Algorytmy i modele inspirowane biologicznie

1. Algorytmy ewolucyjne i ich mechanizmy

Maciej Komosiński

Fundusze Europejskie
Polska Cyfrowa

Rzeczpospolita Polska

Unia Europejska
Europejski Fundusz Rozwoju Regionalnego



Cele przedmiotu

- Założenie:
  - znana podstawowa wiedza o optymalizacji, w tym ewolucyjnej

Cele przedmiotu

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia

- Założenie:
  - znana podstawowa wiedza o optymalizacji, w tym ewolucyjnej
- Cele:
  - integracja dotychczasowej wiedzy, syntetyczne i dojrzałe spojrzenie na optymalizację
  - dyskusja mechanizmów działania – nie tylko „bo tak jest", ale też „dlaczego?"
  - poznanie bardziej zaawansowanych mechanizmów ewolucyjnych
  - proponowanie własnych pomysłów
  - nauczenie się argumentowania „za" (zalety) i „przeciw" (wady)

Cele przedmiotu

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia

- Założenie:
  - znana podstawowa wiedza o optymalizacji, w tym ewolucyjnej
- Cele:
  - integracja dotychczasowej wiedzy, syntetyczne i dojrzałe spojrzenie na optymalizację
  - dyskusja mechanizmów działania – nie tylko „bo tak jest", ale też „dlaczego?"
  - poznanie bardziej zaawansowanych mechanizmów ewolucyjnych
  - proponowanie własnych pomysłów
  - nauczenie się argumentowania „za" (zalety) i „przeciw" (wady)
- Laboratoria:
  - poznanie jednego z najpopularniejszych frameworków ewolucyjnych w Pythonie
  - doświadczenie zupełnie innego problemu optymalizacji – uwolnienie się od intuicji zdominowanych i ograniczonych przez TSP (nawet QAP daje często odwrotne konkluzje!)
  - koewolucja konstrukcji i neuronowego systemu sterującego – robotyka ewolucyjna
  - porównanie skuteczności różnych topologi przeszukiwania w tej samej przestrzeni rozwiązań
  - ocena różnych krajobrazów optymalizacji w tej samej przestrzeni rozwiązań
  - transformacje między topologiami przeszukiwania i przestrzeniami rozwiązań

Przypomnienie z poprzedniego semestru (i ew. uzupełnienie) #1

- Czym różni się *Random Search* od *Random Walk*?
- Jakie wady i zalety mają *Random Search* i *Full Search*?
  Stwórz algorytm łączący zalety obu, eliminując zarazem ich wady.
- Jakie znasz dwa warianty algorytmu lokalnego przeszukiwania?
- Czy wiadomo, że któryś z nich jest lepszy (daje ostatecznie lepsze wyniki – chodzi o jakość, nie o czas działania)?
- Jakie są metody intensyfikacji i dywersyfikacji w *Tabu Search* i *Simulated Annealing*?
- Jaki algorytm jest najbardziej intensyfikujący (eksploatacja), a jaki – najbardziej dywersyfikujący (eksploracja)?

Przypomnienie z poprzedniego semestru (i ew. uzupełnienie) #2

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia

- Jakie znasz operatory krzyżowania w algorytmach ewolucyjnych/genetycznych?
- Jakie znasz metody selekcji w algorytmach ewolucyjnych?
- Jakie znasz metody zwiększania różnorodności rozwiązań podczas ewolucji?
- Czym się różni architektura *steady state* od *generational replacement*?
- Jakie znasz metody radzenia sobie z ograniczeniami?

Dwa scenariusze zastosowania algorytmów optymalizacji

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia

Zastosowania algorytmów optymalizacji można z grubsza podzielić na interakcyjne (*on-line*) i wsadowe (*off-line*).

Przy podejściu wsadowym zainteresowani jesteśmy najlepszym znalezionym podczas pracy algorytmu rozwiązaniem. Przy podejściu interakcyjnym („bieżącym") interesuje nas, by dany algorytm optymalizacji dawał jak najlepsze wyniki przez cały czas. Dla oceny działania w trybach *on-line* i *off-line* De Jong zaproponował specjalne wskaźniki [Gol03, str. 125]; wymyśl dwa najprostsze.

Warianty algorytmów ewolucyjnych

- Algorytmy genetyczne (GA)
- Strategie ewolucyjne (ES)
- Programowanie ewolucyjne (EP)
- Programowanie genetyczne (GP)
- Systemy klasyfikatorowe (CFS) i ewolucyjne uczenie maszynowe (GBML)
- Architektury koewolucyjne
- ...

GA: John Holland (1973, 1975), David Goldberg (1989)
EP: Lawrence Fogel (1963), David Fogel (1992)
ES: Ingo Rechenberg (1973), Thomas Bäck (1996)
GP: John Koza (1992)

Przypomnienie: budowa algorytmu i parametry

Główna pętla:

$t := 0$
inicjalizuj $P(t)$
oceń $P(t)$
dopóki (**nie** warunek-zakończenia)
{
    $t := t + 1$
    wybierz $P(t)$ z $P(t-1)$
    zmień $P(t)$
    oceń $P(t)$
}

Przypomnienie: budowa algorytmu i parametry

Parametry:

- wielkość populacji *POPSIZE*
- prawdopodobieństwo krzyżowania *PXOVER*
- prawdopodobieństwo mutacji *PMUT*

- wybór kryterium stopu
- wybór mechanizmu selekcji (pozytywnej i ew. negatywnej)
- wybór wartości parametrów mechanizmu selekcji

Rola selekcji

- Do czego potrzebna jest selekcja w algorytmie ewolucyjnym?
- Co działoby się, gdyby selekcja była czysto losowa?
- Co działoby się, gdyby selekcja była deterministyczna i dawała równą szansę każdemu osobnikowi?
- Czy można wyrazić siłę presji selekcyjnej liczbowo?

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia

Dyskusja o rodzajach niedeterminizmu i roli niedeterminizmu w algorytmach i w informatyce.

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia

Dyskusja o rodzajach niedeterminizmu i roli niedeterminizmu w algorytmach i w informatyce.

Rozważ cztery generatory sekwencji losowych: oparty o kolejne liczby i *modulo*, kiepski pseudolosowy, dobry pseudolosowy, i prawdziwie losowy.

Rola niedeterminizmu w algorytmice

Dyskusja o rodzajach niedeterminizmu i roli niedeterminizmu w algorytmach i w informatyce.

Rozważ cztery generatory sekwencji losowych: oparty o kolejne liczby i *modulo*, kiepski pseudolosowy, dobry pseudolosowy, i prawdziwie losowy.

```
int random1(int n)
    static int c=-1
    c++
    c=c%n
    return c
```

```
int random2(int n)
    static int c=0
    c=(c·...+...)%n
    return c
```

```
int random3(int n)
    //very
    //complicated
    //logic
    return ...
```

```
int random4(int n)
    return ☼%n
```

Rola niedeterminizmu w algorytmice

Dyskusja o rodzajach niedeterminizmu i roli niedeterminizmu w algorytmach i w informatyce.

