

Podstawy sztucznej inteligencji

wykład 3

Lokalne metody optymalizacji

Joanna Kołodziejczyk

10 listopad 2011

Plan wykładu

- 1 Lokalne metody przeszukiwania (optymalizacji)
 - Ogólne cechy metod
 - Hill-climbing
- 2 Globalne metody przeszukiwania (optymalizacji)

Jakie problemy wynikają z metod przeszukiwania

Duże wymagania pamięciowe

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

Przeformułowanie celu

W wielu problemach takich jak: projektowanie układów cyfrowych, harmonogramowanie, wyznaczanie drogi pojazdu, optymalizacja sieci telekomunikacyjnych itd. nie znane jest rozwiązanie oczekiwane (stan końcowy). Natomiast ważne jest znalezienie **rozwiązania (stanu, ścieżki)** o jak najlepszych walorach.

Cechy metod lokalnych

Lokalne metody przeszukiwania operują na pojedynczym stanie z pewnej przestrzeni 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 (spośród wielu kandydatów) według **funkcji celu**, która jest funkcją heurystyczną.

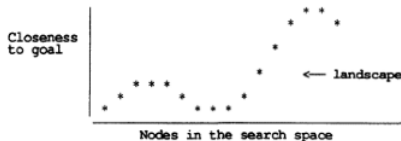
Problemy, w których wykorzystuje się metody lokalne

- pokrycie wierzchołków w grafie (odnaleźć rozwiązanie z najmniejszą liczbą węzłów)
- problem komiwojażera (cykl w grafie o najkrótszej długości cyklu)
- problem spełnialności w rachunku zdań (znalezienie wszystkich klauzul spełniających pewne zadane przypisanie)
- Nurse scheduling problem (NSP) (plan zmian z uwzględnieniem zbioru ograniczeń)
- problemy klasteryzacji (znajdowanie najlepszego dopasowania)

Krajobraz przestrzeni stanów

Krajobraz przestrzeni stanów (landscape) posiada:

- położenie (oś X) — definiowane przez stan
- wzniesienia (oś Y) — zdefiniowane przez wartość funkcji celu (funkcji heurystycznej) oceniającej „closeness to goal”.



Każdy stan jest bardzo często równy „pełnym konfiguracjom”, które można traktować jako potencjalne rozwiązanie problemu, a zadanie dla strategii polega na poszukiwaniu optymalnej konfiguracji/rozwiązania.

Metoda największego wzrostu - Hill climbing

- Metoda próbuje maksymalizować/minimalizować funkcję $f(n)$, gdzie n jest stanem dyskretnym.
- Hill climbing będzie przechodził od węzła do węzła lokalnie zwiększając/zmniejszając wartość funkcji f .
- Czysty hill-climbing nie pozwala na cofanie się do stanów już odwiedzonych, tylko wybiera najlepszego w danej chwili potomka (spośród wszystkich wygenerowanych dla danego węzła) i zapomina o innych. Co jest zaletą, bo zużywa się mało pamięci, ale jest wadą bo najczęściej nie jest optymalny.
- Algorytm może być stosowany dla przestrzeni ciągłych i ta jego odmiana jest nazywana metodą gradientową.

Przykład dla n hetmanów

Założenia

- W każdej kolumnie znajduje się jeden hetman. Zadanie sprowadza się wówczas do znalezienia dla każdego z nich odpowiedniej kolumny.
- Funkcja generująca potomka przesuwa pojedynczego hetmana w tej samej kolumnie.
- Ocena stanu przez funkcję $f(n)$ liczącą liczbę par hetmanów, które się atakują.
- Oczekuje się zmniejszania wartości $f(n)$ w miarę poszukiwania.

Przykład dla n hetmanów

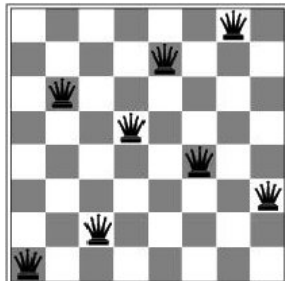
Stan początkowy (Rys. lewy)

$f(n) = 17$ i w polu podane wartości f dla każdego możliwego potomka, który powstanie przez przesunięcie hetmana w odpowiednie pole.

Stan końcowy (Rys. prawy)

minimum lokalne o wartości $f(n) = 1$.

