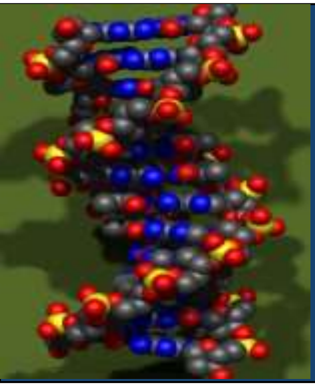
Wydział Zarządzania AGH
Katedra Informatyki Stosowanej

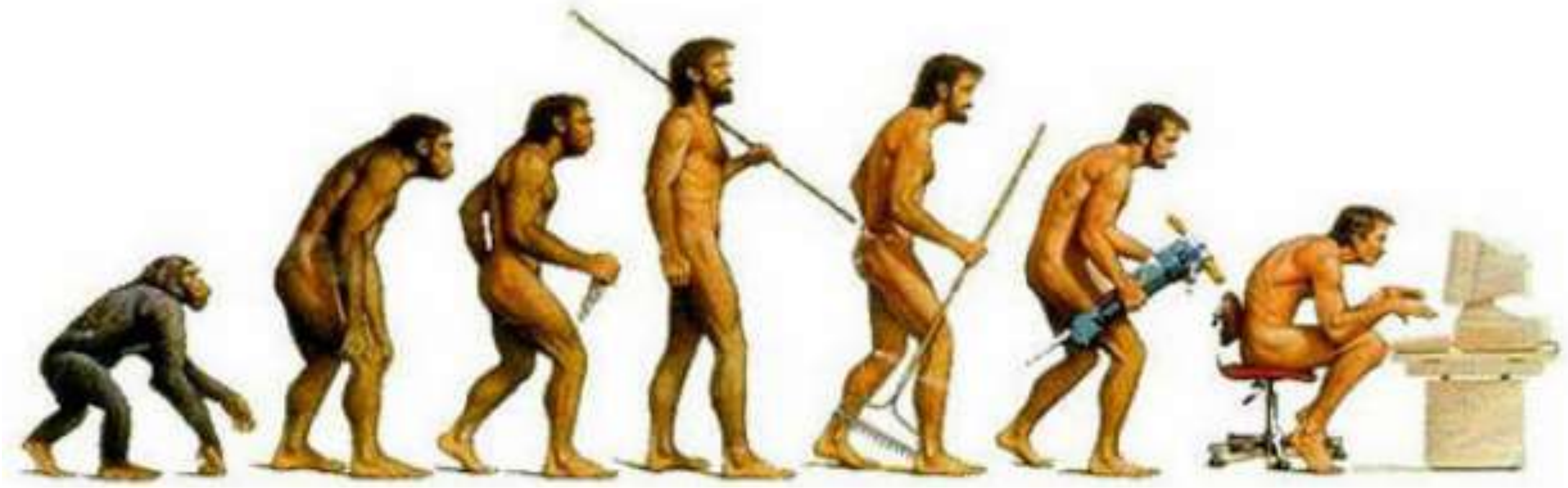


Algorytm genetyczny

Systemy wspomagania decyzji

Treść wykładu

- Wprowadzenie
- Zasada działania
- Podział EA
- Algorytm genetyczny
- Przykłady



EA - wprowadzenie

- Algorytmy ewolucyjne (ang. Evolutionary Algorithms - EA) są szeroko stosowaną heurystyką przeszukiwania i optymalizacji opartą na zasadach przejętych z teorii ewolucji. Naturalność oraz prostota działania sprawiły, że algorytmy te są chętnie wykorzystywane w naukach zarządzania do rozwiązywania problemów optymalizacji kombinatorycznej, a w szczególności - do szeroko rozumianych problemów alokacji zasobów.
- Algorytmy ewolucyjne nie gwarantują znalezienia optimum globalnego, jednak generalnie zapewniają znalezienie rozwiązania wystarczająco dobrego w akceptowalnym przedziale czasu. Stąd głównym zastosowaniem tych algorytmów powinny być problemy, dla których nie istnieją techniki specjalizowane. Nawet jeśli techniki takie istnieją, można osiągnąć poprawę ich działania poprzez ich połączenie z algorytmami ewolucyjnymi.

