

Wydział Zarządzania AGH

Katedra Informatyki Biznesowej i Inżynierii Zarządzania



Metaheurystyki w zarządzaniu

Narzędzia informatyczne w zarządzaniu

Metaheurystyki – treści zajęć

- Wprowadzenie
- Metaheurystyki i algorytmy heurystyczne
- Podstawy algorytmów ewolucyjnych
- Wprowadzenie do SSN
- Zastosowania

Metaheurystyki, heurystyki

- Wprowadzenie, podstawowe pojęcia
- Proste techniki przeszukiwania
- Symulowane wyżarzanie
- Tabu search

Wprowadzenie, podstawowe pojęcia

- Inteligencja obliczeniowa (Computational Intelligence, CI)
 - Dziedzina nauki zajmująca się rozwiązywaniem problemów, które nie są efektywnie algorytmizowane.
 - Computational intelligence to inteligencja zbudowana przy użyciu systemu komputerowego.
 - CI z reguły próbuje naśladować inteligencję człowieka, aczkolwiek nie jest to ani ograniczenie, ani wymaganie.
 - Synonim inteligencji maszynowej.
 - Synonim sztucznej inteligencji.



Wprowadzenie, podstawowe pojęcia

- Organizacje (przede wszystkim gospodarcze i rządowe) są głównie zainteresowane dwoma zadaniami:
 1. przewidywaniem tego, co się ma wydarzyć (predykcja),
 2. podejmowaniem najlepszych decyzji w warunkach ryzyka i niepewności (optymalizacja).
- Zadaniem CI jest dostarczyć narzędzia do modelowania, symulacji i optymalizacji zaspokajające te potrzeby.

Klasyczne metody prognozowania, i:

1. Sieci neuronowe,
2. Systemy rozmyte,
3. Programowanie genetyczne,
4. Systemy agentowe,
5. Techniki DM.

Klasyczne metody OR, i:

1. Algorytmy ewolucyjne,
2. Inteligencja roju,
3. Symulowane wyżarzanie,
4. Tabu search,
5. Algorytmy mrówkowe.

Źródło: Z. Michalewicz, *Adaptive business intelligence*.

Wprowadzenie, podstawowe pojęcia

- **Przypadek Timken Steel**

System planowania produkcji opracowany przez amerykańską firmę **i2 Technologies** o nazwie **RYTHM** dostarcza inteligentne rozwiązania dla planowania i harmonogramowania w przedsiębiorstwach. Integrują się one z systemami typu **ERP** i z transakcyjnymi systemami baz danych.

- W **Timken Steel**, produkującej pręty ze stali stopowej i rury bez szwu, realizowanych jest równocześnie od 8 do 15 tys. zamówień. Dzięki zastosowaniu systemu cykl produkcyjny w Timken został skrócony o 30 – 40%, zapasy zostały zredukowane o 25% i znacznie poprawiono terminowość realizacji zamówień. Ponieważ wyroby przepływają szybciej przez wydziały/oddziały, Timken zredukował wydatki operacyjne.

Heurystyka, metaheurystyka

- *Heuriskein* – szukam, sztuka znajdowania rozwiązań problemów.
- Heurystyka:
 - niepewność wyniku,
 - niekompletność dostępnej wiedzy,
 - poprawianie rozwiązania,
 - doświadczenie, intuicja.
- W inteligencji obliczeniowej termin *heurystyka* oznacza właściwie przeciwieństwo metody dającej optymalne rozwiązanie. Początek – lata 60.

Heurystyka, metaheurystyka

Heurystyka jest specjalizowaną metodą rozwiązywania konkretnego problemu, która znajduje dobre rozwiązania przy akceptowalnych nakładach obliczeniowych, ale bez gwarancji osiągnięcia optymalności celu, czy nawet - w wielu przypadkach - jak blisko optymalnego jest otrzymane rozwiązanie.

Metaheurystyka - ogólna metoda służąca za szkielet do konstrukcji heurystyki rozwiązującej dowolny problem, który można opisać za pomocą pewnych definiowanych przez tę metodę pojęć.

