

Sztuczne inteligencja i inżynieria wiedzy

laboratorium

Ćwiczenie 1. Algorytmy genetyczne

opracowanie: P.Myszkowski. H.Kwaśnicka, aktualizacja: 28.02.2013

Cel ćwiczenia

Zapoznanie się z metaheurystyką algorytmów genetycznych w praktyczny sposób poprzez samodzielną implementację.

Realizacja ćwiczenia

- Zapoznanie się z metaheurystyką algorytmów genetycznych
- Określenie problemu optymalizacyjnego (3 do wyboru: kolorowanie grafu, optymalizacja funkcji lub sztuczny malarz) do rozwiązania
- Zbudowanie algorytmu genetycznego: osobnik, funkcja oceny, krzyżowanie, mutacja, selekcja
- Implementacja modelu
- Zbadanie wpływu różnych parametrów (prawd. mutacji, krzyżowania, selekcji, rozmiar populacji, liczba pokoleń) na efektywność i skuteczność metody
- Sporządzenie sprawozdania z ćwiczenia
- Pokazanie na wykresach zmianę wartości przystosowania (wartość optymalizowanej funkcji celu) w poszczególnych pokoleniach: najlepszy osobnik, średnia wartość w populacji, odchylenie standardowe (lub najgorszy osobnik) (**uwaga:** ten punkt nie obowiązuje w ćwiczeniu *Sztuczny artysta*)
- Porównaj działanie algorytmu genetycznego z wybraną przez siebie, nieewolucyjną metodą optymalizacji (wybierz jedną: *przegląd zupełny*, *metoda gradientowa*, *metoda bezgradientowa*, *metoda losowego przeszukiwania*, *hill climbing*). Zwróć uwagę na uzyskany wynik, czas działania oraz liczbę wartościowań optymalizowanej funkcji celu (**uwaga:** ten punkt nie obowiązuje w ćwiczeniu *Sztuczny artysta*)
- Pokaż najciekawsze, Twoim zdaniem, wyniki
- Omów uzyskane wyniki.
- Raport z ćwiczenia powinien zawierać wszystkie punkty (różnica jest w ćwiczeniu *Sztuczny malarz*)

Informacje pomocnicze

Algorytm genetyczny (AG) jest metodą, w której naśladuje ewolucję naturalną metodą ciśnienia selekcyjnego i doboru naturalnego. Aby ją zastosować należy określić zbiór cechy rozwiązania (osobnik), sposoby jego zmiany (mutacja), łączenia (krzyżowania) oraz oceny jakości rozwiązania (funkcja oceny). Algorytm genetyczny opiera się na następującym schemacie:

```
begin
t:=0;
initialise(pop0);
evaluate(pop0);
    while (not stop_condition) do
        begin
```

```

        popt+1 := selection(popt);
        popt+1 := crossover(popt+1);
        popt+1 := mutation(popt+1);
        evaluate(popt+1);
        t:=t+1;
    end
end

```

AG wstępnie inicjalizuje (zwykle losowo) populację rozwiązań. Następnie ocenia jakość poszczególnych osobników. W kolejnym kroku sprawdzane są warunki zatrzymania (czy osiągnięto akceptowalne rozwiązanie i/lub limit pokoleń został przekroczony). Wybór osobników (metoda ruletki lub turnieju) do następnej populacji następuje przed działaniem operatora krzyżowania i mutacji, które tworzą osobniki nowego pokolenia. Dalej, następuje ocena nowych rozwiązań i cykl metody się zamyka przy sprawdzaniu warunków zatrzymania.

Ze względu na stochastyczny charakter AG, przy podawaniu wyniku należy go oprzeć na minimum 10 uruchomieniach i wtedy podajemy średnią wraz z odchyleniem standardowym. Dodatkowo możemy podać maksymalną i minimalną osiągniętą wartość.

Problem 1. Kolorowanie grafu

Problem kolorowania wierzchołków grafu (podstawowa wersja: nieskierowanego nieważonego) jest NP-trudnym problemem optymalizacyjnym, który ma przełożenie na bardzo wiele problemów praktycznych: od harmonogramowania po problemy alokacji zasobów i wiele innych. Założeniem ćwiczenia jest zapoznanie się z problemem i przebadanie skuteczności AG dla 5 wybranych grafów oraz porównanie z jedną z metod kolorowania grafu (może być metoda zachłanne *LargestFirst*, *SmallestFirst* lub *SaturatedLargestFirst*)

Opis problemu: <http://mathworld.wolfram.com/VertexColoring.html>, ewentualnie http://en.wikipedia.org/wiki/Graph_coloring

Zbiór grafów testowych: <http://mat.gsia.cmu.edu/COLOR/instances.html> (uwaga! Niektóre są bardzo duże)

„mała” odpowiedź: http://www.google.pl/url?sa=t&rct=j&q=graph%20coloring%20problem%20genetic%20algorithm&source=web&cd=3&cad=rja&ved=0CD4QFjAC&url=http%3A%2F%2Fciteseerx.ist.psu.edu%2Fviewdoc%2Fdownload%3Fdoi%3D10.1.1.28.9267%26rep%3Drep1%26type%3Dpdf&ei=SO8lUeT4JuLe4QTE_IDYAg&usg=AFQjCNGWC6JGHltSV0Bv6aY02fgZCY24RA&bvm=bv.42661473.d.d2k

Problem 2. Optymalizacja funkcji

Problem optymalizacji funkcji jest problemem benchmarkowym, dla którego bada się skuteczność metod optymalizacji. Celem zadania jest przeszukanie przestrzeni za pomocą GA dla 5 wybranych funkcji testowych i porównania skuteczności z innymi metodami optymalizacji.

