

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

Wstęp

Algorytmy ewolucyjne

Selekcja

Ocena osobników

Krzyżowanie

Mutacja

Parametry

Bibliografia

W poprzednim semestrze, na przedmiocie *Inteligentne metody optymalizacji*, poznaliśmy różne algorytmy optymalizacji kombinatorycznej.

Teraz przyjrzymy się bardziej szczegółowo niektórym mechanizmom w tych algorytmach, poznamy ich warianty, przemyślimy wady i zalety, a na laboratoriach będziemy rozwiązywali zadanie optymalizacji trudniejsze w wielu aspektach od typowych zadań kombinatorycznych.

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

Wstęp

Algorytmy ewolucyjne

Selekcja

Ocena osobników

Krzyżowanie

Mutacja

Parametry

Bibliografia

- Czym różni się *random search* od *random walk*?
- 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*?

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 („biejącym”) 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

Wstęp

Algorytmy ewolucyjne

Selekcja

Ocena osobników

Krzyżowanie

Mutacja

Parametry

Bibliografia

- 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

Wstęp

Algorytmy ewolucyjne

Selekcja

Ocena osobników

Krzyżowanie

Mutacja

Parametry

Bibliografia

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

Wstęp

Algorytmy ewolucyjne

Selekcja

Ocena osobników

Krzyżowanie

Mutacja

Parametry

Bibliografia

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

Wstęp

Algorytmy ewolucyjne

Selekcja

Ocena osobników

Krzyżowanie

Mutacja

Parametry

Bibliografia

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

Rola niedeterminizmu w algorytmice

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

Wstęp

Algorytmy ewolucyjne

Selekcja

Ocena osobników

Krzyżowanie

Mutacja

Parametry

Bibliografia

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

Rozważ cztery generatorы sekwencji losowych: oparty o kolejne liczby i *modulo*, pseudolosowy [kiepski](#), pseudolosowy [dobry](#), i prawdziwie losowy.

Rola niedeterminizmu w algorytmice

Wstęp

Algorytmy
ewolucyjne

Selekcja

Ocena
osobników

Krzyżowanie

Mutacja

Parametry

Bibliografia

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

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

```
int random1(int n)      int random3(int n)
    static int c=-1        //very
    c++                   //complicated
    c=c%n                 //logic
    return c               return ...
```

```
int random2(int n)      int random4(int n)
    static int c=0          return ...%n
    c=(c+...)%n            return ...
```

Rola niedeterminizmu w algorytmice

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: oparte o kolejne liczby i *modulo*, pseudolosowy **kiepski**, pseudolosowy **dobry**, i prawdziwie losowy.

- przykład suplementacji (np. 4 substancje 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 – wnioski

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()?

Wstęp

Algorytmy ewolucyjne

Selekcja

Ocena osobników

Krzyżowanie

Mutacja

Parametry

Bibliografia

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, ...

Wstęp

Algorytmy ewolucyjne

Selekcja

Ocena osobników

Krzyżowanie

Mutacja

Parametry

Bibliografia

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

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?
- w jakich sytuacjach „nieskrępowana”, pełna losowość jest korzystna?

Wstęp

Algorytmy ewolucyjne

Selekcja

Ocena osobników

Krzyżowanie

Mutacja

Parametry

Bibliografia

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?
- w jakich sytuacjach „nieskrępowana”, pełna losowość jest korzystna?
- dlaczego przejmujemy się ...1080~~777777~~96980348..., ale nie ...1080~~735172~~96980348... ?

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?
- w jakich sytuacjach „nieskrępowana”, pełna losowość jest korzystna?
- dlaczego przejmujemy się ...1080~~777777~~96980348..., ale nie ...1080~~735172~~96980348... ?

Rola niedeterminizmu w algorytmice – wnioski

Wstęp

Algorytmy ewolucyjne

Selekcja

Ocena osobników

Krzyżowanie

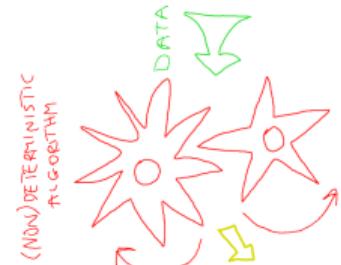
Mutacja

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?
- w jakich sytuacjach „nieskrępowana”, pełna losowość jest korzystna?
- dlaczego **przejmujemy się** ...1080~~777777~~96980348..., ale nie ...1080~~735172~~96980348... ? 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

Wstęp

Algorytmy ewolucyjne

Selekcja

Ocena osobników

Krzyżowanie

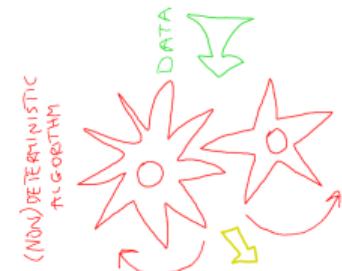
Mutacja

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?
- w jakich sytuacjach „nieskrępowana”, pełna losowość jest korzystna?
- dlaczego **przejmujemy się** ...1080~~777777~~96980348..., ale nie ...1080~~735172~~96980348... ? 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??



Selekcja – popularne metody

Wstęp

Algorytmy ewolucyjne

Selekcja

Ocena osobników

Krzyżowanie

Mutacja

Parametry

Bibliografia

f_i – ocena osobnika i -tego ($i = 1..POPSIZE$)

e_i – liczba jego oczekiwanych kopii w nowej (kolejnej) populacji,

$$e_i = POPSIZE \cdot f_i / \sum f_j$$

- Wybór losowy proporcjonalnie do jakości z powtórzeniami (tj. ze zwracaniem) – „zasada 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 właściwości 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, $f=[1,3,5,6]$.
- 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.

*https://en.wikipedia.org/wiki/Stochastic_universal_sampling

Selekcja – inne techniki

Wstęp

Algorytmy ewolucyjne

Selekcja

Ocena osobników

Krzyżowanie

Mutacja

Parametry

Bibliografia

- 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 powtóreniami: 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 nieliniiowe, 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.

Selekcja – dodatkowe własności

Wstęp

Algorytmy ewolucyjne

Selekcja

Ocena osobników

Krzyżowanie

Mutacja

Parametry

Bibliografia

- Model elitarny (elitarystyczny): spełnia oczekiwanie, że proces selekcji nie powinien prowadzić do utraty najlepszego znalezionej 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].

Meta-schematy selekcji

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

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. 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 *Equal/Number* i *Equal/Width* [KM18, Rys. 3], kiedy w podpopulacjach selekcja będzie losowa (np. rozmiar turnieju = 1), w porównaniu do modelu wyspowego i do standardowego, jednopolulacyjnego AE z losową selekcją?

Selekcja negatywna

Wstęp

Algorytmy ewolucyjne

Selekcja

Ocena osobników

Krzyżowanie

Mutacja

Parametry

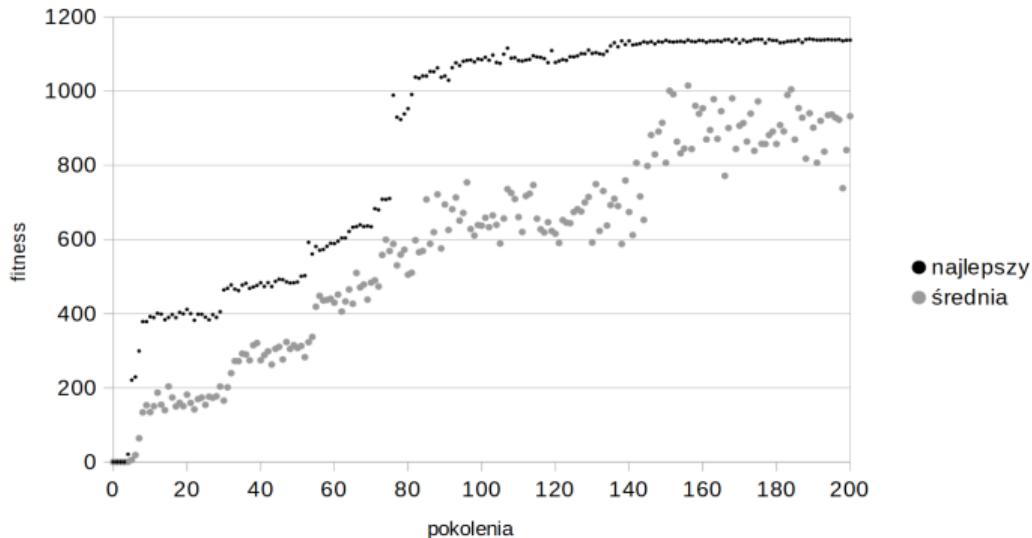
Bibliografia

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:



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?

Wstęp

Algorytmy ewolucyjne

Selekcja

Ocena osobników

Krzyżowanie

Mutacja

Parametry

Bibliografia

Odpowiedź na pytanie 1: przyszłość jest nieznana!

Wstęp

Algorytmy ewolucyjne

Selekcja

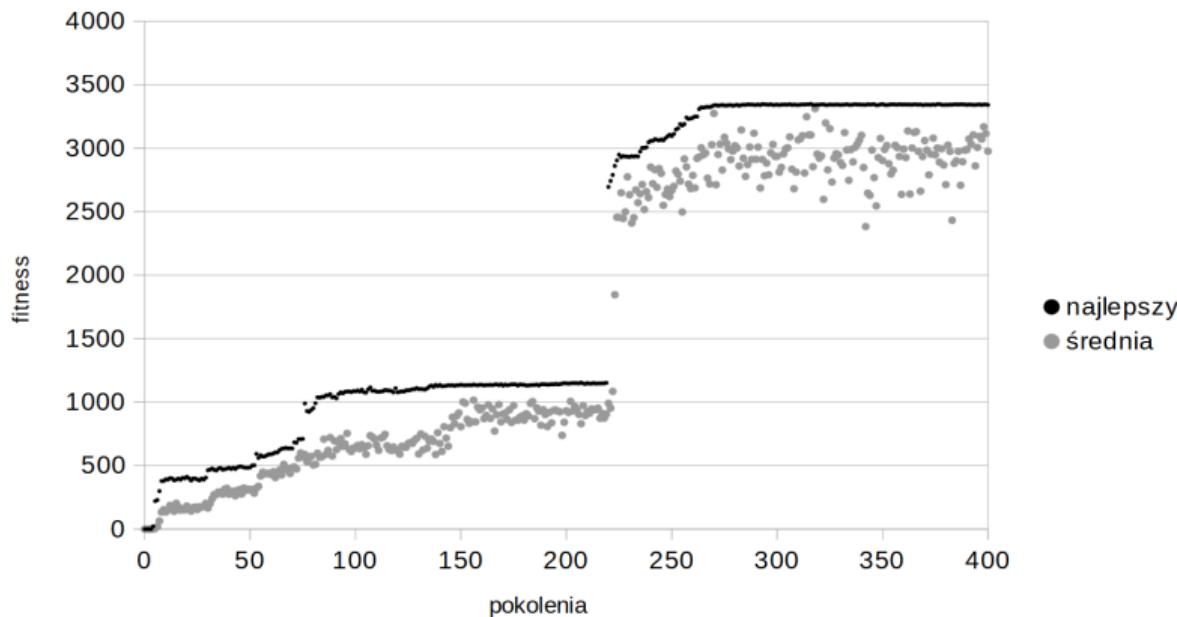
Ocena osobników

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

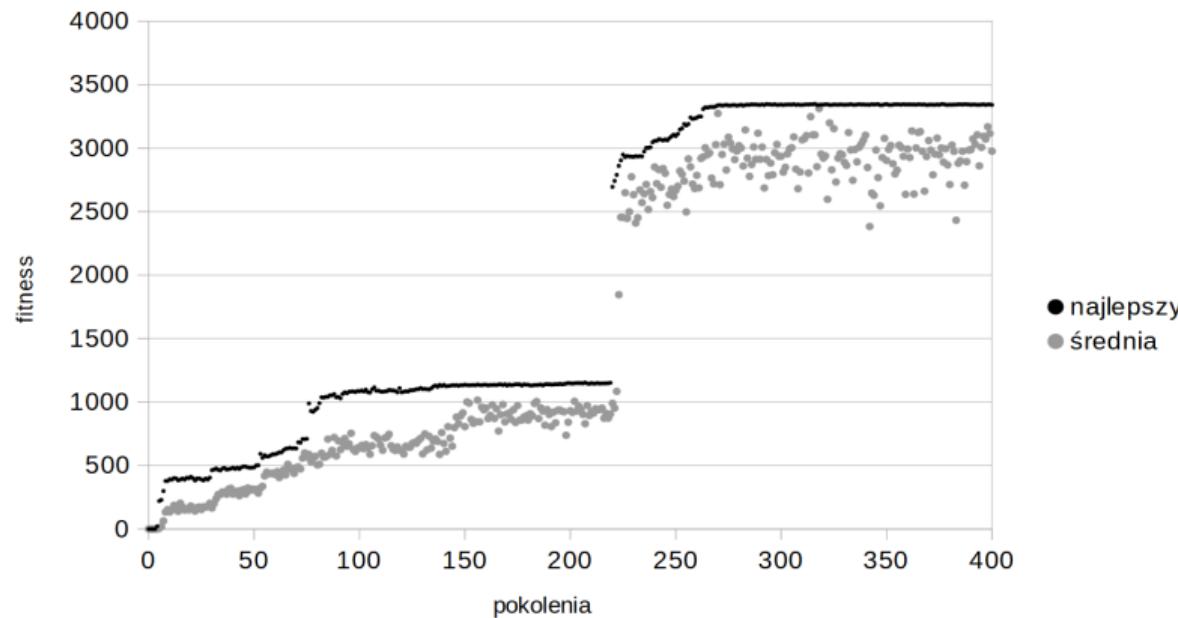
Ocena osobników

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

Wstęp

Algorytmy ewolucyjne

Selekcja

Ocena osobników

Krzyżowanie

Mutacja

Parametry

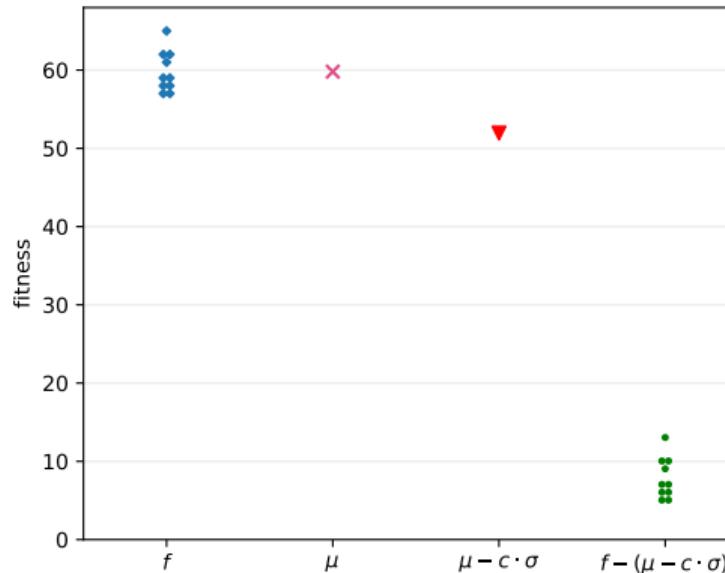
Bibliografia

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

Np. 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 $c = 3$.

Wstęp

Algorytmy ewolucyjne

Selekcja

Ocena osobników

Krzyżowanie

Mutacja

Parametry

Bibliografia

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 ($j = 1..n$).

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

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

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

Zadanie obliczeniowe (i interpretacyjne) – odpowiedź

Wstęp

Algorytmy ewolucyjne

Selekcja

Ocena osobników

Krzyżowanie

Mutacja

Parametry

Bibliografia

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

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.

Zadanie obliczeniowe (i interpretacyjne) – odpowiedź

Wstęp

Algorytmy ewolucyjne

Selekcja

Ocena osobników

Krzyżowanie

Mutacja

Parametry

Bibliografia

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

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.

Przypomnienie: jaki brzmiał jeden z pomysłów pozwalający reprezentować „presión selekcyjną” jako pojedynczą wartość?

Sens skalowania ocen

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...

Sens skalowania ocen

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?

Wstęp

Algorytmy ewolucyjne

Selekcja

Ocena osobników

Krzyżowanie

Mutacja

Parametry

Bibliografia

Średnia ocena jako „punkt neutralny” populacji

Czy średnia jest najlepszym pomysłem?

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

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

f_i – wartość funkcji celu osobnika,

M – **medianą** 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), więc zniknie 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.

Wstęp

Algorytmy ewolucyjne

Selekcja

Ocena osobników

Krzyżowanie

Mutacja

Parametry

Bibliografia

Przykładowy rozkład ocen w populacji podczas ewolucji

Wstęp

Algorytmy
ewolucyjne

Selekcja

Ocena
osobników

Krzyżowanie

Mutacja

Parametry

Bibliografia

Zobacz [KU17, górnny 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

Dyskusja: czy krzyżowanie jest niezbędne w algorytmie ewolucyjnym?

Przypomnienie: jednopunktowe, wielopunktowe, jednorodne, . . .

Krzyżowanie

Wstęp

Algorytmy ewolucyjne

Selekcja

Ocena osobników

Krzyżowanie

Mutacja

Parametry

Bibliografia

Dyskusja: czy krzyżowanie jest niezbędne w algorytmie ewolucyjnym?

Przypomnienie: jednopunktowe, wielopunktowe, jednorodne, . . .

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

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 (zastanów się, jaki wpływ ma liczba rodziców?)

Mutacja

Wstęp

Algorytmy
ewolucyjne

Selekcja

Ocena
osobników

Krzyżowanie

Mutacja

Parametry

Bibliografia

Dyskusja: czy mutacja jest niezbędna w algorytmie ewolucyjnym?

Mutacja

Wstęp

Algorytmy ewolucyjne

Selekcja

Ocena osobników

Krzyżowanie

Mutacja

Parametry

Bibliografia

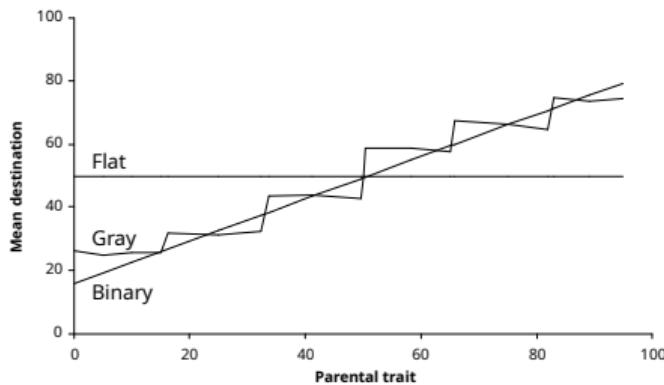
Dyskusja: czy mutacja jest niezbędna w algorytmie ewolucyjnym?

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.

Wstęp

Algorytmy ewolucyjne

Selekcja

Ocena osobników

Krzyżowanie

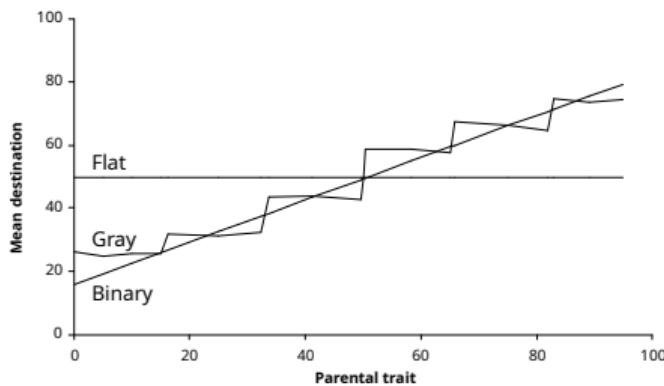
Mutacja

Parametry

Bibliografia

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 ?

Wstęp

Algorytmy ewolucyjne

Selekcja

Ocena osobników

Krzyżowanie

Mutacja

Parametry

Bibliografia

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. Trzeba jednak tak dobrąć wielkość populacji, żeby zdążyła ona zbiec do regionu dobrych rozwiązań w dostępnym czasie.

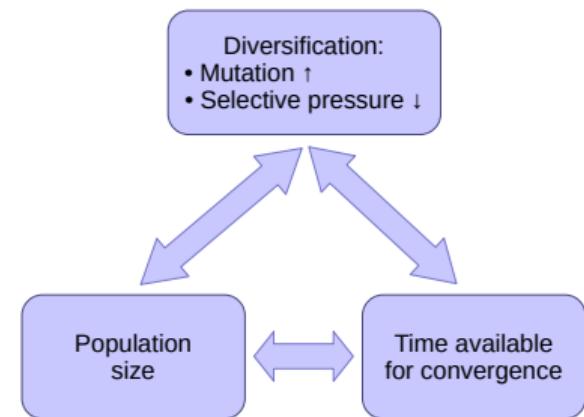
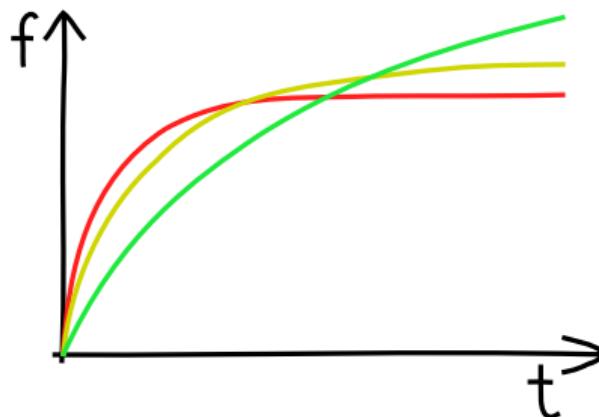
Prawdopodobieństwo mutacji odpowiada za liczbę losowych zmian bitów osobników. Zaleca się, by było ono odwrotnością liczby zmiennych decyzyjnych (jeśli zmienne decyzyjne są binarne, to mutacji ilu genów spodziewamy się w każdym pokoleniu?). Czasem spotyka się prawdopodobieństwo mutacji będące odwrotnością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

Parametryzacja – selekcja, warunek stopu i interakcje

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia

Mechanizm selekcji o właściwościach korzystniejszych od mechanizmu ruletki to opisany wcześniej wybór losowy według reszt, bez powtórzeń lub z powtórzeniami, albo metody turniejowe. Polecanym mechanizmem 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 (choćż 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 zniknie różnorodność.



Bibliografia |

Wstęp
Algorytmy ewolucyjne
Selekcja
Ocena osobników
Krzyżowanie
Mutacja
Parametry
Bibliografia

- [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](https://doi.org/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". Prac. dokt. University of Michigan, 1975. URL: <https://deepblue.lib.umich.edu/bitstream/handle/2027.42/4507/bab6360.0001.001.pdf>.
- [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*. Red. J. David Schaffer. Morgan Kaufmann, 1989, s. 10–19.
- [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](https://doi.org/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](https://doi.org/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*. Red. 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>.



**Fundusze
Europejskie**
Polska Cyfrowa



**Rzeczpospolita
Polska**

Unia Europejska
Europejski Fundusz
Rozwoju Regionalnego



Algorytmy i modele inspirowane biologicznie

2. Teoria algorytmów ewolucyjnych

Maciej Komosiński



**Fundusze
Europejskie**
Polska Cyfrowa



**Rzeczpospolita
Polska**

Unia Europejska
Europejski Fundusz
Rozwoju Regionalnego



Przypomnienie z poprzedniego semestru (i ew. uzupełnienie)

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

- O czym mówi *Twierdzenie o schematach* Hollanda?
- O czym mówi *Hipoteza cegiełek*?
- O czym mówi *Twierdzenie „No Free Lunch”*?

Budowa problemów zawodnych (zwodniczych)

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

Aby skonstruować taki problem, który sprawi trudność algorytmowi genetycznemu, zanegujmy hipotezę cegiełek: dobre „cegiełki” po połączeniu mają tworzyć niekorzystną strukturę.

Najprostszy taki przypadek można uzyskać dla genotypów o długości 2 (jest to problem zwodniczy rzędu drugiego).

Budujemy problem zwodniczy rzędu drugiego

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

Założymy, że istnieją cztery schematy:

*** 0 *** 0 ***
*** 0 *** 1 ***
*** 1 *** 0 ***
*** 1 *** 1 ***

Gwiazdki odpowiadają dowolnej liczbie symboli nieustalonych (ale pozycje ustalone są we wszystkich schematach na tych samych miejscach). Oznaczmy średnie oceny schematów przez f_{00} , f_{01} , f_{10} , f_{11} i schemat z dwoma jedynkami jest globalnym optimum (f_{11}). Aby problem był zwodniczy, chcemy, by schematy z jednym ustaloną zerem były lepsze od odpowiednich schematów z jedynką – to znaczy, by spełniona była przynajmniej jedna z nierówności:

$$f_{0*} > f_{1*}$$

$$f_{0*} = (f_{00} + f_{01})/2, \text{ itd.}$$

$$f_{*0} > f_{*1}$$

Krajobraz przystosowania minimalnego problemu zwodniczego

Wstęp

Problemy zawodne

Epistaza

Ocena AE

Neutralność

Dryf genetyczny

Brak mutacji

Zadanie

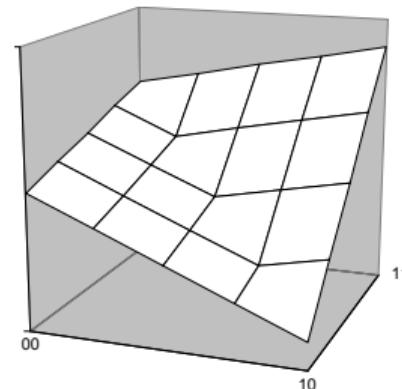
Bibliografia

Taki problem nazywany jest minimalnym problemem zwodniczym (ang. *minimal deceptive problem, MDP*), ponieważ nie istnieje problem zwodniczy rzędu pierwszego.

Są dwa typy problemów zwodniczych rzędu drugiego (wykres obok pokazuje przykładowy problem zwodniczy rzędu drugiego, typu I):

Typ I: $f_{01} > f_{00}$

Typ II: $f_{00} \geq f_{01}$



W *MDP* funkcja przystosowania *nie może* być przedstawiona jako liniowa kombinacja poszczególnych allelei, czyli w postaci $f(x_1, x_2) = b + \sum_{i=1}^2 a_i x_i$.

Trudność problemów zwodniczych w praktyce

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

To, że problem jest zwodniczy nie oznacza jeszcze, iż algorytm genetyczny nie znajdzie optimum. Oznacza to jednak, że funkcja przystosowania (przedstawiona na osi pionowej) nie może być wyrażona jako liniowa kombinacja poszczególnych bitów, zatem występuje zjawisko epistazy – nieliniowości. Zwykle taki problem nie okazuje się jednak AG-trudny, czyli algorytm genetyczny może znaleźć rozwiązanie optymalne.

Zachowanie algorytmu zależy jednak od wielu czynników, takich jak na przykład początkowa obecność schematów w populacji osobników. W szczególności, jeśli wszystkie cztery schematy występują w populacji początkowej, problem zwodniczy rzędu 2 typu I nie jest AG-trudny. Dla zadań typu II zachowanie algorytmu zależy od proporcji występowania schematów w populacji: jeśli schemat 00 będzie występował w przewadze, algorytm może być zbieżny do rozwiązania nieoptymalnego (choć taka sytuacja występuje rzadko). Bardziej szczegółowa analiza problemów zwodniczych znajduje się w [Gol03, str. 63–70, 375–382].

Epistaza

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

Cecha reprezentacji (i ew. operatorów dla niej) – stopień zależności pomiędzy różnymi genami w chromosomie [Dav90]. Jeśli dana reprezentacja posiada wysoką epistazę, fenotypowy efekt pewnych genów zależy od alleli (wartości) innych genów (poligeniczność).

Zerowa epistaza: każdy gen niezależnie wpływa na wartość funkcji celu (i wtedy nie warto stosować AE). Dla skuteczności AE tym lepiej, im mniejsza epistaza. Dla niektórych definicji funkcji celu, projektując reprezentację i operatory może opłacić się zaakceptować niewielki wzrost epistazy, jeśli zyskamy korzystniejszy związek między topologią przestrzeni przeszukiwania, a krajobrazem przystosowania.

Oszacuj mniej więcej epistazę dla następujących funkcji celu $f(x_1, x_2)$, gdzie x to liczby – wartości genów: $x_1 + x_2$, $x_1 - x_2$, $x_1 \cdot x_2$, $x_1 + x_2 \cdot x_2$, $x_1 + x_1 \cdot x_2$, $\frac{x_1}{x_2}$, $x_2 + \frac{x_1}{x_2}$.

Epistaza – przykład

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

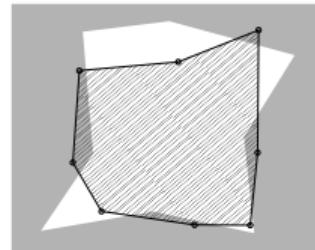
Brak mutacji

Zadanie

Bibliografia

Optymalizujemy kształt figury 2D (albo przekrój obiektu 3D); geny opisują rozmieszczenie wierzchołków w przestrzeni (współrzędne x_i, y_i).

Porównaj epistazę dla następujących scenariuszy:



- ① Same geny x_i, y_i oraz operator mutacji przesuwający losowy wierzchołek.
- ② Wprowadzamy dwa dodatkowe geny: obrotu i skali. Mutacja tych pojedynczych genów zmienia orientację (obrót) i rozmiar całej figury.
- ③ Zamiast tych dwóch dodatkowych genów wprowadzamy specjalne operatory mutacji-obrotu i mutacji-skalowania.

Epistaza – przykład

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

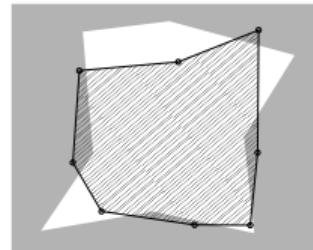
Brak mutacji

Zadanie

Bibliografia

Optymalizujemy kształt figury 2D (albo przekrój obiektu 3D); geny opisują rozmieszczenie wierzchołków w przestrzeni (współrzędne x_i, y_i).

Porównaj epistazę dla następujących scenariuszy:



- ① Same geny x_i, y_i oraz operator mutacji przesuwający losowy wierzchołek.
- ② Wprowadzamy dwa dodatkowe geny: obrotu i skali. Mutacja tych pojedynczych genów zmienia orientację (obrót) i rozmiar całej figury.
- ③ Zamiast tych dwóch dodatkowych genów wprowadzamy specjalne operatory mutacji-obrotu i mutacji-skalowania.

W scenariuszu (1) zmiana orientacji i rozmiaru figury jest możliwa jedynie przez bardzo wiele niezależnych mutacji współrzędnych wierzchołków. Tymczasem w pewnych zadaniach (np. w optymalizacji online, gdy docelowy kształt otworu się zmienia), możliwość szybkiego obrotu i skalowania może przyspieszyć zbieżność i polepszyć jakość uzyskiwanych rozwiązań. W scenariuszu (2) rozważ wpływ konkretnej implementacji (matematyczne działania przekształcające genotyp \rightarrow fenotyp) – chodzi o efekty złożenia mutacji (1) i (2).

Epistaza – opis formalny

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

Epistaza jako stopień interakcji pomiędzy genami [RW95; NK00]:

$$f(s) = \text{stała}$$

$$+ \sum_{i=0}^{l-1} \text{efekt } s_i$$

$$+ \sum_{i=0}^{l-2} \sum_{j=i+1}^{l-1} \text{interakcja między } s_i \text{ i } s_j$$

+ ...

$$+ \text{interakcja między } s_0, s_1, \dots, s_{l-1}$$

+ losowy błąd.

s_i – gen i -ty (lub **allel** genu i -tego, jeśli s jest konkretnym rozwiązaniem),

l – liczba genów (numerowane od zera).

Interakcje mogą być dodatnie i ujemne: jeśli np. interakcja pomiędzy s_i i s_j ma taki sam znak, co efekty s_i i s_j , to taka interakcja wzmacnia działanie tych dwóch genów i może być pożądana.

Epistaza i mutacje

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

Epistaza genów ma swoje konsekwencje, kiedy są one mutowane: przemyśl [ten wykres](#).

Empiryczna i teoretyczna ocena AE

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

Badanie efektywności i zachowania AE można prowadzić teoretycznie albo eksperymentalnie ([empirycznie](#)).

Analiza teoretyczna daje ugruntowaną, pewną wiedzę, ale często da się tak badać tylko bardzo proste modele (tj. z wieloma założeniami).

Analiza empiryczna AE prowadzi do mniej pewnej wiedzy, trudniej o uogólnienia, ale zawsze da się taką analizę przeprowadzić. Oprócz testowania skuteczności algorytmu na docelowym problemie optymalizacji, często używa się znanych problemów testowych* albo sparametryzowanych modeli problemów pozwalających na sterowanie nasileniem epistazy – takich jak model NK** oraz jego warianty z neutralnością: NKP (probabilistyczny) i NKQ (z kwantyzacją).

