

Algorytmy ewolucyjne

strategie ewolucyjne

Piotr Lipiński

Podstawowe algorytmy ewolucyjne

- Podstawowe algorytmy ewolucyjne
 - algorytmy genetyczne
 - zwykle przestrzeń poszukiwań to $\{0, 1\}^d$
 - niektóre wersje działają na przestrzeni poszukiwań złożonej z permutacji (na przykład SGA z operatorami PMX, CX czy OX)
 - strategie ewolucyjne
 - przestrzeń poszukiwań to R^d
 - występuje mechanizm auto-adaptacji
 - programowanie genetyczne
 - przestrzeń poszukiwań to pewien zbiór drzew
 - programowanie ewolucyjne
 - przestrzeń poszukiwań to zbiór wyrażeń pewnej gramatyki

Strategie ewolucyjne

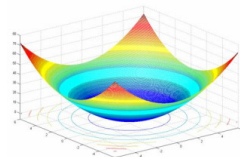
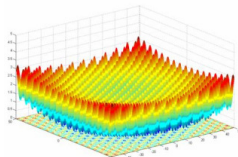
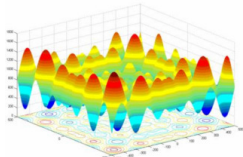
- Strategie ewolucyjne (ang. *Evolution Strategies*, ES) zostały opracowane w Niemczech w latach sześćdziesiątych ubiegłego wieku jako jedna z metod rozwiązywania problemów optymalizacji.
- Badania nad strategiami ewolucyjnymi rozpoczął I. Rechenberg i H.-P. Schwefel na Politechnice Berlińskiej w 1964 r.
- Strategie ewolucyjne:
 - zajmują się optymalizacją funkcji celu $F: \Omega \rightarrow \mathbb{R}$ określonych na przestrzeni poszukiwań $\Omega = \mathbb{R}^d$,
 - kładą duży nacisk na operator mutacji, a mniejszy na operator rekombinacji (w wielu algorytmach operator rekombinacji albo nie występuje albo jest małoistotny),
 - operator mutacji opiera się zazwyczaj na dodawaniu do osobnika zaburzenia losowego generowanego z rozkładem normalnym o zadanych parametrach,
 - używają mechanizmów auto-adaptacji dostosowujących parametry operatorów ewolucyjnych (m.in. zasięg mutacji) w trakcie działania algorytmu do aktualnego stanu poszukiwań.
- Podstawowe strategie ewolucyjne:
 - ES(1+1),
 - ES($\mu + \lambda$) i ES(μ, λ),
 - ES($\mu, \lambda, \rho, \kappa$),
 - CMA-ES($\mu/\mu_w, \lambda$).

Piotr Lipiński, Wykład z algorytmów ewolucyjnych

3

Strategie ewolucyjne

- Popularne testowe funkcje celu to m.in.:
 - Sphere Function
 - Ackley Function
 - Griewank Function
 - Powell Function
 - Rastrigin Function
 - Schwefel Function
- Na konkursach towarzyszących konferencjom *Congress of Evolutionary Computation* (CEC) publikuje się i uaktualnia zbiory testowych problemów optymalizacji służących do rzetelnego testowania efektywności strategii ewolucyjnych.



Źródło: A. Hedar, *Global Optimization Test Problems*, http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO.htm.

Piotr Lipiński, Wykład z algorytmów ewolucyjnych

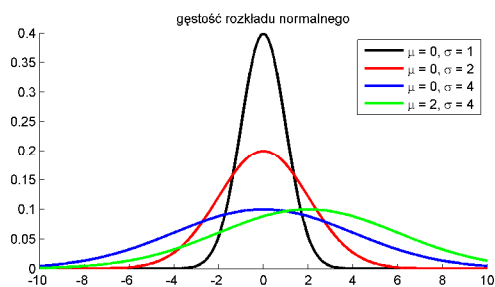
4

Parę słów o rozkładzie normalnym

- gęstość rozkładu normalnego $N(\mu, \sigma^2)$ o wartości oczekiwanej μ i wariancji σ^2 to

$$f(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

- wykresem gęstości jest krzywa Gaussa, wartość oczekiwana wpływa na przesunięcie krzywej, a wariancja na jej kształt



