

Strategie ewolucyjne
(ang. evolution strategies)

Szybki przegląd

- Rozwijane w Niemczech w latach 60-70.
- Wcześni badacze: I. Rechenberg, H.-P. Schwefel (student Rechenberga).
- Typowe zastosowanie: Optymalizacja numeryczna.
- Cechy (tak przynajmniej twierdzą ich zwolennicy).
 - Szybkie.
 - Dobry optymalizator dla funkcji zmiennych rzeczywistych.
 - Dużo teorii :(
- Cecha specjalna: Samo-adaptacja parametrów mutacji jest standardem.

Pierwszy przykład: (1+1)-ES

- Celem jest minimalizacja funkcji $f: \mathbb{R}^L \rightarrow \mathbb{R}$.
- Algorytm używa:
 - wektorów z przestrzeni \mathbb{R}^L jako chromosomów. (fenotyp identyczny z genotypem).
 - populacji o rozmiarze 1.
 - wyłącznie mutacji generującej jednego potomka.
 - zachłannej selekcji (lepszy osobnik z tymczasowej populacji obejmującej 1 rodzica i 1 potomka przeżywa).

(1+1) - ES: pseudokod

```
t=0;
initialize(x0);
while( !termination_condition(xt) ) {
    z=rand_norm();
    yt=xt+z;
    if (f(yt)<f(xt))
        xt+1=yt;
    else
        xt+1=xt;
    t=t+1;
}
```

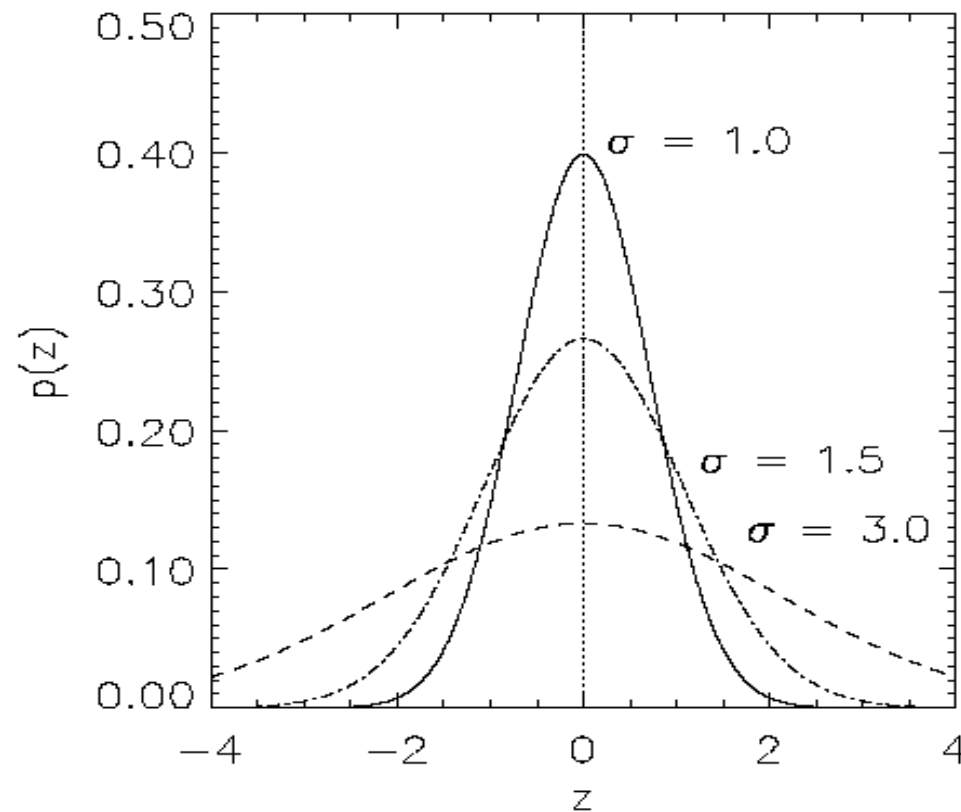
} to jest mutacja

- Funkcja `rand_norm` generuje losowo wektor L – elementowy z , przy czym każdy element z_i ($i=1,2,\dots,L$) tego wektora jest otrzymywany jako:

$$z_i = N_i(0, \sigma)$$

gdzie $N_i(0, \sigma)$ jest liczbą losową z rozkładu normalnego o wartości oczekiwanej 0 i odchyleniu standardowym σ , przy czym losowanie jest wykonywane niezależnie dla każdej wartości i (na co wskazuje indeks przy N).