*https://en.wikipedia.org/wiki/Test_functions_for_optimization

**https://en.wikipedia.org/wiki/NK_model

Konsekwencje neutralności w krajobrazach przystosowania [Gea+02]

- potocznie/tradycyjnie uważa się, że krajobraz przystosowania „składa się z pagórków”, a mutacje powodują zmianę przystosowania

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

- potocznie/tradycyjnie uważa się, że krajobraz przystosowania „składa się z pagórków”, a mutacje powodują zmianę przystosowania
- prowadzi to do koncepcji „optimów lokalnych”, w których rozwiązania lub ich populacje mogą utknąć

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

- potocznie/tradycyjnie uważa się, że krajobraz przystosowania „składa się z pagórków”, a mutacje powodują zmianę przystosowania
- prowadzi to do koncepcji „optimów lokalnych”, w których rozwiązania lub ich populacje mogą utknąć
- badania molekularne nad krajobrazami różnych sposobów zwijania RNA sugerują, że duża część mutacji na poziomie molekularnym jest selekcyjnie neutralna

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

- potocznie/tradycyjnie uważa się, że krajobraz przystosowania „składa się z pagórków”, a mutacje powodują zmianę przystosowania
- prowadzi to do koncepcji „optimów lokalnych”, w których rozwiązania lub ich populacje mogą utknąć
- badania molekularne nad krajobrazami różnych sposobów zwijania RNA sugerują, że duża część mutacji na poziomie molekularnym jest selekcyjnie neutralna
- neutralność jest również obecna w wielu rzeczywistych problemach optymalizacyjnych (popularne „płaskowyże” lub sąsiedzi tej samej jakości)

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

- potocznie/tradycyjnie uważa się, że krajobraz przystosowania „składa się z pagórków”, a mutacje powodują zmianę przystosowania
- prowadzi to do koncepcji „optimów lokalnych”, w których rozwiązania lub ich populacje mogą utknąć
- badania molekularne nad krajobrazami różnych sposobów zwijania RNA sugerują, że duża część mutacji na poziomie molekularnym jest selekcyjnie neutralna
- neutralność jest również obecna w wielu rzeczywistych problemach optymalizacyjnych (popularne „płaskowyże” lub sąsiedzi tej samej jakości)
- zatem: wiele genotypów [\rightarrow jeden fenotyp] \rightarrow jedna wartość funkcji celu

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

- potocznie/tradycyjnie uważa się, że krajobraz przystosowania „składa się z pagórków”, a mutacje powodują zmianę przystosowania
- prowadzi to do koncepcji „optimów lokalnych”, w których rozwiązania lub ich populacje mogą utknąć
- badania molekularne nad krajobrazami różnych sposobów zwijania RNA sugerują, że duża część mutacji na poziomie molekularnym jest selekcyjnie neutralna
- neutralność jest również obecna w wielu rzeczywistych problemach optymalizacyjnych (popularne „płaskowyże” lub sąsiedzi tej samej jakości)
- zatem: wiele genotypów [\rightarrow jeden fenotyp] \rightarrow jedna wartość funkcji celu
- neutralność: jedna z przyczyn równowag przestankowych

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

- potocznie/tradycyjnie uważa się, że krajobraz przystosowania „składa się z pagórków”, a mutacje powodują zmianę przystosowania
- prowadzi to do koncepcji „optimów lokalnych”, w których rozwiązania lub ich populacje mogą utknąć
- badania molekularne nad krajobrazami różnych sposobów zwijania RNA sugerują, że duża część mutacji na poziomie molekularnym jest selekcyjnie neutralna
- neutralność jest również obecna w wielu rzeczywistych problemach optymalizacyjnych (popularne „płaskowyże” lub sąsiedzi tej samej jakości)
- zatem: wiele genotypów [\rightarrow jeden fenotyp] \rightarrow jedna wartość funkcji celu
- neutralność: jedna z przyczyn równowag przestankowych
- jeśli neutralność występuje często w krajobrazie, ryzyko uwięzienia populacji AE w lokalnych optimach jest niskie

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

- potocznie/tradycyjnie uważa się, że krajobraz przystosowania „składa się z pagórków”, a mutacje powodują zmianę przystosowania
- prowadzi to do koncepcji „optimów lokalnych”, w których rozwiązania lub ich populacje mogą utknąć
- badania molekularne nad krajobrazami różnych sposobów zwijania RNA sugerują, że duża część mutacji na poziomie molekularnym jest selekcyjnie neutralna
- neutralność jest również obecna w wielu rzeczywistych problemach optymalizacyjnych (popularne „płaskowyże” lub sąsiedzi tej samej jakości)
- zatem: wiele genotypów [\rightarrow jeden fenotyp] \rightarrow jedna wartość funkcji celu
- neutralność: jedna z przyczyn równowag przestankowych
- jeśli neutralność występuje często w krajobrazie, ryzyko uwięzienia populacji AE w lokalnych optimach jest niskie
- w związku z tym rośnie rolą charakterystyk operatorów rekonfiguracji (mutacji/sąsiedztwa, krzyżowania) i rolą dryfu genetycznego

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

- potocznie/tradycyjnie uważa się, że krajobraz przystosowania „składa się z pagórków”, a mutacje powodują zmianę przystosowania
- prowadzi to do koncepcji „optimów lokalnych”, w których rozwiązania lub ich populacje mogą utknąć
- badania molekularne nad krajobrazami różnych sposobów zwijania RNA sugerują, że duża część mutacji na poziomie molekularnym jest selekcyjnie neutralna
- neutralność jest również obecna w wielu rzeczywistych problemach optymalizacyjnych (popularne „płaskowyże” lub sąsiedzi tej samej jakości)
- zatem: wiele genotypów [\rightarrow jeden fenotyp] \rightarrow jedna wartość funkcji celu
- neutralność: jedna z przyczyn równowag przestankowych
- jeśli neutralność występuje często w krajobrazie, ryzyko uwięzienia populacji AE w lokalnych optimach jest niskie
- w związku z tym rośnie rolą charakterystyk operatorów rekonfiguracji (mutacji/sąsiedztwa, krzyżowania) i rolą dryfu genetycznego
- kluczowa rola ostrej i nieostrej nierówności w implementacjach przeszukiwania lokalnego, o której mówiliśmy wcześniej (ostra nierówność \rightarrow wszystkie neutralne ruchy są potencjalnymi końcami)

Główne siły prowadzące ewolucję

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

- co kieruje ewolucją? co wpływa na trajektorię populacji?

Główne siły prowadzące ewolucję

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

- co kieruje ewolucją? co wpływa na trajektorię populacji?
- krajobraz przystosowania kontra operatory rekonfiguracyjne

Główne siły prowadzące ewolucję

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

- co kieruje ewolucją? co wpływa na trajektorię populacji?
- krajobraz przystosowania kontra operatory rekonfiguracyjne
- co się stanie, gdy wszystkie lub większość mutacji będzie niekorzystna, a dawne osobniki nie będą zachowywane?

Główne siły prowadzące ewolucję

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

- co kieruje ewolucją? co wpływa na trajektorię populacji?
- krajobraz przystosowania kontra operatory rekonfiguracyjne
- co się stanie, gdy wszystkie lub większość mutacji będzie niekorzystna, a dawne osobniki nie będą zachowywane?
- „bias” / „neutralność” krajobrazu przystosowania vs. bias/neutralność operatorów rekonfiguracyjnych

Główne siły prowadzące ewolucję

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

- co kieruje ewolucją? co wpływa na trajektorię populacji?
- krajobraz przystosowania kontra operatory rekonfiguracyjne
- co się stanie, gdy wszystkie lub większość mutacji będzie niekorzystna, a dawne osobniki nie będą zachowywane?
- „bias” / „neutralność” krajobrazu przystosowania vs. bias/neutralność operatorów rekonfiguracyjnych
- operatory rekonfiguracyjne: zmiany w częstości allelei, ale zwykle chcemy uniknąć „biasu” !

Główne siły prowadzące ewolucję

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

- co kieruje ewolucją? co wpływa na trajektorię populacji?
- krajobraz przystosowania kontra operatory rekonfiguracyjne
- co się stanie, gdy wszystkie lub większość mutacji będzie niekorzystna, a dawne osobniki nie będą zachowywane?
- „bias” / „neutralność” krajobrazu przystosowania vs. bias/neutralność operatorów rekonfiguracyjnych
- operatory rekonfiguracyjne: zmiany w częstości allelei, ale zwykle chcemy uniknąć „biasu” !
- widzisz populację osób o niebieskich, zielonych i brązowych oczach; po iluś pokoleniach wszyscy mają zielone oczy. Dlaczego?

Dryf genetyczny

- wyłączmy presję selekcyjną, mutację i krzyżowanie

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

Dryf genetyczny

- wyłączmy presję selekcyjną, mutację i krzyżowanie
- przykład kamików w słoiku

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

Dryf genetyczny

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

- wyłączmy presję selekcyjną, mutację i krzyżowanie
- przykład kamików w słoiku
- dyskretny charakter populacji (składających się z odrębnych osobników), więc idealnie równy rozkład alleli nie jest możliwy (np. 1/64 wśród 5000 osobników)

Dryf genetyczny

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

- wyłączmy presję selekcyjną, mutację i krzyżowanie
- [przykład kamików w słoiku](#)
- dyskretny charakter populacji (składających się z odrębnych osobników), więc idealnie równy rozkład alleli nie jest możliwy (np. 1/64 wśród 5000 osobników)
- wąskie gardła (populacja tymczasowo kurczy się do bardzo małych rozmiarów)

Dryf genetyczny

- wyłączmy presję selekcyjną, mutację i krzyżowanie
- [przykład kamików w słoiku](#)
- dyskretny charakter populacji (składających się z odrębnych osobników), więc idealnie równy rozkład alleli nie jest możliwy (np. 1/64 wśród 5000 osobników)
- wąskie gardła (populacja tymczasowo kurczy się do bardzo małych rozmiarów)
 - [efekt założyciela](#)

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

Dryf genetyczny

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

- wyłączmy presję selekcyjną, mutację i krzyżowanie
- przykład kamików w słoiku
- dyskretny charakter populacji (składających się z odrębnych osobników), więc idealnie równy rozkład alleli nie jest możliwy (np. 1/64 wśród 5000 osobników)
- wąskie gardła (populacja tymczasowo kurczy się do bardzo małych rozmiarów)
 - efekt założyciela
- wpływ rozkładu cech i wielkości populacji

Dryf genetyczny

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

- wyłączmy presję selekcyjną, mutację i krzyżowanie
- przykład kamików w słoiku
- dyskretny charakter populacji (składających się z odrębnych osobników), więc idealnie równy rozkład alleli nie jest możliwy (np. 1/64 wśród 5000 osobników)
- wąskie gardła (populacja tymczasowo kurczy się do bardzo małych rozmiarów)
 - efekt założyciela
- wpływ rozkładu cech i wielkości populacji
 - prawdopodobieństwo, że dana cecha ostatecznie zdominuje całą populację, to jej częstość w populacji

Dryf genetyczny

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

- wyłączmy presję selekcyjną, mutację i krzyżowanie
- [przykład kamików w słoiku](#)
- dyskretny charakter populacji (składających się z odrębnych osobników), więc idealnie równy rozkład alleli nie jest możliwy (np. 1/64 wśród 5000 osobników)
- wąskie gardła (populacja tymczasowo kurczy się do bardzo małych rozmiarów)
 - [efekt założyciela](#)
- wpływ rozkładu cech i wielkości populacji
 - prawdopodobieństwo, że dana cecha ostatecznie zdominuje całą populację, to jej częstość w populacji
 - oczekiwana liczba pokoleń do wystąpienia całkowitej dominacji jest [proporcjonalna do wielkości populacji](#)

Dryf genetyczny

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

- wyłączmy presję selekcyjną, mutację i krzyżowanie
- [przykład kamików w słoiku](#)
- dyskretny charakter populacji (składających się z odrębnych osobników), więc idealnie równy rozkład alleli nie jest możliwy (np. 1/64 wśród 5000 osobników)
- wąskie gardła (populacja tymczasowo kurczy się do bardzo małych rozmiarów)
 - [efekt założyciela](#)
- wpływ rozkładu cech i wielkości populacji
 - prawdopodobieństwo, że dana cecha ostatecznie zdominuje całą populację, to jej częstość w populacji
 - oczekiwana liczba pokoleń do wystąpienia całkowitej dominacji jest [proporcjonalna do wielkości populacji](#)
- wpływ losowości (nieograniczonej) vs. „sprawiedliwość” – przypomnienie: nasza wcześniejsza szczegółowa dyskusja

Dryf genetyczny

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

- wyłączmy presję selekcyjną, mutację i krzyżowanie
- [przykład kamików w słoiku](#)
- dyskretny charakter populacji (składających się z odrębnych osobników), więc idealnie równy rozkład alleli nie jest możliwy (np. 1/64 wśród 5000 osobników)
- wąskie gardła (populacja tymczasowo kurczy się do bardzo małych rozmiarów)
 - [efekt założyciela](#)
- wpływ rozkładu cech i wielkości populacji
 - prawdopodobieństwo, że dana cecha ostatecznie zdominuje całą populację, to jej częstość w populacji
 - oczekiwana liczba pokoleń do wystąpienia całkowitej dominacji jest [proporcjonalna do wielkości populacji](#)
- wpływ losowości (nieograniczonej) vs. „sprawiedliwość” – przypomnienie: nasza wcześniejsza szczegółowa dyskusja

Dryf genetyczny

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

- wyłączmy presję selekcyjną, mutację i krzyżowanie
- [przykład kamików w słoiku](#)
- dyskretny charakter populacji (składających się z odrębnych osobników), więc idealnie równy rozkład alleli nie jest możliwy (np. 1/64 wśród 5000 osobników)
- wąskie gardła (populacja tymczasowo kurczy się do bardzo małych rozmiarów)
 - [efekt założyciela](#)
- wpływ rozkładu cech i wielkości populacji
 - prawdopodobieństwo, że dana cecha ostatecznie zdominuje całą populację, to jej częstość w populacji
 - oczekiwana liczba pokoleń do wystąpienia całkowitej dominacji jest [proporcjonalna do wielkości populacji](#)
- wpływ losowości (nieograniczonej) vs. „sprawiedliwość” – przypomnienie: nasza wcześniejsza szczegółowa dyskusja

-
- jeśli nie znasz tych zjawisk, ich konsekwencje mogą być sprzeczne z intuicją, błędnie interpretowane i mylnie przypisywane innym mechanizmom!
 - interakcja krajobrazu przystosowania (selekcja), operatorów rekonfiguracji i dryfu genetycznego

Przykład analizy teoretycznej AG

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

Wyobraźmy sobie algorytm genetyczny (reprezentacja binarna) z selekcją ruletkową i krzyżowaniem, bez mutacji.

Co się stanie po długim czasie działania algorytmu?

Przykład analizy teoretycznej AG

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

Wyobraźmy sobie algorytm genetyczny (reprezentacja binarna) z selekcją ruletkową i krzyżowaniem, bez mutacji.

Co się stanie po długim czasie działania algorytmu?

Jak to udowodnić?

Przykład analizy teoretycznej AG

Stany i przejścia

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

Rozważamy populację m osobników o długości k . Ile jest różnych stanów populacji?

*https://en.wikipedia.org/wiki/Stochastic_matrix

**<https://www.youtube.com/watch?v=nnssRe5DewE>

***https://en.wikipedia.org/wiki/Absorbing_Markov_chain

Przykład analizy teoretycznej AG

Stany i przejścia

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

Rozważamy populację m osobników o długości k . Ile jest różnych stanów populacji? Ile jest stanów, które są atraktorami?

*https://en.wikipedia.org/wiki/Stochastic_matrix

**<https://www.youtube.com/watch?v=nnssRe5DewE>

***https://en.wikipedia.org/wiki/Absorbing_Markov_chain

Przykład analizy teoretycznej AG

Stany i przejścia

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

Rozważamy populację m osobników o długości k . Ile jest różnych stanów populacji? Ile jest stanów, które są atraktorami?

Użyjemy łańcuchów Markowa; niech π – stan, P – macierz prawdopodobieństw przejścia*. Jest 2^{mk} różnych stanów. Rozkład prawdopodobieństwa stanów, w których możemy wylądować po n krokach** od stanu π to π -ty wiersz macierzy P^n – zapiszmy go jako πP^n . Ponieważ nie ma mutacji, pewne stany są absorbujące (nie ma z nich wyjścia) – wszystkie osobniki są równe; takich stanów jest $a = 2^k$. Zatem macierz P możemy wyrazić jako***

$$P = \begin{bmatrix} I_a & 0 \\ R & Q \end{bmatrix}$$

gdzie I_a to macierz $a \times a$ wypełniona zerami oprócz jedynek po przekątnej, R to podmacierz $t \times a$ opisująca przejścia do stanu absorbującego, Q to podmacierz $t \times t$ opisująca przejścia do stanów nie-absorbujących; $t = 2^{mk} - a$.

*https://en.wikipedia.org/wiki/Stochastic_matrix

**<https://www.youtube.com/watch?v=nnssRe5DewE>

***https://en.wikipedia.org/wiki/Absorbing_Markov_chain

Przykład analizy teoretycznej AG

Agregacja prawdopodobieństw przejścia między stanami

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

Dla n kroków będziemy mieli

$$P^n = \begin{bmatrix} I_a & 0 \\ N_n R & Q^n \end{bmatrix}$$

gdzie $N_n = I_t + Q + Q^2 + Q^3 + \dots + Q^{n-1}$, I_t to macierz $t \times t$ wypełniona zerami oprócz jedynek po przekątnej.

Granicznie

$$\lim_{n \rightarrow \infty} P^n = \begin{bmatrix} I_a & 0 \\ (I_t - Q)^{-1} R & 0 \end{bmatrix}.$$

Przykład analizy teoretycznej AG

Prawdopodobieństwo absorpcji

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

A zatem, co można łatwo zobaczyć mnożąc przykładowe macierze w numpy czy nawet w arkuszu kalkulacyjnym, nasz algorytm zaczynając ze stanu nie-absorbowującego (prawdopodobieństwa $(I_t - Q)^{-1}R$) wyląduje na pewno w jakimś stanie absorbującym i w nim pozostanie (I_a). Obliczmy prawdopodobieństwo trafienia do stanu absorbującego [Fog00, str. 105]. Niech $\Gamma = \{0, 1\}$. Po n iteracjach, nasz algorytm znajdzie się w stanie γ , $\gamma \in (\Gamma^k)^m$:

$$\Pr(\gamma \in A) = \sum_{i=1}^a (\pi^* P^n)_i = \sum_{i=1}^a \left(\pi^* \begin{bmatrix} I_a \\ N_n R \end{bmatrix} \right)_i$$

gdzie $(\cdot)_i$ oznacza i -ty element poziomego wektora, A to zbiór wszystkich stanów absorbujących, a π^* to wektor poziomy opisujący prawdopodobieństwa rozpoczęcia algorytmu w każdym stanie populacji. Graniczne prawdopodobieństwo absorpcji

$$\lim_{n \rightarrow \infty} \sum_{i=1}^a \left(\pi^* \begin{bmatrix} I_a \\ N_n R \end{bmatrix} \right)_i = \sum_{i=1}^a \left(\pi^* \begin{bmatrix} I_a \\ (I - Q)^{-1}R \end{bmatrix} \right)_i = 1.$$

Zadanie dla chętnych

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

Stosujemy pokoleniowy algorytm ewolucyjny, populacja n osobników, g z n osobników jest dobrych, $n - g$ jest złych. Wykorzystujemy selekcję turniejową (wielkość turnieju k) ze zwracaniem, w której jeśli osobnik dobry spotka się z złym, wygrywa dobry.

Załącz, że krzyżowanie i mutacja nie zmieniają jakości osobnika (dobry pozostaje dobrym, zły pozostaje złym).

- ① Zakładając, że w populacji jest połowa osobników dobrych ($g = \frac{n}{2}$), jak zmieni się ta proporcja w kolejnym pokoleniu (tzn. po jednej selekcji)?
- ② Zakładając, że wszystkie osobniki podlegają mutacji i mutacja nigdy nie polepsza złego osobnika, ale jednak pogarsza osobnika dobrego z prawdopodobieństwem m , jakie musi być co najmniej g , żeby liczba dobrych w populacji nie spadała?

Bibliografia |

Wstęp

Problemy
zawodne

Epistaza

Ocena AE

Neutralność

Dryf
genetyczny

Brak mutacji

Zadanie

Bibliografia

- [Dav90] Yuval Davidor. "Epistasis variance: Suitability of a representation to genetic algorithms". W: *Complex Systems* 4.4 (1990), s. 369–383. URL: <http://www.complex-systems.com/pdf/04-4-1.pdf>.
- [Fog00] David B. Fogel. *Evolutionary computation: toward a new philosophy of machine intelligence*. 2 wyd. IEEE Press, 2000.
- [Gea+02] Nicholas Gead i in. "A comparison of neutral landscapes – NK, NK_p and NK_q". W: *Proceedings of the 2002 Congress on Evolutionary Computation, CEC'02*. T. 1. IEEE. 2002, s. 205–210. URL: <https://eprints.soton.ac.uk/264208/1/cec1-comp.pdf>.
- [Gol03] David Edward Goldberg. *Algorytmy genetyczne i ich zastosowania*. Wydawnictwa Naukowo-Techniczne, 2003.
- [NK00] B. Naudts i L. Kallel. "A comparison of predictive measures of problem difficulty in evolutionary algorithms". W: *IEEE Transactions on Evolutionary Computation* 4.1 (2000), s. 1–15. DOI: [10.1109/4235.843491](https://doi.org/10.1109/4235.843491).
- [RW95] Colin R. Reeves i Christine C. Wright. "Epistasis in Genetic Algorithms: An Experimental Design Perspective". W: *ICGA*. 1995, s. 217–224. URL: https://www.researchgate.net/profile/Christine_Wright5/publication/2504851_Epistasis_in_Genetic_Algorithms_An_Experimental_Design_Perspective/links/0912f50bf2d3d75799000000.pdf.



**Fundusze
Europejskie**
Polska Cyfrowa



**Rzeczpospolita
Polska**

Unia Europejska
Europejski Fundusz
Rozwoju Regionalnego



Algorytmy i modele inspirowane biologicznie

3. Warianty i specjalizacje algorytmów ewolucyjnych

Maciej Komosiński



**Fundusze
Europejskie**
Polska Cyfrowa



**Rzeczpospolita
Polska**

Unia Europejska
Europejski Fundusz
Rozwoju Regionalnego



Nieporządkny algorytm genetyczny

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Ang. *messy genetic algorithm*.

Podejście „nieporządkne” (niechlujne) ma na celu polepszenie własności algorytmu genetycznego poprzez skuteczniejsze konstruowanie, wykorzystanie i przekształcanie schematów. Taki sam cel ma realizować operator inwersji, który omówimy w kolejnej prezentacji poświęconej mechanizmom inspirowanym naturą. Nieporządkny algorytm genetyczny [Gol+93][Mic96, str. 122] wykorzystuje szczególną reprezentację osobników: genotypy są zmiennej długości, złożone z par (*etykieta, wartość*). Etykieta to opis znaczenia genu – podobnie jak w operatorze inwersji, etykieta może być pierwotnym numerem genu.

Dozwolone są genotypy niekompletne (niedospecyfikowane), tzn. nie określające wartości wszystkich genów. Genotyp może zawierać również geny nadmiarowe, a nawet sprzeczne.

Nieporządkny algorytm genetyczny: działanie

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Używa się trzech operatorów: cięcia, łączenia i mutacji. Operator cięcia rozcina z pewnym prawdopodobieństwem, w losowo wybranym miejscu, łańcuch bitów. Operator łączenia skleja z pewnym prawdopodobieństwem dwa genotypy. Mutacja jest identyczna z mutacją prostą.

Stosowana jest selekcja turniejowa. Proces ewolucji składa się z dwóch (potencjalnie wielokrotnie wykonywanych) faz: wyboru bloków budujących i stosowania operatorów. Liczebność populacji jest zmieniona w trakcie działania algorytmu.

Nadmiarowość informacji w genotypie można prosto rozwiązać wybierając pierwszą napotkaną wartość danego genu w genotypie, ale istnieją też inne metody – np. uśrednianie wszystkich wartości genu albo stosowanie pewnego rodzaju głosowania, którą wybrać. Niedoprecyzowane genotypy, o ile nie są akceptowalne w danym problemie optymalizacji, rozwiązuje się uzupełniając brakujące geny najlepszymi znanyymi wartościami danego genu z wcześniejszej fazy algorytmu.

Nieporządkne algorytmy genetyczne w problemach zwodniczych działały kilkakrotnie lepiej od klasycznych algorytmów genetycznych z krzyżowaniem punktowym.

Hierarchiczny algorytm genetyczny – cel

Nieporządky
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp \rightarrow
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Dyskusja: czy można w jakiś sposób „wykryć” epistazę?

Hierarchiczny algorytm genetyczny

Nieporządky
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Podobnie jak w przypadku nieporządkowych algorytmów genetycznych, motywacją do rozwoju H-GA była chęć automatycznego odkrywania stopnia współzależności (ang. *linkage*, nie mylić z [genetic linkage](#)) elementów rozwiązania w celu dekompozycji problemu. Poprzez próbkowanie specjalnie skonstruowanych rozwiązań można z pewnym prawdopodobieństwem określić zależność lub niezależność genów i grup genów, a następnie dla wykrytych niezależnych grup (modułów) prowadzić niezależną optymalizację [JTW04].

Aby zbadać w pełni niezależność dwóch genów od reszty rozwiązania, należałyby wygenerować zbiór rozwiązań, w których wszystkie możliwe pary wartości tych dwóch genów są otoczone wszystkimi możliwymi wartościami pozostałych genów (stanowiących „kontekst”). Następnie wszystkie te rozwiązania należałyby ocenić i wyznaczyć zależność między wartościami genów a wartością funkcji celu. Byłoby to bardzo kosztowne obliczeniowo, a przecież to byłby tylko test dla jednej pary genów! Dlatego też stosuje się próbkowanie, które umożliwia oszacowanie potencjalnej niezależności dla wszystkich podzbiorów genów w rozwiązaniu (por. GOMEA: Gene-pool Optimal Mixing EA [Thi18]).

Wykrywanie zależności: techniki statystyczne i empiryczne

Nieporządny
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp \rightarrow
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Wizja automatycznej dekompozycji problemu optymalizacji jest bardzo atrakcyjna – pozostaje kwestią metod i ich wydajności, dlatego to zagadnienie jest aktywnie badane.

Metody wykrywania zależności między genami dzieli się czasem na statystyczne (np. H-GA albo rodzina metod GOMEA) i empiryczne (np. DLED: Direct Linkage Empirical Discovery). Te pierwsze bazują na osobnikach w populacji oraz ich ocenach i na tej podstawie statystycznie szacują niezależność genów. Te drugie próbują i oceniają pełne otoczenie konkretnego osobnika, zatem w jego lokalnym sąsiedztwie i dla jego konkretnego zestawu wartości genów uzyskują kompletną informację o (nie)zależności.

Bazując na tej informacji dowiadujemy się, czy i jak można dokonać dekompozycji problemu – co może mieć miejsce już w trakcie działania algorytmu [PKF21, Rozdział 5]. Algorytm może dzięki temu odpowiednio zarządzać podpopulacjami optymalizującymi potencjalnie niezależne podproblemy, dostosowywać operator krzyżowania, mutacji, itp.

Szacowanie epistazy i dekompozycja – technika DLED

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Założymy, że mamy osobnika, którego genotyp składa się z co najwyżej pięciu elementów. Istnienie każdego elementu reprezentujemy jednym bitem (np. cztery z pięciu elementów to np. osobnik 10111).

Współzależności między genami uwypuklają się w optimach lokalnych. Dlatego jeśli nam na tym zależy, można poddawać analizie osobniki będące optimami lokalnymi – można je np. wcześniej zoptymalizować metodą Greedy (z losową kolejnością sąsiadów-genów) zamieniając pojedyncze $1 \rightarrow 0$ i $0 \rightarrow 1$.

Szacowanie epistazy i dekompozycja – technika DLED

Nieporządkny AG
Hierarchiczny AG
Epistaza – DLED Epistaza – przykład ciągły
Strategie ewolucyjne
Ewolucja różnicowa
Programowanie ewolucyjne
Liczby rzeczywiste Mapowanie genotyp → fenotyp
Programowanie genetyczne
Bibliografia

Założymy, że mamy osobnika, którego genotyp składa się z co najwyżej pięciu elementów. Istnienie każdego elementu reprezentujemy jednym bitem (np. cztery z pięciu elementów to np. osobnik 10111).

Współzależności między genami uwypuklają się w optimach lokalnych. Dlatego jeśli nam na tym zależy, można poddawać analizie osobniki będące optimami lokalnymi – można je np. wcześniej zoptymalizować metodą Greedy (z losową kolejnością sąsiadów–genów) zamieniając pojedyncze $1 \rightarrow 0$ i $0 \rightarrow 1$.

Na analizowanym genotypie osobnika wykonujemy dekompozycję typu DLED [PKF21]:

- ① Dla każdego genu A wprowadzamy perturbację (zmieniamy wartość na odwrotną).
- ② Sprawdzamy wszystkie pozostałe geny, jak dla genu A po perturbacji wartość innego genu B wpływa na fitness jeśli pozostaje niezmieniona oraz jeśli będzie zmieniona.
- ③ Decyzja o zależności jest binarna i wynika ze spełnienia warunku/warunków z kolejnego slajdu.

Technika DLED: warunki zależności genów

Warunki z artykułu [PTK23]:

The situation changed when the Direct Empirical Linkage Discovery (DLED) was proposed [17]. To check the dependency between genes g, h , DLED requires computing $f(\mathbf{x})$, $f(\mathbf{x}, g)$, $f(\mathbf{x}, h)$, $f(\mathbf{x}, g, h)$ values, where (\mathbf{x}, g) and (\mathbf{x}, h) are the genotypes of individual \mathbf{x} with genes g and h flipped, respectively. Finally, $f(\mathbf{x}, g, h)$ is the genotype of \mathbf{x} with both genes flipped. In DLED, genes g and h are considered dependent if at least one of the conditions holds:

- C1. $f(\mathbf{x}) < f(\mathbf{x}, g) \text{ } \& \text{ } f(\mathbf{x}, h) \geq f(\mathbf{x}, g, h)$
- C2. $f(\mathbf{x}) = f(\mathbf{x}, g) \text{ } \& \text{ } f(\mathbf{x}, h) \neq f(\mathbf{x}, g, h)$
- C3. $f(\mathbf{x}) > f(\mathbf{x}, g) \text{ } \& \text{ } f(\mathbf{x}, h) \leq f(\mathbf{x}, g, h)$
- C4. $f(\mathbf{x}) < f(\mathbf{x}, h) \text{ } \& \text{ } f(\mathbf{x}, g) \geq f(\mathbf{x}, g, h)$
- C5. $f(\mathbf{x}) = f(\mathbf{x}, h) \text{ } \& \text{ } f(\mathbf{x}, g) \neq f(\mathbf{x}, g, h)$
- C6. $f(\mathbf{x}) > f(\mathbf{x}, h) \text{ } \& \text{ } f(\mathbf{x}, g) \leq f(\mathbf{x}, g, h)$

The above conditions can be interpreted as the following statement. If the modification of one gene changes the fitness relations for the values of the other gene, then these two genes are dependent. DLED is an ELL technique proven to report only the direct dependencies.