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

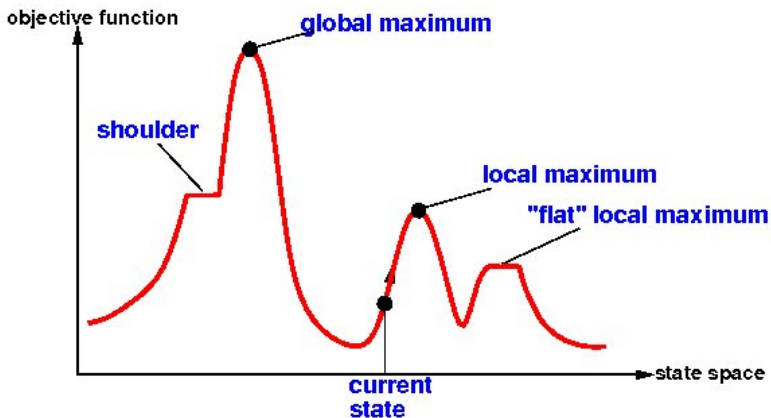


Wady algorytmu największego wzrostu

- Lokalne maksima — jest to szczyt w krajobrazie przestrzeni stanów, który jest niższy niż globalne optimum. Algorytm lubi utykać w takich miejscach.
- Grzbiet (flat local maximum) — sekwencja lokalnych maksimów. Wszystkie punkty powstałe przez ograniczony zestaw operatorów są na tym samym poziomie lub niżej (mają tę samą wartość funkcji f).
- Obszary płaskie (shoulder) — obszary przestrzeni przystosowawczej, gdzie na dużym obszarze wartość funkcji celu jest stała lub słabo zmienna. Algorytm może nie znaleźć wyjścia z obszaru (błądzenie).

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

Wady (krajobraz przestrzeni stanów)



Warianty algorytmu metody największego wzrostu (1)

Stochastyczna metoda największego wzrostu

Wybiera losowo pomiędzy ścieżkami wrostającymi, przy czym prawdopodobieństwo wyboru zależy od nachylenia zbocza. Im bardziej strome, tym większe prawdopodobieństwo.

Metoda największego wzrostu pierwszego wyboru (simple H-C)

Generuje potomków tak długo, dopóki nie znajdzie stanu (potomka) lepszego od bieżącego.

Losowe przemieszczenia

Jest to wariant predykcyjny, który na płaskich obszarach próbuje generować potomków w przód (liczba kroków w przód jest z góry zadana) i odszukać narastające zbocza.

Warianty algorytmu metody największego wzrostu (2)

Losowo restartowana metoda największego wzrostu (Shotgun H-C)

- Próbuje pokonać problem utykania w lokalnych minimach.
- Iteracyjnie wykonuje H-C za każdym razem startując z nowego stanu początkowego x_0 .
- Najlepsz x_m jest przechowywany: jeżeli kolejne uruchomienie wyprodukuje lepsze x_m to zostaje ono podmienione.
- Metoda jest zadziwiająco dobra, gdyż zazwyczaj lepiej wykorzystać czas na obliczenia eksplorując szerzej przestrzeń stanów niż próbować optymalizować punkt startu.

Plan wykładu

- 1 Lokalne metody przeszukiwania (optymalizacji)
- 2 Globalne metody przeszukiwania (optymalizacji)
 - Cechy metod globalnych
 - Algorytmy ewolucyjne

Globalna optymalizacja

Lokalna a globalna?

Najczęstszą formą jest optymalizacji wyrażania zadania optymalizacji jest minimalizacja funkcji f o zmiennych rzeczywistych w przestrzeni parametrów $\vec{x} \in P$. Zazwyczaj dany jest zbiór ograniczeń na wektor rozwiązań \vec{x}_{min} .

W problemach rzeczywistych funkcje wielu zmiennych mają dużą liczbą minimów i maksimów lokalnych. Znajdowanie dowolnego lokalnego optimum jest w miarę łatwo osiągalne przez metody lokalne. Znajdowanie globalnego optimum jest znacząco bardziej wymagające i dla wielu problemów do dnia dzisiejszego niewykonalne.

