

*Strategie przeszukiwania lokalnego i problemy
optymalizacji*

wykład 4

dr inż. Joanna Kołodziejczyk

`jkolodziejczyk@wi.ps.pl`

Zakład Sztucznej Inteligencji ISZiMM

Plan wykładu

Przeszukiwanie lokalne:

- Algorytm największego wzrostu
- Symulowane wyżarzanie
- Szukanie wiązką
- Algorytmy genetyczne

Lokalne przeszukiwanie

Metody dotychczas omawiane systematycznie eksplorowały przestrzeń zachowując w pamięci alternatywne ścieżki do momentu odnalezienia rozwiązania.

W wielu problemach takich jak: projektowanie układów cyfrowych, harmonogramowanie, wyznaczanie drogi pojazdu, optymalizacja sieci telekomunikacyjnych itd. ścieżka nie jest istotna. Natomiast ważne jest **rozwiązanie** (o jak najlepszych walorach).

Lokalne metody przeszukiwania operują na pojedynczym stanie i generują krok do stanu sąsiedniego. Algorytmy nie są systematyczne, ale mają następujące zalety:

- Małe wymagania pamięciowe: zazwyczaj stała wielkość
- Często znajdują akceptowalne rozwiązania w dużej i nieskończonej przestrzeni rozwiązań, gdy metody systematyczne zawodzą.

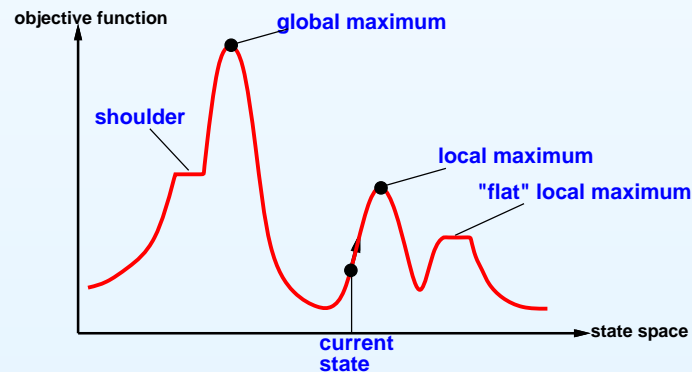
Metody lokalne są dedykowane do zadań **optymalizacji** w których celem jest znalezienie najlepszego stanu według **funkcji celu**.

Krajobraz przestrzeni stanów

Krajobraz posiada:

- położenie — definiowane przez stan
- wzniesienia — zdefiniowane przez wartość funkcji celu (funkcji heurystycznej)

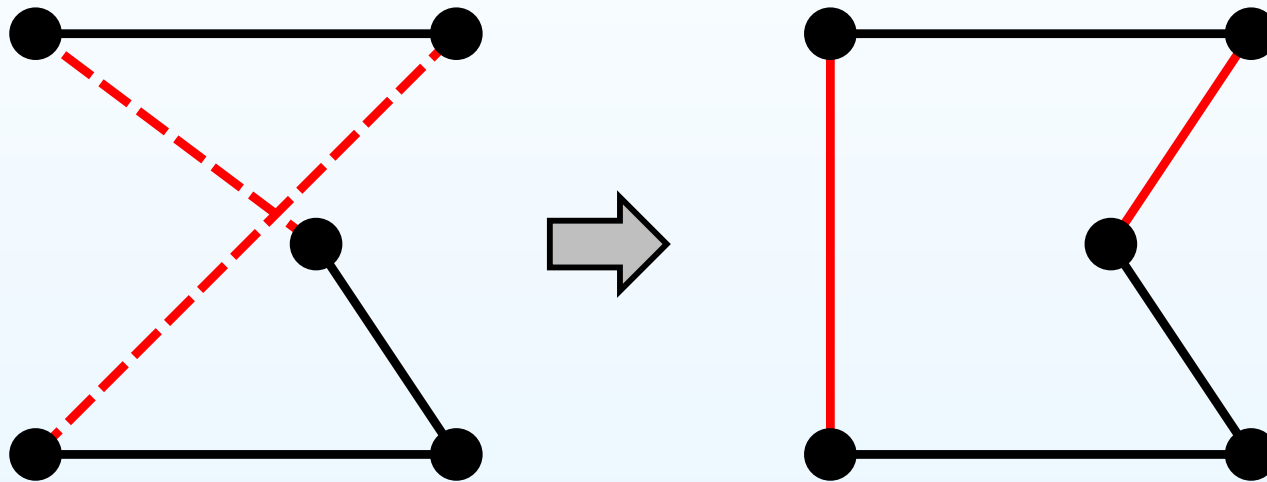
Jeżeli wzniesienia odpowiadają kosztowi, to oczywiście poszukujemy najniższej doliny (**poszukiwanie globalnego minimum**), w przeciwnym przypadku szukamy najwyższego wzniesienia w krajobrazie (**globalne maksimum**).



Przestrzeń stanów jest równa „pełnym konfiguracjom”, a zadanie polega na poszukiwaniu optymalnej konfiguracji.

Przykład — komiwojażer

Rozpocznij z dowolnej pełnej trasy. Wymieniaj miasta parami.

