



Katedra Wytrzymałości Materiałów  
i Metod Komputerowych Mechaniki  
[www.kwmimkm.polsl.pl](http://www.kwmimkm.polsl.pl)

Wydział Mechaniczny Technologiczny  
Politechnika Śląska

---

## METODY HEURYSTYCZNE

---

### **LABORATORIUM 4:** Algorytmy ewolucyjne cz. 2 – wpływ operatorów krzyżowania i mutacji na skuteczność poszukiwań AE

opracował: dr inż. Witold Beluch  
[witold.beluch@polsl.pl](mailto:witold.beluch@polsl.pl)

## Cel ćwiczenia

W ramach ćwiczenia przeprowadzisz testy mające na celu wykazanie skuteczności działania poszczególnych operatorów w algorytmie ewolucyjnym dla wybranej funkcji celu. Sprawdzisz też, jak parametry (a dokładniej w tym przypadku prawdopodobieństwa) poszczególnych operatorów wpływają na skuteczność poszukiwań algorytmu ewolucyjnego.

## Trochę teorii

W ramach niniejszego ćwiczenia sprawdzisz działanie różnych operatorów krzyżowania i mutacji w algorytmie ewolucyjnym. Program *Evolutionary Algorithms* korzysta z kodowania rzeczywistoliczbowego<sup>1</sup>, w związku z tym stosowane w programie operatory ewolucyjne muszą być dopasowane do tego rodzaju kodowania zmiennych projektowych. Poniżej skrótkowo opisano tylko te operatory krzyżowania i mutacji, które są zaimplementowane w programie *Evolutionary Algorithms*. Więcej informacji na temat tych i innych operatorów można znaleźć np. w literaturze zebranej na końcu niniejszej instrukcji.

### ✚ Operatory krzyżowania

- **Krzyżowanie proste** (inaczej: jednopunktowe) jest operatorem krzyżowania wymieniającego, czyli takiego, które nie modyfikuje genów osobników rodzicielskich, a jedynie powoduje ich przetasowanie. Może być, podobnie jak i inne operatory krzyżowania wymieniającego, stosowane zarówno przy kodowaniu binarnym, jak i rzeczywistoliczbowym. Polega na losowym wybraniu dla dwu osobników rodzicielskich ( $X^1$  i  $X^2$  na Rys. 1) punktu ich przecięcia  $c$  i zamianie fragmentów kodu. W wyniku powstają osobniki potomne ( $Y$  oraz  $Z$ ):

$$Y = [X^1_1, \dots, X^1_c, X^2_{c+1}, \dots, X^2_n]$$

$$Z = [X^2_1, \dots, X^2_c, X^1_{c+1}, \dots, X^1_n]$$

$X^1$	$X^2$		$Y$	$Z$
3.24	2.22		3.24	2.22
-0.22	3.14		-0.22	3.14
1.32	7.72		1.32	7.72
3.22	1.22	$c$	1.22	3.22
1.20	2.40		2.40	1.20
7.23	4.28		4.28	7.23
-2.21	-2.42		-2.42	-2.21

Rys. 1 Krzyżowanie proste (jednopunktowe)

- **Krzyżowanie arytmetyczne** jest operatorem krzyżowania uśredniającego, a więc takim, który modyfikuje geny osobników rodzicielskich. Jest operatorem typowym dla kodowania rzeczywistoliczbowego (zastanów się, dlaczego nie można go zastosować przy kodowaniu binarnym?).

<sup>1</sup> Jak pamiętasz z wykładu, w klasycznych algorytmach genetycznych stosowano kodowanie binarne. Takie kodowanie w typowych zadaniach inżynierskich, gdzie zmienne są liczbami rzeczywistymi, jest nieco kłopotliwe (wymagane jest przekształcanie ciągów binarnych na liczby rzeczywiste, ponadto zmiana dziedziny lub żądanej dokładności obliczeń wpływa na długość osobnika). W związku z tym należy przyjąć zasadę, iż kodowanie powinno w możliwie naturalny sposób reprezentować zadanie – i tak np. kodowanie binarne jest naturalne w przypadku tzw. zadania plecakowego, gdzie 1 na danej pozycji odpowiada zabraniu danej rzeczy do plecaka, zaś 0 odpowiada jej pozostawieniu.