Metody takie nie służą do rozwiązywania konkretnych problemów, a jedynie podają sposób na utworzenie odpowiedniego algorytmu heurystycznego.

Cechy metaheurystyk

- strategie określające sposób przeszukiwania przestrzeni dopuszczalnych rozwiązań problemu
- celem działania jest efektywne przeszukiwanie przestrzeni
 - znajdowanie dobrych rozwiązań w określonym regionie (**eksploatacja**)
 - przeglądanie możliwie najszerszego obszaru przestrzeni problemu (**eksploracja**)
- metody przybliżone i przeważnie niedeterministyczne
- stosują różne techniki: od prostego przeszukiwania lokalnego do skomplikowanych procesów ewolucyjnych
- wykorzystują mechanizmy zapobiegające utknięciu metody w ograniczonym obszarze przestrzeni problemu
- nie są specjalizowane dla żadnego specyficznego problemu
- wykorzystują wiedzę o problemie i/lub doświadczenie zgromadzone w procesie przeszukiwania przestrzeni

Heurystyka, metaheurystyka

Argumenty za:

- Akceptowalna złożoność obliczeniowa,
- Elastyczność.

Podstawowe heurystyki:

- Losowe przeszukiwanie
- Algorytmy zachłanne

Podstawowe metaheurystyki:

- Metoda wspinaczki (hill climbing)
- Symulowane wyżarzanie
- Tabu Search
- Iterated Local Search
- Algorytmy ewolucyjne
- Algorytmy mrówkowe
- Inteligencja roju

1. Szczegółowe informacje na temat omawianych tu metod są łatwe do znalezienia.
2. Jeśli znasz inne metody, które mogą być interesujące dla uczestników tego kursu, daj znać – podyskutujemy.

Proste techniki heurystyczne

- **Pełny przegląd** przestrzeni poszukiwań.
- **Przeszukiwanie losowe**
Obecnie rzadko stosowane podejście: polega na losowym lub dokonywanym wg przyjętych reguł wyborze punktów w przestrzeni dopuszczalnych rozwiązań i obliczaniu dlań funkcji celu. Najlepsze rozwiązanie jest uznawane za rozwiązanie problemu.
- **Algorytm zachłanny (greedy algorithm)**
algorytm, który w celu wyznaczenia rozwiązania w każdym kroku dokonuje zachłannego, tj. najlepiej rokującego w danym momencie wyboru **rozwiązania częściowego**. Algorytm zachłanny nie bada, jaki jest skutek danego działania w kolejnych krokach, ale dokonuje decyzji lokalnie optymalnej, kontynuując rozwiązanie podproblemu wynikającego z podjętej decyzji.
- Przykład: NN dla TSP, NEH dla FP, wydawanie reszty.