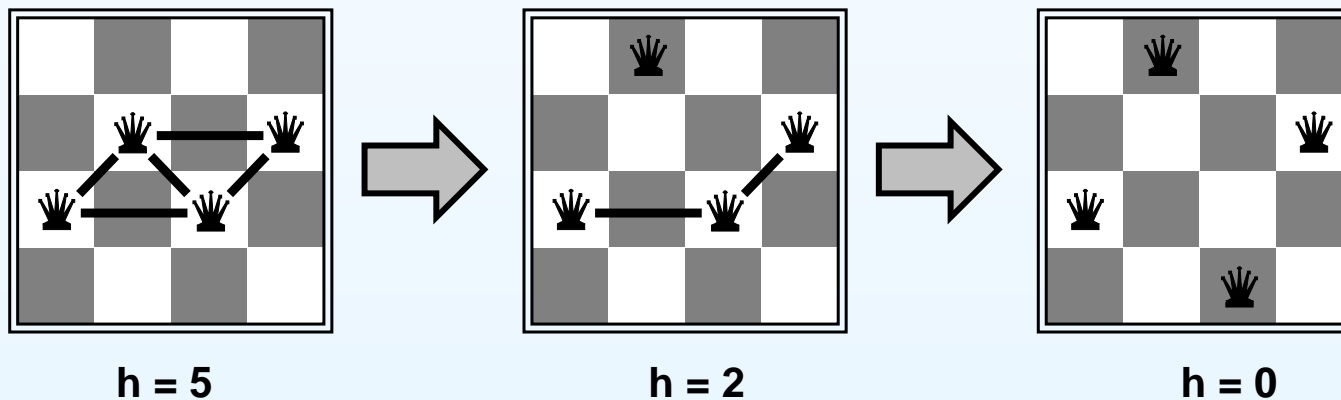


Prezentowane podejście lub jego warianty bardzo szybko znajdują do 1% optimów, nawet dla problemów z tysiącem miast.

Przykład — n hetmanów

Zadanie: Ustaw n hetmanów na szachownicy $n \times n$ z hetmanami ustawionymi w różnych wierszach, kolumnach lub przekątnych.

Poruszaj hetmanem by zredukować liczbę konfliktów.



Podejście prawie zawsze w krótkim czasie rozwiązuje problem n -hetmanów, nawet dla dużego n (np, $n = 1,000,000$).

Metoda największego wzrostu (hill-climbing)

W pętli, nieustannie przesuwamy się w kierunku **rosnących wartości** funkcji celu.

Kończy swoje działanie kiedy osiągnie szczyt, tj. wszyscy sąsiedzi mają mniejsze wartości funkcji celu.

Metoda nie wymaga zastosowania drzewa. Bieżący stan jako struktura danych musi zapamiętywać stan i wartość funkcji celu. „Nie wybiega w przód ani nie patrzy za siebie”.

Metoda największego wzrostu **wybiera losowo** ze zbioru stanów potomnych, jeżeli kilka z nich ma taką samą wartość funkcji celu.

W języku angielskim metoda znana również jako: greedy local search.

Metoda największego wzrostu

```
1  function HILL-CLIMBING( problem) return a state
2      that is a local maximum
3  input: problem, a problem
4  local variables: current, a node.
5      neighbor, a node.
6
7  current = MAKE-NODE(INITIAL-STATE[problem])
8  loop do
9      neighbor = a highest valued successor of current
10     if VALUE [neighbor] <= VALUE[current] then
11         return STATE[current]
12     current = neighbor
```

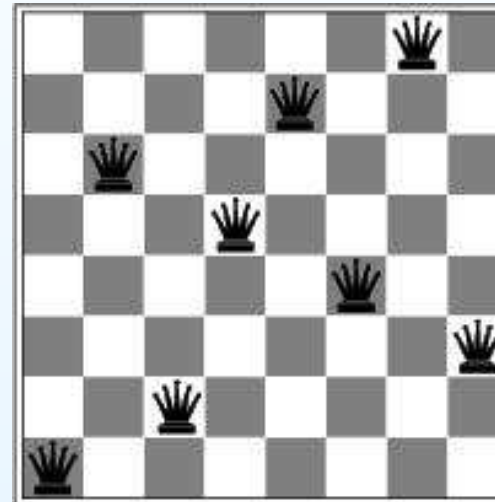
W każdym kroku bieżący węzeł jest zastępowany sąsiadem z największą wartością VALUE. Jednak jeżeli funkcja heurystyczna jest oszacowanie, to sąsiad może nie być bezwzględnie lepszy.

Przykład dla n hetmanów

Założenia: Funkcja generująca potomka — przesuń pojedynczego hetmana w tej samej kolumnie.

Funkcja heurystyczna $h(x)$: liczba par hetmanów, które się atakują.

