

Zadanie 2. Lokalne przeszukiwanie

Zadanie polega na implementacji lokalnego przeszukiwania w wersjach stromej (steepest) i zachłannej (greedy), z dwoma różnymi rodzajami sąsiedztwa, starując albo z rozwiązań losowych, albo z rozwiązań uzyskanych za pomocą jednej z heurystyk opracowanych w ramach poprzedniego zadania. W sumie 8 kombinacji - wersji lokalnego przeszukiwania.

Jako punkt odniesienia należy zaimplementować algorytm losowego błędzenia, który w każdej iteracji wykonuje losowo wybrany ruch (niezależnie od jego oceny) i zwraca najlepsze znalezione w ten sposób rozwiązanie. Algorytm ten powinien działać w takim samym czasie jak średnio najwolniejsza z wersji lokalnego przeszukiwania.

Sąsiedztwa

W przypadku rozważanego problemu potrzebne będą dwa typy ruchów: ruchy zmieniające zbiory wierzchołków tworzące dwa cykle i ruchy wewnątrztrasowe, które jedynie zmieniają kolejność wierzchołków na trasie.

Jako ruch zmieniający zbiór wierzchołków wykorzystujemy wymianę dwóch wierzchołków pomiędzy dwoma cyklami:



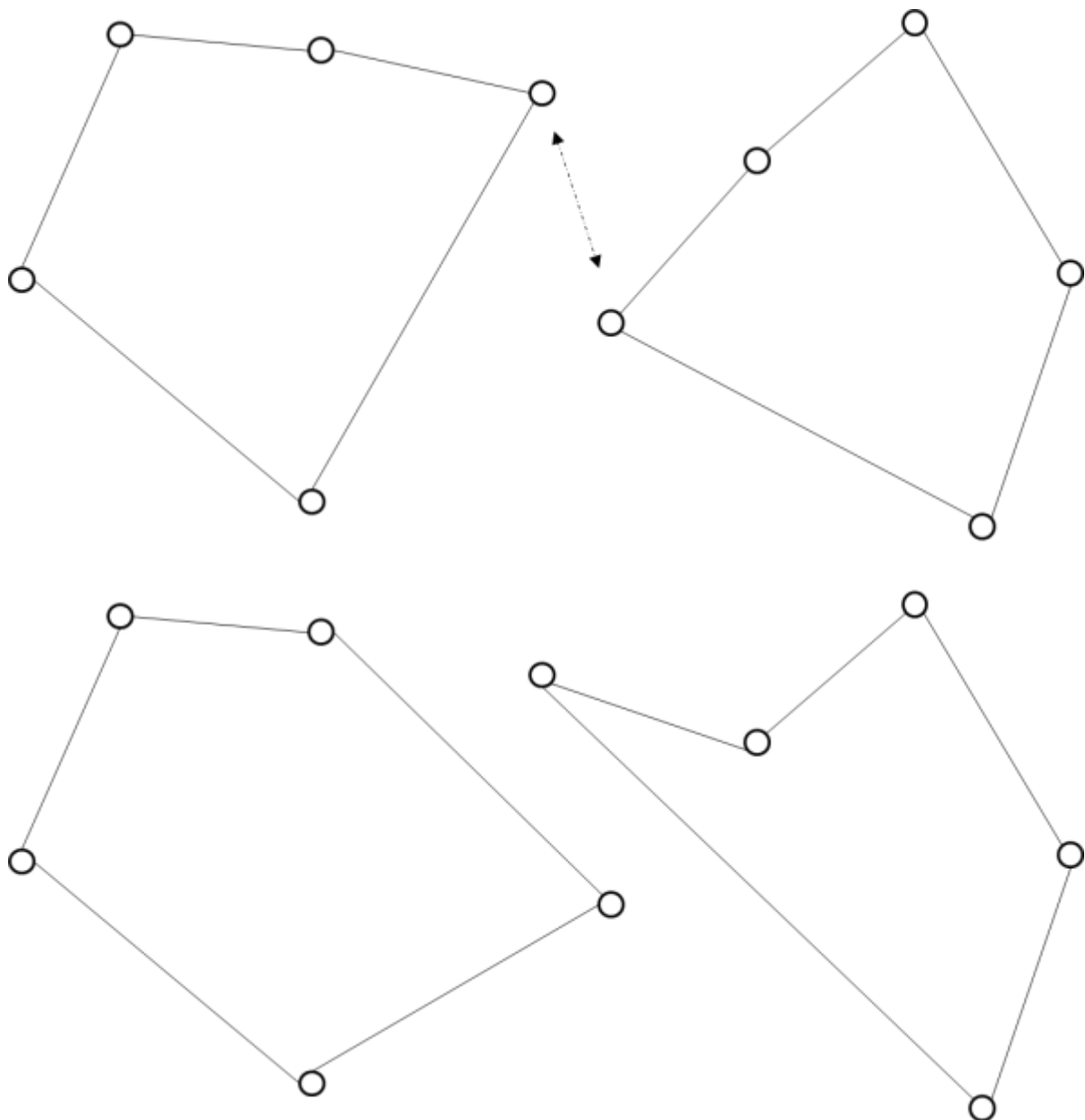
Fundusze
Europejskie
Polska Cyfrowa



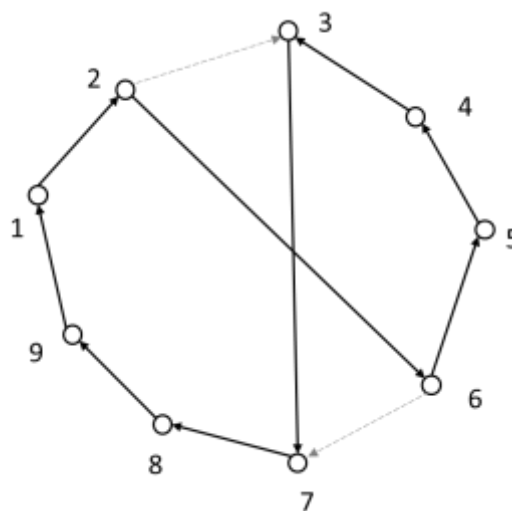
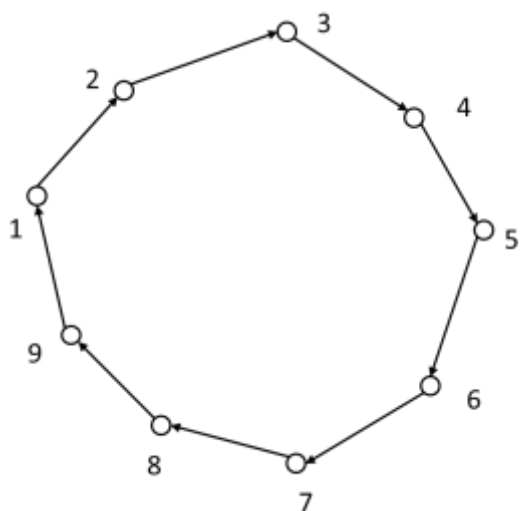
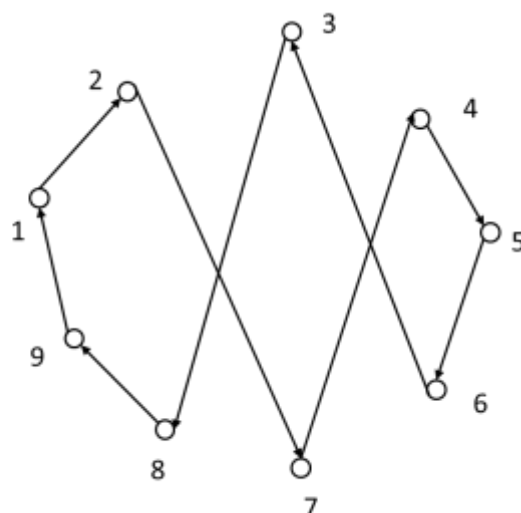
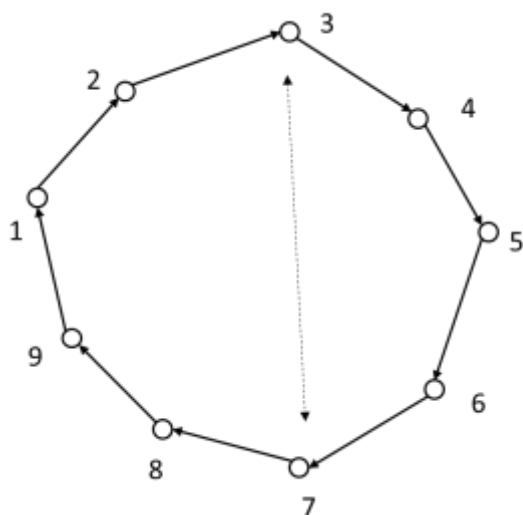
Rzeczpospolita
Polska

