

Metody Rozmyte i Algorytmy Ewolucyjne

Projekty

Piotr Kaczyński

8 czerwca 2007

1 Wstęp

Projekty poniższe polegają na implementacji, a przede wszystkim analizie algorytmu ewolucyjnego jako metody (lub jednej z metod) rozwiązania praktycznego problemu. Projekty realizowane są w ramach przedmiotu *Metody Rozmyte i Algorytmy Ewolucyjne* na Wydziale Matematyczno-Przyrodniczym. Szkoła Nauk Ścisłych na Uniwersytecie Kardynała Stefana Wyszyńskiego w Warszawie.

2 Wymagania

Do zaliczenia projektu konieczne jest

- Wykonanie wszystkich poleceń projektowych,
- Napisanie sprawozdania z projektu zawierającego
 - Opis zastosowanych algorytmów ewolucyjnych (w szczególności operatorów, strategii i parametrów),
 - Informację o wydajności algorytmu (jak długo trwały obliczenia etc),
 - Wyniki przeprowadzonych analiz i porównań (najlepiej w postaci wykresów),
 - *Wnioski z projektu*
- Przesłanie gotowego projektu na adres e-mail prowadzącego *przed* oddaniem,
- Oddanie projektu (pokazanie kodu źródłowego, odpowiedź ustna)

Projekt może być wykonany na dowolnej platformie sprzętowej i programistycznej.

Do danego projektu nie może być przypisanych więcej niż *dwie* osoby — niemniej jednak projekty te muszą być wykonywane *samodzielnie*, bez jakichkolwiek konsultacji z osobami wykonującymi ten sam projekt. Mnogość możliwych rozwiązań i analiz umożliwia bardzo prostą weryfikację samodzielności wykonania projektu.

Zrozumienie treści zadania nie jest jego częścią, a więc wszelkie niejasności mogą chętnie rozwiązać podczas konsultacji. Niezrozumienie treści i błędne rozwiązanie zadania nie jest jednak powodem usprawiedliwienia podczas oddawania projektu.

3 Treści projektów

Projekt 1

Analiza algorytmu genetycznego dla różnych rodzajów kodowania (reprezentacji genów). Należy rozważyć zadania całkowitoliczbowe i przeanalizować wpływ kodowania na jakość algorytmu. Uwzględnić kodowanie naturalne w \mathbb{R}^n oraz binarne (z krzyżowaniem w układzie naturalnym i kodzie Graya). Przeanalizować wpływ różnych operatorów genetycznych. Jako zadania przykładowe obrać:

- Problem plecakowy - chromosom n -genowy z genami binarnymi lub chromosom 1-genowy z liczbą całkowitą powstałą z wyrażenia chromosomu w kodowaniu binarnym w systemie dziesiętnym (wybrać $n = 32$),
- Zagadnienie programowania liniowego całkowitoliczbowego (przypadek 2-wymiarowy i przypadek większej ilości wymiarów (powyżej 10)),

Projekt 2

Analiza algorytmu genetycznego dla różnych rodzajów kodowania (reprezentacji genów). Należy rozważyć zadania ciągłe i przeanalizować wpływ kodowania na jakość algorytmu. Uwzględnić kodowanie naturalne w \mathbb{R}^n oraz kodowanie binarne z przekształcaniem do systemu IEEE 754. Przeanalizować wpływ różnych operatorów genetycznych. Jako zadania przykładowe obrać:

- Problem plecakowy - chromosom n -genowy z genami binarnymi lub chromosom 1-genowy z liczbą całkowitą powstałą z wyrażenia chromosomu w kodowaniu binarnym w systemie dziesiętnym,
- Zagadnienie programowania nieliniowego
 - Dla funkcji Griewanka

$$GR(x) = 1 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + \frac{1}{40} \sum_{i=1}^n x_i^2$$
$$-40 \leq x_i \leq 40$$

$$n = 2, 10, 20$$

- Dla funkcji Ackleya

$$AC(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20$$
$$-30 \leq x_i \leq 30$$

$$n = 2, 10, 20$$

Projekt 3

Analiza szybkości działania i jakości algorytmu genetycznego dla różnych długości reprezentacji genów. Przeanalizować jak należy zmieniać parametry operatorów genetycznych w zależności od wielkości zadania. Zbadać wpływ liczebności populacji na czas działania algorytmu. Jako zadania przykładowe obrać:

