On evolutionary optimization of large problems with small populations

Yaochu Jin, Markus Olhofer, Bernhard Sendhoff

2005

On Evolutionary Optimization of Large Problems Using Small Populations

Yaochu Jin, Markus Olhofer, and Bernhard Sendhoff

Honda Research Institute Europe, Carl-Legien-Str. 30,
63073 Offenbach/Main, Germany
yaochu.jin@honda-ri.de

Abstract. Small populations are very desirable for reducing the required computational resources in evolutionary optimization of complex real-world problems. Unfortunately, the search performance of small populations often reduces dramatically in a large search space. To addresses this problem, a method to find an optimal search dimension for small populations is suggested in this paper. The basic idea is that the evolutionary algorithm starts with a small search dimension and then the search dimension is increased during the optimization. The search dimension will continue to increase if an increase in the search dimension improves the search performance. Otherwise, the search dimension will be decreased and then kept constant. Through empirical studies on a test problem with an infinite search dimension, we show that the proposed algorithm is able to find the search dimension that is the most efficient for the given population size.

1 Introduction

To reduce the computational time in solving expensive optimization problems using evolutionary algorithms, a commonly adopted approach is to parallelize the fitness evaluation process so that each individual is evaluated on a separate machine. In this case, use of a relatively small population size will be very helpful in reducing the computational cost for the evolutionary optimization.

Unfortunately, we are left in a dilemma when we use small populations for solving complex real-world problems. On the one hand, many real-world optimization problems, e.g., design optimization where splines are used to describe the geometry of a structure [1], have a very large number of design parameters. On the other hand, the search efficiency decreases seriously when small populations are used to optimize problems with a high search dimension.

Two approaches could be employed to alleviate, if not solve, the difficulty mentioned above. One method is to develop an efficient evolutionary algorithm with a small population size whose performance is less sensitive to the search dimension. One good example is the derandomized evolution strategy with covariance matrix adaptation (CMA-ES) [2], which has shown to be robust on various unimodal test functions. Nevertheless, the search efficiency of the CMA-ES still greatly depends on the population size. A conclusion from empirical studies is that the population size should be scaled between linear and cubic with the problem dimension to locate the global optimum [3].

Another method is to adapt the search dimension to the population size in use. To this end, an adaptive coding scheme has been suggested where the CMA-ES is employed in aerodynamic shape optimization [4]. The basic idea is to encode the number of parameters to be optimized (the search dimension) in the chromosome and to mutate during the optimization. One issue that arises in the adaptive coding scheme is that the self-adaptation of the evolution strategy can be disturbed due to the mutation in the search dimension, which is harmful to the search performance. One measure to address this problem is to ensure that the mutations are neutral, i.e., the shape of the geometry will be kept the same before and after a new point is inserted in the spline representation.

In this paper, we will explicitly monitor the performance change after the search dimension is increased. If the increase in the search dimension is beneficial, the search dimension will be further increased. Otherwise, the search dimension will be decreased and then will be kept constant until the end of the optimization. To minimize the disturbance on the self-adaptation mechanism, the dimension is increased only by 1 in each change in dimension. Through simulations on various population sizes, it is shown that our method is able to find an optimal or nearly optimal search dimension for the given population size on a test problem with an infinite search dimension.

The test problem used in this study will be briefly described in Section 2. The search capacity of the CMA-ES with regard to the population size on the test problem are investigated empirically in Section 3. The algorithm to find the optimal search dimension is given in Section 4 and a number of simulations are conducted in Section 5, where we show that the algorithm is able to find the optimal or sub-optimal search dimension for different population sizes. Conclusion and further research topics are discussed in Section 6.