Unia Europejska
Europejski Fundusz
Rozwoju Regionalnego





Stosujemy dwa rodzaje ruchów wewnątrztrasowych (jeden albo drugi, stąd dwa rodzaje sąsiedztwa). jeden to wymiana dwóch wierzchołków wchodzących w skład trasy, drugi to wymiana dwóch krawędzi.



Implementacja musi wykorzystywać obliczanie delty funkcji celu.

Sąsiedztwo składa się więc z ruchów dwóch typów. W wersji stromej przeglądamy wszystkie ruchy obu typów i wybieramy najlepszy. W wersji zachłannej należy zrandomizować (kolejność nie musi być całkowicie losowa) kolejność przeglądania. W sprawozdaniu proszę opisać sposób randomizacji.

Każdy algorytm na każdej instancji uruchamiany 100 razy startując z rozwiązań losowych lub rozwiązań uzyskanych za pomocą jednej (najlepszej) z heurystyk opracowanych w poprzednim zadaniu.

Lokalne przeszukiwanie w wersji zachłannej (greedy):

Wygeneruj rozwiązanie startowe x

Powtarzaj

dla każdego $m \in M(x)$ w losowej kolejności

jeżeli $f(m(x)) > f(x)$ to

$x := y$

dopóki nie znaleziono lepszego rozwiązania po przejrzeniu całego $N(x)$



Fundusze Europejskie
Polska Cyfrowa



Rzeczpospolita
Polska

Unia Europejska
Europejski Fundusz
Rozwoju Regionalnego



Lokalne przeszukiwanie w wersji stromej (steepest):

Wygeneruj rozwiązanie startowe \mathbf{x}

powtarzaj

znajdź najlepszy ruch $m \in M(\mathbf{x})$

jeżeli $f(m(\mathbf{x})) > f(\mathbf{x})$ **to**

$\mathbf{x} := m(\mathbf{x})$

dopóki nie znaleziono lepszego rozwiązania po przejrzaniu całego $M(\mathbf{x})$

Sprawozdanie

W sprawozdaniu należy umieścić:

- Krótki opis zadania.
- Opis wszystkich zaimplementowanych algorytmów w pseudokodzie.
- Wyniki eksperymentu obliczeniowego. Dla każdej kombinacji instancja/algorytm należy podać wartość średnią, minimalną i maksymalną funkcji celu oraz, w odrębne tabeli, analogiczne dane dla czasu obliczeń. Sugerowany format tabeli jak poprzednio.
- Wizualizacje najlepszych rozwiązań dla każdej kombinacji podobnie jak poprzednio.
- Wnioski.
- Kod programu (np. w postaci linku).



Fundusze
Europejskie
Polska Cyfrowa



Rzeczpospolita
Polska

Unia Europejska
Europejski Fundusz
Rozwoju Regionalnego

