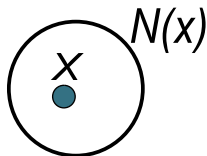


# Optymalizacja. Przeszukiwanie lokalne

dr hab. inż. Maciej Komosiński, prof. PP

Instytut Informatyki  
Politechnika Poznańska  
[www.cs.put.poznan.pl/mkomosinski](http://www.cs.put.poznan.pl/mkomosinski)

$S$



# Definicja sąsiedztwa

- $x \in S$
- zbiór  $N(x) \subseteq S$  rozwiązań, które leżą „blisko” rozwiązania  $x$
- funkcja odległości

$$dist : S \times S \rightarrow \mathbb{R}$$

- sąsiedztwo

$$N(x) = \{y \in S : dist(x, y) \leq \varepsilon\}$$

# Definicja sąsiedztwa

- $x \in S$
- zbiór  $N(x) \subseteq S$  rozwiązań, które leżą „blisko” rozwiązania  $x$
- funkcja odległości

$$dist : S \times S \rightarrow \mathbb{R}$$

- sąsiedztwo

$$N(x) = \{y \in S : dist(x, y) \leq \varepsilon\}$$

- każde rozwiązanie  $y \in N(x)$  jest nazywane rozwiązaniem sąsiednim lub po prostu sąsiadem  $x$
- zakładamy, że  $y \in N(x) \Leftrightarrow x \in N(y)$

# Cechy sąsiedztwa

- ograniczenie na rozmiar
  - dla każdego  $x$ , jego  $N(x)$  zawiera co najmniej jedno rozwiązanie  $y$  różne od  $x$
  - $N(x)$  nie może obejmować całej przestrzeni rozwiązań dopuszczalnych (nie może być wyczerpujące)
- podobieństwo sąsiadów
  - $y \in N(x)$  niewiele różni się od  $x$ , tak by przejście (elementarny ruch) od  $x$  do  $y$  nie wymagało za każdym razem konstruowania nowego rozwiązania „od podstaw”
- „równouprawnienie”
  - niezależnie od wyboru rozwiązania początkowego, każde rozwiązanie należące do  $S$  powinno być osiągalne

# Przykłady sąsiedztw dla permutacji $n$ elementów

- $k$ -zamiana (ang.  $k$ -swap,  $k$ -opt)  
 $N(x)$  – zbiór rozwiązań powstałych przez usunięcie  $k$  elementów i wstawienie ich w innej kolejności
- 2-zamiana z zachowaniem pozycji

1—2—3—4—5—6—7—8—9

1—2—8—4—5—6—7—3—9

$$|N_{2P}(x)| =$$

# Przykłady sąsiedztw dla permutacji $n$ elementów

- $k$ -zamiana (ang.  $k$ -swap,  $k$ -opt)  
 $N(x)$  – zbiór rozwiązań powstałych przez usunięcie  $k$  elementów i wstawienie ich w innej kolejności
- 2-zamiana z zachowaniem pozycji

1—2—3—4—5—6—7—8—9

1—2—8—4—5—6—7—3—9

$$|N_{2P}(x)| = \frac{n(n-1)}{2}$$

- 3-zamiana z zachowaniem pozycji.  $|N_{3P}(x)| = \dots$

# Przykłady sąsiedztw dla permutacji $n$ elementów

- $k$ -zamiana (ang.  $k$ -swap,  $k$ -opt)  
 $N(x)$  – zbiór rozwiązań powstałych przez usunięcie  $k$  elementów i wstawienie ich w innej kolejności
- 2-zamiana z zachowaniem pozycji

1—2—3—4—5—6—7—8—9

1—2—8—4—5—6—7—3—9

$$|N_{2P}(x)| = \frac{n(n-1)}{2}$$

- 3-zamiana z zachowaniem pozycji.  $|N_{3P}(x)| = \dots$
- a czy zamienianie tylko sąsiadujących elementów byłoby dobrym sąsiedztwem?
- ...

