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

Wydział Mechaniczny Technologiczny
Politechnika Śląska

OBLICZENIA EWOLUCYJNE

LABORATORIUM 3: Wpływ operatorów krzyżowania na skuteczność poszukiwań AE

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

Gliwice 2012

Cel ćwiczenia

Wykonując ćwiczenia laboratoryjne przeprowadzisz badania, w wyniku których określisz wpływ różnych operatorów krzyżowania na skuteczność poszukiwań optimum za pomocą algorytmu ewolucyjnego (AE). Sprawdzisz działanie algorytmu dla wybranych funkcji celu stosując zarówno każdy z dostępnych w programie *Evolutionary Algorithms* operatorów krzyżowania osobno jak i wszystkie jednocześnie.

Trochę teorii

Krzyżowanie jest operatorem genetycznym, który tworzy osobniki potomne z wykorzystaniem osobników rodzicielskich. Zazwyczaj dwa osobniki rodzicielskie przyczyniają się do powstania dwóch osobników potomnych, choć istnieje bardzo wiele różnych operatorów krzyżowania – istnieją np. wersje z wieloma osobnikami rodzicielskimi czy też z jednym potomkiem[1].

Z krzyżowaniem jest związany parametr zwany *prawdopodobieństwem krzyżowania* p_c określający, jaka część osobników z populacji pomocniczej (wybranej w procesie reprodukcji) zostanie poddana krzyżowaniu. Dla każdego osobnika jest generowana liczba pseudolosowa r z przedziału [0,1]. Jeśli $r < p_c$ to dany osobnik zostaje wybrany do krzyżowania. Następnie wybrane osobniki są losowo kojarzone w pary. W następnym etapie tworzone są (różnie dla różnych operatorów krzyżowania) osobniki potomne.

Poniżej omówiono tylko te operatory krzyżowania, które zostały zaimplementowane w programie *Evolutionary Algorithms*.

Krzyżowanie proste

Krzyżowanie proste (jednopunktowe) jest operatorem krzyżowania, który może być stosowany zarówno w przypadku kodowania binarnego jak i rzeczywistoliczbowego. Jest to historycznie najstarszy operator krzyżowania zastosowany w algorytmie J. Hollanda [5].

Krzyżowanie proste należy do grupy operatorów krzyżowania wymieniającego, tj. takich, które nie modyfikują wartości poszczególnych genów, a jedynie dokonują ich „przetasowania", składając osobniki potomne z genów osobników rodzicielskich. Dla wybranej pary osobników $\mathbf{X}^1$ i $\mathbf{X}^2$ losuje się punkt rozcięcia c, będący liczbą całkowitą ze zbioru $\{1,2,...,n\text{-}1\}$, gdzie n jest długością osobnika. Następnie zamienia się fragmenty osobników od punktu rozcięcia tworząc osobniki potomne $\mathbf{Y}^1$ i $\mathbf{Y}^2$:

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

Krzyżowanie arytmetyczne

Krzyżowanie arytmetyczne należy do grupy operatorów krzyżowania zmieniających wartości genów osobników rodzicielskich i jako taki nie może być stosowany w przypadku kodowania binarnego. Zapewne domyślasz się, co jest tego przyczyną? Powstałe w wyniku krzyżowania osobniki charakteryzują się tym, że wartości poszczególnych genów zawierają się w przedziale pomiędzy wartościami odpowiednich genów osobników rodzicielskich.

[1] W tomie 1 serii sześciu książek „Algorytmy genetyczne. Kompendium" Tomasza Gwiazdy [4] opisano ok. 180 różnych operatorów krzyżowania dla problemów kodowanych zarówno liczbami binarnymi jak i rzeczywistymi...