Proste techniki heurystyczne

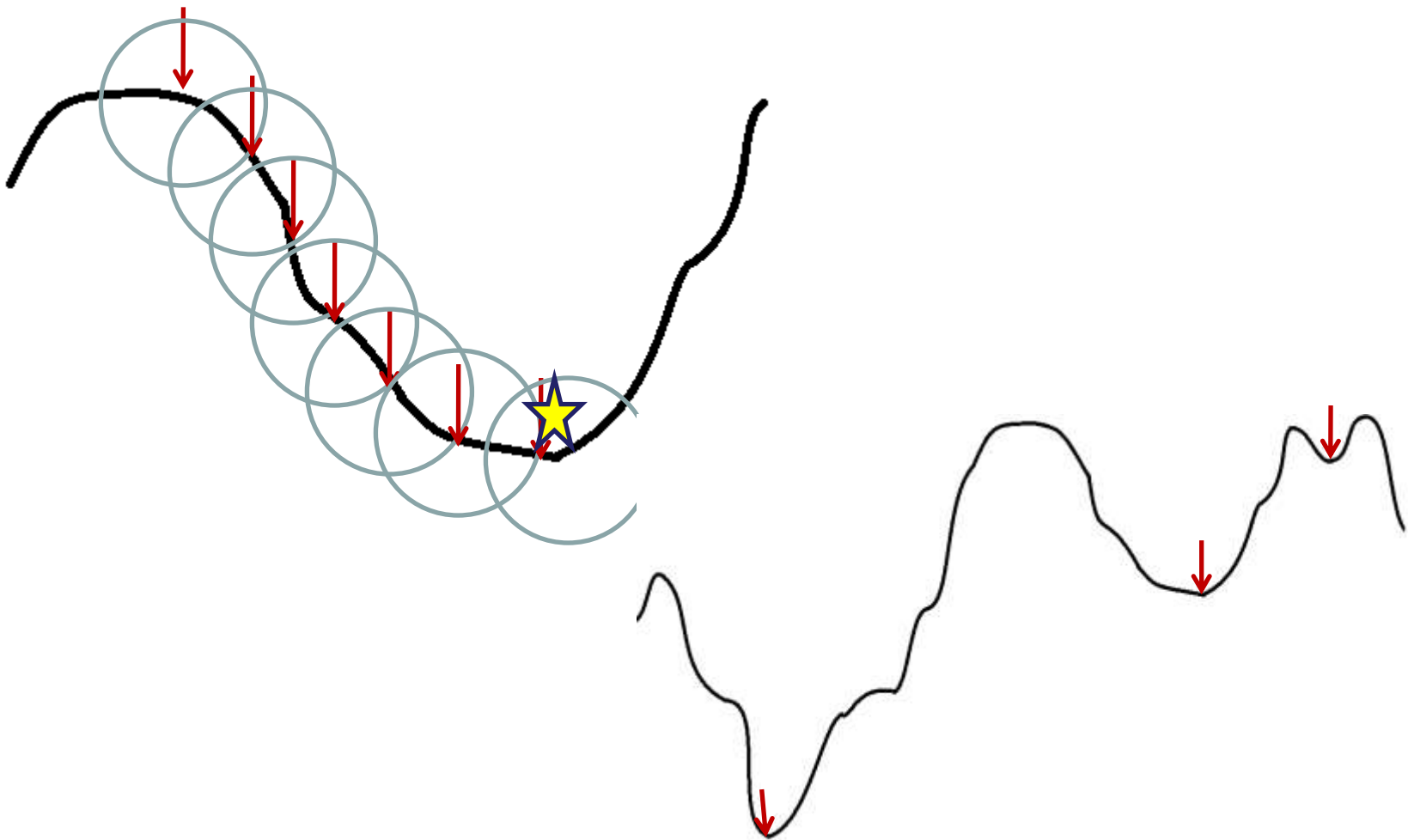
- **Metoda wspinaczki (hill climbing)**

1. Wybierz rozwiązanie początkowe (najczęściej losowo).
2. Przeanalizuj sąsiadów rozwiązania i określ ich jakość.
3. Wybierz najlepsze rozwiązanie jako początkowe do następnej iteracji.
4. Powtarzaj 2-4, aż znalezienie rozwiązania lepszego nie będzie możliwe.
5. Wyprowadź rozwiązanie **aktualne** jako rozwiązanie problemu.

Przykłady.

Proste techniki heurystyczne

Metoda wspinaczki (hill climbing)



Proste techniki heurystyczne

- **Przeszukiwanie iteracyjne**

Połączenie przeszukiwania losowego i wspinaczki daje ciekawą technikę przeszukiwania iteracyjnego. Po znalezieniu optimum lokalnego, procedura przeszukiwania jest wznawiana w nowym, losowo wybranym, punkcie startowym. Jest to bardzo prosta technika i bardzo efektywna dla problemów z niewielką liczbą optimów lokalnych.

Każde przeszukiwanie jest samodzielnym procesem wykonywanym **niezależnie** od pozostałych, stąd nie wykorzystuje się wiedzy o poprzednich rozwiązaniach. Tak więc przeszukiwanie jest równie prawdopodobne dla obszarów o zróżnicowanych wartościach funkcji celu.