Rozważ cztery generatory sekwencji losowych: oparty o kolejne liczby i *modulo*, kiepski pseudolosowy, dobry pseudolosowy, i prawdziwie losowy.

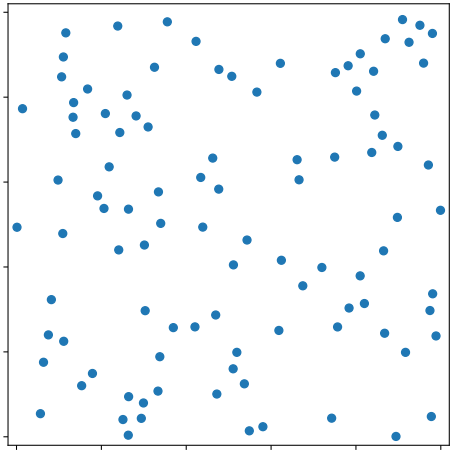
Jaki rodzaj losowości (o ile w ogóle) będzie adekwatny w poniższych zastosowaniach?

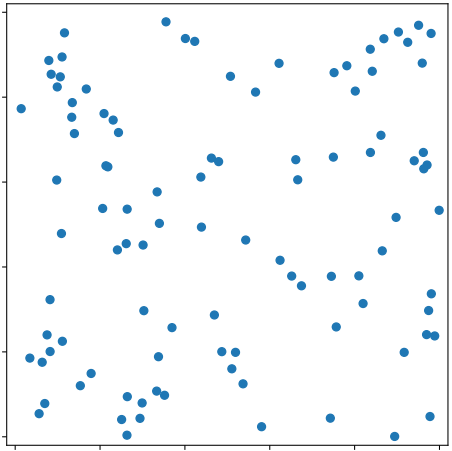
- przykład suplementacji (np. 4 substancje A,B,C,D – średnio co 2 dni każda, nie znamy interakcji)
- przykład dawania prezentów
- przykład *dithering*'u sygnału (dźwięk, obraz, ...) i zaokrąglania
- i wreszcie przykład algorytmu...

Rola niedeterminizmu w algorytmice – „losowość”

Które próbkowanie przejawia większy „bias”?

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia

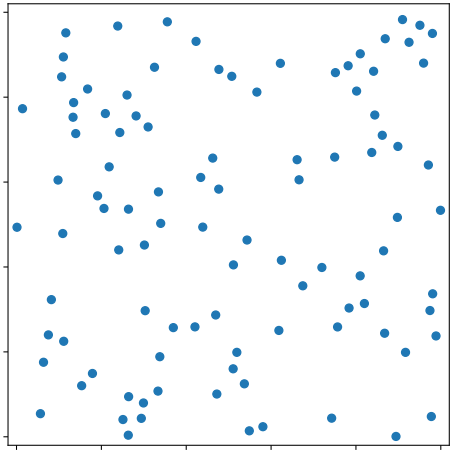


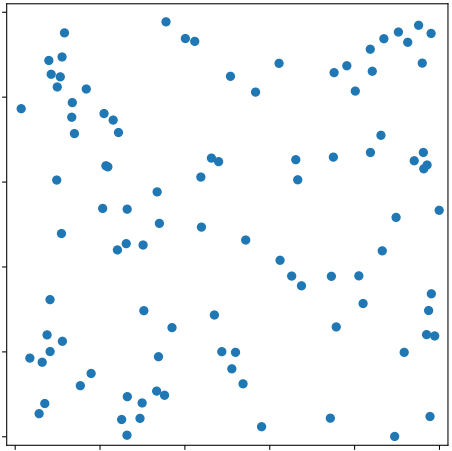


Rola niedeterminizmu w algorytmice – „losowość”

Które próbkowanie przejawia większy „bias”?

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia



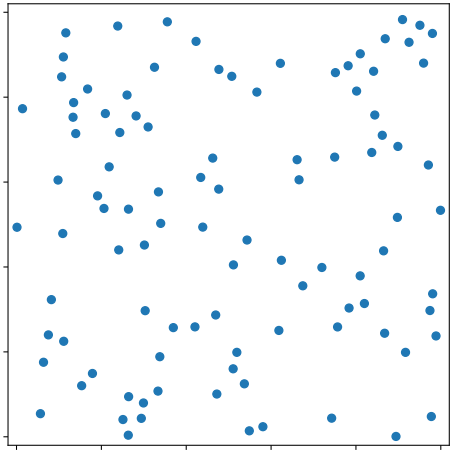


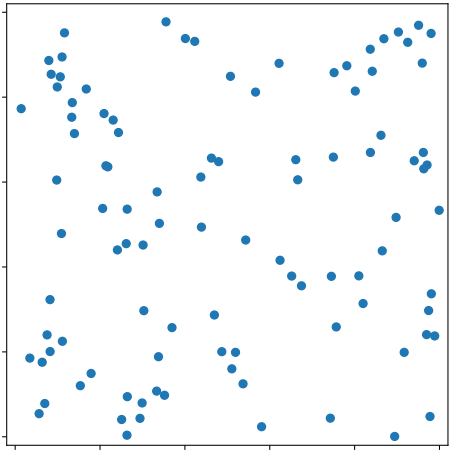
zależność vs. niezależność

Rola niedeterminizmu w algorytmice – „losowość”

Które próbkowanie przejawia większy „bias”?

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia





zależność vs. niezależność

Losowość redukuje „bias”? Zdefiniuj go: zależność albo niejednorodność

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia

Dyskusja o rodzajach niedeterminizmu i roli niedeterminizmu w algorytmach i w informatyce.

Rola niedeterminizmu w algorytmice – wnioski

Dyskusja o rodzajach niedeterminizmu i roli niedeterminizmu w algorytmach i w informatyce.

- wymyślasz jakiś algorytm, np. optymalizacyjny (albo inny). Co skłoni Cię do użycia w nim funkcji random()?

Rola niedeterminizmu w algorytmice – wnioski

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia

Dyskusja o rodzajach niedeterminizmu i roli niedeterminizmu w algorytmach i w informatyce.

- wymyślasz jakiś algorytm, np. optymalizacyjny (albo inny). Co skłoni Cię do użycia w nim funkcji random()?
  - przykłady: *Greedy* i *Steepest* z wieloma sąsiadami o takiej samej jakości, indukcja drzew decyzyjnych z wieloma atrybutami o równej entropii, ...

Rola niedeterminizmu w algorytmice – wnioski

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia

Dyskusja o rodzajach niedeterminizmu i roli niedeterminizmu w algorytmach i w informatyce.

