Platform Design of Particle Swarm Optimization Algorithm based on Scilab

Rui Qi[1,2], Paul-Henry Cournède[1,3], and Baogang Hu[2]
[1] *Ecole Centrale Paris, Laboratory of Applied Mathematics, 92295, Chatenay-Malabry, France*
[2] *LIAMA/NLPR, Institute of Automation, Chinese Academy of Sciences, 100080, Beijing, China*
[3] *INRIA Saclay Ile-de-France, EPI Digiplante, Parc Orsay Universit´e, 91893 Orsay cedex, France*
qiruitree@gmail.com
hubaogang@gmail.com
paul-henry.cournede@ecp.fr

Abstract

As fast as industries are developing, optimization algorithms encounter more and more challenges. On one hand, some optimization issues may have no explicit expressions and solutions. Their gradient information cannot be gained analytically. On the other hand, dimensionality of problems becomes higher and higher. Especially, optimization procedure may be required to be performed based on multi-modal data. In recent research, multi-objective optimization problems become interesting issues, whose optimal solutions called Pareto front [1] are not unique. Thus, classical optimization algorithms are sometimes not competent. Rather than explicit ways, those problems are tackled successfully using heuristic optimization algorithms.

In order to apply heuristic optimization algorithms conveniently and widely, it is better to integrate them into toolbox in open source platforms, so that they could be modified by users as they want without charge, and could be distributed and shared freely. The Particle Swarm Optimization (PSO) algorithm is an iterative, population-based heuristic optimization algorithm, which originally simulates the behavior of bird flocking by Kennedy and Eberhart in 1995 [2]. Generally, PSO has a faster convergence rate than other heuristic algorithms for some problems [3] and it has few parameters to adjust. As a result, we design a Particle Swarm Optimization Toolbox in an open source platform for numerical computation Scilab (PSOTS), which is the main contribution of this work. In comparison with other existing open-source PSO toolboxes which are developed on non-scientific computing software, PSOTS will be more convenient for beginners and users to learn and to improve the given algorithm.

PSOTS has a friendly interface, which is developed in TCL/TK language, in order to be convenient for users to set parameter values. PSOTS can either maximize or minimize optimization problems optionally, without transforming the formula of optimization problems. It includes five variants of PSO. It considers different strategies for choosing the best particle, including global PSO [1], where the best particle is searched in the whole swarm and local PSO[4], where the best particle is searched in part of swarm. Other two variants aim to improve convergence rate and precise of standard PSO, namely Constriction PSO (CPSO) [5] and PSO with Passive Congregation (PSOPC) [6]. The last variant of PSO is used specifically for traveling salesman problem [7]. PSOTS does not require the types of optimized parameters, either real values, or integer values or mixed values. Convergence curves in real-time and particle distribution in the searching space are optional in PSOTS.

Key words: Particle Swarm Optimization, open source, Scilab, numerical computation

References

[1] S. Mostaghim and J. Teich, “Strategies for Finding Good Local Guides in Multi-objective Particle Swarm Optimization (MOPSO),” in *IEEE Swarm Intelligence Symposium*, 2003, pp. 26–33.
[2] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proc. IEEE Conf. on Neural Networks*, vol. IV, Piscataway, NJ, 1995, pp. 1942–1948.
[3] ——, *Swarm Intelligence*. Morgan Kaufmann Publishers, 2001.
[4] M. Song and G. Gu, “Research on particle swarm optimization: A review,” in *Proc. the Third International Conference on Machine Learning and Cybernetics*, Shanghai, 2004, pp. 2236–2241.
[5] M. Clerc and J. Kennedy, “The Particle Swarm: Explosion, Stability and Convergence in a Multi-Dimensional Complex Space,” *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 58–73, 2002.
[6] S. He, Q. Wu, J. Wen, J. Saunders, and R. Paton, “A particle swarm optimizer with passive congregation,” *BioSystem*, vol. 78, pp. 135–147, 2004.
[7] M. Clerc, “Discrete particle swarm optimization – illustrated by the traveling salesman problem,” 2002, http://www.mauriceclerc.net.