Zasada działania

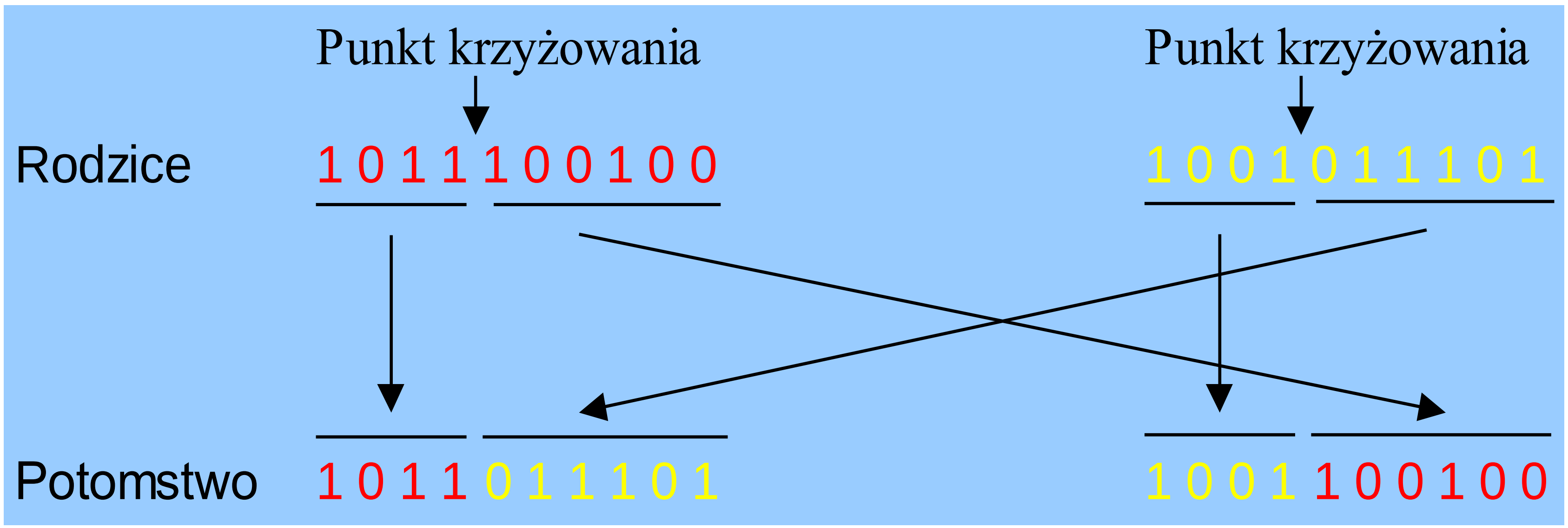
- Drogą rozwoju przyrody ożywionej jest ewolucja czyli metoda prób i błędów oparta na doborze naturalnym. Najlepiej udokumentowanym mechanizmem doboru naturalnego jest proces genetyczny: można go postrzegać jako proces optymalizacyjny, w którym osobniki najlepiej dostosowane do środowiska mają największe szanse przeżycia i utworzenia potomków. Nośnikiem informacji o cechach indywidualnych osobnika są geny - bloki DNA - tworzące chromosomy: kod ten determinuje budowę osobnika i jego rozwój, a w szczególności - jego dopasowanie do środowiska naturalnego. Istnieje silna zależność między chromosomem osobnika a jego żywotnością i zdolnością do przekazywania genotypu kolejnym pokoleniom.

Zasada działania

- Ewolucyjny rozwój populacji chromosomów odbywa się poprzez mechanizm reprodukcji, na który składają się procesy krzyżowania (ang. crossover), mutacji (ang. mutation) i inwersji (ang. inversion). W procesie krzyżowania, z dwóch chromosomów rodzicielskich wybierane są geny, które po zespoleniu tworzą jeden lub więcej chromosomów potomnych. W procesie mutacji dochodzi do przekłamania kodu poprzez zmianę jednego genu lub ich ciągu, natomiast inwersja odwraca fragment chromosomu. Przy pomocy tych mechanizmów tworzą się kolejne pokolenia (populacje chromosomów), zawierające coraz "doskonalsze" osobniki.

Zasada działania

Krzyżowanie osobników



Zasada działania

Mutacja i inwersja

Punkt mutacji

Potomek 1 0 1 1 0 1 1 1 0 1

Zmutowany potomek 1 0 1 1 0 0 1 1 0 1

Punkty inwersji

Potomek 1 0 1 1 0 1 1 1 0 1

Potomek po inwersji 1 0 1 1 1 1 1 0 0 1

Podział AE

Ta ogólnie przedstawiona zasada działania jest wykorzystywana w różnych wariantach algorytmów ewolucyjnych; wyróżnia się 4 typy EA:

- algorytmy genetyczne (ang. Genetic Algorithms - GA),
- strategie ewolucyjne (ang. Evolution Strategies - ES),
- programowanie ewolucyjne (ang. Evolutionary Programming - EP),
- programowanie genetyczne (ang. Genetic Programming - GP).