- Problem plecakowy,
- Zagadnienie programowania nieliniowego
 - Dla funkcji Griewanka

$$GR(x) = 1 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + \frac{1}{40} \sum_{i=1}^n x_i^2$$
$$-40 \leq x_i \leq 40$$

$$n = 2, 10, 20$$

- Dla funkcji Ackleya

$$AC(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20$$
$$-30 \leq x_i \leq 30$$

$$n = 2, 10, 20$$

Projekt 4

Analiza zachowań algorytmu genetycznego dla różnych częstotliwości operatora mutacji. Przeanalizować działanie algorytmu z wyłączonym i włączonym operatorem mutacji. Przeanalizować możliwość zastąpienia tego operatora przez inne operacje genetyczne (odpowiednio dobrane krzyżowanie i selekcję). Przeanalizować różnice dla kodowania binarnego i ciągłego. Zbadać wpływ przy różnych wielkościach populacji i ilości wymiarów. Jako zadania przykładowe obrać:

- Problem plecakowy (kodowanie binarne),
- Zagadnienie programowania nieliniowego (kodowanie ciągłe)
 - Dla funkcji Griewanka

$$GR(x) = 1 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + \frac{1}{40} \sum_{i=1}^n x_i^2$$
$$-40 \leq x_i \leq 40$$

$$n = 2, 10, 20$$

- Dla funkcji Ackleya

$$AC(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20$$
$$-30 \leq x_i \leq 30$$

$$n = 2, 10, 20$$

Projekt 5

Porównanie algorytmu ewolucyjnego z innymi algorytmami optymalizacji lokalnej dla zadań całkowitoliczbowych. Należy porównać algorytm ewolucyjny (z jak najlepiej dobranymi operatorami genetycznymi) pod względem wydajności i jakości uzyskiwanych rozwiązań z algorytmami

- Hill-climbing,
- Symulowanego wyżarzania,
- Sympleksu nieliniowego Nelder-Meada (odpowiednio zmodyfikować do rozwiązywania zadań całkowitoliczbowych),

Należy przeanalizować możliwość stosowania powyższych algorytmów z opcją wielokrotnego startu. Jako przykładowe zadania obrać:

- Problem plecakowy,
- Problem komiwojażera

Projekt 6

Porównanie algorytmu ewolucyjnego z innymi algorytmami optymalizacji lokalnej dla zadań ciągłych. Należy porównać algorytm ewolucyjny (z jak najlepiej dobranymi operatorami genetycznymi) pod względem wydajności i jakości uzyskiwanych rozwiązań z algorytmami

- Hill-climbing (największego wzrostu z wyliczeniem aproksymacji gradientu),
- Symulowanego wyżarzania,
- Sympleksu nieliniowego Nelder-Meada,

Należy przeanalizować możliwość stosowania powyższych algorytmów z opcją wielokrotnego startu. Jako przykładowe zadanie obrać:

- Zagadnienie programowania nieliniowego
 - Dla funkcji Griewanka

$$GR(x) = 1 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + \frac{1}{40} \sum_{i=1}^n x_i^2$$
$$-40 \leq x_i \leq 40$$

$$n = 2, 10, 20$$

- Dla funkcji Ackleya

$$AC(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20$$
$$-30 \leq x_i \leq 30$$

$$n = 2, 10, 20$$

Projekt 7

Porównanie algorytmu ewolucyjnego z innymi algorytmami optymalizacji lokalnej dla zadań ciągłych i dyskretnych. Należy porównać algorytm ewolucyjny (z jak najlepiej dobranymi operatorami genetycznymi) pod względem wydajności i jakości uzyskiwanych rozwiązań z algorytmami

