

Zadanie 3. Wykorzystanie ocen ruchów z poprzednich iteracji i ruchów kandydackich w lokalnym przeszukiwaniu

Celem zadania jest poprawa efektywności czasowej lokalnego przeszukiwania w wersji stromej (steepest) z sąsiedztwem, które okazało się lepsze w poprzednim zadaniu.

Stosujemy dwa mechanizmy poprawy efektywności:

- Wykorzystanie ocen ruchów z poprzednich iteracji z uporządkowaną listą ruchów. Na liście należy umieszczać ruchy zarówno między-, jak i wewnątrztrasowe. W przypadku ruchów wewnątrztrasowych wymiany dwóch krawędzi, należy dokładnie zapoznać się opisem z wykładów dotyczącym problemu komiwojażera.
- Ruchy kandydackie.

Mechanizmy te stosujemy oddzielnie, czyli implementujemy dwie różne wersje lokalnego przeszukiwania. Opcjonalnie można zaimplementować trzecią wersję łączącą oba te mechanizmy.

Przypomnienie z wykładu:

Wykorzystanie ocen ruchów z poprzednich iteracji

Algorytm lokalnego przeszukiwania z listą ruchów przynoszących poprawę:

Zainicjuj LM – listę ruchów przynoszących poprawę uporządkowaną od najlepszego do najgorszego

Wygeneruj rozwiązanie startowe x

powtarzaj

przejrzyj wszystkie **nowe** ruchy i dodaj do LM ruchy przynoszące poprawę

Przeglądaj ruchy m z LM od najlepszego do znalezienia aplikowalnego ruchu

Sprawdź czy m jest aplikowalny i jeżeli nie, usuń go z LM

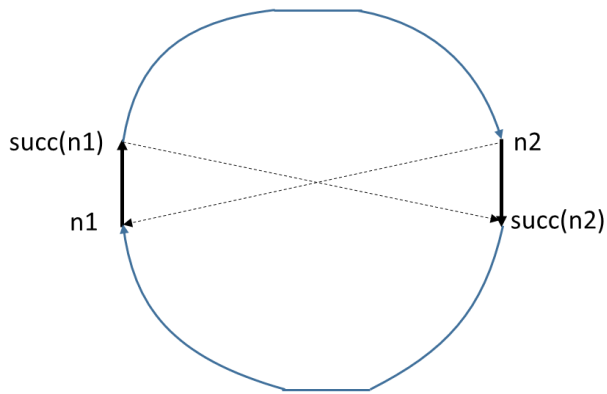
jeżeli znaleziono aplikowalny ruch m **to**

$x := m(x)$ (zaakceptuj $m(x)$)

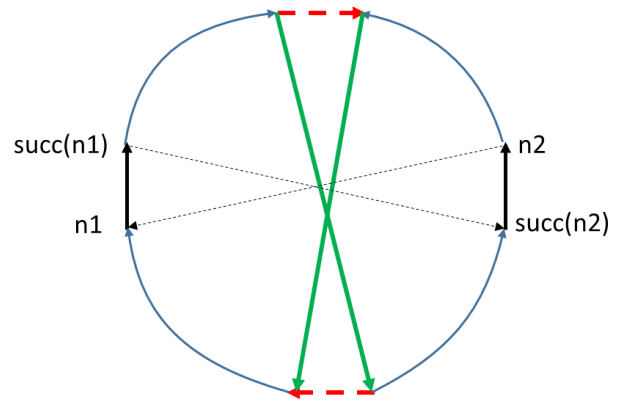
dopóki nie znaleziono ruchu aplikowalnego m po przejrzaniu całej listy LM

Przykład dla TSP

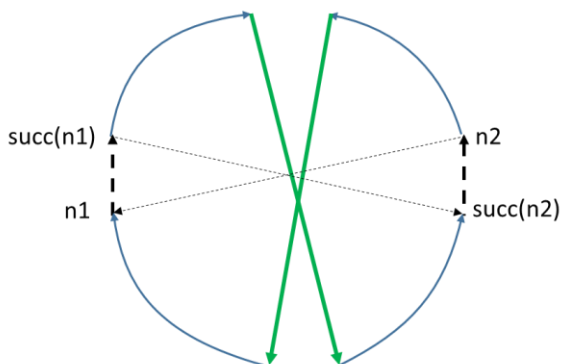
Oceniamy ruch związany z usunięciem krawędzi $(n1, succ(n1)), (n2, succ(n2))$



