



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

Wydział Mechaniczny Technologiczny  
Politechnika Śląska

---

## OBLICZENIA EWOLUCYJNE

---

### **LABORATORIUM 2:** Wpływ wielkości populacji i liczby pokoleń na skuteczność poszukiwań AE

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

## Cel ćwiczenia

Wykonując ćwiczenia laboratoryjne przeprowadzisz badania, w wyniku których określisz wpływ wielkości populacji i liczby pokoleń na skuteczność poszukiwań optimum za pomocą algorytmu ewolucyjnego (AE). W wyniku przeprowadzonych badań określisz, jakie kombinacje powyższych parametrów okazały się najbardziej efektywne dla zadanych funkcji celu (przystosowania) i ograniczeń.

## Trochę teorii

### + Wielkość populacji jako parametr AE

Jak już zapewne wiesz, algorytmy ewolucyjne są metodami optymalizacji, w których w każdym pokoleniu (iteracji, pętli algorytmu) przetwarzany jest cały zbiór osobników (potencjalnych rozwiązań zadania optymalizacji). W związku z tym przed uruchomieniem obliczeń należy określić, jak liczny ma być ten zbiór, czyli zdefiniować liczebność populacji bazowej. Nie ma niestety jednoznacznej odpowiedzi na pytanie, jak liczna powinna być populacja – dobór liczebności populacji wynika np. z rodzaju i stopnia skomplikowania zadania, postaci funkcji celu i ograniczeń, jak również zależy od pewnego doświadczenia osób przeprowadzających obliczenia ewolucyjne. Należy też mieć na uwadze fakt, że koszt (np. czasowy) obliczeń ewolucyjnych jest wprost proporcjonalny do liczby jednocześnie przetwarzanych osobników.

Jak wynika z bogatej literatury dotyczącej tematyki obliczeń ewolucyjnych, opinie na temat optymalnej liczebności populacji wśród badaczy są częstokroć bardzo rozbieżne. Niektórzy twierdzą, że niewielkie liczebnie populacje skutkują bardziej „elastycznym” zachowaniem się algorytmu, co wyraża się poprzez większą łatwość opuszczania ekstremów lokalnych, podczas gdy duże populacje charakteryzują się większą „bezwładnością”. Z kolei inni badacze (szczególnie zajmujący się tzw. programowaniem genetycznym) uznają, że lepsze wyniki uzyskuje się przy dużych populacji rzędu tysięcy osobników. Można znaleźć też podejście, w którym sugeruje się, że można znaleźć na tyle dużą liczebność populacji, że dalsze jej zwiększanie nie przynosi poprawy działania algorytmu – przy czym wielkość ta nie jest stała, lecz zależy od rozwiązywanego problemu.

Interesującym pomysłem na rozwiązanie problemu doboru liczebności populacji jest koncepcja algorytmu genetycznego ze zmienną liczebnością populacji (AGzZLP) [2]. Idea algorytmu bazuje na obserwacji, że optymalna liczebność populacji może być różna w różnych etapach działania AE. W algorytmie tym każdemu osobnikowi przypisywany jest (jednorazowo) jego czas życia, który zastępuje procedurę selekcji. Czas życia osobnika, ustalany w momencie jego powstania, zależy od jego przystosowania. Wiek osobnika jest zwiększany w kolejnych pętlach algorytmu; gdy wiek osobnika przekroczy czas jego życia, osobnik taki jest usuwany z populacji. Oczywiście czas życia należy dobrać tak, by uniknąć niekontrolowanego wzrostu liczebności populacji. Schemat działania AGzZLP w postaci zapisu procedury (funkcji) w pseudokodzie został przedstawiony na rys. 1.