Piotr Lipiński, Wykład z algorytmów ewolucyjnych

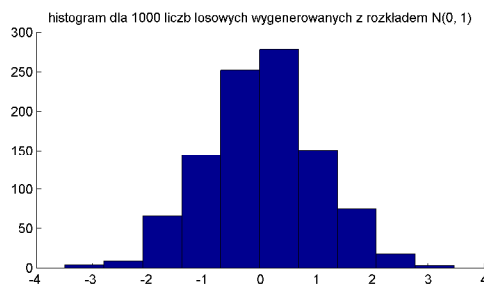
5

Parę słów o rozkładzie normalnym

- generując liczby losowe z rozkładem $N(\mu, \sigma^2)$ powinniśmy dostawać histogramy zgodne z odpowiednią krzywą Gaussa

UWAGA: Jest kilka popularnych algorytmów generowania liczb pseudolosowych z rozkładem normalnym (zazwyczaj są one już zaimplementowane w popularnych narzędziach programistycznych), m.in. algorytm Boxa-Mullera czy algorytm Ziggurata.

- przykład: histogram dla 1000 liczb losowych wygenerowanych z rozkładem $N(0, 1)$



Piotr Lipiński, Wykład z algorytmów ewolucyjnych

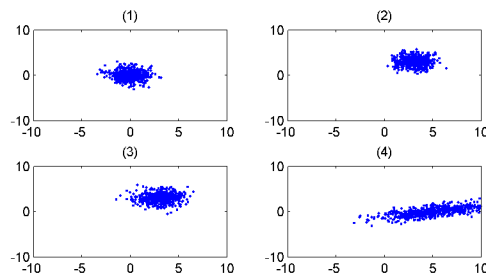
6

Parę słów o wielowymiarowym rozkładzie normalnym

- gęstość d-wymiarowego rozkładu normalnego $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ o wartości oczekiwanej $\boldsymbol{\mu} \in \mathbb{R}^d$ i macierzy kowariancji $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$ to

$$f(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right)$$

- przykład: 500 losowych punktów wygenerowanych z dwuwymiarowym rozkładem normalnym (dla czterech różnych rozkładów)



Piotr Lipiński, Wykład z algorytmów ewolucyjnych

7

Parę słów o wielowymiarowym rozkładzie normalnym

- Podstawowe własności wielowymiarowego rozkładu normalnego:
 - Jeśli $X \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, to $AX + b \sim N(A\boldsymbol{\mu} + b, A\boldsymbol{\Sigma}A^T)$, dla dowolnej macierzy A i dowolnego wektora b (odpowiednich rozmiarów).

- estymator wartości oczekiwanej:

$$\hat{\boldsymbol{\mu}} = \frac{1}{N} \sum_{i=1}^N X_i$$

- estymator kowariancji (o największej wiarygodności):

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{N} \sum_{i=1}^N (X_i - \hat{\boldsymbol{\mu}})(X_i - \hat{\boldsymbol{\mu}})^T$$

- estymator kowariancji (nieobciążony):

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{N-1} \sum_{i=1}^N (X_i - \hat{\boldsymbol{\mu}})(X_i - \hat{\boldsymbol{\mu}})^T$$

- jak generować dane?

- rozkład Choleskiego macierzy kowariancji $\boldsymbol{\Sigma} = AA^T$,
 - $X = Az + \boldsymbol{\mu}$,
- gdzie z to zmienna losowa o standardowym rozkładzie normalnym

Piotr Lipiński, Wykład z algorytmów ewolucyjnych

8

ES(1+1)

- Rozpatrujemy problem minimalizacji funkcji celu $F : \Omega \rightarrow \mathbb{R}$ określonej na przestrzeni poszukiwań $\Omega = \mathbb{R}^d$.
- Algorytm ES(1 + 1) to prosta strategia ewolucyjna (stosowana jedynie do bardzo prostych problemów optymalizacji)
 - populacja złożona z jednego osobnika generuje jednego potomka
 - kolejne (jednoelementowe) populacje są wybierane z sumy mnogościowej (jednoelementowej) populacji rodziców i (jednoelementowej) populacji potomków
 - mechanizm auto-adaptacji realizowany jest przez tzw. *regulę 1/5 sukcesów Rechenberga*

ES(1+1)