Cechy algorytmów ewolucyjnych

Algorytmy powyższe, oprócz terminologii oraz operatorów przeniesionych z biologii, różnią się istotnie od innych technik przeszukiwania:

- wykorzystują operatory pseudoewolucyjne, które działają na sprecyzowanych reprezentacjach rozwiązań (osobnikach),
- przetwarzają całą populację rozwiązań, badając przy tym przestrzeń przeszukiwania równocześnie z wielu punktów,
- do prawidłowego działania nie potrzebują żadnej szczegółowej wiedzy o charakterze problemu, a jedynie informację o jakości rozwiązań,
- w sposób celowy stosują procesy stochastyczne; oznacza to nie losowe, lecz inteligentne badanie przestrzeni przeszukiwania: wysiłek obliczeniowy koncentruje się na obiecujących regionach przestrzeni przeszukiwań.

Zalety algorytmów ewolucyjnych

- efektywna technika, o szerokich możliwościach zastosowania,
- wiarygodność,
- dobrze przystosowane do przeszukiwania wielowymiarowej, złożonej przestrzeni rozwiązań,
- względnie łatwe do opracowania i implementacji,
- nie ma ograniczeń co do postaci funkcji celu,
- wstępna wiedza o problemie nie jest potrzebna,
- możliwość optymalizacji wielokryterialnej,
- duży wybór postaci algorytmu,
- łatwa współpraca z innymi technikami (heurystyki inicjalizacyjne, przeszukiwanie lokalne),
- naturalna paralelność algorytmu.

Wady algorytmów ewolucyjnych

- heurystyczny charakter techniki (nie daje pewności osiągnięcia optimum w określonym czasie),
- czasochłonność obliczeń (łagodzona przez gwałtowny postęp techniczny),
- często nieefektywna w końcowej fazie przeszukiwań,
- określenie właściwych wartości parametrów nie jest łatwe.

Algorytm genetyczny

Model procesu ewolucyjnego do rozwiązywania zagadnień, których rozwiązanie jest sformalizowane w postaci ciągu binarnego, przyjęto nazywać algorytmem genetycznym. Ogólnie biorąc, algorytm genetyczny jest iteracyjną procedurą poprawiania rozwiązań zawartych w zbiorze najlepszych rozwiązań danego zagadnienia, realizowaną przez zbiór pseudogenetycznych operatorów.

- Chromosom jest ciągiem genów (elementów) o ustalonej długości; standardowo jest to ciąg binarny.
- Populację stanowi zbiór chromosomów o zadanej liczebności.

Algorytm genetyczny

Podstawowymi elementami algorytmu genetycznego są:

- genetyczna reprezentacja rozwiązania problemu,
- sposób generowania populacji początkowej,
- postać funkcji dopasowania (ang. fitness function), oceniającej dopasowanie rozwiązań do otoczenia,
- sposób doboru rodziców,
- stosowane operatory genetyczne,
- sposób selekcji następnego pokolenia,
- wartości parametrów:
  -rozmiar populacji *P*,
  -prawdopodobieństwo zastosowania operatorów krzyżowania *pc*,
  -prawdopodobieństwo zastosowania operatorów mutacji *pm*,
  -warunek zatrzymania.

Algorytm genetyczny

```
BEGIN
t := 0;
inicjalizacja populacji początkowej P(t);
REPEAT
    badanie dopasowania P(t);
    t := t+1;
    P(t) = { };
    WHILE  (nie zakończona selekcja P(t)) DO
    BEGIN
            wybór dwóch osobników zgodnie z ich wartością funkcji
            dopasowania;
            rekombinacja osobników w celu utworzenia dwóch potomków;
            mutacja potomków;
            dodanie potomków do populacji P(t);
    END;
UNTIL (spełniony warunek zatrzymania)
END.
```

Algorytm genetyczny - podstawy

Reprezentacja

- Dostosowana do problemu.
- Powinna kodować jak najwięcej informacji specyficznych dla problemu.
- Może wymagać specjalizowanych operatorów.

Funkcja dopasowania

- Dopasowanie osobnika może być obliczone na podstawie genotypu.
- Najczęściej jest to wprost funkcja celu.
- Często funkcja zawiera ograniczenia nałożone na zmienne np. w postaci funkcji kary.