W przypadku krzyżowania arytmetycznego w pierwszym etapie generuje się liczbę pseudolosową *k* z przedziału [0,1]. Następnie tworzy się osobniki potomne zgodnie ze schematem:

$$\mathbf{Y}^1 = \mathbf{X}^1 + k\,(\mathbf{X}^2 - \mathbf{X}^1)$$

$$\mathbf{Y}^2 = \mathbf{X}^2 + \mathbf{X}^1 - \mathbf{Y}^1$$

Należy zauważyć, że przyjęcie $k = 0.5$ powoduje powstanie dwóch identycznych osobników potomnych o wartościach poszczególnych genów równych średniej arytmetycznych odpowiednich genów osobników rodzicielskich. Oczywiście nie wydaje się to być dobrym pomysłem, jako że wówczas dwa takie same osobniki wskazują na to samo miejsce w przestrzeni poszukiwań[1].

Krzyżowanie heurystyczne

Krzyżowanie heurystyczne jest operatorem w swym działaniu nieco podobnym do krzyżowania arytmetycznego, również powodując modyfikację genów osobników rodzicielskich (choć nie ich uśrednienie). W związku z tym jest również operatorem, który nie może być stosowany w przypadku kodowania binarnego.

Operator krzyżowania heurystycznego jest dość nietypowy, gdyż tworzy maksymalnie jednego potomka. Powyższe stwierdzenie oznacza, że może w ogóle nie utworzyć osobnika potomnego. Osobnik potomny jest tworzony jako liniowa kombinacja genów osobników rodzicielskich:

$$\mathbf{Y} = \mathbf{X}^2 + k\,(\mathbf{X}^2 - \mathbf{X}^1)$$

gdzie *k* jest liczbą pseudolosową z przedziału [0,1] a rodzic $\mathbf{X}^2$ jest nie gorszy niż rodzic $\mathbf{X}^1$ (wartość funkcji celu $f(\mathbf{X}^2) \geq f(\mathbf{X}^1)$). Jeśli utworzony potomek nie jest dopuszczalny wówczas generuje się nową wartość *k* i tworzy innego potomka. Jeżeli po *m* próbach[2] operator nie utworzy osobnika dopuszczalnego to potomek nie jest w ogóle tworzony. W takim przypadku zaleca się dodatkowo skorzystanie z innego operatora krzyżowania.

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*).

[1] Przestrzeń poszukiwań (ang. *search space*) to zbiór wszystkich możliwych rozwiązań danego problemu.
[2] Parametr *m* nazywany jest też *wskaźnikiem cierpliwości krzyżowania heurystycznego*.

Do wykonania

Przeprowadź obliczenia dla następujących funkcji:

Funkcja nr 1:

$$f_1(x,y) = 8/\left(1+\mathrm{sqr}\left((x-6)\cdot(x-6)+(y-6)\cdot(y-6)\right)\right)+$$
$$+5/\left(1+\mathrm{sqr}\left((x-3)\cdot(x-3)+(y-3)\cdot(y-3)\right)\right)$$

Ograniczenia na zmienne: $0 < x < 10, \quad 0 < y < 10$.

Funkcja nr 2 (funkcja Rastrigina[1]):

$$f_2(x,y) = 30-\left(x\cdot x-10\left(\cos(2\cdot\pi\cdot x)\right)\right)-\left(y\cdot y-10\left(\cos(2\cdot\pi\cdot y)\right)\right)$$

Ograniczenia na zmienne: $-5 < x < 5, \quad -5 < y < 5$.

A. Wpisz funkcję nr 1 oraz:

- Przyjmij liczebność populacji (*population size*) równą **20** i liczbę pokoleń (*Last generation*) równą **50**.
- Wprowadź następujące parametry:
  - mutacja równomierna (*uniform mutation*), p_{um}=0.03;
  - pozostałe operatory mutacji nieaktywne;
  - funkcja kary: kara śmierci (*death penalty*);
  - selekcja: turniejowa (*tournament selection*), p_{ts}=0.1.
- Przeprowadź **trzykrotnie** obliczenia dla następujących operatorów krzyżowania:
  - jedynie krzyżowanie proste (*simple crossover*), p_{sc}=0.3;
  - jedynie krzyżowanie arytmetyczne (*arithmetic crossover*), p_{ac}=0.3;
  - jedynie krzyżowanie heurystyczne (*heuristic crossover*), p_{hc}=0.3;
  - wszystkie powyższe (p_{sc}=0.1, p_{ac}=0.1, p_{hc}=0.1).
- Wypełnij tabelę 1.