18	12	14	13	13	12	14	14
14	16	13	15	12	14	12	16
14	12	18	13	15	12	14	14
15	14	14	♔	13	16	13	16
♔	14	17	15	♔	14	16	16
17	♔	16	18	15	♔	15	♔
18	14	♔	15	15	14	♔	16
14	14	13	17	12	14	12	18

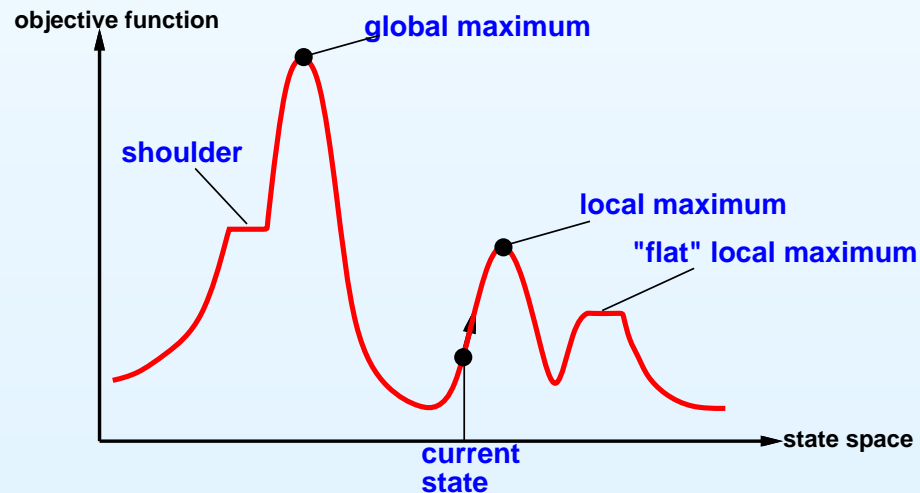


Rys. lewy: stan $h = 17$ i wartości h dla każdego możliwego potomka.

Rys. prawy: minimum lokalne o wartości $h = 1$.

Wady

- Lokalne maksima — jest to szczyt w krajobrazie przystosowawczym, który jest niższy niż globalne optimum. Algorytm lubi utykać w takich miejscach.
- Grzbiet — sekwencja lokalnych maksimów. Trudno poruszać się w takich krajobrazach.
- Obszary płaskie — obszary przestrzeni przystosowawczej, gdzie na dużym obszarze wartość funkcji celu jest stała. Może nie znaleźć drogi wyjściowej z obszaru.



Dla 8-hetmanów utyka w 86% przypadków, gdy stan początkowy jest losowany.

Warianty metody największego wzrostu

- **Stochastyczna metoda największego wzrostu:** losowy wybór pomiędzy ścieżkami wzrastającymi. Prawdopodobieństwo wyboru zależy od nachylenia zbocza.
- **Metoda największego wzrostu pierwszego wyboru:** generuje losowo potomków dopóki nie znajdzie lepszego od bieżącego stanu.
- **Losowo restartowana metoda największego wzrostu:** próbuje pokonać problem utykania w lokalnych minimach.
- **Losowe przemieszczenia:** ucieczka z płaskich obszarów prowadzących do zbocza narastającego, ale zapętli się na płaskim lokalnym maksimum.

Symulowane wyżarzanie (simulated annealing)

Unika lokalnych maksimów poprzez zezwalanie na kroki w kierunku malejącej wartości funkcji heurystycznej.
Częstotliwość i wielkość takich skoków maleje z czasem.

Zazwyczaj funkcja celu (heurystyczna) jest minimalizowana.

Symulacja procesu studzenia kryształów do osiągnięcia stanu minimalnej energii.

Jeżeli T zmniejsza się dostatecznie wolno, to osiągnie się najlepszy stan.

Symulowane wyżarzanie

```
1  function SIMULATED-ANNEALING( problem, schedule) return
2                                     a solution state
3  input: problem- a problem
4         schedule- a mapping from time to temperature
5  local variables: current - a node.
6         next - a node.
7         T - a "temperature" controlling the probability
8               of downward steps
9
10 current = MAKE-NODE(INITIAL-STATE[problem])
11 for t = 1 to inf do
12     T = schedule[t]
13     if T = 0 then return current
14     next = a randomly selected successor of current
15     deltaE = VALUE[next] - VALUE[current]
16     if deltaE > 0 then current = next
17     else current = next only with probability  $e^{(\text{deltaE}/T)}$ 
```

Lokalne szukanie wiązką (local beam search)

Jednocześnie śledzi k ścieżek (stanów) zamiast jednego:

1. Wylosuj k stanów początkowych.
2. Określ wszystkich potomków k stanów z kroku poprzedniego.
3. Jeżeli którykolwiek potomek jest celem, to zatrzymaj.
W przeciwnym przypadku wybierz k najlepszych potomków i powtarzaj od kroku 2.

Algorytm cechuje losowo wygenerowany stan początkowy. Korzysta się z k niezależnych rozwiązań i wybiera lepsze z nich.

Jeżeli jakiś stan wygeneruje tylko dobrych potomków, a inny złych to na drodze wyboru tylko potomkowie dobrego będą się dalej rozwijać. Oznacza to znalezienie obiecującego ukształtowania płaszczyzny przystosowawczej i przyciągnięcie do niej.

Jest podatny na brak różnorodności.

Wariant stochastyczny: wybiera k potomków losowo z prawdopodobieństwem proporcjonalnym do ich funkcji celu.

Algorytmy genetyczne (genetic algorithms)

AG jest wariantem szukania wiązki z rozmnażaniem.