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

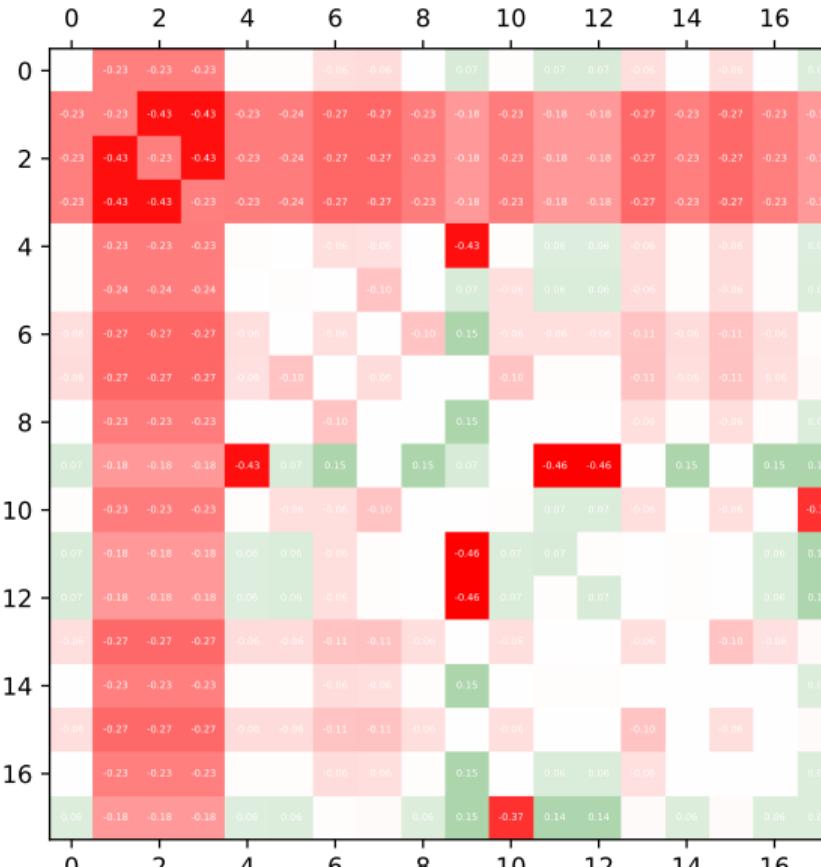
Szacowanie epistazy i dekompozycja – konkretny przykład

Zmiana w wartości fitness po wyłączeniu par genów w przykładowym osobniku

Szacowanie epistazy i dekompozycja – konkretny przykład

Zmiana w wartości fitness po wyłączeniu par genów; na przekątnej efekt wyłączenia jednego genu

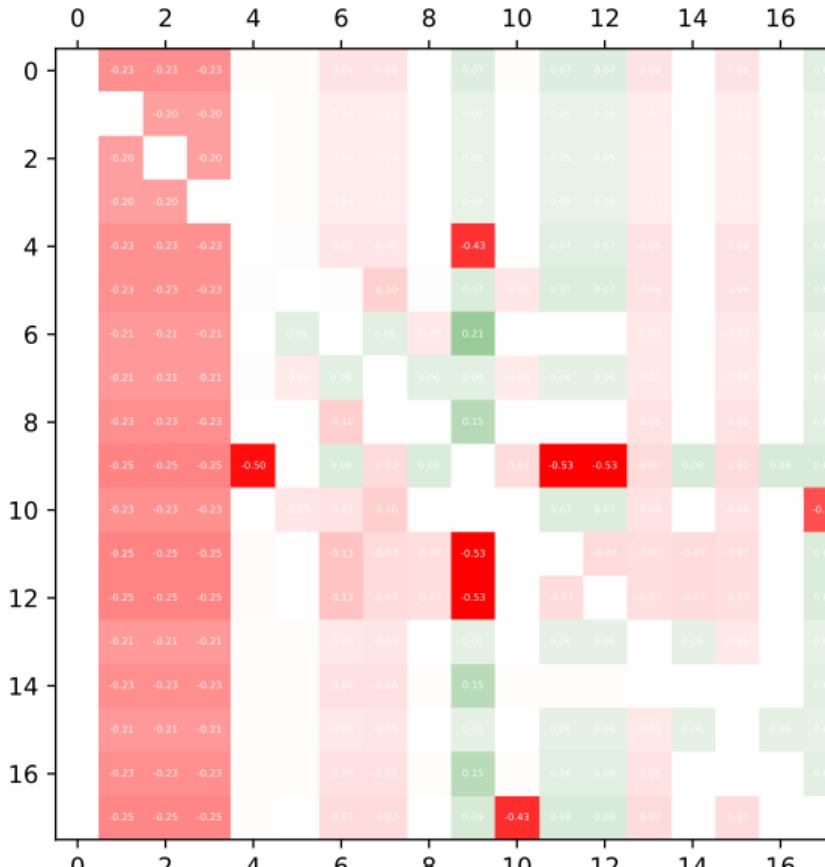
Nieporządkowy AG
Hierarchiczny AG
Epistaza – DLED
Epistaza – przykład ciągły
Strategie ewolucyjne
Ewolucja różnicowa
Programowanie ewolucyjne
Liczby rzeczywiste
Mapowanie genotyp → fenotyp
Programowanie genetyczne
Bibliografia



Szacowanie epistazy i dekompozycja – konkretny przykład

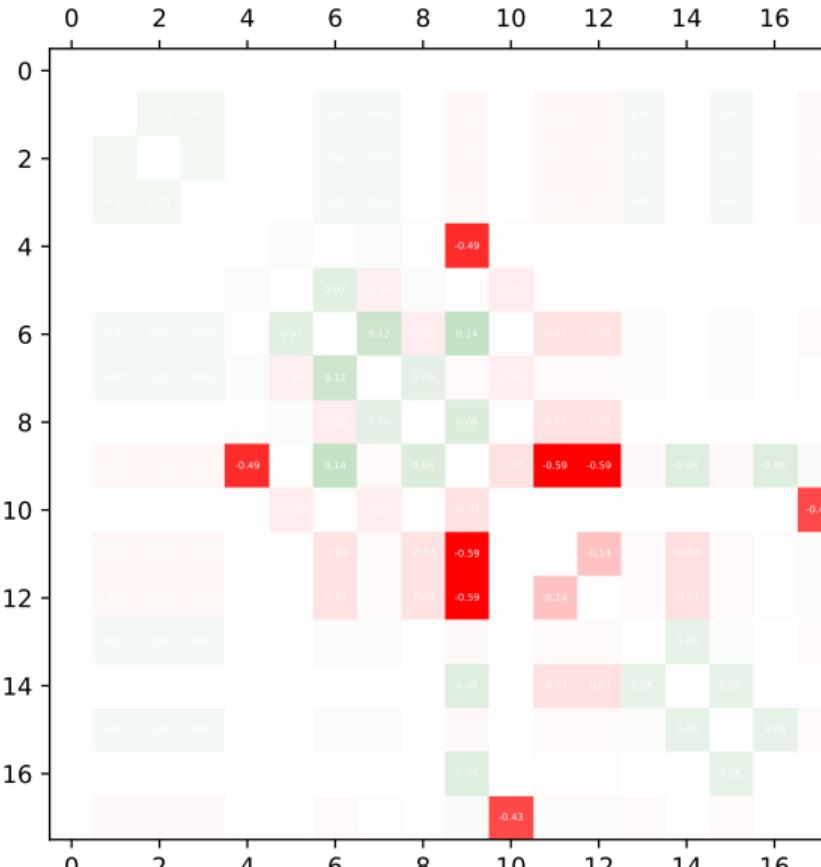
Ojęcie wartości na przekątnej od wierszy: jak interpretować efekt?

Nieporządkowy AG
Hierarchiczny AG
Epistaza – DLED
Epistaza – przykład ciągły
Strategie ewolucyjne
Ewolucja różnicowa
Programowanie ewolucyjne
Liczby rzeczywiste
Mapowanie genotyp \rightarrow fenotyp
Programowanie genetyczne
Bibliografia



Szacowanie epistazy i dekompozycja – konkretny przykład

Addytywnośc wpływu lub jej brak



Szacowanie epistazy i dekompozycja

Nieporządny
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Co powinno być wierszami i kolumnami (w tym przykładzie to było „wyłączanie genów”, ale czy o to zawsze chodzi?)

Szacowanie epistazy i dekompozycja

Co powinno być wierszami i kolumnami (w tym przykładzie to było „wyłączanie genów”, ale czy o to zawsze chodzi?)

Jak wykorzystać te informacje w algorytmie podczas optymalizacji?

Nieporządny
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Szacowanie epistazy i dekompozycja

Nieporządkny
AG

Hierarchiczny
AG
Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Co powinno być wierszami i kolumnami (w tym przykładzie to było „wyłączanie genów”, ale czy o to zawsze chodzi?)

Jak wykorzystać te informacje w algorytmie podczas optymalizacji?

- optymalizować niezależne podzbiory genów osobno → zmniejszenie złożoności obliczeniowej
- zaprojektować krzyżowanie i mutację tak, aby zachowywały korzystne epistatyczne interakcje genów → traktować współpracujące epistatycznie grupy genów jako integralne zespoły → ochrona i skuteczna propagacja „cegiełek” budujących dobre rozwiązania [GT12; TB13]
- przykład: operator *Optimal Mixing* (OM) w algorytmie GOMEA. Wybiera z populacji rodzica i donora, a następnie tworzy potomka przenosząc z donora do rodzica allele współzależnych (współdziałających) genów. Akceptuje potomka tylko gdy jest on co najmniej tak dobry, jak rodzic [Thi18].

Szacowanie epistazy i dekompozycja

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Co powinno być wierszami i kolumnami (w tym przykładzie to było „wyłączanie genów”, ale czy o to zawsze chodzi?)

Jak wykorzystać te informacje w algorytmie podczas optymalizacji?

- optymalizować niezależne podzbiory genów osobno → zmniejszenie złożoności obliczeniowej
- zaprojektować krzyżowanie i mutację tak, aby zachowywały korzystne epistatyczne interakcje genów → traktować współpracujące epistatycznie grupy genów jako integralne zespoły → ochrona i skuteczna propagacja „cegiełek” budujących dobre rozwiązania [GT12; TB13]
- przykład: operator *Optimal Mixing* (OM) w algorytmie GOMEA. Wybiera z populacji rodzica i donora, a następnie tworzy potomka przenosząc z donora do rodzica allele współzależnych (współdziałających) genów. Akceptuje potomka tylko gdy jest on co najmniej tak dobry, jak rodzic [Thi18].

Jak zastosować analogiczne podejście do wykrycia współzależności trzeciego rzędu? (między trójkami genów?)

Szacowanie epistazy i dekompozycja – konkretny przykład

Bohater analizy: genotyp, fenotyp i fitness

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

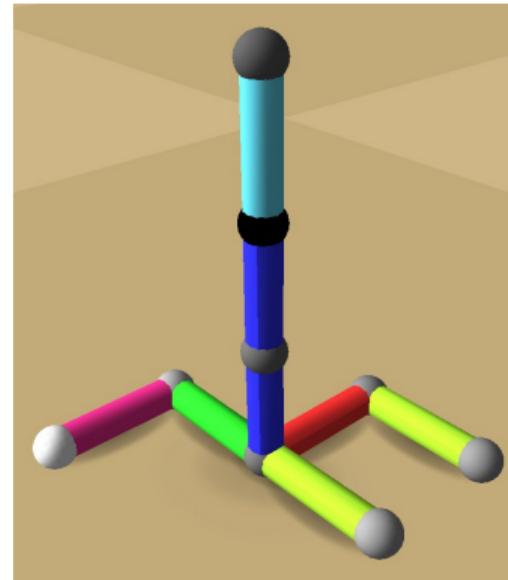
Liczby rzeczywiste

Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Genotyp: /*9*/UDDDLFBFBRFBBFBR



Fitness: wysokość środka ciężkości konstrukcji
(0.47 dla oryginalnego genotypu – dobrego, ale nie lokalnie optymalnego).

Strategie ewolucyjne: początki

Nieporządkny
AG

Hierarchiczny
AG
Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne
Liczby rzeczywiste
Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Strategie ewolucyjne (ang. *Evolutionary Strategies*, ES) rozwijały się przez pewien czas niezależnie od AG, jako metody służące do optymalizacji numerycznej. Wiele aspektów różni je od AG; wspólnie jest wykorzystanie mechanizmów ewolucji podczas optymalizacji.

Strategie ewolucyjne: początki

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp \rightarrow
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Strategie ewolucyjne (ang. *Evolutionary Strategies*, ES) rozwijały się przez pewien czas niezależnie od AG, jako metody służące do optymalizacji numerycznej. Wiele aspektów różni je od AG; wspólnie jest wykorzystanie mechanizmów ewolucji podczas optymalizacji.

ES – naturalne pochodzenie: jedno z pierwszych zastosowań (1964) to *evolutionary design*, czyli projektownie konstrukcji (powiemy o tym problemie później i poeksperymentujemy na laboratorium). W celu minimalizacji oporów przepływu wody i optymalizacji kształtów rur, aby ocenić konstrukcję lub rurociąg, nie symulowano jej, tylko budowano [Rec84, zob. rys. na str. 123]; zmiany konstrukcji odpowiadały „mutacjom”. Był to zatem sposób postępowania realizujący algorytm optymalizacji.

Strategie ewolucyjne: najpierw dwuelementowe

Nieporządkowy AG

Hierarchiczny AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie ewolucyjne

Ewolucja różnicowa

Programowanie ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp \rightarrow fenotyp

Programowanie genetyczne

Bibliografia

Wczesne strategie ewolucyjne używały jedynie operatora mutacji, który modyfikował jedynego przetwarzanego osobnika. Inaczej niż w algorytmach genetycznych, osobnik był parą składającą się z wektora wartości zmiennych i wektora odchyлеń standardowych (stałego podczas całego procesu ewolucji). Mutacja polegała na zmianie każdej zmiennej wektora wartości o losowy czynnik wygenerowany zgodnie z rozkładem normalnym o odpowiednim odchyleniu standardowym (określonym w wektorze odchyłeń standardowych). Osobnik po mutacji zastępował swojego przodka jedynie wówczas, gdy był od niego lepszy i dopuszczalny.

Taka strategia została nazwana dwuelementową (ponieważ w pewnym momencie istnieje jeden potomek i jeden przodek) i jest oznaczana (1+1)-ES, a jej działanie przypomina *Local Search*.

Strategie ewolucyjne: populacja i krzyżowanie

Nieporządny
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Ulepszeniem strategii dwuelementowej jest strategia wieloelementowa, w której, podobnie jak w AG, istnieje populacja osobników. Wprowadza się dodatkowo operację krzyżowania jednorodnego, jednak nie stosuje się jej do wszystkich osobników, tylko do dwóch z nich – tak, że powstaje jeden potomek, który zastępuje osobnika najsłabszego (jeden nowy osobnik – a więc analogicznie, jak w algorytmach ewolucyjnych typu *steady-state*).

Kolejnym udoskonaleniem było stosowanie krzyżowania wiele razy w jednym kroku (powstawało wielu potomków), a następnie wybór z przodków i potomków *POPSIZE* osobników (tzw. selekcja typu „plus”). W innym podejściu wybiera się osobników do następnego pokolenia tylko z grupy potomków (tzw. selekcja typu „przecinek”, ang. *comma selection*), co jest korzystne w zadaniach ze zmieniającym się optimum.

Strategie ewolucyjne: specjalna notacja

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp \rightarrow
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Ogólny i zwięzły zapis architektury ES ma postać $(\mu/\rho, \lambda)$ -ES albo $(\mu/\rho + \lambda)$ -ES, gdzie μ to liczba rodziców, $\rho \leq \mu$ to liczba rodziców z których wywodzą się potomkowie, λ to liczba potomków, przecinek oznacza selekcję tylko ze zbioru potomków, a plus – z obu zbiorów: rodziców i potomków.

Strategie ewolucyjne: mutacja i krzyżowanie

Nieporządkowy AG

Hierarchiczny AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie ewolucyjne

Ewolucja różnicowa

Programowanie ewolucyjne

Liczby rzeczywiste

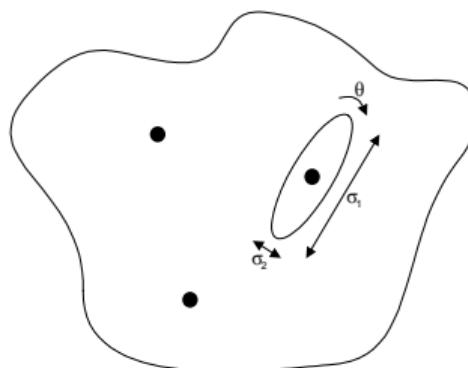
Mapowanie genotyp \rightarrow fenotyp

Programowanie genetyczne

Bibliografia

Stosuje się ulepszony operator mutacji, który zmienia nie tylko wartość zmiennej, ale także odchylenie standardowe zmian, które podlega również ewolucji. Do reprezentacji osobnika, oprócz wartości zmiennych i odchyleń standardowych, można wprowadzić dodatkowo informację o preferowanym kącie odchylenia podczas procesu przeszukiwania i w ten sposób poprawić szybkość zbieżności strategii ewolucyjnych. Zmienna jest wówczas reprezentowana przez jej wartość, odchylenie standardowe i kąt odchylenia, i wszystkie te wielkości podlegają ewolucji pozwalając na samoadaptację i umożliwiając dokładne dostrojenie lokalne.

Używa się również krzyżowania arytmetycznego (średnia ważona rodziców).



Rysunek: Parametry mutacji w strategiach ewolucyjnych.

Strategie ewolucyjne: macierz kowariancji mutacji

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Znana z wydajności jest strategia ewolucyjna dostosowująca macierz kowariancji mutacji, CMA-ES,* której implementacja jest dostępna np. w bibliotece DEAP.** Ta strategia jest też adekwatna dla problemów **źle uwarunkowanych** (ang. ill-conditioned).

Metoda CMA-ES ma wiele parametrów, istnieje też wiele alternatywnych mechanizmów dla każdego kroku tej metody. Można stosować wartości domyślne oraz polityki wielu uruchomień uwalniające użytkownika od konieczności podawania wartości jakichkolwiek parametrów.

*<https://en.wikipedia.org/wiki/CMA-ES>

**<https://deap.readthedocs.io/en/master/examples/cmaes.html>

Strategie ewolucyjne: główna idea CMA-ES

- ustalamy środek populacji,

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Strategie ewolucyjne: główna idea CMA-ES

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

- ustalamy środek populacji,
- próbujemy rozwiązania z wielowymiarowego (n) rozkładu normalnego (dany jednym parametrem – izometryczny, albo n parametrami – skalowanie równoległe do osi, albo $\binom{n}{2}$ parametrami czyli macierzą kowariancji – umożliwia obracanie),

Strategie ewolucyjne: główna idea CMA-ES

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

- ustalamy środek populacji,
- próbujemy rozwiązań z wielowymiarowego (n) rozkładu normalnego (dany jednym parametrem – izometryczny, albo n parametrami – skalowanie równoległe do osi, albo $\binom{n}{2}$ parametrami czyli macierzą kowariancji – umożliwia obracanie),
- oceniamy wszystkie rozwiązania,

Strategie ewolucyjne: główna idea CMA-ES

Nieporządkowy AG

Hierarchiczny AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie ewolucyjne

Ewolucja różnicowa

Programowanie ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp → fenotyp

Programowanie genetyczne

Bibliografia

- ustalamy środek populacji,
- próbujemy rozwiązań z wielowymiarowego (n) rozkładu normalnego (dany jednym parametrem – izometryczny, albo n parametrami – skalowanie równoległe do osi, albo $\binom{n}{2}$ parametrami czyli macierzą kowariancji – umożliwia obracanie),
- oceniamy wszystkie rozwiązania,
- przesuwamy środek populacji: ustawiamy go w miejscu średniej ważonej jakością (rankingową) najlepszych osobników. Rankingowość powoduje nieczułość na niewielkie zaburzenia oceny („chropowatość” krajobrazu) oraz jego wyginanie – stopień wkleśności,

Strategie ewolucyjne: główna idea CMA-ES

Nieporządkowy AG

Hierarchiczny AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie ewolucyjne

Ewolucja różnicowa

Programowanie ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp → fenotyp

Programowanie genetyczne

Bibliografia

- ustalamy środek populacji,
- próbujemy rozwiązań z wielowymiarowego (n) rozkładu normalnego (dany jednym parametrem – izometryczny, albo n parametrami – skalowanie równoległe do osi, albo $\binom{n}{2}$ parametrami czyli macierzą kowariancji – umożliwia obracanie),
- oceniamy wszystkie rozwiązania,
- przesuwamy środek populacji: ustawiamy go w miejscu średniej ważonej jakością (rankingową) najlepszych osobników. Rankingowość powoduje nieczułość na niewielkie zaburzenia oceny („chropowatość” krajobrazu) oraz jego wyginanie – stopień wklęsłości,
- rozproszenie nowych (próbkowanych) osobników jest proporcjonalne do prędkości, z jaką przemieszcza się środek populacji: wolniejsze przemieszczanie → mniejsze rozproszenie,

Strategie ewolucyjne: główna idea CMA-ES

Nieporządkowy AG

Hierarchiczny AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie ewolucyjne

Ewolucja różnicowa

Programowanie ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp \rightarrow fenotyp

Programowanie genetyczne

Bibliografia

- ustalamy środek populacji,
- próbujemy rozwiązań z wielowymiarowego (n) rozkładu normalnego (dany jednym parametrem – izometryczny, albo n parametrami – skalowanie równoległe do osi, albo $\binom{n}{2}$ parametrami czyli macierzą kowariancji – umożliwia obracanie),
- oceniamy wszystkie rozwiązania,
- przesuwamy środek populacji: ustawiamy go w miejscu średniej ważonej jakością (rankingową) najlepszych osobników. Rankingowość powoduje nieczułość na niewielkie zaburzenia oceny („chropowatość” krajobrazu) oraz jego wyginanie – stopień wklęsłości,
- rozproszenie nowych (próbkowanych) osobników jest proporcjonalne do prędkości, z jaką przemieszcza się środek populacji: wolniejsze przemieszczanie \rightarrow mniejsze rozproszenie,
- aktualizujemy macierz kowariancji tak, żeby rozciągnąć nieco wielowymiarowy rozkład normalny w kierunku przemieszczenia środka populacji. W ten sposób będziemy dalej podążali zgodnie z aproksymowanym gradientem oczekiwanej jakości rozwiązań.

Ewolucja różnicowa (*differential evolution*)

Nieporządkowy AG

Hierarchiczny AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie ewolucyjne

Ewolucja różnicowa

Programowanie ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp \rightarrow fenotyp

Programowanie genetyczne

Bibliografia

Specyficzną cechą ewolucja różnicowej jest mutacja różnicowa [SP97]. W każdej iteracji ewolucji, dla każdego osobnika o z populacji N osobników powtarzamy:

- losujemy n unikatowych z N osobników, wybieramy z nich osobnika bazowego β i osobnika różnicy δ (dla $n = 3$, β może być wybrany z nich losowo, a δ może być różnicą dwóch pozostałych osobników),
- tworzymy osobnika tymczasowego („donora”) $\omega = \beta + F\delta$ (F – stała),
- krzyżujemy ω z o ,
- decydujemy czy efekt krzyżowania ma zastąpić oryginalnego o , czy nie (selekcja).

DE jest znana ze swojej prostoty, małej liczby parametrów ([przykładowa implementacja](#)) i szybkiej zbieżności. Nie wymaga określenia osobnego, niezależnego rozkładu prawdopodobieństwa dla mutacji – mutacja wynika ze stanu populacji. Warianty DE są konkurencyjne w stosunku do innych algorytmów w corocznych konkursach optymalizacyjnych.

Tworzenie osobnika tymczasowego ω : porównaj później omawiane krzyżowanie simpleksowe.

Programowanie ewolucyjne (*evolutionary programming*)

Nieporządkowy AG

Hierarchiczny AG
Epistaza – DLED
Epistaza – przykład ciągły

Strategie ewolucyjne

Ewolucja różnicowa

Programowanie ewolucyjne
Liczby rzeczywiste
Mapowanie genotyp \rightarrow fenotyp

Programowanie genetyczne

Bibliografia

Trzy główne różnice pomiędzy EP a GA:

- ① Reprezentacja rozwiązania nie musi być binarna – wynika naturalnie z problemu.
- ② Mutacja zmienia fragmenty rozwiązania, przy czym małe zmiany są częstsze, a duże – rzadsze.
- ③ Krzyżowanie może nie występować.

Obecnie „*evolutionary programming*” jest nazwą rzadko używaną, zamiast niej mówi się o algorytmie ewolucyjnym – co oznacza ogólnie, że użyto algorytmu przystosowanego do danego problemu. Stopień jego przystosowania bywa różny; najczęściej obejmuje reprezentację i operatory.

Wykorzystuje się wiele reprezentacji osobników: zbiór, lista, permutacja*, drzewo, graf nieskierowany, graf skierowany, macierz, wyrażenia logiczne, reguły (jak w genetycznym uczeniu maszynowym, [LCS/GBML](#)), sieci neuronowe, automaty, wyrażenia opisane gramatyką (np. zapisane w ONP), wyrażenia o strukturze drzewa, programy (jak w omawianym dalej GP), ...

*Krzyżowanie dla permutacji: [OX](#), [PMX](#), [ERO](#), inne: <https://hrcak.srce.hr/file/163313>

Reprezentacja liczb rzeczywistych – operatory

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp \rightarrow
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Bardzo często w AE (i w optymalizacji w ogóle!) stosuje się reprezentację wartości ciągłych. Geny kodują liczby rzeczywiste w takim formacie, jak jest to przyjęte na procesorach (zmienna precyzja w zależności od bezwzględnej wartości liczby).

Pytanie: jakie można zaproponować operatory krzyżowania i mutacji (oprócz typowych, takich jak wymiana genów albo wielopunktowe) dla wektora liczb? Proponując operatory pamiętaj o celu krzyżowania i mutacji.

Reprezentacja liczb rzeczywistych – krzyżowanie

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Krzyżowanie: np. średnia z rodziców, lub średnia ważona by uzyskać dwóch różnych potomków. Średnia ważona to krzyżowanie **arytmetyczne** – dzieci są liniową kombinacją rodziców: $d_1 = r_1 \cdot a + r_2 \cdot (1 - a)$, $d_2 = \dots$

Wagę a można losować przy każdym wykonaniu.

Krzyżowanie simpleksowe

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

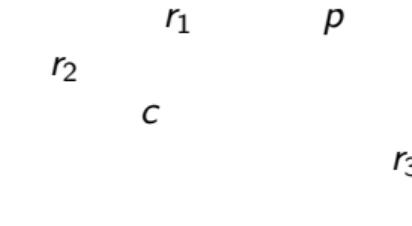
Liczby rzeczywiste
Mapowanie genotyp \rightarrow
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Wyznaczamy centroid c rodziców r .

- Wariant bez dostępu do jakości rozwiązań – SPX [TYH99]: losujemy potomka z (powiększonej) przestrzeni kombinacji liniowych rodziców (odsuniętych od c o ε – *the expanding rate*).
- Wariant z uwzględnieniem jakości: tworzymy potomka p jako przesunięcie od najgorszego osobnika/rodzica dalej poprzez punkt c .



Reprezentacja liczb rzeczywistych – mutacja

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp \rightarrow
fenotyp

Programo-
wanie
genetyczne

Bibliografia

- *uniform random (Flat)* – ustal gen na wartość losową z dozwolonego przedziału
- *creep* – zmień gen o wartość losowaną z pewnego rozkładu (np. normalnego albo jednostajnego – np. $-3..+3$, itp.)

Aby uniezależnić się od „osi” (tzn. aby mutacja nie przebiegała tylko równolegle do osi, czyli nie dotyczyła pojedynczych parametrów, co byłoby niekorzystne gdyby funkcja celu była np. obróconą wersją funkcji zależnej wprost od parametrów), mutuje się naraz wszystkie geny (i wtedy stosujemy rozkład normalny losowanej zmiany a nie równomierny – zastanów się dlaczego).

Chcąc zapewnić, że taka mutacja naraz n elementów wektora przesunie bieżące rozwiązanie o taką samą odległość w n -wymiarowej przestrzeni, jak zrobiłaby to mutacja tylko jednego wymiaru, przez jaką wartość należy podzielić (znormalizować) każdą z n wylosowanych wartości zmiany w n -wymiarowym wektorze?

Co zrobić, jeśli po zmutowaniu wartość genu wychodzi poza dozwolony zakres? Jakie metody rozwiązania tego problemu można zaproponować? [Bul99; Bul01]

Reprezentacja liczb rzeczywistych – mutacja

Metody radzenia sobie z mutantami spoza dozwolonego przedziału

Nieporządkny AG
Hierarchiczny AG
Epistaza – DLED Epistaza – przykład ciągły
Strategie ewolucyjne
Ewolucja różnicowa
Programowanie ewolucyjne
Liczby rzeczywiste Mapowanie genotyp \rightarrow fenotyp
Programowanie genetyczne
Bibliografia

- **Pochłaniaj:** niepoprawna po mutacji wartość jest obcinana do najbliższej granicy przedziału – znany trick $\max(\min())$.
- **Powtarzaj:** zmutowana wartość jest generowana tak długo, aż uzyskamy poprawną wartość.
- **Zastępuj:** zmutowana wartość jest generowana tak długo, aż uzyskamy poprawną wartość, za każdym razem wybierając od nowa przodka.

- **Ignoruj:** niepoprawna po mutacji wartość jest zapominana; mutant uzyskuje oryginalną wartość swojego przodka.
- **Odbijaj:** niepoprawna po mutacji wartość leżąca d powyżej (lub poniżej) najbliższej jej granicy przedziału jest zastępowana przez wartość leżącą d poniżej (lub powyżej) tej granicy.
- **Zawijaj:** mutowana cecha jest traktowana tak, jakby była cykliczna. Przedział się „zawija”, a zmutowana wartość jest wyznaczana modulo szerokość przedziału.

Czy ma znaczenie, którą z metod wybierzemy? Tak! [dyskusja pożądanych cech mutacji i wybór zwycięskiej metody].

Reprezentacja liczb rzeczywistych – mutacja

Działanie metod radzenia sobie z niepoprawnymi mutantami

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

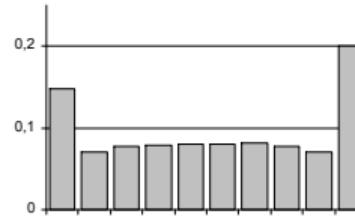
Liczby rzeczywiste
Mapowanie genotyp \rightarrow
fenotyp

Programo-
wanie
genetyczne

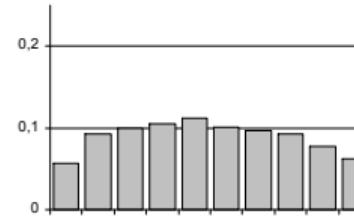
Bibliografia

Wyniki eksperymentalne: jak często zdarzały się po mutacji i „naprawie” różnymi metodami określone wartości genu? Przodek miał równe szanse na każdą wartość. Oś pozioma – przedział zmienności genu, oś pionowa – częstotliwość występowania wartości potomka w podprzedziałach:

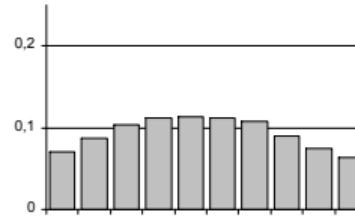
Pochłaniaj



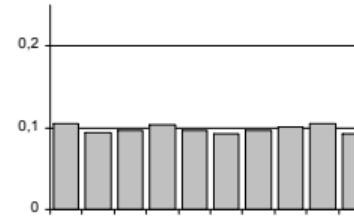
Powtarzaj



Zastępuj



Ignoruj, Odbijaj, Zawijaj,
Flat – podobne do:



Reprezentacja liczb rzeczywistych – mutacja

Potrzeba dokładniejszej analizy zachowania mechanizmów naprawy

Nieporządny
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

To, że w metodach **Ignoruj**, **Odbijaj**, **Zawijaj**, uzyskano taki sam rozkład jak we **Flat** nie oznacza, że metody te zachowują się tak samo. Faktycznie zasada ich działania jest zupełnie inna. Przykład: metoda **Ignoruj**. W okolicy granic przedziału więcej mutacji będzie nielegalnych i ignorowanych. Skoro ignorowanych, to rzadziej też będą trafiać się w ogóle wartości bliskie granicom. Kiedy już się trafią, rzadko doprowadzą do poprawnej mutacji...

Zatem to, że rozkład jest taki sam, nie gwarantuje jeszcze, że tak samo często zdarzają się efektywne (faktycznie zmieniające wartość genu) mutacje w poszczególnych podprzedziałach. Stąd oprócz rozkładu częstości wartości genu dla różnych metod należy też porównać inne parametry – np. ile liczb ubywa z określonego podprzedziału (przed mutacją) i przybywa (po mutacji), jaki jest między nimi związek, itp. W tym świetle metody **Ignoruj**, **Odbijaj**, **Zawijaj** różnią się między sobą, i żadna z nich nie zachowuje się tak jak **Flat**.

Reprezentacja liczb rzeczywistych – mutacja

Wpływ mechanizmu naprawy na proces optymalizacji

Nieporządny
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Rozpatrywaliśmy tu nieskomplikowany rodzaj mutacji i proste mechanizmy, a jednak te drobne elementy mają duży wpływ na proces ewolucji. Na przykład **Pochłaniaj** ukierunkowuje (*bias*) **cały czas** dryf genetyczny ku krańcowym wartościom dozwolonych przedziałów. Nie będąc tego świadomym można wyciągnąć wniosek, że są one optymalne i ewolucja je faworyzuje – tymczasem jest to ciągły wpływ mutacji.

Przegląd operatorów mutacji i krzyżowania dla problemów numerycznych zawierają książki–zbiory [Gwi07a; Gwi07b].

Przypomnienie z poprzedniego semestru (i ew. uzupełnienie)

Nieporządky
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp \rightarrow
fenotyp

Programo-
wanie
genetyczne

Bibliografia

- Mając kilka pomysłów na operatory krzyżowania, skąd wiemy, który z nich będzie prawdopodobnie lepszy?

Przypomnienie z poprzedniego semestru (i ew. uzupełnienie)

Nieporządkowy
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp \rightarrow
fenotyp

Programo-
wanie
genetyczne

Bibliografia

- Mając kilka pomysłów na operatory krzyżowania, skąd wiemy, który z nich będzie prawdopodobnie lepszy?
- Jak świadomie tworzyć (projektować) skuteczne operatory krzyżowania?

Przypomnienie z poprzedniego semestru (i ew. uzupełnienie)

Nieporządkowy
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp \rightarrow
fenotyp

Programo-
wanie
genetyczne

Bibliografia

- Mając kilka pomysłów na operatory krzyżowania, skąd wiemy, który z nich będzie prawdopodobnie lepszy?
- Jak świadomie tworzyć (projektować) skuteczne operatory krzyżowania?
- Czym charakteryzuje się DPX – *distance-preserving crossover*?

Przypomnienie z poprzedniego semestru (i ew. uzupełnienie)

Nieporządkowy
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp \rightarrow
fenotyp

Programo-
wanie
genetyczne

Bibliografia

- Mając kilka pomysłów na operatory krzyżowania, skąd wiemy, który z nich będzie prawdopodobnie lepszy?
- Jak świadomie tworzyć (projektować) skuteczne operatory krzyżowania?
- Czym charakteryzuje się DPX – *distance-preserving crossover*?
- Jak wiedza z powyższych pytań przenosi się na operator mutacji?

Przypomnienie z poprzedniego semestru (i ew. uzupełnienie)

Nieporządkowy
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp \rightarrow
fenotyp

Programo-
wanie
genetyczne

Bibliografia

- Mając kilka pomysłów na operatory krzyżowania, skąd wiemy, który z nich będzie prawdopodobnie lepszy?
- Jak świadomie tworzyć (projektować) skuteczne operatory krzyżowania?
- Czym charakteryzuje się DPX – *distance-preserving crossover*?
- Jak wiedza z powyższych pytań przenosi się na operator mutacji?
- Czym jest *globalna wypukłość* i jak ją wykryć?

Przypomnienie z poprzedniego semestru (i ew. uzupełnienie)

Nieporządkowy
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp \rightarrow
fenotyp

Programo-
wanie
genetyczne

Bibliografia

- Mając kilka pomysłów na operatory krzyżowania, skąd wiemy, który z nich będzie prawdopodobnie lepszy?
- Jak świadomie tworzyć (projektować) skuteczne operatory krzyżowania?
- Czym charakteryzuje się DPX – *distance-preserving crossover*?
- Jak wiedza z powyższych pytań przenosi się na operator mutacji?
- Czym jest *globalna wypukłość* i jak ją wykryć?
- Co ocenia miara FDC?