! 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 (*bes_popu.dat*). Jeśli zapomnisz zapisać plik dla danego wiersza tabeli – powtórz obliczenia i zapisz odpowiedni plik, gdyż wyniki w tabeli muszą się zgadzać z plikami.

[1] Funkcja Rastrigina (tu: dla 2 zmiennych) jest jedną z typowych funkcji testowych, tj. takich, na których testuje się działanie różnych algorytmów optymalizacji – między innymi algorytmów ewolucyjnych. Funkcja ta ma wiele ekstremów, w tym jedno globalne (w oryginale – w punkcie x_i=0, i=1...n o wartości 0). W niniejszym przykładzie wykorzystano funkcję Rastrigina z przeciwnym niż „normalnie" znakiem – w oryginalnej funkcji poszukuje się jej minimum lokalnego, a jak pamiętasz algorytmy ewolucyjne „ze swej natury" poszukują maksimum globalnego. Ponadto do oryginalnej funkcji dodano stałą wartość równą 50, by (w związku z wymogami programu) sprostać założeniu, że funkcja celu jest dodatnia w całej dziedzinie.

B. Wpisz funkcję nr 2 oraz:

- Przyjmij liczebność populacji (*population size*) równą **30** i liczbę pokoleń (*Last generation*) równą **100**.
- Wprowadź następujące parametry:
  - mutacja równomierna (*uniform mutation*), p_{um}=0.03;
  - pozostałe operatory mutacji nieaktywne;
  - funkcja kary: kara śmierci (*death penalty*);
  - selekcja: turniejowa (*tournament selection*), p_{ts}=0.1.
- Przeprowadź **trzykrotnie** obliczenia dla następujących operatorów krzyżowania:
  - jedynie krzyżowanie proste (*simple crossover*), p_{sc}=0.3;
  - jedynie krzyżowanie arytmetyczne (*arithmetic crossover*), p_{ac}=0.3;
  - jedynie krzyżowanie heurystyczne (*heuristic crossover*), p_{hc}=0.3;
  - wszystkie powyższe (p_{sc}=0.1, p_{ac}=0.1, p_{hc}=0.1).
- Wypełnij tabelę 2.

Sprawozdanie

- Sprawozdanie ma być dostarczone wyłącznie w formie elektronicznej.
- Nazwa pliku wg wzorca: OE_lab3_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 12 na jednym wykresie) wygenerowane na podstawie posiadanych plików oraz wartości uśrednione z 3 przebiegów (po 4 na jednym wykresie).
  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.

[4] T. Gwiazda: Algorytmy genetyczne. Kompendium. T. 1. Operator krzyżowania dla problemów numerycznych. Wydawnictwo Naukowe PWN, 2007.

[5] J. H. Holland: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence, MIT Press, Cambridge, 1992 (1975).

Protokół do laboratorium 3: Wpływ operatorów krzyżowania 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)

L.p.	Rodzaj krzyżo-wania	Wartość f. celu najlepszego osob-nika	x[1]	x[2]	Znaleziono w pokoleniu:
1	proste				
2					
3					
4	arytmetyczne				
5					
6					
7	heurystyczne				
8					
9					
10	p+a+h				
11					
12					

Tabela 2. Wyniki dla funkcji drugiej (wielomodalnej)

L.p.	Rodzaj krzyżo-wania	Wartość f. celu najlepszego osob-nika	x[1]	x[2]	Znaleziono w pokoleniu:
1	proste				
2					
3					
4	arytmetyczne				
5					
6					
7	heurystyczne				
8					
9					
10	p+a+h				
11					
12					

Notatki (na drugiej stronie)**:**