Rodzaje metaheurystyk

Inspirowane przyrodą (naturą)

Zjawiska biologiczne i społeczne:

- Algorytmy ewolucyjne,
- Algorytmy mrówkowe,
- Sztuczne systemy immunologiczne,
- Inteligencja roju (owady, ptaki),
- Tabu search.

Zjawiska fizyczne i chemiczne: symulowane wyżarzanie.

Inspirowane pozaprzyrodniczo: ILS, VNS

Populacyjne vs. niepopulacyjne (pojedyncze rozwiązanie)

Symulowane wyżarzanie

Simulated annealing (SA)

Podstawy działania

Ogólnie ujmując proces odprężania (wyżarzania) polega na powolnym obniżaniu temperatury tak, by w każdej temperaturze materiał mógł osiągnąć stan równowagi. Takie postępowanie zapobiega powstawaniu wewnętrznych naprężeń i pozwala na osiągnięcie globalnego minimum energii, odpowiadające idealnemu kryształowi. Wyznaczenie stanu najniższej energii jest problemem optymalizacyjnym, stąd fizyczny proces odprężania posłużył jako wzór do opracowania nowej techniki optymalizacji globalnej.

Symulowane wyżarzanie

Podstawy działania

Pomysł działania opracował Metropolis z zespołem w 1953 roku. Punktem wyjścia były prace M. z dziedziny termodynamiki statystycznej, gdzie obowiązuje:

$$p(E) = \exp(-dE/kt)$$

Stała
Bolzmana

- Kirkpatrick z zespołem wykazali, że algorytm ten może być zastosowany do problemów optymalizacyjnych drogą analogii do procesów fizycznych.

Symulowane wyżarzanie

- SA należy do technik przeszukiwania sąsiedztwa.
- **Sąsiedztwo** $N(x,m)$ rozwiązania x jest zbiorem rozwiązań, które mogą być osiągnięte z x drogą prostej operacji m , zwanej ruchem. Jeśli rozwiązanie y jest lepsze od dowolnego z jego sąsiedztwa $N(y,m)$, wtedy y jest optimum lokalnym.
- Wymagania odnośnie **sąsiedztwa**:
 - niewielki rozmiar - liczba rozwiązań sąsiednich,
 - możliwość osiągnięcia każdego rozwiązania z dowolnego rozwiązania początkowego,
 - rozwiązania sąsiednie muszą być podobne.

Symulowane wyżarzanie

BEGIN

$R := R_0$ {rozwiązanie początkowe};

$T := T_0$ {temperatura początkowa > 0 };

wybierz funkcję redukcji temperatury α ;

WHILE (nie spełniony warunek zatrzymania) DO

BEGIN

WHILE (nie osiągnięty stan równowagi) DO

BEGIN

$R' := f(R)$ {nowe, sąsiednie rozwiązanie,
generowane losowo};

$dE := E(R') - E(R)$;

IF ($dE < 0$) THEN $R := R'$

ELSE IF ($\text{RANDOM} < \exp(-dE/T)$) THEN $R := R'$

END;

$T := \alpha(T)$;

END;

END.

Symulowane wyżarzanie

Warunkiem poprawnego działania przedstawionej techniki jest prawidłowy dobór dwóch grup czynników:

- warunek zatrzymania obliczeń,
- prawdopodobieństwo akceptacji "gorszego" rozwiązania,
- liczba generowanych rozwiązań dla danej temperatury,
- sposób generowania sąsiednich rozwiązań,
- postać funkcji kosztowej.