2 Test Problem

The test problem used in this study is very simple. However, it serves our purpose well where an infinitely large search dimension is needed theoretically. We consider the approximation of a one-dimensional function using a Taylor series. If a function $f(x)$ has continuous derivatives, then this function can be expanded as follows:

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)(x-a)^2}{2!} + \cdots + \frac{f^{(n)}(a)(x-a)^n}{n!} + R_n, \quad (1)$$

where R_n is the remainder after $n+1$ terms defined by:

$$\begin{aligned} R_n &= \int_a^x f^{(n+1)}(u)\frac{(x-u)^n}{n!}du \\ &= \frac{f^{(n+1)}(\xi)(x-a)^{n+1}}{(n+1)!}, \end{aligned} \quad (2)$$

where $a < \xi < x$. When this expansion converges over a certain range of x, i.e., $lim_{n \to} R_n = 0$, then the expansion is known as em Taylor Series of function $f(x)$ about a. For example, the Taylor expansion of sine function is as follows:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots, -\infty < x < \infty. \tag{3}$$

The optimization problem is to find the coefficients of the Taylor series by minimizing the squared approximation error:

$$E(x) = (\sum_{i=0}^{n} a_i x^i - \sin(x))^2, \tag{4}$$

where x is the point about which the Taylor series is expanded, $a_i, i{=}0,1,2,\cdots,n$ is the number of terms of the Taylor series. Theoretically, an infinite search dimension is needed to realize a perfect approximation of a sinusoidal function using Taylor series. To estimate the approximation error reliably, we sample 100 points uniformly within the range of $0 \leq x \leq 1$:

$$E = \sum_{j=1}^{100} E(x_j). \tag{5}$$

An interesting fact in the above test function is that the influence of each term on the function value decreases as the order increases. Thus, terms in the Taylor expansion are added in the search algorithm from lower orders to higher ones. This is reasonable because in optimization of real-world problems, we try to account for at first the most important factors and then try to include those with minor influence.

3 Search Efficiency of Small EAs

As we mentioned in the Introduction, the derandomized evolution strategy with covariance matrix adaptation (CMA-ES) proposed in [5] was designed for small populations. It has shown to be efficient on a large number of unimodal optimization problems, particularly on ill-conditioned and non-separable problems [2]. In the (μ, λ)-CMA-ES without recombination, the λ offspring of generation $g + 1$ is generated as follows:

$$\mathbf{x}_k^{(g+1)} = \mathbf{x}_j^{(g)} + \sigma^{(g)} \mathbf{B}^{(g)} \mathbf{D}^{g)} \mathbf{z}_k^{(g+1)},\ j = 1, \cdots, \mu;\ k = 1, \cdots, \lambda, \tag{6}$$

where k is randomly chosen from the μ selected parents, $\mathbf{z}$ is an n-dimensional (n is the search dimension) vector of normally distributed random numbers with expectation zero and identity covariance matrix, $\mathbf{BD}\left(\mathbf{BD}\right)^T = \mathbf{C}$ is the covariance matrix. During the evolution, the covariance matrix is updated as follows:

$$\mathbf{C}^{(g+1)} = (1 - c_{\text{cov}})\mathbf{C}^{(g)} + c_{\text{cov}} \mathbf{p}_{\text{C}}^{(\mathbf{g+1})} \left(\mathbf{p}_{\text{C}}^{(\mathbf{g+1})}\right)^T, \tag{7}$$

where $\mathbf{p}_{\mathrm{C}}^{(\mathbf{g+1})}$ is known as the evolution path calculated by:

$$\mathbf{p}_{\mathrm{C}}^{(g+1)} = (1 - c_{\mathrm{C}})\mathbf{p}_{\mathrm{C}}^{(g)} + \sqrt{c_{\mathrm{C}} \cdot (2 - c_{\mathrm{C}})}\mathbf{B}^{(g)}\mathbf{D}^{(g)}\mathbf{z}_{k}^{(g)}. \tag{8}$$

The adaptation of the global step-size $\sigma^{(g+1)}$ is calculated by:

$$\sigma^{(g+1)} = \sigma^{(g)}\exp\left(\frac{1}{d_\sigma}\frac{||\mathbf{p}_\sigma^{(g)}|| - c\hat{h}i_n}{\hat{\chi}_n}\right), \tag{9}$$

where $\hat{\chi}_n$ is the expected length of a $(\mathbf{0}, \mathbf{I})$-normally distributed random vector and can be approximated by $\sqrt{n}(1 - \frac{1}{4n} - \frac{1}{21n^2})$, d_σ is a damping coefficient, and $\mathbf{p}_\sigma^{(g+1)}$ is a "conjugate" evolution path:

$$\mathbf{p}_\sigma^{(g+1)} = (1 - c_\sigma)\mathbf{p}_\sigma^{(g)} + \sqrt{c_\sigma \cdot (2 - c_\sigma)}\mathbf{B}^{(g)}\mathbf{z}_k^{(g)}. \tag{10}$$