Ilościowe ocenianie podobieństwa/odległości rozwiązań

Nieporządkowy AG

Hierarchiczny AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie ewolucyjne

Ewolucja różnicowa

Programowanie ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp \rightarrow fenotyp

Programowanie genetyczne

Bibliografia

Miara podobieństwa rozwiązań ma liczne zastosowania – przydaje się m.in. do:

- testowania pomysłów na operator krzyżowania – różne cechy rozwiązań i wartości FDC,
- prowadzenia selekcji ze ściskiem – omówionej wcześniej,
- szacowania różnorodności w populacji i oceny zbieżności,
- analizy struktury populacji; analizy skupień w zbiorze rozwiązań,
- utrzymywania „gatunków” podczas ewolucji – te metody będą jeszcze omawiane,

oraz wszędzie tam, gdzie występuje potrzeba wyznaczenia różnicy dwóch rozwiązań, np. w przedstawionych już: ewolucji różnicowej i krzyżowaniu simpleksowym.

Jeśli rozwiązania posiadają prostą reprezentację (rozważ kilka przykładów), wtedy pomysły na miary podobieństwa mogą nasuwać się same. Dla złożonych reprezentacji (rozważ kilka przykładów) przydatne mogą być pojęcia odległości edycyjnej* oraz odległości spychaczowej** (tłum. MK).

*https://en.wikipedia.org/wiki/Edit_distance

**https://en.wikipedia.org/wiki/Earth_mover%27s_distance

Embriogeneza

Nieporządkowy
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Embriogeneza: mapowanie genotyp → fenotyp. Dla prostych reprezentacji i równomiernych, jednorodnych przestrzeni takich jak kompletna przestrzeń bitów, liczb lub permutacji, pierwszym (domyślnym) pomysłem jest trywialne, bezpośrednie mapowanie 1:1.

Ale czy takie mapowanie jest najlepszym wyborem?

Embriogeneza

Nieporządkowy
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

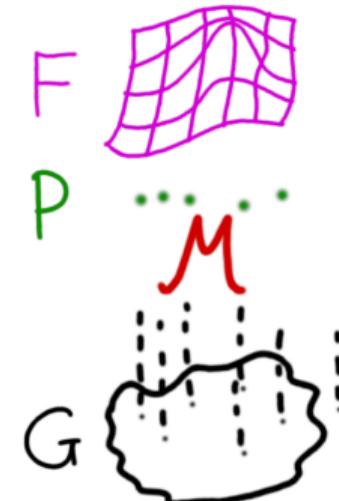
Liczby rzeczywiste
Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Embriogeneza: mapowanie genotyp → fenotyp. Dla prostych reprezentacji i równomiernych, jednorodnych przestrzeni takich jak kompletna przestrzeń bitów, liczb lub permutacji, pierwszym (domyślnym) pomysłem jest trywialne, bezpośrednie mapowanie 1:1.

Ale czy takie mapowanie jest najlepszym wyborem?



Embriogeneza

Nieporządkowy
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

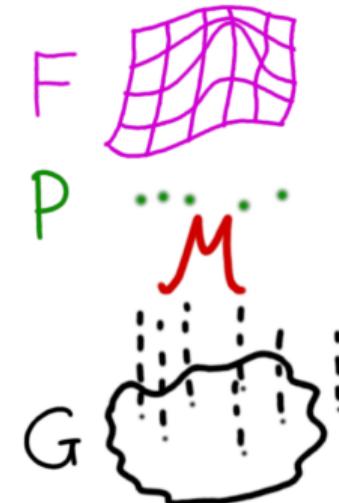
Liczby rzeczywiste
Mapowanie genotyp \rightarrow
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Embriogeneza: mapowanie genotyp \rightarrow fenotyp. Dla prostych reprezentacji i równomiernych, jednorodnych przestrzeni takich jak kompletna przestrzeń bitów, liczb lub permutacji, pierwszym (domyślnym) pomysłem jest trywialne, bezpośrednie mapowanie 1:1.

Ale czy takie mapowanie jest najlepszym wyborem?



Przypomnij sobie $RGB \leftrightarrow HSL$, sygnał \leftrightarrow widmo, kran $\bullet\text{---}\bullet \leftrightarrow \text{---}\text{---}$, ...

Embriogeneza – po co?

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Zastanów się, w jakich sytuacjach mapowanie genotyp → fenotyp powinno (albo musi?) być bardziej skomplikowane. Jakie cechy mapowania powinna zapewnić procedura mapująca przestrzeń genotypów w przestrzeń fenotypów?

Embriogeneza – po co?

Nieporządkowy
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

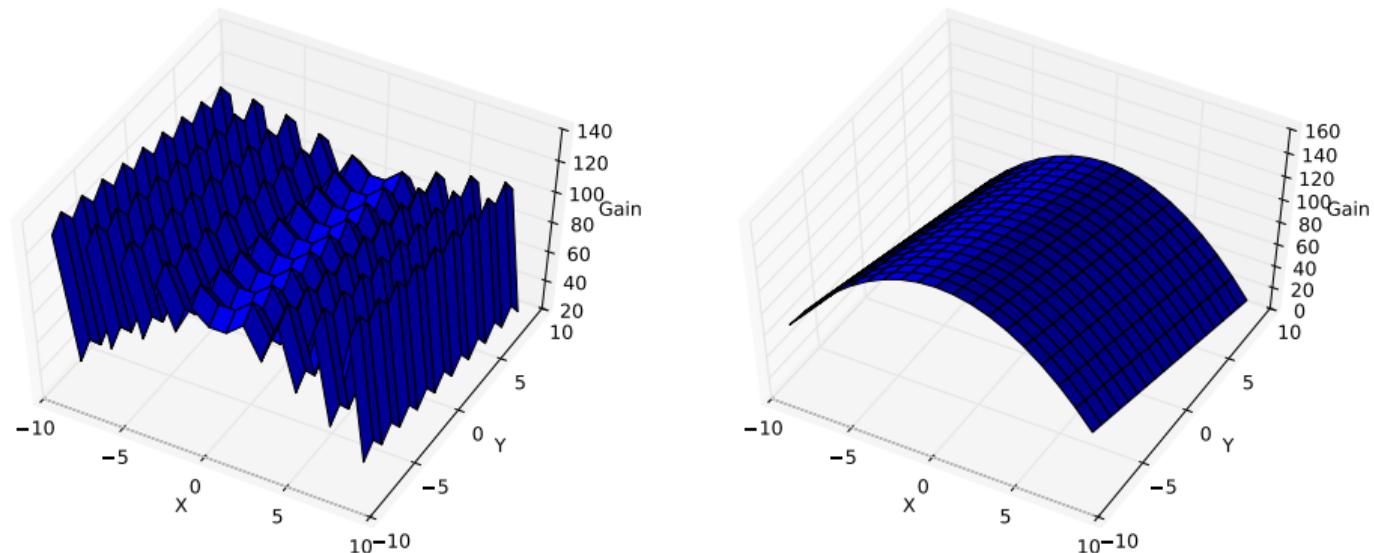
Liczby rzeczywiste

Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Zastanów się, w jakich sytuacjach mapowanie genotyp → fenotyp powinno (albo musi?) być bardziej skomplikowane. Jakie cechy mapowania powinna zapewnić procedura mapująca przestrzeń genotypów w przestrzeń fenotypów?



Embriogeneza – po co?

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Zastanów się, w jakich sytuacjach mapowanie genotyp → fenotyp powinno (albo musi?) być bardziej skomplikowane. Jakie cechy mapowania powinna zapewnić procedura mapująca przestrzeń genotypów w przestrzeń fenotypów?

Pomyśl teraz o naturze, o organizmach żywych i o biologicznym mapowaniu genotyp → fenotyp. Jakie ono jest i czy jest **korzystne**? Czy dałoby się zrealizować to mapowanie lepiej?

Embriogeneza – po co?

Nieporządkowy
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Zastanów się, w jakich sytuacjach mapowanie genotyp → fenotyp powinno (albo musi?) być bardziej skomplikowane. Jakie cechy mapowania powinna zapewnić procedura mapująca przestrzeń genotypów w przestrzeń fenotypów?

Pomyśl teraz o naturze, o organizmach żywych i o biologicznym mapowaniu genotyp → fenotyp. Jakie ono jest i czy jest korzystne? Czy dałoby się zrealizować to mapowanie lepiej?

Jeśli przestrzeń fenotypów jest inna niż genotypów (co ma miejsce np. wtedy, kiedy rozwiązania są bardzo skomplikowane – wyobraź sobie optymalizację mostu, samochodu, robota, . . .), potrzebna jest procedura „mapowania” jednej przestrzeni w drugą. W biologii ten proces to embriogeneza (rozwój z genotypu do stadium zarodka – budowanie ciała). Ale nawet dla identycznych przestrzeni, niebezpośrednie mapowanie może przynieść korzyści.

Embriogeneza – decyzje i ich konsekwencje [Rot06]

Nieporządkowy AG

Hierarchiczny AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie ewolucyjne

Ewolucja różnicowa

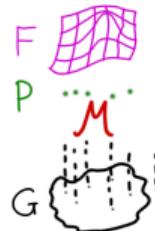
Programowanie ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp → fenotyp

Programowanie genetyczne

Bibliografia



- nadmiarowość: wiele genotypów → jeden fenotyp
 - sąsiednia (*synonymous*): genotypy tworzące ten sam fenotyp są sąsiadami
 - równoliczna każdy fenotyp powstaje z takiej samej liczby genotypów
 - różnoliczna: przeciwnie
 - odległa (*non-synonymous*): niekorzystne dla optymalizacji
- skalowanie alleli: na ile jednakowy jest wpływ alleli na fitness
- lokalność: podobieństwo (bliskość) genotypów skorelowane z podobieństwem odpowiadających im fenotypów
 - wysoka: dobrze! mapowanie nie zwiększa trudności problemu
 - niska: zwiększa trudność problemu

Powysze własności można oszacować liczbowo.

Embriogeneza – właściwości i zalety

Nieporządkowy AG

Hierarchiczny AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie ewolucyjne

Ewolucja różnicowa

Programowanie ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp → fenotyp

Programowanie genetyczne

Bibliografia

Możliwe powody skłaniające do stosowania nietrywialnego mapowania [Ben99]:

- ograniczenie (zmniejszenie) przestrzeni przeszukiwania (rekurencyjne, hierarchiczne, itd.),
- lepsze próbkowanie przestrzeni przeszukiwania (tworzące topologię zwiększającą FDC),
- bardziej złożone rozwiązania w przestrzeni fenotypów („procedura wzrostu” zapisana w genotypie),
- lepsza obsługa ograniczeń (mapowanie **każdego** genotypu na poprawny fenotyp),

oraz:

- kompresja: proste genotypy opisują skomplikowane fenotypy,
- powtarzanie: genotypy mogą opisywać symetrię, modularność, podprocedury, itd.,
- adaptacja: fenotypy mogą rozwijać się „adaptacyjnie” (żeby dopasować się do ograniczeń, albo żeby naprawiać błędy).

Embriogeneza – trudności

Nieporządkowy AG

Hierarchiczny AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie ewolucyjne

Ewolucja różnicowa

Programowanie ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp → fenotyp

Programowanie genetyczne

Bibliografia

- potrzeba dużego doświadczenia, aby ręcznie zdefiniować embriogenezę zapewniającą wymienione na poprzednim slajdzie zalety,
- trudno automatycznie wyewoluować embriogenę (potrzeba specjalizowanych operatorów zabezpieczających przed „puchnięciem” (ang. *bloat*) genotypów i fenotypów, epistazą i psuciem potomków przez operatory lub słabym dziedziczeniem informacji).

W większości zastosowań embriogeneza to niezmienne, zaprojektowane przez człowieka reguły odwzorowujące genotyp w jego znaczenie (*external embryogeny*). Więcej informacji – patrz późniejszy wykład o optymalizowaniu konstrukcji/ewolucyjnym projektowaniu.

Embriogeneza – ilustracja

Nieporządkowy AG

Hierarchiczny AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie ewolucyjne

Ewolucja różnicowa

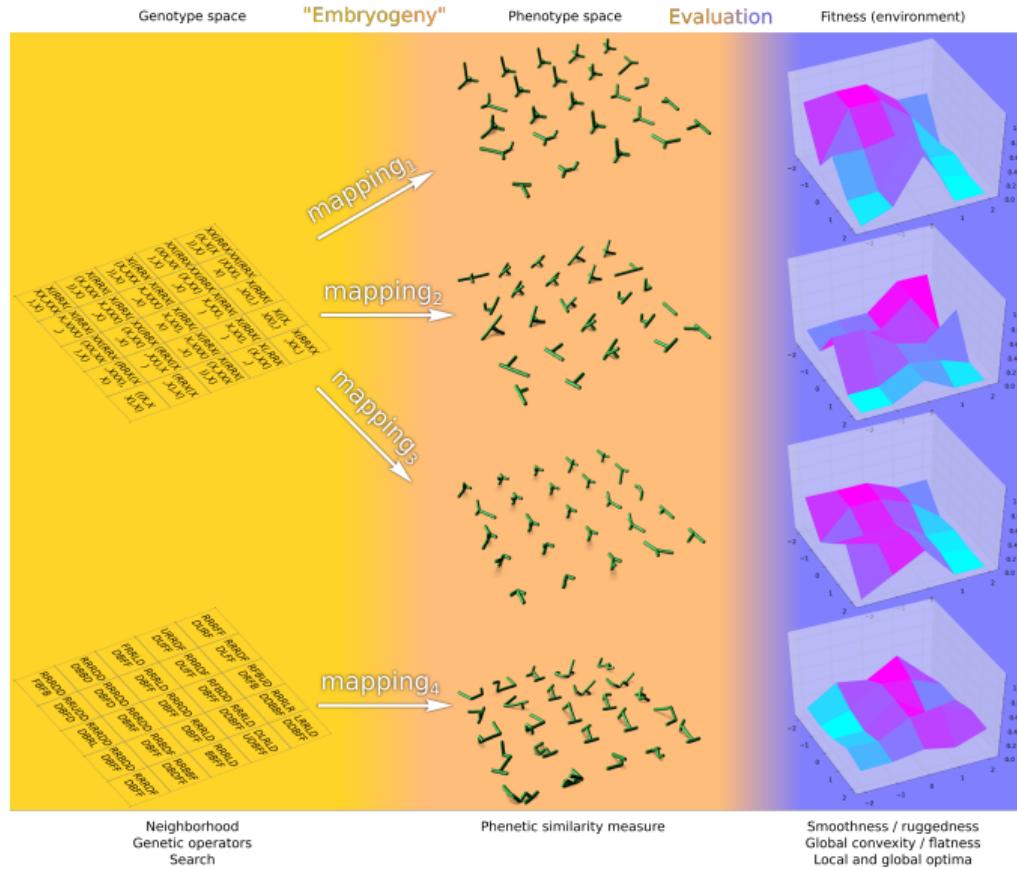
Programowanie ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp \rightarrow fenotyp

Programowanie genetyczne

Bibliografia



Embriogeneza – opis ilustracji

Nieporządkowy
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Schemat na poprzednim slajdzie pokazuje związki między przestrzenią genetyczną, fenetyczną, i krajobrazem przystosowania.

Zauważ, że różne embriogenezy (zatem też różne zbiory fenotypów, ich topologie i krajobrazy przystosowania) mogą być wynikiem:

- ① różnych reprezentacji genetycznych i przeznaczonych dla nich operatorów (na schemacie są pokazane dwie),
- ② różnych interpretacji (pokazano trzy) genów w ramach danej reprezentacji,
- ③ tej samej reprezentacji genetycznej i tej samej interpretacji genów, ale różnych operatorów mutacji/sąsiedztwa (schemat tej sytuacji nie pokazuje).

Programowanie genetyczne (*genetic programming*)

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

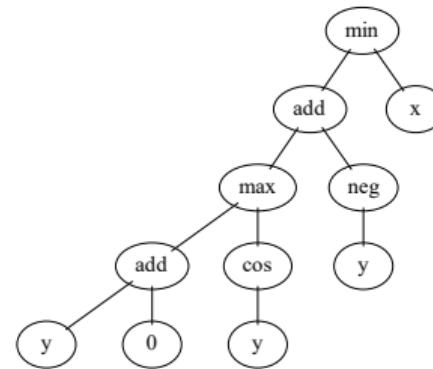
Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp \rightarrow
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Programowanie genetyczne* służy do optymalizacji wyrażeń i programów. Cechą charakterystyczną jest drzewiasta reprezentacja rozwiązań pozwalająca na zakodowanie programów, choć istnieje też mniej popularny wariant z kodowaniem liniowym.**



Rysunek: Wyrażenie $\min(\text{add}(\text{max}(\text{add}(y, 0), \cos(y)), \text{neg}(y)), x)$ czyli $\min(\max(y + 0, \cos(y)) + (-y), x)$ czyli $\min(x, \max(y, \cos(y)) - y)$.

*Darmowa książka:

http://www0.cs.ucl.ac.uk/staff/W.Langdon/ftp/papers/poli08_fieldguide.pdf

**https://en.wikipedia.org/wiki/Linear_genetic_programming

Programowanie genetyczne – budowa drzewa

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp \rightarrow
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Wyrażenia przechowywane w populacji składają się z elementów należących do zbioru funkcji **F** (węzły drzewa) i zbioru terminali **T** (liście drzewa). Zbiory te można dobrać wedle potrzeb i dostosować do rozwiązywanego problemu. Przestrzeń rozwiązań stanowią wszystkie kombinacje wyrażeń złożonych z elementów obu zbiorów.

Zbiór funkcji	
Rodzaj	Przykłady
Arytmetyczne	$+$, $*$, $/$
Matematyczne	\sin , \cos , \exp
Logiczne	AND, OR, NOT
Warunkowe	IF-THEN-ELSE
Pętle	FOR, REPEAT

Zbiór terminali	
Rodzaj	Przykłady
Zmienne	\vec{x} , y , $x172$
Stałe	3, 0.45, π
Procedury	rand, go_left, read_proximity

„Procedury” mogą być funkcjami lub akcjami bezargumentowymi.

Programowanie genetyczne – przykład użycia biblioteki DEAP

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

```
from deap import gp
# https://deap.readthedocs.io/en/master/tutorials/advanced/gp.html
# https://deap.readthedocs.io/en/master/examples/gp_symbreg.html

pset = gp.PrimitiveSet("MAIN", 2) # two arguments (x and y)
pset.addPrimitive(operator.add, 2)
pset.addPrimitive(operator.sub, 2)
pset.addPrimitive(operator.mul, 2)
pset.addPrimitive(operator.neg, 1)
pset.addPrimitive(min, 2)
pset.addPrimitive(max, 2)
pset.addPrimitive(math.cos, 1)
pset.addPrimitive(math.sin, 1)
pset.addEphemeralConstant("rand101", lambda: random.randint
    (-1,1))
pset.renameArguments(ARG0='x')
pset.renameArguments(ARG1='y')
```

Programowanie genetyczne – pożądane własności

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Pożądane są dwie cechy zbiorów F i T :

- ❶ *domknięcie* (ang. *closure*) – każda funkcja działa dla dowolnych wartości i typów argumentów zwracanych przez dowolną funkcję lub terminal,
- ❷ *wystarczalność* (ang. *sufficiency*) – dostępne w obu zbiorach elementy pozwalają na zbudowanie rozwiązania stawianego problemu.

Programowanie genetyczne – domknięcie

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Zastanów się, jak można zapewnić właściwość *domknięcia*.

Programowanie genetyczne – domknięcie

Nieporządky
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Zastanów się, jak można zapewnić właściwość *domknięcia*.

Właściwość *domknięcia* można uzyskać zabezpieczając odpowiednio funkcje (np. licząc zawsze wartość bezwzględną dla argumentu pierwiastka) albo karając niepoprawne wyrażenia (obniżając im wartość dopasowania). Albo ustawić flagi procesora/programu/systemu operacyjnego tak, żeby wszelkie operacje nie powodowały wyjątków... (tutaj wspomnienie długiej symulacji numerycznej pod linuxem i różnica tej samej symulacji pod Windows).

```
def protectedDiv(left, right):
    try:
        return left / right
    except ZeroDivisionError:
        return 1

pset.addPrimitive(protectedDiv, 2)
```

Programowanie genetyczne – wystarczalność

Nieporządky
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp \rightarrow
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Jeśli nie zapewnmy *wystarczalności*, GP będzie starało się znaleźć (najlepsze) przybliżenie rozwiązania za pomocą dostępnych środków.

Podstawowe metody tworzenia populacji początkowej

Nieporządny
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp \rightarrow
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Podstawowe metody tworzenia populacji początkowej

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp \rightarrow
fenotyp

Programo-
wanie
genetyczne

Bibliografia

- *Full*: wybieraj węzły z F jeśli głębokość jest poniżej ustalonego progu, a jeśli nie, to z T . Wszystkie drzewa będą miały taką samą głębokość – przykłady na kolejnym slajdzie.
- *Grow*: wybieraj węzły z $F \cup T$ jeśli głębokość jest poniżej ustalonego progu, a jeśli nie, to z T . Drzewa będą miały różną głębokość i kształt – przykłady na slajdzie następnym.
- *Ramped half-and-half*: połowa populacji metodą *full*, połowa metodą *grow* – zapewnia zróżnicowanie populacji początkowej.

Tworzenie osobników metodą *Full*

Nieporządky
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

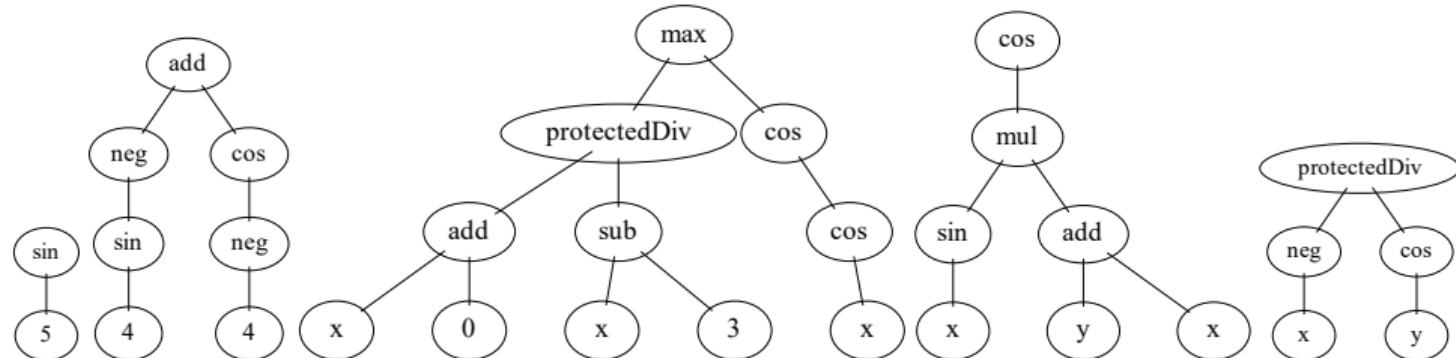
Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp \rightarrow
fenotyp

Programo-
wanie
genetyczne

Bibliografia



Rysunek: Pięć osobników wygenerowanych metodą *Full*, `gp.genFull(pset, 1, 3)` (DEAP wymaga dwóch parametrów, nie jednego) dla $T = \{x, y, 0, 1, 2, 3, 4, 5\}$.

Tworzenie osobników metodą *Grow*

Nieporządky
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

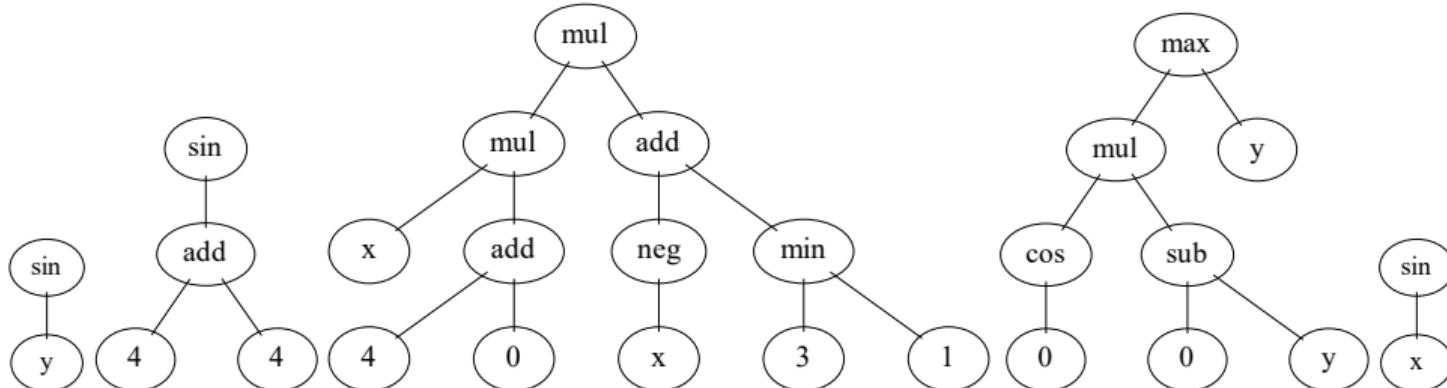
Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp \rightarrow
fenotyp

Programo-
wanie
genetyczne

Bibliografia



Rysunek: Pięć osobników wygenerowanych metodą *Grow*, `gp.genGrow(pset, 1, 3)`, dla $T = \{x, y, 0, 1, 2, 3, 4, 5\}$. W metodzie `genGrow()` DEAP'a nie ma sensu ustawienie argumentów `min_depth` i `max_depth` na taką samą wartość, bo generowane drzewa będą miały wtedy wszystkie liście na tej samej głębokości – tak samo, jakby drzewa generowała metoda `genFull()`.

Krzyżowanie

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Krzyżowanie

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste

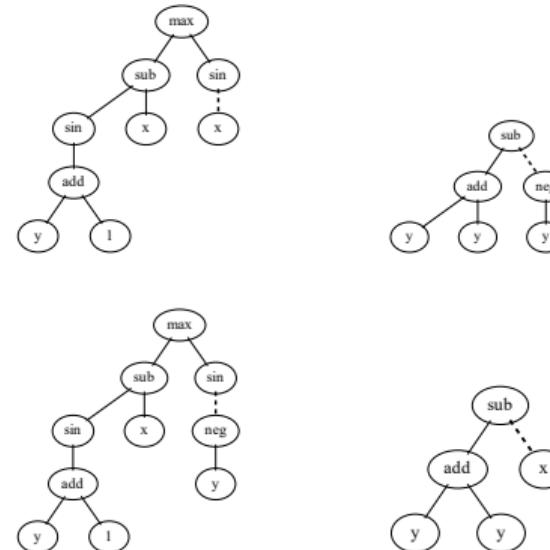
Mapowanie genotyp \rightarrow
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Krzyżowanie w GP to najczęściej wymiana losowo wybranych poddrzew u obu rodziców.

```
toolbox.register("mate", gp.cxOnePoint)
```



Rysunek: Krzyżowanie w GP. U góry rodzice wygenerowani metodą `gp.genGrow(pset, 2, 4)`. Na dole dzieci utworzone metodą `gp.cxOnePoint(parent1, parent2)`.

Mutacja

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Mutacja

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste

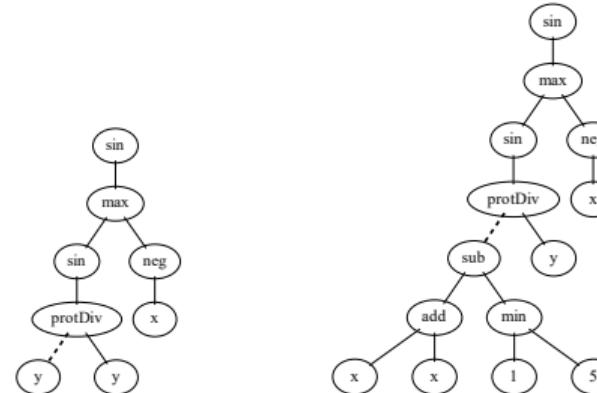
Mapowanie genotyp \rightarrow
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Typowa mutacja to wybranie losowego miejsca w oryginalnym drzewie i zastąpienie poddrzewa nowym, wygenerowanym jedną z powyżej opisanych metod.

```
toolbox.register("expr_mut", gp.genFull, min_=0, max_=2)
toolbox.register("mutate", gp.mutUniform, expr=toolbox.expr_mut,
                pset=pset)
```



Rysunek: Po lewej przodek wygenerowany metodą `gp.genGrow(pset, 2, 5)`. Po prawej mutant utworzony metodą `gp.mutUniform(parent, toolbox.expr_mut, pset=pset)`, przy wcześniejszym `toolbox.register("expr_mut", gp.genFull, min_=0, max_=2)`.

„Puchnięcie” rozwiązań

Nieporządny
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

„Puchnięcie” rozwiązań

Nieporządky
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Żeby zabezpieczyć się przed niekontrolowanym „puchnięciem” wyrażeń (ang. *bloat*), można włączyć do oceny rozwiązań kary za rozmiar wyrażenia lub zastosować ograniczenia na głębokość drzewa.

```
toolbox.decorate("mate", gp.staticLimit(key=operator.attrgetter(  
    "height"), max_value=13))  
toolbox.decorate("mutate", gp.staticLimit(key=operator.  
    attrgetter("height"), max_value=11))
```

Ponieważ wyrażenia lub programy generowane przez GP mają przypadkowy charakter, trudno byłoby je uruchamiać bezpośrednio w systemie operacyjnym – bezpieczniej jest je interpretować lub oceniać w wirtualnym środowisku (np. w maszynie wirtualnej albo „sandboxie”). Ocena jakości rozwiązania wymaga najczęściej jego wyliczenia lub użycia w wielu sytuacjach (różne wartości argumentów, różne lokalizacje robota, itp.).

```
# Exception: MemoryError - Error in tree evaluation: Python  
cannot evaluate a tree higher than 90.
```

Selekcja

Nieporządkowy AG

Hierarchiczny AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie ewolucyjne

Ewolucja różnicowa

Programowanie ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp \rightarrow fenotyp

Programowanie genetyczne

Bibliografia

Można stosować omówione wcześniej, standardowe metody selekcji, ale często lepszych wyników dostarcza selekcja Lexicase. W tej metodzie nie agreguje się błędów każdego rozwiązania na wszystkich testach do jednej liczby. Zamiast tego, aby wybrać jednego osobnika z populacji, najpierw losujemy kolejność testów, a potem wybieramy te osobniki, które na pierwszym teście (z tej losowej kolejności) uzyskały najlepszy wynik w populacji. Jeśli takich osobników jest więcej niż jeden, rozważamy wśród nich tak samo drugi test, potem ewentualnie trzeci, itd.

Dyskusja: jak takie podejście różni się od metod selekcji z ewolucyjnej optymalizacji wielokryterialnej takich jak NSGA czy SPEA?

Skuteczność GP

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Dyskusja: krajobraz przystosowania, globalna wypukłość i skuteczność optymalizacji w GP.

Regresja symboliczna

Nieporządkny
AG

Hierarchiczny
AG
Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp \rightarrow
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Regresja symboliczna to typowe zastosowanie GP, w którym szukamy funkcji opisującej możliwie dokładnie zadane punkty. O ile w tradycyjnych metodach regresji postać szukanej funkcji jest ustalona (poszukujemy tylko współczynników), o tyle w GP można łatwo manipulować postacią funkcji, a nawet szukać pewnych klas funkcji lub dowolnych funkcji (stąd ta metoda regresji nazywana jest *symboliczną*).

Sterowanie postacią szukanego wyrażenia odbywa się za pomocą odpowiedniego doboru elementów zbioru funkcji F i terminali T , oraz narzucaniem ewentualnych ograniczeń na głębokość drzewa, liczbę wystąpień funkcji ze zbioru F , itp.

Regresja symboliczna – szukamy $f(x) = x^2 - x$

Nieporządkowy
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp \rightarrow
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Przykładowy eksperyment #1: Znajdź wyrażenie najlepiej opisujące zbiór punktów należących do funkcji $f(x) = x^2 - x$. Pamiętajmy, że w praktyce funkcja ta jest nieznana i chcemy ją odkryć! Do dyspozycji GP dajemy to co w przykładowych źródłach powyżej, czyli oprócz x operatory neg, $+$, $-$, $*$, $/$, max, min, sin, cos, i dodatkowo też stałe $-1, 0, 1$.