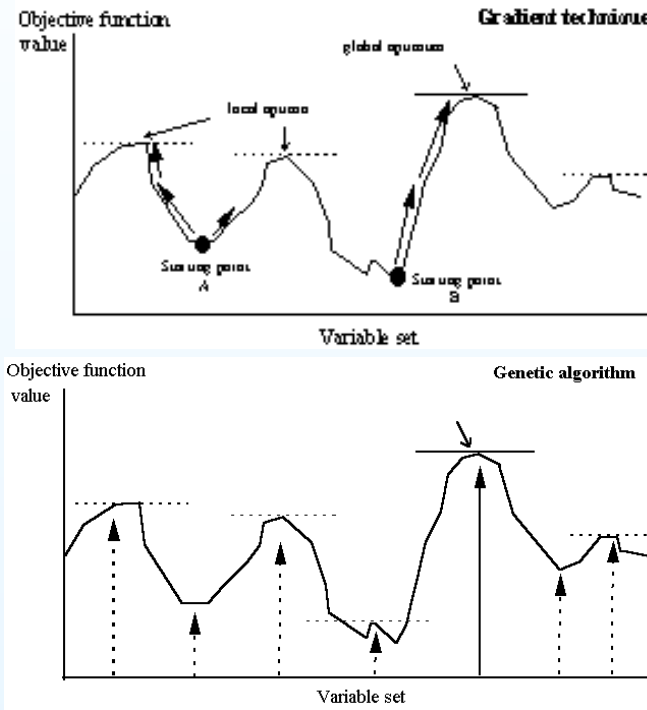
Oparty jest na darwinowskiej teorii doboru naturalnego: **przeżywają i rozmnażają się osobniki najlepiej przystosowane do warunków środowiska.**

Darwin zauważył, że na świat przychodzi dużo więcej potomstwa, niż może pomieścić środowisko zatem większość musi zginąć. Przeżywają tylko nieliczni, ale za to najlepsi. I to jest mechanizm zmian: selekcja - nieustająca, powolna, powszechna.

W procesie rozrodu osobniki przekazują potomstwu swoje własne cechy, jednocześnie w procesie **krzyżowania** cechy najsilniejszych osobników mieszają się, dając kombinacje dotąd nie występujące.

Ewolucja: Każdy organizm rozwija się w środowisku i w populacji osobników tego samego gatunku. Osobniki różnią się między sobą. Lepsze są w stanie się rozmnażać. Środowisko ocenia osobniki na podstawie ich cech zewnętrznych (fenotypu), które to zależą od zestawu genów osobnika.

Algorytm genetyczny - zalety



Rozpoczynając poszukiwanie w różnych (początkowo losowo wybranych punktach przestrzeni) daje większe szanse znalezienia globalnego optimum.

Generowanie nowych stanów nie odbywa się na podstawie określonych dla danego zdania funkcji generowania potomków, tylko za pomocą operatorów genetycznych.

Algorytm genetyczny - pojęcia

- **chromosom**: ciąg kodowy, reprezentuje stan z przestrzeni
- **gen**: cecha rozwiązania, stanu np. zmienna x
- **allel**: wariant cechy (przy kodowaniu binarnym 0 lub 1)
- **locus**: pozycja genu w chromosomie
- **genotyp**: struktura — stan złożony z kilku chromosomów
- **fenotyp**: zbiór parametrów, rozwiązanie alternatywne, zdekodowana struktura genotypu
- **funkcja przystosowania**: miernik zysku – ocena przydatności danego osobnika na podstawie jego chromosomu
- **mutacja**: zmiana allelu w chromosomie osobnika
- **krzyżowanie**: operacja rozmnażania
- **selekcja**: wybór osobników do następnej populacji

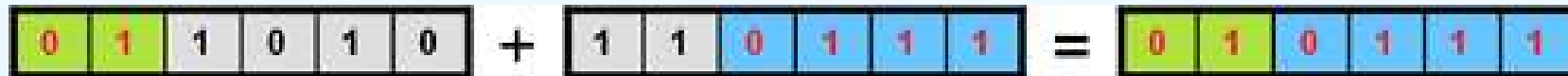
Algorytm genetyczny

```
1  function GENETIC_ALGORITHM( population, FITNESS-FN) return an individual
2      input: population, a set of individuals
3           FITNESS-FN, a function which determines the quality of the
4           individual
5
6      repeat
7          new_population = empty set
8          loop for i from 1 to SIZE(population) do
9              x = RANDOM_SELECTION(population, FITNESS_FN)
10             y = RANDOM_SELECTION(population, FITNESS_FN)
11             child = REPRODUCE(x,y)
12             if (small random probability) then child = MUTATE(child)
13             add child to new_population
14             population = new_population
15 until some individual is fit enough or enough time has elapsed
16 return the best individual
```

Algorytm genetyczny — funkcja reproduce

```
1 function REPRODUCE(x,y) return an individual
2   input: x, y --- parent individuals
3
4   n = LENGTH(x)
5   c = random number from 1 to n
6   return APPEND(SUBSTRING(x,1,c),SUBSTRING(y,c+1,n))
```

Przykład krzyżowania jednopunktowego chromosomów według powyższej procedury:



Co należy ustalić przed wykorzystaniem AG

- **Ustalenie kodowania:** ustalenie genomu jako reprezentanta wyniku
- **Ustalenie funkcji przystosowania:** jak oceniać osobniki?
- **Wybranie operatorów genetycznych:** jakie krzyżowanie, mutacja i selekcja
- **Ustalenie wartości współczynników stałych:** np. licznosc populacji, prawdopodobienstwo krzyżowania, prawdopodobienstwo mutacji, itp.
- **Warunek zatrzymania AG.**

Rodzaje Kodowania

Kodowanie — reprezentacja danych — zbiór stanów przestrzeni zadania przedstawiony w postaci skończonego alfabetu znaków.