The default parameter setting suggested in [2] is as follows:

$$c_{\mathrm{C}} = \frac{4}{n+4},\; c_{\mathrm{COV}} = \frac{2}{(n+\sqrt{2})^2},\; c_\sigma = \frac{4}{n+4}, d_\sigma = c_\sigma^{-1} + 1. \tag{11}$$

In this study, a slightly modified variant of the algorithm presented in [5] has been adopted, where a separate covariance matrix is maintained for each parent individual. Though the CMA-ES is designed for small populations, recent studies have found that CMA-ESs with a large population can improve the search performance significantly [3,6].

However, little work has been reported on what is the optimal search dimension for a CMA-ES with a small population size when the theoretic search dimension is very large or even infinite. In the following, we investigate the search performance of CMA-ES with small populations on the test problem described in Section 2. In our simulations, the CMA-ES without recombination has been adopted and a maximum of 2000 generations are run for search dimensions $5, 7, \cdots, 47, 49$. For each search dimension, the results are averaged over 50 independent runs. The results from a (1, 4)-CMA-ES and a (2, 10)-CMA-ES are presented in Figures 1 and 2, respectively.

From Fig. 1, we can see that the search performance of the (1, 4)-CMA-ES heavily depends on the search dimension. For a search dimension smaller than 7, the approximation error is quite large due to the limited number of free parameters. The minimal approximation error (0.002936) is achieved when the search dimension is 11, where the approximation error is mostly smaller than 0.01. When the search dimension further increases, the search performance degrades seriously due to the limited search capacity of the (1,4)-CMA-ES.

Similar simulations are carried out for the (2, 10)-CMA-ES. The minimal approximation error (0.000003) is achieved when the search dimension is 11. This implies that the (1, 4)-CMA-ES failed to locate the global optimum for an eleven-dimensional optimization problem in 50 runs. Even the (2, 10)-CMA-ES is able to locate the best found solution only once in the 50 runs. These results indicate that the search efficiency of CMA-ES with small populations is

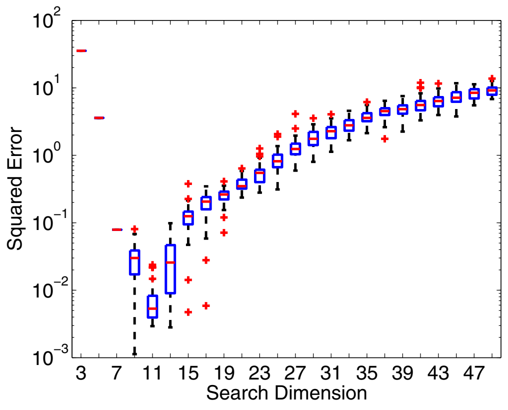


Fig. 1. Search performance of the (1,4)-CMA-ES for search dimensions ranging from 3 to 49. Results averaged over 50 runs.

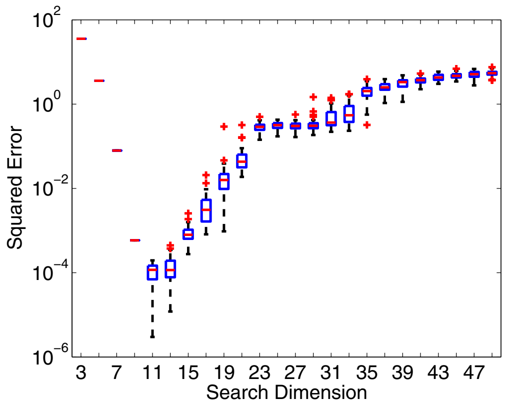
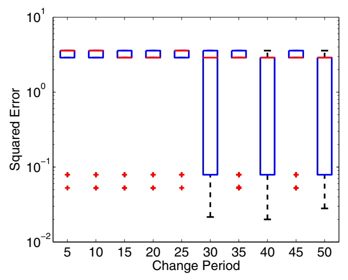


Fig. 2. Search performance of the (2, 10)-CMA-ES for search dimensions ranging from 3 to 49. Results averaged over 50 runs.