Ograniczenia na dziedzinę (parametry) lub przeciwdziedzinę (wartości funkcji)

Rozważmy zadanie minimalizacji funkcji Rosenbrocka

$$f(x) = 100(x_2 + x_1^2)^2 + (1 + x_1)^2$$

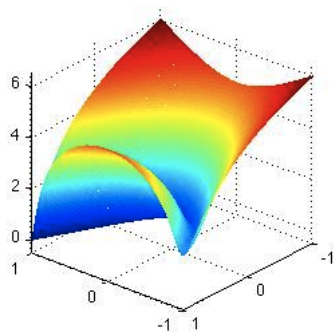
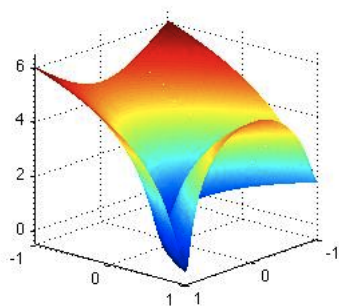
w przestrzeni ograniczonej przez koło o promieniu 1 ze środkiem w początku układu współrzędnych.

Czyli należy znaleźć minimum funkcji $f(x)$, gdy parametry funkcji spełniają nierówność:

$$x_1^2 + x_2^2 \leq 1$$

Zatem jest to problem optymalizacji funkcji nieliniowej z nieliniowym ograniczeniem.

Funkcja Rosenbrocka



Przykład zadania optymalizacji z wieloma funkcjami celu

Optymalizacja wielokryterialna to często znalezienie kompromisu pomiędzy sprzecznymi celami. Np. patrząc na pewien zakład produkcyjny możemy określić następujące kryteria optymalizacji:

- 1 minimalizacja czasu pomiędzy przyjęciem zlecenia a dostawą
- 2 maksymalizacja zysków
- 3 minimalizacja kosztów reklamy, zatrudnienia personelu, zużycia materiałów itp.
- 4 maksymalizacja jakości produktu
- 5 minimalizacja negatywnego wpływu na środowisko naturalne

Pewne zależności pomiędzy celami są od razu widoczne, ale wiele jest ukrytych i dopiero wnikliwa analiza danych może wskazać pewne ukryte zależności.

Taksonomia metod optymalizacji <http://www.it-weise.de/projects/book.pdf>

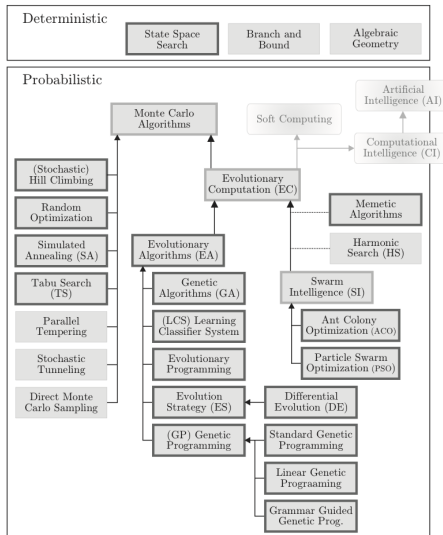


Figure 1.1: The taxonomy of global optimization algorithms.

Podstawy — dobór naturalny

Dobór naturalny

Podstawowy mechanizm ewolucji. Proces, który prowadzi do ukierunkowanych zmian w populacji. Dzięki temu zwiększa się średnie przystosowanie populacji do środowiska.



Ewolucja

Organizm rozwija się w środowisku i w populacji osobników tego samego gatunku. Osobniki różnią się między sobą. Środowisko ocenia osobniki na podstawie ich cech zewnętrznych (fenotypu), które to zależą od zestawu genów osobnika.

Selekcja, krzyżowanie

Selekcja

jest nieustająca, powolna, powszechna. Darwin zauważył, że na świat przychodzi więcej potomstwa, niż może pomieścić środowisko zatem większość musi zginąć. Przeżywają tylko nieliczni, ale za to najlepiej przystosowani do środowiska.

Krzyżowanie

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

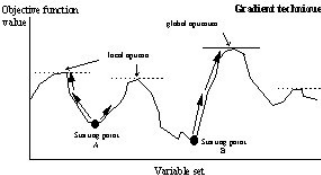
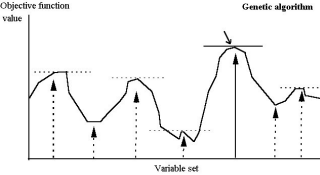
Algorytm ewolucyjny

Jest to metoda rozwiązywania problemów optymalizacji bez/z ograniczeniami jedno/wielokryterialnych w oparciu o selekcję naturalną.