Algorytm genetyczny - podstawy

Reprezentacja

- Potencjalne rozwiązanie problemu może być reprezentowane przez ciąg parametrów (np. wymiary przęseł w przypadku projektu mostu). Parametry te (geny) są łączone tworząc ciąg wartości (chromosom). Przyjmuje się, że alfabet binarny jest najlepszy do tworzenia takiego ciągu. Jeśli np. problemem jest maksymalizacja wartości funkcji dwóch zmiennych F(x,y), reprezentacją każdej zmiennej może być 10-bitowa liczba binarna. Tak więc chromosom będzie się składał z 2 genów i liczył 20 cyfr binarnych.

Algorytm genetyczny - przykład

- Maksymalizacja f(x)=2*x dla 0<=x<=31
- Założenia:
  - rozmiar populacji *P* = 4,
  - krzyżówka jednopunktowa,
  - dobór rodziców zgodnie z funkcją dopasowania (ruletka),
  - prawd. zastosowania operatora krzyżowania *pc* = 0,8,
  - prawd. zastosowania operatora mutacji typu flip *pm* =0,05,
  - selekcja następnego pokolenia – 50% najlepszych dzieci ,
  - warunek zatrzymania.

Algorytm genetyczny - przykład

- Maksymalizacja f(x)=2x dla 0<=x<=31
- Krok 0 – wylosowanie populacji początkowej.
- Pętla 1 – krok 1: obliczenie funkcji dopasowania osobników.

osobnik	4	3	2	1	0	x_i	$FD(x_i)$
1	1	1	0	0	0	24	48
2	0	0	1	0	1	5	10
3	1	0	1	1	0	22	44
4	0	1	1	0	0	12	24
suma						63	126
średnia						15,75	31,50

- Pętla 1 – krok 2: wybór rodziców zgodnie z funkcją dopasowania.

0	x_i	$FD(x_i)$	Prawd.	Dystr.
0	24	48	0,381	0,381
1	5	10	0,079	0,460
0	22	44	0,349	0,810
0	12	24	0,190	1,000
	63	126	1,000	

L.losowa	1. rodzic	L.losowa	2. rodzic
0,258	1	0,462	3
0,593	3	0,030	1
0,984	4	0,340	1
0,704	3	0,963	4

Algorytm genetyczny - przykład

- Maksymalizacja f(x)=2x dla 0<=x<=31
- Pętla 1 – krok 3: krzyżówka.

L.losowa	Rodzice											
0,642	1	1	1	0	0	0	3	1	0	1	1	0
0,117	3	1	0	1	1	0	1	1	1	0	0	0
0,987	4	0	1	1	0	0	1	1	1	0	0	0
0,694	3	1	0	1	1	0	4	0	1	1	0	0

Dzieci											
1	1	1	1	1	0	2	1	0	0	0	0
3	1	0	1	0	0	4	1	1	0	1	0
5	0	1	1	0	0	6	1	1	0	0	0
7	1	1	1	0	0	8	0	0	1	1	0

Algorytm genetyczny - przykład

- Maksymalizacja f(x)=2x dla 0<=x<=31
- Pętla 1 – krok 4: mutacja.

osobnik	4	3	2	1	0	x_i	FD(x_i)	L.losowa	L.losowa	L.losowa	L.losowa	L.losowa
1	1	1	1	1	0	30	60	0,848	0,409	0,962	0,407	0,118
2	1	0	0	0	0	16	32	0,158	0,543	0,472	0,141	0,074
3	1	0	1	0	0	20	40	0,596	0,328	0,284	0,368	0,136
4	1	1	0	1	0	26	52	0,824	0,260	0,116	0,333	0,821
5	0	1	1	0	0	12	24	0,843	0,600	0,344	0,695	0,617
6	1	1	0	0	0	24	48	0,883	0,724	0,922	0,320	0,713
7	1	1	1	0	0	28	56	0,715	0,281	0,790	0,924	0,547
8	0	0	1	1	0	6	12	0,015	0,646	0,399	0,318	0,918
suma						162	324					
średnia						20,25	40,50					

osobnik	4	3	2	1	0	x_i	FD(x_i)
1	1	1	1	1	0	30	60
2	1	0	0	0	0	16	32
3	1	0	1	0	0	20	40
4	1	1	0	1	0	26	52
5	0	1	1	0	0	12	24
6	1	1	0	0	0	24	48
7	1	1	1	0	0	28	56
8	1	0	1	1	0	22	44
suma						178	356
średnia						22,25	44,50

Algorytm genetyczny - przykład

- Maksymalizacja f(x)=2x dla 0<=x<=31
- Pętla 1 – krok 5: selekcja następnego pokolenia.

