Algorytmy ewolucyjne

ES(μ,λ,ρ,κ)

Piotr Lipiński

lipinski@ii.uni.wroc.pl

Algorytm ES$(\mu, \lambda, \kappa, \rho)$

W $ES(\mu, \lambda, \rho, \kappa)$ populacja złożona z μ osobników generuje λ potomków. Każdy potomek jest generowany przez ρ rodziców. Kolejne populacje są wybierane z sumy mnogościowej populacji rodziców i populacji potomków, jednak z wyłączeniem osobników, które osiągnęły wiek κ, czyli przetrwały κ iteracji algorytmu.

W $ES(\mu, \lambda, \rho, \kappa)$, podobnie jak w dwóch poprzednich algorytmach, mechanizm autoadaptacji oparty jest na kodowaniu parametrów operatorów ewolucyjnych w osobniku.

Algorytm ES$(\mu, \lambda, \kappa, \rho)$

Każdy osobnik $\hat{\mathbf{x}}$ składa się z chromosomu $\mathbf{x} \in \mathbb{R}^d$, wieku $\kappa \in \mathbb{R}$ oraz dodatkowych parametrów $\alpha \in \mathbb{R}^{\frac{d(d+1)}{2}}$ i $\sigma \in \mathbb{R}^d$ wykorzystywanych podczas mutacji

$$\hat{\mathbf{x}} = (\mathbf{x}, \kappa, \alpha, \sigma),$$

gdzie

$$\mathbf{x} = (x_1, x_2, \ldots, x_d),$$

$$\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_{\frac{d(d+1)}{2}}),$$

$$\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_d).$$

Algorytm ES$(\mu, \lambda, \kappa, \rho)$

Evolution-Strategy$(F, N, M, n, \beta, \tau, \tau_0)$

```
P ← Random-Population(N);
Population-Evaluation(P, F);
while not Termination-Condition(P)
do
    P^(C) = Reproduction(P, M, n, β, τ, τ_0);
    Replacement(P, P^(C));
    Population-Evaluation(P, F);
```

$\mathcal{P} \leftarrow$ Random-Population(N);
Population-Evaluation$(\mathcal{P}, F)$;
while not Termination-Condition$(\mathcal{P})$
do
    $\mathcal{P}^{(C)} =$ Reproduction$(\mathcal{P}, M, n, \beta, \tau, \tau_0)$;
    Replacement$(\mathcal{P}, \mathcal{P}^{(C)})$;
    Population-Evaluation$(\mathcal{P}, F)$;

ES$(\mu, \lambda, \kappa, \rho)$: Reproduction

Reproduction$(\mathcal{P}, M, n, \beta, \tau, \tau_0)$

$\mathcal{P}^{(C)} \leftarrow \emptyset$;
for $k = 1$ to M
do
    $\{\hat{\mathbf{x}}_1^{(P)}, \hat{\mathbf{x}}_2^{(P)}, \ldots, \hat{\mathbf{x}}_n^{(P)}\} \leftarrow$ Parent-Selection$(\mathcal{P}, n)$;
    $\hat{\mathbf{x}}^{(C)} \leftarrow$ Crossover$(\hat{\mathbf{x}}_1^{(P)}, \hat{\mathbf{x}}_2^{(P)}, \ldots, \hat{\mathbf{x}}_n^{(P)})$;
    Mutation$(\hat{\mathbf{x}}^{(C)}, \beta, \tau, \tau_0)$;
    $\mathcal{P}^{(C)} \leftarrow \mathcal{P}^{(C)} \cup \{\hat{\mathbf{x}}^{(C)}\}$;
return $\mathcal{P}^{(C)}$

ES$(\mu, \lambda, \kappa, \rho)$: Reproduction

Z bieżącej populacji $\mathcal{P}$ algorytm tworzy nową populacje $\mathcal{P}^{(C)} = \{\breve{\mathbf{x}}_1^{(C)}, \breve{\mathbf{x}}_2^{(C)}, \ldots, \breve{\mathbf{x}}_M^{(C)}\}$, zwaną *populacją potomków* lub *populacją dzieci*.