Cechy algorytmu

- Wielokrotnie zmienia populację osobników (reprezentacja rozwiązania problemu).
- Losowo wybiera z populacji osobniki, które staną się rodzicami tworzącymi nowe potomne osobniki do następnej generacji osobników.
- Z każdą następną generacją populacja ewoluuje w kierunku rozwiązania optymalnego.
- Można rozwiązywać problemy nierozwiązywalne innymi metodami: nieciągła, nieróżniczkowalna, stochastyczna lub silnie nieliniowa funkcja celu.

Porównanie z innymi metodami optymalizacji

Algorytm klasyczny	Algorytm ewolucyjny
 <p>The graph shows a jagged line representing an objective function. A path starts at a point labeled 'Start point A' and moves towards a 'local optimum' (a smaller peak). From there, it moves to a 'global optimum' (the highest peak). The path is labeled 'Gradient technique'.</p>	 <p>The graph shows the same jagged line. Multiple paths, represented by dashed lines and arrows, start from various points and explore the search space, eventually finding the 'global optimum'. The process is labeled 'Genetic algorithm'.</p>
<p>W każdym kroku tworzy pojedynczy punkt przestrzeni. Sekwencja punktów prowadzi do rozwiązania (sub)optimalnego.</p>	<p>W każdej iteracji tworzy populację punktów. Najlepszy punkt z populacji prowadzi do rozwiązania (sub)optimalnego</p>
<p>Wybiera kolejny punkt w ciągu deterministycznie.</p>	<p>Losowo wybiera osobniki do następnej populacji.</p>
<p>Funkcja generowanie potomków wynika z problemu</p>	<p>Funkcja generowania potomków jest niezależna od problemu (operatory genetyczne)</p>

Terminologia algorytmów ewolucyjnych

Funkcja przystosowania (celu)

Jest to funkcja, która jest optymalizowana np.

$$f(x_1, x_2, x_3) = (2x_1 + 1)^2 + (3x_2 + 4)^2 + (x_3 - 2)^2$$

Dla klasycznych algorytmów optymalizacji jest to funkcja celu.

Funkcja ta mierzy jak dobry (użyteczny) jest osobnik jako rozwiązanie danego problemu.

Osobnik/chromosom/gen

Osobnik jest to punkt w przestrzeni (potencjalne rozwiązanie problemu). Osobnik ma postać zakodowanego ciągu o określonej długości zwanego chromosomem. Poszczególne elementy chromosomu to geny.

Na przykład wektor $(2, -3, 1)$ jest osobnikiem z genami 2, -3, 1.

Wartość funkcji przystosowania dla tego osobnika to

$$f(2, -3, 1) = 51.$$

Terminologia algorytmów ewolucyjnych cd.

Populacja

Populacja to tablica osobników, np. jeżeli liczność populacji jest 100, a długość osobnika 3, to populacja jest reprezentowana przez tablicę 100×3 .

Ten sam osobnik może pojawić się w populacji więcej niż raz.

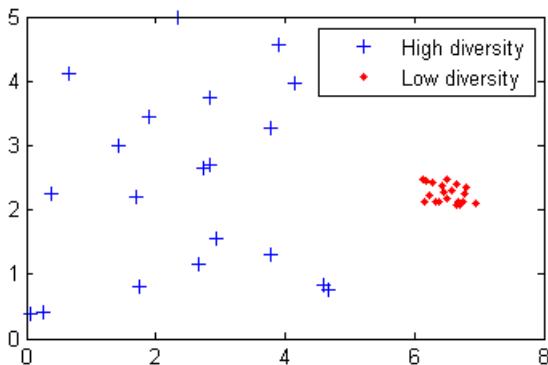
Generacja

W każdej iteracji algorytm wykonuje serię operacji, obliczeń na osobnikach i populacji. W wyniku tego zmieniają się osobniki i powstaje **nowa populacja**. Każda kolejna populacja nazywana jest nową generacją.

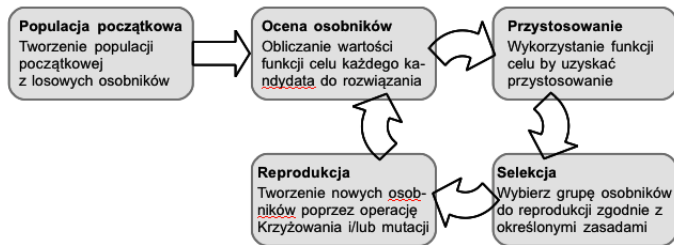
Terminologia algorytmów ewolucyjnych cd.