- wymyślasz jakiś algorytm, np. optymalizacyjny (albo inny). Co skłoni Cię do użycia w nim funkcji random()?
  - przykłady: *Greedy* i *Steepest* z wieloma sąsiadami o takiej samej jakości, indukcja drzew decyzyjnych z wieloma atrybutami o równej entropii, ...
  - spróbuj zupełnie „udeterministycznić" SA. Czy będą jakieś negatywne konsekwencje?

Rola niedeterminizmu w algorytmice – wnioski

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia

Dyskusja o rodzajach niedeterminizmu i roli niedeterminizmu w algorytmach i w informatyce.

- wymyślasz jakiś algorytm, np. optymalizacyjny (albo inny). Co skłoni Cię do użycia w nim funkcji random()?
  - przykłady: *Greedy* i *Steepest* z wieloma sąsiadami o takiej samej jakości, indukcja drzew decyzyjnych z wieloma atrybutami o równej entropii, ...
  - spróbuj zupełnie „udeterministycznić" SA. Czy będą jakieś negatywne konsekwencje?

Rola niedeterminizmu w algorytmice – wnioski

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia

Dyskusja o rodzajach niedeterminizmu i roli niedeterminizmu w algorytmach i w informatyce.

- wymyślasz jakiś algorytm, np. optymalizacyjny (albo inny). Co skłoni Cię do użycia w nim funkcji random()?
  - przykłady: *Greedy* i *Steepest* z wieloma sąsiadami o takiej samej jakości, indukcja drzew decyzyjnych z wieloma atrybutami o równej entropii, ...
  - spróbuj zupełnie „udeterministycznić" SA. Czy będą jakieś negatywne konsekwencje? [DS90]
- w jakich sytuacjach „nieskrępowana", pełna losowość jest korzystna?

Rola niedeterminizmu w algorytmice – wnioski

Dyskusja o rodzajach niedeterminizmu i roli niedeterminizmu w algorytmach i w informatyce.

- wymyślasz jakiś algorytm, np. optymalizacyjny (albo inny). Co skłoni Cię do użycia w nim funkcji random()?
  - przykłady: *Greedy* i *Steepest* z wieloma sąsiadami o takiej samej jakości, indukcja drzew decyzyjnych z wieloma atrybutami o równej entropii, ...
  - spróbuj zupełnie „udeterministycznić" SA. Czy będą jakieś negatywne konsekwencje? [DS90]
- w jakich sytuacjach „nieskrępowana", pełna losowość jest korzystna?
- dlaczego przejmujemy się ...1080777777796980348..., ale nie ...1080735172796980348... ?

Rola niedeterminizmu w algorytmice – wnioski

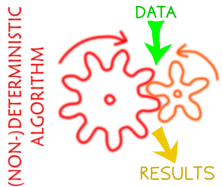
Dyskusja o rodzajach niedeterminizmu i roli niedeterminizmu w algorytmach i w informatyce.

- wymyślasz jakiś algorytm, np. optymalizacyjny (albo inny). Co skłoni Cię do użycia w nim funkcji random()?
  - przykłady: *Greedy* i *Steepest* z wieloma sąsiadami o takiej samej jakości, indukcja drzew decyzyjnych z wieloma atrybutami o równej entropii, ...
  - spróbuj zupełnie „udeterministycznić" SA. Czy będą jakieś negatywne konsekwencje? [DS90]
- w jakich sytuacjach „nieskrępowana", pełna losowość jest korzystna?
- dlaczego przejmujemy się ...1080777777796980348..., ale nie ...1080735172796980348... ?

Rola niedeterminizmu w algorytmice – wnioski

Dyskusja o rodzajach niedeterminizmu i roli niedeterminizmu w algorytmach i w informatyce.

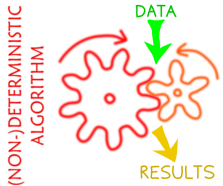
- wymyślasz jakiś algorytm, np. optymalizacyjny (albo inny). Co skłoni Cię do użycia w nim funkcji random()?
  - przykłady: *Greedy* i *Steepest* z wieloma sąsiadami o takiej samej jakości, indukcja drzew decyzyjnych z wieloma atrybutami o równej entropii, ...
  - spróbuj zupełnie „udeterministycznić" SA. Czy będą jakieś negatywne konsekwencje? [DS90]
- w jakich sytuacjach „nieskrępowana", pełna losowość jest korzystna?
- dlaczego przejmujemy się ...1080777777796980348..., ale nie ...1080735172796980348... ? Pierwsza sekwencja jest wprost z RNG, druga jest „poprawiona" (czy na pewno?)
- kiedy prawdziwa losowość jest preferowana względem dobrej pseudolosowości?



Rola niedeterminizmu w algorytmice – wnioski

Dyskusja o rodzajach niedeterminizmu i roli niedeterminizmu w algorytmach i w informatyce.

- wymyślasz jakiś algorytm, np. optymalizacyjny (albo inny). Co skłoni Cię do użycia w nim funkcji random()?
  - przykłady: *Greedy* i *Steepest* z wieloma sąsiadami o takiej samej jakości, indukcja drzew decyzyjnych z wieloma atrybutami o równej entropii, ...
  - spróbuj zupełnie „udeterministycznić" SA. Czy będą jakieś negatywne konsekwencje? [DS90]
- w jakich sytuacjach „nieskrępowana", pełna losowość jest korzystna?
- dlaczego przejmujemy się ...1080777777796980348..., ale nie ...1080735172796980348... ? Pierwsza sekwencja jest wprost z RNG, druga jest „poprawiona" (czy na pewno?)

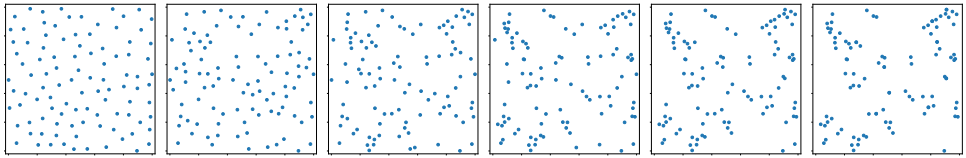


- kiedy prawdziwa losowość jest preferowana względem dobrej pseudolosowości?
- i wreszcie: czy powinien dostać to finansowanie i dlaczego??

Rola niedeterminizmu w algorytmice – „losowość”

Które próbkowanie przestrzeni rozwiązań jest najlepsze?

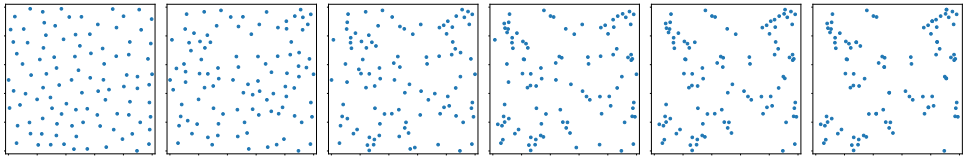
Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia



Rola niedeterminizmu w algorytmice – „losowość”

Które próbkowanie przestrzeni rozwiązań jest najlepsze?