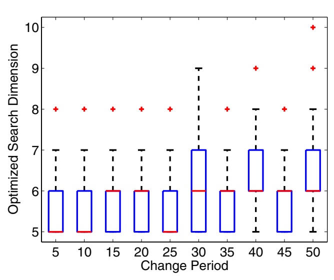
limited even for a relatively low dimensional problem. Meanwhile, as in the (1, 4)-CMA-ES case, the search performance becomes worse when the search dimension increases, though not as serious as the (1, 4)-CMA-ES. Again, there is an optimal search dimension where the (2, 10)-CMA-ES achieves the best performance and the search performance is acceptable when the search dimension is from 9 up to 17 (approximation error smaller than 0.01).

4 Adaptation of Search Dimension

It can be seen from the results in the previous section that there is an optimal search dimension for a given population size that is able to achieve the minimal



(a)



(b)

Fig. 3. Adaptation of search dimension for (1, 4)-CMA-ES with various change periods. The design parameters are randomly re-initialized during dimension changes. (a) The best fitness value, and (b) the optimized search dimension. Results averaged over 50 runs.

approximation error. The optimal search dimension is unknown beforehand and is presumably dependent on the population size and the problem at hand.

In this section, we suggest a simple approach to address this problem by adapting the search dimension during the optimization to find an approximately optimal search dimension for a given population size. The basic idea is to start the optimization from a relatively low search dimension and let the search dimension increase in every k generations during the optimization. k is called *change period*. To determine whether an increase in search dimension is beneficial, we compare the best fitness values before and after dimension increase. Assume the best (minimal in this work) fitness values before and after an increase in search dimension are $PBest$ and $CBest$, respectively. Note that $CBest$ is the best fitness value after k generations with an increased search dimension. The increase in search dimension is considered to be beneficial if $CBest$ is smaller than $Pbest$ for minimization problems. If an increase in dimension is regarded as beneficial, then the search dimension will be further increased by one. Otherwise, the search dimension will be decreased by one and fixed until the end of the optimization.

To implement the above idea, the change period k needs to be determined. We conduct simulations to investigate the influence of this parameter on the adaptation performance. Another parameter to be determined is the initial search dimension. This parameter should depend on the problem at hand. In our simulations, the initial dimension is set to 5.

When the search dimension is increased, we have the following three alternatives:

– Re-initialize all design parameters randomly;
– Inherit the value for existing design parameters and initialize new design parameter randomly;
– Inherit the value for existing design parameters and set the new parameter to zero, so that the fitness function does not change after the inclusion of the new dimension.

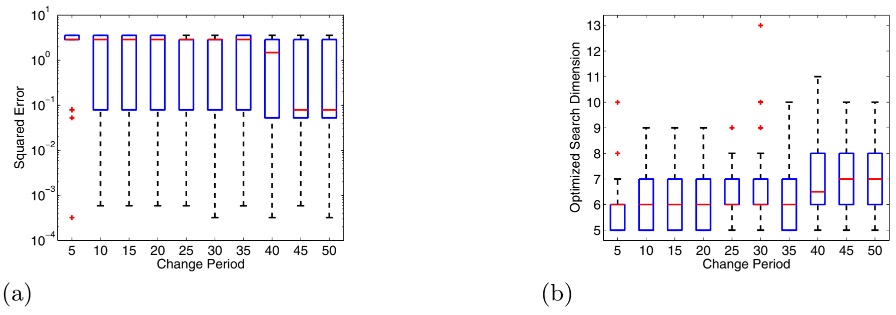


Fig. 4. Adaptation of search dimension for (2, 10)-CMA-ES with various change periods. The design parameters are randomly re-initialized during dimension changes. (a) The best fitness value, and (b) the optimized search dimension. Results averaged over 50 runs.