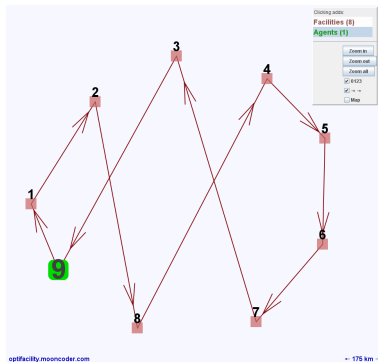
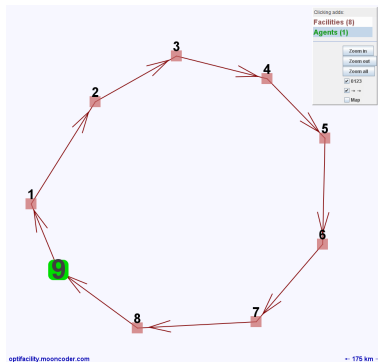


# Sąsiedztwo w TSP (zamiana miast)

$N_{2P}$ , indeksy (3, 8):

1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9

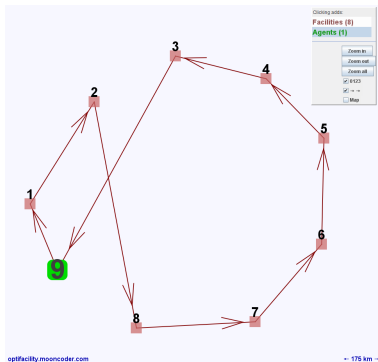
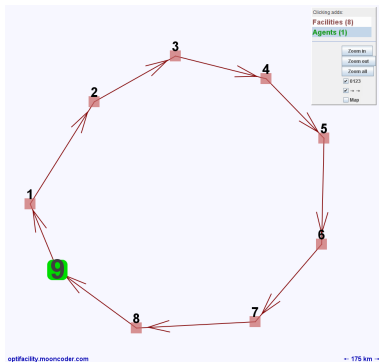
1 – 2 – 8 – 4 – 5 – 6 – 7 – 3 – 9



# Sąsiedztwo w TSP (odwrócenie ścieżki)

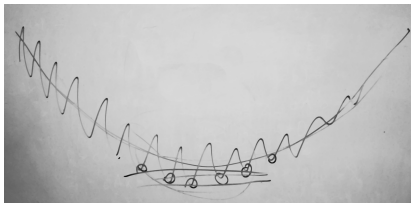
$N_{2R}$ , indeksy (3, 8):

1 — 2 —  $\overrightarrow{3 - 4 - 5 - 6 - 7 - 8}$  — 9  
1 — 2 —  $\overleftarrow{8 - 7 - 6 - 5 - 4 - 3}$  — 9



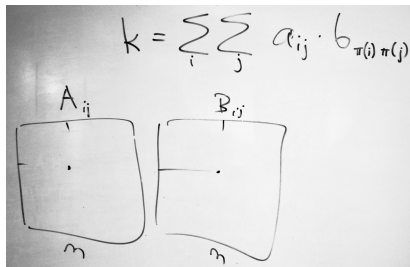
# Które sąsiedztwo bardziej ułatwi optymalizację TSP?

Krajobraz funkcji celu ma być łagodny... FDC ma być wysoka...

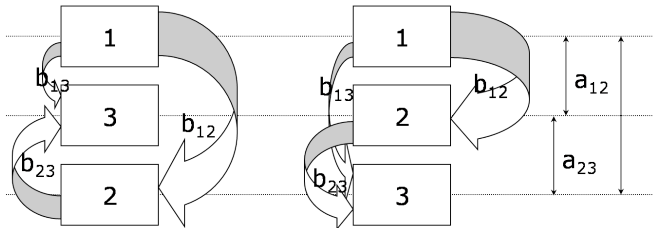


	STSP	ATSP
zamiana miast		
odwrócenie ścieżki		

# Sąsiedztwo w QAP (*quadratic assignment problem*)

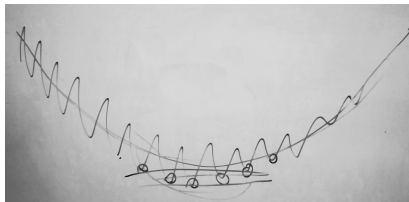
$$K = \sum_i \sum_j a_{ij} \cdot b_{\pi(i)\pi(j)}$$


Zamiana przypisania, np. 1-3-2  $\rightarrow$  1-2-3:



# Które sąsiedztwo bardziej ułatwi optymalizację QAP?

Krajobraz funkcji celu ma być łagodny... FDC ma być wysoka...