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia



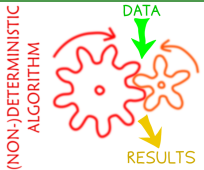
← lepsza eksploracja, ryzyko interakcji z danymi

słaba eksploracja, niezależność

gorsza eksploracja, ryzyko interakcji z danymi →

Rola niedeterminizmu w algorytmice – podsumowanie

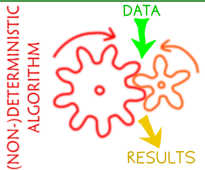
Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia



Analogia do szyfru z kluczem jednorazowym: DATA to komunikat do zaszyfrowania, a ALGORITHM to (losowy) klucz szyfrujący.

Rola niedeterminizmu w algorytmice – podsumowanie

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia



Analogia do szyfru z kluczem jednorazowym: DATA to komunikat do zaszyfrowania, a ALGORITHM to (losowy) klucz szyfrujący.

Prawdziwa losowość albo dobra pseudolosowość:

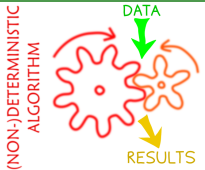
- idealnie lub wystarczająco dobrze separuje dane i logikę algorytmu,
- może być „niesprawiedliwa" i powodować powstawanie artefaktów ze względu na niekontrolowane fluktuacje.

„Ulepszona" (ograniczona) losowość albo kiepska pseudolosowość:

- zwiększa ryzyko interakcji danych i logiki algorytmu (co może sprzyjać artefaktom),
- w zależności od konkretnych celów, może być bardziej „sprawiedliwa" i równomierna, niż $\leftarrow$.

Rola niedeterminizmu w algorytmice – podsumowanie

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia



Analogia do szyfru z kluczem jednorazowym: DATA to komunikat do zaszyfrowania, a ALGORITHM to (losowy) klucz szyfrujący.

Prawdziwa losowość albo dobra pseudolosowość:

- idealnie lub wystarczająco dobrze separuje dane i logikę algorytmu,
- może być „niesprawiedliwa" i powodować powstawanie artefaktów ze względu na niekontrolowane fluktuacje.

„Ulepszona" (ograniczona) losowość albo kiepska pseudolosowość:

- zwiększa ryzyko interakcji danych i logiki algorytmu (co może sprzyjać artefaktom),
- w zależności od konkretnych celów, może być bardziej „sprawiedliwa" i równomierna, niż $\leftarrow$.

- Kompromis: decydujesz zależnie od wymagań.
- Artefakty mogą na różne nieoczywiste sposoby wpływać na eksplorację i eksploatację.
- Artefakty mogą polepszać lub pogarszać rezultaty optymalizacji dla danej instancji.

Selekcja – popularne metody

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia

f_i – ocena osobnika i-tego ($i = 1..POPSIZE$)
e_i – oczekiwana liczba jego kopii w nowej (kolejnej) populacji, $e_i = POPSIZE \cdot f_i / \sum f_j$

- Wybór losowy proporcjonalnie do jakości z powtórzeniami (tj. ze zwracaniem), nazywany potocznie zasadą ruletki: osobnikom przydzielane są pola na tarczy. Wielkość pól jest proporcjonalna do f_i. Następnie kręci się tarczą „ruletki" $POPSIZE$ razy wybierając wylosowanego osobnika. Ta sama zasada, ale lepsze własności wyboru losowego*: *stochastic universal sampling*.**
- Wybór losowy według reszt bez powtórzeń: każdy osobnik otrzymuje tyle kopii w nowej populacji, ile wynosi część całkowita jego e_i. Pozostałe* wolne miejsca zapełnia się podejmując losową decyzję, dla każdego osobnika z prawdopodobieństwem będącym częścią ułamkową jego e_i, czy ma trafić do nowej populacji. Przykład: 4 osobniki, $\boldsymbol{f}$=[1,3,5,6].
- Wybór według turniejów losowych (parametr: k – wielkość turnieju). Bardziej staranny wariant* tej techniki dba o to, żeby każdy osobnik wziął udział w tej samej liczbie turniejów.

*Por. wcześniejsza dyskusja o losowości, jej własnościach i motywacjach jej ograniczania.
**https://en.wikipedia.org/wiki/Stochastic_universal_sampling

Selekcja – inne techniki

- Wybór deterministyczny według reszt: każdy osobnik otrzymuje tyle kopii w nowej populacji, ile wynosi część całkowita jego e_i, a pozostałe wolne miejsca w populacji zapełnia się w kolejności malejących części ułamkowych e_i osobników.
- Wybór losowy według reszt z powtórzeniami: każdy osobnik otrzymuje tyle kopii w nowej populacji, ile wynosi część całkowita jego spodziewanej liczby kopii (e_i). Pozostałe miejsca zapełnia się według zasady ruletki, wykalibrowanej według części ułamkowych e_i.
- Wybór porządkowy: osobnikom nadaje się rangi odpowiadające ich pozycjom w uszeregowaniu od najlepszego do najgorszego. Wybór następuje na podstawie funkcji prawdopodobieństwa określonej nie na ocenach osobników, a na ich pozycjach w rankingu. Stosuje się różne funkcje prawdopodobieństwa – linowe i nieliniowe, przy czym parametry tych funkcji pozwalają na sterowanie naporem selekcyjnym.

Zadanie: podziel wymienione 6 technik na dwie kategorie – zależnie od sposobu, w jaki wykorzystują wartości funkcji przystosowania.

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia

- Model elitarny (elitarystyczny): spełnia oczekiwanie, że proces selekcji nie powinien prowadzić do utraty najlepszego znalezionego dotąd osobnika. Jeśli taki osobnik nie trafia w naturalny (wynikający z zastosowanej metody selekcji) sposób do kolejnej populacji, włącza się go do niej i w ten sposób informacja o najlepszym dotychczasowym rozwiązaniu zostaje zawsze zachowana.
- Model ze współczynnikiem zatłoczenia (ze ściskiem): podobnie jak w naturze, gdzie gatunki wypełniające niszę ekologiczną muszą walczyć o ograniczone zasoby – w modelu ze ściskiem (ang. *crowding model*) nowe osobniki zastępują osobniki stare (z poprzedniej populacji) z uwzględnieniem ich podobieństwa, tzn. nowe osobniki zajmują miejsce najbardziej podobnych do nich starych osobników. Współczynnik ścisku (parametr) wpływa na sposób zastępowania osobników [DJ75; Mah92].