osobnik	4	3	2	1	0	x_i	$FD(x_i)$
1	1	1	1	1	0	30	60
7	1	1	1	0	0	28	56
4	1	1	0	1	0	26	52
6	1	1	0	0	0	24	48
8	1	0	1	1	0	2[illegible]	[illegible]
3	1	0	1	0	0	2[illegible]	[illegible]
2	1	0	0	0	0	1[illegible]	[illegible]
5	0	1	1	0	0	1[illegible]	[illegible]
suma						17[illegible]	[illegible]
średnia						22,[illegible]	[illegible]

osobnik	4	3	2	1	0	x_i	$FD(x_i)$
1	1	1	0	0	0	24	48
2	0	0	1	0	1	5	10
3	1	0	1	1	0	22	44
4	0	1	1	0	0	12	24
suma						63	126
średnia						15,75	31,50

osobnik	4	3	2	1	0	x_i	$FD(x_i)$
1	1	1	1	1	0	30	60
7	1	1	1	0	0	28	56
4	1	1	0	1	0	26	52
6	1	1	0	0	0	24	48
suma						108	216
średnia						27,00	54,00

EA - zastosowania

- Ekstrema funkcji numerycznych

  Tradycyjne zastosowanie GA, szczególnie dla trudnych, nieciągłych funkcji, o wielu ekstremach.

- Projektowanie

  Zagadnienia te można traktować jako połączenie optymalizacji kombinatorycznej i funkcyjnej. Znane zastosowania dotyczą:

  - projektowanie mostów,
  - optymalizacja dyszy węża ppoż.,
  - projektowanie struktury sieci neuronowej,
  - projektowanie sieci elektrycznej,
  - projektowanie sieci zaopatrzenia w wodę.

EA - zastosowania

- Optymalizacja kombinatoryczna

  Zagadnienia te obejmują problemy rozmieszczenia dyskretnych obiektów w czasie i/lub przestrzeni.

  Najszerzej znanym jest problem komiwojażera (TSP).

  Problem załadunku (BBP) ma duże znaczenie praktyczne.

  Szeregowanie zadań produkcyjnych.

  Problem transportowy.

  Kontrola lotów.

  Planowanie ruchu pacjentów w szpitalu.

  Sterowanie pracą centrali telefonicznej.

  Podział godzin w szkole.

  Zagadnienia te wymagają stosowania niestandardowych reprezentacji i operatorów.

Praktyczne aspekty EA

- Projektując zastosowanie GA nie wystarczy posłużyć się standardowym schematem. Badania empiryczne dowiodły, że krytyczne są następujące elementy:
  - reprezentacja,
  - postać funkcji dopasowania,
  - zróżnicowanie dopasowania,
  - technika wyboru rodziców,
  - technika wyboru pokolenia potomków.

Reprezentacja

Inne reprezentacje

- Dla prostych problemów kolejnościowych naturalną reprezentacją jest lista uporządkowana:
  1 3 2 4 7 5 6 8
- Podstawowy problem - nie działają klasyczne operatory.
- Czy ciąg liczb rzeczywistych może być reprezentacją dla takich problemów?

Sekwencja wejściowa

1	2	3	4	5	6	7	8
0,485223	0,681597	0,489335	0,17524	0,441197	0,699367	0,114374	0,370359

Sekwencja po sortowaniu

7	4	8	5	1	3	2	6
0,114374	0,17524	0,370359	0,441197	0,485223	0,489335	0,681597	0,699367

Reprezentacja

Inne reprezentacje

Prosta krzyżówka
Rodzice

7	4	8	5	1	3	2	6

6	2	3	4	8	7	1	5

Dzieci

7	4	8	5	8	7	1	5

6	2	3	4	1	3	2	6

Prosta krzyżówka
Rodzice

0,485223	0,681597	0,489335	0,17524	0,441197	0,699367	0,114374	0,370359

0,754165	0,176504	0,329154	0,329504	0,762581	0,006147	0,611928	0,402365

Dzieci

0,485223	0,681597	0,489335	0,17524	0,762581	0,006147	0,611928	0,402365

0,754165	0,176504	0,329154	0,329504	0,441197	0,699367	0,114374	0,370359

6	4	8	1	3	7	2	5
0,006147	0,17524	0,402365	0,485223	0,489335	0,611928	0,681597	0,762581

7	2	3	4	8	5	6	1
0,114374	0,176504	0,329154	0,329504	0,370359	0,441197	0,699367	0,754165

Reprezentacja

Inne reprezentacje

- Reprezentacja macierzowa:

np. dla problemu komiwojażera:

0	0	0	1	0	
1	0	1	1	1	
1	0	0	1	1	koduje trasę 2 3 5 1 4
0	0	0	0	0	
1	0	0	1	0	

  - jest w niej dokładnie $n(n-1)/2$ jedynek,
  - na przekątnej są zera,
  - jeśli $m_{ij}=1$ i $m_{jk}=1$, to $m_{ik}=1$.
- Podobna reprezentacja - dla układania podziału godzin.

