An Efficient Fitness-based Stagnation Detection Method for Particle Swarm Optimization

Hazem Radwan Ahmed
School of Computing, Queen's University
25 Union Street, Kingston, Ontario,
Canada K7L 3N6.
hazem@cs.queensu.ca

ABSTRACT

Stagnation is a prevalent issue in many heuristic search algorithms, such as Particle Swarm Optimization (PSO). PSO stagnation occurs when the rate of position changes (or velocities) that attract particles to the global best position approaches zero, potentially leading the swarm to being trapped in a local optimum, especially for deceptive multimodal optimization problems. This paper proposes a novel fitness-based stagnation detection method that effectively and efficiently restarts the search process to escape potential local optima. The main idea of the proposed method is to make use of the already calculated fitness values of swarm particles, instead of their pairwise distance values, to predict an imminent stagnation situation. That is, the proposed fitness-based method does not require any computational overhead of repeatedly calculating pairwise distances between all particles at each iteration. The proposed fitness-based method substantially outperforms the commonly used distance-based method when tested on several classical and advanced (shifted/rotated) benchmark optimization functions in three ways: 1) The optimization performance is significantly better performing (using Wilcoxon rank-sum test). 2) The optimization performance is considerably faster (up to three times). 3) The proposed fitness-based method is less dependent on the problem search space, compared with the distance-based method.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*

Keywords

multi-start particle swarm optimization; fitness-based stagnation detection; search diversification; speedup technique.

GECCO'14, July 12–16, 2014, Vancouver, BC, Canada..

ACM 978-1-4503-2881-4/14/07…$15.00.
http://dx.doi.org/10.1145/2598394.2605669

1. INTRODUCTION

Drawing inspiration from the sociological behavior associated with bird flocking, Particle Swarm Optimization (PSO) has had growing scientific attention in many diverse domains [16] since its inception in 1995 [10]. While PSO algorithms showed efficient, robust and fast-convergence behavior on several optimization problems, they suffered from a number of key issues, such as swarm explosion and swarm stagnation. The early versions of PSO suffered from the swarm explosion issue [4], in which particle velocities indefinitely grow, causing divergence of swarm particles for some values of the inertia and learning coefficients. Among early attempts to address this issue was the 'velocity clamping' strategy, which limits the velocity magnitude to a maximum velocity threshold. This strategy avoids increasing particle velocities indefinitely and prevents particles from taking extremely large shifts from their current position, realistically simulating the incremental change of human learning [9].

A better strategy to address the swarm explosion issue is the use of a 'constriction coefficient', which was first proposed by Clerc in 1999 [2]. Then shortly after, in 2000, Eberhart and Shi [6] analyzed the convergence behavior of PSO and suggested the popular settings of the constriction and learning coefficients (i.e., χ= 0.729 and c_1=c_2=2.05), which ensures that the particle velocities will shrink over time, rather than indefinitely growing to infinity. While the Constriction Coefficient PSO (CC-PSO) guarantees convergence to a stable point, there is no guarantee that this point is a quality point in the search space. Thus, Van der Berg, in 2002 [18], introduced the Guaranteed Convergent PSO (GC-PSO) to address this issue. Nonetheless, the effectiveness of the GC-PSO was only remarkable on unimodal optimization problems, rather than complex multimodal problems.

Among earlier attempts to better handle multimodal problems was the Species-based PSO (SPSO), proposed by Li in 2004 [11]. Although SPSO worked fairly well on multimodal function optimization using sub-swarm (or species), it was challenging to define the best species radius, especially for multimodal problems with little (or no) prior knowledge about their search space. A number of years later, Van den Bergh provided a formal convergence proof for PSO [19], and argued that the use of a multi-start strategy can potentially convert the PSO algorithm from a local to global optimizer, even on complex multi-modal optimization problems.

This paper is organized as follows: Section 2 provides a brief background on different stagnation detection methods used in the literature to restart PSO. Section 3 describes the proposed fitness-based stagnation detection method for PSO. Section 4 discusses the experimental results and observations. The last section concludes the study and suggests possible future work.

2. BACKGROUND

2.1 Stagnation Detection Methods

PSO stagnation, or premature convergence, issue occurs when the private experiences of the particles (or, their individual best positions) do not change relative to the swarm social experience (or, the global best position) for a number of iterations [3]. This critical issue vanishes the rate of position changes (or velocities) that attract particles to the global best position prematurely, making the swarm prone to being trapped into local optima, especially for deceptive multimodal optimization problems.

Different stagnation detection methods have been studied in the literature [21] to restart the search process of PSO, such as Maximum Swarm Radius, Cluster Analysis and Objective Function Slope [17]. The Maximum Swarm Radius method restarts the swarm when the most-distant particle from the global best particle reaches a predefined minimum distance, while the Cluster Analysis method restarts the swarm when the majority of swarm particles (e.g., >60%) reaches a predefined minimum distance from the global best particle. Both the Maximum Swarm Radius and Cluster Analysis are distance-based stagnation detection methods that check how close the particles are to the global best particle. The third method, Objective Function Slope, detects stagnation and restarts the swarm when the change rate in the objective function remains negligible for a predefined number of iterations. It has been claimed that distance-based methods outperform the Objective Function Slope method [7]; thus, the Maximum Swarm Radius method was adopted in Regrouping PSO [7], as well as in a number of other more recent studies [14; 23]. The main disadvantages, however, of the distance-based methods is that they are computationally expensive and dependent on the range of the search space [15].