- Podział ze względu na przyjmowane wartości kodowe:
 - binarne
 - całkowitoliczbowe
 - zmiennoprzecinkowe
 - ciągi znaków
- Podział ze względu na strukturę:
 - standardowe
 - permutacyjne
 - drzewiaste

Kodowanie heterogeniczne

Geny są heterogeniczne, tzn. geny na różnych pozycjach w chromosomie przechowują informacje różnego typu. Każda kodowana cecha fenotypu ma ściśle przypisaną na stałe pozycję w chromosomie. W wyniku krzyżowania i mutacji geny nie przemieszczają się, tylko zmieniają wartości

Przykłady:

- **Przykład problemu:** Problem plecakowy
Kodowanie: Każdy gen ma wartość 0 lub 1 i określa, czy element jest, czy go nie ma w plecaku. Długość chromosomu zależy od liczby elementów w zbiorze.
- **Przykład problemu:** Optymalizacja funkcji n zmiennych danej wzorem
Kodowanie: Każdy gen jest liczbą rzeczywistą i jest jedną z n zmiennych. Zazwyczaj kolejność jest zachowana x_1, x_2, \dots, x_n . Wartości x -ów zależą od dziedziny
- **Przykład problemu:** n -hetmanów
Kodowanie: Gen jest liczbą całkowitą od 1 do n i wskazuje wiersz położenia hetmana. Kolejne geny, to kolejne kolumny.

Kodowanie permutacyjne

Geny są homogeniczne, tzn. przechowują podobne informacje, są wymienne. W wyniku krzyżowania i mutacji geny nie zmieniają wartości, natomiast zmieniają pozycje.

Przykład problemu: Problem komiwojażera

Kodowanie: Chromosom podaje numery miast (kolejność) w jakiej komiwojażer przejechał trasę.

Chromosom	A	1	5	3	2	6	4	7	9	8
Chromosom	B	8	5	6	7	2	3	1	4	9

Poprawność kodowania

Kodowanie jest bardzo istotnym etapem projektowania algorytmu. Sposób kodowania wydatnie wpływa na szybkość i jakość znajduwanych wyników, na sposób w jaki przeszukiwana jest przestrzeń rozwiązań. Złe kodowanie może spowodować, że nigdy nie zostanie przeszukany fragment przestrzeni, w którym znajdują się najlepsze rozwiązania.

D - zbiór fenotypów

G - zbiór genotypów

$c : D \rightarrow G$ - funkcja kodująca

Kodowanie powinno zapewnić:

$$\forall d \in D \exists g \in G c(d) = g$$

Każde rozwiązanie zadania można przedstawić jako genotyp i kodowanie nie wprowadza dodatkowych ekstremów lokalnych.

Funkcja przystosowania

Funkcja przystosowania (ang. FF — fitness function) przypisuje każdemu chromosomowi pewną wartość liczbową, która mówi jak daleko od rozwiązania jest badany osobnik (f. heurystyczna) $f(ch_i)$, gdzie ch_i jest i -tym chromosomem w populacji. FF podaje kryteria oceny osobników.

FF jest zawsze zależna od rozwiązywanego problemu!

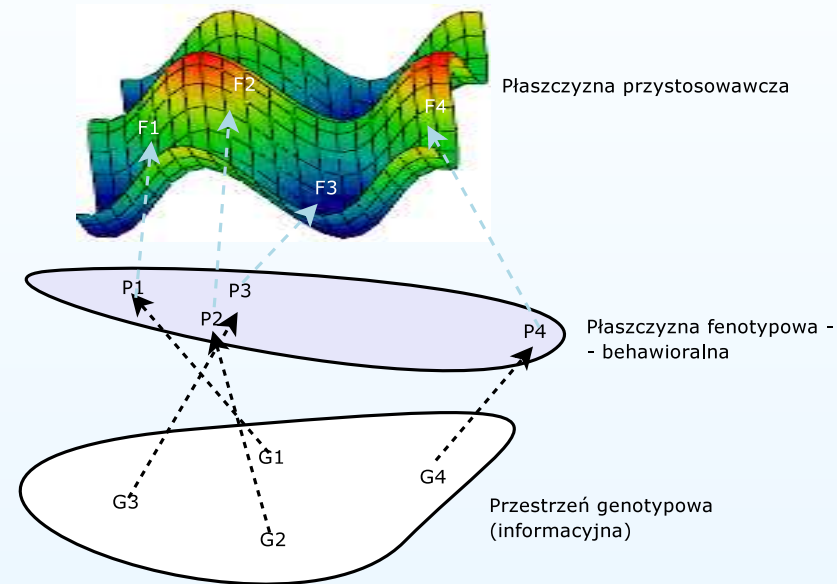
Wartość $f(ch)$ jest zazwyczaj maksymalizowana. Można zadanie minimalizacji przekształcić na zadanie szukania maksimum.

Wyższa wartość funkcji oznacza lepsze cechy i osobnik (rozwiązanie) ma większe szanse przejścia do następnego pokolenia. (Darwinowska przeżywalność najlepiej przystosowanych).