Kluczowym czynnikiem jest szybkość (sposób) redukcji temperatury. Najczęściej stosuje się dwa sposoby:

- redukcja geometryczna

$$\alpha(T) = a * T, \quad a < 1$$

w danej temperaturze testowanych jest wiele rozwiązań,

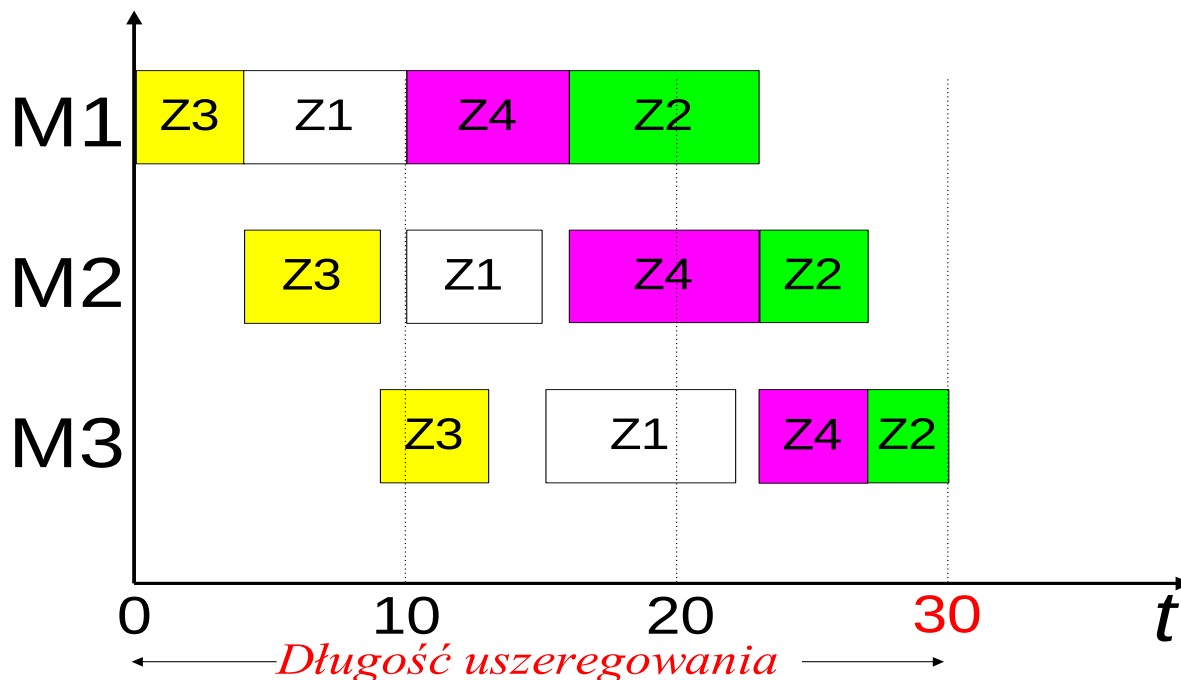
- powolny spadek

$$\alpha(T) = T / (1 + b * T), \quad b \ll 1$$

w danej temperaturze testowane jest jedno rozwiązanie.

Symulowane wyżarzanie - przykład

- Udane implementacje schematu Metropolisa dla problemu szeregowania zadań w systemie przepływowym podali Ogbu i Smith, Osman i Pots oraz Gangadharan i Rajendran.



Symulowane wyżarzanie - przykład

- Ogbu i Smith przyjęli, że działanie algorytmu przebiega w $LE = 15$ etapach, liczba sprawdzanych rozwiązań na każdym etapie jest funkcją liczby zadań n i równa się $5 \cdot (n-1)^2$ dla $n < 15$ oraz $7 \cdot (n-1)^2$ dla $n \geq 15$, natomiast prawdopodobieństwo akceptacji $AP(k)$ dla $k = 1, \dots, LE$ jest dane wzorem:

$$AP(k) = \begin{cases} AP(1) \bullet (a)^{k-1} & \text{jeżeli } du(R') > du(R) \\ 1 & \text{jeżeli } du(R') \leq du(R) \end{cases}$$

Prawd. bazowe = 0.2

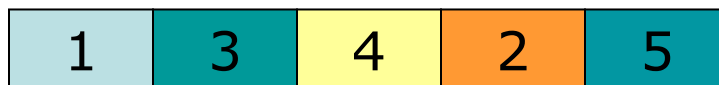
Czynnik redukujący = 0.75

Symulowane wyżarzanie - przykład

- Nowe, sąsiednie rozwiązania generowane są przy użyciu schematu nazywanego *insertion*. Polega on na tym, że po wylosowaniu dwóch liczb i, j ($i < > j$), zadanie nr i jest wstawiane na pozycję nr j sekwencji wejściowej.