Rozkład normalny - funkcja gęstości prawdopodobieństwa



- Odchylenie standardowe σ jest nazywane **krokiem mutacji**. Większy krok mutacji \rightarrow większe prawdopodobieństwo dużych odchyień od zera.
- Około 68.3% wylosowanych liczb w przedziale $(-\sigma, \sigma)$. Około 95.5% wylosowanych liczb w przedziale $(-2\sigma, 2\sigma)$. Około 99.7% wylosowanych liczb w przedziale $(-3\sigma, 3\sigma)$ (reguła 3 sigm).
- Jak dobrać wartość σ ?

Reguła 1/5 sukcesu (Rechenberg, 1973)

- Jeżeli przez k kolejnych generacji więcej niż $1/5$ mutacji kończy się sukcesem, to zwiększ krok mutacji, w przeciwnym wypadku zmniejsz ją.
 - Za mutację zakończoną sukcesem uważamy mutację, w wyniku której potomek ma lepsze dopasowanie niż rodzic.
 - Reguła jest stosowana co k pokoleń.
- Niech $\varphi(k)$ jest częstością mutacji zakończonych sukcesem w ostatnich k generacjach algorytmu, Rechenberg zaproponował aby:

$$\sigma^{t+1} = \begin{cases} c_d * \sigma^t & \text{jeżeli } \varphi(k) < 1/5 \\ c_e * \sigma^t & \text{jeżeli } \varphi(k) > 1/5 \\ \sigma^t & \text{jeżeli } \varphi(k) = 1/5 \end{cases}$$

gdzie $c_e > 1$ reguluje szybkość wzrostu, a $c_d < 1$ szybkość malenia kroku mutacji. Schwefel (1981) w swoich eksperymentach użył: $c_d = 0.82$, $c_e = 1.22$.

Algorytm $(\mu+1)$ -ES

- $(1+1)$ -ES to algorytm typu random hill-climbing, wrażliwy na minima lokalne optymalizowanej funkcji.
- Reguła 1/5 sukcesu może prowadzić do przedwczesnej zbieżności.
- Rozszerzenie $(\mu+1)$ – ES
 - Utrzymuj populację μ osobników.
 - Wybierz losowo jednego osobnika i poddaj go mutacji tworząc potomka.
 - Potomek wraz z pozostałymi osobnikami tworzą tymczasową populację o rozmiarze $(\mu+1)$.
 - Z tej populacji μ najlepszych osobników przechodzi do nowej generacji (co jest równoważne odrzuceniu najgorszego).
- Cechy charakterystyczne schematu zarządzania populacją
 - Selekcja rodzica czysto losowa, bez naporu selekcyjnego.
 - Sukcesja deterministyczna z naporem selekcyjnym.
- Powyższe cechy są typowe dla strategii ewolucyjnych.

Algorytm $(\mu+\lambda)$ -ES

- Jest to uogólnienie $(\mu+1)$ – ES
 - Utrzymuj populację μ osobników.
 - Wybierz losowo, ze zwracaniem λ osobników.
 - Poddaj każdego z λ osobników mutacji tworząc λ potomków.
 - λ potomków oraz μ oryginalnych osobników tworzy tymczasową populację o rozmiarze $(\mu+\lambda)$. (stąd nazwa algorytmu)
 - Z tej tymczasowej populacji μ najlepszych osobników przechodzi do nowej generacji (co jest równoważne odrzuceniu λ najgorszych).
 - Czy to jest sukcesja elitarystyczna ?
 - Typowe wartości $\mu=20$, $\lambda=7*\mu$.
- Pozostaje pytanie jak wybierać krok mutacji σ ? W strategiach ewolucyjnych zastosowano rewolucyjny (na owe czasy) pomysł:

Parametr algorytmu (w naszym przykładzie krok mutacji) jest częścią osobnika i podlega ewolucji wraz z nim.