Przykłady:

- Problem komiwojażera: $f(ch_i) = 1 / \sum_{j=1}^{n-1} dist(ch_{ij}, ch_{i,j+1})$
- Problem plecakowy: $f(ch_i) = \sum_{j=1}^n value(ch_{ij})$ dla $f(ch_i) < Capacity$, lub $f(ch_i) = 0$
- Funkcja zadana wzorem: $f(ch_i) = funkcja$

Krajobraz przystosowawczy



Topografia adaptacyjna (krajobraz przystosowawczy - fitness landscape) to odwzorowanie fenotypu na płaszczyznę przystosowania (zależy od wymiarowości zadania).

Ewolucja przesuwa populację w kierunku szczytu krajobrazu.

Metody selekcji

Jak wybierać chromosomy z istniejącej populacji, by zapewnić Darwinowską ewolucję?

- Selekcja metodą ruletki
- Metoda rankingowa
- Turniej
- **Elitaryzm**: jako dodatkowy element do wymienionych selekcji. Chromosom o największej wartości funkcji przystosowania jest zawsze kopiowany do następnej generacji.

Selekcja metodą ruletki

Powtarzaj wielkość populacji razy:

1. Oblicz sumę ($S = \sum_{i=1}^N f(x_i)$) przystosowania wszystkich osobników w populacji, gdzie N - licznosc populacji.
2. Wygeneruj losowa liczbe (R) z przedzialu 0 do S
3. Sumuj kolejno po $S_c = \sum_j f(x_j)$ kolejnych chromosomow. Gdy $S_c > R$, to chromosom j przepiszesz do nowej populacji.

Wady: Gdy wartosci funkcji przystosowania sa bardzo rozne, to moze sie zdarzyc, ze jeden chromosom zajmie duza przestrzen z kola. Inne beda mialy mala szanse przejscia reprodukcji. Nadaje sie dla FF maksymalizowanej.

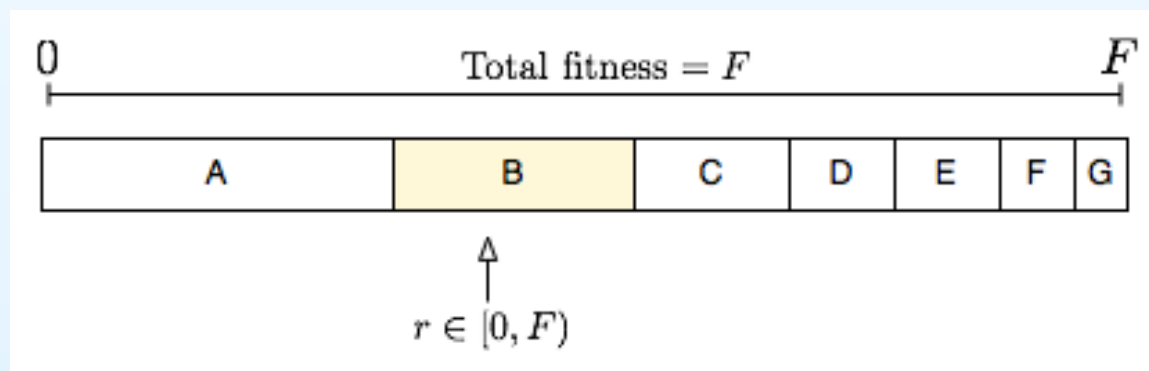


Image:en.wikipedia.org

Selekcja rankingowa

Selekcja rankingowa nadaje rangi chromosomom w populacji na podstawie ich przystosowania. Najgorszy dostaje rangę 1, przedostani 2 a najlepszy N jeżeli N jest licząnością populacji.

W korku następnym traktuje się ranking jako wartości oszacowania użyteczności chromosomów i uruchamia się metodę ruletki.

Cechy:

Rozwiązuje problem ruletki, czyli daje szansę chromosomom o małej FF na wylosowanie. Ma jednak słabą zbieżność, bo najlepszy chromosom różni się od innych chromosomów nieznacznie.

Selekcja: turniej

Powtarzaj wielkość populacji razy:

1. wybierz losowo k (rozmiar turnieju) osobników z populacji
2. wybierz najlepszego osobnika z grupy turniejowej z prawdopodobieństwem p
3. wybierz drugiego osobnika z grupy turniejowej z prawdopodobieństwem $p(1 - p)$
4. wybierz trzeciego osobnika z grupy turniejowej z prawdopodobieństwem $p(1 - p)^2$
itd.

Deterministyczny turniej następuje, gdy wybiera się najlepszego osobnika z $p = 1$.
Jeżeli $k = 1$, to wybór nowej populacji jest losowy.

Wybrany do nowej populacji osobnik zazwyczaj pozostaje w populacji, co daje mu szansę ponownego wylosowania.

Zalety: Sprawdza się w implementacjach równoległych, łatwo może manipulować presją selekcyjną (skłonność algorytmu do częstszego wybierania lepiej przystosowanych osobników).

Metody krzyżowania (reprodukcji)

Dla chromosomów heterogenicznych.

