

# Programy Ewolucyjne

Tadeusz Puźniakowski

PJATK

15 czerwca 2024

## Co na dzisiaj

Bardziej praktyczne podejście do rozwiązywania problemów za pomocą algorytmów inspirowanych naturą. Ponad to kilka uwag związanych z projektowaniem AG.

# Spis treści

## 1 Problemy Z AG

## 2 Przegląd popularnych operatorów genetycznych

- Metody selekcji
- Metody krzyżowania
- Metody mutacji

## 3 Programy ewolucyjne

- Przykład zastosowania algorytmu ewolucyjnego

## 4 Zakończenie

# O zbieżności

## Krzyżowanie – eksploracja

Pozwala na dotarcie do nowych obszarów poszukiwań (potencjalnie ciekawych).

## Krzyżowanie – eksploatacja

Działa jak optymalizacja lokalna – poprawia rozwiązania w niewielkim ich otoczeniu.

# O warunkach zakończenia

## Warunki zakończenia

- Liczba iteracji
- Liczba wykonień obliczania funkcji fitness
- Zadana jakość rozwiązań
- Zadana różnorodność (albo raczej jej brak) rozwiązań
- Czas obliczeń

# O zbieżności

## Przedwczesna zbieżność

Jest to utrata przez algorytm optymalizacyjny zdolności przeszukiwania przestrzeni rozwiązań przed osiągnięciem ekstremum globalnego

# O zbieżności

Skąd brak zbieżności?

- Kodowanie powoduje, że algorytm przeszukuje przestrzeń nie zawierającą optimum globalnego
- Liczba iteracji musi być skończona
- Liczebność populacji musi być skończona

# Superosobniki

## Superosobniki

Osobniki o bardzo wysokim stopniu przystosowania

- Przeszkadzają na początku algorytmu (przedwczesna zbieżność)
- Bardzo przydatne pod koniec działania algorytmu (zawężenie przestrzeni poszukiwań)

# Niepoprawne rozwiązania

Zdarza się, że osobniki po krzyżowaniu/mutacji są całkowicie niepoprawne. Co z tym zrobić?

## Metoda kary

Osobnik niepoprawny otrzymuje tak zwaną karę - oznacza to, że jego wartość dostosowania jest w znaczący sposób zmniejszana, a w skrajnym przypadku osobnik jest usuwany.

## Algorytm Naprawy

Osobnik niepoprawny jest poprawiany tak, aby mieścił się w założeniach zadania.

# Spis treści

- 1 Problemy Z AG
- 2 Przegląd popularnych operatorów genetycznych
  - Metody selekcji
  - Metody krzyżowania
  - Metody mutacji
- 3 Programy ewolucyjne
  - Przykład zastosowania algorytmu ewolucyjnego
- 4 Zakończenie

# Przypomnienie

Do tej pory poznaliśmy selekcję metodą ruletki (i wspomniałem o innych metodach). Teraz czas na inne podejścia.

# Podwójna selekcja turniejowa

## Podwójna\* selekcja turniejowa (selekcja turniejowa)



wylosuj osobniki  $s_1, s_2, \dots, s_n$  z  $P(t)$

wybierz najlepszego z nich

Często turnieje rozmiaru 2.

\* trzeba jej dokonać 2x, aby uzyskać parę rodziców.

# Podwójna selekcja turniejowa

## Podwójna selekcja turniejowa - przykład

Przykład na tablicy. Niech będzie 6 osobników i bardzo skrajne wartości funkcji przystosowania.

# Podwójna selekcja turniejowa

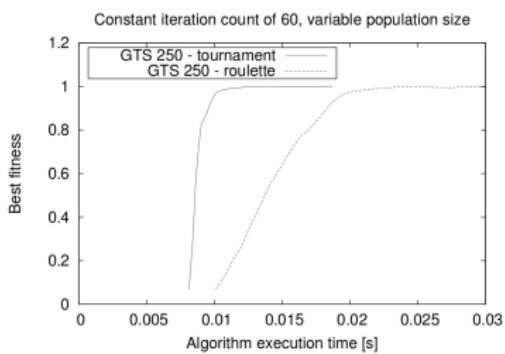
## Podwójna selekcja turniejowa – cechy

- świetnie nadaje się do maszyn wieloprocesorowych
- jest bardzo prosta w implementacji
- zapobiega przedwczesnej zbieżności algorytmu
- pozwala na przeszukanie szerszego zakresu potencjalnych rozwiązań nawet przy kiepskiej funkcji fitness

# Podwójna selekcja turniejowa

## Podwójna selekcja turniejowa vs metoda ruletki

Dla problemu OneMax okazuje się, że selekcja metodą turniejową daje lepsze rezultaty (na GPU).



# Pojedyncza selekcja turniejowa

## Pojedyncza selekcja turniejowa

mieszamy populację

dzielimy na grupy rozmiaru  $n$

z każdej grupy wybieramy 2 osobników o najlepszym  
przystosowaniu

wybcone osobniki posłużą jako rodzice w następnym etapie AG

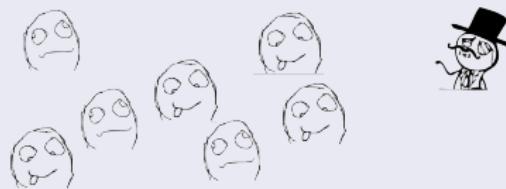
# Pojedyncza selekcja turniejowa

## Pojedyncza selekcja turniejowa – cechy

- gwarantuje że zawsze wybierzemy co najmniej  $n - 2$  najlepszych osobników
- nawet najlepiej przystosowane osobniki produkują jednego potomka (zapobiega przedwczesnej zbieżności)

# Selekcja elitarna

## Selekcja elitarna (z elitą)



Zawsze na początku wybierz  $n$  najlepszych osobników

# Selekcja elitarna

## Selekcja elitarna – cechy

- zapewnia, że zawsze zachowamy najlepszego osobnika
- przy niewielkiej elicie, zwykle nie ma problemu z przedwczesną zbieżnością
- przy większych elitach może się zdarzyć, że elita z początkowych kroków ewolucji zdominuje całą populację (czyli pewnie nie znajdziemy optimum globalnego)

# Selekcja rangowa

## Selekcja rangowa (porządkowa)

sortujemy osobniki wg funkcji przystosowania  
rangi osobników służą za funkcję przystosowania

# Metody krzyżowania

## Metody krzyżowania

Są na tyle proste, że tylko przypomnę (i szybko opiszę):

- jednopunktowe
- wielopunktowe
- jednostajne (zamieniamy geny z zadanym prawdopodobieństwem)

# Metody krzyżowania

## Metody mutacji

Teraz jeszcze krótki przegląd wybranych metod mutacji.

# Metody mutacji

## Mutacja punktowa

Polega na tym, że modyfikujemy pojedynczy losowy bit chromosomu.

# Metody mutacji

## Mutacja wielopunktowa

Polega na tym, że modyfikujemy  $n$  pojedynczych losowych bitów chromosomu.

# Metody mutacji

Mutacja probabilistyczna (ta z klasycznego AG)

Sprawdzamy każdy bit chromosomu i z zadanym prawdopodobieństwem zamieniamy go na przeciwny.

# Metody mutacji

## Mutacja użyteczna

Po wykonaniu mutacji (metoda dowolna) porównujemy nowego osobnika z rodzicem i wybieramy lepsze rozwiązanie.

# Metody mutacji

## Mutacja lamarckowska rzędu $k$

Sprawdzamy wyniki mutacji  $n$  punktowych dla  $0 \leq n \leq k$  i wybranie najlepszego rozwiązania.

# Spis treści

1 Problemy Z AG

2 Przegląd popularnych operatorów genetycznych

- Metody selekcji
- Metody krzyżowania
- Metody mutacji

3 Programy ewolucyjne

- Przykład zastosowania algorytmu ewolucyjnego

4 Zakończenie

## Co dalej?

Teraz zajmiemy się modyfikacjami klasycznego algorytmu genetycznego.

Dokładne trzymanie się zasad klasycznego algorytmu genetycznego nie zawsze jest konieczne.