- Monte-Carlo
- Symulowanego wyżarzania,

Należy przeanalizować możliwość stosowania powyższych algorytmów z opcją wielokrotnego startu. Jako przykładowe zadanie obrać:

- Problem plecakowy,
- Zagadnienie programowania nieliniowego
 - Dla funkcji Griewanka

$$GR(x) = 1 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + \frac{1}{40} \sum_{i=1}^n x_i^2$$
$$-40 \leq x_i \leq 40$$

$$n = 2, 10, 20$$

- Dla funkcji Ackleya

$$AC(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20$$
$$-30 \leq x_i \leq 30$$

$$n = 2, 10, 20$$

Projekt 8

Zaproponować co najmniej 2 różne operatory krzyżowania i 2 różne operatory mutacji (możliwe różne rodzaje kodowania) wykorzystujące specyfikę problemu dla zadania komiwojażera. Przeanalizować ich wpływ na uzyskiwane rozwiązania dla różnych wymiarów problemu przy różnej liczebności populacji. Zbadać wpływ nacisku selektywnego. Porównać z rozwiązaniami uzyskiwanymi przy pomocy standardowych operatorów genetycznych dla kodowania całkowitoliczbowego. Określić najlepsze parametry dla tego typu zadań.

Projekt 9

Zaproponować co najmniej 2 różne operatory krzyżowania i 2 różne operatory mutacji (możliwe różne rodzaje kodowania) wykorzystujące specyfikę problemu dla zagadnienia plecakowego. Przeanalizować ich wpływ na uzyskiwane rozwiązania dla różnych wymiarów problemu przy różnej liczebności populacji. Zbadać wpływ nacisku selektywnego. Porównać z rozwiązaniami uzyskiwanymi przy pomocy standardowych operatorów genetycznych dla kodowania binarnego. Określić najlepsze parametry dla tego typu zadań.

Projekt 10

Porównać działanie standardowego algorytmu ewolucyjnego z algorytmem stosującym ewolucję lamarkowską realizowaną przy pomocy algorytmu hill climbing (zagadnienie dyskretne) lub metodą optymalizacji gradientowej (najszybszy wzrost lub quasi-newton dla zagaznienia ciągłego). Porównanie przedstawić dla zagadnień:

- Problemu plecakowego (n -genowy osobnik binarny),
- Programowania nieliniowego
 - Dla funkcji Griewanka

$$GR(x) = 1 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + \frac{1}{40} \sum_{i=1}^n x_i^2$$
$$-40 \leq x_i \leq 40$$

$$n = 2, 10, 20$$

- Dla funkcji Ackleya

$$AC(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20$$
$$-30 \leq x_i \leq 30$$

$$n = 2, 10, 20$$

Projekt 11

Porównać działanie standardowego algorytmu ewolucyjnego z algorytmem koewolucyjnym. Porównać skuteczność algorytmu koewolucyjnego ze względu na ilość podpopulacji (przy zachowaniu stałej liczebności wszystkich osobników sumując po wszystkich podpopulacjach). Wykonać testy dla różnej liczebności całej populacji (suma osobników we wszystkich podpopulacjach), ilość osobników emigrujących i imigrujących. Testy przeprowadzić dla zagadnień:

- Problemu plecakowego (n -genowy osobnik binarny),
- Programowania nieliniowego
 - Dla funkcji Griewanka

$$GR(x) = 1 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + \frac{1}{40} \sum_{i=1}^n x_i^2$$
$$-40 \leq x_i \leq 40$$

$$n = 2, 10, 20$$

- Dla funkcji Ackleya

$$AC(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20$$
$$-30 \leq x_i \leq 30$$

$$n = 2, 10, 20$$

Dla chętnych: Wykonanie projektu w środowisku wieloprocesorowym.

