



Katedra Wytrzymałości Materiałów
i Metod Komputerowych Mechaniki
www.kwmimkm.polsl.pl

Wydział Mechaniczny Technologiczny
Politechnika Śląska

METODY HEURYSTYCZNE

LABORATORIUM 3: Algorytmy ewolucyjne cz. 1 – wpływ wielkości populacji i liczby pokoleń na skuteczność poszukiwań AE

opracował: dr inż. Witold Beluch
witold.beluch@polsl.pl

Cel ćwiczenia

Wykonując ćwiczenie zapoznasz się z programem *Evolutionary Algorithms* w zakresie obliczeń ewolucyjnych. Oprócz tego przeprowadzisz badania, w wyniku których dla zadanych funkcji celu (przystosowania) określisz wpływ wielkości populacji i liczby pokoleń na skuteczność poszukiwań optimum za pomocą algorytmu ewolucyjnego.

Program *Evolutionary Algorithms* – wprowadzenie

Z zagadnieniami związanymi z algorytmami genetycznymi (AG) i ewolucyjnymi (AE) spotkałeś się na wykładzie z tego przedmiotu. AG i AE należą do grupy tzw. algorytmów optymalizacji globalnej, tzn. są algorytmami wysoce odpornymi na ekstrema lokalne. Do ich działania nie jest potrzebna informacja o gradiencie funkcji celu (przystosowania), która jest często trudna bądź nawet niemożliwa do uzyskania. Z kolei praca na populacji (zbiorze) potencjalnych rozwiązań skutkuje wielokierunkowym przeszukiwaniem przestrzeni poszukiwań, co ułatwia znalezienie optimum globalnego. Algorytmy te zalicza się do tzw. metaheurystyk. Określenie „metaheurystyka” jest połączeniem słów „meta”, czyli „nad” czy też „wyższego poziomu” oraz „heurystyka” (z greckiego *heuriskein* – szukać). Metaheurystyki to zazwyczaj algorytmy służące do rozwiązywania różnorodnych problemów optymalizacji.

Program *Evolutionary Algorithms* jest programem komputerowym autorstwa Piotra Orantka realizującym obliczenia ewolucyjne jak również umożliwiającym pracę ze sztucznymi sieciami neuronowymi. Wersja 7.07 programu jest dostępna wyłącznie jako anglojęzyczna. Program można pobrać ze strony Katedry (adres strony znajduje się na okładce niniejszej instrukcji). Program nie wymaga instalacji. Jeśli program odmawia współpracy należy sprawdzić, czy w ustawieniach systemu Windows jako separator dziesiętny jest ustawiona kropka (domyślny dla polskich ustawień regionalnych przecinek może powodować błędy w programie).

W ramach zapoznania się z programem będziemy rozpatrywać poniższe zadanie optymalizacji.

Zadanie

Znajdź maksimum¹ globalne funkcji dwóch zmiennych:

$$f(x, y) = 10 / (1 + \text{sqr}((x-4)*(x-4) + (y-6)*(y-6)))$$

gdzie **sqr** (*square*) oznacza podniesienie do kwadratu. Nie pomył tej funkcji z podobną – **sqrt** – oznaczającą pierwiastek kwadratowy (*square root*)...

Ograniczenia na zmienne:

$$0 < x < 10, \quad 0 < y < 10.$$

Parametry algorytmu ewolucyjnego:

Używane (aktywne) operatory ewolucyjne:

- mutacja równomierna (*uniform mutation*), type: linear, $p_{um}=0.07$;
- krzyżowanie proste (*simple crossover*), type: linear, $p_{sc}=0.3$;

¹ Algorytmy genetyczne i ewolucyjne z założenia rozwiązują zadania maksymalizacji funkcji. Jeśli konieczna jest minimalizacja pewnej funkcji $f(x)$, to należy maksymalizować funkcję $g(x) = -f(x)$, by po znalezieniu ekstremum zmienić jej znak:

$$\min f(x) = \max g(x) = \max \{-f(x)\}$$

Inne:

- funkcja kary: kara śmierci (*death penalty*);
- selekcja: turniejowa (*tournament selection*), $p_t=0.5$.

Liczebność populacji: 25 osobników. Należy wykonać kilkakrotnie obliczenia dla liczby pokoleń (pętli algorytmu) równej 10, 20 oraz 100.

Zaczynamy



Uruchom program *Evolutionary Algorithms*. Jeśli pojawi się okienko z informacją o programie naciśnij *OK*.

1. Ograniczenia. Przejdź do menu *Data*:

- Sprawdź w *Number of variables*, czy liczba zmiennych projektowych wynosi 2; jeśli tak nie jest, to wyniki mogą być błędne (wykres funkcji wyświetlany jest dla pierwszych 2 zmiennych, obliczenia zaś przeprowadzane są dla wszystkich zmiennych).
- W pozycji *Constraints* ustaw odpowiednie ograniczenia na zmienne x (zmienna 1) i y (zmienna 2).
- Sprawdź (i ewentualnie usuń) czy w pozycjach *Linear constraints* oraz *Nonlinear constraints* nie ma aktywnych ograniczeń liniowych i nieliniowych. Zauważ, że górne pole to tylko baza danych, aktywne ograniczenia są (ewentualnie) widoczne w dolnym polu. W obydwu przypadkach powinna w dolnej części okna widnieć informacja, że liczba aktywnych ograniczeń to 0.