Zróżnicowanie dopasowania

- Przedwczesna zbieżność

  Dominacja dobrych (ale nie optymalnych) kilku chromosomów powoduje utknięcie w lokalnym ekstremum. Gdyby populacja była nieskończona nie byłoby problemu, w rzeczywistych zastosowaniach należy zmodyfikować sposób wyboru rodziców do reprodukcji. Wyjściem jest kontrola liczby szans na reprodukcję poszczególnych osobników, tak aby nie była ona ani zbyt duża, ani zbyt mała.

- Powolne finiszowanie

  Odwrotne zagadnienie; po wielu generacjach populacja jest w większości zbieżna, ale globalne maximum nie jest precyzyjnie zlokalizowane. Średnie dopasowanie jest wysokie, a różnica między najlepszym i najgorszym osobnikiem - niewielka. W konsekwencji brak impulsu do skierowania EA ku maksimum. Wyjściem jest rozszerzenie efektywnego zróżnicowania wartości funkcji dopasowania przez jej skalowanie.

Techniki wyboru rodziców

- Selekcja na zasadzie turnieju

  W najprostszej wersji para osobników jest losowana z populacji; osobnik o lepszym dopasowaniu z tej dwójki jest wybierany do krzyżowania. Turniej trwa aż do stworzenia wszystkich par. Możliwa jest również wersja, w której „ściera się" ze sobą *n* losowo wybranych osobników.

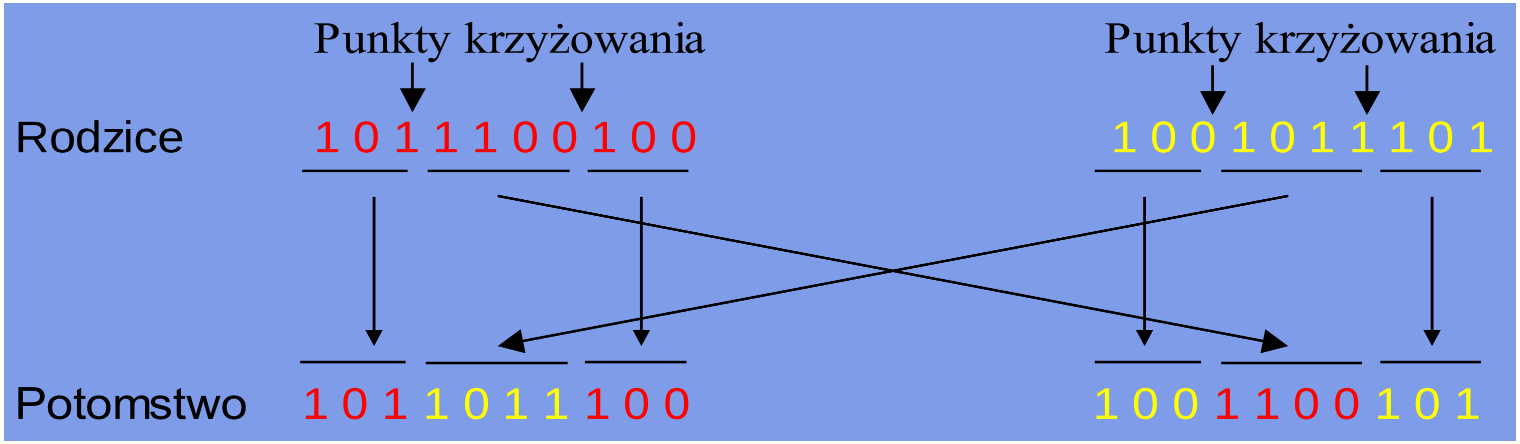
  W turnieju probabilistycznym, z pary (lub *n*) wylosowanych osobników wygrywa lepszy z prawdopodobieństwem $0{,}5<p<1{,}0$. Selekcja jest dużo łagodniejsza.

Technika wyboru potomstwa

- Standardowo, cała populacja rodziców jest wymieniana przez potomstwo. Ostatnie badania wykazują, że lepiej wymieniać tylko część populacji. Ma to uzasadnienie w naturze.
  Technika wyboru osobników, które „mają pecha” i zostaną usunięte z populacji może być różna: najczęściej stosuje się wybór losowy lub deterministyczny najgorszych osobników.