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia

W poniższych metodach selekcji, części populacji (podpopulacje) mogą być niezależnie przetwarzane – metody te mogą zatem pełnić również rolę schematu rozpraszania i zrównoleglania ewolucji.

- Model wyspowy: dzieli całą populację na podpopulacje, w których działa wybrany schemat selekcji (np. turniejowa, ruletkowa czy inny). Ewolucja odbywa się niezależnie na każdej wyspie, z okresową migracją części genotypów między wyspami. Jakie to ma efekty?
- Selekcja konwekcyjna: inaczej niż w tradycyjnym modelu wyspowym, podział na podpopulacje następuje wedle podobieństwa wartości funkcji celu rozwiązań. Jakie to ma efekty?

Meta-schematy selekcji

W poniższych metodach selekcji, części populacji (podpopulacje) mogą być niezależnie przetwarzane – metody te mogą zatem pełnić również rolę schematu rozpraszania i zrównoleglania ewolucji.

- Model wyspowy: dzieli całą populację na podpopulacje, w których działa wybrany schemat selekcji (np. turniejowa, ruletkowa czy inny). Ewolucja odbywa się niezależnie na każdej wyspie, z okresową migracją części genotypów między wyspami. Ten model zwiększa zdolność eksploracji.
- Selekcja konwekcyjna: inaczej niż w tradycyjnym modelu wyspowym, podział na podpopulacje następuje wedle podobieństwa wartości funkcji celu rozwiązań. Selekcja konwekcyjna poprawia zdolność eksploracji AE dzięki odpowiedniemu zbalansowaniu presji selekcyjnej [KU17; KM18]. Animacje sposobu działania tutaj. Dyskusja: jak zadziała ten meta-schemat w wariantach *EqualNumber* i *EqualWidth* [KM18, Rys. 3], kiedy w podpopulacjach selekcja będzie losowa (np. rozmiar turnieju $= 1$), w porównaniu do modelu wyspowego i do standardowego, jednopopulacyjnego AE z losową selekcją?

Selekcja negatywna

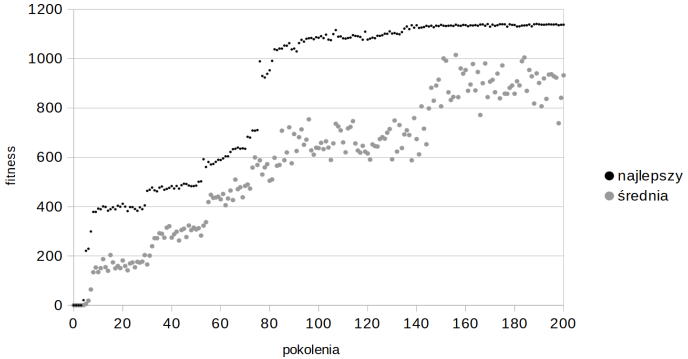
Niekiedy (zależnie od przyjętej architektury AE) oprócz zastosowania selekcji pozytywnej konieczne jest wykorzystanie również selekcji negatywnej.

Jej rolą jest zrobienie miejsca w populacji na nowe genotypy: selekcja negatywna decyduje, które genotypy usunąć z populacji. Można stosować podobne mechanizmy, jak przy selekcji pozytywnej; dwie przykładowe, naiwne metody to usuwanie najgorszego i usuwanie losowego.

Uruchomiłeś/aś algorytm ewolucyjny z selekcją ruletkową

i otrzymujesz taki przebieg:

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia

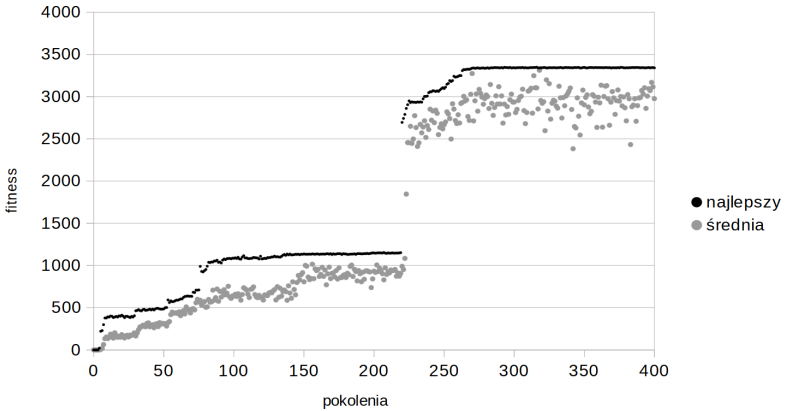


Pytanie 1. Czy to jest dobry moment, żeby zakończyć optymalizację?

Pytanie 2. Czy z tego przebiegu można wyciągnąć wniosek, że coraz mniejsze (i w końcu zerowe) polepszenia wynikają z tego, że coraz trudniej w otoczeniu populacji znaleźć lepsze rozwiązania?

Odpowiedź na pytanie 1: przyszłość jest nieznana!

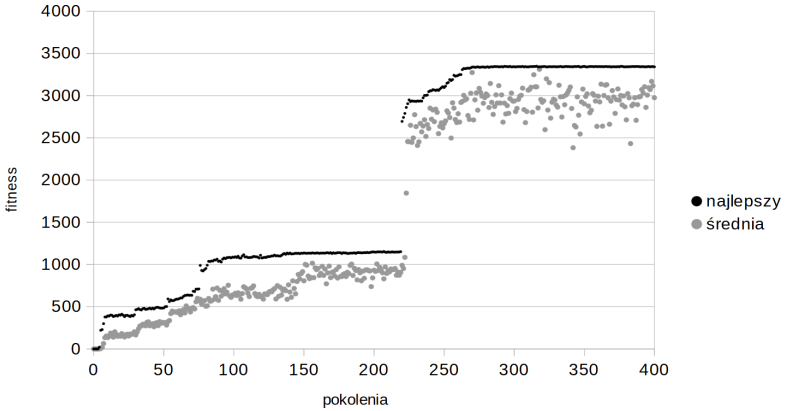
Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia



Jakie inne kryteria (oprócz ustabilizowanych wartości funkcji celu) można wykorzystać, żeby zbudować lepszy warunek stopu i uniknąć powyższej sytuacji?

Odpowiedź na pytanie 1: przyszłość jest nieznana!

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia



Miary różnorodności rozwiązań w populacji oraz związek z presją selekcyjną i potencjałem operatorów do zmiany rozwiązań. Ewentualnie wiedza (choćby pobieżna) o krajobrazie przystosowania.