- Algorytm

```
σ = σ0;
x = Random-Individual();
Individual-Evaluation(x, F);
while not Termination-Condition()
    y = Mutation(x, σ);
    Individual-Evaluation(y, F);
    if F(y) ≤ F(x) then
        x = y;
    Sigma-Updating(σ, k, θ1, θ2);
return x;
```
- Znaczenie parametrów:
 - $\sigma_0 > 0$ – początkowy zasięg mutacji
 - $k, \theta_1 > 1, \theta_2 < 1$ – parametry auto-adaptacji

ES(1+1)

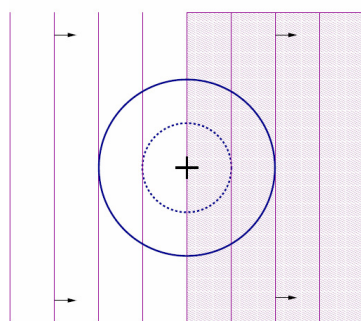
- Operator mutacji przekształca osobnika $\mathbf{x} = (x_1, x_2, \dots, x_d)$ w osobnika $\mathbf{y} = (y_1, y_2, \dots, y_d)$ przez dodanie zaburzenia losowego, niezależnie do każdej współrzędnej $i = 1, 2, \dots, d$, w następujący sposób:

$$y_i = x_i + \varepsilon_i,$$

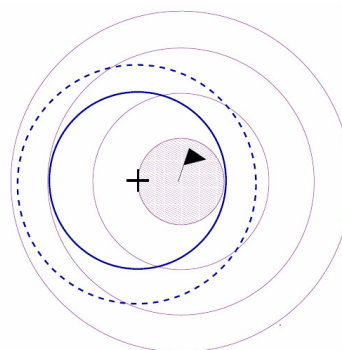
gdzie ε_i jest liczbą rzeczywistą wygenerowaną losowo z rozkładem normalnym $N(0, \sigma^2)$ o średniej 0 i wariancji σ^2 (parametr σ określający zasięg mutacji zmienia się w czasie działania algorytmu).

- Parametr σ aktualizowany jest w każdej iteracji algorytmu przez *regułę 1/5 sukcesów Rechenberga* z parametrami θ_1 i θ_2 .
 - przez udaną mutację rozumiemy mutację, której wynik \mathbf{y} jest lepszy od argumentu \mathbf{x}
 - jeśli przez ostatnie k iteracji liczba udanych mutacji była większa niż 1/5 wszystkich wykonanych mutacji, to zasięg mutacji jest zwiększany, tzn. $\sigma = \theta_1 \sigma$ (dla $\theta_1 > 1$)
 - jeśli przez ostatnie k iteracji liczba udanych mutacji była mniejsza niż 1/5 wszystkich wykonanych mutacji, to zasięg mutacji jest zmniejszany, tzn. $\sigma = \theta_2 \sigma$ (dla $\theta_2 < 1$)
 - (k jest parametrem odpowiadającym za uwzględniany horyzont czasowy)
- Badania pokazały, że dobre wyniki osiąga się dla $\theta_1 = 1/0.82$ i $\theta_2 = 0.82$.

ES(1+1)



increase σ



decrease σ

ES($\mu+\lambda$) i ES(μ, λ)

- Rozpatrujemy problem minimalizacji funkcji celu $F : \Omega \rightarrow \mathbb{R}$ określonej na przestrzeni poszukiwań $\Omega = \mathbb{R}^d$.
- Algorytm ES($\mu+\lambda$) to strategia ewolucyjna, w której
 - populacja złożona z μ osobników generuje λ potomków
 - kolejne populacje są wybierane z sumy mnogościowej populacji rodziców i populacji potomków
 - mechanizm auto-adaptacji opiera się na parametrach zakodowanych w dodatkowych chromosomach osobników
- W algorytmie ES($\mu+\lambda$) osobnik składa się z dwóch chromosomów:
 - tradycyjnego chromosomu $\mathbf{x} \in \mathbb{R}^d$ kodującego element przestrzeni poszukiwań
 - dodatkowego chromosomu $\boldsymbol{\sigma} \in \mathbb{R}^d$ kodującego parametry mutacji
- Algorytm ES(μ, λ) różni się od ES($\mu+\lambda$) jedynie sposobem wyboru kolejnych populacji – są one wybierane wyłącznie z populacji potomków.