W pierwszym etapie losowana jest liczba  $k$  z zakresu  $\langle 0,1 \rangle$ . W drugim etapie wartości genów chromosomów rodzicielskich są uśredniane (Rys. 2):

$$Y = X^1 + k(X^2 - X^1)$$

$$Z = X^2 + X^1 - Y$$



Rys. 2 Krzyżowanie arytmetyczne

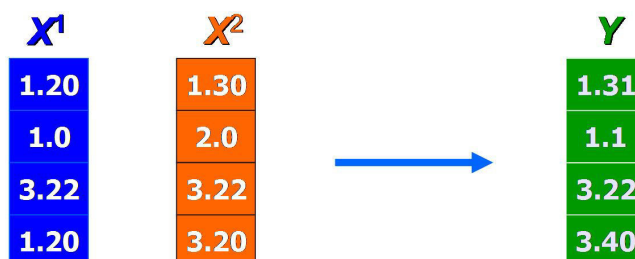
Wartości genów osobników potomnych leżą zawsze pomiędzy wartościami genów osobników rodzicielskich, więc jeśli rodzice spełniają ograniczenia kostkowe na zmienne<sup>1</sup>, to osobniki potomne również je spełniają.

- **Krzyżowanie heurystyczne** jest operatorem podobnym w działaniu do krzyżowania arytmetycznego i również jest operatorem typowym dla kodowania rzeczywistoliczbowego. W pierwszym etapie losowana jest liczba  $k$  przyjmująca wartości z zakresu  $\langle 0,1 \rangle$ . W drugim etapie tworzony jest z osobników rodzicielskich maksymalnie 1 potomek taki, że:

$$Y = X^2 + k(X^2 - X^1)$$

przy założeniu, że  $f(X^2) \geq f(X^1)$ , czyli że osobnik 2 jest nie gorszy od osobnika 1.

Ten rodzaj krzyżowania może utworzyć osobnika, który nie spełnia ograniczeń kostkowych na zmienne. W takim przypadku procedura ponownie generowania jest liczba losowa i próba utworzenia osobnika potomnego jest powtarzana. Jeśli przez założoną liczbę prób nie uda się utworzyć osobnika dopuszczalnego wówczas potomek nie jest tworzony w ogóle. Przykład krzyżowania heurystycznego przedstawia Rys. 3 (zwróć uwagę, że wartości genów potomka mogą być większe, niż wartości genów obojga rodziców).



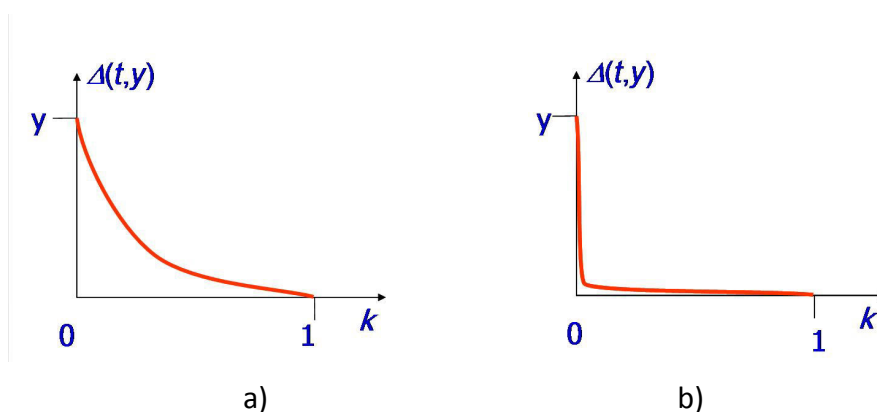
Rys. 3 Krzyżowanie heurystyczne

<sup>1</sup> Ograniczenia kostkowe (proste) to takie, że dla  $i$ -tej zmiennej projektowej  $x_i$ :  $a_i \leq x_i \leq b_i$ , gdzie liczby  $a_i$  oraz  $b_i$  ograniczają z dołu i z góry wartości, które może przyjmować zmienna  $x_i$ .