Projekt 12

Zbadać, czy rozkład uzyskiwanych rozwiązań w algorytmie genetycznym jest zbieżny do postaci funkcji celu. Należy więc porównać histogram znajdujących rozwiązań z funkcją przystosowania. Przeanalizować zadania jedno i dwuwymiarowe w przypadkach

- Funkcji wypukłych ciągłych (o małej i dużej stałej Lipschitza),
- Funkcji niewypukłych (o małej i dużej stałej Lipschitza) z co najmniej trzema optimumami lokalnymi,
- Z ograniczeniami kosztowymi i ograniczeniami tworzącymi niewypukły obszar rozwiązań dopuszczalnych

Pokazać wpływ nacisku selektywnego. Operatory genetyczne dobrać jak najlepiej (uzyskać jak najszybszą zbieżność) i dla nich przeprowadzać analizy. Przeanalizować co stanie się, gdy któryś z operatorów nie jest używany. Przeanalizować rozkłady generowane przez najlepszych osobników bądź całą populację z ostatniej generacji.

Projekt 13

Zaproponować algorytm genetyczny tworzący model NAR (MISO) w postaci statycznej sieci neuronowej o jednej warstwie ukrytej. Funkcjami aktywacji powinny być funkcja sigmoidalna (np. tanh) oraz funkcja liniowa w odpowiednio warstwie ukrytej i wyjściowej sieci. Należy uwzględnić istnienie niezerowej wartości średniej dla każdej z warstw (wejście jednego z neuronów każdej z warstw stale równe 1).

Algorytm genetyczny powinien być odpowiedzialny za

- Dobór liczby neuronów w warstwie ukrytej sieci,
- Dobór wag sieci neuronowej,
- Wybór wejść modelu (tzn. które zmienne objaśniające należy uwzględnić, maksymalny rząd przyjąć jako z góry ograniczony),

Algorytm genetyczny należy przetestować na dwóch zbiorach danych

- Wygenerowanych „ręcznie” z symulacji następującego równania różnicowego (można wygenerować jeden długi zestaw danych)

$$y_{k+1} = 0.1y_{k-1} + 0.8 \tanh(0.1y_k + 0.2) + 2 \tanh(0.2y_{k-3} - 0.3) + 0.1 * \xi_k$$

gdzie ξ_k jest realizacją standardowej normalnej zmiennej losowej.

- Danych rzeczywistych (zastosowanie algorytmu genetycznego dobranego do danych poprzednich), np. danych z Giełdy Papierów Wartościowych (WIG i inne indeksy).

Pokazać dokładność tak uzyskanych modeli NAR.

Projekt 14

Zaproponować algorytm genetyczny do trenowania klasyfikujących sieci neuronowych. Postać sieci może być dowolna. Przykładowe sieci można utworzyć do klasyfikacji

- Cyfr w różnych krojach czcionki drukowanej,
- Pisma odręcznego,

Pokazać skuteczność trenowania algorytmem ewolucyjnym (ew. porównać z sieciami uzyskanymi innymi metodami optymalizacji, przy użyciu gotowych narzędzi).

Projekt 15

Zaproponować jak najlepszy algorytm gry w „kartofla” (w celu wyjaśnienia zasad gry należy skontaktować się z prowadzącym). Algorytm ten oprzeć na algorytmie genetycznym — zaproponować odpowiednie operatory

genetyczne i kodowanie. Wydajność algorytmu porównywać z algorytmem rozwiązywania gier $\alpha - \beta$ dla różnych głębokości przeszukiwanego drzewa. Pokazać wpływ zastosowania operatorów genetycznych (ich używania bądź nieużywania).

Projekt 16

Zaproponować jak najlepszy algorytm gry w kółko i krzyżyk na nieograniczonej planszy z wygraną przy pięciu symbolach w jednej linii. Algorytm ten oprzeć na algorytmie genetycznym — zaproponować odpowiednie operatory genetyczne i kodowanie. Wydajność algorytmu porównywać z algorytmem rozwiązywania gier $\alpha - \beta$ dla różnych głębokości przeszukiwanego drzewa. Pokazać wpływ zastosowania operatorów genetycznych (ich używania bądź nieużywania).