# Program ewolucyjny

$t \leftarrow 0$

ustal początkowe  $P(t)$

ocen  $P(t)$

**while** NOT warunek zakończenia **do**

$t \leftarrow t + 1$

wybierz  $P(t)$  z  $P(t - 1)$

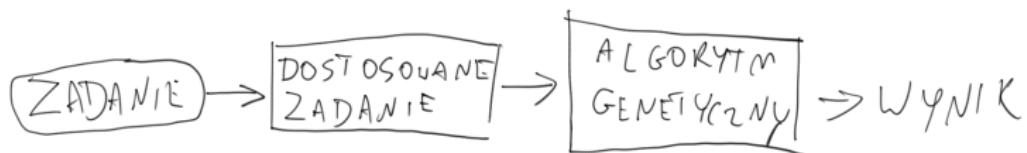
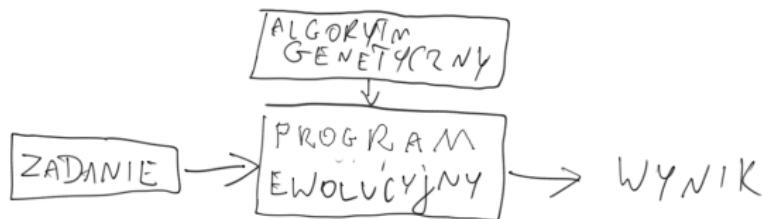
zmień  $P(t)$

ocen  $P(t)$

**end while**

(Widać pewne podobieństwo do algorytmów genetycznych?)

# Różne podejścia



# Różne podejścia

## Algorytm genetyczny

Uniwersalny sposób na rozwiązywanie problemów. Mamy zawsze taki sam algorytm, natomiast zadanie dostosowujemy do niego.

## Program ewolucyjny

Mamy ogólny schemat algorytmu i dostosowujemy go do wymagań problemu.

# Algorytm ewolucyjny

## Mała uwaga

Algorytm genetyczny można uznać za szczególny przypadek algorytmu ewolucyjnego.

# Program ewolucyjny

Ale po co nam to?



\*Screenshot z gry The Incredible Machine

# Program ewolucyjny

Ale po co nam to?

(SPOILER) Okazuje się, że takie podejście jest często znacznie tańsze.

Na następnych slajdach pokażę pewien przykład, gdzie tradycyjne, binarne chromosomy się nie sprawdzają.

# Komiwojażer

## Problem komiwojażera – definicja

Problem komiwojażera (TSP) jest to zagadnienie optymalizacyjne, polegające na znalezieniu minimalnego cyklu Hamiltona w pełnym grafie ważonym.

Cykł Hamiltona to taki cykl w grafie, w którym każdy wierzchołek grafu przechodzony jest tylko jeden raz (oprócz pierwszego wierzchołka).

## Problem komiwojażera – trochę jaśniej

Chcemy odwiedzić wszystkie miasta jak najniższym kosztem.

# Komiwojażer

## Problem komiwojażera - przykładowe zastosowania

- (oczywisty) wyznaczanie trasy turystycznej :)
- wyznaczenie optymalnego położenia kabli na jakimś terenie
- wiercenie otworów w płytach drukowanych

# Komiwojażer - przykład

Problem komiwojażera - złożoność

Zbiór permutacji  $n$  miast. (wersja naiwna)

Kto pamięta ile jest możliwych permutacji zbioru  $n$  elementowego?

# Komiwojażer

## Problem komiwojażera - złożoność

Możliwych permutacji  $n$  miast jest  $n!$ .

Na przykład, dla 20 miast jest to 2432902008176640000 możliwych permutacji.

Powiedzmy że sprawdzenie jednej możliwości trwa 50 cykli procesora. Na komputerze z procesorem 3GHz zajęłoby to 1300 lat.

# Komiwojażer

## Problem komiwojażera - złożoność

Problem komiwojażera jest NP-trudny. Złożoność algorytmu naïwnego to  $O(n!)$  natomiast algorytm korzystający z zasady włącz-wyłącz ma złożoność  $O(2^n)$ .

## Wniosek

Jak widać, nie opłaca się tego liczyć dokładnie. Tutaj przychodzą z pomocą rozwiązania przybliżone, a jednym z nich jest program ewolucyjny.