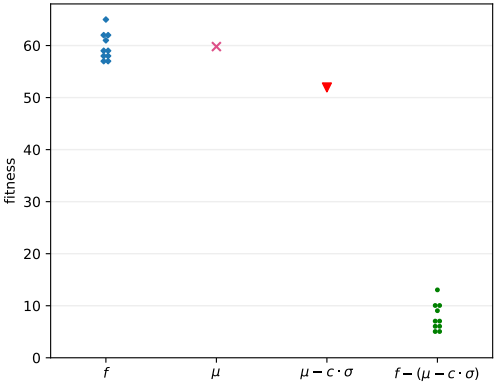
Skalowanie

Dyskusja: skąd wynika potrzeba skalowania? Analiza zachowania selekcji ruletkowej na początku ewolucji oraz w późniejszych etapach (por. poprzedni wykres).

- Skalowanie liniowe: $f' = af + b$. Współczynniki a i b dobiera się w ten sposób, by przystosowanie f' osobnika najlepszego było zadaną wielokrotnością (na przykład $2\times$) dopasowania osobnika „średniego". Po skalowaniu mogą pojawić się ujemne wartości przystosowania – można wówczas wyzerować je lub dokonać innego przekształcenia liniowego.
- Skalowanie potęgowe: $f' = f^k$. Współczynnik k zależy od konkretnego rozwiązywanego zadania, stąd metoda ta nie jest specjalnie przydatna.
- Skalowanie σ-obcinające (obcinanie na poziomie zależnym od odchylenia standardowego). Wartości przystosowania zależą nie tylko od wartości pierwotnych ocen osobników, ale także od rozkładu ocen w populacji. Wyznacza się średnie dopasowanie populacji μ oraz odchylenie standardowe ocen w populacji σ, a przystosowanie (w przypadku maksymalizacji) $f' = f - (\mu - c \cdot \sigma)$. Ujemne wartości f' zastępuje się zerem. Współczynnik c określa napór selekcyjny: im większe c, tym mniejszy napór.

Skalowanie σ-obcinające – przykład

10 osobników o ocenach 57, 57, 58, 58, 59, 59, 61, 62, 62, 65.
Średnia μ wynosi 59.8, a odchylenie standardowe $\sigma = 2.6$.



Rysunek: Wykres skalowania (z odcięciem na poziomie zależnym od odchylenia standardowego) dla współczynnika $c = 3$.

Zadanie obliczeniowe (i interpretacyjne)

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia

Przed skalowaniem, prawdopodobieństwo wybrania osobnika i-tego p_i to iloraz wartości dopasowania osobnika (f_i) i sumy dopasowań wszystkich osobników ($j = 1..n$).

$$p_i = \frac{f_i}{\sum_{j=1}^{n} f_j} = \frac{f_i}{n\mu}$$

Ile wyniesie p'_i, jeśli $f'_i = f_i - (\mu - c\sigma)$?

Wyraź p'_i jako funkcję p_i i innych możliwie łatwo interpretowalnych wyrażeń.

Zadanie obliczeniowe (i interpretacyjne) – odpowiedź

$$p'_i = \frac{f_i - \mu}{nc\sigma} + \frac{1}{n} \quad = \quad \frac{\mu}{c\sigma}\left(p_i - \frac{1}{n}\right) + \frac{1}{n}$$

Zinterpretuj powyższy wzór i odnieś do poprzedniego wykresu skalowania.

- Czym jest $\frac{1}{n}$?
- Rozważ sytuację, gdzie μ jest małe a σ duże (początek ewolucji) oraz odwrotną, gdy ewolucja jest już zaawansowana.
- Czy to skalowanie posiada punkt neutralności, w którym nie ma ono żadnego wpływu?

Zadanie obliczeniowe (i interpretacyjne) – odpowiedź

$$p_i' = \frac{f_i - \mu}{nc\sigma} + \frac{1}{n} \quad = \quad \frac{\mu}{c\sigma}\left(p_i - \frac{1}{n}\right) + \frac{1}{n}$$

Zinterpretuj powyższy wzór i odnieś do poprzedniego wykresu skalowania.

- Czym jest $\frac{1}{n}$?
- Rozważ sytuację, gdzie μ jest małe a σ duże (początek ewolucji) oraz odwrotną, gdy ewolucja jest już zaawansowana.
- Czy to skalowanie posiada punkt neutralności, w którym nie ma ono żadnego wpływu?

Przypomnienie: jaki brzmiał jeden z pomysłów pozwalający reprezentować „presję selekcyjną" jako pojedynczą wartość?

Sens skalowania ocen

Mechanizm skalowania pomaga utrzymać stały napór selekcyjny podczas całej ewolucji, niezależnie od własności optymalizowanej funkcji. Kontrolowanie naporu selekcyjnego jest bardzo ważne – bez tego niemożliwa jest efektywna optymalizacja.

Stosowanie skalowania ma sens, jeśli użyta metoda selekcji...

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia

Mechanizm skalowania pomaga utrzymać stały napór selekcyjny podczas całej ewolucji, niezależnie od własności optymalizowanej funkcji. Kontrolowanie naporu selekcyjnego jest bardzo ważne – bez tego niemożliwa jest efektywna optymalizacja.

Stosowanie skalowania ma sens, jeśli użyta metoda selekcji wykorzystuje ilorazowość skali wartości przystosowania (a np. selekcja turniejowa i rankingowa nie wykorzystują).

Średnia ocena jako „punkt neutralny” populacji

Czy średnia jest najlepszym pomysłem?

Średnia ocena jako „punkt neutralny" populacji

Czy średnia jest najlepszym pomysłem?

Inna (lepsza niż σ-obcinające?) funkcja skalowania: prawdopodobieństwo wybrania osobnika,

$$p'_i = \frac{1}{1 + \exp(\frac{f_i - M}{\sigma})}$$

f_i – wartość funkcji celu osobnika,
M – **mediana** funkcji celu wszystkich osobników w populacji,
σ – odchylenie standardowe funkcji celu w populacji.

Dla maksymalizacji należy odwrócić znak argumentu exp().

- Minimalny wpływ krańcowo dobrze/źle przystosowanych osobników (w ciągu całego procesu ewolucji), co zmniejsza problem przedwczesnej zbieżności. Minimalny – bo mediana (krańcowy osobnik wpłynie tylko na umiarkowane zwiększenie σ).
- Skuteczne rozdzielenie osobników przeciętnie przystosowanych na dwie grupy (powyżej i poniżej mediany).
- Na początku ewolucji – zwykle duże σ, czyli małe zróżnicowanie p'_i osobników. Pod koniec ewolucji – zbieżność więc małe σ, zatem osobniki lepsze dużo silniej promowane.

Przykładowy rozkład ocen w populacji podczas ewolucji

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia

Zobacz [KU17, górny wykres na rys. 9].

Jak wyjaśnisz utrzymujące się przez długi czas poziome linie (grupy osobników o podobnych, szczególnych wartościach funkcji celu)? Postaraj się podać co najmniej dwie możliwe przyczyny.

Krzyżowanie

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia

Przypomnienie: jednopunktowe, wielopunktowe, jednorodne, ...

