A new algorithm hybridizing differential evolution with particle swarm optimization

Jin Xue, K-J. Wang, Phuc Hong Nguyen, and Shaw Yi. Wu
Department of Industrial Management
National Taiwan University of Science & Technology, Taipei, Taiwan
Tel: (+886) 2-2737-6769, Email: kjwang@mail.ntust.edu.tw, 15876517929@163.com, nhphuc72@gmail.com, quae7031@gmail.com

Abstract. This study proposes a novel hybrid algorithm named PSO-DE, which integrates particle swarm optimization (PSO) algorithm with differential evolution (DE) algorithm. This hybridizing approach aims to combine two algorithms in a judicious manner such that the resulting algorithm contains the positive features of both the algorithms. For example, speeds up the convergence and improves the algorithm's performance. Experimental results show that our solution approach outperforms DE, PSO, and genetic algorithm (GA).

Keywords: Differential evolution, particle swarm optimization, hybrid algorithm.

1. Introduction

Optimization problems are of great importance for the industrial as well as various spheres of human activities, where decisions need to be taken in some complex situation that can be represented by a mathematical model. In the practical utility of optimization problems, there is a need for efficient and robust computational algorithms, which can numerically solve on computers the mathematical models of a medium as well as large size optimization problem arising in industrial (Akbaripour and Masehian, 2013). In the past few decades, evolutionary algorithms such as differential evolution (DE), PSO and GA have been very popular for programming model (Kachitvichyanukul, 2012).

There has been a growing popular in evolutionary algorithms for diverse fields of industries. Optimization algorithms such as DE, PSO, are popular for making decisions. However, despite having several attractive features, it has been observed that when the number of the decision variables is large and the solution space is complicated, these algorithms do not always perform well. Some people found hybrid algorithms exploit the good properties of different methods by applying them to problems they can always solve problems more efficiently than a single algorithm. The focus of the present study is the hybrid algorithm in which integrates PSO and DE.

2. Literature review

2.1 Differential evolution

DE is a population heuristic algorithm, which is simple to implement, requires little or no parameter tuning and is known for its remarkable performance for combinatorial optimization (Chang, 2009). This algorithm had first introduced to solve the optimization by Storn and Price (1995). DE creates new candidate solutions by perturbing the parent individual with the weighted difference of several other randomly chosen individuals of the same population. A candidate replaces the parent only if it is better than its parent. Thereafter, DE guides the population towards the vicinity of the optimum through repeated the cycles of mutation, crossover and selection. DE has several advantages: it can search randomly, requires only fewer parameters setting, high performance and applicable to high-dimensional complex optimization problems (Wu et al., 2011). A disadvantage of DE is that DE has no mechanism to memorize the previous process. Therefore, it easily results in a waste of computing power and gets tapped in local optima (Hao et al., 2007).

2.2 Particle swarm optimization

PSO is a stochastic population based optimization approach, which is inspired by social interaction of animals living in group likes birds, fish, termites, ants and even human beings. The conception of PSO has first published by Kennedy (2011). Since its first publication, a large body of research has been done to study the performance of PSO, and to improve its performance (Van den Bergh and Engelbrecht, 2006). PSO has become one of the most promising optimizing techniques for solving optimization problems. The PSO system consists of a population of potential solutions called particles. These particles move through the search domain with a specified velocity in search of optimal solution. Each particle maintains a memory, which helps it keep the track of its previous best position. The positions of the particles are distinguished as personal best and global best. In the past several years, PSO has been successfully applied in many research and application area (Thangaraj et al., 2009). The details of PSO are given in Kennedy (2011). PSO is efficient, its ability to handle optimization problems with multiple local optima reasonably well and its simplicity of implementation (Meissner et al., 2006). It also has some critical problems such as it easily stuck in local optimal when updating personal best and global best after finding the best position in the overall population (Hao et al., 2007).

3. Algorithm development

The idea of this approach is to use DE as a global searcher while PSO will work as a local searcher. The procedure of the proposed DE-PSO is illustrated in Figure 1.

First, the proposed algorithm starts with initialization in which the initial population, with size Z, is generated. Second, it will be followed by DE operations where crossover, mutation and infeasible repairing mechanism will be performed. The role of DE is to create a high level of diversity of the DE-PSO algorithm.

Third, we will select *n* number of best solutions achieved from the DE operation. Each of these points is considered as a G-best point. Given each G-best point, PSO algorithm will perform and generate one new population. Therefore, we will have totally *n* new populations. The next step is updating new population where we will select Z number of best solutions from *n* population that have just generated by PSO. Finally, if one of stopping conditions is satisfied, the algorithm will be stopped. Otherwise, the new population will be sent to the DE, and the DE-PSO operation will be repeated.

4. Experiments

To verify the proposed DE-PSO, this section examines several important concerns regarding algorithmic design in applying DE-PSO and model properties. In this experiment, all programs are coded in Matlab and executed on a computer with Intel core i7-3770, CPU 3.4GHz and 12 GB RAM.