- **Jednopunktowe**: Losowo wybierz z populacji 2 rodziców. Wylosuj punkt krzyżowania. Rozmnoż osobniki poprzez zamianę ciągów od punktu krzyżowania.
- **Dwupunktowe**: Wylosuj dwóch rodziców z populacji. Wylosuj dwa punkty krzyżowania. Rozmnoż osobniki poprzez zamianę ciągów pomiędzy punktami krzyżowania.
- **Jednorodne**: Wylosuj dwóch rodziców do krzyżowania. Dla każdego genu losowo określ czy nastąpi zamiana genów w chromosomach czy też nie.
- **Krzyżowanie uśredniające**: (dla chromosomów o wartościach rzeczywistych). Dla rodziców x_1 oraz x_2 , z rozkładem jednostajnym losujemy liczbę α . Wartości potomków: $y_1 = x_1 + \alpha(x_2 - x_1)$ i $y_2 = x_2 + (x_1 - y_1)$.

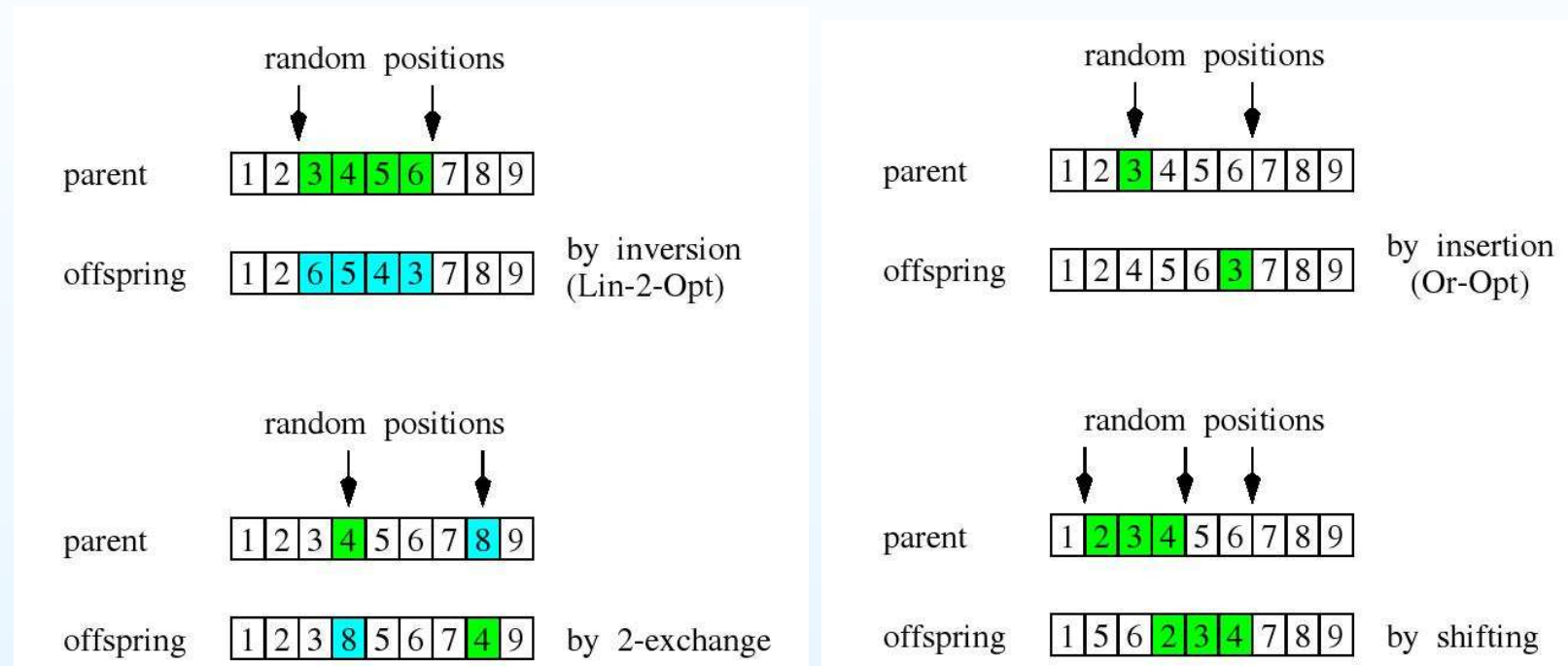
Metody mutacji

Dla chromosomów heterogenicznych.

- **Jednopunktowa**: Wylosuj osobnika do mutacji. Wylosuj punkt mutacji. Zmień wartość genu na dozwoloną wartość z alfabetu
- **Na określonej długości**: Wylosuj osobnika do mutacji. Wylosuj punkt mutacji i długość zmutowanego fragmentu. Zmień wartości genów na dozwolone wartości z alfabetu.
- **Dla chromosomów o wartościach rzeczywistych**. Losujemy liczbę α . Wartość po mutacji: $y = x_1 + \alpha x_1$.