Samoadaptacja kroku mutacji σ (1)

- Duży krok mutacji jest pożyteczny w początkowej fazie algorytmu (lokalizacja obiecujących obszarów przestrzeni rozwiązań).
- Mały krok mutacji pożyteczny w końcowej fazie algorytmu (dokładne dostrajanie lokalne).
- Ale możliwa jest również sytuacja w której algorytm potrzebuje najpierw dużego kroku mutacji, następnie małego następnie dużego ...
- Idea adaptacji kroku mutacji:
 - Pozwól algorytmowi samemu optymalizacji dobierać krok mutacji.
 - Na podstawie obserwacji zachowania algorytmu (reguła 1/5 sukcesów).
 - Poddając parametr algorytmu (w naszym wypadku krok mutacji) ewolucji (selekcja, mutacja, sukcesja) wraz ze zmiennymi x_1, x_2, \dots, x_L . (samoadaptacja).
- A zatem, przy wykorzystaniu samoadaptacji osobnik w populacji ma postać wektora:

$$y = [x_1, x_2, \dots, x_L, \sigma]$$

Samoadaptacja kroku mutacji σ (2)

L pierwszych elementów wektora y reprezentuje wektor x , a dopasowanie:

$$f(y) = f(x) = f([x_1, x_2, \dots, x_L])$$

(a zatem wiemy jak przeprowadzić selekcję i sukcesję).

- Mutacja osobnika $y = [x_1, x_2, \dots, x_L, \sigma]$ daje nowego osobnika $y' = [x'_1, x'_2, \dots, x'_L, \sigma']$ i jest wykonywana w sposób następujący:
- Najpierw obliczany jest nowy krok mutacji σ' .

$$\sigma' = \sigma * \exp(\tau_0 * N(0, 1))$$

gdzie $N(0, 1)$ jest liczbą losową z rozkładu normalnego o wartości oczekiwanej 0 i odchyleniu standardowym 1, τ_0 jest (niestety) parametrem.

- Sprawdzamy, czy krok mutacji nie jest zbyt mały:

Jeżeli $\sigma' < \varepsilon_0$ to $\sigma' = \varepsilon_0$.

- następnie nowo obliczona wartość kroku σ' jest wykorzystywana do mutacji elementów wektora x :

$$x_i' = x_i + N_i(0, \sigma')$$

przy czym liczba losowa z rozkładu normalnego jest losowana niezależnie dla każdego i ($i=1, 2, \dots, L$).

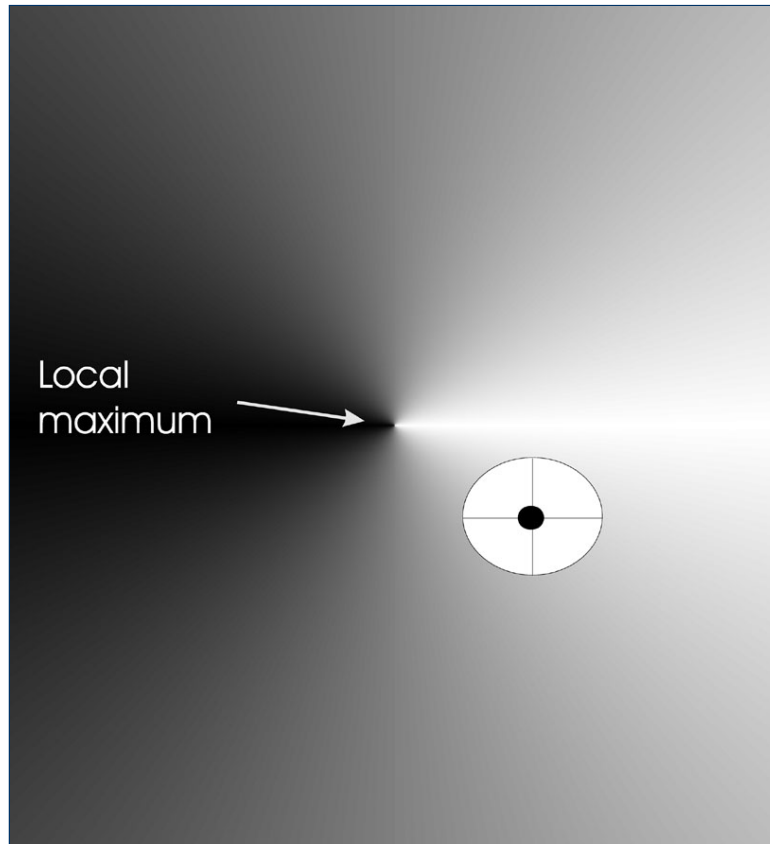
Samoadaptacja kroku mutacji σ (3)