An experiment design is conducted to examine the DE_PSO, DE, PSO, GA and random search (RS) algorithms in two merits: CPU seconds consumed and fitness value. Those five methods are applied to the problem based onWang et al. (2008). Three sets of parameters of DE, PSO, GA algorithms are suggested by literatures (Prett and Morari, 2013; Meadowcroft, 1992; Wang and Wang, 2013) and the value of parameters of DE-PSO are shown in Table 1.

Figure 2 shows that the GA and RS computed the solution quickly; however, their performance is the worst among those algorithms to investigate the objective function, even the worst solution found by DE-PSO is better than the best solution reported in. DE and PSO performance better than GA and RS and CPU time is better than DE-PSO. DE-PSO outperforms the two compared approach, in terms of quality and SD while robustness is not as well as the two approaches mentioned above. DE-PSO performance is significantly better than the others.

5. Conclusion

In this study, a novel algorithm DE-PSO is proposed. The strategy makes DE-PSO have the advantages of two algorithms and maintain diversity of the population. The performance of the DE-PSO algorithm is compared with DE, PSO, GA, which demonstrate that it is a powerful optimization algorithm with rapid convergence rate, high solution quality and algorithm robustness. The major contribution of this study is to present a hybrid algorithm that can approach an optimal solution while consuming a tiny amount of computational time as compared to other common algorithms for large-size problems.

In comparison with the DE-PSO, PSO, DE, GA, RS, DE-PSO is more effective in obtaining better quality solutions in a more effective way, and finds better quality solutions more frequently.

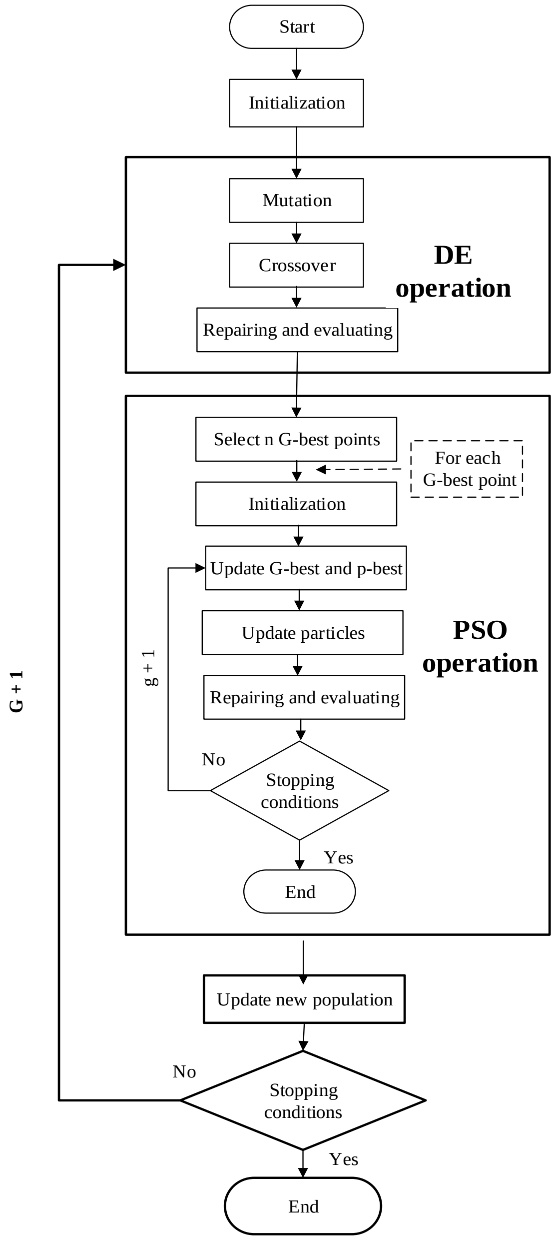

Figure 1 DE-PSO algorithm.

Table 1 Control factors of DE－PSO

Parameter	Value
Population size (M)	80
Mutation scale (F)	0.5
Crossover probability (CR)	0.2
G_best points	5
Inertia weight (w)	0.4
Positive constant (C_1&C_2)	1.5
Time limit	1600

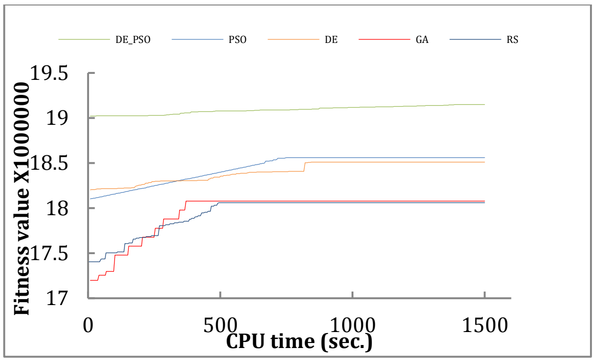

Figure 2 Comparing of the results of 5 algorithms

Appendix

The model by Wang et al. (2008): A nonlinear stochastic optimization model is developed to maximize the expected profit under demand uncertainty. For solution efficiency, a stochastic programming-based genetic algorithm (SPGA) is proposed to determine a profitable capacity planning and task allocation plan. The algorithm improves a conventional two-stage stochastic programming by integrating a genetic algorithm into a stochastic sampling procedure to solve this large-scale nonlinear stochastic optimization on a real-time basis. Finally, the tradeoff between profits and risks is evaluated under different settings of algorithmic and hedging parameters. Experimental results have shown that the proposed algorithm can solve the problem efficiently.