Regresja symboliczna – przykład użycia biblioteki DEAP

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

```
def target_function(x):
    return x**2 - x # in a real application, this is what we
    look for!

def eval_expr(individual, points):
    # transform the tree expression into a callable function
    func = toolbox.compile(expr=individual)
    # evaluate the mean squared error between the expression and
    # the target function
    sqerrors = ((func(x) - target_function(x))**2 for x in points
    )
    return math.fsum(sqerrors) / len(points),

toolbox.register("evaluate", eval_expr, points=[x/10. for x in
    range(-10,11)])
```

Szukamy $f(x) = x^2 - x$. Pierwsze pokolenie

Nieporządkny AG

Hierarchiczny AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie ewolucyjne

Ewolucja różnicowa

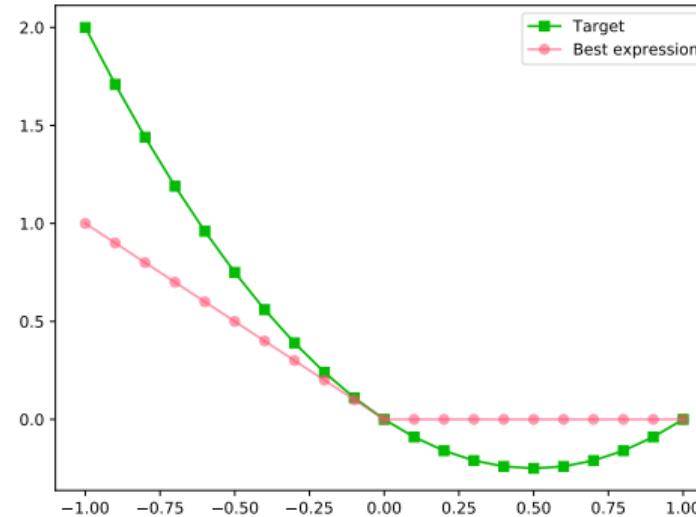
Programowanie ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp \rightarrow fenotyp

Programowanie genetyczne

Bibliografia



Rysunek: Rozwiążanie najlepsze w pierwszym pokoleniu (czyli w losowo wygenerowanej populacji).

`mul(min(0, x), neg(1))`

Szukamy $f(x) = x^2 - x$. Ostatnie pokolenie

Nieporządkny AG

Hierarchiczny AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie ewolucyjne

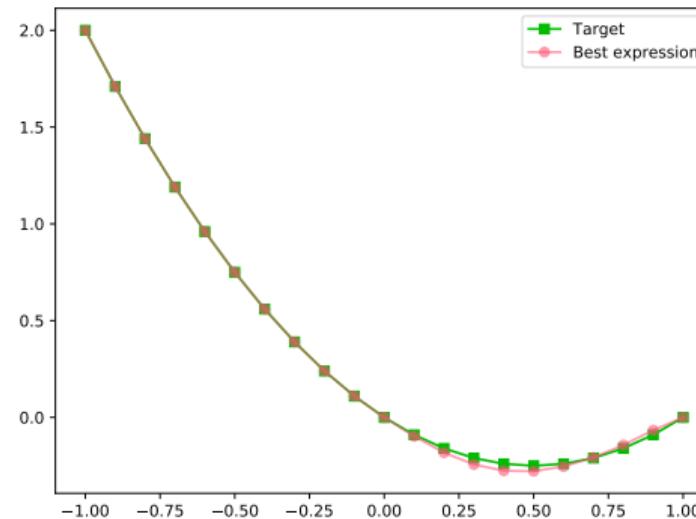
Ewolucja różnicowa

Programowanie ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp \rightarrow fenotyp

Programowanie genetyczne

Bibliografia



Rysunek: Rozwiążanie najlepsze po zakończeniu ewolucji.

Szukamy $f(x) = x^2 - x$. Ostatnie pokolenie

Nieporządkny AG

Hierarchiczny AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie ewolucyjne

Ewolucja różnicowa

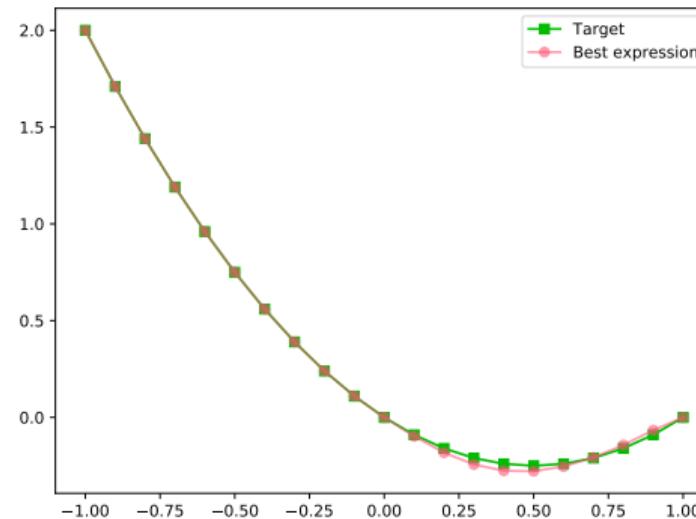
Programowanie ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp \rightarrow fenotyp

Programowanie genetyczne

Bibliografia



Rysunek: Rozwiążanie najlepsze po zakończeniu ewolucji.

```
sub(x, add(min(min(min(0, x), mul(0, add(0, max(1, 0)))),  
add(x, max(x, mul(add(0, x), neg(x)))), max(add(min(min(x, 0),  
add(min(sin(x), x), max(sin(x), add(add(0, 0), sin(sin(sin(x))))))),  
max(sin(add(min(sin(x), sin(sin(sin(x))))), max(sin(sin(x)), -1))),  
x)), x)))
```

Szukamy $f(x) = x^2 - x$. Inne parametry...

Nieporządky
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp \rightarrow
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Po zwiększeniu rozmiaru populacji i liczby pokoleń: `mul(add(-1, x), min(x, x))`.
Podobnie, po ograniczeniu złożoności wyrażeń (intensyfikacja przeszukiwania prostych wyrażeń): `mul(add(-1, x), protectedDiv(x, 1))`.

Regresja symboliczna – szukamy XOR

Nieporządky
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Przykładowy eksperyment #2: Znajdź układ logiczny realizujący funkcję XOR, czyli $\{x_1, x_2, y\} = \{(0, 0, 0); (0, 1, 1); (1, 0, 1); (1, 1, 0)\}$.

W tym eksperymencie GENERATIONS=100 oraz POPSIZE=150, a w przypadku niepowodzenia – kolejna próba z POPSIZE=1500.

Regresja symboliczna – szukamy XOR: funkcje i terminale

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

```
def nand(input1, input2):
    return not(input1 and input2)

def if_then_else(input, output1, output2):
    return output1 if input else output2

pset = gp.PrimitiveSetTyped("main", [bool, bool], bool) # let's
    use strongly-typed GP as an example
pset.addPrimitive(operator.xor, [bool, bool], bool)
pset.addPrimitive(operator.or_, [bool, bool], bool)
pset.addPrimitive(operator.and_, [bool, bool], bool)
pset.addPrimitive(operator.not_, [bool], bool)
pset.addPrimitive(nand, [bool, bool], bool) # custom
pset.addPrimitive(if_then_else, [bool, bool, bool], bool) #
    custom
pset.addTerminal(True, bool)

pset.renameArguments(ARG0="x1")
pset.renameArguments(ARG1="x2")
```

Regresja symboliczna – szukamy XOR: ocenianie

Nieporządky
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp \rightarrow
fenotyp

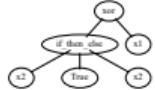
Programo-
wanie
genetyczne

Bibliografia

```
def eval_expr(individual):
    # transform the tree expression into a callable function
    func = toolbox.compile(expr=individual)
    # evaluate the error between the expression and the target
    # function
    err = 0
    for x1 in (False, True):
        for x2 in (False, True):
            target = x1 ^ x2
            actual = func(x1, x2)
            if target != actual:
                err += 1
    return err,
```

Regresja symboliczna – szukamy XOR: wyniki

- Wszystkie operatory oraz stała True jak w kodzie powyżej:
 $\text{xor}(\text{if_then_else}(x_2, \text{True}, x_2), x_1)$



Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste

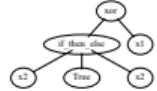
Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Regresja symboliczna – szukamy XOR: wyniki

- Wszystkie operatory oraz stała True jak w kodzie powyżej:
 $\text{xor}(\text{if_then_else}(x_2, \text{True}, x_2), x_1)$



- Samo if-then-else: brak idealnego rozwiązania (najmniejszy błąd = 1)

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

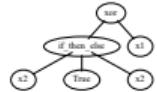
Liczby rzeczywiste
Mapowanie genotyp \rightarrow
fenotyp

Programo-
wanie
genetyczne

Bibliografia

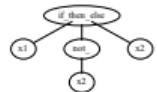
Regresja symboliczna – szukamy XOR: wyniki

- Wszystkie operatory oraz stała True jak w kodzie powyżej:
`xor(if_then_else(x2, True, x2), x1)`



- Samo if-then-else: brak idealnego rozwiązania (najmniejszy błąd = 1)
- Tylko if-then-else oraz not:

`if_then_else(x1, not_(x2), x2)`



Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste

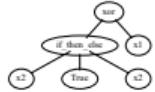
Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

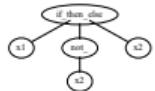
Regresja symboliczna – szukamy XOR: wyniki

- Wszystkie operatory oraz stała True jak w kodzie powyżej:
`xor(if_then_else(x2, True, x2), x1)`



- Samo if-then-else: brak idealnego rozwiązania (najmniejszy błąd = 1)
- Tylko if-then-else oraz not:

`if_then_else(x1, not_(x2), x2)`



- Tylko not oraz and: brak idealnego rozwiązania (najmniejszy błąd = 1)

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Regresja symboliczna – szukamy XOR: wyniki

Nieporządky
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

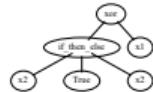
Liczby rzeczywiste
Mapowanie genotyp \rightarrow fenotyp

Programo-
wanie
genetyczne

Bibliografia

- Wszystkie operatory oraz stała True jak w kodzie powyżej:

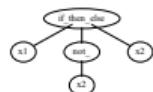
$\text{xor}(\text{if_then_else}(x_2, \text{True}, x_2), x_1)$



- Samo if-then-else: brak idealnego rozwiązania (najmniejszy błąd = 1)

- Tylko if-then-else oraz not:

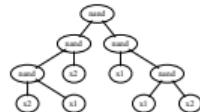
$\text{if_then_else}(x_1, \text{not_}(x_2), x_2)$



- Tylko not oraz and: brak idealnego rozwiązania (najmniejszy błąd = 1)

- Samo nand:

$\text{nand}(\text{nand}(\text{nand}(x_2, x_1), x_2), \text{nand}(x_1, \text{nand}(x_1, x_2)))$



Regresja symboliczna – szukamy XOR: wyniki

Nieporządky
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

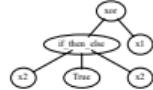
Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp \rightarrow fenotyp

Programo-
wanie
genetyczne

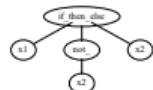
Bibliografia

- Wszystkie operatory oraz stała True jak w kodzie powyżej:
 $\text{xor}(\text{if_then_else}(x_2, \text{True}, x_2), x_1)$



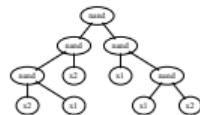
- Samo if-then-else: brak idealnego rozwiązania (najmniejszy błąd = 1)
- Tylko if-then-else oraz not:

$\text{if_then_else}(x_1, \text{not_}(x_2), x_2)$

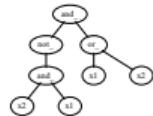


- Tylko not oraz and: brak idealnego rozwiązania (najmniejszy błąd = 1)
- Samo nand:

$\text{nand}(\text{nand}(\text{nand}(x_2, x_1), x_2), \text{nand}(x_1, \text{nand}(x_1, x_2)))$



- Trójka and, or, not:
 $\text{and_}(\text{not_}(\text{and_}(x_2, x_1)), \text{or_}(x_1, x_2))$



Programowanie genetyczne – konkluzje i dyskusja

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Dyskusja: czy warto stosować upraszczanie wyrażeń podczas ewolucji?

Programowanie genetyczne – konkluzje i dyskusja

Nieporządny
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Dyskusja: czy warto stosować upraszczanie wyrażeń podczas ewolucji?

Dyskusja: w jakich dziedzinach GP ma szansę konkurować z człowiekiem, w jakich go przewyższyć, a w jakich nie ma szans? Dlaczego?

Programowanie genetyczne – polepszanie skuteczności

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp \rightarrow
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Semantyczne GP (semantyka = zbiór efektów działania osobnika na zbiorze testów) oraz geometryczne semantyczne GP (operatorzy genetyczni uwzględniają topologię przestrzeni semantyk) [Bak+19].

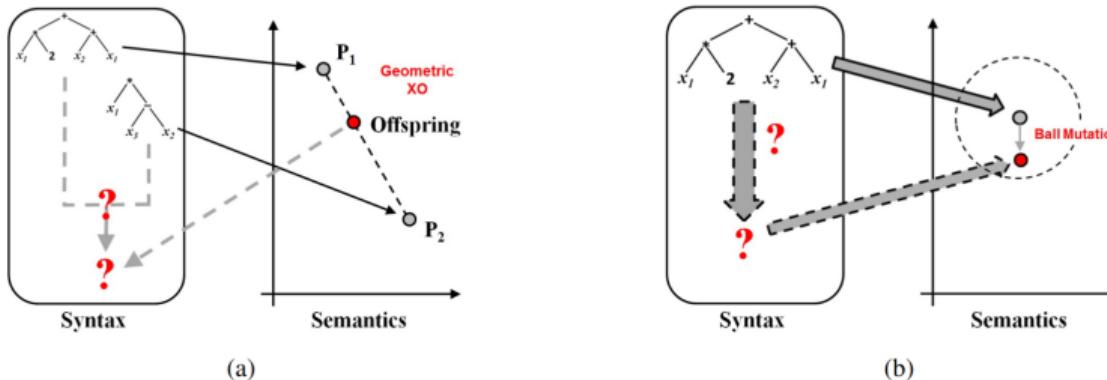


Figure 1: Geometric semantic crossover (plot (a)) (respectively geometric semantic mutation (plot (b))) performs a transformation on the syntax of the individual that corresponds to geometric crossover (respectively geometric mutation) on the semantic space. In this figure, the unrealistic case of a bidimensional semantic space is considered, for simplicity.

Por. wcześniejsze przypomnienie FDC, DPX, "Jak świadomie tworzyć (projektować) skuteczne operatorzy krzyżowania?" oraz embriogenezy/mapowania.

Szukamy algorytmu uczenia sieci neuronowej

Przykładowy eksperyment #3: Znajdź algorytm uczenia sieci neuronowej...

Architektura ewolucyjna: „ewolucja regulatoryzowana” (Rys. 2) [Rea+20].

Odkrycia ewolucji – Rys. 6:

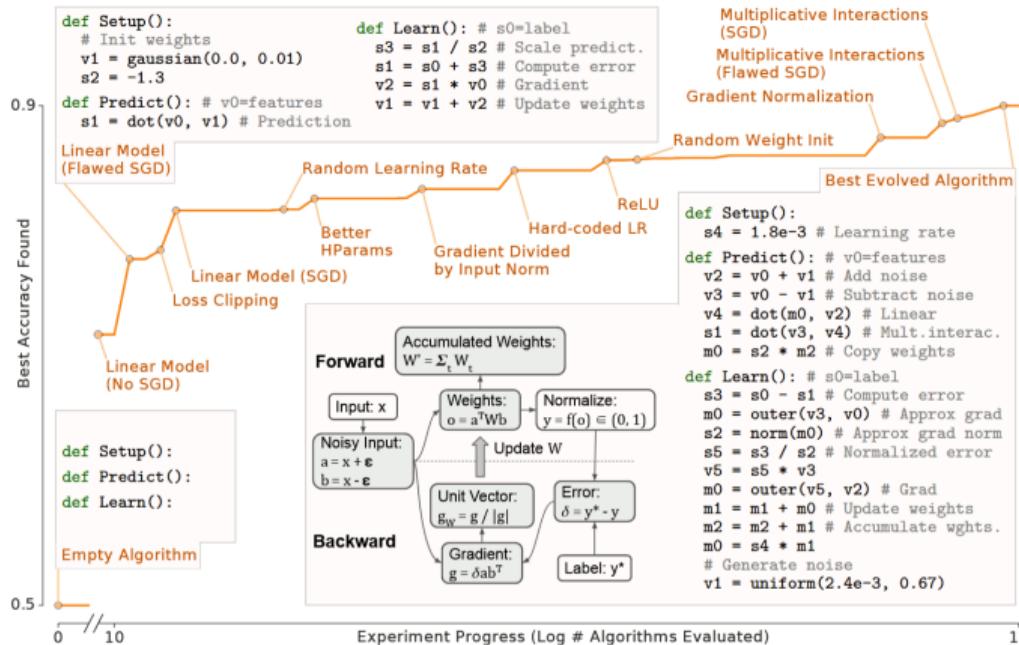


Figure 6: Progress of one evolution experiment on projected binary CIFAR-10. Callouts indicate some beneficial discoveries. We also print the code for the initial, an intermediate, and the final algorithm. The last is explained in the flow diagram. It outperforms a simple

Nieporządkowy AG

Hierarchiczny AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie ewolucyjne

Ewolucja różnicowa

Programowanie ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp → fenotyp

Programowanie genetyczne

Bibliografia

Hiperheurystyki i samo-programujące się algorytmy

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

*<http://en.wikipedia.org/wiki/Hyper-heuristic>

Hiperheurystyki i samo-programujące się algorytmy

Nieporządny
AG

Hierarchiczny
AG
Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne
Liczby rzeczywiste
Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Struktura algorytmu ewolucyjnego (rodzaj selekcji, krzyżowania, mutacji, ...) może podlegać kontroli GP (czyli ewolucyjnemu ulepszaniu) [Wąs97; BT96; OG03; Olt05]. GP pozwala na „zbudowanie” algorytmu optymalizacji z modułów, przy czym możliwe są nietypowe architektury: wiele rodzajów mutacji, nietypowe operatory działające na części populacji, wielokrotna selekcja w jednym kroku itp. – zależnie od stopni swobody GP.

Wyniki – lepsze od tradycyjnego, ustalonego AE, ale kosztem...

Porównaj: twierdzenie *No Free Lunch* oraz hiperheurystyki* przeszukujące przestrzeń heurystyk i ich kombinacji [Ros05; ÖBK08; Bur+10].

*<http://en.wikipedia.org/wiki/Hyper-heuristic>

A gdyby tak...

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED

Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste

Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

Jeśli masz czas i lubisz SF, przeczytaj

<https://www.teamten.com/lawrence/writings/coding-machines/>.

Bibliografia |

Nieporządkny
AG

Hierarchiczny
AG
Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne
Liczby rzeczywiste
Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

- [Bak+19] Illya Bakurov i in. "A regression-like classification system for geometric semantic genetic programming". W: *Proceedings of the 11th International Joint Conference on Computational Intelligence (IJCCI)*. T. 1. 2019, s. 40–48. URL: [https://run.unl.pt/bitstream/10362/87064/1/Regression_like_Classification_System_Geometric.pdf](https://run.unl.pt/bitstream/10362/87064/1/Regression_like_Classification_System_Geometric_Semantic_Genetic.pdf).
- [Ben99] Peter Bentley. *Evolutionary design by computers*. Morgan Kaufmann, 1999.
- [BT96] Andreas Bölte i Ulrich Wilhelm Thonemann. "Optimizing simulated annealing schedules with genetic programming". W: *European Journal of Operational Research* 92.2 (1996), s. 402–416. ISSN: 0377-2217. DOI: [10.1016/0377-2217\(94\)00350-5](https://doi.org/10.1016/0377-2217(94)00350-5).
- [Bul01] Seth Bullock. "Smooth operator? Understanding and visualising mutation bias". W: *European Conference on Artificial Life*. Springer. 2001, s. 602–612. DOI: [10.1007/3-540-44811-X_68](https://doi.org/10.1007/3-540-44811-X_68).
- [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](https://doi.org/10.1007/3-540-48304-7_11). URL: <https://eprints.soton.ac.uk/261452/1/10.1.1.40.2753.pdf>.
- [Bur+10] E. K. Burke i in. "A classification of hyper-heuristic approaches". W: *Handbook of Metaheuristics* (2010), s. 449–468.
- [Gol+93] David E. Goldberg i in. *Rapid, Accurate Optimization of Difficult Problems Using Fast Messy Genetic Algorithms*. Spraw. tech. 93004. 1993, s. 1–16. URL: <http://repository.ias.ac.in/81677/1/110-a.pdf>.
- [GT12] Brian W. Goldman i Daniel R. Tauritz. "Linkage tree genetic algorithms: variants and analysis". W: *Proceedings of the 14th annual conference on Genetic and evolutionary computation*. 2012, s. 625–632. URL: <http://www.cmap.polytechnique.fr/~nikolaus.hansen/proceedings/2012/GECCO/proceedings/p625.pdf>.

Bibliografia II

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

- [Gwi07a] Tomasz Dominik Gwiazda. *Algorytmy genetyczne – kompendium. Tom I. Operator krzyżowania dla problemów numerycznych*. PWN, 2007.
- [Gwi07b] Tomasz Dominik Gwiazda. *Algorytmy genetyczne – kompendium. Tom II. Operator mutacji dla problemów numerycznych*. PWN, 2007.
- [JTW04] E. D. de Jong, D. Thierens i R. A. Watson. "Hierarchical genetic algorithms". W: *Lecture notes in computer science* (2004), s. 232–241. URL: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=20bde9fe796a5e15c9007c7dc770b35aa731751d>.
- [Mic96] Z. Michalewicz. *Algorytmy genetyczne + struktury danych = programy ewolucyjne*. Wydawnictwa Naukowo-Techniczne, 1996.
- [ÖBK08] E. Özcan, B. Bilgin i E. E. Korkmaz. "A comprehensive analysis of hyper-heuristics". W: *Intelligent Data Analysis* 12.1 (2008), s. 3–23.
- [OG03] Mihai Oltean i Crina Grosan. "Evolving evolutionary algorithms using multi expression programming". W: *European Conference on Artificial Life*. Springer. 2003, s. 651–658. URL: https://www.researchgate.net/profile/Mihai_Oltean2/publication/226167912_Evolving_Evolutionary_Algorithms_Using_Multi_Expression_Programming/links/55dac32308aed6a199aaaf916.pdf.
- [Olt05] Mihai Oltean. "Evolving evolutionary algorithms using linear genetic programming". W: *Evolutionary Computation* 13.3 (2005), s. 387–410. URL: https://mihaioltean.github.io/oltean_mit_draft_2005.pdf.
- [PKF21] Michał W. Przewozniczek, Marcin M. Komarnicki i Bartosz Frej. "Direct linkage discovery with empirical linkage learning". W: *Proceedings of the Genetic and Evolutionary Computation Conference*. 2021, s. 609–617. URL: <https://www.cs.put.poznan.pl/mkomosinski/lectures/optimization/extras/DLED-epistasis-GECCO2021.pdf>.

Bibliografia III

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

Bibliografia

- [PTK23] Michal W. Przewozniczek, Renato Tinós i Marcin M. Komarnicki. "First Improvement Hill Climber with Linkage Learning – on Introducing Dark Gray-Box Optimization into Statistical Linkage Learning Genetic Algorithms". W: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '23*. ACM, 2023, s. 946–954. DOI: [10.1145/3583131.3590495](https://doi.org/10.1145/3583131.3590495).
- [Rea+20] Esteban Real i in. "AutoML-Zero: Evolving machine learning algorithms from scratch". W: *International Conference on Machine Learning*. PMLR, 2020, s. 8007–8019. URL: <https://proceedings.mlr.press/v119/real20a/real20a.pdf>.
- [Rec84] Ingo Rechenberg. "The Evolution Strategy. A Mathematical Model of Darwinian Evolution". W: *Synergetics – From Microscopic to Macroscopic Order*. Red. Eckart Frehland. Berlin, Heidelberg: Springer Berlin Heidelberg, 1984, s. 122–132. DOI: [10.1007/978-3-642-69540-7_13](https://doi.org/10.1007/978-3-642-69540-7_13).
- [Ros05] P. Ross. "Hyper-heuristics". W: *Search Methodologies* (2005), s. 529–556.
- [Rot06] Franz Rothlauf. *Representations for genetic and evolutionary algorithms*. Springer, 2006. DOI: [10.1007/3-540-32444-5](https://doi.org/10.1007/3-540-32444-5).
- [SP97] Rainer Storn i Kenneth Price. "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces". W: *Journal of Global Optimization* 11.4 (1997), s. 341–359. ISSN: 1573-2916. DOI: [10.1023/A:1008202821328](https://doi.org/10.1023/A:1008202821328).
- [TB13] Dirk Thierens i Peter A. N. Bosman. "Hierarchical problem solving with the linkage tree genetic algorithm". W: *Proceedings of the 15th annual conference on Genetic and evolutionary computation*. 2013, s. 877–884. URL: https://homepages.cwi.nl/~bosman/publications/2013_hierarchicalproblemsolving.pdf.
- [Thi18] Dirk Thierens. *Model-Based Evolutionary Algorithms, Part 2: Linkage Tree Genetic Algorithm*. 2018. URL: https://ics-websites.science.uu.nl/docs/vakken/ea/slides/LTGA_GOMEA.pdf.

Bibliografia IV

Nieporządkny
AG

Hierarchiczny
AG

Epistaza – DLED
Epistaza – przykład ciągły

Strategie
ewolucyjne

Ewolucja
różnicowa

Programo-
wanie
ewolucyjne

Liczby rzeczywiste
Mapowanie genotyp →
fenotyp

Programo-
wanie
genetyczne

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.

[Wąs97]

Piotr Wąsiewicz. "Self-programming of Algorithms". W: *Proceedings of the 2nd National Conference on Evolutionary Computation and Global Optimization*. Rytro: Oficyna Wydawnicza Politechniki Warszawskiej, wrz. 1997, s. 293–297.



**Fundusze
Europejskie**
Polska Cyfrowa



**Rzeczpospolita
Polska**

Unia Europejska
Europejski Fundusz
Rozwoju Regionalnego



Algorytmy i modele inspirowane biologicznie

5. Sterowanie różnorodnością

Maciej Komosiński

Standardowe techniki selekcji

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform

Selection Scheme

FUDS: Fitness Uniform
Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna
vs. lokalna

NSLC: Novelty Search with
Local Competition

Przykładowy eksperyment porównawczy

Bibliografia

● Kiedy są skuteczne?

- Kiedy szukamy lokalnych optimów
- Kiedy zachodzi globalna wypukłość
- W niezbyt chropowatym krajobrazie przystosowania
- W niezwodniczym krajobrazie przystosowania

Standardowe techniki selekcji

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform

Selection Scheme

FUDS: Fitness Uniform
Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna
vs. lokalna

NSLC: Novelty Search with
Local Competition

Przykładowy eksperyment porównawczy

Bibliografia

● Kiedy są skuteczne?

- Kiedy szukamy lokalnych optimów
- Kiedy zachodzi globalna wypukłość
- W niezbyt chropowatym krajobrazie przystosowania
- W niezwodniczym krajobrazie przystosowania

● Kiedy nie są szczególnie skuteczne?

Standardowe techniki selekcji

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

Bibliografia

- Kiedy są skuteczne?
 - Kiedy szukamy lokalnych optimów
 - Kiedy zachodzi globalna wypukłość
 - W niezbyt chropowatym krajobrazie przystosowania
 - W niezwodniczym krajobrazie przystosowania
- Kiedy nie są szczególnie skuteczne?
- Jak można je usprawnić?

Sterowanie różnorodnością – przypomnienie

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Poznamy teraz nowoczesne, efektywne metody sterowania różnorodnością stosowane w algorytmach populacyjnych. Ich zastosowanie jest uzasadnione w problemach, w których standardowe techniki selekcji nie są wystarczająco skuteczne (poprzedni slajd). Tę grupę metod nazywa się zwykle *quality-diversity*.

Przypomnij sobie poznane do tej pory metody wymuszania/utrzymywania różnorodności w wartościach dopasowania rozwiązań.

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

Bibliografia

Sterowanie różnorodnością – przypomnienie

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

Poznamy teraz nowoczesne, efektywne metody sterowania różnorodnością stosowane w algorytmach populacyjnych. Ich zastosowanie jest uzasadnione w problemach, w których standardowe techniki selekcji nie są wystarczająco skuteczne (poprzedni slajd). Tę grupę metod nazywa się zwykle *quality-diversity*.

Przypomnij sobie poznane do tej pory metody wymuszania/utrzymywania różnorodności w wartościach dopasowania rozwiązań.

- selekcja konwekcyjna (prezentacja #1, meta-schematy selekcji)

Przypomnij sobie poznane do tej pory metody wymuszania/utrzymywania różnorodności w treści rozwiązań.

Bibliografia

Sterowanie różnorodnością – przypomnienie

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

Bibliografia

Poznamy teraz nowoczesne, efektywne metody sterowania różnorodnością stosowane w algorytmach populacyjnych. Ich zastosowanie jest uzasadnione w problemach, w których standardowe techniki selekcji nie są wystarczająco skuteczne (poprzedni slajd). Tę grupę metod nazywa się zwykle *quality-diversity*.

Przypomnij sobie poznane do tej pory metody wymuszania/utrzymywania różnorodności w wartościach dopasowania rozwiązań.

- selekcja konwekcyjna (prezentacja #1, meta-schematy selekcji)

Przypomnij sobie poznane do tej pory metody wymuszania/utrzymywania różnorodności w treści rozwiązań.

- model ze współczynnikiem zatłoczenia (prezentacja #1, selekcja – dodatkowe własności)
- dyskusja: w jaki sposób uzyskał(a)byś stabilnie utrzymujące się pokrycie osobnikami optimów lokalnych, z licznością osobników proporcjonalną do jakości optimów?

Sterowanie różnorodnością – przypomnienie

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

Bibliografia

Poznamy teraz nowoczesne, efektywne metody sterowania różnorodnością stosowane w algorytmach populacyjnych. Ich zastosowanie jest uzasadnione w problemach, w których standardowe techniki selekcji nie są wystarczająco skuteczne (poprzedni slajd). Tę grupę metod nazywa się zwykle *quality-diversity*.

Przypomnij sobie poznane do tej pory metody wymuszania/utrzymywania różnorodności w wartościach dopasowania rozwiązań.

- selekcja konwekcyjna (prezentacja #1, meta-schematy selekcji)

Przypomnij sobie poznane do tej pory metody wymuszania/utrzymywania różnorodności w treści rozwiązań.

- model ze współczynnikiem zatłoczenia (prezentacja #1, selekcja – dodatkowe własności)
- dyskusja: w jaki sposób uzyskał(a)byś stabilnie utrzymujące się pokrycie osobnikami optimów lokalnych, z licznością osobników proporcjonalną do jakości optimów? → „niszowanie”

Sterowanie różnorodnością wartości funkcji celu

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

Bibliografia

W kolejności złożoności:

- FUSS
- FUDS
- Selekcja konwekcyjna – już poznana
- HFC
- Nieco inne założenia: MAP-Elites (wymaga pomocniczych funkcji oceniających)

FUSS (Fitness Uniform Selection Scheme)

- Wyznacz najniższe i najwyższe wartości przystosowania w populacji: f_{min} , f_{max} .
- Wybierz wartość przystosowania f losowo z równym prawdopodobieństwem z zakresu $[f_{min}, f_{max}]$.
- Osobnikiem wybranym zostaje ten z przystosowaniem najbliższym f [HL06].

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Niszowanie

Pozyskiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

FUSS (Fitness Uniform Selection Scheme)

- Wyznacz najniższe i najwyższe wartości przystosowania w populacji: f_{min} , f_{max} .
- Wybierz wartość przystosowania f losowo z równym prawdopodobieństwem z zakresu $[f_{min}, f_{max}]$.
- Osobnikiem wybranym zostaje ten z przystosowaniem najbliższym f [HL06].

- Nie wywiera bezpośrednio presji selekcyjnej.
- Ponieważ lepsze rozwiązania trudniej odkryć, wywiera presję selekcyjną niejawnie.

FUSS – prawdopodobieństwo wybrania rozwiązania

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

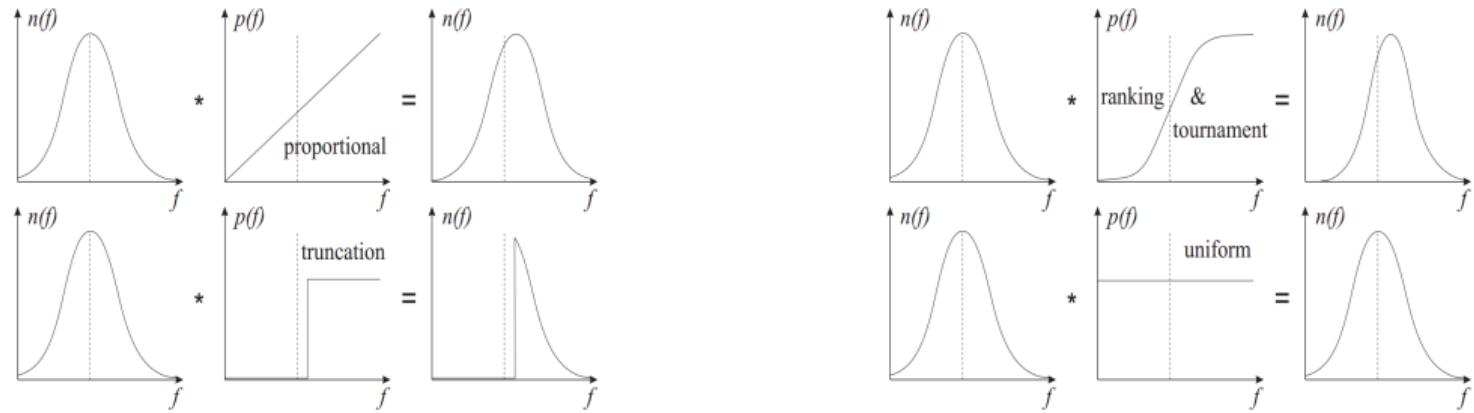
Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy



Bibliografia

FUSS – prawdopodobieństwo wybrania rozwiązania

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

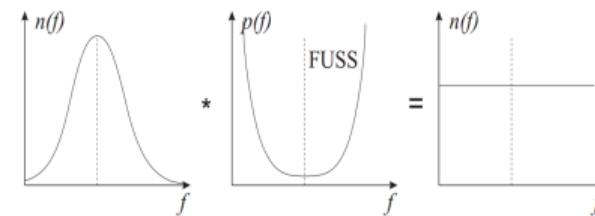
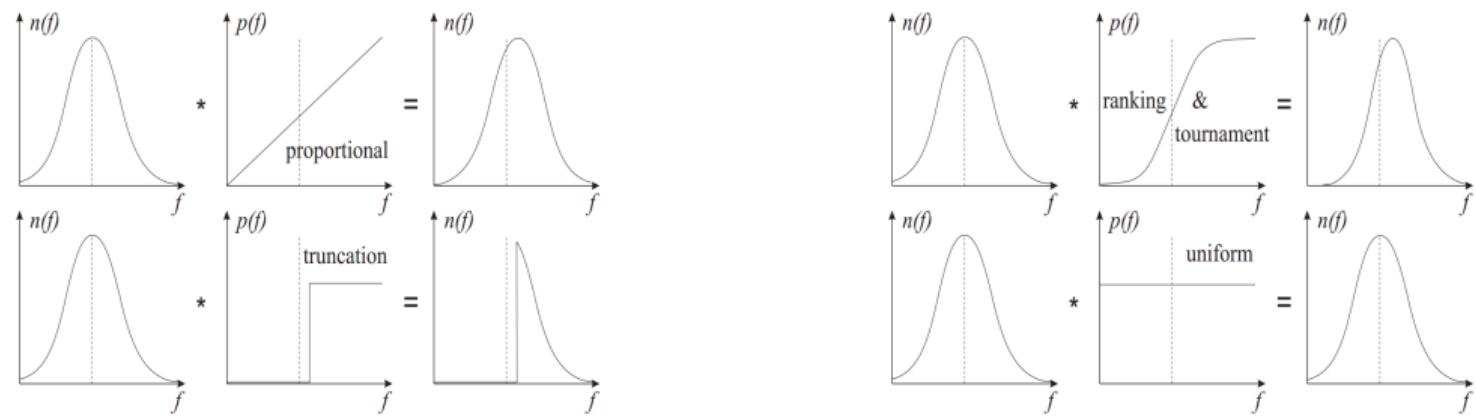
Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy



Rysunek: Prawdopodobieństwo wybrania $p(f)$ w zależności od jakości rozwiązania dla różnych metod selekcji, i oczekiwany rozkład jakości rozwiązań w kolejnym pokoleniu $n(f)$ [HL06].

