

# Algorytmy ewolucyjne

---

Piotr Lipiński

## Klasyczne algorytmy ewolucyjne

---

- Klasyczne algorytmy ewolucyjne
  - algorytmy genetyczne
    - zwykłe przestrzeń poszukiwań to  $\{0, 1\}^d$
    - niektóre wersje działają na przestrzeni poszukiwań złożonej z permutacji
  - strategie ewolucyjne
    - przestrzeń poszukiwań to  $R^d$
    - występuje mechanizm auto-adaptacji
  - programowanie genetyczne
    - przestrzeń poszukiwań to zbiór drzew
  - programowanie ewolucyjne
    - przestrzeń poszukiwań to wyrażenia pewnej gramatyki

## Algorytmy genetyczne

- Poznane dotychczas algorytmy genetyczne:
  - PBIL
  - CGA
  - SSGA
  - SGA
- Algorytm SGA często dostosowuje się do konkretnych problemów przez wprowadzenie nowych operatorów ewolucyjnych, które pozwalają zmienić działanie algorytmu i zwiększyć jego efektywność.
- Algorytm SGA można łatwo dostosować do optymalizowania funkcji celu zadanych na przestrzeniach poszukiwań innych niż  $\Omega = \{0, 1\}^d$ .
  - Zazwyczaj wystarczy zmienić jedynie operatory krzyżowania i mutacji.
- W praktyce rzadko kiedy stosuje się oryginalny algorytm SGA bez wprowadzania zmian dostosowujących ten algorytm do konkretnego problemu praktycznego.

## Popularne benchmarki

- Proste popularne benchmarki dla algorytmów genetycznych na przestrzeni poszukiwań  $\Omega = \{0, 1\}^d$ :
  - OneMax – liczba jedynek w wektorze binarnym,
  - Pattern – liczba pozycji wektora binarnego zgodna z ustaloną wzorcem,
  - DeceptiveOneMax – jeśli wektor składa się z samych zer, wartością funkcji jest  $d+1$ , w przeciwnym przypadku wartością funkcji jest liczba jedynek,
  - K-DeceptiveOneMax – wektor binarny dzieli się na bloki długości  $K$ , na każdym bloku liczy się funkcję DeceptiveOneMax, ostatecznie funkcja zwraca sumę uzyskanych wyników ( $K$  powinno dzielić  $d$  bez reszty).
- Przykłady (dla  $d = 8$ ):
  - OneMax(00100100) = 2
  - Pattern(00100100) = 6 (dla ustalonego wzorca 01100110)
  - DeceptiveOneMax(00100100) = 2
  - DeceptiveOneMax(00000000) = 9
  - 4-DeceptiveOneMax(0010 0100) = 2 (odstęp pokazuje podział wektora)
  - 4-DeceptiveOneMax(0000 1111) = 9 (odstęp pokazuje podział wektora)
  - 2-DeceptiveOneMax(00 10 01 00) = 8 (odstęp pokazuje podział wektora)

## Operatory ewolucyjne – selekcja

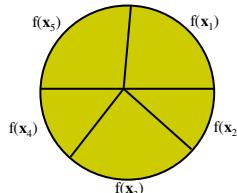
- W niektórych podejściach selekcji nie uważa się za operator przeszukiwania, mimo że ma bardzo duży wpływ na działanie algorytmu.
- Selekcja może być zastosowana zarówno przed jak i po zastosowaniu pozostałych operatorów przeszukiwania.
- Kiedy selekcja jest stosowana przed użyciem innych operatorów przeszukiwania, proces konstrukcji następnej populacji przy tych operatorów przeszukiwania zwany jest reprodukcją.
- Generational EA – algorytm, w którym wymianie podlega większość populacji.
- Steady-state EA – algorytm, w którym wymianie podlegają nieliczne osobniki.
- Generational Gap – w niektórych algorytmach pewne osobniki przechodzą do następnego pokolenia bez zastosowania żadnych operatorów przeszukiwania, zjawisko takie zwane jest Generational Gap.
- Elitism – w niektórych algorytmach najlepsze osobniki z bieżącego pokolenia są automatycznie włączane do następnego pokolenia, zjawisko takie zwane jest Elitism.