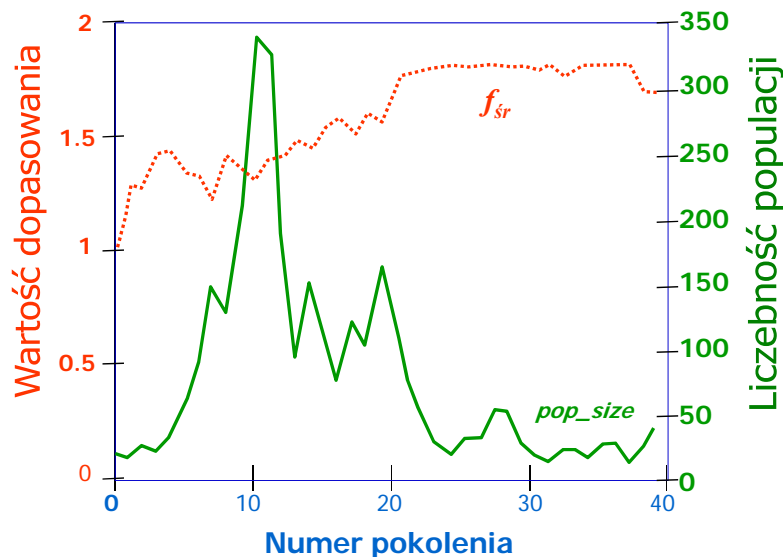
```

procedure AGzZLP
begin
  t:=0
  wybierz populację początkową P(t)
  oceń P(t)
  while (not warunek zakończenia) do
    begin
      t:=t+1
      podnieś wiek każdego osobnika o 1
      zmień P(t) (utwórz P'(t))
      oceń P'(t)
      usuń osobniki o wieku większym od ich czasu życia
    end
  end

```

Rys. 1 procedura AGzZLP w pseudokodzie

Na rys. 2 przedstawiono przykładowe przebiegi wartości średniego dopasowania populacji<sup>1</sup>  $f_{sr}$  oraz liczby osobników ( $pop\_size$ ) w kolejnych pokoleniach AGzZLP.



Rys. 2 Przykładowe przebiegi: wartości dopasowania populacji i liczebności populacji bazowej w kolejnych pokoleniach AGzZLP

W programie *Evolutionary Algorithms* liczebność populacji musi być określona przed rozpoczęciem obliczeń i jest stała podczas danego uruchomienia AE. W ramach wykonywania badań będziesz między innymi zmieniać liczebność populacji bazowej przy ustalonej liczbie pokoleń dla dwu różnych funkcji testowych.

<sup>1</sup> Średnie dopasowanie populacji  $f_{sr}$  jest liczone jako:

$$f_{sr} = \frac{1}{pop\_size} \sum_{i=1}^{pop\_size} f_i$$

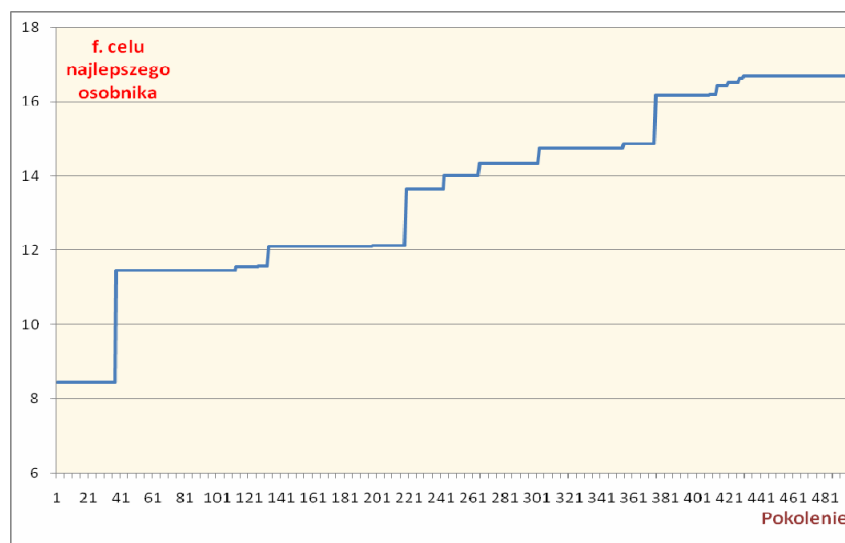
gdzie:  $f_i$  – funkcja przystosowania  $i$ -go osobnika,  $pop\_size$  – liczebność populacji

## ✚ Liczba pokoleń w AE

Jednym z najczęściej przyjmowanych kryteriów zatrzymania AE jest osiągnięcie założonej liczby pokoleń. Jest to jedna z wersji kryterium maksymalnego kosztu<sup>1</sup>. Należy pamiętać, że – inaczej niż w metodach przetwarzających jedno rozwiązanie – całkowity koszt obliczeniowy zależy nie tylko od liczby pętli algorytmu (pokoleń), ale również od liczebności populacji. Wynika to z faktu, że dla każdego nowopowstałego rozwiązania musi być wyliczona wartość jego przystosowania.

