Towards a Repulsive and Adaptive Particle Swarm Optimization Algorithm

Simone A. Ludwig
Department of Computer Science
North Dakota State University
Fargo, USA
simone.ludwig@ndsu.edu

ABSTRACT

This paper proposes a Repulsive Adaptive PSO (RAPSO) variant that adaptively optimizes the velocity weights of every particle at every iteration. RAPSO optimizes the velocity weights during every outer PSO iteration, and optimizes the solution of the problem in an inner PSO iteration. We compare RAPSO to Global Best PSO (GBPSO) on nine benchmark problems, and the results show that RAPSO outperforms GBPSO on difficult optimization problems.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Particle Swarm Optimization

Keywords

Evolutionary computation, benchmark problems, adaptive particle swarm optimization

1. INTRODUCTION

There are many adaptive PSO variants that iteratively change all or some of the weights. For example, PSO with dynamic adaption [1] uses an evolutionary speed factor measuring personal best value changes in order to calculate the inertia weight. Another example is given in [2], where the inertia weight of every particle is based on its objective value, the global best value, and the global worst value. [3] changes its inertia weight based on swarm diversity to reduce premature convergence. The swarm diversity is calculated as a function of positions. Different variations of the self-tuning adaptive PSO are given in [4, 5, 6].

Self-tuning adaptive PSO [5] assigns every particle its own personal best weight, and global best weight. Self-tuning adaptive PSO initializes the personal best weights and the global best weights randomly for every particle, and moves the personal and global best weights towards values of the particle that yielded the most updates of the global best position based on the total number of iterations [4].

2. REPULSIVE AND ADAPTIVE PSO

Our Repulsive and Adaptive PSO (RAPSO) variant is inspired by other adaptive PSO variants that assign every particle its own velocity weights [5, 6]. All these variants move the velocity weights of all particles toward the velocity weights of a certain particle that is selected based on a measure of superior performance [6].

For example, self-tuning adaptive PSO moves the velocity weights towards the settings of the particle that yielded the most updates of the global best position [5, 6]. Controlled APSO [7] adaptively changes the personal best weights and the global best weights based on the distance between the positions and the global best position. Inertia weight adaptive PSO [2] allows every particle its own inertia weight that is changed using a function of the objective values and the global best value. Optimized PSO [8] uses multiple PSO subswarms, each having their own parameter settings, in an inner iteration to solve the original optimization problem. The parameter settings are then optimized in an outer iteration of PSO for a fixed number of iterations.

Inspired by the optimized PSO variant [8], we treat the problem of finding good velocity weights as an optimization problem. In RAPSO every particle has its own velocity weights, i.e., its inertia weight, personal best weight, and global best weight. A particular setting of the velocity weights is referred to as the position of the velocity weights. An objective function for the velocity weights is used to quantify how well the positions of the velocity weights perform for solving the overall optimization problem. Using the calculated objective values of the velocity weights, RAPSO takes a step toward optimizing the velocity weights. The velocity weights are optimized in a fixed auxiliary search space.

3. EXPERIMENTS AND RESULTS

3.1 Global Best PSO

Global Best PSO (GBPSO) [9] is used for comparison and works as follows. The particle positions and velocities are first randomly initialized within the search space [10]. Then, the objective values of the particles are calculated. The global best value and global best position are set to the objective value and position of the particle with the best objective value in the entire swarm. Velocities for all particles are then calculated, and are moved to their new positions. The objective values are evaluated again. Personal best positions are updated for particles that have a new objective value that is better than their previous personal best value.

3.2 Benchmark Problems

Nine optimization benchmark problems are used to compare our RAPSO algorithm with the GBPSO algorithm.

GECCO'13, July 6-10, 2013, Amsterdam, The Netherlands.
ACM TBA.

Table 1: Performance for benchmark F1 to F9 for 10,000, 100,000 and 1,000,000 FE.

Algorithm	FE	Quality	F1	F2	F3	F4	F5	F6	F7	F8	F9
GBPSO	10,000	Best	9.18e-01	6.10e+01	1.32e+03	1.11e+02	1.68e+00	0.00e+00	2.84e-14	3.19e+03	3.00e+00
		Mean	1.49e+00	7.30e+01	**1.61e+03**	1.23e+02	**1.75e+01**	6.48e-03	**3.79e-14**	**5.28e+03**	**9.33e+00**
		Std	4.17e-01	1.09e+02	6.37e+04	1.34e+02	2.27e+02	3.15e-05	2.69e-28	5.78e+06	6.03e+01
RAPSO	10,000	Best	4.50e-01	4.51e+01	2.08e+03	4.22e+01	7.72e+01	0.00e+00	5.68e-14	6.46e+03	2.90e+01
		Mean	**1.03e+00**	**7.28e+01**	2.47e+03	**5.75e+01**	2.05e+02	**0.00e+00**	7.58e-14	1.02e+04	2.00e+02
		Std	4.32e-01	5.77e+02	1.70e+05	1.76e+02	1.40e+04	0.00e+00	2.69e-28	1.06e+07	3.12e+04
GBPSO	100,000	Best	5.16e-01	1.10e+01	1.10e+02	8.26e+01	4.28e+00	0.00e+00	0.00e+00	2.02e+00	0.00e+00
		Mean	8.18e-01	3.53e+01	1.53e+02	9.29e+01	**1.29e+01**	**0.00e+00**	**2.84e-14**	2.12e+02	2.33e+00
		Std	7.69e-02	4.64e+02	4.47e+03	1.35e+02	5.91e+01	0.00e+00	2.42e-27	1.30e+05	1.03e+01
RAPSO	100,000	Best	0.00e+00	5.06e+00	5.46e+01	3.28e+01	2.32e+01	0.00e+00	2.84e-14	1.89e-02	0.00e+00
		Mean	**1.33e-02**	**9.02e+00**	**1.14e+02**	**5.60e+01**	2.32e+01	**0.00e+00**	5.68e-14	**4.78e-01**	**0.00e+00**
		Std	5.29e-04	1.28e+01	6.74e+03	4.53e+02	3.16e-04	0.00e+00	8.08e-28	1.79e-01	0.00e+00
GBPSO	1,000,000	Best	2.53e-01	6.00e+00	4.42e-04	2.59e+01	7.57e-01	0.00e+00	0.00e+00	4.25e-18	0.00e+00
		Mean	6.05e-01	1.10e+01	7.03e-01	4.94e+01	4.23e+00	**0.00e+00**	**3.79e-14**	8.31e-16	**0.00e+00**
		Std	1.30e-01	4.90e+01	1.48e-02	5.25e+02	2.34e+01	0.00e+00	1.08e-27	1.08e-30	0.00e+00
RAPSO	1,000,000	Best	0.00e+00	0.00e+00	4.42e-04	1.09e+01	1.73e-03	0.00e+00	0.00e+00	0.00e+00	0.00e+00
		Mean	**1.18e-15**	**3.33e-01**	**7.03e+00**	**1.23e+01**	**1.36e+00**	**0.00e+00**	**3.79e-14**	**0.00e+00**	**0.00e+00**
		Std	1.05e-30	3.33e-01	1.48e-02	2.31e+00	5.25e+00	0.00e+00	1.88e-27	0.00e+00	0.00e+00