Reprodukcja składa się z trzech faz:

- *parent selection*,

- *crossover*,

- *mutation*,

powtarzanych M razy. W każdej iteracji tworzony jest jeden potomek.

Parent Selection

Z bieżącej populacji $\mathcal{P}$ wybieranych jest n osobników $\breve{\mathbf{x}}_1^{(P)}, \breve{\mathbf{x}}_2^{(P)}, \ldots, \breve{\mathbf{x}}_n^{(P)}$.

Osobniki są wybierane losowo w taki sposób; że prawdopodobieństwo wyboru osobnika $\breve{\mathbf{x}}_i$ jest równe wartości jego przystosowania $f(\breve{\mathbf{x}}_i)$ (metoda ruletki).

Krzyżowanie

n osobników $\breve{\mathbf{x}}_1^{(P)}, \breve{\mathbf{x}}_2^{(P)}, \ldots, \breve{\mathbf{x}}_n^{(P)}$ tworzy jednego potomka $\breve{\mathbf{x}}^{(C)}$. Każda składowa potomka, tzn. $\mathbf{x}^{(C)}$, $\alpha^{(C)}$ i $\sigma^{(C)}$, jest tworzona niezależnie przez zastosowanie jednego z operatorów rekombinacji. Naturalnie, do tworzenia różnych składowych mogą być użyte różne operatory.

Zazwyzaj używanych jest kilka operatorów rekombinacji: *random selection*, *global intermediary recombination*, *local intermediary recombination* oraz *uniform crossover*.

Po utowrzeniu, wiek potomka $\kappa^{(C)}$ jest ustawiany na zero.

Krzyżowanie

random selection: algorytm wybiera losowo liczbę całkowitą k taką że $1 \leqslant k \leqslant n$; potomek dziedziczy cały chromosom od rodzica $\breve{\mathbf{x}}_k^{(P)}$

$$x_j^{(C)} = x_{kj}^{(P)}, \quad \text{dla } j = 1, 2, \ldots, d.$$

global intermediary recombination: chromosom potomka jest tworzony jako średnia arytmetyczna chromosomów jego wszystkich rodziców

$$x_j^{(C)} = \frac{1}{n} \sum_{k=1}^{n} x_{kj}^{(P)}, \quad \text{dla } j = 1, 2, \ldots, d.$$

Krzyżowanie

local intermediary recombination: algorytm wybiera losowo liczbę rzeczywistą $u \in (0,1)$ oraz dwie liczby całkowite k_1 i k_2 takie że $1 \leqslant k_1, k_2 \leqslant n$; chromosom potomka jest tworzony jako kombinacja chromosomów dwóch jego rodziców $\breve{\mathbf{x}}_{k_1}^{(P)}$ i $\breve{\mathbf{x}}_{k_2}^{(P)}$

$$x_j^{(C)} = u \cdot x_{k_1 j}^{(P)} + (1-u) \cdot x_{k_2 j}^{(P)}, \quad \text{dla } j = 1, 2, \ldots, d.$$

uniform crossover: podobnie jak w SSGA, potomek dziedziczt każdy gen od losowo wybranego rodzica.

Mutacja

Mutacja jest najważniejszym operatorem ewolucyjnym ES (inaczej niż w algorytmach genetycznych).

Potomek $\breve{\mathrm{x}}^{(C)}$ jest modyfikowany przez dodanie pewnego losowego zaburzenia. Każdy potomek jest modyfikowany oddzielnie przy użyciu zakodowanych w nim parametrów mutacji α i σ.

Początkowo algorytm modyfikuje zakodowany w osobniku parametr $\alpha^{(C)}$

$$\alpha_j = \alpha_j + \epsilon_j, \quad \text{dla } j = 1, 2, \ldots, d,$$

gdzie ϵ_j jest liczbą rzeczywistą wygenerowaną losowo z rozkładem normalnym o średniej 0 i wariancji β^2, zaś β jest parametrem algorytmu.