## 🚧 Operatory mutacji

Wszystkie zastosowane w programie *Evolutionary Algorithms* operatory mutacji są typowe dla kodowania rzeczywistoliczbowego. Operatory mutacji działają zazwyczaj w ten sposób, że dla każdego genu każdego osobnika w populacji podejmuje z prawdopodobieństwem równym prawdopodobieństwu mutacji decyzję, czy gen ten ulegnie mutacji. W przedstawionych poniżej operatorach różny jest sposób, w jaki dochodzi do modyfikacji wybranego genu.

- **Mutacja równomierna** jest operatorem mutacji, który modyfikuje wybrany gen tak, iż przyjmuje on z losową wartość z zakresu zmienności danej zmiennej z rozkładem równomiernym. Operator ten jest szczególnie przydatny w początkowej fazie działania algorytmu, gdzie wskazane jest szerokie przeszukiwanie przestrzeni rozwiązań.
- **Mutacja nierównomierna** jest operatorem należącym do tzw. grupy mutacji ze strojeniem. Wybrany gen jest modyfikowany o wartość pewnej funkcji  $\Delta(t,y)$ . Funkcja ta w początkowych pokoleniach pozwala na istotne modyfikacje genu z dużym prawdopodobieństwem, które to prawdopodobieństwo zmniejsza się w kolejnych pokoleniach (Rys. 4). Takie podejście ogranicza wady poprzedniego operatora mutacji w końcowej fazie działania AE.



Rys. 4 Przebieg funkcji  $\Delta(t,y)$  a) początkowa faza obliczeń, b) końcowa faza obliczeń

- **Mutacja brzegowa** jest operatorem mutacji, który powoduje przyjęcie przez wybrany gen (z jednakowym prawdopodobieństwem) wartości z lewego bądź z prawego krańca przedziału dla danej zmiennej. Operator ten jest szczególnie użyteczny, gdy ekstremum globalne leży na, bądź w pobliżu brzegu obszaru dopuszczalnego.

## Program *Evolutionary Algorithms*

Program *Evolutionary Algorithms* został opisany (w zakresie algorytmów ewolucyjnych) w instrukcji do ćwiczenia laboratoryjnego numer 3. W razie potrzeby sięgnij do tej instrukcji. Pamiętaj o ważnych elementach związanych z funkcjonowaniem programu – są one przypomniane poniżej:

! Zawsze rozpoczynając pracę z programem należy sprawdzić, czy:

- liczba zmiennych projektowych wynosi 2 (*Data*->*Number of variables*);
- nie ma przypadkowo wprowadzonych ograniczeń liniowych oraz nieliniowych (*Data* -> *Linear/Nonlinear constraints*).

## Do wykonania

Przeprowadź poszukiwania maksimum globalnego dla następującej wielomodalnej funkcji celu (przystosowania):

$$f(x, y) = ((36 - (x - 6) * (x - 6)) * \cos(2 * (x - 6))) + \\ + ((36 - (y - 6) * (y - 6)) * \cos(2 * (y - 6))) + 78$$

### A. Przyjmij:

- ograniczenia na zmienne:  $0 < x < 12$ ,  $0 < y < 12$ .
- wielkość populacji: 20 osobników;
- liczba pokoleń: 50;
- funkcja kary: kara śmierci (*death penalty*);
- selekcja: turniejowa (*tournament selection*),  $p_t=0.1$



Wyniki w tabelach zapisuj z dokładnością do 5-go miejsca po przecinku.



Pamiętaj o kopiowaniu pliku *bes\_popu.dat* (jeden plik dla każdego wiersza obu tabel).