2. Funkcja przystosowania (celu). Przejdź do menu *Tools->Function Editor*. Pojawi się okno edytora funkcji.

- Naciśnięcie przycisku *Read* powoduje wczytanie poprzednio zapisanej w pliku (tekstowym) funkcji. Nie naciskaj tego przycisku po wpisaniu własnej funkcji a przed naciśnięciem *Apply*, bo utracisz to, co wpisałaś/eś! Wpisz funkcję (tylko to, co ma kolor niebieski). Zwróć szczególną uwagę na liczbę nawiasów i znaki mnożenia!
- Naciśnij przycisk *Apply* a następnie *Show*. Pojawi się okno wykresu funkcji. Jeśli po naciśnięciu przycisku *Compute* pojawi się błąd, wróć do okna edytora funkcji i popraw błędny zapis funkcji. Jeśli zaś uaktywni się klawisz *Show* – naciśnij go. Powinien się wyświetlić wykres funkcji. Możesz zmienić typ wykresu z 3D na warstwicowy (*Graph type*, wymaga ponownego przeliczenia), ustawić opcje koloru i wypełnienia (*Options*) oraz obrócić wykres 3-D lub zmienić dokładność wyświetlania wykresu warstwicowego. Zauważ, że w wersji warstwicowej przemieszczenie kursora nad wykresem powoduje wyświetlanie wartości zmiennych oraz wartości funkcji, co pomaga oszacować wartość funkcji np. w okolicy optimum. Sprawdź działanie poszczególnych opcji. By wyjść z okna wykresu naciśnij *Close*.
- Naciśnij *OK* w oknie edytora funkcji.

3. Opcje programu. Przejdź do menu *Options->Computing Options*.

Sprawdź, czy aktywne są opcje: *Save all* (zapisywanie wyników w każdym pokoleniu) oraz *Random* (losowe tworzenie populacji startowej).

4. Ustawienia algorytmu. Wybierz *Settings->Evolutionary algorithm*.

Skorzystaj z danych w opisie zadania i wykonaj poniższe kroki:

- W ramce *Selection* wybierz rodzaj selekcji. Naciśnij przycisk *Settings* i ustaw odpowiednią wartość prawdopodobieństwa.
- Wybierz (uaktywnij) odpowiednie operatory. Ustaw (*Settings*) typ prawdopodobieństwa na liniowy. W ramce *Probability* wpisz w odpowiednich polach dwukrotnie tę samą wartość (oznacza to, że prawdopodobieństwo zmienia się co prawda liniowo, lecz jego wartości w pierwszym i ostatnim pokoleniu są jednakowe, czyli prawdopodobieństwo jest w efekcie stałe).
- Wybierz odpowiedni rodzaj selekcji. Kara śmierci oznacza, że ewentualnie powstałe (w wyniku działania operatorów ewolucyjnych) osobniki niedopuszczalne są eliminowane z populacji i zastępowane nowymi, co zwykle wymaga powtórzenia działania danego operatora.
- W polu *Population* wstaw odpowiednią wartość liczebności populacji.
- Naciśnij *OK*.

5. Obliczenia i wyniki. Przejdź do menu *Compute->Evolutionary algorithm*.

- W ramce *Generations* pozostaw w polu *First generation* numer pierwszego pokolenia równy 1, natomiast jako ostatnie pokolenie (*Last generation*) wpisz liczbę pokoleń z treści zadania.
- W polu *No. of generations without improvement* wpisz 1000 – nie będziemy z tej opcji¹ w ramach niniejszego ćwiczenia korzystać.
- Uruchom obliczenia naciskając *Compute*. Najlepsza wartość znaleziona przez AE podczas działania prezentowana jest (z niezbyt wysoką precyzją) w polu *Best eval in history*. Zauważ, że naciskając ponownie *Compute* uzyskujesz (zazwyczaj) inne wyniki, choć nie były zmieniane żadne parametry. Zapewne pamiętasz, że AE to algorytmy niedeterministyczne (ale nie losowe).
- W celu wyświetlenia obszerniejszej informacji na temat wyników wybierz w głównym menu *Results->Complete*. Najistotniejsze informacje zawierają pola *Best fitness function value* (wynik działania AE), *Best chromosome* (wartości zmiennych projektowych) oraz *Found in generation* (w którym pokoleniu znaleziono najlepsze rozwiązanie²).
- Zmieniaj odpowiednio do treści zadania liczbę pokoleń i wykonuj obliczenia.

6. Zapoznaj się z innymi funkcjami programu.



Pamiętaj – **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*).

¹ Ta opcja pozwala na ustawienie warunku zatrzymania algorytmu w postaci kryterium minimalnej szybkości poprawy. W tym przypadku algorytm jest zatrzymywany, jeśli przez podaną liczbę pokoleń nie zostanie znalezione lepsze rozwiązanie.

² Jeśli jest to ostatnie lub jedno z ostatnich pokoleń to może to sugerować, że przyjęto zbyt małą liczbę pokoleń...

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*), $p_t=0.5$.

B. Wpisz funkcję nr 1.

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

10, 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 pokoleniu 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, 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: [MH_lab3_Jan_Kowalski.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 4 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] Z. Michalewicz: Algorytmy genetyczne + struktury danych = programy ewolucyjne. WNT, Warszawa, 1996.
- [3] L. Rutkowski: Metody i techniki sztucznej inteligencji. PWN, Warszawa, 2006.

Protokół do laboratorium 3: Wpływ wielkości populacji i liczby pokoleń na skuteczność poszukiwań AE

Imię i nazwisko	Rok ak.	Gr.	Sem.	Komp.	Data	Podpis prowadzącego
_____	20__/__	ME3	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				
50					
100					
250					
20	10				
	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				
50					
100					
250					
20	10				
	50				
	100				
	250				

Notatki: