An Improved CMA-ES for Solving Large Scale Optimization Problem

Jin Jin(✉), Chuan Yang, and Yi Zhang

Chengdu Neusoft University, Chengdu 610844, China
jinjin@nsu.edu.cn

Abstract. In solving large scale optimization problems, CMA-ES has the disadvantages of high complexity and premature stagnation. To solve this problem, this paper proposes an improved CMA-ES, called GI-ES, for large-scale optimization problems. GI-ES uses all the historical information of the previous generation of individuals to evaluate the parameters of the distribution of the next generation. These estimates can be considered as approximate gradient information, which complete covariance information is not required. Thus GI-ES is friendly to large scale optimization problems. Comparative experiments have been done on state-of-the-art algorithms. The results proved the effectiveness and efficiency of GI-ES for large scale optimization problems.

Keywords: CMA-ES · Approximate gradients · Information utilization · Large scale optimization

1 Introduction

In the era of big data, large scale optimization has been applied more and more widely in many engineering and research fields. These optimization problems are difficult to obtain mathematical optimization models, such as simulation software, can be regarded as black box optimization problems. However, the stochastic global optimization method does not have high requirements on the characteristics of the optimization problem itself and does not depend on the specific problem. Therefore, it becomes a common method to solve optimization problems.

Among the stochastic global optimization methods, the most representative algorithm is evolutionary computation, which is a modern optimization algorithm [2]. Its main characteristics are parallelism and self-adaptability, self-learning habits and self-organization. Since 1960s when evolutionary computing was designed, it developed rapidly and formed many branches. These branches include genetic algorithm (GA)[8], evolutionary strategy (ES)[13], particle swarm optimization (PSO)[20], differential evolution algorithm (DE) [16], Covariance matrix adaptation evolutionary computation (CMA-ES) [6] and so on. These algorithms do not need the domain knowledge of the problem, only need to be able to calculate the fitness of the optimization target to be applied.

Y. Tan et al. (Eds.): ICSI 2020, LNCS 12145, pp. 386–396, 2020.
https://doi.org/10.1007/978-3-030-53956-6_34

But as the size of the problem increases, traditional evolutionary computation algorithms take hours or even days to find the optimal solution.

Covariance adaptive optimization algorithm is an optimization algorithm proposed in recent years [5]. The adaptive evolutionary strategy of covariance matrix can automatically adjust the standard deviation according to the distribution of the population. In addition, because CMA-ES can use the information of the optimal solution to adjust its parameters at the same time. CMA-ES as one of the most popular gradient-free optimization algorithms, has become the choice of many researchers and practitioners.

Although it has many advantages, CMA-ES has high space and time complexity when dealing with large scale optimization problems. Another obvious disadvantage is CMA-ES assessed some of the best individuals. Although it can speed up convergence to some extent, this strategy discards most of the information. We all know that great people in life have certain qualities that we can learn from, but some people who fail also keep a record of "not doing" something. It is important for better calculation and evaluation of the next generation. These two limitations may prevent large scale optimization using CMA-ES.

This paper proposes an evolutionary strategy based on gradient information utilization (GI-ES), which extends the application of CMA-ES in the field of large scale optimization.

To summarise, this work make the following contributions.

- The calculation of covariance matrix is replaced by the expected fitting degree scoring strategy.
- The gradient information is simulated through all individual information, and the approximate gradient information is used to guide the search direction.
- The extensive experiments are conducted on basic test problems. Experiments results prove the effectiveness and efficiency of the proposed algorithm.

Organization. Following the introduction section. We first discuss the related work of the utilization of the information of evolutionary computation in Sect. 2. Section 3 describes the detailed implementations of GI-ES. Thereafter, the simulation results on the benchmark test suites are conducted to evaluate the effectiveness of the proposed approach in Sect. 4. Finally, Sect. 5 summarizes this paper.

2 Related Work

Almost all evolutionary strategy (ES) frameworks are similar, with the main step being 1) to generate a number of candidates as needed, which can be either fixed or dynamic; 2) update the candidate scheme according to certain rules, and use the history information of the objective function in the update process. In the first step, information can be obtained according to the candidate scheme. In the second step, according to the information obtained, the algorithm can discard and retain the candidate schemes.

Many algorithms [4,12] that use objective function guidance information show that the use of this information plays an important role in the improvement of the algorithm. [11] proposed the information utilization ratio (IUR) to evaluate the performance of the heuristic algorithm. The IUR can be used as a metric to reflect how finely and advanced an algorithm is designed.

The research on gradient information originates from the Policy gradient proposed by Williams [18], which uses the reinforcement learning as a means of ES. Literature [15] adopts Policy Exploring Policy Gradients extends the using of gradient information. More extensions of ES that modify the search distribution use natural gradient or non-Gaussian (such as longtail distribution) search distributions [17]. [10] uses gradients information of a network with respect to the weights to increases the ability of traditional ES. Guided evolutionary strategies [12] uses the surrogate gradient information which combine the first-order methods and the random search.

In this study, information about the approximate gradient is used to optimize original CMA-ES. Our algorithm is different from these because our method estimates the gradient information but does not use first-order information. Because it is relatively difficult to solve the first order information of large scale optimization problems. The algorithm does not require absolute accuracy of the gradient, which is friendly for large scale problems.

3 Proposed Approach

The proposed algorithm GI-ES adopts the basic framework of CMA-ES, but makes some improvements to CMA-ES. In the proposed scheme, keep all the information about each scheme in each generation, good or bad. In this way, with these gradient signal assessments, we can move the whole scheme in a better direction for the next generation. Since we need to evaluate the gradient, we can use the standard stochastic gradient descent algorithm (SGD) applied to deep learning [3].

3.1 GI-ES

The fitness score was optimized for each sampling scheme in GI-ES. If the expected results are good enough, the best-performing scheme in the sampling generation may perform better. Maximization of the expected fitness score of a sampling scheme is actually equivalent to maximization of the whole fitness score.

3.2 Search Gradient Adaptation

GI-ES adopts approximate gradient as the direction of search, so that the algorithm can adapt to the fitness terrain dependent on variables. This process generates a fitting degree evaluation by the expected fitting degree score and obtains

the gradient signal by the maximum likelihood estimation. It differs from traditional evolutionary computation in that it represents this "population" as a parameterized distribution, when it actually updates the parameters of this distribution using a search gradient, which is calculated using fitness values.

In the above derivation, the distribution is expressed as $\pi(z, \theta)$. In the actual scheme, the most typical distribution has multivariate normal distribution, but the algorithm can still extend to other distributions.

Multinormal Distribution. Multivariate normal distribution is the most widely used distribution in evolutionary computation. The fisher information matrix of multiple normal distribution can be easily solved. For normal distribution, the parameter θ is (μ, Σ), where $\boldsymbol{\mu} \in \mathbb{R}^d$ is the center of the alternative solution, and $\Sigma \in \mathbb{R}^{d\times d}$ is the covariance matrix. To sample more efficiently, you need a matrix $A \in \mathbb{R}^{d\times d}$, meet $\mathbf{A}^\top\mathbf{A} = \boldsymbol{\Sigma}$, then $\mathbf{z} = \boldsymbol{\mu}+\mathbf{A}^\top$ can transfer the standard normal distribution $\mathbf{s} \sim \mathcal{N}(0, \mathbb{I})$ to $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. $\mathbb{I} = \mathrm{diag}(1, \ldots, 1) \in \mathbb{R}^{d\times d}$ denotes the identity matrix. $\pi(\mathbf{z}|\theta)$ represents the probability density function of the multinormal distribution.

$$\begin{aligned}\pi(\mathbf{z}|\theta) &= \frac{1}{(\sqrt{2\pi})^d|\det(\mathbf{A})|} \cdot \exp\left(-\frac{1}{2}\left\|\mathbf{A}^{-1}\cdot(\mathbf{z}-\boldsymbol{\mu})\right\|^2\right) \\ &= \frac{1}{\sqrt{(2\pi)^d\det(\boldsymbol{\Sigma})}} \cdot \exp\left(-\frac{1}{2}(\mathbf{z}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{z}-\boldsymbol{\mu})\right)\end{aligned} \tag{1}$$

In order to calculate the gradient information of the multivariate gaussian variable, the logarithm of the probability density is obtained, so that the gradient can be estimated by summation:

$$\log\pi(\mathbf{z}|\theta) = -\frac{d}{2}\log(2\pi) - \frac{1}{2}\log\det\boldsymbol{\Sigma} - \frac{1}{2}(\mathbf{z}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{z}-\boldsymbol{\mu}) \tag{2}$$

So $\nabla_\mu \log\pi(\mathbf{z}|\theta)$ and $\nabla_\Sigma \log\pi(\mathbf{z}|\theta)$ can be obtained. Then update the parameters with the calculated gradient information.

$$\theta \leftarrow \theta + \eta\nabla_\theta J \tag{3}$$

The algorithm can dynamically change the shape as needed to continue exploring or adjusting the solution space.

The Technique of GI-ES. Further, A can be decomposed into a scale parameter σ, and a normalized covariance factor B satisfying $det(B) = 1$. This decoupling form of two orthogonal components can be independently learned.

The advantage of overall information utilization is to prevent information loss, but outliers still need to be considered. In this method, according to the fitness value, the population individuals are ranked according to the fitness from

small to large. Calculate the utility value according to the fitness value $u_1 \geq \cdots \geq u_{\lambda-2}$. Here, to reduce the impact of outliers on the performance of the algorithm. Get rid of the best and the worst.

$$u_i = \frac{u_i}{u_1 - u_{\lambda-2}} \tag{4}$$

The complexity of each covariance matrix update is $O(d^3)$. The complexity can be reduced to $O(d^2)$ by calculating the update of local non-exponential coordinates. In this case, the update of gradient information can be decomposed into the following components,

$$\nabla_{\mathbf{M}} J \leftarrow \sum_{k=1}^{\lambda-2} u_k \cdot \left(\mathbf{s}_k \mathbf{s}_k^{\top} - \mathbb{I}\right) \tag{5}$$

$$\nabla_{\sigma} J \leftarrow \operatorname{tr}\left(\nabla_{\mathbf{M}} J\right)/d \tag{6}$$

$$\nabla_{\mathbf{B}} J \leftarrow \nabla_{\mathbf{M}} J - \nabla_{\sigma} J \cdot \mathbb{I} \tag{7}$$

The Implementation of GI-ES. In this section, we summarize the pseudo-code of the proposed algorithm in Algorithm 1.

Algorithm 1: The pseudo-code of GI-ES

Require: $f(x)$: objective funtion; μ_{init}: initial μ; $\boldsymbol{\Sigma}_{init} = \mathbf{A}^{\top}\mathbf{A}$;
Ensure: optimal x^*
Initial $\sigma \leftarrow \sqrt[d]{|\det(\mathbf{A})|}$ and $\mathbf{B} \leftarrow \mathbf{A}/\sigma$;
while *Iter* $\leq$ *MaxFE* **do**
  while $k = 1 \ldots \lambda$ **do**
    draw sample $\mathbf{s}_k \sim \mathcal{N}(0, \mathbb{I})$;
    $\mathbf{z}_k \leftarrow \boldsymbol{\mu} + \sigma \mathbf{B}^{\top}\mathbf{s}_k$;
    evaluate the fitness value
  end
  sort the sampling particles according to the fitness value and compute utilities function u_k according to (4)
  compute gradients according to (5)-(7)
end

4 Experiments and Analysis

In this section, GI-ES is used to compare with the state-of-the-art algorithms to verify the effectiveness of the proposed algorithm.

4.1 Experiment and Settings

Parameter Setting. All parameters used in GI-ES are as follows,

$\lambda = 4 + \lfloor 3\log(d) \rfloor$,
$\eta_\mu = \lfloor \frac{\lambda}{2} \rfloor$,
$\eta_\sigma = \eta_B = \frac{(9+3\log(d))}{5d\sqrt{d}}$,
$\eta_\delta = \frac{(3+\log(d))}{5\sqrt{d}}$.

Most of the parameters are recommended in [6].

In the algorithm, the number of population and the learning rate of gradient information are the parameters that need to be specified artificially.

Benchmark Function. Basic test problems. The test suite contains 11 classical problems that are widely used in evolutionary algorithms. As shown in Table 1 Sphere function is the simplest test function, which is used to test the basic performance of the algorithm. Ellipsoid Rosenbrock, and Cigar is a test of complex functions, used to test the function in the ill-conditioning, nonlinear scaling and flat region on the issue of test performance. Rotated function is through the rotating test function, matrix as references [14]. Persuasive in order to make the experiment. Each function is executed 21 times independently to record statistics. The end condition is the maximum number of iterations, MAXFE=10E08.

Table 1. Test problems

Set 1: Basic Test Problems	
Name	Functions
Sphere	$f_{\text{Sphere}}(\boldsymbol{x}) = \sum_{i=1}^{n} x_i^2$
Ellipsoid	$f_{\text{Elli}}(\boldsymbol{x}) = \sum_{i=1}^{n} 10^6 \frac{i-1}{n-1} x_i^2$
Rastrigin	$f_{\text{Ras}}(x) = 10n + \sum_{i=1}^{n} (x_i^2 - 10cos(2\pi x_i))$
Cigar	$f_{\text{Cigar}}(\boldsymbol{x}) = x_1^2 + 10^6 \sum_{i=2}^{n} x_i^2$
Rotated Ellipsoid	$f_{\text{RotElli}}(\boldsymbol{x}) = f_{\text{Elli}}(\boldsymbol{Rx})$
Rotated Cigar	$f_{\text{RotCigar}}(\boldsymbol{x}) = f_{\text{Cigar}}(\boldsymbol{Rx})$

* $\boldsymbol{R}$ is a rotation matrix by Gram-Schnidt orthogonalization

Algorithms for Comparison. CMA-ES and 2 algorithm variants for solving large scale optimization, including search direction adaptation evolution strategy(SDA-ES)[7]. Some other algorithms are not derived from CMA-ES, but are the state-of-the-art ones. For example, CC-based differential evolution (DECC-G) [19], the multiple offspring sampling(MOS)[9].

Effectiveness of the Gradient Information. The adaptation of the mutation strength is crucial for evolutionary calculation. It determines the direction of the next generation population and the convergence characteristics of the algorithm [1]. Before testing the overall performance of the GI-ES, we first investigate the effectiveness of gradient information. In the basic test section, the test algorithms we used were SDA-ES, MOS, DECC-G. The parameters of these algorithms are given in the original literature.

The 1000-dimensional sphere function f_{sphere} and Rastrigin function $f_{rastrigin}$ are used to test the effectiveness of gradient information. Sphere function is a spherical function, and many algorithms can be solved quickly, but the convergence performance of the algorithm is different. Rastrigin function is a relatively complex function with only one optimal solution, but there are many local optimizations in the fitness landscape of the function. In this part, we analyze the validity of gradient information by running GI-ES and ES-based algorithms.

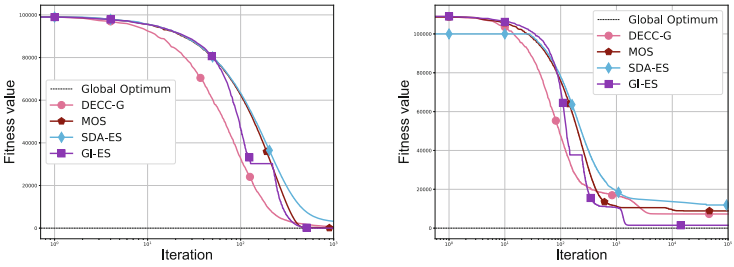
To avoid the sensitivity of the algorithm to the origin, the test function is operated in the experimental design and shifted by 10. Units away from the origin. As can be seen from Fig. 1, DECC-G converge faster in the initial stage, because the DECC-G adopts the group-based problem decomposition strategy for searching. However the time required to reach the optimal solution is longer than GI-ES. As can be seen from Fig. 1(a), in the subsequent convergence curve, GI-ES showed better convergence characteristics. For large scale optimization problems, the solution complexity is high and the computation is large. The Sphere function test shows that the use of historical information by GI-ES is useful for solving large scale optimization problems.

The rastrigin function is a very complex function. There are many local optimal solutions with disturbing properties in the adaptive terrain. As can be seen from Fig. 1(b), DECC-G converges faster in the early convergence process, and other algorithms gradually fall into local optimal solution as the convergence process proceeds. And GI-ES has a very good ability to jump out of the local optimal solution.

In sphere and rastrigin function tests, GI-ES was superior to other algorithms. This shows that for the application of single locally optimal objective function and multiple locally optimal objective functions, it is effective for GI-ES to use gradient information to determine the direction of the next generation of individuals in the iterative process.

Effectiveness of the GI-ES. In this part, to further verify the effectiveness of the proposed algorithm, the performance of the algorithm is tested on the 1000d basic test problems. In order to verify the invariance of the algorithm, some basic test functions are rotated. In general, the test of the rotation function can well illustrate the invariance of the algorithm. And the use of rotation functions is common in many mainstream test functions.

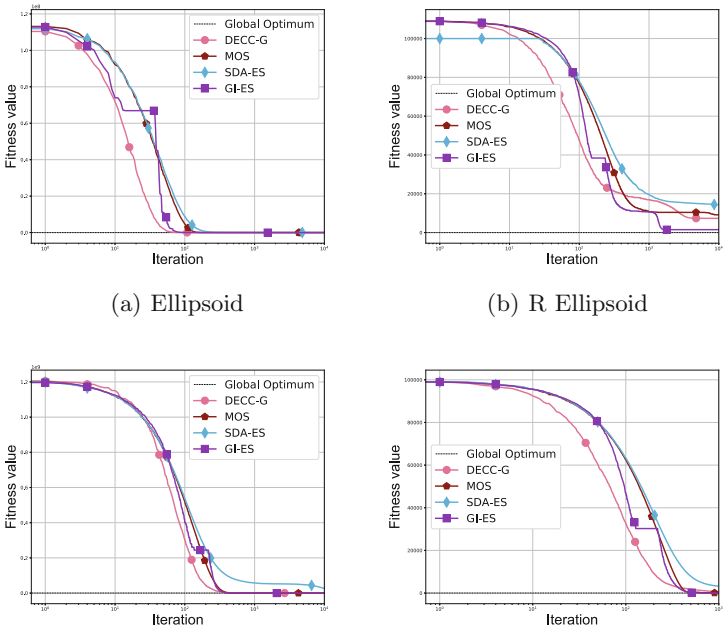
A visual representation of the convergence comparison of several algorithms is given in Fig. 2. In general, GI-ES is good for rotating and non-rotating functions.



(a) Sphere (b) Rastrigin

Fig. 1. Convergence plot on the 1000-d Sphere and Rastrigin function

Follows by DECC-G and MOS. On the Ellipsoid problem, GI-ES showed excellent performance in solving the rotation function. DECC-G shows sensitivity to the problem. DECC-G converges prematurely to the local optimal solution on the rotation function. However, GI-ES algorithm can jump out of local optimal solution in a short time after encountering local optimal solution, which benefits from the excellent ability of GI-ES algorithm to jump out of local optimal solution.



(a) Ellipsoid (b) R Ellipsoid

(c) Cigar (d) R Cigar

Fig. 2. Convergence plot on the 1000-d Ellipsoid and Cigar function

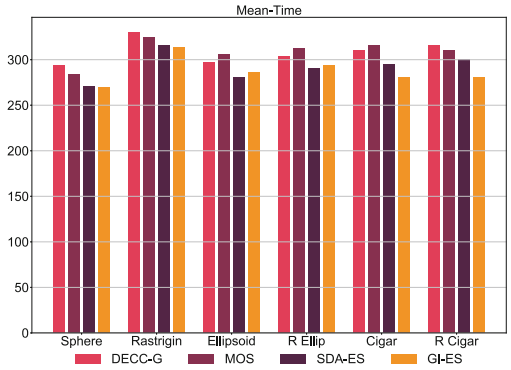


Fig. 3. Mean time consumed

Although the smooth of the cigar problem, it has a narrow ridge to be followed. And the overall shape deviates remarkably from being quadratic. DECC-G performs best for non-rotated cigar problem, but fails to GI-ES for rotated ones as shown in Fig. 2(c and d). In this case, the use of historical information can be a good guide for the algorithm to exploitation, so as to produce high-quality solutions.

The average running time of the algorithm is also a criterion that measures the performance of the algorithm. Figure 3 shows the average running time of the algorithm for each function. It can be seen from the figure that the average time of GI-ES on the Ellipsoid function is slightly higher than that of SDA-ES. For other functions, the average time of the GI-ES is less than that of other algorithms.

5 Conclusion

This research introduced the use of guiding gradient information to improve the performance of CMA-ES. The problem of low utilization of historical information by ES was solved by guiding the information to generate the distribution of the next generation solution. The guidance information was obtained by the approximation of the gradient. This strategy not only increased the diversity of knowledge, but also made full use of the optimal information in the heuristic algorithm. The theoretical analysis and experimental results showed that this method incorporating guidance information is accurate and stable.