### B. Zbadaj wpływ różnych operatorów krzyżowania na skuteczność poszukiwań EA.

- Jako jedyny aktywny operator mutacji przyjmij mutację równomierną (*uniform mutation*),  $p_{um}=0.03$ .
- Przeprowadź trzykrotnie obliczenia dla następujących aktywnych operatorów krzyżowania:
  - ♦ tylko krzyżowanie proste (*simple crossover*),  $p_{sc}=0.3$ ;
  - ♦ tylko krzyżowanie arytmetyczne (*arithmetic crossover*),  $p_{ac}=0.3$ ;
  - ♦ tylko krzyżowanie heurystyczne (*heuristic crossover*),  $p_{hc}=0.3$ ;
  - ♦ wszystkie powyższe operatory jednocześnie ( $p_{sc}=p_{ac}=p_{hc}=0.1$ ).

Wypełnij tabelę 1.

### C. Zbadaj wpływ różnych operatorów mutacji na skuteczność poszukiwań EA.

- Jako jedyny aktywny operator krzyżowania przyjmij krzyżowanie proste,  $p_{sc}=0.3$ .
- Przeprowadź trzykrotnie obliczenia dla następujących aktywnych operatorów mutacji:
  - ♦ tylko mutacja równomierna  $p_{um}=0.03$ ;
  - ♦ tylko mutacja nierównomierna, (*non-uniform mutation*),  $p_{nm}=0.03$ ;
  - ♦ tylko mutacja brzegowa (*boundary mutation*),  $p_{bm}=0.03$ ;
  - ♦ wszystkie powyższe operatory jednocześnie ( $p_{um}=p_{nm}=p_{bm}=0.01$ ).

Wypełnij tabelę 2.

## Sprawozdanie

- Sprawozdanie ma być dostarczone wyłącznie w formie elektronicznej.
- Nazwa pliku wg wzorca: [MH\\_lab4\\_Jan\\_Kowalski.doc/pdf](#).
- Strona pierwsza to strona tytułowa.
- W sprawozdaniu należy zamieścić (poza stroną tytułową):
  1. Cel ćwiczenia.
  2. Optymalizowane funkcje (w tym ich postaci graficzne) oraz ograniczenia na zmienne.
  3. Parametry AE.
  4. Skan/fotografię protokołu.
  5. Wykresy wartości funkcji celu najlepszych osobników w kolejnych pokoleniach (po 12 na jednym wykresie) oraz wartości uśrednione z 3 przebiegów (po 4 na jednym wykresie), wygenerowane na podstawie posiadanych plików.
  6. Wnioski z ćwiczenia z podziałem na wnioski dotyczące krzyżowania oraz mutacji.

## Literatura i źródła

- [1] J. Arabas: Wykłady z algorytmów ewolucyjnych. WNT, Warszawa, 2003.
- [2] Z. Michalewicz: Algorytmy genetyczne + struktury danych = programy ewolucyjne. WNT, Warszawa, 1996.
- [3] L. Rutkowski: Metody i techniki sztucznej inteligencji. PWN, Warszawa, 2006.
- [4] T. Gwiazda: Algorytmy genetyczne kompendium. Tom I: Operator krzyżowania dla problemów numerycznych, tom II: Operator mutacji dla problemów numerycznych. PWN, Warszawa, 2007.

**Protokół do laboratorium 4: Wpływ operatorów krzyżowania i mutacji na skuteczność poszukiwań AE**

Imię i nazwisko	Rok ak.	Gr.	Sem.	Komp.	Data	Podpis prowadzącego
_____	20__/__	ME3	I	_____	_____	

Tabela 1. Wyniki dla różnych operatorów krzyżowania

L.p.	Rodzaj krzyżowania	Wartość f. celu najlepszego osobnika	x[1]	x[2]	Znaleziono w pokoleniu:
1	proste				
2					
3					
4	arytmetyczne				
5					
6					
7	heurystyczne				
8					
9					
10	wszystkie				
11					
12					

Tabela 2. Wyniki dla różnych operatorów mutacji

L.p.	Rodzaj mutacji	Wartość f. celu najlepszego osobnika	x[1]	x[2]	Znaleziono w pokoleniu:
1	równomierna				
2					
3					
4	nierównomierna				
5					
6					
7	brzegowa				
8					
9					
10	wszystkie				
11					
12					

**Notatki** (na drugiej stronie):