- Parametr τ_0 jest nazywany szybkością uczenia.
 - duże wartości \rightarrow adaptacja kroku σ szybka, ale mało precyzyjna.
 - małe wartości \rightarrow adaptacja wolna, ale bardziej precyzyjna.
- Schwefel (1995) zaproponował, aby:

$$\tau_0 \propto \frac{1}{\sqrt{L}}$$

- Czy jeden krok mutacji jest właściwy dla wszystkich zmiennych x_1, x_2, \dots, x_L ?
 - Co jeżeli $-1000 < x_1 < 1000$, a $0.001 < x_2 < 0.001$?

Wspólny krok mutacji dla wszystkich zmiennych (rys.-Eiben & Smith)



- Funkcja dwóch zmiennych.
- Czarna kropka to osobnik.
- Osobniki na okręgu wokół kropki mają takie same prawdopodobieństwo utworzenia w wyniku mutacji.

- Prawdopodobieństwo kroku wzdłuż osi x (duża zmiana dopasowania) takie same jak wzdłuż osi y (duża zmiana dopasowania).
- Ale przy jednym parametrze mutacji nie ma możliwości sterowania krokiem mutacji dla poszczególnych zmiennych.

L indywidualnych kroków mutacji

- Osobnik y w populacji ma postać: $y = [x_1, x_2, \dots, x_L, \sigma_1, \sigma_2, \dots, \sigma_L]$
- Wynikiem mutacji jest osobnik $y' = [x'_1, x'_2, \dots, x'_L, \sigma'_1, \sigma'_2, \dots, \sigma'_L]$
- Operator mutacji najpierw dokonuje zmian w krokach mutacji:

$$\sigma'_i = \sigma_i * \exp(\tau_0 * N(0,1) + \tau_1 * N_i(0,1))$$

gdzie $N(0,1)$ jest losowane raz dla osobnika y , a $N_i(0,1)$ niezależnie dla każdego

i. Parametry

$$\tau_0 \propto \frac{1}{\sqrt{2L}} \qquad \tau_1 \propto \frac{1}{\sqrt{2}\sqrt{L}}$$

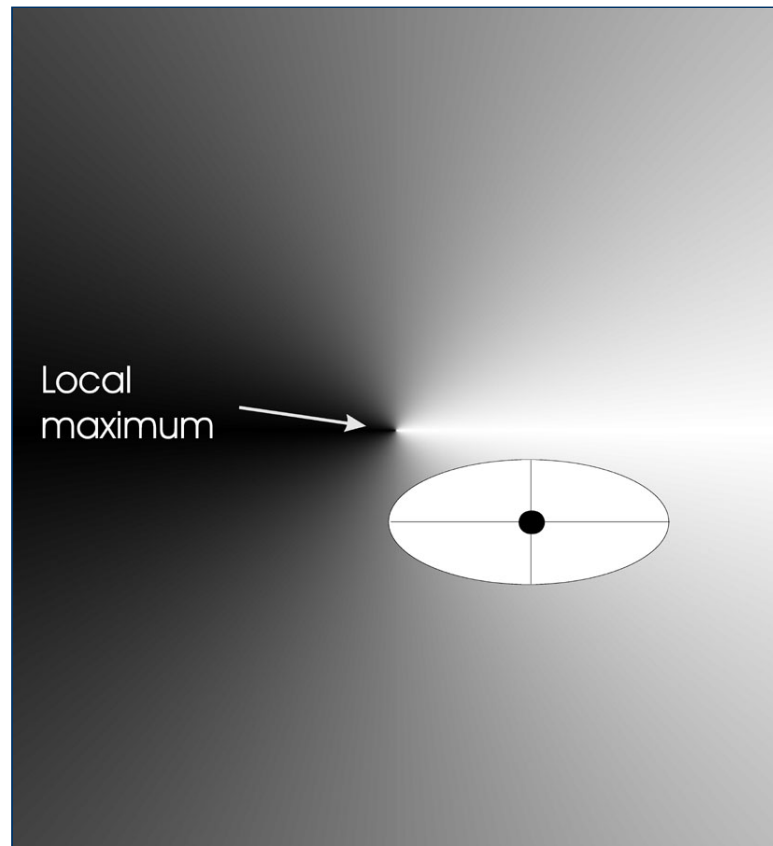
- Następnie sprawdzamy, czy krok mutacji nie jest zbyt mały:

Jeżeli $\sigma' < \varepsilon_0$ to $\sigma' = \varepsilon_0$.