$i=2, j=4$

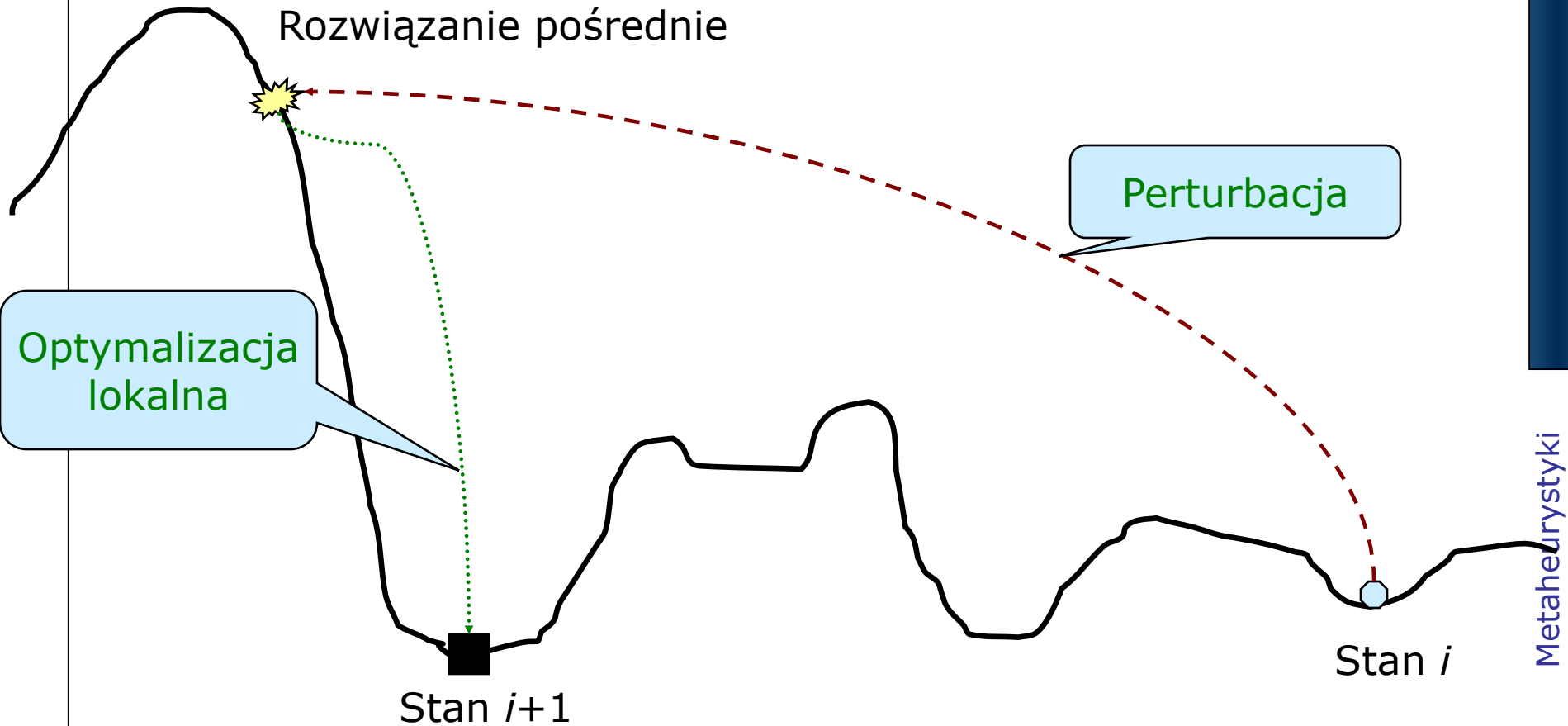


$i=4, j=2$



- Po wygenerowaniu na danym etapie wszystkich sekwencji wybierana jest ostatnia zaakceptowana jako bazowa dla następnego etapu.