Dyskusja: czy krzyżowanie jest niezbędne w algorytmie ewolucyjnym?

Krzyżowanie

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia

Przypomnienie: jednopunktowe, wielopunktowe, jednorodne, ...

Dyskusja: czy krzyżowanie jest niezbędne w algorytmie ewolucyjnym?

Dyskusja: czy AE bez krzyżowania to to samo, co wiele niezależnie uruchomionych LS?

Krzyżowanie

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia

Przypomnienie: jednopunktowe, wielopunktowe, jednorodne, ...

Dyskusja: czy krzyżowanie jest niezbędne w algorytmie ewolucyjnym?

Dyskusja: czy AE bez krzyżowania to to samo, co wiele niezależnie uruchomionych LS?

Dyskusja: mutacja i krzyżowanie a eksploracja i eksploatacja.

Krzyżowanie – metody zaawansowane

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia

- *Tasowanie*: dodatkowy mechanizm [ECS89] stosowany przy niektórych rodzajach krzyżowania. Polega on na losowej zamianie miejsc bitów w łańcuchach rodziców (u obu tak samo), dokonaniu krzyżowania i przywróceniu pierwotnego porządku bitów u potomków (zastanów się, po co stosuje się taką operację i dla jakich operatorów krzyżowania ma sens?)
- Krzyżowanie adaptacyjne: w genotypie oprócz bitów rozwiązania mogą być przechowywane informacje o miejscach, w których nastąpiło krzyżowanie. Rozwiązania słabe zanikają (wraz z informacją o miejscach krzyżowania), podczas gdy rozwiązania dobre (i informacja o korzystnych punktach krzyżowania) utrzymują się i ulepszają. Tak więc miejsca cięcia podlegają ewolucji wraz z populacją rozwiązań. Rozszerzeniem tego pomysłu jest rejestrowanie dopasowania (oceny) różnych operatorów genetycznych i wybór operatora z uwzględnieniem jego dopasowania.
- Krzyżowanie z wieloma przodkami: rekombinacji podlegają geny losowane z puli genów rodziców uzyskanych w wyniku selekcji, włączając w to przypadek tzw. „orgii", w których rozwiązanie-potomek może mieć więcej niż dwóch rodziców [EKK95; TYH99; Ahm15; AA19] (zastanów się, jaki wpływ ma liczba rodziców?)

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia

Dyskusja: czy mutacja jest niezbędna w algorytmie ewolucyjnym?

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia

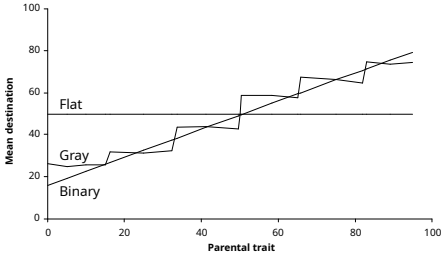
Dyskusja: czy mutacja jest niezbędna w algorytmie ewolucyjnym?

Z reguły korzystne jest to, że po mutacji osobnik jest podobny w pewnym stopniu do rodzica, oraz że każda mutacja powoduje zmiany zbliżonej wielkości. Czy taka sytuacja ma miejsce, kiedy zastosujemy standardowe kodowanie binarne liczb? Nie. Mutacje będą się zdarzać raz na bitach mniej, raz bardziej znaczących, powodując czasem niewielkie zmiany, czasem ogromne.

Czy kojarzysz jakiś specjalny kod, który może tu pomóc? Kod związany z sąsiednimi sekwencjami różniącymi się wartością tylko jednego bitu. Czy taki kod by tu pomógł?

Mutacja – sprytne kodowanie?

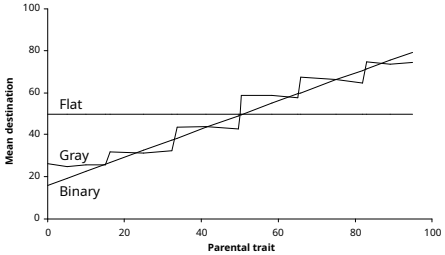
Jeśli zastosujemy kod Gray'a, w którym sąsiednie liczby różnią się tylko jednym bitem (odległość Hamminga), to zawsze sąsiednie liczby są osiągalne przez pojedynczą mutację. Ale czy to naprawdę pomoże?



Rysunek: Silne nieciągłości operatora mutacji mają stały wpływ (*bias*) na ewolucję [Bul99, str. 69]. Flat: mutacja polega na wylosowaniu wartości genu z dozwolonego przedziału. Kod Gray'a nie pozwala na uniknięcie wad związanych z mutowaniem zakodowanych wartości liczbowych.

Mutacja – sprytne kodowanie?

Jeśli zastosujemy kod Gray'a, w którym sąsiednie liczby różnią się tylko jednym bitem (odległość Hamminga), to zawsze sąsiednie liczby są osiągalne przez pojedynczą mutację. Ale czy to naprawdę pomoże?



Rysunek: Silne nieciągłości operatora mutacji mają stały wpływ (*bias*) na ewolucję [Bul99, str. 69]. Flat: mutacja polega na wylosowaniu wartości genu z dozwolonego przedziału. Kod Gray'a nie pozwala na uniknięcie wad związanych z mutowaniem zakodowanych wartości liczbowych.

Dyskusja: jakie kodowanie spełniłoby oczekiwanie, że każda mutacja zmienia zakodowaną wartość o ± 1?

Parametryzacja

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia

Wprowadzanie różnych udoskonaleń i rozszerzeń AG (przykład: adaptacyjne prawdopodobieństwa mutacji) prowadzi niekiedy do powstania kolejnych parametrów. Z drugiej strony część takich modyfikacji pozwala na automatyczne dostrojenie wartości parametrów. Nowe parametry są często łatwiej interpretowalne i/lub ich wpływ na działanie algorytmu i uzyskiwane wyniki staje się bardziej przewidywalny.

Dyskusja: wyobraź sobie, że masz jakiś konkretny problem optymalizacji. Jakie wartości parametrów ustalisz i jakie mechanizmy wybierzesz? Jakimi kryteriami kierujesz się wybierając akurat te, a nie inne wartości?

Parametryzacja – wielkość populacji i mutacja

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia

Wielkość populacji ma wpływ przede wszystkim na bezwładność algorytmu. Przy większych populacjach algorytm reaguje wolniej, dlatego przy zastosowaniach *on-line* zaleca się względnie małe populacje. Duże populacje, szczególnie na początku ewolucji, potrzebują więcej czasu by dotrzeć do rozwiązań dobrych. Z kolei dla zastosowań *off-line* duże populacje są korzystne, ponieważ dysponują większą ilością informacji i pozwalają na dokładniejsze przeszukanie przestrzeni rozwiązań. Duża liczba osobników zmniejsza zarazem ryzyko utknięcia w optimum lokalnym.