- Po czym stosujemy indywidualny krok mutacji σ'_i do wygenerowania x'_i (dla $i=1,2,\dots,L$):

$$x'_i = x_i + N_i(0, \sigma'_i)$$

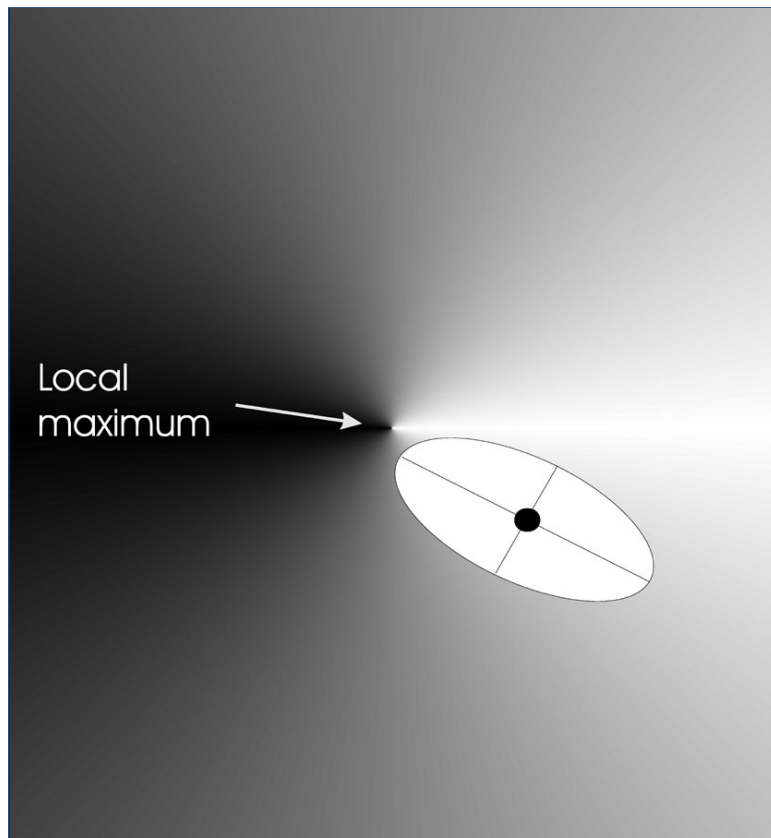
Indywidualne kroki mutacji dla poszczególnych zmiennych (rys.-Eiben & Smith)



- Funkcja dwóch zmiennych.
- Czarna kropka to osobnik.
- Osobniki na elipsie wokół kropki mają takie same prawdopodobieństwo utworzenia w wyniku mutacji.

- Prawdopodobieństwo kroku wzdłuż osi x (duża zmiana dopasowania) większe jak wzdłuż osi y.

Mutacje skorelowane (rys. Eiben & Smith)



- Funkcja dwóch zmiennych.
- Czarna kropka to osobnik.
- Osobniki na elipsie wokół kropki mają takie same prawdopodobieństwo utworzenia w wyniku mutacji.

- Optymalny krok mutacji, po trajektorii największego spadku.
- Wymaga perturbacji wektora x o wektor generowany z rozkładu $N(0, \Sigma)$. Jest to wielowymiarowy rozkład normalny z nediagonalną macierzą kowariancji Σ . Wymaga samoadaptacji współczynników kowariancji.
- Taki algorytm istnieje (CMA-ES) (ang. covariance matrix adaptation - ES)

Problem z metodą sukcesji ($\mu+\lambda$) i algorytm (μ,λ) -ES

- W algorytmie sukcesji ($\mu+\lambda$) osobnik pozostaje w populacji tak długo, jak zostanie zastąpiony przez lepszego potomka.
- Osobnik może pozostawać w populacji dowolnie długo, a jeżeli jego krok(i) mutacji stanie się bardzo mały nie będzie w stanie generować ulepszonych potomków.
- Prowadzi to do przedwczesnej zbieżności do minimum lokalnego.
- Dlatego algorytm sukcesji (μ,λ) jest preferowany
 - W tym algorytmie do nowej generacji przechodzi wyłącznie μ najlepszych spośród λ potomków.
 - A zatem musi być spełniony warunek $\mu < \lambda$. W strategiach ewolucyjnych typowo $\lambda \approx 7 * \mu$. (Np. $\mu=30$, $\lambda=200$).
 - Wszyscy rodzice giną po jednej generacji.
- (μ,λ) nie jest selekcją elitarną (dlaczego?) ale w eksperymentach numerycznych daje na ogół (dla funkcji wielomodalnych) lepsze wyniki niż $(\mu+\lambda)$.