Metaheurystyka ILS (CLO)



Metaheurystyka ILS (CLO)

s_0 = GenerateInitialSolution

s^* = LocalSearch(s_0)

REPEAT

s' = Perturbation(s^* , history)

$s^{*'} = \text{LocalSearch}(s')$

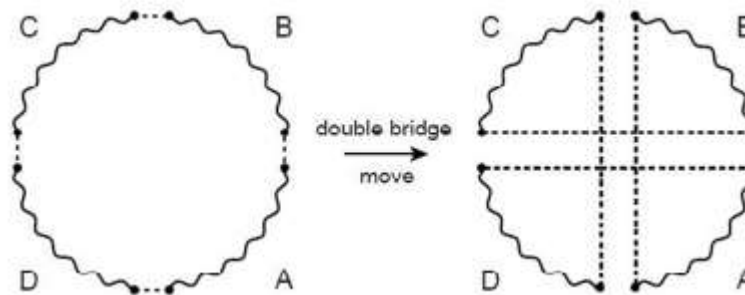
$s^* = \text{AcceptanceCriterion}(s^*, s^{*'}, \text{history})$

UNTIL termination condition met

Metaheurystyka ILS (CLO)

Klasyczne podejście *ILS* do *TSP* składa się z następujących elementów:

- sąsiedztwo: *zamiana* dwóch miast;
- perturbacja: *double-bridge* move;



- kryterium akceptacji: "Czy nastąpiła poprawa najlepszego rozwiązania?"

$$s'^* < s^*$$

Tabu search

Tabu (ang. *Taboo*) (z polinezyjskiego tapu)

- 1) w religiach ludów pierwotnych zakaz podejmowania różnych działań w stosunku do osób, miejsc, zwierząt, rzeczy bądź stanów rzeczy lub też wypowiedania pewnych słów. Przekonanie o wartości i celowości takich zachowań wynika z wiary, że są one święte lub nieczyste w sensie rytualnym. Wskutek złamania tabu następuje stan skażenia związany z odpowiednimi sankcjami, głównie ze strony sił nadprzyrodzonych. Instytucja tabu została odkryta przez J. Cooka w 1771 wśród tubylców wyspy Tonga.
 - 2) przedmiot, osoba, zwierzę, słowo, miejsce lub czynność objęte zakazem.
 - 3) w psychoanalizie zakaz ograniczający realizację dążeń seksualnych, którego konsekwencją jest stłumienie i ukrycie w podświadomości tendencji do zachowań objętych zakazem.
 - 4) w szerokim sensie tabu oznacza jakikolwiek zakaz.
- Początek dały prace Glovera z lat 70.
 - TS rozumie się jako nakładanie ograniczeń; tu jednak chodzi raczej o wykorzystanie ograniczeń do kierowania przeszukiwania w obiecujące rejony przestrzeni dopuszczalnych rozwiązań.

Tabu search

- Podstawy działania

Ruch jest funkcją transformującą dane rozwiązanie w inne rozwiązanie. Podzbiór ruchów generuje dla danego rozwiązania podzbiór rozwiązań zwany *sąsiedztwem*. Na każdym stopniu iteracji przeszukiwane jest sąsiedztwo w celu wyboru najlepszego rozwiązania, które staje się bazowym dla następnego kroku. Ostatnio wykonany ruch jest wprowadzany na *listę zabronionych ruchów*, które nie mogą być wykonane w ciągu najbliższych s iteracji. Mechanizm ten umożliwia wyjście z lokalnego minimum i chroni przed cyklicznością ruchów. Niekiedy możliwe jest wykonanie zabronionego ruchu, o ile zdefiniowana *funkcja aspiracji* określi zyskowność takiego ruchu. Koniec obliczeń następuje w momencie przekroczenia limitu czasu, po wykonaniu dopuszczalnej liczby iteracji albo po wykonaniu dopuszczalnej liczby ruchów bez poprawy wartości funkcji kryterium.