Bibliografia

FUSS – działanie

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

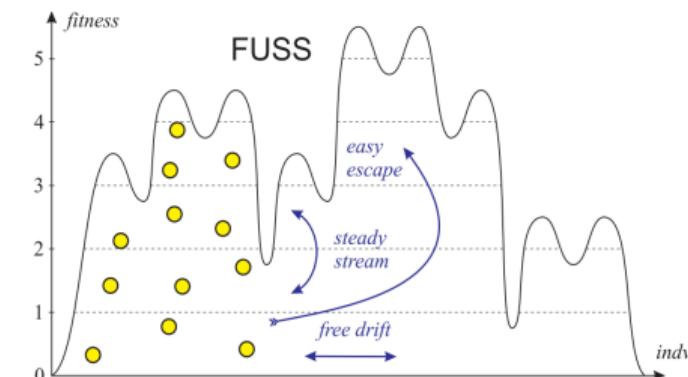
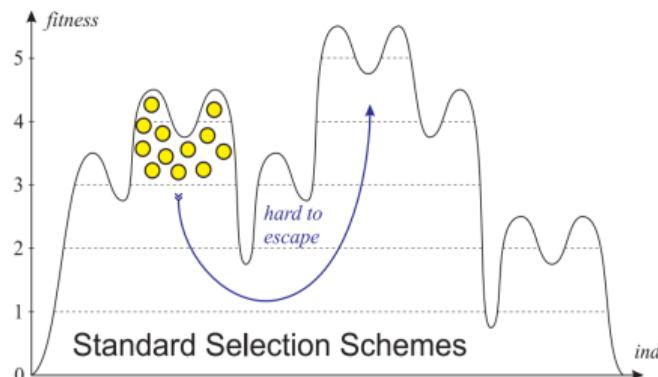
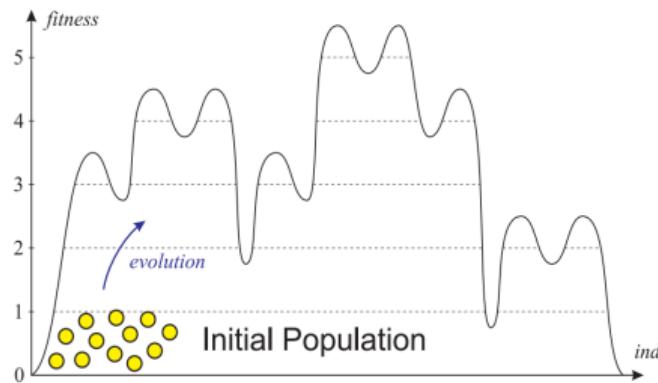
Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

Bibliografia



Rysunek: Wymuszone „pionowe” rozpraszanie pomaga w ciągłym odkrywaniu nowych optimów [HL06].

Właściwości FUSS

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

Bibliografia

- FUSS skutecznie odkrywa (także) dobre osobniki – z tego samego powodu, co dyskutowane wcześniej: dlaczego selekcja konwekcyjna typu *EqualWidth* z losową selekcją w podpopulacjach nadal potrafi optymalizować?

* „Przejęcie” oznacza, że najlepszy osobnik kolonizuje i zajmuje całą populację. „Czas przejęcia”: zaczynając od populacji z jednym unikatowym najlepszym osobnikiem (pozostałe są gorsze), ilu iteracji danej metody selekcji oczekujemy, aby populacja stała się jednorodna i składała się tylko z kopii najlepszego osobnika? Ta pojedyncza wartość może być wykorzystana do porównania presji selekcyjnych różnych technik selekcji.

Własności FUSS

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

Bibliografia

- FUSS skutecznie odkrywa (także) dobre osobniki – z tego samego powodu, co dyskutowane wcześniej: dlaczego selekcja konwekcyjna typu *EqualWidth* z losową selekcją w podpopulacjach nadal potrafi optymalizować?
- Brak problemu dominacji („przejęcia” – *takeover*)* – czyli brak przedwczesnej zbieżności, a raczej brak w ogóle zbieżności!

* „Przejęcie” oznacza, że najlepszy osobnik kolonizuje i zajmuje całą populację. „Czas przejęcia”: zaczynając od populacji z jednym unikatowym najlepszym osobnikiem (pozostałe są gorsze), ilu iteracji danej metody selekcji oczekujemy, aby populacja stała się jednorodna i składała się tylko z kopii najlepszego osobnika? Ta pojedyncza wartość może być wykorzystana do porównania presji selekcyjnych różnych technik selekcji.

Własności FUSS

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

Bibliografia

- FUSS skutecznie odkrywa (także) dobre osobniki – z tego samego powodu, co dyskutowane wcześniej: dlaczego selekcja konwekcyjna typu *EqualWidth* z losową selekcją w podpopulacjach nadal potrafi optymalizować?
- Brak problemu dominacji („przejęcia” – *takeover*)* – czyli brak przedwczesnej zbieżności, a raczej brak w ogóle zbieżności!
- Faworyzuje osobniki o rzadkich wartościach przystosowania (w porównaniu z często występującymi wartościami przystosowania) – może faworyzować osobniki słabe, jeśli są rzadkie.

* „Przejęcie” oznacza, że najlepszy osobnik kolonizuje i zajmuje całą populację. „Czas przejęcia”: zaczynając od populacji z jednym unikatowym najlepszym osobnikiem (pozostałe są gorsze), ilu iteracji danej metody selekcji oczekujemy, aby populacja stała się jednorodna i składała się tylko z kopii najlepszego osobnika? Ta pojedyncza wartość może być wykorzystana do porównania presji selekcyjnych różnych technik selekcji.

Własności FUSS

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

Bibliografia

- FUSS skutecznie odkrywa (także) dobre osobniki – z tego samego powodu, co dyskutowane wcześniej: dlaczego selekcja konwekcyjna typu *EqualWidth* z losową selekcją w podpopulacjach nadal potrafi optymalizować?
- Brak problemu dominacji („przejęcia” – *takeover*)* – czyli brak przedwcześnie zbieżności, a raczej brak w ogóle zbieżności!
- Faworyzuje osobniki o rzadkich wartościach przystosowania (w porównaniu z często występującymi wartościami przystosowania) – może faworyzować osobniki słabe, jeśli są rzadkie.
- Swobodny dryf działa na różnych poziomach przystosowania i sprzyja różnorodności.

* „Przejęcie” oznacza, że najlepszy osobnik kolonizuje i zajmuje całą populację. „Czas przejęcia”: zaczynając od populacji z jednym unikatowym najlepszym osobnikiem (pozostałe są gorsze), ilu iteracji danej metody selekcji oczekujemy, aby populacja stała się jednorodna i składała się tylko z kopii najlepszego osobnika? Ta pojedyncza wartość może być wykorzystana do porównania presji selekcyjnych różnych technik selekcji.

Własności FUSS

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

Bibliografia

- FUSS skutecznie odkrywa (także) dobre osobniki – z tego samego powodu, co dyskutowane wcześniej: dlaczego selekcja konwekcyjna typu *EqualWidth* z losową selekcją w podpopulacjach nadal potrafi optymalizować?
- Brak problemu dominacji („przejęcia” – *takeover*)* – czyli brak przedwcześnie zbieżności, a raczej brak w ogóle zbieżności!
- Faworyzuje osobniki o rzadkich wartościach przystosowania (w porównaniu z często występującymi wartościami przystosowania) – może faworyzować osobniki słabe, jeśli są rzadkie.
- Swobodny dryf działa na różnych poziomach przystosowania i sprzyja różnorodności.
- Metoda samobalansująca się i pozbawiona parametrów.

* „Przejęcie” oznacza, że najlepszy osobnik kolonizuje i zajmuje całą populację. „Czas przejęcia”: zaczynając od populacji z jednym unikatowym najlepszym osobnikiem (pozostałe są gorsze), ilu iteracji danej metody selekcji oczekujemy, aby populacja stała się jednorodna i składała się tylko z kopii najlepszego osobnika? Ta pojedyncza wartość może być wykorzystana do porównania presji selekcyjnych różnych technik selekcji.

Własności FUSS

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

Bibliografia

- FUSS skutecznie odkrywa (także) dobre osobniki – z tego samego powodu, co dyskutowane wcześniej: dlaczego selekcja konwekcyjna typu *EqualWidth* z losową selekcją w podpopulacjach nadal potrafi optymalizować?
- Brak problemu dominacji („przejęcia” – *takeover*)* – czyli brak przedwczesnej zbieżności, a raczej brak w ogóle zbieżności!
- Faworyzuje osobniki o rzadkich wartościach przystosowania (w porównaniu z często występującymi wartościami przystosowania) – może faworyzować osobniki słabe, jeśli są rzadkie.
- Swobodny dryf działa na różnych poziomach przystosowania i sprzyja różnorodności.
- Metoda samobalansująca się i pozbawiona parametrów.
- Prosta implementacja i niski koszt obliczeniowy.

* „Przejęcie” oznacza, że najlepszy osobnik kolonizuje i zajmuje całą populację. „Czas przejęcia”: zaczynając od populacji z jednym unikatowym najlepszym osobnikiem (pozostałe są gorsze), ilu iteracji danej metody selekcji oczekujemy, aby populacja stała się jednorodna i składała się tylko z kopii najlepszego osobnika? Ta pojedyncza wartość może być wykorzystana do porównania presji selekcyjnych różnych technik selekcji.

FUDS – logika

FUDS (Fitness Uniform Deletion Scheme)

- Technika usuwania (tj. selekcji negatywnej). W połączeniu z nią można zastosować dowolną technikę selekcji pozytywnej.
- Dzieli pełny zakres przystosowania rozwiązań na podprzedziały o równej szerokości.
- Usuwa losowe rozwiązanie z najbardziej wypełnionego podprzedziału.

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

Bibliografia

FUDS (Fitness Uniform Deletion Scheme)

- Technika usuwania (tj. selekcji negatywnej). W połączeniu z nią można zastosować dowolną technikę selekcji pozytywnej.
- Dzieli pełny zakres przystosowania rozwiązań na podprzedziały o równej szerokości.
- Usuwa losowe rozwiązanie z najbardziej wypełnionego podprzedziału.

- Niech f_{min} i f_{max} będą minimalną i maksymalną wartością przystosowania osiągalną dla danego problemu, lub sensownym górnym i dolnym ograniczeniem.
- Dzielimy przedział $[f_{min}, f_{max}]$ na zestaw podprzedziałów o równej szerokości $[f_{min}, f_{min} + \epsilon], [f_{min} + \epsilon, f_{min} + 2\epsilon], \dots, [f_{max} - \epsilon, f_{max}]$.
- Wyznaczany jest podprzedział z największą liczbą osobników, a następnie usuwany jest losowy osobnik z tego podprzedziału.
- W przypadku wielu podprzedziałów o tej samej liczbie osobników wybierany jest najniższy podprzedział [LH05].

Właściwości FUDS

Wstęp

Różnorodność
wartości
funkcji celu

FUSS: Fitness Uniform
Selection Scheme

FUDS: Fitness Uniform
Deletion Scheme

ConvSel: selekcja
konwekcyjna

HFC: Hierarchical Fair
Competition

Przykładowy eksperyment
porównawczy

MAP-Elites

Różnorodność
treści
rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna
vs. lokalna

NSLC: Novelty Search with
Local Competition

Przykładowy eksperyment
porównawczy

Bibliografia

- Brak problemu dominacji („przejęcia” – *takeover*).

Właściwości FUDS

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

- Brak problemu dominacji („przejęcia” – *takeover*).
- Umożliwia ciągłe tworzenie osobników na każdym poziomie przystosowania.

Bibliografia

Właściwości FUDS

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

- Brak problemu dominacji („przejęcia” – *takeover*).
- Umożliwia ciągłe tworzenie osobników na każdym poziomie przystosowania.
- Działanie odporne na różne nasilenia pozytywnej presji selekcyjnej: nawet bardzo niska pozytywna presja selekcyjna (nawet losowa pozytywna selekcja!) nie spowoduje utraty najlepszych osobników – zostaną one zachowane tak długo, jak długo będą rzadkie.

Bibliografia

Właściwości FUDS

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

Bibliografia

- Brak problemu dominacji („przejęcia” – *takeover*).
- Umożliwia ciągłe tworzenie osobników na każdym poziomie przystosowania.
- Działanie odporne na różne nasilenia pozytywnej presji selekcyjnej: nawet bardzo niska pozytywna presja selekcyjna (nawet losowa pozytywna selekcja!) nie spowoduje utraty najlepszych osobników – zostaną one zachowane tak długo, jak długo będą rzadkie.
- Prosta implementacja i niski koszt obliczeniowy.

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Niszowanie

Poسعکیوانه nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

ConvSel (Convection Selection)

- Była omówiona wcześniej.
- Jest wariantem *modelu wyspowego*.
- Dzieli populację na M podpopulacje, z których każda ma przypisany rozłączny zakres przystosowania o równej szerokości.
- Podpopulacje ewoluują niezależnie przez $N \cdot R$ ocen osobników, gdzie N to rozmiar pełnej populacji, a R to parametr skalujący.
- Następnie ponownie wyznaczane są zakresy dopasowania podpopulacji i rozwiązania są przypisywane (migrują) do nowych podpopulacji.
- Podczas niezależnej ewolucji w podpopulacjach można użyć dowolnego AE z dowolną selekcją pozytywną/negatywną.

Selekcja konwekcyjna – zjawisko konwekcji

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

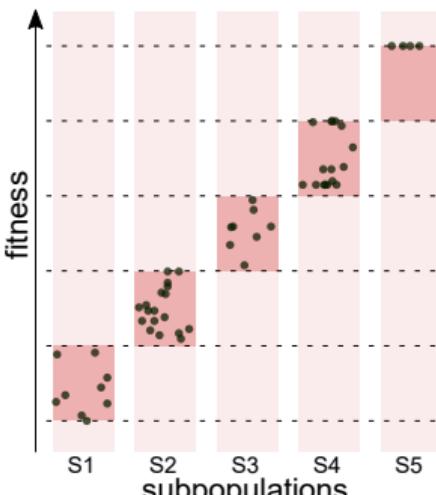
Poszukiwanie nowości

Dwukryterialność

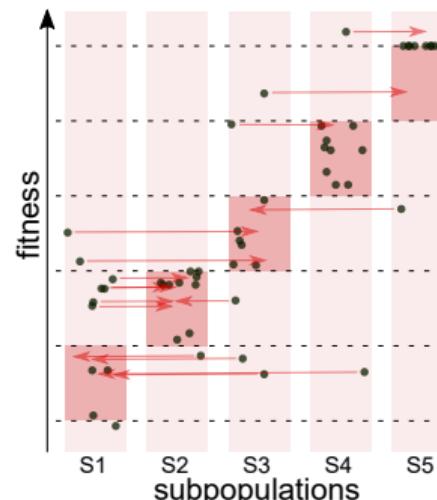
Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

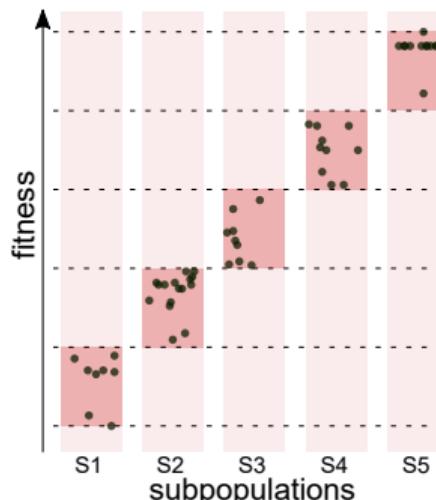
Przykładowy eksperyment porównawczy



Po i -tej migracji



Przed $(i+1)$ -tą migracją



Po $(i+1)$ -tej migracji

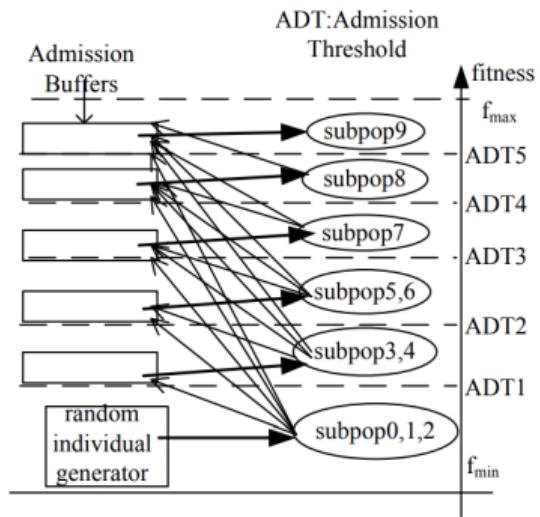
Rysunek: Poprawione potomstwo słabych rozwiązań może wzrosnąć się w górę łańcucha podpopulacji. <https://www.cs.put.poznan.pl/mkomosinski/convection-vpos.svg>

Bibliografia

HFC – logika

HFC (Hierarchical Fair Competition)

- Populacja dzielona na podpopulacje.
- Każda podpopulacja ma ustawione progowe wartości przystosowania: próg przyjęcia i próg odesłania; próg odesłania i -tej jest progiem przyjęcia $(i + 1)$ -szej.
- Rozwiązania, których przystosowanie przekracza próg odesłania swojej podpopulacji są przenoszone do bufora przyjęć (pozwala to uniknąć dominacji „mistrzów” w każdej podpopulacji).
- Zepsute rozwiązania, które mogą się pojawić i spaść poniżej progu przyjęcia nie są traktowane w żaden specjalny sposób i pozostają bez zmian w podpopulacjach.
- W momencie migracji rozwiązania przesuwają się z buforów przyjęć w górę hierarchii podpopulacji.
- Najgorsza podpopulacja jest dopełniana losowo generowanymi rozwiązaniami.



Rysunek: Model HFC [Hu+05].

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Pozukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

Bibliografia

HFC – metafory

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

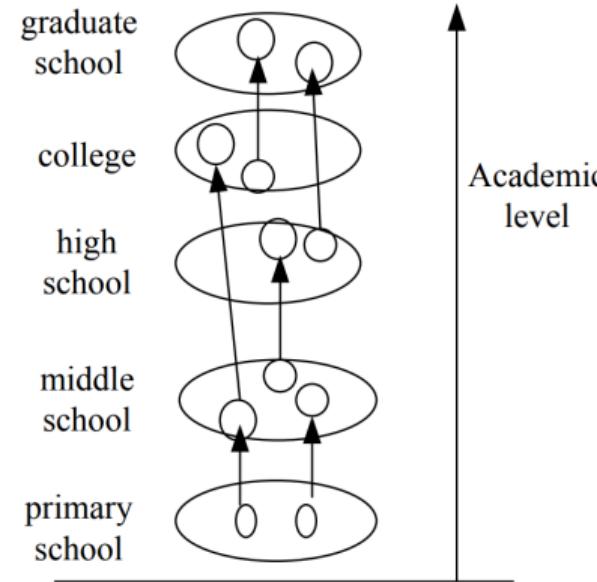
Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy



Rysunek: Metafora: zasada uczciwej konkurencji z systemów społecznych i gospodarczych [Hu+05]. Brak opadania w edukacji (?)

Bibliografia

Właściwości HFC

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

- Ciągłe dostarczanie i rozpowszechnianie nowego materiału genetycznego niskiej jakości.

Bibliografia

Właściwości HFC

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

Bibliografia

- Ciągłe dostarczanie i rozpowszechnianie nowego materiału genetycznego niskiej jakości.
- Unikanie efektu założyciela.

Właściwości HFC

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

Bibliografia

- Ciągłe dostarczanie i rozpowszechnianie nowego materiału genetycznego niskiej jakości.
- Unikanie **efektu założyciela**.
- Zepsute rozwiązania nie mogą pogorszyć podpopulacji niższego poziomu...

Właściwości HFC

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

Bibliografia

- Ciągłe dostarczanie i rozpowszechnianie nowego materiału genetycznego niskiej jakości.
- Unikanie **efektu założyciela**.
- Zepsute rozwiązań nie mogą pogorszyć podpopulacji niższego poziomu...
- ... ale też nie mają szansy na poprawę – zabije je presja selekcyjna w ich podpopulacji.

Właściwości HFC

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

Bibliografia

- Ciągłe dostarczanie i rozpowszechnianie nowego materiału genetycznego niskiej jakości.
- Unikanie **efektu założyciela**.
- Zepsute rozwiązań nie mogą pogorszyć podpopulacji niższego poziomu...
- ... ale też nie mają szansy na poprawę – zabije je presja selekcyjna w ich podpopulacji.
- Podobnie jak w selekcji konwekcyjnej, podczas ewolucji w podpopulacjach można użyć dowolnego AE z dowolną selekcją pozytywną/negatywną.

Porównanie: warunki eksperymentu

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

Bibliografia

Cztery algorytmy:

- jednopoludacyjny standardowy AE (StdEA)
- AE z fitness uniform selection scheme (FUSS)
- AE z fitness uniform deletion scheme (FUDS)
- wielopoludacyjny AE z selekcją konwekcyjną (ConvSel)
- brak HFC :(nie starczyło czasu na implementację [BKM22], ale prace trwają!

Cztery zadania:

- Trzy proste testy matematyczne: Drop-wave function, Shaffer function N.2, Shaffer function N.4.
- Jeden trudny problem projektowania ewolucyjnego (ewolucja szybkiego ruchu symulowanych agentów w płaskim środowisku 3D).

Wszystkie algorytmy wykorzystywały ewolucję *steady-state* z selekcją turniejową i losowym usuwaniem osobników.

Porównanie neutralne względem parametrów testowanych metod!

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Pośzukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

Bibliografia

Nazwa parametru	StdEA	FUSS	FUDS	ConvSel	Wartości
<i>rozmiar populacji</i>	✓	✓	✓	✓	[100, 200, 500]
<i>wielkość turnieju</i>	✓		✓	✓	[3, 5, 7]
<i>prawd. krzyżowania</i>	✓	✓	✓	✓	[0.5, 0.75]
<i>prawd. mutacji</i>	✓	✓	✓	✓	[0.25, 0.5]
M				✓	[5, 10]
R				✓	[10, 25]
α (tylko matemat.)	✓	✓	✓	✓	[0.2, 0.4]

Tabela: Wartości parametrów użytych w eksperymentach. Dla każdego parametru zaznaczono metody, które go używają. Parametr α (zakres mutacji) był używany tylko w testach matematycznych.

Parametry eksperymentu

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Pośzukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

- Testowanie każdej kombinacji algorytmu, wartości parametrów i funkcji przystosowania.
- 30 (testy matematyczne) albo 10 (projektowanie ewolucyjne) niezależnych przebiegów ewolucyjnych na kombinację.
- Średnia i odchylenie standardowe obliczone dla każdej kombinacji wartości parametrów.
- $5 \cdot 10^5$ rozwiązań (testy matematyczne) albo 10^6 rozwiązań (projektowanie ewolucyjne) na przebieg.
- Porównanie neutralne względem parametrów porównywanych metod (w każdym momencie ewolucji uwzględnia najlepszy wynik ze wszystkich kombinacji wartości parametrów).

Bibliografia

Zadanie projektowania ewolucyjnego (prędkość na lądzie)

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Pośzukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

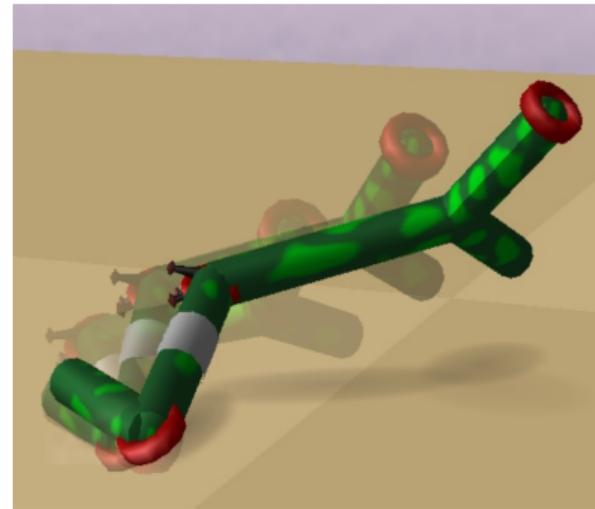
NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

Bibliografia

Metoda	Średnia	Odch.std.
StdEA	0.0143	0.0147
FUSS	0.0219	0.0141
FUDS	0.0177	0.0090
ConvSel	0.0216	0.0198

Tabela: Porównanie najlepszych średnich wartości uzyskanych metodami utrzymywania różnorodności wartości funkcji celu dla zadania maksymalizacji prędkości agenta.



Rysunek: Przykład wyewoluowanego agenta w ruchu, symulowanego w środowisku Framsticks.

Wyniki: zadanie projektowania ewolucyjnego (prędkość na lądzie)

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

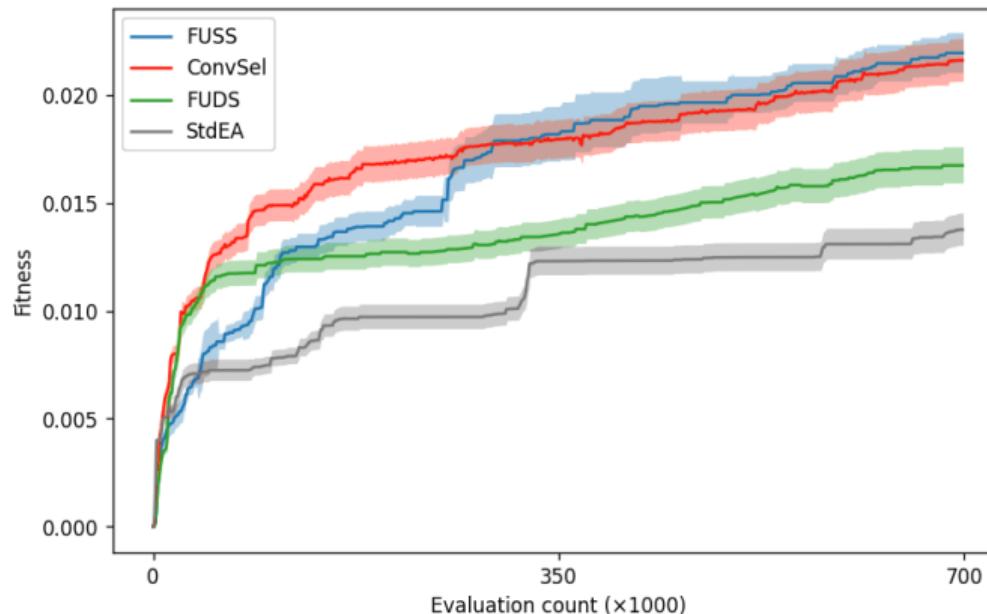
Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy



Rysunek: Porównanie skuteczności metod utrzymywania różnorodności wartości funkcji celu. Każda linia przedstawia najwyższą wartość średniej z zestawu parametrów (dla tego każdy punkt może być uzyskany z różnych wartości parametrów). Pas wokół każdej linii odpowiada 5% odchylenia standardowego.

Bibliografia

Eksperyment: wnioski

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

Bibliografia

- Dla trudnego problemu projektowania ewolucyjnego, wszystkie testowane metody utrzymywania różnorodności przystosowania osiągają lepsze wyniki niż jednopolulacyjny AE typu *steady-state* z selekcją turniejową, przy czym FUSS i ConvSel wyprzedzają FUDS.
- Dla łatwiejszych, matematycznych testów porównawczych, FUDS i ConvSel przewyższają tradycyjne podejścia ewolucyjne.

MAP-Elites

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

Bardziej skomplikowanym mechanizmem dywersyfikacji (aczkolwiek wymagającym wprowadzenia dodatkowych kryteriów oceny) jest MAP-Elites (Multi-dimensional Archive of Phenotypic Elites) [MC15].

- Zdefiniuj miarę przystosowania jak zwykle (na przykład: maksymalizacja prędkości robota).
- Zdefiniuj inne cechy każdego rozwiązania (na przykład: rozmiar robota, waga, zużycie energii).
- Zdyskretyzuj zakresy tych innych cech, tworząc w ten sposób wielowymiarową siatkę z „komórkami”.
- Podczas ewolucji wybieraj osobnika z losowej komórki, mutuj/krzyżuj go, ocenij jego przystosowanie i cechy, a następnie umieszczaj w odpowiedniej komórce.
- W każdej komórce przechowuj tylko jedno rozwiązanie z najlepszym przystosowaniem znalezionym do tej pory.

Bibliografia

MAP-Elites

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

Bibliografia

Bardziej skomplikowanym mechanizmem dywersyfikacji (aczkolwiek wymagającym wprowadzenia dodatkowych kryteriów oceny) jest MAP-Elites (Multi-dimensional Archive of Phenotypic Elites) [MC15].

- Zdefiniuj miarę przystosowania jak zwykle (na przykład: maksymalizacja prędkości robota).
- Zdefiniuj inne cechy każdego rozwiązania (na przykład: rozmiar robota, waga, zużycie energii).
- Zdyskretyzuj zakresy tych innych cech, tworząc w ten sposób wielowymiarową siatkę z „komórkami”.
- Podczas ewolucji wybieraj osobnika z losowej komórki, mutuj/krzyżuj go, ocenij jego przystosowanie i cechy, a następnie umieszczaj w odpowiedniej komórce.
- W każdej komórce przechowuj tylko jedno rozwiązanie z najlepszym przystosowaniem znalezionym do tej pory.

Warianty:

- Przechowywanie więcej niż jednego rozwiązania w każdej komórce.
- Rozpoczęcie od zgrubnej dyskretyzacji cech i stopniowe zwiększanie liczby przedziałów (rozdzielczości).

Niszowanie

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

Bibliografia

Omawiane już wcześniej:

nowa ocena := pierwotna ocena/(suma podobieństw do pozostałych osobników)

... albo równoważnie:

nowa ocena := pierwotna ocena · (1 + suma odległości do pozostałych osobników)

Poszukiwanie nowości

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

Egzotycznym pomysłem na algorytm optymalizacji wydaje się zupełnie zignorowanie funkcji celu i oparcie logiki algorytmu na samym mianowniku wzoru dla niszowania („pierwotna ocena” staje się wtedy stałą). Jednak przy pewnych problemach optymalizacji i odpowiedniej mierze podobieństwa (opartej zwykle o złożone cechy fenotypowe), takie podejście (ang. *novelty search*) może wystarczyć do skutecznego optymalizowania.

Wymuszanie jedynie różnorodności przydaje się też wszędzie tam, gdzie chcemy równomiernie pokryć pewną złożoną przestrzeń – np. do generowania możliwie różnych testów programistycznych, funkcji oceniających, konfiguracji w grach, generatywnych procedur, antygenów w biologii, itp.

Bibliografia

Dwukryterialność

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

Bibliografia

W przeciwieństwie do wzoru dla niszowania, który agreguje jakość rozwiązania i odmiennosć jego treści od innych znanych rozwiązań, można potraktować oba te czynniki jako osobne kryteria (adekwatnie do nazwy omawianej teraz grupy metod: *quality–diversity*) i użyć dwukryterialnej optymalizacji (np. NSGA-2).

Główną zaletą takiego podejścia jest brak kompensacji (typowe przy przejściu z agregacji kryteriów do ich osobnego traktowania).