ES($\mu+\lambda$)

- Algorytm

```
P = Random-Population( $\mu$ );
Population-Evaluation(P, F);
while not Termination-Condition()
    Pc = Parent-Selection(P,  $\lambda$ );
    Mutation(Pc,  $\tau$ ,  $\tau_0$ );
    Replacement(P, Pc);
    Population-Evaluation(P, F);
return best of P;
```
- Znaczenie parametrów:
 - μ – liczba osobników w populacji głównej **P**
 - λ – liczba osobników w populacji potomnej **P**^c
 - τ, τ_0 – parametry auto-adaptacji

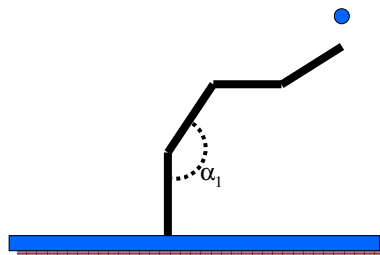
ES($\mu+\lambda$)

- Operator mutacji przetwarza każdego osobnika $(\mathbf{x}^c, \boldsymbol{\sigma}^c) = ((x_1^c, x_2^c, \dots, x_d^c), (\sigma_1^c, \sigma_2^c, \dots, \sigma_d^c))$ z populacji \mathbf{P}^c .
 - Najpierw modyfikowany jest chromosom $\boldsymbol{\sigma}^c$ przez dodanie zaburzenia losowego, niezależnie do każdej współrzędnej $i = 1, 2, \dots, d$, w następujący sposób:
$$\sigma_i^c = \sigma_i^c \exp(\epsilon_i + \epsilon_0),$$
gdzie ϵ_i jest liczbą rzeczywistą wygenerowaną losowo z rozkładem normalnym $N(0, \tau^2)$ o średniej 0 i wariancji τ^2 oraz ϵ_0 jest liczbą rzeczywistą wygenerowaną losowo z rozkładem normalnym $N(0, \tau_0^2)$ o średniej 0 i wariancji τ_0^2 wspólną dla wszystkich współrzędnych.
 - Następnie modyfikowany jest chromosom \mathbf{x}^c przez dodanie zaburzenia losowego, niezależnie do każdej współrzędnej $i = 1, 2, \dots, d$, w następujący sposób:
$$x_i^c = x_i^c + \epsilon_i,$$
gdzie ϵ_i jest liczbą rzeczywistą wygenerowaną losowo z rozkładem normalnym $N(0, \sigma_i^c{}^2)$ o średniej 0 i wariancji $\sigma_i^c{}^2$.
- Badania pokazały, że dobre wyniki osiąga się dla $\tau = K/\sqrt{(2d)}$ i $\tau_0 = K/\sqrt{(2\sqrt{d})}$, gdzie K jest pewnym współczynnikiem uczenia.

Przykład – problem Inverse Kinematics

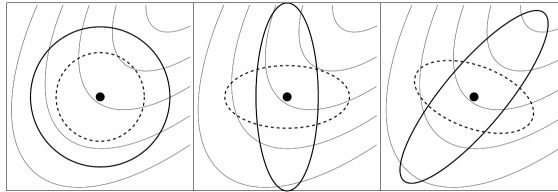
Inverse Kinematics:

- Dane jest ramię robota, przytwierdzone do podłoża, składające się z kilku segmentów o ustalonej długości, które może się wyginać (może zmieniać się kąt między segmentami).
- Celem jest takie ustawienie ramienia robota (określenie wartości kątów między segmentami), żeby koniec ramienia osiągnął podanego punktu docelowego.
- Problem rozpatrujemy w dwóch (łatwiej) lub trzech (trudniej) wymiarach.
- Problem można skomplikować:
 - dodając ograniczenia na zakres wartości każdego kąta (segmenty nie mogą przekreślać się o 360 stopni),
 - wprowadzając przeszkody, które ramię musi ominąć,
 - rozpatrując problem w aspekcie dynamicznym (punkt docelowy lub przeszkody mogą się poruszać).
- Problem taki można sformułować jako problem optymalizacji i rozwiązywać strategiami ewolucyjnymi.