„Właściwa” liczba pokoleń zależy zazwyczaj od specyfiki zadania. Przyjęcie zbyt małej liczby pokoleń może spowodować przedwczesne zakończenie poszukiwań i znalezienie np. optimum lokalnego. Z kolei przyjęcie zbyt dużej liczby pokoleń skutkuje niepotrzebnym wydłużeniem czasu obliczeń, podczas gdy algorytm nie jest w stanie znaleźć lepszego rozwiązania (co wcale nie musi oznaczać, że odnalazł optimum globalne). Jak już wiesz pewną informacją o poprawnym oszacowaniu liczby pokoleń jest to, w którym pokoleniu znaleziono najlepsze rozwiązanie w danym przebiegu AE. Jeśli jest to ostatnie lub jedno z ostatnich pokoleń to może to sugerować, że przyjęto zbyt małą liczbę pokoleń. Z kolei znalezienie rozwiązania dużo wcześniej, niż zakładana liczba pokoleń może sugerować zmniejszenie liczby pokoleń. Jeszcze lepiej jest to widoczne na wykresie zależności najlepszego rozwiązania od numeru pokolenia (jedna z tzw. krzywych zbieżności algorytmu).

Przykładowa krzywa zbieżności dla najlepszego rozwiązania przedstawiona jest na rys. 3. Jak sądzisz – czy przyjęta liczba pokoleń (tu: 500) była właściwa? A może zmniejszyłbyś ją lub zwiększył? Uzasadnij swoją odpowiedź.



Rys. 3 Przykładowa krzywa zbieżności – przebieg najlepszego rozwiązania w kolejnych pokoleniach

Samodzielnie stosowane kryterium zatrzymania w postaci maksymalnej liczby pokoleń wydaje się być nie najlepszym rozwiązaniem. Jak wiesz z instrukcji co laboratorium 1 (przypis dolny na stronie 4) w programie *Evolutionary Algorithms* jest również zaimplementowane dodatkowe kryterium (kryterium minimalnej szybkości poprawy). Więcej na temat kryteriów zatrzymania dowiesz się na wykładzie.

W programie *Evolutionary Algorithms* maksymalna liczba pokoleń jest określana przed rozpoczęciem obliczeń. W ramach przeprowadzanych badań będziesz między innymi zmieniał maksymalną liczbę pokoleń przy ustalonej liczebności populacji bazowej dla dwu różnych funkcji testowych.

<sup>1</sup> Według tego kryterium algorytm kończy działanie, jeśli koszt algorytmu przekroczy założoną wartość maksymalną. Często przyjmuje się wariant z maksymalną liczbę pokoleń (ważne - koszt zależy niejawnie od liczby nowych osobników w każdym pokoleniu).

## Program *Evolutionary Algorithms*

Program *Evolutionary Algorithms* został opisany (w zakresie algorytmów ewolucyjnych) w instrukcji do laboratorium numer 1. 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 obliczenia dla następujących funkcji:

Funkcja nr 1:

$$f_1(x, y) = 10 / (1 + \text{sqr}((x-5)*(x-5) + (y-5)*(y-5))) + 4 / (1 + \text{sqr}((x-2)*(x-2) + (y-2)*(y-2)))$$

Funkcja nr 2:

$$f_2(x, y) = ((25 - (x-5)*(x-5)) * \cos(2*(x-5))) + ((25 - (y-5)*(y-5)) * \cos(2*(y-5))) + 50$$

A. Dla obydwu funkcji przyjmij:

- Ograniczenia na zmienne:  $0 < x < 10$ ,  $0 < y < 10$ .
- Używane operatory:
  - ♦ mutacja równomierna (*uniform mutation*),  $p_{um}=0.05$ ;
  - ♦ krzyżowanie proste (*simple crossover*),  $p_{sc}=0.2$ ;
- Inne:
  - ♦ funkcja kary: kara śmierci (*death penalty*);
  - ♦ selekcja: turniejowa (*tournament selection*),  $pt=0.5$ .