Różnorodność globalna vs. lokalna

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

Bibliografia

- $dist$ – miara odległości
- x_i – i -te rozwiązanie
- n – rozmiar populacji
- k – liczba rozwiązań, $k < n$, najbliższych i -temu rozwiązaniu wedle $dist$

Metryka globalnej nowości dla rozwiązania i -tego to

$$distance_global_i = \frac{1}{n} \sum_{\substack{j=1 \\ j \neq i}}^n dist(x_i, x_j)$$

Metryka lokalnej nowości dla rozwiązania i -tego to

$$distance_local_i = \frac{1}{k} \sum_{\substack{j=1 \\ j \neq i}}^k dist(x_i, x_j)$$

W poszukiwaniu nowości jest to jedyne optymalizowane kryterium.

W niszowaniu, pierwotna ocena jest mnożona przez $(1 + \text{metryka globalnej lub lokalnej nowości})$.

NSLC: Novelty Search with Local Competition

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Pozyskiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

Bibliografia

Cechy [LS11]: archiwum (pamięć: monitorowanie odwiedzonych obszarów) i lokalna konkurencja (dywersyfikacja tylko lokalnie). Obie te cechy mogą być (i są) wykorzystywane także w niszowaniu i w poszukiwaniu nowości (te dwie metody zostały właśnie omówione).

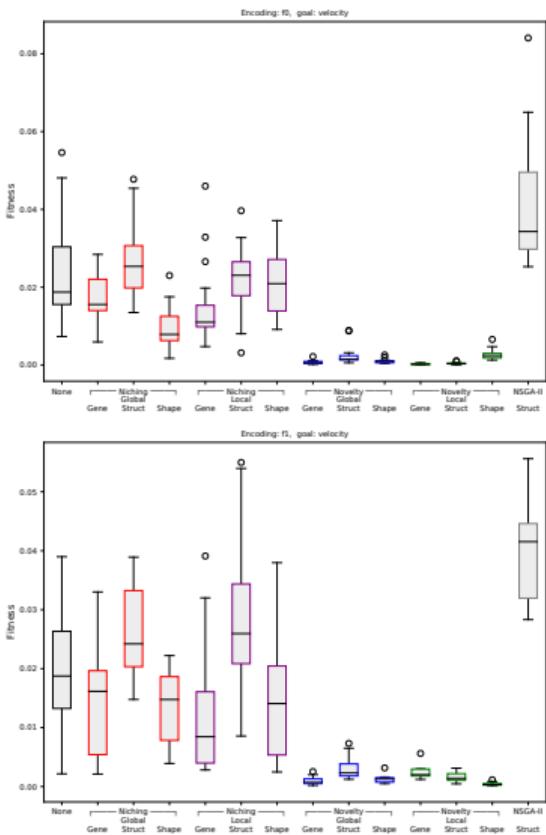
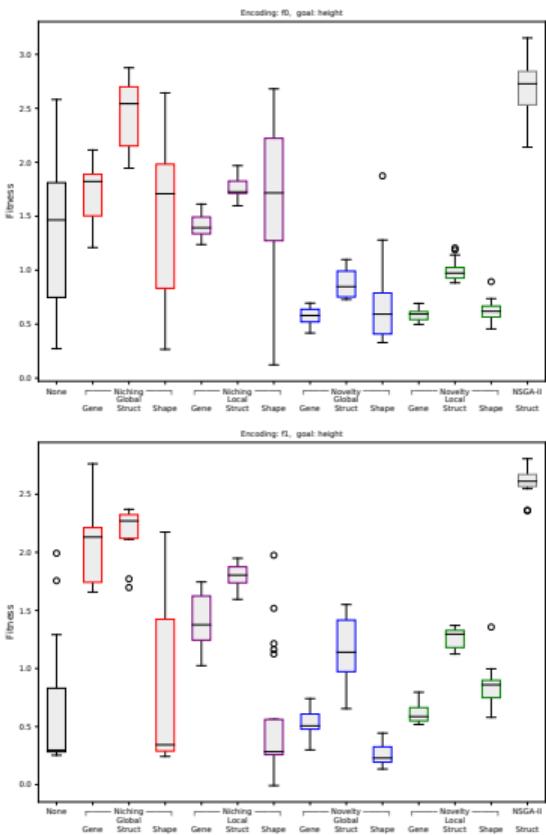
Z [CD17]:

«

In NSLC, the exploration focuses on solutions that are both novel (according to the novelty score) and locally high-performing. The main insight consists of comparing the performance of a solution only to those that are close in the descriptor space. This is achieved with a “local quality score” that is defined as the number of the k -nearest neighboring solutions in the novelty archive with a lower performance (e.g., slower walking speed) than the considered solution. The exploration is then achieved with a multi-objective optimization algorithm (e.g., NSGA-II) that optimizes both the novelty and local quality scores of the solutions.

However, the local quality score does not influence the threshold used to select whether an individual is added to the novelty archive. The final result of NSLC is the population containing solutions that are both novel and high-performing compared to other local solutions. In other words, the population gathers solutions that are both different from those saved in the novelty archive, and high-performing when compared to similar types of solutions.

»



Rysunek: Surowe wartości przystosowania najlepszych wyewoluowanych rozwiązań dla różnych technik utrzymywania różnorodności. Kolumny to kryteria przystosowania (po lewej: wysokość, po prawej: prędkość). Wiersze to reprezentacje genetyczne (góra: f_0 , dół: f_1). 14·10 rozwiązań na każdym wykresie. Miara odległości: genetyczna Levenshteina (Gene), fenetyczna struktura grafu (Struct), fenetyczna używająca deskryptorów kształtu (Shape).

Bibliografia |

Wstęp

Różnorodność wartości funkcji celu

FUSS: Fitness Uniform Selection Scheme

FUDS: Fitness Uniform Deletion Scheme

ConvSel: selekcja konwekcyjna

HFC: Hierarchical Fair Competition

Przykładowy eksperyment porównawczy

MAP-Elites

Różnorodność treści rozwiązań

Niszowanie

Poszukiwanie nowości

Dwukryterialność

Różnorodność globalna vs. lokalna

NSLC: Novelty Search with Local Competition

Przykładowy eksperyment porównawczy

Bibliografia

- [BKM22] Kamil Basiukajc, Maciej Komosinski i Konrad Miazga. "Fitness Diversification in the Service of Fitness Optimization: a Comparison Study". W: *Genetic and Evolutionary Computation Conference Companion (GECCO '22)*. Boston, USA: ACM, 2022, s. 471–474. DOI: [10.1145/3520304.3528949](https://doi.org/10.1145/3520304.3528949). URL: <http://www.framsticks.com/files/common/FitnessDiversity.pdf>.
- [CD17] Antoine Cully i Yiannis Demiris. "Quality and diversity optimization: A unifying modular framework". W: *IEEE Transactions on Evolutionary Computation* 22.2 (2017), s. 245–259. URL: <https://arxiv.org/pdf/1708.09251.pdf>.
- [HL06] Marcus Hutter i Shane Legg. "Fitness uniform optimization". W: *IEEE Transactions on Evolutionary Computation* 10.5 (2006), s. 568–589. URL: <https://arxiv.org/pdf/cs/0610126.pdf>.
- [Hu+05] Jianjun Hu i in. "The Hierarchical Fair Competition (HFC) framework for sustainable evolutionary algorithms". W: *Evolutionary Computation* 13.2 (2005), s. 241–277. URL: <https://citeserx.ist.psu.edu/document?repid=rep1&type=pdf&doi=9a9418ae55bdff2f1a988effe8554e8ec0050df0>.
- [LH05] Shane Legg i Marcus Hutter. "Fitness uniform deletion: A simple way to preserve diversity". W: *Proceedings of the 7th annual conference on Genetic and evolutionary computation*. 2005, s. 1271–1278. URL: <https://arxiv.org/pdf/cs/0504035.pdf>.
- [LS11] Joel Lehman i Kenneth O. Stanley. "Evolving a diversity of virtual creatures through novelty search and local competition". W: *Proceedings of the 13th annual conference on genetic and evolutionary computation*. 2011, s. 211–218. URL: http://plex.cs.ucf.edu/papers/lehman_gecco11.pdf.
- [MC15] Jean-Baptiste Mouret i Jeff Clune. "Illuminating search spaces by mapping elites". W: *arXiv preprint arXiv:1504.04909* (2015). URL: <https://arxiv.org/pdf/1504.04909.pdf>.

Algorytmy i modele inspirowane biologicznie

6. Koewolucja kooperatywna i konkurencyjna

Maciej Komosiński



**Fundusze
Europejskie**
Polska Cyfrowa



**Rzeczpospolita
Polska**

Unia Europejska
Europejski Fundusz
Rozwoju Regionalnego



Koewolucja

Koewolucja
koopera-
tywna

Koewolucja
konkuren-
cyjna

Bibliografia

Wiele gatunków (grup organizmów) – co najmniej dwie – wpływa nawzajem na swoje procesy ewolucyjne.

Koewolucja

Koewolucja
koopera-
tywna

Koewolucja
konkuren-
cyjna

Bibliografia

Wiele gatunków (grup organizmów) – co najmniej dwie – wpływa nawzajem na swoje procesy ewolucyjne.

Pytanie: jak to wpływa na krajobraz funkcji celu?

Podstawowe zagadnienia kooperacji w optymalizacji

Koewolucja
koopera-
tywna

Koewolucja
konkuren-
cyjna

Bibliografia

Omawiany w tej części przykład architektury kooperatywnej bazuje na artykule [PD00].

Implementacja tej architektury jest dostępna w bibliotece

DEAP: https://deap.readthedocs.io/en/master/examples/coev_coop.html.

Podstawowe zagadnienia kooperacji w optymalizacji

Koewolucja
koopera-
tywna

Koewolucja
konkuren-
cyjna

Bibliografia

Omawiany w tej części przykład architektury kooperatywnej bazuje na artykule [PD00].

Implementacja tej architektury jest dostępna w bibliotece

DEAP: https://deap.readthedocs.io/en/master/examples/coev_coop.html.

- Optymalizacja złożonych problemów będzie skuteczniejsza, jeśli będziemy umieli dokonać ich dekompozycji na części.

Podstawowe zagadnienia kooperacji w optymalizacji

Koewolucja
koopera-
tywna

Koewolucja
konkuren-
cyjna

Bibliografia

Omawiany w tej części przykład architektury kooperatywnej bazuje na artykule [PD00].

Implementacja tej architektury jest dostępna w bibliotece

DEAP: https://deap.readthedocs.io/en/master/examples/coev_coop.html.

- Optymalizacja złożonych problemów będzie skuteczniejsza, jeśli będziemy umieli dokonać ich dekompozycji na części.
- Jak wiemy z wcześniejszych wykładów (por. epistaza, hierarchiczny AG), niestety wpływ poszczególnych części rozwiązania na wartość funkcji celu jest zwykle silnie powiązany.

Podstawowe zagadnienia kooperacji w optymalizacji

Koewolucja koopera-tywna

Koewolucja konkuren-cyjna

Bibliografia

Omawiany w tej części przykład architektury kooperatywnej bazuje na artykule [PD00].

Implementacja tej architektury jest dostępna w bibliotece

DEAP: https://deap.readthedocs.io/en/master/examples/coev_coop.html.

- Optymalizacja złożonych problemów będzie skuteczniejsza, jeśli będziemy umieli dokonać ich dekompozycji na części.
- Jak wiemy z wcześniejszych wykładów (por. epistaza, hierarchiczny AG), niestety wpływ poszczególnych części rozwiązania na wartość funkcji celu jest zwykle silnie powiązany.
- W tradycyjnym AE osobniki są oceniane zupełnie niezależnie – zatem nie ma pola do ich współpracy między sobą.

Podstawowe zagadnienia kooperacji w optymalizacji

Koewolucja kooperatywna

Koewolucja konkurencyjna

Bibliografia

Omawiany w tej części przykład architektury kooperatywnej bazuje na artykule [PD00].

Implementacja tej architektury jest dostępna w bibliotece

DEAP: https://deap.readthedocs.io/en/master/examples/coev_coop.html.

- Optymalizacja złożonych problemów będzie skuteczniejsza, jeśli będziemy umieli dokonać ich dekompozycji na części.
- Jak wiemy z wcześniejszych wykładów (por. epistaza, hierarchiczny AG), niestety wpływ poszczególnych części rozwiązania na wartość funkcji celu jest zwykle silnie powiązany.
- W tradycyjnym AE osobniki są oceniane zupełnie niezależnie – zatem nie ma pola do ich współpracy między sobą.
- Gdyby udało się zdekomponować problem (nawet ręcznie, np. dzieląc osobnika – wektor zmiennych na poszczególne zmienne), będziemy musieli rozwiązać problemy: 1) zależności krajobrazu przystosowania z punktu widzenia każdej części od wartości innych współpracujących części, 2) „przypisania zasług” (ang. *credit assignment*) częściom, 3) utrzymania różnorodności w ramach populacji części.

Podstawowe zagadnienia kooperacji w optymalizacji

Koewolucja kooperatywna

Koewolucja konkurencyjna

Bibliografia

Omawiany w tej części przykład architektury kooperatywnej bazuje na artykule [PD00].

Implementacja tej architektury jest dostępna w bibliotece

DEAP: https://deap.readthedocs.io/en/master/examples/coev_coop.html.

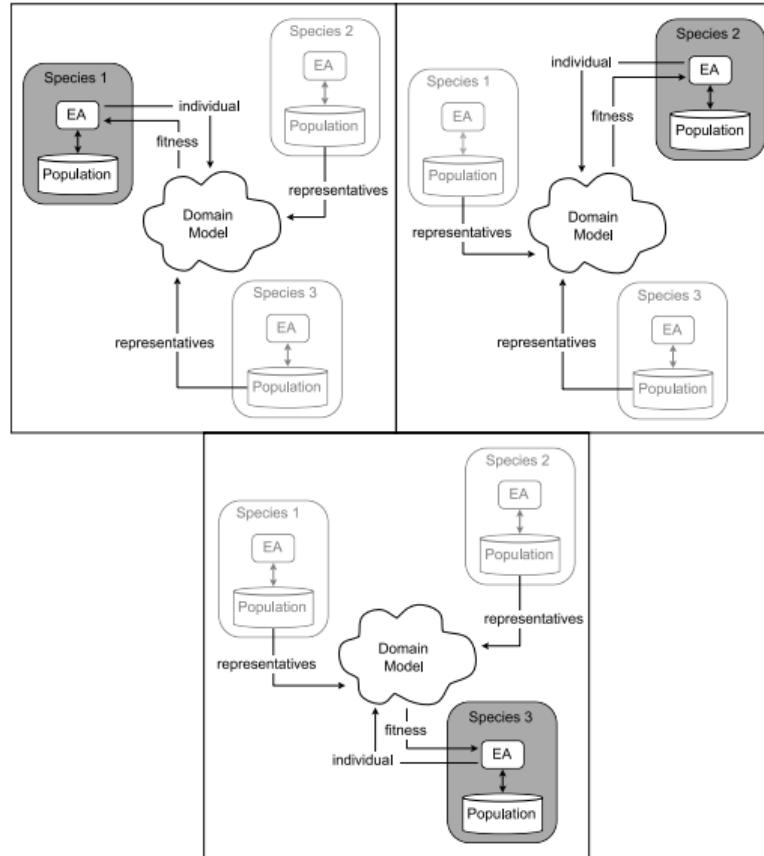
- Optymalizacja złożonych problemów będzie skuteczniejsza, jeśli będziemy umieli dokonać ich dekompozycji na części.
- Jak wiemy z wcześniejszych wykładów (por. epistaza, hierarchiczny AG), niestety wpływ poszczególnych części rozwiązania na wartość funkcji celu jest zwykle silnie powiązany.
- W tradycyjnym AE osobniki są oceniane zupełnie niezależnie – zatem nie ma pola do ich współpracy między sobą.
- Gdyby udało się zdekomponować problem (nawet ręcznie, np. dzieląc osobnika – wektor zmiennych na poszczególne zmienne), będziemy musieli rozwiązać problemy: 1) zależności krajobrazu przystosowania z punktu widzenia każdej części od wartości innych współpracujących części, 2) „przypisania zasług” (ang. *credit assignment*) częściom, 3) utrzymania różnorodności w ramach populacji części.
- Założymy, że dzielimy rozwiązanie na części, które trafiają do osobnych (genetycznie niezależnych) populacji – zatem reprezentacja genetyczna i operatory w każdej populacji mogą być zupełnie różne. Jak ocenić jakość każdej części?

Architektura badana przez Pottera i De Jonga

Koewolucja koopera-tywna

Koewolucja konkuren-cyjna

Bibliografia



Adaptacja liczby gatunków

Koewolucja
koopera-
tywna

Koewolucja
konkuren-
cyjna

Bibliografia

- Dla funkcji n zmiennych możemy zdekomponować zadanie ręcznie na n gatunków.

Adaptacja liczby gatunków

Koewolucja
koopera-
tywna

Koewolucja
konkuren-
cyjna

Bibliografia

- Dla funkcji n zmiennych możemy zdekomponować zadanie ręcznie na n gatunków.
- Dla agenta (np. robota) z regułowym systemem sterującym, możemy ręcznie podzielić zbiór reguł na dwa zbiory odpowiedzialne za dwa zachowania (np. podążanie do celu i czekanie, aż cel się pojawi).

Adaptacja liczby gatunków

Koewolucja
koopera-
tywna

Koewolucja
konkuren-
cyjna

Bibliografia

- Dla funkcji n zmiennych możemy zdekomponować zadanie ręcznie na n gatunków.
- Dla agenta (np. robota) z regułowym systemem sterującym, możemy ręcznie podzielić zbiór reguł na dwa zbiory odpowiedzialne za dwa zachowania (np. podążanie do celu i czekanie, aż cel się pojawi).
- Jeśli mielibyśmy zadanie, w którym liczba części może się dynamicznie zmieniać, byłoby świetnie gdyby algorytm sam potrafił dobierać liczbę gatunków oraz ich funkcję (\rightarrow „nisze” ...) we współpracy z innymi. Co może wyzwalać dodanie nowego gatunku oraz usunięcie istniejącego?

Adaptacja liczby gatunków

Koewolucja
koopera-
tywna

Koewolucja
konkuren-
cyjna

Bibliografia

- Dla funkcji n zmiennych możemy zdekomponować zadanie ręcznie na n gatunków.
- Dla agenta (np. robota) z regułowym systemem sterującym, możemy ręcznie podzielić zbiór reguł na dwa zbiory odpowiedzialne za dwa zachowania (np. podążanie do celu i czekanie, aż cel się pojawi).
- Jeśli mielibyśmy zadanie, w którym liczba części może się dynamicznie zmieniać, byłoby świetnie gdyby algorytm sam potrafił dobierać liczbę gatunków oraz ich funkcję (\rightarrow „nisze” ...) we współpracy z innymi. Co może wyzwalać dodanie nowego gatunku oraz usunięcie istniejącego?
 - Dodanie nowego, np. losowo zainicjowanego: kiedy system jest w stagnacji – brak wzrostu jakości najlepszego osobnika.

Adaptacja liczby gatunków

Koewolucja
koopera-
tywna

Koewolucja
konkuren-
cyjna

Bibliografia

- Dla funkcji n zmiennych możemy zdekomponować zadanie ręcznie na n gatunków.
- Dla agenta (np. robota) z regułowym systemem sterującym, możemy ręcznie podzielić zbiór reguł na dwa zbiory odpowiedzialne za dwa zachowania (np. podążanie do celu i czekanie, aż cel się pojawi).
- Jeśli mielibyśmy zadanie, w którym liczba części może się dynamicznie zmieniać, byłoby świetnie gdyby algorytm sam potrafił dobierać liczbę gatunków oraz ich funkcję (\rightarrow „nisze” ...) we współpracy z innymi. Co może wyzyskać dodanie nowego gatunku oraz usunięcie istniejącego?
 - Dodanie nowego, np. losowo zainicjowanego: kiedy system jest w stagnacji – brak wzrostu jakości najlepszego osobnika.
 - Usunięcie istniejącego: np. jeśli jego wkład do współpracy (różnica między jakością osobników z nim i bez niego) jest poniżej zadanego progu.

[PD00] Eksperyment 1: pokrycie zbioru

- Znajdź zbiór M z m ciągami binarnymi pokrywający zbiór T z t ciągami binarnymi, $m \ll t$.

Koewolucja
koopera-
tywna

Koewolucja
konkuren-
cyjna

Bibliografia

[PD00] Eksperyment 1: pokrycie zbioru

- Znajdź zbiór M z m ciągami binarnymi pokrywający zbiór T z t ciągami binarnymi, $m \ll t$.
- Pokrycie dla dwóch wektorów binarnych to liczba ich identycznych bitów.

Koewolucja koopera-tywna

Koewolucja konkuren-cyjna

Bibliografia

[PD00] Eksperyment 1: pokrycie zbioru

- Znajdź zbiór M z m ciągami binarnymi pokrywający zbiór T z t ciągami binarnymi, $m \ll t$.
- Pokrycie dla dwóch wektorów binarnych to liczba ich identycznych bitów.
- Pokrycie M to średnia z (dla każdego wektora z T znajdź najlepiej pokrywający go wektor z M).

Koewolucja koopera-tywna

Koewolucja konkuren-cyjna

Bibliografia

[PD00] Eksperyment 1: pokrycie zbioru

- Znajdź zbiór M z m ciągami binarnymi pokrywający zbiór T z t ciągami binarnymi, $m \ll t$.
- Pokrycie dla dwóch wektorów binarnych to liczba ich identycznych bitów.
- Pokrycie M to średnia z (dla każdego wektora z T znajdź najlepiej pokrywający go wektor z M).
- Zatem M musi zawierać wzorce częste w T – czyli musi generalizować.

Koewolucja koopera-tywna

Koewolucja konkuren-cyjna

Bibliografia

[PD00] Eksperyment 1: pokrycie zbioru

Koewolucja
koopera-
tywna

Koewolucja
konkuren-
cyjna

Bibliografia

- Znajdź zbiór M z m ciągami binarnymi pokrywający zbiór T z t ciągami binarnymi, $m \ll t$.
- Pokrycie dla dwóch wektorów binarnych to liczba ich identycznych bitów.
- Pokrycie M to średnia z (dla każdego wektora z T znajdź najlepiej pokrywający go wektor z M).
- Zatem M musi zawierać wzorce częste w T – czyli musi generalizować.
- Jak gatunki stanowiące M odkryły wzorce ukryte celowo w T ?

[PD00] Eksperyment 1: pokrycie zbioru

Koewolucja
koopera-
tywna

Koewolucja
konkuren-
cyjna

Bibliografia

- Znajdź zbiór M z m ciągami binarnymi pokrywający zbiór T z t ciągami binarnymi, $m \ll t$.
- Pokrycie dla dwóch wektorów binarnych to liczba ich identycznych bitów.
- Pokrycie M to średnia z (dla każdego wektora z T znajdź najlepiej pokrywający go wektor z M).
- Zatem M musi zawierać wzorce częste w T – czyli musi generalizować.
- Jak gatunki stanowiące M odkryły wzorce ukryte celowo w T ?

- Znajdź zbiór M z m ciągami binarnymi pokrywający zbiór T z t ciągami binarnymi, $m \ll t$.
 - Pokrycie dla dwóch wektorów binarnych to liczba ich identycznych bitów.
 - Pokrycie M to średnia z (dla każdego wektora z T znajdź najlepiej pokrywający go wektor z M).
 - Zatem M musi zawierać wzorce częste w T – czyli musi generalizować.
 - Jak gatunki stanowiące M odkryły wzorce ukryte celowo w T ?

T_1 :

T_2 :

T_3 :

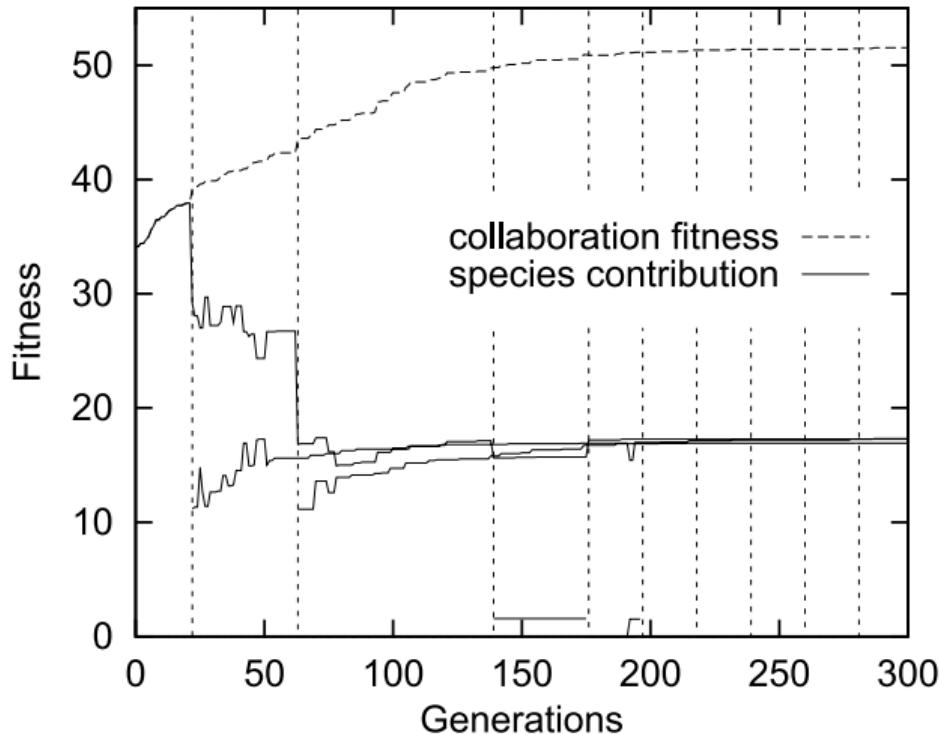
[PD00] Eksperyment 1: pokrycie zbioru – przebieg koewolucji

Koewolucja kooperatywna

Koewolucja konkurencyjna

Bibliografia

Automatycznie ustalana liczba gatunków; pionowe linie to momenty dodania gatunku w wyniku wykrycia stagnacji (22, 63, 138) lub usunięcia z powodu jego niewielkiego wkładu (oscylacje począwszy od 176).

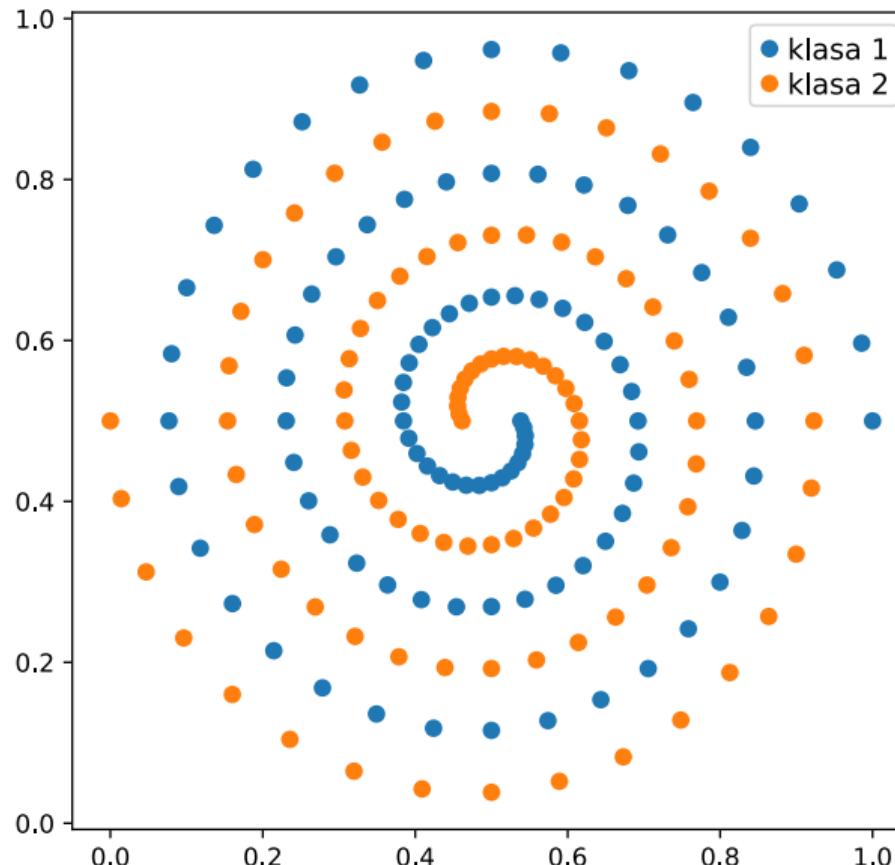


[PD00] Eksperyment 2: klasyfikacja złośliwego zbioru przypadków

Koewolucja koopera-tywna

Koewolucja konkuren-cyjna

Bibliografia



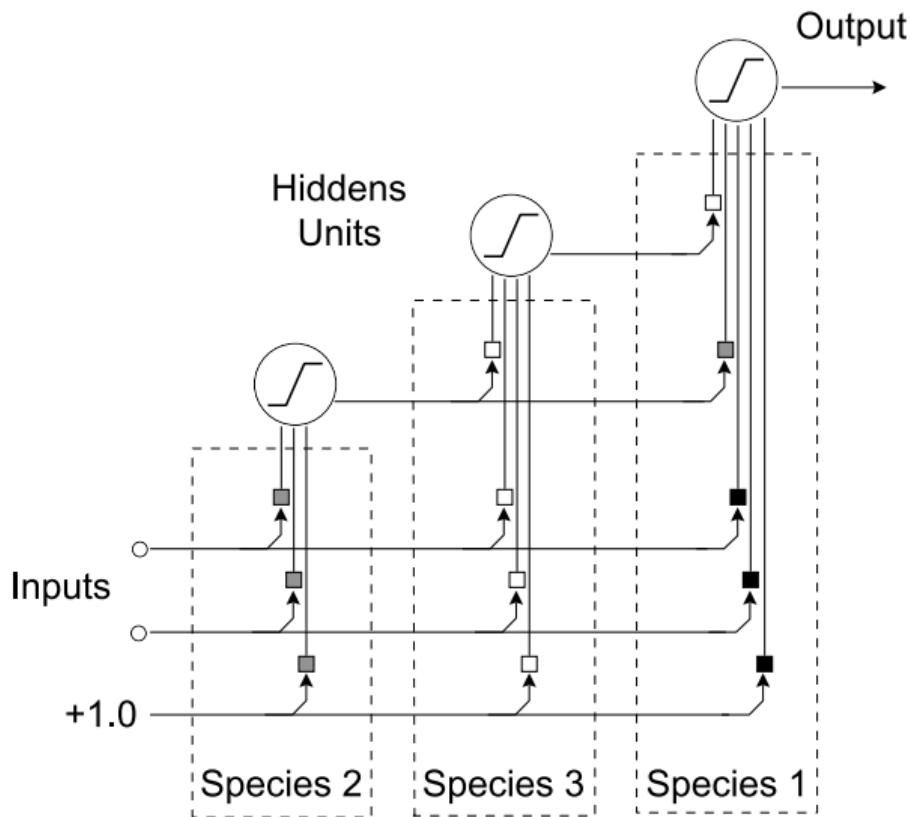
[PD00] Eksperyment 2: klasyfikacja złośliwego zbioru przypadków

Sieć kaskadowa (por. [gradient boosting](#)): prosta heurystyka jako odniesienie oraz gatunki w koewolucji

Koewolucja
koopera-
tywna

Koewolucja
konkuren-
cyjna

Bibliografia



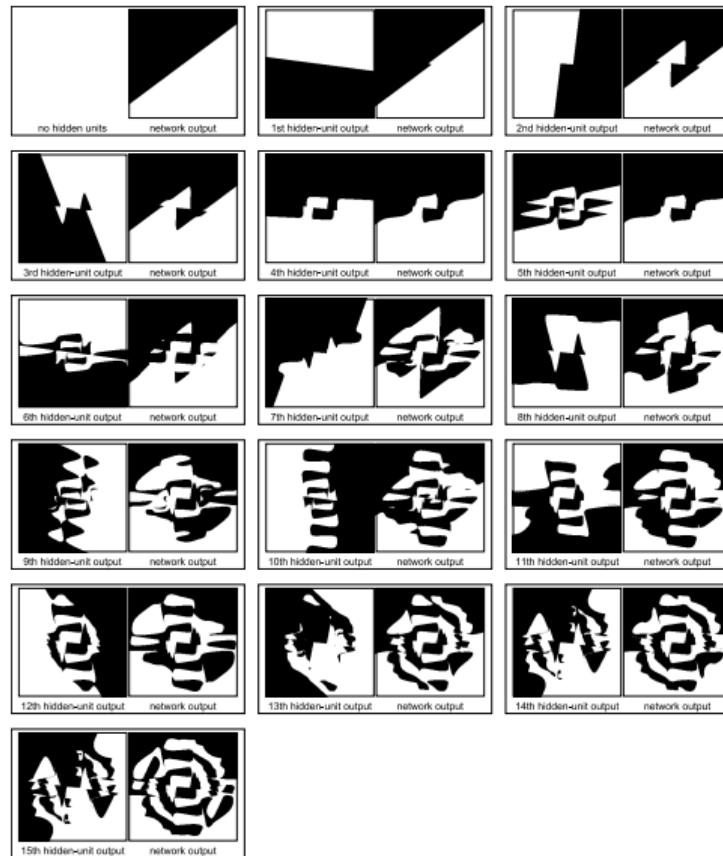
[PD00] Eksperyment 2: klasyfikacja złośliwego zbioru przypadków

Wyniki sieci kaskadowej: heurystyka ucząca

Koewolucja koopera-tywna

Koewolucja konkuren-cyjna

Bibliografia



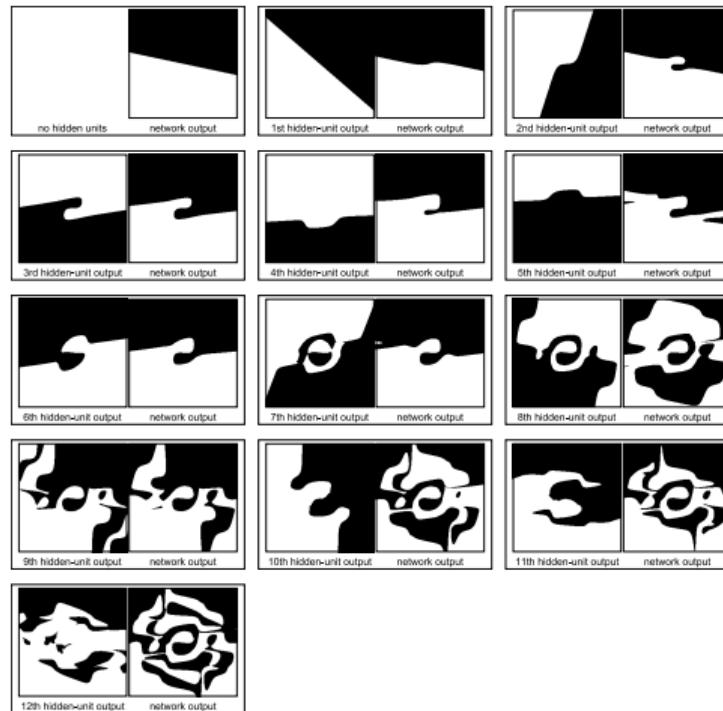
[PD00] Eksperyment 2: klasyfikacja złośliwego zbioru przypadków

Wyniki sieci kaskadowej: koewolucja kooperatywna

Koewolucja kooperatywna

Koewolucja konkurencyjna

Bibliografia



Koewolucja konkurencyjna – pożądany sposób działania

Koewolucja kooperatywna

Koewolucja konkurencyjna

Bibliografia

Typowy przykład: koewolucja (optymalizacja) strategii [Elf+21]. Osobnik reprezentuje wiedzę zawartą w strategii (np. mogą być to wagi kryteriów używanych do oceny sytuacji na planszy w grze). Ocenę osobnika uzyskuje się np. przez rozegranie wielu partii z pozostałymi osobnikami (każdy gra według własnej strategii).