Metody mutacji

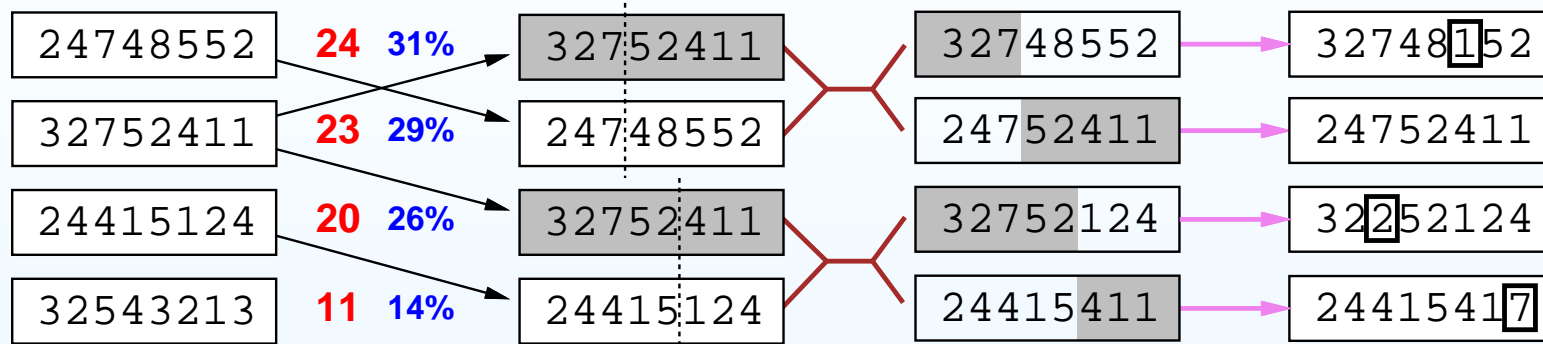
Dla chromosomów homogenicznych. Fitness zależy od kolejności elementów osobnika.



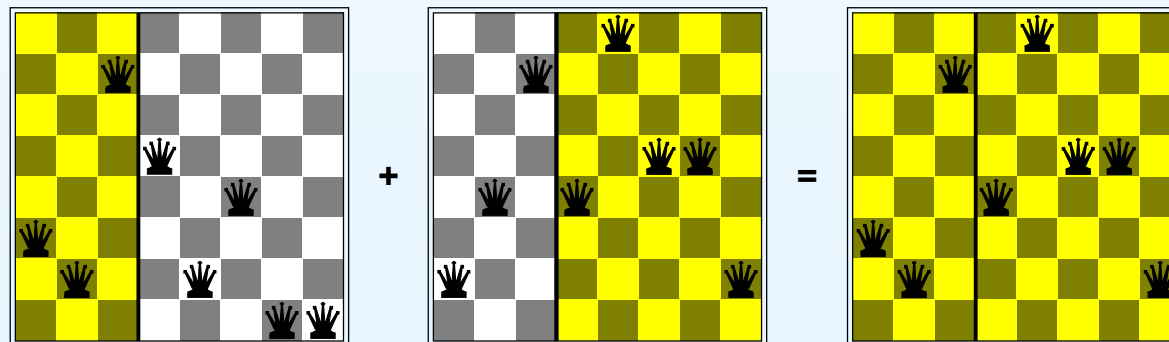
Neighborhood — liczba stanów możliwych do osiągnięcia ze stanu rodzicielskiego w jednym kroku. Ten parametr jest siłą mutacji w przestrzeni permutacyjnej.

Sąsiedztwo zazwyczaj stałe — stąd zmiennym paramentrem jest liczba mutacji przeprowadzana w pojedynczym kroku. $1 < s < const \cdot N$

Przykład



Fitness Selection Pairs Cross-Over Mutation



FF = liczba nie atakujących się par.
 Najlepsze rozwiązanie FF=28.

Warunki zakończenia AG

Generowanie nowych populacji odbywa się w cyklu, który może się zakończyć, gdy spełniony zostanie warunek zakończenia. Warunek ten może być pojedynczy lub złożony. Do najczęściej stosowanych należą:

- Znalezione rozwiązanie optymalne lub suboptymalne
- Osiągnięto założoną liczbę generacji
- Przekroczono założony budżet (czas obliczeniowy/zysk)
- W kolejnych iteracjach nie uzyskuje się polepszenia najlepszego osobnika lub średniego przystosowania populacji
- Kontrola manualna

Parametry od których zależy efektywność AG

Pewne wartości sterujące zachowaniem metod selekcji, krzyżowania i mutacji są określane przed uruchomieniem AG i zazwyczaj pozostają niezmiennie. Do nich zaliczyć można np.:

- Liczność populacji
- Prawdopodobieństwo krzyżowania
- Prawdopodobieństwo mutacji
- Liczność grupy turniejowej

Wielkości wymienionych parametrów są najczęściej dobierane empirycznie. Ponadto rezultaty działania AG zależą od stosowanych selekcji i operatorów zmian kodu.

Ocena AG

Zalety:

- Szerokie zastosowanie w rozwiązywaniu problemów, dla których nie zawsze znane są dobre inne techniki.
- Może być uruchamiany interaktywnie (zmiana parametrów).
- Niewielki (pozornie) koszt adaptacji algorytmu dla nowego problemu czy nowej przestrzeni stanów.
- Rozwiązania są łatwo interpretowane.
- Dobre wyniki w poszukiwaniu lokalnych optimum dzięki operowaniu na genotypie, a nie fenotypie.

Trudności z AG:

- Znalezienie odpowiedniej reprezentacji rozwiązania.
- Odpowiedni dobór parametrów i operatorów (liczność populacji, prawdopodobieństwa krzyżowania i mutacji).
- Czas uzyskania satysfakcjonującego rozwiązania.