

## Algorytmy ewolucyjne

### Algorytmy ewolucyjne:

- Inspirowane ewolucją naturalną – zarówno sposób działania i nazewnictwo.
- Początkowo algorytmy z tej rodziny wykorzystywały kodowanie binarne. Algorytmy takie nazywamy **algorytmami genetycznymi**.
- Prosty algorytm genetyczny to już stary pomysł (Holland, 1975 rok; rozwój i zastosowania: De Jong i Goldberg).

### Ewolucyjne nazewnictwo

- Osobnik – reprezentuje punkt w przeszukiwanej przestrzeni ( $\mathbf{x}$ ), może zawierać dodatkowe informacje.
- Populacja – zbiór osobników przetwarzanych w każdej iteracji ( $P$ ).
- Mutacja – losowe zaburzenie zmiennych  $x_i$  osobnika (**genów**),  $i \in 1 \dots n$ , gdzie  $n$  to wymiarowość zadania.
- Krzyżowanie – nowy osobnik powstaje na podstawie genów istniejących osobników (najczęściej dwóch).
- Chromosom – wszystkie geny. Najczęściej osobnik ma jeden chromosom. W niektórych algorytmach osobnik ma dodatkowe, pomocnicze chromosomy.

### Pseudokod algorytmu ewolucyjnego

---

---

```
Data:  $q(x), P_0, \mu, \sigma, p_c, t_{max}$ 
Result:  $\hat{x}^*, \hat{o}^*$ 
1 begin
2    $t \leftarrow 0$ 
3    $o \leftarrow \text{ocena}(q, P_0)$ 
4    $\hat{x}^*, \hat{o}^* \leftarrow \text{znajdź najlepszego}(P_0, o)$ 
5   while nie spełnione kryterium stopu(  $t, t_{max}, P_t, o$  ) do
6      $R \leftarrow \text{reprodukcja}(P_t, o, \mu)$ 
7      $M \leftarrow \text{operacje genetyczne}(R, \sigma, p_c)$ 
8      $o_m \leftarrow \text{ocena}(q, M)$ 
9      $x_t^*, o_t^* \leftarrow \text{znajdź najlepszego}(M, o_m)$ 
10    if  $o_t^* \leq \hat{o}^*$  then
11       $\hat{o}^* \leftarrow o_t^*$ 
12       $\hat{x}^* \leftarrow x_t^*$ 
13    end
14     $P_{t+1}, o \leftarrow \text{sukcesja}(P_t, M, o, o_m)$ 
15     $t \leftarrow t + 1$ 
16  end
17 end
```

---

### Składniki algorytmu ewolucyjnego

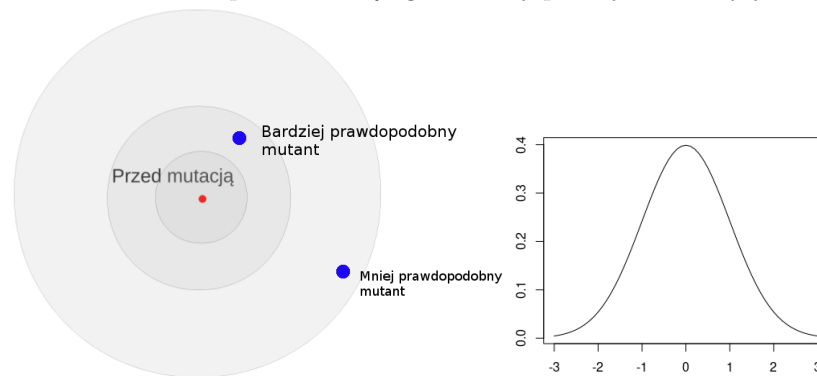
- Reprodukacja – ma za zadanie wybrać lepsze punkty z  $P_t$  z większym prawdopodobieństwem niż gorsze.
- Operatory genetyczne – to typowo krzyżowanie i mutacja, zachowujemy rozmiar populacji (tyle mutantów ile rodziców).
- Krzyżowanie – ma za zadanie wygenerować punkt „pośredni” pomiędzy rodzicami, należy używać z umiarem, tylko jak problem pozwala na sensowne zdefiniowanie tego operatora.
- Mutacja – ma za zadanie wygenerować punkt z otoczenia punktu mutowanego.
- Sukcesja (zastępowanie) – ma zdecydować które osobniki przeżyją do następnej generacji.

## Reprodukcja turniejowa

Reprodukcja turniejowa – w typowym wariancie tego procesu najpierw wybieramy losowo dwa osobniki, następnie najlepszy z nich grupy jest wybierany.

## Idea mutacji

Generacja punktu **z otoczenia** punktu bieżącego, tak aby punkty bliskie były bardziej prawdopodobne



niż te dalekie.

Najczęściej stosuje się mutację gaussowską, czyli do aktualnej wartości  $\mathbf{x}$  dodaje się wektor liczb losowych z rozkładu normalnego, przeskalowany przez  $\sigma$ , nazywane dalej **siłą mutacji**:  $\mathbf{x} = \mathbf{x} + \sigma \cdot \mathcal{N}(0, 1)$ .

## Sukcesja

- generacyjna –  $P_{t+1} = M$ ,
- elitarna –  $P_{t+1} = \{k \text{ najlepszych z } P(t)\} \cup M \setminus \{k \text{ najgorszych z tego połączonego zbioru}\}$ ,

## Proponowana konfiguracja algorytmu na początek przygody

- Selekcja turniejowa
- Sukcesja elitarna z elitą o rozmiarze 1
- Bez krzyżowania, każdy osobnik podlega mutacji
- Mutacja wielowymiarowym rozkładem normalnym:  $\mathbf{x} = \mathbf{x} + \sigma \cdot \mathcal{N}(0, 1)$
- Rozmiar populacji: 20
- Liczba iteracji  $\geq 500$