Strategie ewolucyjne z rekombinacją (odpowiednikiem krzyżowania)

- Osobnik który jest poddawany mutacji powstaje w wyniku operatora rekombinacji. Procedura generowania λ potomków wygląda następująco:
 1. Wybierz bez zwracania $\rho > 1$ osobników (z populacji o rozmiarze μ).
 2. Utwórz potomka z ρ osobników stosując operator rekombinacji (o tym na następnym slajdzie)
 3. Poddaj go mutacji.
 4. Dodaj potomka do populacji tymczasowej i zwróć ρ osobników do populacji rodziców.
 5. Powtarzaj krok 1-4 λ razy.
- Liczba rodziców ρ może (ale nie musi) być równa 2 – jak w algorytmach genetycznych.
- Algorytmy z rekombinacją oznaczane są jako $(\mu/\rho+\lambda)$ – ES oraz $(\mu/\rho,\lambda)$ – ES.

Operator rekombinacji (Beyer & Schwefel, 2002)

- Oznaczmy ρ osobników wybranych z populacji o rozmiarze μ jako y^1, y^2, \dots, y^ρ . Niech y^i ($i=1, 2, \dots, \rho$) przyjmuje postać: $y^i = [x_1^i, x_2^i, \dots, x_L^i, \sigma_1^i, \sigma_2^i, \dots, \sigma_L^i]$

- Oznaczmy potomka otrzymanego w wyniku rekombinacji jako y' . Niech

$$y' = [x'_1, x'_2, \dots, x'_L, \sigma'_1, \sigma'_2, \dots, \sigma'_L]$$

- Operator rekombinacji dla zmiennych j -ty ($j=1, 2, \dots, L$, L jest liczbą zmiennych) element potomka losuje spośród j -tych elementów jego rodziców:

$$x'_j = x_j^{r_j}$$

gdzie r_j jest liczbą losową ze zbioru $\{1, 2, \dots, \rho\}$ losowaną niezależnie (od nowa) dla każdego $j=1, 2, \dots, L$.

- Operator rekombinacji dla kroków mutacji oblicza j -ty krok mutacji jako średnią arytmetyczną kroków mutacji wszystkich ρ rodziców.

$$\sigma'_j = \frac{1}{\rho} \sum_{i=1}^{\rho} \sigma_j^i$$

Literatura (oprócz podanej na pierwszym wykładzie)

1. A. E. Eiben, J. E. Smith, *Introduction to Evolutionary Computing*, Springer, 2003.
2. T. Bäck, F. Hoffmeister, H.-P. Schwefel, A survey of evolution strategies, *Proceedings of the Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann, s. 2-9, 1991.
3. T. Bäck, G. Rudolph, H.-P. Schwefel, Evolutionary programming and evolution strategies: similarities and differences, *Proceedings of the Second Annual Conference on Evolutionary Programming*, Evolutionary Programming Society, s. 11-22, 1993.
4. H.-G. Beyer and H.-P. Schwefel, Evolution strategies – A comprehensive introduction, *Natural Computing*, 1(1), s.3-52, 2002.
5. T. Bäck, *Evolutionary Algorithms in Theory and Practice: Evolution strategies, Evolutionary Programming, Genetic Algorithms*, Oxford University Press, 1996

Podsumowanie

- W Państwa projekcie chcę widzieć:
 - Porównanie metod sukcesji $(\mu+\lambda)$ oraz (μ,λ) .
 - Adaptację kroku mutacji indywidualnie dla każdej ze zmiennych.
 - Implementację operatorów rekombinacji i porównanie:
 - wersji działającej bez tych operatorów.
 - wersji z $\rho=2$.
 - wersji z $\rho=\mu$.
- Następny wykład:
 - Algorytmy ewolucyjne dla problemu komiwojażera.