The experimental results showed that the use of guiding information is effective. The algorithm was also compared with other typical meta-heuristic algorithms and demonstrated good average performance and pair-wise comparison performance across a wide range of test functions.

The experiments showed that this algorithm is an effective global optimization method for large scale problems, which makes it applicable to a large number of practical applications. The principle of using guidance information is simple, but effective, and has certain guiding significance for heuristic optimization algorithms.

References

1. Beyer, H.G., Hellwig, M.: The dynamics of cumulative step size adaptation on the ellipsoid model. Evol. Comput. **24**(1), 25–57 (2016)
2. Beyer, H.G., Schwefel, H.P.: Evolution strategies - a comprehensive introduction. Nat. Comput. **1**(1), 3–52 (2002). https://doi.org/10.1023/a:1015059928466
3. Bordes, A., Bottou, L., Gallinari, P.: SGD-QN: careful quasi-newton stochastic gradient descent. J. Mach. Learn. Res. **10**(Jul), 1737–1754 (2009)
4. Bringmann, K., Friedrich, T., Neumann, F., Wagner, M.: Approximation-guided evolutionary multi-objective optimization. In: Twenty-Second International Joint Conference on Artificial Intelligence (2011)
5. Hansen, N.: The CMA evolution strategy: a tutorial. arXiv preprint arXiv:1604.00772 (2016)
6. Hansen, N., Müller, S.D., Koumoutsakos, P.: Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evol. Comput. **11**(1), 1–18 (2003)
7. He, X., Zhou, Y., Chen, Z., Zhang, J., Chen, W.N.: Large-scale evolution strategy based on search direction adaptation. IEEE Trans. Cybern., 1–15 (2019). https://doi.org/10.1109/tcyb.2019.2928563
8. Holland, J.H.: Genetic algorithms. Sci. Am. **267**(1), 66–73 (1992)
9. LaTorre, A., Muelas, S., Pena, J.M.: Multiple offspring sampling in large scale global optimization. In: 2012 IEEE Congress on Evolutionary Computation. IEEE, June 2012. https://doi.org/10.1109/cec.2012.6256611
10. Lehman, J., Chen, J., Clune, J., Stanley, K.O.: Safe mutations for deep and recurrent neural networks through output gradients. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 117–124. ACM (2018)
11. Li, J., Tan, Y.: Information utilization ratio in heuristic optimization algorithms. arXiv preprint arXiv:1604.01643 (2016)
12. Maheswaranathan, N., Metz, L., Tucker, G., Sohl-Dickstein, J.: Guided evolutionary strategies: escaping the curse of dimensionality in random search. arXiv preprint arXiv:1806.10230 (2018)
13. Roubos, J., van Straten, G., van Boxtel, A.: An evolutionary strategy for fed-batch bioreactor optimization; concepts and performance. J. Biotechnol. **67**(2–3), 173–187 (1999). https://doi.org/10.1016/s0168-1656(98)00174-6
14. Salomon, R.: Evolutionary algorithms and gradient search: similarities and differences. IEEE Trans. Evol. Comput. **2**(2), 45–55 (1998). https://doi.org/10.1109/4235.728207
15. Sehnke, F., Osendorfer, C., Rückstieß, T., Graves, A., Peters, J., Schmidhuber, J.: Parameter-exploring policy gradients. Neural Netw. **23**(4), 551–559 (2010)
16. Storn, R., Price, K.: Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. **11**(4), 341–359 (1997). https://doi.org/10.1023/A:1008202821328

17. Wierstra, D., Schaul, T., Peters, J., Schmidhuber, J.: Natural evolution strategies. In: 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence). IEEE, June 2008. https://doi.org/10.1109/cec.2008.4631255
18. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Mach. Learn. **8**(3–4), 229–256 (1992). https://doi.org/10.1007/bf00992696
19. Yang, Z., Tang, K., Yao, X.: Large scale evolutionary optimization using cooperative coevolution. Inf. Sci. **178**(15), 2985–2999 (2008). https://doi.org/10.1016/j.ins.2008.02.017
20. Zeugmann, T., et al.: Particle swarm optimization. In: Sammut, C., Webb, G.I. (eds.) Encyclopedia of Machine Learning, pp. 760–766. Springer, Heidelberg (2011). https://doi.org/10.1007/978-0-387-30164-8_630