Przykładowe funkcje de Jonga <http://www2.denizyuret.com/pub/aitr1569/node19.html>

Większy zbiór funkcji:

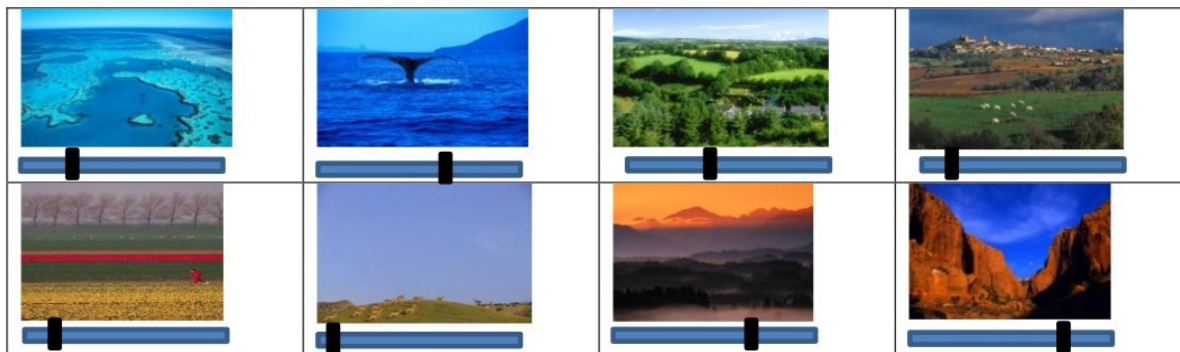
http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO_files/Page364.htm

Problem 3. Sztuczny malarz

Problem wymaga wizualizacji. Zakoduj obraz w postaci osobnika/chromosomu i odpowiednie operatory genetyczne działające na osobnikach. Ewoluuuj niewielką liczbę osobników-obrazów tak, abyś mógł je wyświetlić jednocześnie na ekranie komputera, np. 8 osobników obrazów, po 4 w rzędzie, dwa rzędy obrazków. Pod każdym obrazkiem umieść pasek oceny funkcji przystosowania, czyli oceny na ile dany obrazek Ci się podoba.

Przykładowy rozkład ekranu: generowane przez komputer (algorytm genetyczny) obrazy :) a pod nimi pasek do oceny, na ile nam się dany obraz podoba. Mamy wyświetloną całą populację

konkurujących (ewoluujących) obrazów wraz z paskiem oceny pod każdym osobnikiem.



Można spróbować też zautomatyzować funkcje oceny i wtedy możemy mieć podejście wzorowane na <http://rogeralsing.com/2008/12/07/genetic-programming-evolution-of-mona-lisa/>.

W przypadku wyboru innego problemu, należy skonsultować ten wybór wcześniej z prowadzącym.

Ocena ćwiczenia

2pkt	Zbudowanie modelu algorytmu genetycznego
2pkt	Implementacja algorytmu genetycznego
2pkt	Zbadanie działania na 2 innych plikach danych
2pkt	Zbadanie wpływu prawd. krzyżowania i mutacji na wyniki działania metody
2pkt	Zbadanie wpływu rozmiaru populacji i liczbie pokoleń na wyniki działania metody

Uwaga! Przy testach należy brać pod uwagę przynajmniej 10 uruchomień i uśrednianie wyniku (nie dotyczy zadania sztucznego malarza, chociaż i tam warto sprawdzić powtarzalność wyników)

Literatura

1. Materiały do ćwiczenia: dokument na Boardzie w katalogu /Kwasnicka/Sztuczna_Inteligencja/ Calosc_ZeszytNaukowyNr1_Ostatni.pdf
2. Książka dostępna pod adresem: http://www.google.pl/url?sa=t&source=web&cd=2&ved=0CCcQFjAB&url=http%3A%2F%2Fwww.boente.eti.br%2Ffuzzy%2Febook-fuzzy-mitchell.pdf&ei=DKGMTteRC46M4gSlxImtCQ&usq=AFOjCNGfU_i6oWloPS75Dj6Pa216CrB83Q
3. Arabas J. Wykłady z algorytmów ewolucyjnych (<http://staff.elka.pw.edu.pl/~jarabas/ksiazka.html>)
2. Goldberg D. *Algorytmy genetyczne i ich zastosowanie*
3. Michalewicz Z. *Algorytmy genetyczne + struktury danych = programy ewolucyjne*
4. Michalewicz Z., Fogel D.B. *Jak to rozwiązać, czyli nowoczesna heurystyka*