# Komiwojażer

## Dlaczego komiwojażer

Warto mieć chociaż pobiczoną wiedzę na temat sposobów rozwiązywania bardziej zaawansowanych problemów kombinatorycznych.

Na następnych slajdach przedstawię sposoby na reprezentację osobników i metody krzyżowania dla problemu komiwojażera.

# Komiwojażer

Kodowanie dla klasycznego algorytmu genetycznego

Jak to zrobić? (pamiętamy - ciągi bitów)

# Komiwojażer

## Przykład na tablicy

Weźmy problem dla 7 miast.

Wymyślmy 2 osobniki które zostały wybrane do krzyżowania.

Spróbujmy je skrzyżować.

Co nam wychodzi?

# Komiwojażer

## Kodowanie dla klasycznego algorytmu genetycznego

Okazuje się, że kodowanie permutacji jako ciąg bitów jest wyjątkowo nieefektywne.

Problemy z operatorami genetycznymi.

# Komiwojażer

Wg. Whitley, Starkweather, Shaner - cytat

Proste krzyżowanie powoduje podwójne wystąpienie miast, lub ich brak.

Okazuje się, że tradycyjne operatory genetyczne nie nadają się do tego problemu.

# Komiwojażer

## Reprezentacje osobnika w programie ewolucyjnym

- Reprezentacja porządkowa
- Reprezentacja ścieżkowa
- \*Reprezentacja przyległościowa

Omówimy dwie pierwsze z nich.

# Komiwojażer - r. porządkowa

## Reprezentacja porządkowa

Ciąg  $n$  miast zapisany jest tak, że  $i$ -ty element jest liczbą z zakresu 1 do  $n - i + 1$ . Mamy także uporządkowany ciąg miast  $C$  jako punkt odniesienia.

## Przykład (na tablicy)

$$C = (1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9)$$

$$l = (1 \ 1 \ 2 \ 1 \ 4 \ 1 \ 3 \ 1 \ 1)$$

reprezentuje trasę:

$$(1 \ 2 \ 4 \ 3 \ 8 \ 5 \ 9 \ 6 \ 7)$$

# Komiwojażer - r. porządkowa

## Pytanie

Czy widać tu jakąś potencjalną zaletę tej reprezentacji?

Jak zrobić krzyżowanie?

# Komiwojażer - r. porządkowa

## Krzyżowanie

Można zrobić krzyżowanie jednopunktowe i będzie ono działało bez żadnych poprawek.

# Komiwojażer - r. porządkowa

## Krzyżowanie - sprawdźmy (na tablicy)

Przeprowadźmy krzyżowanie jednopunktowe dla:

(1 1 2 1 | 4 1 3 1 1)

(5 1 5 5 | 5 4 4 3 1)

Zobaczmy:

- jakie to są trasy (rodzice)?
- jakie trasy powstaną po krzyżowaniu?
- czy widać tu jakiś problem?

# Komiwojażer - r. porządkowa

## Krzyżowanie - mały problem

Okazuje się, że wyniki eksperymentalne stosowania tego typu reprezentacji i krzyżowania są bardzo słabe.

Jest to związane z tym, że od punktu krzyżowania kolejne miasta są pomieszane w sposób przypadkowy.

# Komiwojażer - r. przyległościowa

## Reprezentacja przyległościowa

Miasto  $j$  znajduje się na pozycji  $i$  wtedy i tylko wtedy, gdy trasa wiedzie z miasta  $i$  do miasta  $j$ .

## Przykład

(2 4 8 3 9 7 1 5 6)

reprezentuje

$1 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 8 \rightarrow 5 \rightarrow 9 \rightarrow 6 \rightarrow 7$

# Komiwojażer - r. ścieżkowa

## Reprezentacja ścieżkowa

Naturalna reprezentacja trasy, czyli chromosom bezpośrednio tłumaczy się na trasę.

## Krzyżowania - przykłady

- PMX - krzyżowanie z częściowym odwzorowaniem
- OX - krzyżowanie z porządkowaniem
- CX - krzyżowanie cykliczne
- ...