## Selekcja według funkcji przystosowania

- Rozpatrujemy problem maksymalizacji funkcji celu  $F$  na przestrzeni poszukiwań  $\Omega$ .
- Niech  $P = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$  będzie populacją złożoną z  $N$  osobników. Dla każdego osobnika  $\mathbf{x}_i$  można policzyć wartość funkcji celu  $F(\mathbf{x}_i)$ .
- Sama wartość funkcji celu jest trudna do interpretacji kiedy nie znamy prawdziwych minimów i maksimów funkcji celu (trudno powiedzieć czy osobnik jest dobry czy nie).
- Jednym z pomysłów może być rozpatrywanie wartości funkcji celu danego osobnika w porównaniu do wartości funkcji celu innych osobników w populacji.
- Pierwsze podejście:
  - Zamiast  $F(\mathbf{x}_i)$  rozpatrujmy: 
$$\frac{F(\mathbf{x}_i)}{\sum_{j=1}^N F(\mathbf{x}_j)}$$
  - Problem dla funkcji celu przyjmującej ujemne wartości.
  - Jeżeli różnica między maksymalną i minimalną wartością funkcji celu w populacji jest niewielka, to osobniki w populacji będą miały zbliżone wartości proponowanej funkcji.
- Drugie podejście:
  - Zamiast  $F(\mathbf{x}_i)$  rozpatrujmy: 
$$\frac{F(\mathbf{x}_i) - F_{\min}}{\sum_{j=1}^N (F(\mathbf{x}_j) - F_{\min})}$$
 gdzie  $F_{\min}$  to minimalna wartość funkcji celu w danej populacji.
  - Wartość tę nazywamy wartością funkcji przystosowania (ang. fitness function) osobnika  $\mathbf{x}_i$  w populacji  $P$  i oznaczamy przez  $f(\mathbf{x}_i)$ .
  - UWAGA: Wartość przystosowania osobnika zależy od populacji, w kontekście której jest rozpatrywany!

## Selekcja według funkcji przystosowania

- Selekcja według funkcji przystosowania (zwana metodą ruletki):
  - z populacji P wybiera się losowo osobnika z prawdopodobieństwami wyboru poszczególnych osobników zadawanymi przez ich wartości przystosowania
  - (wartości przystosowania osobników w populacji są zawsze z przedziału [0, 1] i sumują się do 1 w całej populacji, więc mogą być używane jako prawdopodobieństwa wyboru poszczególnych osobników)



### □ PROBLEMY:

- (techniczny) mianownik ułamka może czasami być zerem
  - można wówczas przyjąć, że przystosowanie każdego osobnika jest takie samo i wynosi 1/N
- (merytoryczny) jeśli w populacji będzie osobnik o wartości funkcji celu znacznie przewyższającej wartości funkcji celu pozostałych osobników, to losowany będzie głównie on (tzw. dominacja super osobników)
  - problem często występuje w początkowych iteracjach algorytmów ewolucyjnych
  - rozwiązaniem może być skalowanie wartości funkcji celu lub przystosowania
  - rozwiązaniem może być też zastosowanie innej metody selekcji (m.in. selekcji rankingowej)

## Selekcja według funkcji przystosowania

### □ Skalowanie wartości funkcji celu:

- Simple Scaling

$$F_{scaled}(x_i) = F_{original}(x_i) - F_{min}$$

gdzie  $F_{min}$  to minimalna wartość funkcji celu znaleziona dotychczas.

- Sigma Scaling

$$F_{scaled}(x_i) = \max(0, F_{original}(x_i) - (F_{mean} - c\sigma))$$

gdzie  $F_{mean}$  to średnia wartość funkcji w aktualnej populacji,  $\sigma$  to odchylenie standardowe, a  $c$  to pewna stała.

- Power Scaling

$$F_{scaled}(x_i) = (F_{original}(x_i))^k$$

dla pewnej stałej  $k > 0$ .

- Exponential Scaling

$$F_{scaled}(x_i) = \exp(F_{original}(x_i)/T)$$

gdzie  $T > 0$  to pewien parametr malejący w kolejnych iteracjach.

## Selekcja rankingowa

- W odróżnieniu od selekcji według funkcji przystosowania, selekcja rankingowa nie przywiązuje wagi do wielkości różnic między wartościami funkcji celu poszczególnych osobników w populacji, a opiera się jedynie na rankingu osobników.
- Ranking polega na posortowaniu osobników z populacji według rosnącej wartości funkcji celu (0 – osobnik najgorszy, N-1 – osobnik najlepszy).
- Linear Ranking – prawdopodobieństwo wyboru osobnika  $\mathbf{x}$  to:  
$$P_{\text{linear}}(\mathbf{x}) = \frac{\alpha + [\text{rank}(\mathbf{x})/(N-1)](\beta - \alpha)}{N}$$
gdzie  $0 < \alpha < \beta$  to parametry, takie że  $\alpha + \beta = 2$  (jakie jest ich znaczenie?).
- Power Ranking  
$$P_{\text{power}}(\mathbf{x}) = \frac{\alpha + [\text{rank}(\mathbf{x})/(N-1)]^k(\beta - \alpha)}{K}$$
gdzie  $0 < \alpha < \beta$  oraz  $k$  to parametry, a  $K$  to stała normalizująca.
- Geometric Ranking  
$$P_{\text{geom}}(\mathbf{x}) = \frac{\alpha(1-\alpha)^{N-1-\text{rank}(\mathbf{x})}}{K}$$
- Exponential Ranking  
$$P_{\text{exp}}(\mathbf{x}) = \frac{1-\exp(-\text{rank}(\mathbf{x}))}{K}$$