Projekt 17

Zaproponować jak najlepszy algorytm gry w warcaby. Algorytm ten oprzeć na algorytmie genetycznym — zaproponować odpowiednie operatory genetyczne i kodowanie. Wydajność algorytmu porównywać z algorytmem rozwiązywania gier $\alpha - \beta$ dla różnych głębokości przeszukiwanego drzewa. Pokazać wpływ zastosowania operatorów genetycznych (ich używania bądź nieużywania).

Projekt 18

Zaproponować algorytm rozwiązujący problem ułożenia zadania Programowania Liniowego. Danymi wejściowymi są

- Wymiar początkowy zadania,
- Ilość kroków prowadzących do znalezienia rozwiązania,
- Ilość zmiennych sztucznej bazy w tablicy początkowej metody sympleks,
- Zadany typ rozwiązania zagadnienia (skończone rozwiązanie optymalne, nieskończone rozwiązanie optymalne, nieskończona ilość rozwiązań optymalnych, zadanie sprzeczne).

Algorytm ewolucyjny powinien zwracać zagadnienie w postaci niestandardowej wyłącznie z ograniczeniami nierównościami. Funkcja celu powinna uwzględniać

- Prostotę rachunków przy kolejnych tablicach sympleks (jak najwięcej liczb całkowitych w tablicy),
- Sensowność parametrów (jak najmniejsze),

przy spełnieniu ograniczeń zadanych na wejściu (można uwzględnić pewną ich relaksację). Należy zaproponować odpowiednią funkcję celu i dobrać odpowiedni algorytm ewolucyjny.

Projekt 19

Dany jest następujący problem; Znaleźć optymalne rozłożenie zadanej ilości elips na płaszczyźnie \mathbb{R}^2 tak, aby okrąg opisany na zbiorze tychże elips (czyli okrąg, którego promień jest wystarczająco duży, aby pomieścić wszystkie elipsy) miał jak najmniejszą średnicę. Parametry i ilość elips są danymi wejściowymi zadania.

Dane są:

- k elips o równaniach

$$\frac{x^2 - x_{k0}}{a^2} + \frac{y^2 - y_{k0}}{b^2} = 1$$

gdzie a, b są danymi wejściowymi zadania

Algorytm powinien zwrócić

- Dobrany środek (x_{k0}, y_{k0}) dla każdej z elips,
- Środek i promień najmniejszego okręgu opisanego na tych elipsach

Należy zaproponować odpowiednią funkcję celu oraz algorytm genetyczny rozwiązujący podane zagadnienie. Przetestować jego skuteczność na rozwiązany, prostym przypadku i pokazać działanie na bardziej skompliko-

wanych przykładach. Dopuszczalne jest użycie pewnych przybliżeń (np. do znalezienia środka okręgu opisanego na elipsach).

Projekt 20

Zastosowanie algorytmu ewolucyjnego do harmonogramowania zadań.

Projekt 21

Napisać aplikację okienkową prezentującą działanie algorytmu genetycznego. Zaimplementować możliwość wizualizacji działania dla problemów jedno i dwuwymiarowych oraz możliwość ustalania parametrów algorytmu takich jak typ operatora krzyżowania, typ operatora mutacji, typ selekcji z odpowiednimi dla nich parametrami, liczebność populacji. Uwzględnić kilka funkcji testowych.

Należy również zaimplementować wizualizację prostych wykresów agregujących przebieg symulacji algorytmu (najlepszy osobnik, średnia etc).

Uwaga! Projekt ten wymaga umiejętności programistycznych (bądź chęci ich nauki) w języku C++ oraz chęci nauki bibliotek OpenGL i QT. W związku z tym osoba wykonująca ten projekt zwolniona jest z pisania jakiegokolwiek sprawozdania z projektu. W przypadku satysfakcjonującego mnie wykonania można otrzymać dodatkowe benefity ¹.

¹Projekt ten może być realizowany przez 2 osoby