Mutacja

Następnie, modyfikowany jest parametr $\sigma^{(C)}$

$$\sigma_j = \sigma_j \cdot \mathrm{e}^{\epsilon_j + \epsilon_0}, \quad \text{dla } j = 1, 2, \ldots, d,$$

gdzie ϵ_j jest liczbą rzeczywistą wygenerowaną losowo z rozkładem normalnym o średniej 0 i wariancji τ^2, gdzie τ jest parametrem algorytmu, zaś ϵ_0 jest liczbą rzeczywistą wygenerowaną losowo z rozkładem normalnym o średniej 0 i wariancji τ_0^2, gdzie τ_0 jest parametrem algorytmu.

Ostatecznie, algorithm modyfikuje chromosom $\mathbf{x}^{(C)}$ używając dwóch parametrów mutacji $\alpha^{(C)}$ i $\sigma^{(C)}$ zawartych w potomku i zmodyfikowanych wcześniej.

Mutacja

Początkowo, generowany jest losowo wektor liczb rzeczywistych $\mathbf{z} = (z_1, z_2, \ldots, z_d)^T$ w taki sposób, że każda współrzędna z_i jest generowana losowo z rozkładem normalnym o średniej 0 i wariancji $(\sigma_i^{(C)})^2$.

Następnie, tworzona jest macierz $\mathbf{T}$

$$\mathbf{T} = \prod_{p=1}^{d} \prod_{q=p+1}^{d} \mathbf{T}_{pq}(\alpha_j^{(C)}),$$

gdzie $j = \frac{1}{2}(2N - p)(p+1) - 2N + q$ oraz

Mutacja

$$\mathbf{T}_{pq}(\alpha) = \begin{pmatrix} 1 & 0 & & & & & \cdots & & & & & 0 \\ 0 & 1 & & & & & & & & & & 0 \\ & & \ddots & & & & & & & & & \\ & & & 1 & & & & & & & & \\ & & & & \cos\alpha & & & -\sin\alpha & & & & \\ & & & & & 1 & & & & & & \\ \vdots & & & & & & \ddots & & & & & \vdots \\ & & & & & & & 1 & & & & \\ & & & & \sin\alpha & & & \cos\alpha & & & & \\ & & & & & & & & 1 & & & \\ & & & & & & & & & \ddots & & \\ & & & & & & & & & & 1 & \\ 0 & 0 & & & & & \cdots & & & & & 1 \end{pmatrix}$$

Mutacja

tzn. $\mathbf{T}_{pq}(\alpha) = \{t_{ij}\}$, $i, j = 1, 2, \ldots, d$, gdzie $t_{ij} = \cos\alpha$ dla $i = p$ i $j = p$, $t_{ij} = -\sin\alpha$ dla $i = p$ i $j = q$, $t_{ij} = \sin\alpha$ dla $i = q$ i $j = p$, $t_{ij} = \cos\alpha$ dla $i = q$ i $j = q$, w pozostałych przypadkach $t_{ij} = 1$ dla $i = j$ i $t_{ij} = 0$ dla $i \neq j$.

Następnie, modyfikowany jest chromosom $\mathbf{x}^{(C)}$

$$\mathbf{x}^{(C)} = \mathbf{x}^{(C)} + \epsilon^T,$$

gdzie

$$\epsilon = \mathbf{T}\mathbf{z}.$$

Replacement

Osobniki z populacji potomków $\mathcal{P}^{(C)}$ zastępują osobników w bieżącej populacji $\mathcal{P}$ w taki sposób, że rozmiar populacji nie zmienia się.

Zazwyczaj używa się kilku metod:
- (μ, λ),
- $(\mu + \lambda)$,
- selekcja deterministyczna,
- selekcja turniejowa.