Techniki krzyżowania

- Wielopunktowe krzyżówki służą intensywniejszemu przeszukiwaniu przestrzeni rozwiązań. Szczególnie dobrze działa krzyżówka dwupunktowa.



Techniki krzyżowania

- Krzyżówka ujednolicająca (ang. uniform crossover) tworzy potomków przez skopiowanie odpowiednich genów z jednego lub drugiego rodzica zgodnie z tworzoną losowo dla każdej pary maską.

	Maska krzyżowania	0 0 1 1 0 0 1 1 0 1
Rodzice	1 0 1 1 1 0 0 1 0 0	1 0 0 1 0 1 1 1 0 1
Potomstwo	1 0 1 1 0 1 0 1 0 0	1 0 0 1 1 0 1 1 0 1

Dla niektórych problemów opisane wyżej krzyżówki nie mogą być stosowane. Należą do nich problemy kolejnościowe, gdzie wartości genów z reguły nie są bitami, a wartość funkcji dopasowania zależy od kolejności, w której występują.

Techniki krzyżowania - OX

- Krzyżówka OX (Order Crossover). Zaproponował ją Davis; buduje potomstwo przez wybór podciągu z uszeregowania i zachowanie względnej kolejności zadań z innego rodzica.
- Np. dwoje rodziców:

  R1 = (1 2 3 | 4 5 6 7 | 8 9)

  R2 = (4 5 2 | 1 8 7 6 | 9 3)

  Na początku segmenty pomiędzy punktami cięcia są kopiowane do potomków:

  P1 = (x x x | 4 5 6 7 | x x)

  P2 = (x x x | 1 8 7 6 | x x)

Techniki krzyżowania - OX

- Następnie poczynając od drugiego punktu cięcia pierwszego rodzica, zadania z drugiego rodzica są kopiowane w tej samej kolejności, pomijając symbole już istniejące. Po osiągnięciu końca ciągu kontynuujemy od pierwszej pozycji ciągu.

  P1 = (2 1 8 | 4 5 6 7 | 9 3)

  P2 = (3 4 5 | 1 8 7 6 | 9 2)

  R1 = (1 2 3 | 4 5 6 7 | 8 9)

  R2 = (4 5 2 | 1 8 7 6 | 9 3)

Techniki krzyżowania

- Operator losowy z uzupełnianiem działający w ten sposób, że losowane są pozycje zadań przenoszonych bezpośrednio z rodziców do potomków, a reszta - jest uzupełniana z drugiego rodzica (z pominięciem zadań powtarzających się).

  Rozważmy np. parę rodziców:

  R1 = (1,2,3,4,5,6,7,8,9) i

  R2 = (2,4,1,7,8,3,9,5,6)

Techniki krzyżowania

- Niech dla pierwszego rodzica wylosowano niezmienione pozycje nr 2,4,5,8, a dla drugiego - 1,2,6,9. Zadania na tych pozycjach są przepisywane do potomków:
  P1 = (x,2,x,4,5,x,x,8,x) i
  P2 = (2,4,x,x,x,3,x,x,6).
- Reszta jest przenoszona z drugiego rodzica dając w rezultacie parę:
  P1 = (1,2,7,4,5,3,9,8,6) i
  P2 = (2,4,1,5,7,3,8,9,6).

Algorytm genetyczny – przykład TSP

- Znaleźć najkrótszą trasę dla n=7 miast; odległości między miastami wynoszą:

	1	2	3	4	5	6	7
1		4	2	5	2	2	1
2	4		3	7	4	4	5
3	2	3		1	8	5	3
4	5	7	1		6	1	4
5	2	4	8	6		5	1
6	2	4	5	1	5		3
7	1	5	3	4	1	3	

- Założenia:
  - rozmiar populacji P = 4;
  - krzyżówka losowa z uzupełnianiem;
  - dobór rodziców zgodnie z funkcją dopasowania (ruletka);
  - prawd. zastosowania operatora krzyżowania pc = 0,8;
  - prawd. zastosowania operatora mutacji typu swap pm = 0,05;
  - selekcja następnego pokolenia – 50% najlepszych dzieci;
  - warunek zatrzymania.

Przykład GA – TSP dla *n* = 7 miast

- Krok 0 – wylosowanie populacji początkowej.
- Pętla 1 – krok 1: obliczenie funkcji dopasowania osobników.

osobnik	m1	m2	m3	m4	m5	m6	m7	$DT(x_i)$	$FD(x_i)$
1	4	2	5	3	7	1	6	26	15
2	7	3	6	1	2	5	4	28	13
3	1	2	3	4	5	6	7	23	18
4	6	7	1	5	3	4	2	26	15
suma								103,0	61,0
średnia								25,8	15,3

- Pętla 1 – krok 2: wybór rodziców zgodnie z funkcją dopasowania.