CMA-ES($\mu/\mu_w, \lambda$)

- ES(1+1) to prosta strategia ewolucyjna (stosowana jedynie do bardzo prostych problemów optymalizacji).
- ES($\mu + \lambda$) i ES(μ, λ) są bardziej efektywne, ale:
 - traktują poszczególne zmienne losowe (wymiar przestrzeni poszukiwań) jako niezależne
 - w wielu popularnych benchmarkach zmienne są niezależne, więc algorytm będzie działał skutecznie
 - w wielu problemach praktycznych zmienne są zależne, więc algorytm mógłby wykorzystać to
 - przykład: budowa klasyfikatora zdjęć satelitarnych z określonych reguł klasyfikujących
 - w niektórych przypadkach używanie nieskorelowanego rozkładu normalnego może opóźnić znajdowanie dobrych punktów przestrzeni poszukiwań



Źródło: Nikolaus Hansen, *The CMA Evolution Strategy: A Tutorial*, <https://www.lri.fr/~hansen/cmatutorial.pdf>

- Bardziej zaawansowane strategie ewolucyjne nie zakładają niezależności zmiennych losowych.
 - ES($\mu, \lambda, \rho, \kappa$)
 - CMA-ES($\mu/\mu_w, \lambda$)

CMA-ES($\mu/\mu_w, \lambda$)

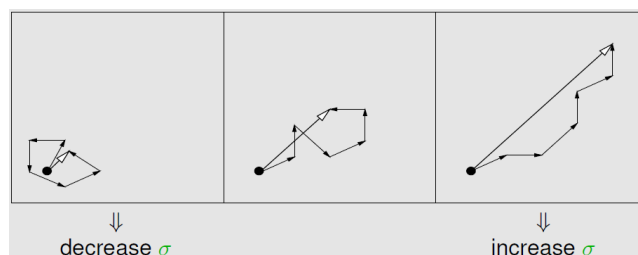
- Algorytm *Covariance Matrix Adaptation Evolution Strategy*, CMA-ES($\mu/\mu_w, \lambda$), to nowoczesna strategia ewolucyjna niezakładająca niezależności zmiennych losowych, ale używająca macierzy kowariancji do określenia tych zależności.
- Podstawowe cechy algorytmu CMA-ES($\mu/\mu_w, \lambda$)
 - maksymalizacja funkcji wiarygodności
 - celem jest zwiększenie prawdopodobieństwa generowania dobrych rozwiązań
 - wartość oczekiwana rozkładu prawdopodobieństwa jest uaktualniana, tak aby funkcja wiarygodności dla poprzednio otrzymanych dobrych rozwiązań (przewyższających dotychczas najlepszych) była maksymalna
 - macierz kowariancji jest uaktualniana, tak aby wiarygodność poprzednich kroków rosła
 - obie modyfikacje są oparte na metodzie *natural gradient descent*

CMA-ES($\mu/\mu_w, \lambda$)

- Rozpatrujemy problem minimalizacji funkcji celu $F : \Omega \rightarrow \mathbb{R}$ określonej na przestrzeni poszukiwań $\Omega = \mathbb{R}^d$.
- Algorytm CMA-ES($\mu/\mu_w, \lambda$) to strategia ewolucyjna, w której
 - populacja złożona z μ osobnika generuje λ potomków
 - kolejne populacje są wybierane z sumy mnogościowej populacji rodziców i populacji potomków
 - mechanizm auto-adaptacji opiera się na parametrach zakodowanych w dodatkowych chromosomach osobników
- Algorytm CMA-ES($\mu/\mu_w, \lambda$) używa trzech zmiennych modelujących rozkład prawdopodobieństwa opisujący rozwiązanie problemu optymalizacji
 - $\mathbf{m} \in \mathbb{R}^d$
 - $\sigma \in \mathbb{R}$
 - $\mathbf{C} \in \mathbb{R}^{d \times d}$
 - osobniki są losowane z użyciem wielowymiarowego rozkładu normalnego $N(\mathbf{m}, \sigma^2 \mathbf{C})$ opisywanego przez powyższe parametry
 - początkowe ustawienie parametrów to $\mathbf{m} = \mathbf{0}$, $\sigma = 1$, $\mathbf{C} = \mathbf{I}$.
- Algorytm CMA-ES($\mu/\mu_w, \lambda$) używa też dwóch zmiennych pomocniczych
 - p_e (isotropic evolution path)
 - p_c (anisotropic evolution path)

CMA-ES($\mu/\mu_w, \lambda$)

- W algorytmie CMA-ES($\mu/\mu_w, \lambda$) auto-adaptacja realizowana jest przez kontrolę długości ścieżki ewolucyjnej (ang. *Path Length Control*) przy użyciu mechanizmu *Cumulative Step-Size Adaptation* (CSA).