Zróżnicowanie populacji

Jest to parametr określający średnią odległość pomiędzy osobnikami w populacji. Duża różnorodność, to duża wartość średniej. Mała różnorodność, to niewielka odległość pomiędzy osobnikami.



Algorytm ewolucyjny — schemat działania



Rodzice/dzieci

Rodzice to osobniki wybrane do reprodukcji służą do tworzenia nowych osobników nazywanych dziećmi. Algorytm ma tendencje do wybierania jako rodziców tych osobników, które mają wyższe przystosowanie.

Dopasowanie algorytmu ewolucyjnego do rozwiązywanego problemu

Zanim przystąpi się do rozwiązywania problemu z użyciem AE należy określić niezbędne do uruchomienia algorytmu parametry.

- 1 **Ustalenie kodowania:** ustalenie chromosomu jako reprezentanta wyniku.
- 2 **Ustalenie funkcji przystosowania:** Co jest celem optymalizacji? Jak oceniać osobniki?
- 3 **Wybranie operatorów genetycznych:** Jakie stosować krzyżowanie, mutacje i selekcje?
- 4 **Ustalenie wartości pewnych parametrów:** np. licznosc populacji, prawdopodobienstwo krzyzowania, prawdopodobienstwo mutacji, itp.
- 5 **Warunek zatrzymania algorytmu.** Kiedy zatrzymać cykl generowania nowych populacji?

Rodzaje chromosomów stosowane w AE

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

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

Kodowanie heterogeniczne

Geny heterogeniczne

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

- **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

Geny 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. W przykładzie poniżej ChromosomA opisuje trasę przejazdu z miasta 1 do 5 potem do 3 itd.

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.

Zdefiniowanie funkcja przystosowania

Funkcja przystosowania (ang. FF — fitness function)

FF przypisuje każdemu chromosomowi pewną wartość liczbową, która mówi jak daleko od rozwiązania jest badany osobnik lub jak użyteczny jest dany osobnik. Jest to funkcja heurystyczna.

FF jest zawsze zależna od rozwiązywanego problemu!

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

Przykłady funkcji przystosowania

$f(ch_i)$ — funkcja przystosowania dla i -tego chromosomu.

Problem komiwojażera (minimalizacja drogi)

$$f(ch_i) = \sum_{j=1}^{n-1} \text{dist}(ch_{ij}, ch_{ij+1})$$

Problem plecakowy (maksymalizacja zysku przy ograniczeniu)

$$f(ch_i) = \begin{cases} \sum_{j=1}^n \text{value}(ch_{ij}) & \text{if } f(ch_i) < \text{Capacity}; \\ f(ch_i) = 0 & \text{otherwise.} \end{cases}$$

Funkcja zadana wzorem

$f(ch_i) = \text{funkcja np.}$

$$f(x_1, x_2, x_3) = (2x_1 + 1)^2 + (3x_2 + 4)^2 + (x_3 - 2)^2$$

Metody selekcji

Selekcja

Jest to metoda, która wybiera osobniki do reprodukcji. Powinna zapewnić zbieżność do optimum. Wybieranie zawsze i tylko najlepszych nazywa się dużą presją selekcyjną i takie metody nie zawsze dają dobre rozwiązania (utykanie w lokalnych optimach).

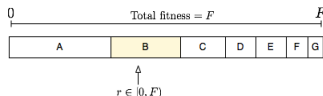
- Selekcja metodą ruletki
- Metoda rankingowa
- Selekcja turniejowa
- **Elitaryzm**: stosuje się jako dodatkowy element do metod selekcji. Chromosom(y) o największej wartości funkcji przystosowania jest zawsze kopiowany do następnej generacji, przez co nie można go utracić.

Selekcja metodą ruletki

Algorytm

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 przedziału 0 do S
- 3 Sumuj kolejno po $S_c = \sum_j f(x_j)$ kolejnych chromosomów. Gdy $S_c > R$, to chromosom j przepisz do nowej populacji.



Cechy/wady

Może się zdarzyć, że jeden chromosom zajmie dużą przestrzeń z koła, zdominuje następną generację. Nadaje się dla FF maksymalizowanej.

