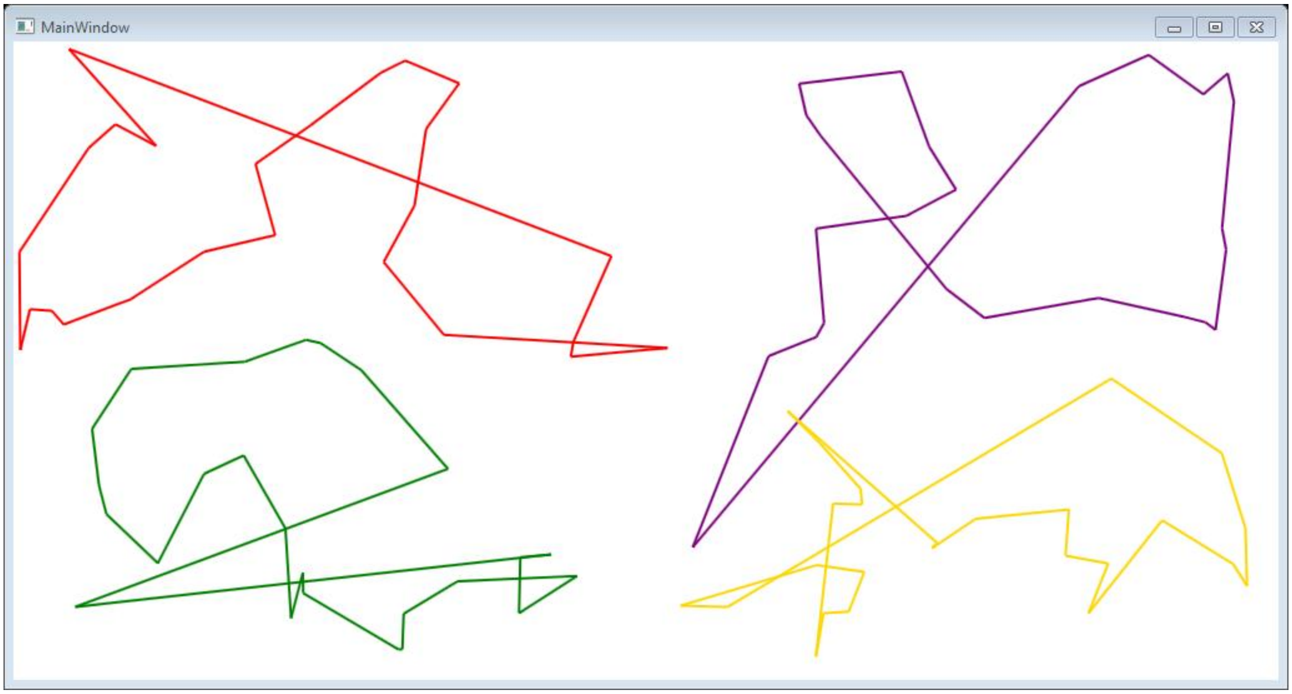
3.1 Opis metody

Analizowana metoda generowania rozwiązania to *rozrywanie (1 ruch) w wersji stromej.*

3.2 Opis słowny metody

Proces poszukiwania lokalnego optimum rozpoczyna się od wykonania kroków z opisanego w rozdziale *Generowanie rozwiązania początkowego*. Następnie na takim rozwiązaniu dokonywane jest lokalne przeszukiwanie.

Kroki metody:



Rysunek 1: Rozwiązanie początkowe dla *kroA100.txt*

1. Wybierz wierzchołek i $k-1$ mu najbliższych wierzchołków.
2. Rozerwij łuki wokół tych wierzchołków.
3. Rozważ wszystkie możliwe sposoby naprawy do rozwiazania tego problemu.
4. Wykonaj ruch, który przynosi najwięcej zysku.

Parametr $k \in 2, 3, 4$ jest definiowany na wejściu programu.