Reference

Akbaripour, H., & Masehian, E. (2013). Efficient and robust parameter tuning for heuristic algorithms. *International Journal of Industrial Engineering & Production Research, 24*(2), 143-150.

Atamtürk, A., & Hochbaum, D. S. (2001). Capacity acquisition, subcontracting, and lot sizing. *Management Science, 47*(8), 1081-1100.

Bard, J. F., Srinivasan, K., & Tirupati, D. (1999). An optimization approach to capacity expansion in semiconductor manufacturing facilities. *International Journal of Production Research, 37*(15), 3359-3382.

Bashyam, T. (1996). Competitive capacity expansion under demand uncertainty. *European Journal of Operational Research, 95*(1), 89-114.

Berman, O., & Larson, R. C. (1994). Determining optimal pool size of a temporary call-in work force. *European Journal of Operational Research, 73*(1), 55-64.

Biswal, B., Dash, P. K., & Mishra, S. (2011). A hybrid ant colony optimization technique for power signal pattern classification. *Expert Systems with Applications, 38*(5), 6368-6375.

Chang, W.-D. (2009). Two-dimensional fractional-order digital differentiator design by using differential evolution algorithm. *Digital Signal Processing, 19*(4), 660-667.

Cuevas, E., Zaldivar, D., & Pérez-Cisneros, M. (2010). A novel multi-threshold segmentation approach based on differential evolution optimization. *Expert Systems with Applications, 37*(7), 5265-5271.

Dong, R. (2009). *Differential evolution versus particle swarm optimization for PID controller design.* Paper presented at the Natural Computation, 2009. ICNC'09. Fifth International Conference on.

Hao, Z.-F., Guo, G.-H., & Huang, H. (2007). *A particle swarm optimization algorithm with differential evolution.* Paper presented at the Machine Learning and Cybernetics, 2007 International Conference on.

Hendtlass, T. (2001). A combined swarm differential evolution algorithm for optimization problems *Engineering of intelligent systems* (pp. 11-18): Springer.

Kachitvichyanukul, V. (2012). Comparison of three evolutionary algorithms: GA, PSO, and DE. *Industrial Engineering and Management Systems, 11*(3), 215-223.

Kennedy, J. (2011). Particle swarm optimization *Encyclopedia of machine learning* (pp. 760-766): Springer.

Kumar, A., & Rajeev, G. (2013). R. Compare the Results of Tuning of PID Controller by Using PSO and GA Technique for AVR System. *Intern. J. Adv. Res. Comput. Engin. Technol, 2*, 2131-2138.

Liu, B., Wang, L., & Jin, Y.-H. (2008). An effective hybrid PSO-based algorithm for flow shop scheduling with limited buffers. *Computers & Operations Research, 35*(9), 2791-2806.

Meissner, M., Schmuker, M., & Schneider, G. (2006). Optimized Particle Swarm Optimization (OPSO) and its application to artificial neural network training. *BMC bioinformatics, 7*(1), 125.

Saad, M. S., Jamaluddin, H., & Darus, I. Z. M. (2012). Implementation of PID controller tuning using differential evolution and genetic algorithms. *International Journal of Innovative Computing Information and Control, 8*(11), 7761-7779.

Storn, R., & Price, K. (1995). *Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces* (Vol. 3): ICSI Berkeley.

Thangaraj, R., Pant, M., & Abraham, A. (2009). *Evolutionary Algorithms based speed optimization of servo motor in optical disc systems.* Paper presented at the Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on.

Törn, A., & Žilinskas, A. (1989). Global optimization.

Uzsoy, R., Ramcharan, D. J., & Martin-Vega, L. A. (1991). An experimental expert system for process planning of sheet-metal parts. *Computers & industrial engineering, 20*(1), 59-69.

Van den Bergh, F., & Engelbrecht, A. P. (2006). A study of particle swarm optimization particle trajectories. *Information sciences, 176*(8), 937-971.

Wang, K.-J., & Wang, S.-M. (2013). Simultaneous resource portfolio planning under demand and technology uncertainty in the semiconductor testing industry. *Robotics and Computer-Integrated Manufacturing, 29*(5), 278-287.

Wang, K.-J., Wang, S.-M., & Chen, J.-C. (2008). A resource portfolio planning model using sampling-based stochastic programming and genetic algorithm. *European Journal of Operational Research, 184*(1), 327-340.

Wu, C.-H., & Chuang, Y.-T. (2010). An innovative approach for strategic capacity portfolio planning under uncertainties. *European Journal of Operational Research, 207*(2), 1002-1013.

Wu, Y.-C., Lee, W.-P., & Chien, C.-W. (2011). *Modified the performance of differential evolution algorithm with dual evolution strategy.* Paper presented at the International Conference on Machine Learning and Computing.