Selekcja rankingowa

Algorytm

- 1 Nadaj rangi chromosomom w populacji na podstawie ich przystosowania. Najgorszy dostaje rangę 1, przedostatni 2 a najlepszy N , gdzie N jest licznością populacji.
- 2 Traktuj ranking jako wartości oszacowania użyteczności chromosomów i urucham metodę ruletki na tych wartościach.

Cechy/wady

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 turniejowa

Algorytm

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 $p(1 - p)$, trzeciego osobnika z grupy turniejowej z prawdopodobieństwem $p(1 - p)^2$ itd.

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

Cechy/wady

Sprawdza się w implementacjach równoległych, łatwo może manipulować presją selekcyjną. FF może być min/maksymalizowana.

Metody krzyżowania dla chromosomów heterogenicznych

W wyniku krzyżowania powstają dwa osobniki, które zazwyczaj podmieniają swoich rodziców w populacji.

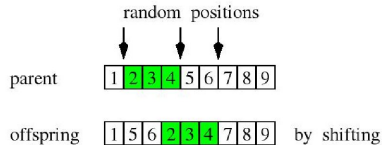
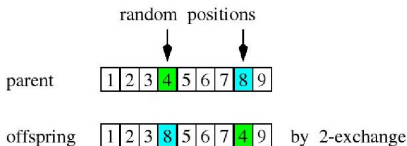
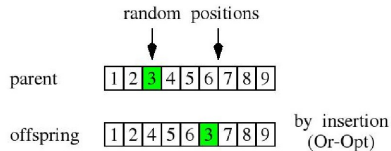
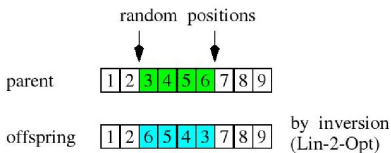
- **Jednopunktowe:** Losowo wybierz z populacji 2 rodziców. Wylosuj punkt krzyżowania. Rozmnoż osobniki poprzez zamianę ciągów od punktu krzyżowania.
- **Dwupunktowe:** Wylosuj dwóch rodziców z populacji. Wylosuj dwa punkty krzyżowania. Rozmnoż osobniki poprzez zamianę ciągów pomiędzy punktami krzyżowania.
- **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

W wyniku mutacji powstaje nowy chromosom, który najczęściej podmienia w populacji chromosom z którego powstaje.

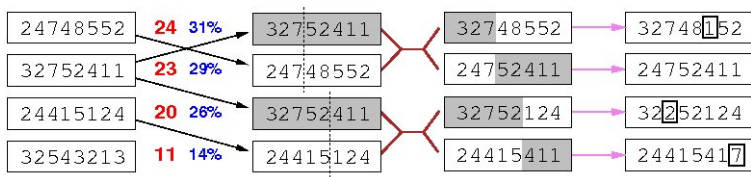
- **Jednopunktowa**: Wylosuj osobnika do mutacji. Wylosuj punkt mutacji. Zmień wartość genu na dozwoloną/ wylosowaną wartość z alfabetu.
- **Na określonej długości**: Wylosuj osobnika do mutacji. Wylosuj punkt mutacji i długość zmutowanego fragmentu. Zmień wartości genów na dozwolone/losowe wartości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

Metody mutacji dla chromosomów homogenicznych

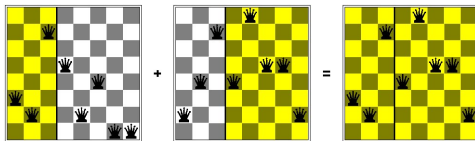


Przykład działania algorytmu ewolucyjnego dla n-hetmanów

FF to liczba nie atakujących się par. Najlepsze rozwiązanie FF=28.
 Selekcja metodą ruletki. Krzyżowanie, mutacja jednopunktowe.



Fitness **Selection** **Pairs** **Cross-Over** **Mutation**



Warunki zatrzymania AE

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:

- Znaleziono 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 AE

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

- licznosc populacji
- prawdopodobienstwo krzyzowania
- prawdopodobienstwo mutacji
- licznosc grupy turniejowej.

Wielkości wymienionych parametrów są najczęściej dobierane empirycznie. Ponadto rezultaty działania AE zależą od stosowanych selekcji i operatorów reprodukcji.

Ocena algorytmów ewolucyjnych

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.
- Dobre wyniki w poszukiwaniu optimów dzięki operowaniu na genotypie, a nie fenotypie.

Trudności

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