We test the performance of the suggested algorithm for 10 change periods, i.e., $k = 5, 10, 15, 20, 25, 30, 35, 40, 45, 50$. The results where all design parameters are randomly initialized are presented in Figures 7 and 8, respectively. Again, 50 runs are conducted for each k.

From Fig. 3 and Fig. 4, we see that neither the (1, 4)-CMA-ES nor the (2, 10)-CMA-ES shows acceptable performance. The search dimension is largely underestimated for all tested change periods. A much larger change period is not practical, since an overly large period will unfavorably increase the needed computational time. Thus, we conclude that randomly re-initialize the design parameters is undesirable in adopting an adaptive search dimension.

The next idea to try out is to inherit the value for each existing design parameter and then initialize the newly added design variable randomly. The simulation results are shown in Figures 5 and 6, respectively. We notice that the performance has been improved significantly. For the (1, 4)-CMA-ES, the per-

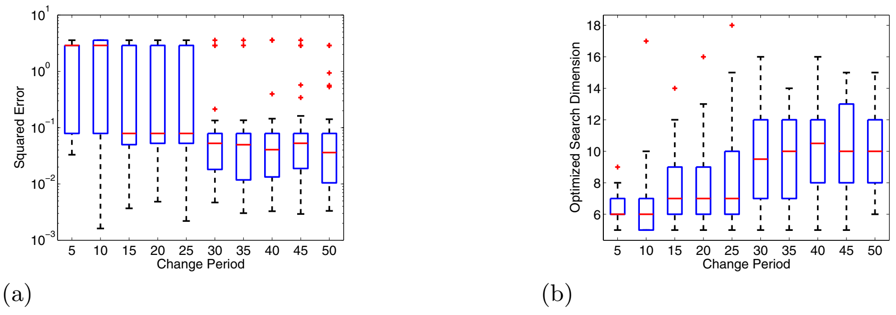
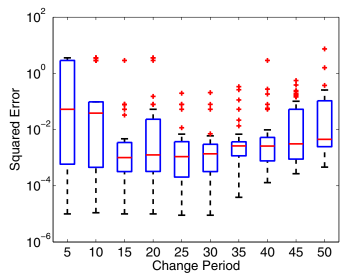
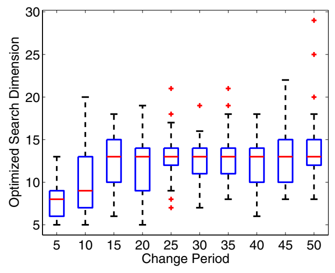


Fig. 5. Adaptation of search dimension for (1, 4)-CMA-ES with various change periods. The value of the existing design parameters are inherited and the new one is randomly initialized during dimension change. (a) The best fitness value, and (b) the optimized search dimension. Results averaged over 50 runs.

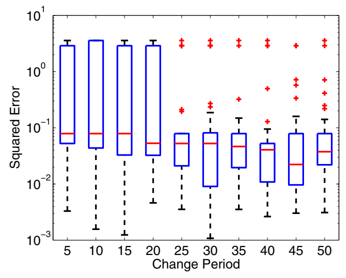


(a)

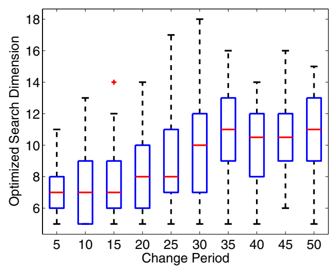


(b)

Fig. 6. Adaptation of search dimension for (2, 10)-CMA-ES with various change periods. The value of the existing design parameters are inherited and the new one is randomly initialized during dimension changes. (a) The best fitness value, and (b) the optimized search dimension. Results averaged over 50 runs.