## Selekcja turniejowa

- Selekcja turniejowa:
  - z populacji wybierana jest losowo ustalona liczba  $K$  osobników z rozkładem jednostajnym (wybór każdego osobnika jest jednakowo prawdopodobny, nie zależy od jego wartości funkcji celu),
  - z wybranych  $K$  osobników wybierany jest najlepszy (zwycięzca turnieju),
  - wybrany zwycięzca turnieju dodawany jest do nowej populacji,
  - powyższe kroki są powtarzane aż do wypełnienia nowej populacji (wylosowania odpowiedniej liczby osobników).
- Jakie znaczenie ma ustalona liczba  $K$ ? Jak wpływa na prawdopodobieństwa wyboru poszczególnych osobników?
- Przykład/Zadanie:
  - Rozważamy problem optymalizacji OneMax o długości 4.
  - Bieżąca populacja składa się z 8 osobników: 0000, 1000, 0100, 0010, 0001, 0011, 0111 i 1111.
  - Jakie jest prawdopodobieństwo, że w selekcji turniejowej z  $K = 4$  w pierwszej iteracji wylosujemy osobnika 0000? A jakie, że 0111?

## Operatory ewolucyjne – rekombinacja

- Operatory rekombinacji są zazwyczaj dwóch rodzajów: krzyżowania i mutacji.
- W zależności od konkretnego algorytmu ich działanie i znaczenie może być różne.
- O ile operatory selekcji są dość uniwersalne i nie zależy od rodzaju przestrzeni poszukiwań, o tyle operatory rekombinacji muszą odpowiadać kodowaniu elementów przestrzeni poszukiwań.
  
- Operatory rekombinacji dla problemów na przestrzeni poszukiwań  $\Omega = \{0, 1\}^d$ :
  - No Recombination (kopianie chromosomu od losowo wybranego rodzica)
  - Uniform Crossover (występuje w SSGA)
  - Multi-Point Recombination (występuje w SGA)

## Operatory ewolucyjne – rekombinacja

- Operatory rekombinacji dla problemów na przestrzeni poszukiwań  $\Omega = \mathbb{R}^d$ :
  - No Recombination (kopianie chromosomu od losowo wybranego rodzica)
  - Rekombinacje dyskretne (nie zmieniają wartości poszczególnych genów)
    - Multi-Point Recombination (analogicznie jak w przypadku binarnym)
    - Global Discrete Recombination (analogicznie do krzyżowania jednostajnego w przypadku binarnym)
  - Global Intermediate Recombination
    - Osobnik potomny jest średnią arytmetyczną wszystkich swoich rodziców.
  - Local Intermediate Recombination
    - Osobnik potomny jest średnią arytmetyczną dwóch losowo wybranych swoich rodziców.
  - Intermediate Recombination
    - Osobnik potomny jest średnią ważoną wszystkich swoich rodziców, z ustalonimi wagami.
  - Geometric Recombination
    - Osobnik potomny jest średnią geometryczną wszystkich swoich rodziców
  - Heuristic Recombination
    - Osobnik potomny to
$$\mathbf{y} = u(\mathbf{x}_2 - \mathbf{x}_1) + \mathbf{x}_2$$
gdzie  $\mathbf{x}_1$  to najgorszy z jego rodziców,  $\mathbf{x}_2$  to najlepszy z jego rodziców, zaś  $u$  to liczba wygenerowana losowo z rozkładem  $N(0, 1)$ .
  - Simplex Recombination
    - Osobnik potomny to
$$\mathbf{y} = \mathbf{c} + (\mathbf{c} - \mathbf{x}_2)$$
gdzie  $\mathbf{x}_1$  to najgorszy z jego rodziców,  $\mathbf{x}_2$  to najlepszy z jego rodziców, a  $\mathbf{c}$  to środek ciężkości grupy rodziców z wyłączeniem rodzica  $\mathbf{x}_1$ .