Tabu search - przykład

- Sytuacja wejściowa

Długość uszeregowania = 60

2	5	7	3	4	6	1
---	---	---	---	---	---	---

- Ruch typu wymiana (ang. swap, exchange)

$i=5, j=6$

2	6	7	3	4	5	1
---	---	---	---	---	---	---

Długość uszeregowania = 62

Tabu search - przykład

- **Lista tabu** (ruchów zabronionych)

Lista wskazuje jakie ruchy i jak długo są zabronione

$s=3$

	2	3	4	5	6	7
1						
2						
3						
4						
5						
6						

zamiana (3,5) jest
zabroniona w ciągu
trzech kolejnych iteracji

zamiana (6,7) jest
zabroniona przez
jeszcze jedną iterację

- **Funkcja aspiracji** określa warunki akceptacji zabronionego ruchu np. kryterium *najlepsze rozwiązanie*

Tabu search - przykład

Iteracja 0

2	5	7	3	4	6	1
---	---	---	---	---	---	---

Długość uszeregowania = 60

Lista tabu

	2	3	4	5	6	7
1						
2						
3						
4						
5						
6						

Top 5

Ruch	Z
5,4	-6
7,4	-4
3,6	-2
2,3	0
4,1	-1

Tabu search - przykład

Iteracja 1 - ruch (5,4)

2	4	7	3	5	6	1
---	---	---	---	---	---	---

Długość uszeregowania = $60 - 6 = 54$

Lista tabu

	2	3	4	5	6	7
1						
2						
3						
4				3		
5						
6						

Top 5

Ruch	Z
3,1	-2
2,3	-1
3,6	1
7,1	2
6,1	4

Tabu search - przykład

Iteracja 2 - ruch (3,1)

2	4	7	1	5	6	3
---	---	---	---	---	---	---

Długość uszeregowania = $54 - 2 = 52$

Lista tabu

	2	3	4	5	6	7
1		3				
2						
3						
4				2		
5						
6						

Top 5

Ruch	Z
1,3	2
2,4	4
7,6	6
4,5	7
5,3	9

Tabu search - przykład

Iteracja 3 - ruch (2,4)

4	2	7	1	5	6	3
---	---	---	---	---	---	---

Długość uszeregowania = $54+4=58$

Lista tabu

	2	3	4	5	6	7
1		2				
2			3			
3						
4				1		
5						
6						

Top 5

Ruch	Z
4,5	-6
5,3	-2
7,1	0
1,3	3
2,6	6

Tabu search - przykład

Iteracja 4 - ruch (4,5)

5	2	7	1	4	6	3
---	---	---	---	---	---	---

Długość uszeregowania = $58 - 6 = 52$

Lista tabu

	2	3	4	5	6	7
1		1				
2			2			
3						
4				3		
5						
6						

Top 5

Ruch	Z
7,1	0
4,3	3
6,3	5
5,4	6
2,6	8

Tabu search

BEGIN

$R := R_0$ {rozwiązanie początkowe};

TL := empty {lista tabu jest pusta};

$R^* := R_0$; $C^* := C(R_0)$ {najlepsze rozwiązanie i funkcja celu};

WHILE (nie spełniony warunek zatrzymania) DO

BEGIN

$C := \text{MAX}$;

WHILE (nie przetestowane sąsiedztwo rozwiązania R) DO

BEGIN

IF $m(R)$ NOT tabu THEN $R' := m(R)$ {nowe rozwiązanie po
wykonaniu ruchu m };

IF $C(R') < C$ THEN BEGIN $C := C(R')$; $R'' := R'$ END;

END;

IF $C < C^*$ THEN BEGIN $C^* := C$; $R^* := R''$ END;

$R := R''$;

aktualizacja TL;

END;

END.