Generalized Self-Adapting Particle Swarm Optimization algorithm with model-based optimization enhancements

Mateusz Zaborski[1], Michał Okulewicz[1], and Jacek Mańdziuk[1]

Warsaw University of Technology
Faculty of Mathematics and Information Science
{M.Zaborski,M.Okulewicz,J.Mandziuk}@mini.pw.edu.pl

Abstract. This paper investigates the performance of an improved version of Generalized Self-Adapting Particle Swarm Optimization (GAPSO) – a hybrid global optimization framework. In particular, the possibility of utilizing model-based optimization in parallel with sampling-based methods (like PSO or DE) within GAPSO framework is discussed. The research on GAPSO approach is based on two assumptions: (1) it is possible to improve the performance of an optimization algorithm through utilization of more function samples than standard PSO sample-based memory, (2) combining specialized sampling methods (i.e. Particle Swarm Optimization, Differential Evolution, locally fitted functions) will result in a better algorithm performance than using each of them separately. The inclusion of these model-based enhancements indicated the necessity of extending GAPSO framework by means of an external samples memory - this enhanced model is referred to as M-GAPSO in the paper. The key feature of M-GAPSO is collection of already computed function samples in R-Tree based index and their subsequent use in model-based enhancements. Moreover, M-GAPSO incorporates a global swarm restart mechanism from JADE algorithm, instead of resetting each particle separately, as opposed to GAPSO. COCO benchmark set is used to assess the M-GAPSO performance against the original GAPSO and the state-of-the-art KL-BIPOP-CMAES algorithm.

Keywords: Particle Swarm Optimization · global optimization · metaheuristics

1 Introduction

Particle Swarm Optimization (PSO) [2] is a well-known global optimization metaheuristic with many possible variants. For instance, Nepomuceno and Engelbrecht [3] proved that appropriate mix of heterogeneous versions of PSO can lead to significant performance improvement. Yamaguchi and Akimoto [6] presented the usage of search history for more efficient algorithm initialization after restart. The above works confirmed that various optimization enhancements and storing samples in memory are all promising directions of global optimization research. This work presents an approach which combines both of the above-mentioned features.

2 M-GAPSO framework description

This section describes the proposed Generalized Self-Adapting Particle Swarm Optimization framework with external samples memory (M-GAPSO), which is an enhancement of the GAPSO approach [5].

The GAPSO optimization framework has been designed on the basis of PSO. It allows the usage of virtually any optimization algorithm behavior, whose performance is evaluated during the optimization process. The enhancements that brought the highest improvement to the estimated optimum and used relatively more frequently.

Within GAPSO, *particles* act independently and may behave differently. From the swarm's point of view internal behavior of the *particle* (i.e. function sampling scheme) is irrelevant. Each *particle* is only obliged to update its velocity and maintain its current and best positions. Therefore, the well-known algorithms, such as Differential Evolution (DE), can be easily implemented within GAPSO framework by means of the appropriate scheme for updating the velocity vector.

However, in order to include efficient model-based enhancements, an additional external memory module has to be implemented for storing the already sampled function values. Moreover, the implementation of the new features revealed that a global JADE-like restart mechanism [4] is more beneficial for the algorithm's performance, than the original GAPSO's particle-by-particle method.

M-GAPSO is maintained in the publicly available source code repository[1].

2.1 External memory

Gathering samples (understood as point coordinates and function values at these points) is a key enhancement compared to the initial work on GAPSO [5]. The main idea is to take advantage of already gathered samples and use them in model-based optimization enhancements. In order to store and retrieve samples in an efficient way M-GAPSO utilizes a multidimensional R-Tree index. It allows quick access to the desired subset of samples, such as surrounding of a selected point.

2.2 Model-based optimization enhancements

Model-based enhancements (quadratic as well as polynomial models) have been applied in order to support quick convergence to local optima. In both cases the same principles of particle's behavior are applied. At the beginning, the model is fitted using specified sample collection. Then, the algorithm finds a function

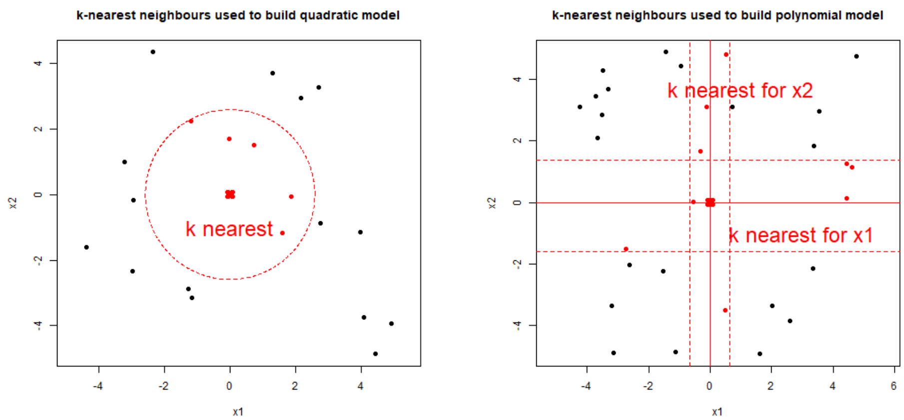

Fig. 1. Comparison of a samples data sets used for fitting quadratic and polynomial models

optimum in relation with the field boundaries. Finally, the particle is moved to coordinates that match the estimated optimum.