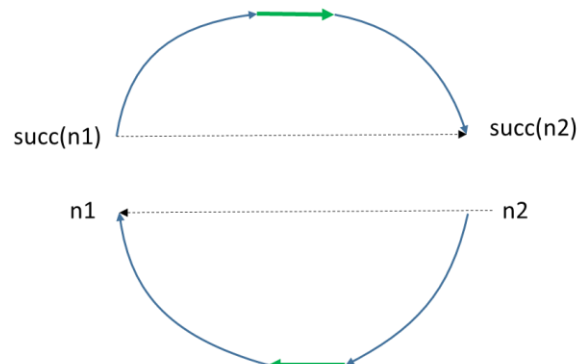
Zapamiętujemy ruch
Dodajemy krawędzie $(n1, n2)$, $(succ(n1), succ(n2))$



Wykonujemy inny ruch



Po wykonaniu innego ruchu



Zapamiętany ruch nie jest poprawny,
ale może stać się poprawny po ponownym odwróceniu

Wnioski z przykładu:

- Musimy też brać pod uwagę kierunek przechodzenia krawędzi w bieżącym rozwiązaniu
- Każdy inny ruch może tę kolejność zmienić
- 3 sytuacje (kiedy przeglądamy ruchy od najlepszego):
 - Usuwane krawędzie nie występują już w bieżącym rozwiązaniu
 - -> usuwamy ruch z *LM*
 - Usuwane krawędzie występują w bieżącym rozwiązaniu w różnym od zapamiętanego kierunku – może być aplikowalny w przyszłości
 - -> zostawiamy ruch w *LM*, ale nie aplikujemy, przechodzimy dalej
 - Usuwane krawędzie występują w bieżącym rozwiązaniu w tym samym kierunku (także obie odrócone)
 - -> aplikujemy i usuwamy z *LM*

Oceniając nowe ruchy trzeba też uwzględniać (dodawać do *LM*) także ruchy dla odróconego (względem obecnego) względnego kierunku krawędzi – nie są one aplikowalne do bieżącego rozwiązania, ale mogą się stać aplikowalne po wykonaniu innego ruchu

Wersja dla TSP i wymiany dwóch krawędzi:

Zainicjuj LM – listę ruchów przynoszących poprawę uporządkowaną od najlepszego do najgorszego

Wygeneruj rozwiązanie startowe x

powtarzaj

przejrzyj wszystkie **nowe** ruchy (uwzględniając także ruchy z odróconym względnym kierunkiem krawędzi) i dodaj do LM ruchy przynoszące poprawę

Przeglądaj ruchy m z LM od najlepszego do znalezienia aplikowalnego ruchu

Jeżeli co najmniej jednej z usuwanych krawędzi nie ma już w rozwiązaniu, usuń m z LM

Jeżeli obie usuwane krawędzie są w rozwiązaniu ale w odwróconym względnym kierunku (względem zapamiętanego), pomiń m ale pozostaw go w LM

jeżeli znaleziono aplikowalny ruch m **to**

$x := m(x)$ (zaakceptuj $m(x)$)

dopóki nie znaleziono aplikowalnego ruchu m po przejrzeniu całej listy LM

Ruchy kandydackie

Jako kandydackie stosujemy ruchy wprowadzające do rozwiązania co najmniej jedną krawędź kandydacką. Krawędzie kandydackie definiujemy wyznaczając dla każdego wierzchołka 10 innych najbliższych wierzchołków. Parametr ten można też dobrać eksperymentalnie, tak aby uzyskiwać jak najlepsze wyniki. Dla przykładu przeglądanie ruchów (między trasami lub wewnątrz trasy) może wyglądać następująco:

dla każdego wierzchołka n_1

dla każdego wierzchołka n_2 z listy wierzchołków najbliższych n_1

oceń ruch (ruchy) wprowadzający krawędź n_1 - n_2 do rozwiązania (**uwaga nie jest to równoważne wymianie wierzchołka n_1 z n_2**)

Opis eksperymentu

Stosujemy większe instancje kroa200 i krob200, ponieważ czas obliczeń powinien być wyraźnie krótszy.

Jako punkty odniesienia uruchamiamy też lokalne przeszukiwanie w wersji stromej bez powyższych mechanizmów oraz najlepszy algorytm z zadania 1 (heurystykę konstrukcyjną).

Każdy z czterech algorytmów na każdej instancji uruchamiany 100 razy startując (lokalne przeszukiwanie) z rozwiązań losowych.

Sprawozdanie – analogiczne jak poprzednio.