

Kryteria aspiracji

Cel: określić kiedy restrykcje „tabu” mogą zostać pominięte

Pierwsze kryterium aspiracji:

- usunąć ograniczenie „tabu” tylko wtedy, gdy ruch da rozwiązanie lepsze od znalezionego do tej pory

Aspiracja domniemana

Jeżeli wszystkie ruchy są „tabu” i nie są dozwolone przez inne kryteria, to wybierany jest ruch najmniej „tabu”

Aspiracja wg kryterium

Forma globalna – ruch „tabu” jest akceptowany jeżeli

$$c(x') < \text{best_cost}$$

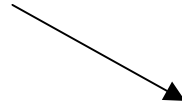
Forma regionalna (przestrzeń dzieli się na podregiony R) – ruch „tabu” jest akceptowany jeżeli

$$c(x') < \text{best_cost}(R)$$

Intensyfikacja i dywersyfikacja w Tabu Search



w „dobrych” rejonach



w rzadko odwiedzanych rejonach

Intensyfikacja

1. Powrót do najlepszego rozwiązania znalezione do tej pory
2. Skrócenie listy Tabu (short term memory)
3. Dla pamięci długotrwałej (long term memory)
 - każde rozwiązanie lub ruch to zbiór komponentów
 - zapamiętywanie komponentów dobrych ruchów lub rozwiązań w czasie obliczeń
 - w czasie intensyfikacji ruchy i rozwiązania włączają „dobre” komponenty

Pamięć długotrwała w TS daje możliwość uczenia się

Dywersyfikacja

1. Karanie często wykonywanych ruchów
2. Duże kary - ucieczka z regionu

Zmodyfikowana funkcja celu

$$f \sim = f + Int + Div$$

Konstrukcja listy kandydatów

$$N^*(x) \subset S$$

Jaki podzbiór sąsiadów ma być kandydatami?

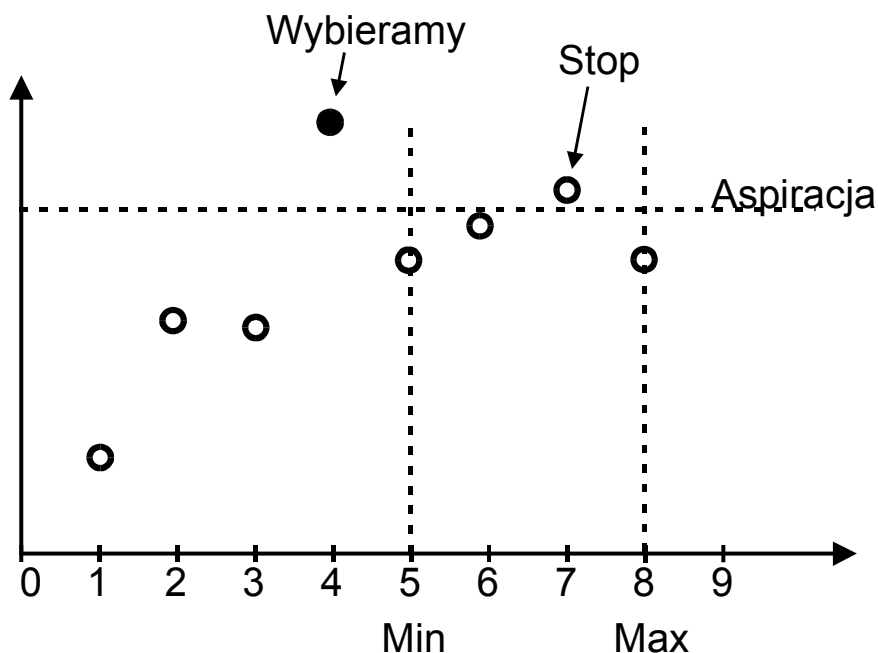
- dobrzy kandydaci
- wyodrębnić regiony sąsiedztwa zawierające ruchy o pewnych pożądanych cechach

Ogólne strategie tworzenia listy kandydatów

Aspiracja plus

- próg jakości ruchu – bazuje na historii przeszukiwania
- przeszukiwanie sąsiedztwa, aż do znalezienia sąsiada lepszego o pewną wartość progową (plus)
- ilość kandydatów rośnie aż do osiągnięcia wartości progowej
- wybieramy najlepszy znaleziony dotąd ruch

Min < ilość ruchów sprawdzonych < Max



Poziom aspiracji może być zmienny w trakcie przeszukiwania

Elitarna lista kandydatów

1. tworzenie master list – sprawdzenie wszystkich lub większości ruchów, wybór k -najlepszych (k jest parametrem)
2. w następnych iteracjach aktualnie najlepszy ruch ze zbioru k -ruchów jest wybierany do wykonania, aż strata jakości ruchu spadnie poniżej danego progu, lub zostanie osiągnięta pewna liczba iteracji

Motywacja – „dobry” ruch jeśli nie jest wykonany w bieżącej iteracji będzie wciąż „dobry” w kilku następnych