Źródło: A. Auger, N. Hansen, Tutorial: CMA-ES — Evolution Strategies and Covariance Matrix Adaptation, <https://www.lri.fr/~hansen/gecco2011-CMA-ES-tutorial.pdf>

CMA-ES($\mu/\mu_w, \lambda$)

- *Cumulative Step-Size Adaptation (CSA)*
 - $\mathbf{m} \in \mathbb{R}^d, \sigma \in \mathbb{R}, \sigma > 0, p_\sigma = 0$
 - $c_\sigma \approx 4/n, d_\sigma \approx 1$ (współczynniki uczenia),
 - \mathbf{y} := średnia wybranych osobników z populacji
 - aktualizacja wektora średniej \mathbf{m} :
 $\mathbf{m} := \mathbf{m} + \sigma \mathbf{y}$,
 - aktualizacja rozmiaru kroku σ :
 $p_\sigma := (1 - c_\sigma) p_\sigma + \text{sqrt}(1 - (1 - c_\sigma)^2) \text{sqrt}(\boldsymbol{\mu}) \mathbf{y}$
 $\sigma = \sigma \exp(c_\sigma / d_\sigma (\|\mathbf{p}_\sigma\| / \mathbb{E}\|\mathbf{N}(\mathbf{0}, \mathbf{I})\|))$
gdzie $\boldsymbol{\mu}$ to czynnik normalizujący (równy sumie odwrotności kwadratów wag użytych przy obliczaniu średniej ważonej \mathbf{y})

CMA-ES($\mu/\mu_w, \lambda$)

- *Covariance Matrix Adaptation (CMA) – Rank-One Update*
 - $\mathbf{m} \in \mathbb{R}^d, \mathbf{C} = \mathbf{I} \in \mathbb{R}^{d \times d}, \sigma = 1 \in \mathbb{R}$
 - $c_{\text{cov}} \approx 2/n^2$ (współczynnik uczenia),
 - generowanie osobników nowej populacji:
 $\mathbf{y}_i := \mathbf{N}(\mathbf{0}, \mathbf{C})$
 $\mathbf{x}_i := \mathbf{m} + \sigma \mathbf{y}_i$
 - \mathbf{y} := średnia wybranych osobników z populacji
 - aktualizacja wektora średniej \mathbf{m} :
 $\mathbf{m} := \mathbf{m} + \sigma \mathbf{y}$,
 - aktualizacja macierzy kowariancji \mathbf{C} :
 $\mathbf{C} := (1 - c_{\text{cov}}) \mathbf{C} + c_{\text{cov}} \boldsymbol{\mu} \mathbf{y}^T$
gdzie $\boldsymbol{\mu}$ to czynnik normalizujący (równy sumie odwrotności kwadratów wag użytych przy obliczaniu średniej ważonej \mathbf{y})

CMA-ES($\mu/\mu_w, \lambda$)

□ Algorytm

```
(m,  $\sigma$ , C) = Model-Initialization();  
P = Random-Population(m, C);  
Population-Evaluation(P, F);  
while not Termination-Condition()  
    Cumulative-Step-size-Adaptation();  
    Covariance-Matrix-Adaptation();  
    P = Random-Population(m, C);  
    Population-Evaluation(P, F);  
return best of P;
```

□ Znaczenie parametrów:

- $\mathbf{m} \in \mathbf{R}^d$
- $\sigma \in \mathbf{R}$
- $\mathbf{C} \in \mathbf{R}^{d \times d}$
- osobniki są losowane z użyciem wielowymiarowego rozkładu normalnego $N(\mathbf{m}, \sigma^2 \mathbf{C})$ opisywanego przez powyższe parametry
- początkowe ustawienie parametrów to $\mathbf{m} = \mathbf{0}$, $\sigma = 1$, $\mathbf{C} = \mathbf{I}$.