The following benchmark functions were used: Michalewicz (F1), Non-continous Rastrigin and multimodal (F2), Parabola (F3), Rastrigin (F4), Rosenbrock (F5), Schaffer's F6 (F6), Shubert (F7), Sphere (F8), and Step (F9).

3.3 Results

Analyzing the results, as shown in Tables 1 (best mean values are given in bold), reveals that RAPSO improves with increasing numbers of Function Evaluations (FE), scoring best compared to GBPSO for 100,000 and 1,000,000 FE.

For 10,000 FE, GBPSO scores best in terms of best mean value on 5 benchmark functions, and RAPSO scores best on only 4 benchmark functions. Starting with 100,000 FE and higher a trend favoring RAPSO can be seen. The table shows that GBPSO scores best on 3 benchmark functions, and RAPSO scores best on 7 benchmark functions. For 1,000,000 FE, GBPSO has the best mean value for 3 benchmark functions, and RAPSO scores best on 9 benchmark functions.

For 1,000,000 and 100,000 FE, the optimum value of 0.0 was achieved by RAPSO, measuring the average value, on 3 benchmark functions, and for 10,000 FE only on 1 benchmark function. This demonstrates that with increasing numbers of FE more benchmark functions are solved optimally.

4. CONCLUSION

We proposed a repulsive and adaptive PSO (RAPSO) algorithm, for which every particle has its own velocity weights. An objective function for the velocity weights is used to measure the suitability of the velocity weights for solving the overall optimization problem, and thereby improving the optimization process. The advantage of RAPSO is that the velocity weights adapt themselves to dynamic changes, e.g., different particle distributions at different iterations.

We evaluated our RAPSO algorithm on nine benchmark functions and compared it with global best PSO (GBPSO). Our RAPSO variant outperforms GBPSO for higher numbers of FE in particular for 100,000 and 1,000,000 FE.

However, since the comparison is done only with GBPSO, a more thorough evaluation needs to be conducted. For example, RAPSO has to be compared with an adaptive PSO variant as well as with more benchmark problems in order to better demonstrate the strength of the RAPSO approach.

5. REFERENCES

[1] X. Yang, J. Yuan, J. Yuan, and H. Mao. A modified particle swarm optimizer with dynamic adaptation. *Applied Mathematics and Computation*, 189(2):1205–1213, 2007.

[2] J. Zhu, J. Zhao, and X. Li. A new adaptive particle swarm optimization algorithm. *International Workshop on Modelling, Simulation and Optimization*, 456–458, 2008.

[3] Y. Bo, Z. Ding-Xue, and L. Rui-Quan. A modified particle swarm optimization algorithm with dynamic adaptive. *2007 Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2:346–349, 2007.

[4] T. Yamaguchi and K. Yasuda. Adaptive particle swarm optimization; self-coordinating mechanism with updating information. *IEEE International Conference on Systems, Man and Cybernetics, 2006. SMC '06*, 3:2303–2308, 2006.

[5] T. Yamaguchi, N. Iwasaki, and K. Yasuda. Adaptive particle swarm optimization using information about global best. *IEEE Transactions on Electronics, Information and Systems*, 126:270–276, 2006.

[6] K. Yasuda, K. Yazawa, and M. Motoki. Particle swarm optimization with parameter self-adjusting mechanism. *IEEE Transactions on Electrical and Electronic Engineering*, 5(2):256–257, 2010.

[7] A. Ide and K. Yasuda. A basic study of adaptive particle swarm optimization. *Denki Gakkai Ronbunshi / Electrical Engineering in Japan*, 151(3):41–49, 2005.

[8] M. Meissner, M. Schmuker, and G. Schneider. Optimized particle swarm optimization (OPSO) and its application to artificial neural network training. *BMC Bioinformatics*, 7(1):125, 2006.

[9] A. Engelbrecht. *Computational Intelligence — An Introduction 2nd Edition*. Wiley, 2007.

[10] A. Ratnaweera, S.K. Halgamuge, and H.C. Watson. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Transaction on Evolutionary Computation*, 8(3):240–255, 2004.