$DT(x_i)$	$FD(x_i)$	Prawd.	Dystr.
26	15	0,246	0,246
28	13	0,213	0,459
23	18	0,295	0,754
26	15	0,246	1,000
103,0	61,0	1,000	
25,8	15,3		

L.los	1. rodzic	L.los	2. rodzic
0,176	1	0,738	3
0,278	2	0,552	3
0,637	3	0,883	4
0,924	4	0,220	1

Przykład GA – TSP dla *n* = 7 miast

- Pętla 1 – krok 3: krzyżówka.
- Pętla 1 – krok 4: ewaluacja populacji.

L.los	1. rodzic	L.los	2. rodzic
0,176	1	0,738	3
0,278	2	0,552	3
0,637	3	0,883	4
0,924	4	0,220	1

Rodzice

0,642	1	4	2	5	3	7	1	6	3	1	2	3	4	5	6	7
0,117	2	7	3	6	1	2	5	4	3	1	2	3	4	5	6	7
0,987	3	1	2	3	4	5	6	7	4	6	7	1	5	3	4	2
0,694	4	6	7	1	5	3	4	2	1	4	2	5	3	7	1	6

Dzieci

1	4	1	5	2	7	3	6	2	1	4	3	2	5	6	7
3	2	3	4	1	6	5	7	4	7	2	3	4	1	6	5
5	1	2	3	4	5	6	7	6	6	7	1	5	3	4	2
7	6	4	1	5	3	7	2	8	4	1	5	3	7	2	6

Przykład GA – TSP dla *n* = 7 miast

- Pętla 1 – krok 5: mutacja.

osobnik	m1	m2	m3	m4	m5	m6	m7	DT(x_i)	FD(x_i)	L.losowa
1	4	1	5	2	7	3	6	25	16	0,848
2	1	4	3	2	5	6	7	22	19	0,158
3	2	3	4	1	6	5	7	22	19	0,596
4	7	2	3	4	1	6	5	22	19	0,824
5	1	2	3	4	5	6	7	23	18	0,843
6	6	7	1	5	3	4	2	26	15	0,883
7	6	4	1	5	3	7	2	28	13	0,715
8	4	1	5	3	7	2	6	28	13	0,015
suma								196,0	132,0	
średnia								24,5	16,5	

osobnik	m1	m2	m3	m4	m5	m6	m7	DT(x_i)	FD(x_i)
1	4	1	5	2	7	3	6	25	16
2	1	4	3	2	5	6	7	22	19
3	2	3	4	1	6	5	7	22	19
4	7	2	3	4	1	6	5	22	19
5	1	2	3	4	5	6	7	23	18
6	6	7	1	5	3	4	2	26	15
7	6	4	1	5	3	7	2	28	13
8	4	2	5	3	7	1	6	26	15
suma								194,0	134
średnia								24,3	16,75

Przykład GA – TSP dla *n* = 7 miast

- Pętla 1 – krok 6: selekcja następnego pokolenia.

osobnik	m1	m2	m3	m4	m5	m6	m7	$DT(x_i)$	$FD(x_i)$
2	1	4	3	2	5	6	7	22	19
3	2	3	4	1	6	5	7	22	19
4	7	2	3	4	1	6	5	22	19
5	1	2	3	4	5	6	7	23	18
1	4	1	5	2	7	3	6	25	16
6	6	7	1	5	3	4	2		
8	4	2	5	3	7	1	6		
7	6	4	1	5	3	7	2		

osobnik	m1	m2	m3	m4	m5	m6	m7	$DT(x_i)$	$FD(x_i)$
1	4	2	5	3	7	1	6	26	15
2	7	3	6	1	2	5	4	28	13
3	1	2	3	4	5	6	7	23	18
4	6	7	1	5	3	4	2	26	15
suma								103,0	61,0
średnia								25,8	15,3

osobnik	m1	m2	m3	m4	m5	m6	m7	$DT(x_i)$	$FD(x_i)$
2	1	4	3	2	5	6	7	22	19
3	2	3	4	1	6	5	7	22	19
4	7	2	3	4	1	6	5	22	19
5	1	2	3	4	5	6	7	23	18
suma								89,0	75
średnia								22,3	18,75

Opt.	3	2	5	7	1	6	4	**DT=13**