B. Wpisz funkcję nr 1.

- Przyjmij liczbę pokoleń równą 20 i wykonaj obliczenia dla wielkości populacji:

**10, 25, 50, 100, 250.**

Wypełnij stosowną część tabeli 1. Wyniki w tabelach zapisuj z dokładnością do 5-go miejsca po przecinku.

! Dla każdego wiersza tabeli oprócz wpisania do niej odpowiednich wartości **skopiuj** do dowolnego folderu (na dysku lub przenośnym nośniku danych) plik z najlepszymi osobnikami w danym poko-

leniu o nazwie *bes\_popu.dat* zmieniając stosownie jego nazwę, np. na *bes\_popu01.dat*, *bes\_popu02.dat* itd. Plik ten znajduje się w katalogu programu i jest modyfikowany przez program po każdorazowym uruchomieniu obliczeń. Uważaj, żeby plik skopiować, nie zaś przenieść (program może odmówić współpracy...)!

! Jeśli pomylisz się i zapomnisz zapisać plik – powtórz obliczenia i zapisz odpowiedni plik, gdyż wyniki w tabeli **muszą** się zgadzać z plikami.

- Przyjmij wielkości populacji równą 20 i wykonaj obliczenia dla liczby pokoleń:

**10, 25, 50, 100, 250.**

Wypełnij drugą część tabeli 1.

C. Wpisz funkcję nr 2 i wypełnij tabelę 2 oraz zapisz stosowne pliki analogicznie jak w przypadku funkcji nr 1.

## Sprawozdanie

- Sprawozdanie ma być dostarczone wyłącznie w formie elektronicznej.
- Nazwa pliku wg wzorca: OE\_lab2\_Jan\_Iksinski.doc/pdf.
- Strona pierwsza to strona tytułowa.
- W sprawozdaniu należy zamieścić:
  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 5 na jednym wykresie) wygenerowane na podstawie posiadanych plików.
  6. Wnioski z ćwiczenia z podziałem na wnioski dotyczące funkcji 1, funkcji nr 2 oraz wnioski wspólne dla obu funkcji.

## Literatura i źródła

- [1] J. Arabas: Wykłady z algorytmów ewolucyjnych. WNT, Warszawa, 2003.
- [2] J. Arabas, Z. Michalewicz, J. Mulawka: GAVaPS - a Genetic Algorithm with Varying Population Size. ICEC'94, 73-78, 1994.
- [3] Z. Michalewicz: Algorytmy genetyczne + struktury danych = programy ewolucyjne. WNT, Warszawa, 1996.
- [4] L. Rutkowski: Metody i techniki sztucznej inteligencji. PWN, Warszawa, 2006.

## Protokół do laboratorium 2: Wpływ wielkości populacji i liczby pokoleń na skuteczność poszukiwań AE

Imię i nazwisko	Rok ak.	Gr.	Sem.	Komp.	Data	Podpis prowadzącego
_____	20__/__	AB3	I	_____	_____	

Tabela 1. Wyniki dla funkcji pierwszej (dwumodalnej)

Liczebność populacji	Liczba pokoleń	Wartość f. celu najlepszego osobnika	x[1]	x[2]	Znaleziono w pokoleniu:
10	20				
25					
50					
100					
250					
20	10				
	25				
	50				
	100				
	250				

Tabela 2. Wyniki dla funkcji drugiej (wielomodalnej)

Liczebność populacji	Liczba pokoleń	Wartość f. celu najlepszego osobnika	x[1]	x[2]	Znaleziono w pokoleniu:
10	20				
25					
50					
100					
250					
20	10				
	25				
	50				
	100				
	250				

**Notatki:**