

Algorytm przeszukiwania z zabronieniami

Algorytm Taboo Search (pamięć krótkoterminowa, kryterium aspiracji)

Krok 1.

Określ długość listy tabu (czas obowiązywania zakazu) T , wybierz $x_{start} \in X$ oraz:

$$x_a := x_{start}, x_{min} := x_{start};$$
$$Q := Q_{min} := Q(x_{start}), TL := \emptyset;$$

Krok 2. Dla $k=1$ do K wykonaj :

$$x_{new} := \arg \min \{Q(x) : x \in N(x_a) \setminus x_a, x \notin TL\}$$
$$x_{tabu} := \arg \min \{Q(x) : x \in N(x_a) \setminus x_a, x \in TL\}$$

Krok 2a.

$$x_a := x_{new};$$

a) Jeśli $Q(x_a) < Q_{min}$ to: $x_{min} := x_a$ oraz $Q_{min} := Q(x_a)$;

b) Sprawdzenie kryterium aspiracji:

Jeśli $Q(x_{tabu}) < Q_{min}$ to $x_a := x_{tabu}$, $x_{min} := x_{tabu}$ oraz $Q_{min} := Q(x_{tabu})$;

Krok 2b. Korekta listy tabu:

$TL := TL \cup \{r : (x_a^{(k)} \rightarrow x_a^{(k+1)})\}$ oraz usuń te ruchy z listy, które przebywają na niej dłużej niż T iteracji algorytmu.

Algorytm symulowanego wyżarzania (SA)

Krok 1.

- Podstaw $T = T_0$
- Wygeneruj rozwiązanie startowe: x_a .
- Podstaw $x^* = x_a$ oraz $\phi^* = \phi(x_a)$

Krok 2. Powtarzaj dopóki $T > T_{min}$

Krok 2.1. Powtarzaj k -krotnie:

- Utwórz nowe rozwiązanie x_n w sąsiedztwie x_a
- Oblicz $\Delta = \phi(x_n) - \phi(x_a)$
- Jeśli $\Delta \leq 0$.
 - to podstaw $x_a = x_n$
 - jeśli $\phi(x_s) < \phi^*$ to podstaw $x^* = x_a$ oraz $\phi^* = \phi(x_s)$

- Jeśli $\Delta > 0$.
 - to wylosuj $\delta = \text{RANDOM}[0, 1]$.
 - jeśli $\delta < e^{-\frac{\Delta}{T}}$ to podstaw $x_a = x_n$

Krok 2.2. Podstaw $T = \alpha \cdot T$

Schemat algorytmu ewolucyjnego

```
procedure EA
   $t \leftarrow 0$ 
  Inicjalizacja( $P(t)$ )
  Oceń( $P(t)$ )
  while (not warunek_stopu)
    do {
       $t \leftarrow t + 1$ 
       $P(t) \leftarrow$  Wybierz( $P(t-1)$ )
      Zmień( $P(t)$ )
      Oceń( $P(t)$ )
    }
  return ( $P(t)$ )
end EA;
```

- $P(t) = \{a^{(t)}_1, a^{(t)}_2, \dots, a^{(t)}_n\}$ – populacja rozwiązań/osobników
- Podprocedury:
 - Inicjalizacja($P(t)$) – tworzy wstępną populację osobników
 - Oceń($P(t)$) – ocenia rozwiązań w oparciu o funkcję oceny
 - Zmień($P(t)$) – przetwarza nowo utworzoną populację za pomocą operatorów genetycznych