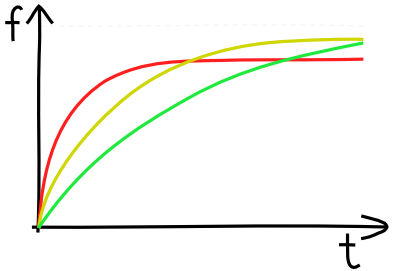
W AG prawdopodobieństwo mutacji odpowiada za liczbę losowych zmian bitów osobników. Zaleca się, by było ono odwrotnością liczby zmiennych decyzyjnych (jeśli zmienne decyzyjne są binarne, to mutacji ilu genów spodziewamy się w każdym pokoleniu?). Czasem spotyka się prawdopodobieństwo mutacji będące odwrotnością liczby osobników w populacji (mutacji ilu genów spodziewamy się w każdym pokoleniu?)

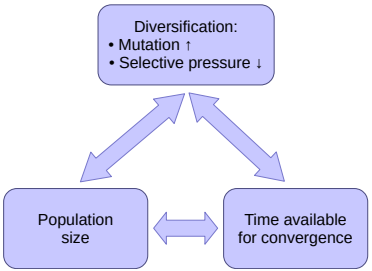
Parametryzacja – selekcja, warunek stopu i interakcje

Mechanizm selekcji o właściwościach korzystniejszych od mechanizmu ruletki to opisany wcześniej wybór losowy według reszt, bez powtórzeń lub z powtórzeniami, albo metody turniejowe. Polecany mechanizm skalowania to odcięcie na poziomie zależnym od odchylenia standardowego, ewentualnie połączone ze skalowaniem liniowym.

Dobrym kryterium stopu przy zastosowaniach *off-line* jest liczba pokoleń bez poprawy (chociaż bez wprowadzenia progu minimalnej zmiany algorytm może działać zbyt długo z powodu ciągłych, nieznacząco małych polepszeń). Można też monitorować średnie podobieństwo w populacji i kończyć optymalizację, gdy zaniknie różnorodność.

Kluczowy jest taki dobór parametrów, by uzyskać zbieżność w dostępnym czasie (por. pojęcie „przedwczesnej zbieżności" oraz *multi-random-start local search*).





Bibliografia I

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia

[AA19] Anas Arram i Masri Ayob. "A novel multi-parent order crossover in genetic algorithm for combinatorial optimization problems". W: *Computers & Industrial Engineering* 133 (2019), s. 267–274. DOI: 10.1016/j.cie.2019.05.012.

[Ahm15] Zakir Hussain Ahmed. "A multi-parent genetic algorithm for the quadratic assignment problem". W: *Opsearch* 52 (2015), s. 714–732. URL: https://www.researchgate.net/profile/Zakir-Ahmed-3/publication/275716879_A_multi-parent_genetic_algorithm_for_the_quadratic_assignment_problem/links/5bee70e54585150b2bba1f07/A-multi-parent-genetic-algorithm-for-the-quadratic-assignment-problem.pdf.

[Bul99] Seth Bullock. "Are artificial mutation biases unnatural?" W: *European Conference on Artificial Life*. Springer. 1999, s. 64–73. DOI: 10.1007/3-540-48304-7_11. URL: https://eprints.soton.ac.uk/261452/1/10.1.1.40.2753.pdf.

[DJ75] Kenneth Alan De Jong. "Analysis of the behavior of a class of genetic adaptive systems". Rozprawa doktorska. University of Michigan, 1975. URL: https://deepblue.lib.umich.edu/bitstream/handle/2027.42/4507/bab6360.0001.001.pdf.

[DS90] Gunter Dueck i Tobias Scheuer. "Threshold accepting: A general purpose optimization algorithm appearing superior to simulated annealing". W: *Journal of Computational Physics* 90.1 (1990), s. 161–175. DOI: https://doi.org/10.1016/0021-9991(90)90201-B. URL: https://www.sciencedirect.com/science/article/pii/002199919090201B.

[ECS89] Larry J. Eshelman, Rich Caruana i J. David Schaffer. "Biases in the crossover landscape". W: *Proceedings of the 3rd International Conference on Genetic Algorithms*. Oprac. J. David Schaffer. Morgan Kaufmann, 1989, s. 10–19.

Bibliografia II

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia

[EKK95] Agoston E. Eiben, Cees H. M. van Kemenade i Joost N. Kok. "Orgy in the computer: Multi-parent reproduction in genetic algorithms". W: *European conference on artificial life*. Springer. 1995, s. 934–945. URL: https://www.researchgate.net/profile/A-Eiben/publication/221530971_Orgy_in_the_Computer_Multi-Parent_Reproduction_in_Genetic_Algorithms/links/02bfe511b6708800d5000000/Orgy-in-the-Computer-Multi-Parent-Reproduction-in-Genetic-Algorithms.pdf.

[Gol03] David Edward Goldberg. *Algorytmy genetyczne i ich zastosowania*. Wydawnictwa Naukowo-Techniczne, 2003.

[KM18] Maciej Komosinski i Konrad Miazga. "Comparison of the tournament-based convection selection with the island model in evolutionary algorithms". W: *Journal of Computational Science* 32 (2018), s. 106–114. ISSN: 1877-7503. DOI: 10.1016/j.jocs.2018.10.001. URL: http://www.framsticks.com/files/common/ConvectionSelectionVsIslandModel.pdf.

[KU17] Maciej Komosinski i Szymon Ulatowski. "Multithreaded computing in evolutionary design and in artificial life simulations". W: *The Journal of Supercomputing* 73.5 (2017), s. 2214–2228. ISSN: 1573-0484. DOI: 10.1007/s11227-016-1923-4. URL: http://www.framsticks.com/files/common/MultithreadedEvolutionaryDesign.pdf.

[Mah92] Samir W. Mahfoud. "Crowding and preselection revisited". W: *Parallel problem solving from nature*. Oprac. R. Männer i B. Manderick. T. 2. Elsevier, 1992, s. 27–36. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.52.3943&rep=rep1&type=pdf.

Bibliografia III

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia

[TYH99] Shigeyoshi Tsutsui, Masayuki Yamamura i Takahide Higuchi. "Multi-parent recombination with simplex crossover in real coded genetic algorithms". W: *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation – Volume 1*. 1999, s. 657–664. URL: https://www.researchgate.net/profile/Shigeyoshi-Tsutsui/publication/243776468_Multi-parent_recombination_with_simplex_crossover_in_real-coded_genetic_algorithms/links/00463531fa92bd4738000000/Multi-parent-recombination-with-simplex-crossover-in-real-coded-genetic-algorithms.pdf.



Fundusze Europejskie
Polska Cyfrowa



Rzeczpospolita Polska

Unia Europejska
Europejski Fundusz Rozwoju Regionalnego