Quadratic model is fitted on a data set composed of k nearest samples (in the sense of the Euclidean metric) to a $particle.x_{best}$ location for which the quadratic behavior has been selected. See Fig. 1 as an example. Quadratic function-based approach fits the following model:

$$\hat{f}_{quadratic.local}(x) = \sum_{d=1}^{dim} \left(a_d x_d^2 + b_d x_d\right) + c \tag{1}$$

Polynomial model enhances the quadratic model in the following way:

$$\hat{f}_{polynomial.local}(x_d) = \sum_{i=1}^{p} a_{i,d} x_d^i + c \tag{2}$$

Furthermore, the polynomial model is fitted on separated data sets in each dimension. These data sets are made of k samples closest to a line with coordinates fixed to the current location, except for the dimension d for which the model is currently fitted. The differences between methods of gathering samples (quadratic vs. polynomial) are depicted in Fig. 1.

3 Results and future work

The enhancements implemented in M-GAPSO improved performance over GAPSO. Moreover, results became comparable with the state-of-the-art CMA-ES[6], mostly for lower dimensions. Comparisons for 5 and 20 dimensions are shown in Fig. 2.

[1] https://bitbucket.org/pl-edu-pw-mini-optimization/basic-pso-de-hybrid

Evaluation was made on 24 noiseless continuous functions from COCO BBOB benchmark data set [1].

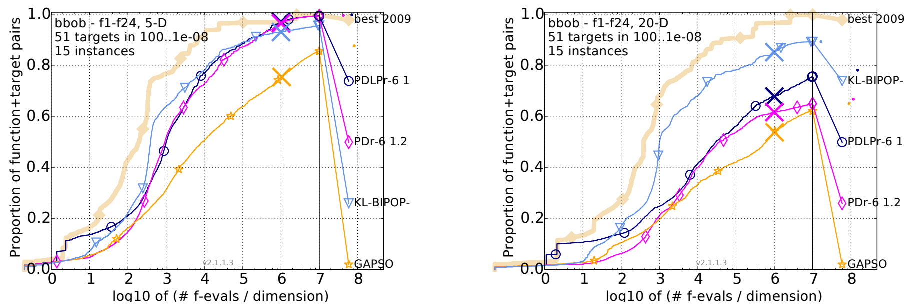

Fig. 2. Results of M-GAPSO configurations including model-based optimization (PDLPr) and without model based optimization (PDr) against GAPSO and state-of-the-art variation of CMA-ES for $DIM * 10^6$ optimization budget.

On a general note, M-GAPSO results are promising, although, many improvements can still be applied, in particular other local methods for handling samples gathered in external memory, as well as global modeling schemes.

References

1. Hansen, N., Brockhoff, D., Mersmann, O., Tusar, T., Tusar, D., ElHara, O.A., Sampaio, P.R., Atamna, A., Varelas, K., Batu, U., Nguyen, D.M., Matzner, F., Auger, A.: COmparing Continuous Optimizers: numbbo/COCO on Github (2019). https://doi.org/10.5281/zenodo.2594848, https://doi.org/10.5281/zenodo.2594848
2. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. Proceedings of IEEE International Conference on Neural Networks. IV pp. 1942–1948 (1995)
3. Nepomuceno, F.V., Engelbrecht, A.P.: A Self-adaptive Heterogeneous PSO Inspired by Ants. In: International Conference on Swarm Intelligence, pp. 188–195 (2012). https://doi.org/10.1007/978-3-642-32650-9_17
4. Poaík, P., Klema, V.: JADE, an adaptive differential evolution algorithm, benchmarked on the BBOB noiseless testbed. In: Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference companion - GECCO Companion '12. p. 197. ACM Press, New York, New York, USA (2012). https://doi.org/10.1145/2330784.2330814
5. Uliński, M., Zychowski, A., Okulewicz, M., Zaborski, M., Kordulewski, H.: Generalized Self-adapting Particle Swarm Optimization Algorithm. In: Lecture Notes in Computer Science, vol. 3242, pp. 29–40. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99253-2_3
6. Yamaguchi, T., Akimoto, Y.: Benchmarking the novel CMA-ES restart strategy using the search history on the BBOB noiseless testbed. In: GECCO '17 Proceedings of the Genetic and Evolutionary Computation Conference Companion. pp. 1780–1787 (2017). https://doi.org/10.1145/3067695.3084203