	STSP	ATSP	QAP
zamiana miast			
odwrócenie ścieżki			

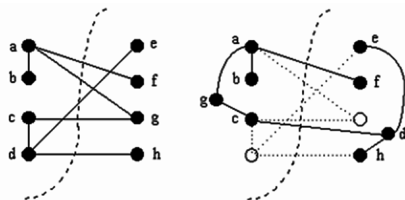
# Sąsiedztwo w GPP

Sąsiedztwo – wszystkie takie podziały  $(V'_1, V'_2)$ , że

$V'_1 = V_1 \cup \{x\}$  i  $V'_2 = V_2 \setminus \{x\}$   
lub

$V'_1 = V_1 \setminus \{y\}$  i  $V'_2 = V_2 \cup \{y\}$

$x \in V_2, y \in V_1$



Step No.	Vertex Pair	Gain	Cutset
0		0	5
1	(d, g)	3	2
2	(c, f)	1	1
3	(b, h)	-2	3
4	(a, e)	-2	5

Podójście alternatywne:

generowanie rozwiązań niedopuszczalnych, tzn.  $|V_1| \neq |V_2|$ .

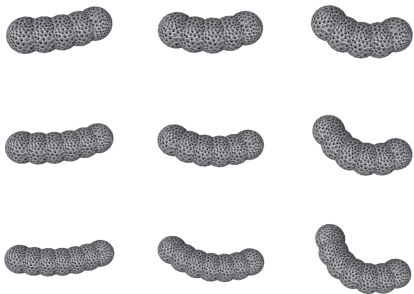
$$F(V_1, V_2) = \sum_{i \in V_1, j \in V_2} E_{ij} + \gamma(|V_1| - |V_2|)^2$$

$\gamma$  – dodatnia stała (waga kary za niedopuszczalny podział).

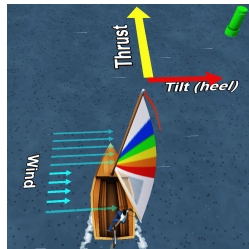
$$[4, 5.017, 3.422, -12.430, 107.819, \dots]$$

# Sąsiedztwo wektora liczb

$[4, 5.017, 3.422, -12.430, 107.819, \dots]$



<http://www.framsticks.com/foraminifera>

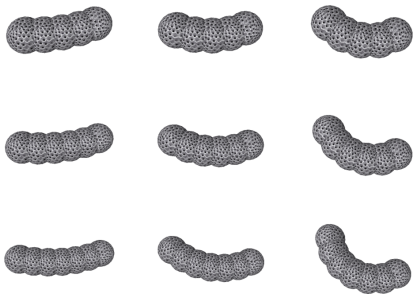


<http://sailor.mooncoder.com/>

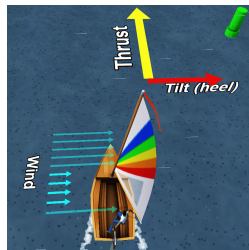


# Sąsiedztwo wektora liczb

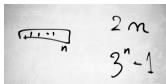
$[4, 5.017, 3.422, -12.430, 107.819, \dots]$



<http://www.framsticks.com/foraminifera>



<http://sailor.mooncoder.com/>



# Eksploracja sąsiedztwa rozwiązania $x$

- 1 wybierz rozwiązanie w  $S$ , oceń je, zdefiniuj jako rozwiązanie *bieżące*
- 2 wygeneruj nowe rozwiązania z rozwiązania bieżącego i oceń je
- 3 jeśli nowe rozwiązanie jest lepsze, uznaj je jako rozwiązanie bieżące, w przeciwnym wypadku odrzuć
- 4 powtarzaj kroki 2 i 3 dopóki można uzyskać poprawę

# Procedura przeszukiwania lokalnego (local search)

procedure PRZESZUKIWANIE\_LOKALNE;

begin

    INICJALIZUJ( $x_{start}$ );

$x_{current} := x_{start}$ ;

repeat

        GENERUJ( $y \in N(x_{current})$ );

if  $f(y) \leq f(x_{current})$  then

$x_{current} := y$ ;

until  $f(y) > f(x_{current})$  dla wszystkich  $y \in N(x_{current})$ ;

end;

# Procedura przeszukiwania lokalnego (local search)

**procedure** PRZESZUKIWANIE\_LOKALNE;

**begin**

INICJALIZUJ( $x_{start}$ );

$x_{current} := x_{start}$ ;

**repeat**

GENERUJ( $y \in N(x_{current})$ );  $\leftarrow$  implementacja?