# Komiwojażer - r. ścieżkowa

Krzyżowanie PMX (zaproponowane przez Goldberga i Lingle'a)

Wybieramy losową podtrasę jednego z rodziców i pozostawiamy porządek i pozycję tak wielu miast rodzica, jak to możliwe.

# Komiwojażer - r. ścieżkowa

## Krzyżowanie PMX przykład - partially mapped

$$p_1 = (1 \ 2 \ 3 \ | \ 4 \ 5 \ 6 \ 7 \ | \ 8 \ 9)$$

$$p_2 = (4 \ 5 \ 2 \ | \ 1 \ 8 \ 7 \ 6 \ | \ 9 \ 3)$$

Wymieniamy fragmenty między punktami cięcia, przy okazji daje nam to odwzorowania 1 – 4, 8 – 5, 7 – 6, 6 – 7

Wstawiamy pozostałe miasta dla których nie ma konfliktu

$$o_1 = (x \ 2 \ 3 \ | \ 1 \ 8 \ 7 \ 6 \ | \ x \ 9)$$

$$o_2 = (x \ x \ 2 \ | \ 4 \ 5 \ 6 \ 7 \ | \ 9 \ 3)$$

Następnie wykorzystując odwzorowania otrzymujemy

$$o_1 = (4 \ 2 \ 3 \ | \ 1 \ 8 \ 7 \ 6 \ | \ 5 \ 9)$$

$$o_2 = (1 \ 8 \ 2 \ | \ 4 \ 5 \ 6 \ 7 \ | \ 9 \ 3)$$

# Komiwojażer - r. ścieżkowa

Krzyżowanie OX (zaproponowane przez Davisa)

Wybieramy podtrase jednego rodzica, i pozostawiamy wzajemne uporządkowanie miast z drugiego rodzica.

# Komiwojażer - r. ścieżkowa

## Krzyżowanie OX przykład

$$p_1 = (1 \ 2 \ 3 \ | \ 4 \ 5 \ 6 \ 7 \ | \ 8 \ 9)$$

$$p_2 = (4 \ 5 \ 2 \ | \ 1 \ 8 \ 7 \ 6 \ | \ 9 \ 3)$$

Wymieniamy fragmenty między punktami cięcia.

$$o_1 = (x \ x \ x \ | \ 4 \ 5 \ 6 \ 7 \ | \ x \ x)$$

$$o_2 = (x \ x \ x \ | \ 1 \ 8 \ 7 \ 6 \ | \ x \ x)$$

następnie, od drugiego cięcia jednego z rodziców kopujemy miasta pomijając już obecne. Czyli z drugiego rodzica:

9 – 3 – 4 – 5 – 2 – 1 – 8 – 7 – 6 usuwamy 4, 5, 6, 7 i dostajemy

9 – 3 – 2 – 1 – 8. Po krzyżowaniu mamy więc:

$$o_1 = (2 \ 1 \ 8 \ | \ 4 \ 5 \ 6 \ 7 \ | \ 9 \ 3)$$

$$o_2 = (3 \ 4 \ 5 \ | \ 1 \ 8 \ 7 \ 6 \ | \ 9 \ 2)$$

# Komiwojażer - r. ścieżkowa

## Krzyżowanie CX (zaproponowane przez Oliver)

Każde miasto i jego pozycja pochodzi od jednego z rodziców.

Wyjaśnienie na przykładzie.

# Komiwojażer - r. ścieżkowa

Krzyżowanie CX - ciekawa właściwość

Krzyżowanie to zachowuje bezwzględne pozycje elementów rodziców.

# Spis treści

## 1 Problemy Z AG

## 2 Przegląd popularnych operatorów genetycznych

- Metody selekcji
- Metody krzyżowania
- Metody mutacji

## 3 Programy ewolucyjne

- Przykład zastosowania algorytmu ewolucyjnego

## 4 Zakończenie

# Zakończenie

Dziękuję za uwagę.

# Podsumowanie

## Źródła

- Algorytmy genetyczne + struktury danych = programy ewolucyjne, Z. Michalewicz
- Wykłady z Algorytmów Ewolucyjnych, J. Arabas
- docs.opencv.org
- Wikipedia