Dyskusja: czy włączając taki proces, jeśli odpowiednio długo poczekamy, otrzymamy mistrzowską strategię? Jeśli nie, to dlaczego? (podaj przyczyny – listę ewentualnych problemów).

Koewolucja konkurencyjna – pożądany sposób działania

Koewolucja koopera-tywna

Koewolucja konkuren-cyjna

Bibliografia

Typowy przykład: koewolucja (optymalizacja) strategii [Elf+21]. Osobnik reprezentuje wiedzę zawartą w strategii (np. mogą być to wagi kryteriów używanych do oceny sytuacji na planszy w grze). Ocenę osobnika uzyskuje się np. przez rozegranie wielu partii z pozostałymi osobnikami (każdy gra według własnej strategii).

Dyskusja: czy włączając taki proces, jeśli odpowiednio długo poczekamy, otrzymamy mistrzowską strategię? Jeśli nie, to dlaczego? (podaj przyczyny – listę ewentualnych problemów).

Trudności: chcemy *arms race* (ciągła konkurencja), lecz możemy zostać w *MSS – Mediocre Stable State* (stagnacji – kiepskim, stabilnym stanie). Zbyt silny przeciwnik nie pozwoli rozróżnić przeciętnego i złego rozwiązania; zbyt słaby – przeciętnego i dobrego. Ocena każdego zależy od innych (zewnętrzny, obiektywny „nauczyciel” rozwiązuje ten problem równocześnie eliminując zalety koewolucji). Ocena strategii może być nieprzechodnia.

Koewolucja konkurencyjna – problemy i środki zaradcze

Koewolucja koopera-tywna

Koewolucja konkuren-cyjna

Bibliografia

- Dyskusja przykładowych sytuacji: GP (populacja wyrażeń i populacja testów), strategie gry w szachy, piłka nożna, tenis i nieprzechodniość „lepszości”, papier–kamień–nożyce, lokalna szkoła szermierki i różnorodność (por. *exploiter agents* w [AlphaStar](#)), natura.

*https://en.wikipedia.org/wiki/Red_Queen_hypothesis

Koewolucja konkurencyjna – problemy i środki zaradcze

Koewolucja koopera-tywna

Koewolucja konkuren-cyjna

Bibliografia

- Dyskusja przykładowych sytuacji: GP (populacja wyrażeń i populacja testów), strategie gry w szachy, piłka nożna, tenis i nieprzechodniość „lepszości”, papier–kamień–nożyce, lokalna szkoła szermierki i różnorodność (por. *exploiter agents* w [AlphaStar](#)), natura.
- Pojęcia: arms race, Red Queen*, MSS.

*https://en.wikipedia.org/wiki/Red_Queen_hypothesis

- Dyskusja przykładowych sytuacji: GP (populacja wyrażeń i populacja testów), strategie gry w szachy, piłka nożna, tenis i nieprzechodniość „lepszości”, papier–kamień–nożyce, lokalna szkoła szermierki i różnorodność (por. *exploiter agents* w [AlphaStar](#)), natura.
- Pojęcia: arms race, Red Queen*, MSS.
- Problemy: brak lub utrata gradientu, zapętlenie (cykle, nieprzechodniość relacji porównania wynikającej z oceniania), brak monotoniczności (postępu) [Mic09].

*https://en.wikipedia.org/wiki/Red_Queen_hypothesis

- Dyskusja przykładowych sytuacji: GP (populacja wyrażeń i populacja testów), strategie gry w szachy, piłka nożna, tenis i nieprzechodniość „lepszości”, papier–kamień–nożyce, lokalna szkoła szermierki i różnorodność (por. *exploiter agents* w [AlphaStar](#)), natura.
- Pojęcia: arms race, Red Queen*, MSS.
- Problemy: brak lub utrata gradientu, zapętlenie (cykle, nieprzechodniość relacji porównania wynikającej z oceniania), brak monotoniczności (postępu) [Mic09].
- Rady: *competitive fitness sharing* (zwiększenie oceny tych rozwiązań, które wygrywają z testami (przeciwnikami) trudnymi dla pozostałych rozwiązań [RB95]), specjalny dobór zbioru testów, utrzymywanie *hall of fame* lub zbiorów Pareto-niezdominowanych rozwiązań i testów.

*https://en.wikipedia.org/wiki/Red_Queen_hypothesis

Bibliografia |

Koewolucja koopera-tywna

Koewolucja konkuren-cyjna

Bibliografia

- [Elf+21] Ehab Z. Elfeky i in. "A systematic review of coevolution in real-time strategy games". W: *IEEE Access* 9 (2021), s. 136647–136665. URL: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9548932>.
- [Mic09] Thomas Miconi. "Why coevolution doesn't "work": superiority and progress in coevolution". W: *12th European Conference on Genetic Programming – EuroGP*. Springer, 2009, s. 49–60. URL: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=c51f7914019c020d2386b4880cdc59b22dbbd7c4>.
- [PD00] Mitchell A. Potter i Kenneth A. De Jong. "Cooperative coevolution: an architecture for evolving coadapted subcomponents". W: *Evolutionary computation* 8.1 (mar. 2000), s. 1–29. DOI: [10.1162/106365600568086](https://doi.org/10.1162/106365600568086). URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.35.5861&rep=rep1&type=pdf>.
- [RB95] Christopher Rosin i Richard Belew. "Methods for competitive co-evolution: finding opponents worth beating". W: *Proceedings of the Sixth International Conference on Genetic Algorithms*. Morgan Kaufmann, 1995, s. 373–380. URL: [https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.52.9359&rep=rep1&type=pdf](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.52.9359&rep=rep1&type=pdf).



**Fundusze
Europejskie**
Polska Cyfrowa



**Rzeczpospolita
Polska**

Unia Europejska
Europejski Fundusz
Rozwoju Regionalnego



Algorytmy i modele inspirowane biologicznie

7. Projektowanie ewolucyjne

Maciej Komosiński



**Fundusze
Europejskie**
Polska Cyfrowa



**Rzeczpospolita
Polska**

Unia Europejska
Europejski Fundusz
Rozwoju Regionalnego



Jak reprezentować rozwiązania w ED (*evolutionary design*)?

Przykłady

Przyczyny
trudności

Rodzaje

Genotyp a
fenotyp

Bibliografia

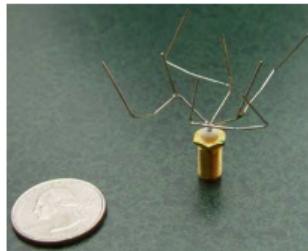
Projektowanie ewolucyjne jest szczególnym przypadkiem [automatyzacji projektowania](#).

Optymalizowane konstrukcje mogą być pasywne (statyczne) albo aktywne (wyposażone w siłowniki–efektory, a czasem również sensory). Jednym z przykładów ED jest zatem [robotyka ewolucyjna](#).

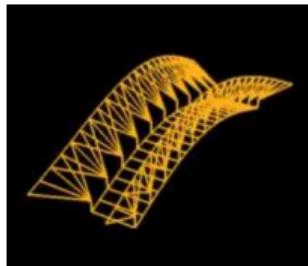
Wymyśl kilka reprezentacji genetycznych do optymalizacji mostu.

Przykłady projektowania ewolucyjnego (1/2)

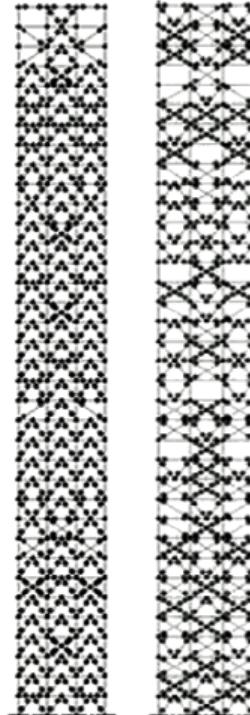
Przykłady
Przyczyny
trudności
Rodzaje
Genotyp a
fenotyp
Bibliografia



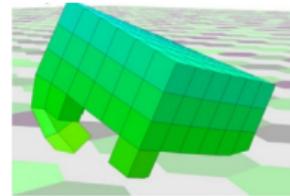
Automated Antenna Design with Evolutionary Algorithms,
G. Hornby et al., 2006



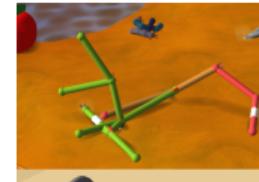
Combining Structural Analysis and Multi-Objective Criteria for Evolutionary Architectural Design,
J. Byrne et al., 2011



Evolutionary Design of Steel Structures in Tall Buildings, R. Kicinger et al., 2005



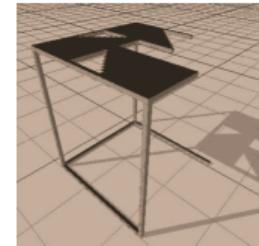
Evolutionary Developmental Soft Robotics As a Framework to Study Intelligence and Adaptive Behavior in Animals and Plants,
F. Corucci, 2017



Framsticks [KU24]



Evolving virtual creatures,
K. Sims [Sim94]



Generative representations,
G. Hornby [Hor03]

Przykłady projektowania ewolucyjnego (2/2)

Przykłady

Przyczyny
trudności

Rodzaje

Genotyp a
fenotyp

Bibliografia

- Wczesne ewoluowane roboty
<https://youtu.be/tgJcYx-yewA?t=237>
- Optymalizacja turbin wiatrowych
<https://www.youtube.com/watch?v=cNaFhhwTpS8>
- ...i wodnych (na co zwracać uwagę, co jest celem)
<https://youtu.be/fcE6HV1g2kk?t=2477>
- Optymalizacja aerodynamiki samochodu
https://www.youtube.com/watch?v=sw7_XWdd56c&t=25



- Optymalizacja struktury samochodu
Czinger 21C: “Using supercomputing and AI (...) the chassis structure is generatively designed. Every component of the structure is pareto optimized for its precise function, not a single gram of material goes to waste.”

<https://youtu.be/Pppne2jcgok?t=1541>

Porównanie trudności optymalizacji

Porównajmy złożoność klasycznego permutacyjnego problemu optymalizacji i problemu projektowania konstrukcji:

Porównanie trudności optymalizacji

Porównajmy złożoność klasycznego permutacyjnego problemu optymalizacji i problemu projektowania konstrukcji:

Porównanie trudności optymalizacji

Porównajmy złożoność klasycznego permutacyjnego problemu optymalizacji i problemu projektowania konstrukcji:

Porównanie trudności optymalizacji

Porównajmy złożoność klasycznego permutacyjnego problemu optymalizacji i problemu projektowania konstrukcji:

Porównanie trudności optymalizacji

Porównajmy złożoność klasycznego permutacyjnego problemu optymalizacji i problemu projektowania konstrukcji:

Porównanie trudności optymalizacji

Porównajmy złożoność klasycznego permutacyjnego problemu optymalizacji i problemu projektowania konstrukcji:

Porównanie trudności optymalizacji

Porównajmy złożoność klasycznego permutacyjnego problemu optymalizacji i problemu projektowania konstrukcji:

Porównanie trudności optymalizacji

Porównajmy złożoność klasycznego permutacyjnego problemu optymalizacji i problemu projektowania konstrukcji:

Własność	QAP/TSP	Optymalizacja konstrukcji
Skończony zbiór rozwiązań		
Przestrzeń dyskretno-ciągła		
Genotyp o stałej długości		
Oczywista, naturalna reprezentacja		
Prosta koncepcja sąsiedztwa		
Wiele optimów lokalnych		
Silne związki między fragmentami rozwiązania		
Liczne ograniczenia		

Przykłady

Przyczyny
trudności

Rodzaje

Genotyp a
fenotyp

Bibliografia

Porównanie trudności optymalizacji

Porównajmy złożoność klasycznego permutacyjnego problemu optymalizacji i problemu projektowania konstrukcji:

Porównanie trudności optymalizacji

Porównajmy złożoność klasycznego permutacyjnego problemu optymalizacji i problemu projektowania konstrukcji:

Porównanie trudności optymalizacji

Porównajmy złożoność klasycznego permutacyjnego problemu optymalizacji i problemu projektowania konstrukcji:

Własność	QAP/TSP	Optymalizacja konstrukcji
Skończony zbiór rozwiązań		
Przestrzeń dyskretno-ciągła		
Genotyp o stałej długości		
Oczywista, naturalna reprezentacja		
Prosta koncepcja sąsiedztwa		
Wiele optimów lokalnych		
Silne związki między fragmentami rozwiązania		
Liczne ograniczenia		
Wiele kryteriów oceniających		
Kryteria trudno sformalizować		
Determinizm oceny		

Porównanie trudności optymalizacji

Przykłady

Przyczyny
trudności

Rodzaje

Genotyp a
fenotyp

Bibliografia

Porównajmy złożoność klasycznego permutacyjnego problemu optymalizacji i problemu projektowania konstrukcji:

Właściwość	QAP/TSP	Optymalizacja konstrukcji
Skończony zbiór rozwiązań		
Przestrzeń dyskretno-ciągła		
Genotyp o stałej długości		
Oczywista, naturalna reprezentacja		
Prosta koncepcja sąsiedztwa		
Wiele optimów lokalnych		
Silne związki między fragmentami rozwiązania		
Liczne ograniczenia		
Wiele kryteriów oceniających		
Kryteria trudno sformalizować		
Determinizm oceny		
Ocena obejmuje aspekt czasowy		

Porównanie trudności optymalizacji

Przykłady

Przyczyny
trudności

Rodzaje

Genotyp a
fenotyp

Bibliografia

Porównajmy złożoność klasycznego permutacyjnego problemu optymalizacji i problemu projektowania konstrukcji:

Właściwość	QAP/TSP	Optymalizacja konstrukcji
Skończony zbiór rozwiązań		
Przestrzeń dyskretno-ciągła		
Genotyp o stałej długości		
Oczywista, naturalna reprezentacja		
Prosta koncepcja sąsiedztwa		
Wiele optimów lokalnych		
Silne związki między fragmentami rozwiązania		
Liczne ograniczenia		
Wiele kryteriów oceniających		
Kryteria trudno sformalizować		
Determinizm oceny		
Ocena obejmuje aspekt czasowy		
Czasochłonna ocena		

Porównanie trudności optymalizacji

Przykłady

Przyczyny
trudności

Rodzaje

Genotyp a
fenotyp

Bibliografia

Porównajmy złożoność klasycznego permutacyjnego problemu optymalizacji i problemu projektowania konstrukcji:

Właściwość	QAP/TSP	Optymalizacja konstrukcji
Skończony zbiór rozwiązań		
Przestrzeń dyskretno-ciągła		
Genotyp o stałej długości		
Oczywista, naturalna reprezentacja		
Prosta koncepcja sąsiedztwa		
Wiele optimów lokalnych		
Silne związki między fragmentami rozwiązania		
Liczne ograniczenia		
Wiele kryteriów oceniających		
Kryteria trudno sformalizować		
Determinizm oceny		
Ocena obejmuje aspekt czasowy		
Czasochłonna ocena		
Przewidywalny czas oceny		

Porównanie trudności optymalizacji

Przykłady

Przyczyny
trudności

Rodzaje

Genotyp a
fenotyp

Bibliografia

Porównajmy złożoność klasycznego permutacyjnego problemu optymalizacji i problemu projektowania konstrukcji:

Właściwość	QAP/TSP	Optymalizacja konstrukcji
Skończony zbiór rozwiązań		
Przestrzeń dyskretno-ciągła		
Genotyp o stałej długości		
Oczywista, naturalna reprezentacja		
Prosta koncepcja sąsiedztwa		
Wiele optimów lokalnych		
Silne związki między fragmentami rozwiązania		
Liczne ograniczenia		
Wiele kryteriów oceniających		
Kryteria trudno sformalizować		
Determinizm oceny		
Ocena obejmuje aspekt czasowy		
Czasochłonna ocena		
Przewidywalny czas oceny		
Proste szacowanie podobieństwa		

Poziom szczegółowości w projektowaniu ewolucyjnym

Przykłady
Przyczyny trudności
Rodzaje
Genotyp a fenotyp
Bibliografia

- Koncepcyjne projektowanie ewolucyjne: tworzenie wysokopoziomowych ram koncepcyjnych dla projektów. Nowe koncepcje projektowe można optymalizować, ale ich elementy są dostarczane przez projektanta. Przykład: system hydroenergetyczny jako kombinacja lokalizacji, typów zapór, długości tuneli i trybów pracy.
- Generatywne projektowanie ewolucyjne: bezpośrednie generowanie projektu. Brak z góry zdefiniowanych wysokopoziomowych pojęć, brak konwencji i narzuconej wiedzy ([efektu nastawienia](#)). Zdefiniowane jedynie niskopoziomowe bloki konstrukcyjne. Złożone reprezentacje. Przykłady: stoły, radiatory, pryzmaty optyczne, modele aerodynamiczne i hydrodynamiczne, mosty, dźwigi, [EHW](#), obwody analogowe.

Efekt nastawienia oraz wytwory ludzkie a naturalne

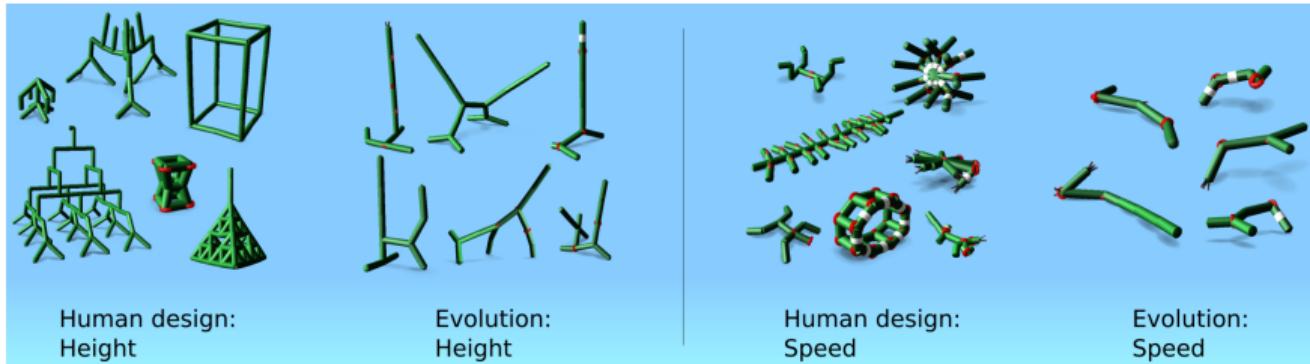
Z materiałów 7th Int. Conf. on Swarm Intelligence, sesja o *Morphogenetic Engineering*:

Konstrukcje zaprojektowane:

- często wykonane z wielu unikatowych elementów składowych połączonych w precyzyjny, skomplikowany sposób,
- działają deterministycznie zgodnie ze specyfikacją daną przez projektantów.

Z kolei (por. rysunek poniżej) samoorganizacja w systemach naturalnych (fizycznych, biologicznych, ekologicznych, społecznych):

- często opiera się na powielaniu identycznych agentów i stochastycznej dynamice,
- nietrywialne zachowanie może wynikać z względnie prostych reguł,
- jednak większość naturalnych wzorców można opisać za pomocą niewielkiej liczby parametrów statystycznych,
- wzorce te są przypadkowe albo ukształtowane przez warunki brzegowe, ale nigdy nie wykazują wewnętrznej architektury, którą mają zaprojektowane produkty.



Embriogeneza w ED

Przykłady

Przyczyny
trudności

Rodzaje

Genotyp a
fenotyp

Bibliografia

W projektowaniu ewolucyjnym fenotypy zwykle znacznie bardziej różnią się od ich reprezentacji genotypowych niż w typowych problemach optymalizacji. Oznacza to, że jest wymagane mapowanie genotypu na fenotyp („embriogeneza”) i ono może być skomplikowane – rozmawialiśmy o tym omawiając programowanie ewolucyjne.

Celem jest wysoka **skalowalność** (zdolność do rozbudowy i tworzenia bardziej złożonych struktur [Hor08]) i **ewoluowalność** (zdolność do tworzenia potomstwa, które jest różnorodne/lepiej oceniane [Gaj+19]) – rozważmy przykład optymalizacji *szczoteczki do zębów* [dyskusja].

Embriogeneza w ED

Przykłady

Przyczyny
trudności

Rodzaje

Genotyp a
fenotyp

Bibliografia

W projektowaniu ewolucyjnym fenotypy zwykle znacznie bardziej różnią się od ich reprezentacji genotypowych niż w typowych problemach optymalizacji. Oznacza to, że jest wymagane mapowanie genotypu na fenotyp („embriogeneza”) i ono może być skomplikowane – rozmawialiśmy o tym omawiając programowanie ewolucyjne.

Celem jest wysoka **skalowalność** (zdolność do rozbudowy i tworzenia bardziej złożonych struktur [Hor08]) i **ewoluwalność** (zdolność do tworzenia potomstwa, które jest różnorodne/lepiej oceniane [Gaj+19]) – rozważmy przykład optymalizacji *szczoteczki do zębów* [dyskusja].

Te same pożądane właściwości mapowania genotyp-fenotyp, które zidentyfikowaliśmy w przykładzie szczoteczki, występują przy optymalizacji mostów, turbin, robotów, samochodów, itd.

Mapowanie genotyp–fenotyp: natura a ED

Przykłady
Przyczyny trudności
Rodzaje
Genotyp a fenotyp
Bibliografia

W naturze embriogeneza jest definiowana przez interakcje między genami, ich efekty fenotypowe i środowisko, w którym zarodek się rozwija. Istnieją łańcuchy oddziałujących na siebie „reguły”; ciąg ich aktywacji nie jest całkowicie określony z góry i zaprogramowany; jest dynamiczny, równoległy i adaptacyjny.*

W optymalizacji (np. w projektowaniu ewolucyjnym) embriogenezy mogą być [Ben99]:

- Zewnętrzne (nie ewoluowane). Stałe reguły określające sposób konstruowania fenotypów z genotypów. Np. $f0$, $f1$, fH , $f7$ i $f9$ we Framsticks [KU21].

*<https://nautil.us/the-strange-inevitability-of-evolution-235189/>

Mapowanie genotyp–fenotyp: natura a ED

Przykłady
Przyczyny trudności
Rodzaje
Genotyp a fenotyp
Bibliografia

W naturze embriogeneza jest definiowana przez interakcje między genami, ich efekty fenotypowe i środowisko, w którym zarodek się rozwija. Istnieją łańcuchy oddziałujących na siebie „reguły”; ciąg ich aktywacji nie jest całkowicie określony z góry i zaprogramowany; jest dynamiczny, równoległy i adaptacyjny.*

W optymalizacji (np. w projektowaniu ewolucyjnym) embriogenezy mogą być [Ben99]:

- Zewnętrzne (nie ewoluowane). Stałe reguły określające sposób konstruowania fenotypów z genotypów. Np. $f0$, $f1$, fH , $f7$ i $f9$ we Framsticks [KU21].
- Jawne (ewoluowane). Genotyp i embriogeneza ewoluują równocześnie, ale embriogeneza składa się z ustalonych bloków/cech – takich jak iterowanie, rekurencja itp., podobnie jak w GP (programowaniu genetycznym). Często potrzebne są specjalizowane operatory i reprezentacje. Np. $f4$ we Framsticks.

*<https://nautil.us/the-strange-inevitability-of-evolution-235189/>

Mapowanie genotyp–fenotyp: natura a ED

Przykłady
Przyczyny trudności
Rodzaje
Genotyp a fenotyp
Bibliografia

W naturze embriogeneza jest definiowana przez interakcje między genami, ich efekty fenotypowe i środowisko, w którym zarodek się rozwija. Istnieją łańcuchy oddziałujących na siebie „reguły”; ciąg ich aktywacji nie jest całkowicie określony z góry i zaprogramowany; jest dynamiczny, równoległy i adaptacyjny.*

W optymalizacji (np. w projektowaniu ewolucyjnym) embriogenezy mogą być [Ben99]:

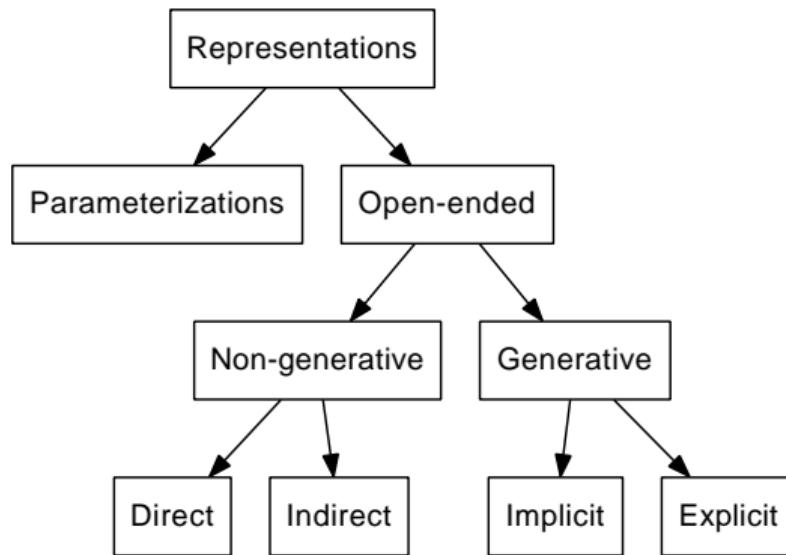
- Zewnętrzne (nie ewoluowane). Stałe reguły określające sposób konstruowania fenotypów z genotypów. Np. $f0$, $f1$, fH , $f7$ i $f9$ we Framsticks [KU21].
- Jawne (ewoluowane). Genotyp i embriogeneza ewoluują równocześnie, ale embriogeneza składa się z ustalonych bloków/cech – takich jak iterowanie, rekurencja itp., podobnie jak w GP (programowaniu genetycznym). Często potrzebne są specjalizowane operatory i reprezentacje. Np. $f4$ we Framsticks.
- Niejawne (ewoluowane). Te same geny mogą być wielokrotnie aktywowane i tłumione oraz mogą określać różne funkcje. Dozwolona jest iteracja, warunki, podprogramy, równoległe interpretowanie genów. Jednak bardzo trudno jest zaprojektować dobrą niejawną reprezentację. Np. fB , $f6$ i fL we Framsticks.

*<https://nautil.us/the-strange-inevitability-of-evolution-235189/>

Mapowanie genotyp–fenotyp: klasyfikacja

Przykłady
Przyczyny trudności
Rodzaje
Genotyp a fenotyp
Bibliografia

Inna, podobna klasyfikacja embriogenez [Hor03]:

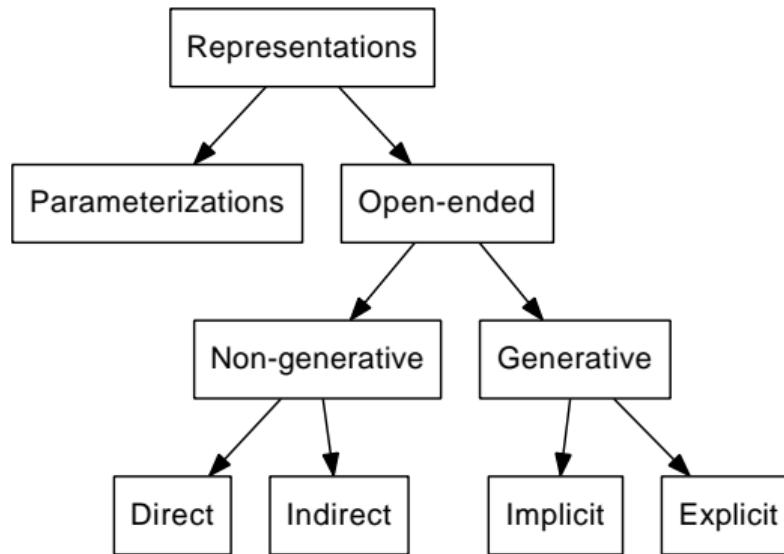


W reprezentacjach niegeneratywnych każdy gen jest aktywowany raz. W bezpośrednich (*direct*) i jawnych (*explicit*) znaczeniu genów jest stałe (nie podlega ewolucji).

Mapowanie genotyp–fenotyp: klasyfikacja

Przykłady
Przyczyny trudności
Rodzaje
Genotyp a fenotyp
Bibliografia

Inna, podobna klasyfikacja embriogenez [Hor03]:



W reprezentacjach niegeneratywnych każdy gen jest aktywowany raz. W bezpośrednich (*direct*) i jawnych (*explicit*) znaczeniu genów jest stałe (nie podlega ewolucji).

Pytanie: do jakiej kategorii trafią: permutacja w TSP, DNA w naturze, *f9* w ED?

Automatyczne tworzenie mapowania genotyp–fenotyp

Przykłady

Przyczyny
trudności

Rodzaje

Genotyp a
fenotyp

Bibliografia

Sam rozwój skutecznej embriogenezy/mapowania może być postawiony jako problem optymalizacji lub uczenia maszynowego („znajdź reprezentację, która daje łagodny krajobraz przystosowania: maksymalizuj FDC” albo „znajdź reprezentację, która sprawia, że podobne fenotypy są genetycznymi sąsiadami”).

Można go rozwiązywać za pomocą technik podobnych do osadzania (zanurzania) słów w nowej przestrzeni* albo (neuronowych) autoenkoderów** [KKM21].

*https://en.wikipedia.org/wiki/Word_embedding

**<https://en.wikipedia.org/wiki/Autoencoder>

Bibliografia |

Przykłady
Przyczyny trudności
Rodzaje
Genotyp a fenotyp
Bibliografia

- [Ben99] Peter Bentley. *Evolutionary design by computers*. Morgan Kaufmann, 1999.
- [Gaj+19] Alexander Gajewski i in. "Evolvability ES: scalable and direct optimization of evolvability". W: *Proceedings of the Genetic and Evolutionary Computation Conference*. 2019, s. 107–115. URL: <https://arxiv.org/pdf/1907.06077.pdf>.
- [Hor03] Gregory S. Hornby. "Creating complex building blocks through generative representations". W: *Proceedings of the 2003 AAAI Spring Symposium: Computational Synthesis: From Basic Building Blocks to High Level Functionality*. 2003, s. 98–105. URL: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.323.8779&rep=rep1&type=pdf>.
- [Hor08] Gregory S. Hornby. "Improving the scalability of generative representations for openended design". W: *Genetic Programming Theory and Practice V* (2008), s. 125–142.
- [KKM21] Piotr Kaszuba, Maciej Komosinski i Agnieszka Mensfelt. "Automated development of latent representations for optimization of sequences using autoencoders". W: *2021 IEEE Congress on Evolutionary Computation (CEC)*. IEEE. 2021, s. 1123–1130. DOI: [10.1109/CEC45853.2021.9504910](https://doi.org/10.1109/CEC45853.2021.9504910). URL: <http://www.framsticks.com/files/common/LatentRepresentationsForSequencesOptimization.pdf>.
- [KU21] Maciej Komosinski i Szymon Ulatowski. *Genetic representations in Framsticks*. http://www.framsticks.com/a/al_genotype. 2021.
- [KU24] Maciej Komosinski i Szymon Ulatowski. *Framsticks website*. 2024. URL: <http://www.framsticks.com>.
- [Sim94] Karl Sims. "Evolving virtual creatures". W: *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*. ACM. 1994, s. 15–22. URL: <https://www.cs.drexel.edu/~david/Classes/Papers/p15-sims.pdf>.



**Fundusze
Europejskie**
Polska Cyfrowa



**Rzeczpospolita
Polska**

Unia Europejska
Europejski Fundusz
Rozwoju Regionalnego