**if**  $f(y) \leq f(x_{current})$  **then**  $\leftarrow \leq ?$

$x_{current} := y$ ;

**until**  $f(y) > f(x_{current})$  dla wszystkich  $y \in N(x_{current})$ ;  $> ?$

**end;**

# Lokalne optimum

$x_{min}$  jest *lokalnym minimum* jeśli

$$\forall_{y \in N(x_{min})} \quad f(x_{min}) \leq f(y)$$

$x_{max}$  jest *lokalnym maksimum* jeśli

$$\forall_{y \in N(x_{max})} \quad f(x_{max}) \geq f(y)$$

# Lokalne optimum

$x_{min}$  jest *lokalnym minimum* jeśli

$$\forall y \in N(x_{min}) \quad f(x_{min}) \leq f(y)$$

$x_{max}$  jest *lokalnym maksimum* jeśli

$$\forall y \in N(x_{max}) \quad f(x_{max}) \geq f(y)$$

ostra vs. nieostra nierówność!



## Dwie odmiany Local Search

- Greedy – first improvement; first descent
- Steepest – best improvement; highest descent

# Dwie odmiany Local Search

- Greedy – first improvement; first descent
- Steepest – best improvement; highest descent

Czy umiał(a)byś już zaimplementować LS?



# Przykład (maksymalizacja na siatce 2D)

<http://www.nlife.pl/opt/p/index.html>

$x_1 \backslash x_2$	1	2	3	4	5
A	1	2	1	7	4
B	0	2	1	2	4
C	1	3	3	4	8
D	3	4	5	3	3
E	6	5	3	2	1

- zacznij od B2
- zasymuluj Greedy i Steepest
- rozważ dwa sąsiedztwa: Moore'a (8) i von Neumanna (4)
- pomyśl o wpływie warunku  $<$  vs.  $\leq$
- sformułuj wnioski

# Efektywne obliczanie kosztu nowego rozwiązania

- TSP: czas liniowy (naiwnie) lub stały (łatwo)
- QAP: czas kwadratowy (naiwnie) lub liniowy (łatwo) lub prawie-stały (tablica delt)

## Wady

- kończą działanie (zbyt szybko?) w optimum lokalnym
- jakość uzyskiwanych rozwiązań może zależeć od wyboru rozwiązania startowego
  - dla większości problemów nie ma żadnych wskazań co do tego, jak najlepiej wybrać rozwiązanie startowe

## Wady

- kończą działanie (zbyt szybko?) w optimum lokalnym
- jakość uzyskiwanych rozwiązań może zależeć od wyboru rozwiązania startowego
  - dla większości problemów nie ma żadnych wskazań co do tego, jak najlepiej wybrać rozwiązanie startowe

## Zalety

- są elastyczne
- można stosować dla każdego problemu optymalizacji kombinatorycznej
- bardzo szybkie i proste

# Unikanie wad algorytmów lokalnego przeszukiwania

- wykonanie algorytmu lokalnego przeszukiwania dla dużej liczby różnych rozwiązań startowych
  - odmiana multi-random start
  - odmiana guided local search: funkcja celu jest modyfikowana przy kolejnych uruchomieniach tak, że fragmenty rozwiązań ze znalezionych wcześniej optimów lokalnych są karane
- wprowadzenie bardziej złożonej definicji sąsiedztwa w celu przeszukania większej części przestrzeni rozwiązań dopuszczalnych
- zmienianie wykorzystywanego sąsiedztwa podczas działania – *variable neighborhood search*
- ograniczone akceptowanie pogorszeń funkcji celu
- ...