(a)

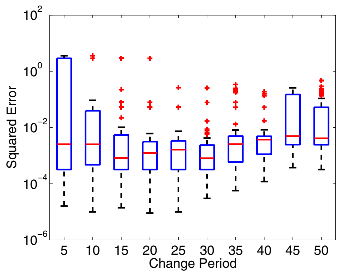


(b)

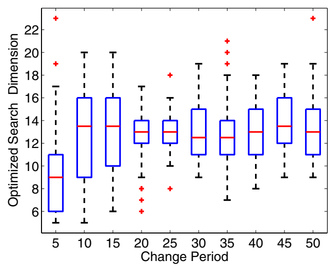
Fig. 7. Adaptation of search dimension for (1, 4)-CMA-ES with various change periods. The value of the existing design parameters are inherited and the new one is set to zero during dimension changes. (a) The best fitness value, and (b) the optimized search dimension. Results averaged over 50 runs.

formance is quite good when the change period is between 30 and 50, where the optimized search dimension is between 9 and 11 on average, which are optimal or sub-optimal if we refer to the empirical results shown in Fig. 1. Similar conclusion can be made to the results obtained for the (2, 10)-CMA-ES. However, the performance of the algorithm seems more robust against the change period in that satisfying performance has been achieved when the change period varies from 15 to 50, where the estimated optimal search dimension is 13 on average, which is one of the optimal search dimension as shown in Fig. 2.

Finally, we investigate the performance of the algorithm when we initialize the newly added design parameter to 0, which in this example makes the inclusion of the new search dimension neutral to the fitness value. Such neutral mutations have shown to be essential to the success of adaptive coding when splines are used for geometry description in design optimization [4]. Comparing



(a)



(b)

Fig. 8. Adaptation of search dimension for (2, 10)-CMA-ES with various change periods. The value of the existing design parameters are inherited and the new one is set to zero during dimension changes. (a) The best fitness value, and (b) the optimized search dimension. Results averaged over 50 runs.

the results in Fig. 5 and those in Fig. 7 regarding the (1, 4)-CMA-ES, we see that minor improvements have been achieved, particularly when the change period is small.

5 Conclusions

Evolutionary optimization of large problems with evolutionary algorithms with a small population is a challenging topic. To efficiently optimize possibly infinite large problems using small populations, a method to adapt the search dimension has been suggested in this paper. The basic idea is that for small populations, we should start from a relatively low search dimension and then increase it gradually during the optimization. The increase in search dimension should continue until performance improvement cannot be achieved in a number of generations after the dimension increase. In this case, the search dimension is decreased by one and and kept constant till the end of the optimization. From our empirical studies, the change period should be between 20 to 50 generations. A too small change period is not desirable because the algorithm needs some time to find the potential improvement after an increase in dimension. Neither is a large change period preferred because a larger change period tends to increase the computational time rapidly.

It is found essential for the success of our algorithm that the value of the existing design parameters should be inherited after an increase in search dimension. This result is consistent with the findings reported in the literature that *a priori* knowledge is beneficial in enhancing the performance of evolutionary algorithms [7].

The strategy parameters are randomly re-initialized during dimension changes in this work, which may not be optimal for evolution strategies. It will be one of our future work to investigate the influence of re-initialization of strategy parameters on the performance of our algorithm using a dynamic search dimension.

References

1. Jin, Y., Olhofer, M., Sendhoff, B.: A framework for evolutionary optimization with approximate fitness functions. IEEE Transactions on Evolutionary Computation. **6** (2002) 481–494
2. Hansen N., Ostermeier A.: Completely derandomized self-adaptation in evolution strategies. Evolutionary Computation **9** (2001) 159–195
3. Hansen, N., Kern S.: Evaluating the CMA evolution strategy on multimodal test functions. Parallel Problem Solving from Nature, Vol. 3242. Springer (2004) 282–291
4. Olhofer, M., Jin, Y., Sendhoff, B.: Adaptive encoding for aerodynamic shape optimization using evolution strategies. Congress on Evolutionary Computation (2001) 576–583
5. Hansen, N., Ostermeier, A.: Adapting arbitrary normal mutation distributions in evolution strategies: The covariance Matrix Adaptation. IEEE Conf. on Evolutionary Computation (1996) 312–317
6. Hansen, N., Müller, S., Koumoutsakos, P.: Reducing the time complexity of the derandomized evolution startegy with covariance matrix adaptation (CMA-ES). Evolutionary Computation. **11** (2003) 1–18
7. Jin, Y. (ed.): Knowledge Incorporation in Evolutionary Computation. Springer, Berlin Heidelberg (2005)