On the contrary, the proposed stagnation detection method restarts particle positions and resets their velocity and memory using a criterion that is not based on the relative distances between particles and the global best particle to the problem search range (as in the Maximum Swarm Radius and Cluster Analysis methods), nor based on the change rate in the objective function (as in the Objective Function Slope method). Instead, the proposed method is based on the swarm-wide performance on the objective function with relative to the global best performance. That is why the proposed fitness-based method is more efficient and less dependent on the search range, compared with the commonly used distance-based methods for stagnation detection.

3. METHODOLOGY

The proposed stagnation detection method uses a more efficient fitness-based criterion to effectively trigger what we called the "Agile Restart" mechanism, as needed. In general, agility is the ability to change the body's position rapidly and efficiently as required, which is basically what the proposed "Agile Restart" attempts to accomplish when particles are trapped in a local optimum. The proposed method uses a Performance-based Stagnation Indicator (PSI) based on Average Swarm Fitness to define a potential stagnation situation to be imminent when the fitness of the global best particle becomes almost the same as the average fitness of the entire swarm, as shown in (1). The proposed fitness-based criterion has a computational advantage over the distance-based criterion of O(nk) for a swarm of n particles in k number of iterations, since it makes use of the already calculated fitness values of swarm particles to trigger the restart mechanism, instead of calculating their pairwise distances at each iteration. Interestingly, the efficiency of the proposed criterion does not come at the cost of the solution quality. The proposed PSI criterion showed competitive results when compared with the distance-based Maximum Swarm Radius criterion, on several classical and advanced benchmark optimization functions.

$$f(P_g) \geq \psi * Average\,(f(P_i)) \qquad (1)$$

Where: f is the objective (minimization) function, $f(P_i)$ is the corresponding fitness vector of particles' historically best positions, $f(P_g)$ is the fitness value of the global best particle position, and ψ is the *PSI threshold*, which is a positive percentage value defined by the user to adaptively control the restart mechanism. In particular, unlike some methods that reset the PSO algorithm every fixed K number of iterations [20], the proposed method does not trigger the restart mechanism according to some artificially fixed cycles of equal length; it is rather dynamically triggered (as needed) according to the '*run-time'* behaviour of the search process. The smaller the value of ψ, the more frequent the restart mechanism will be triggered. Thus, PSI value clearly affects the algorithm sensitivity to triggering the restart mechanism. It was empirically found that this approach works well when the PSI threshold is set between 90% for low-dimensional problems and 70% for high-dimensional problems. The reason why higher-dimensional problems generally require lower PSI threshold is because of the fact that the chance of stagnation situation usually increases as the problem dimensionality increases. It is perhaps worth mentioning that the proposed stagnation detection method was successfully applied to a real-life computational biology problem in our recent work [1].

4. RESULTS AND DISCUSSION

The proposed fitness-based stagnation detection method is tested on a set of 8 classical and advanced (shifted/rotated) benchmark optimization functions, and compared with the commonly used distance-based stagnation detection method on the same set of general optimization benchmarks. This set is sufficient to include various classes of optimization problems with different regularity, modality, separability and dimensionality [7; 22], such as Ackley, Griewank, Quadric, Rastrigin, Rosenbrock and Spherical functions [7], each at 3 to 30 dimensions. A function is called unimodal if it only has one global optimum with no local optima, whereas a function is called multimodal if it has more than one local optimum, besides at least one global optimum. A function of variables is called separable if it can be rewritten as a sum of functions of only a single variable [8].

Three performance evaluation metrics are considered in this experiment: 1) the achieved solution quality or the algorithm's effectiveness, 2) the search effort or the number of function evaluations required to reach the solution, and 3) the algorithm's efficiency, represented as the CPU time needed to reach the solution. It is worth emphasizing here that the goal of this experiment is not to compare two different Multi-Start PSOs (MPSOs) but to compare the effect of using the popular distance-based stagnation detection method with the proposed fitness-based method on any MPSO model. In this paper, an MPSO model based on the Constriction Coefficient PSO (CC-PSO) [4] was used for all of the comparisons. The performance was recorded over 50 independent runs (with different random seeds) per benchmark with a maximum of 800,000 function evaluations per run, using a swarm size of 20 and Clerc's constriction model with χ= 0.729 and c_1=c_2=2.05.

The first comparison metric that we considered in this experiment is the solution quality, or algorithm's effectiveness. We, therefore, performed the Wilcoxon rank-sum test [5], using Matlab statistical toolbox, to validate if (and when) the proposed fitness-based stagnation detection methods is significantly better performing (i.e., P-Value < 0.05, h=1 and Z-Score > 0) [13] than the commonly used distance-based stagnation detection method over the 50 conducted independent runs. Table 1 confirms that the proposed fitness-based method is indeed significantly better performing than the distance-based method, except for a few highlighted values in bold and red color (i.e., when Z-Score < 0 and P-Value < 0.05).

In particular, as shown in Table 1, the proposed fitness-based stagnation detection methods outperformed the distance-based method on most tested benchmark functions at lower dimensionality (n=3 and n=10), except for the Shifted Rosenbrock and Rotated Rastrigin functions. Nonetheless, while the proposed method achieved better performance (Z>0) in the classical Griewank function, the improved performance in this particular case (at n=10) is not, but almost, significantly validated at 95% confidence level (P = 0.0672). Concerning higher-dimensional problems (at n=30), the proposed method showed significantly better performance on unimodal functions, such as Quadric and Spherical functions. As for multimodal functions at higher dimensions (i.e., n=30), the proposed method showed overall better performance on Ackley and Shifted Rosenbrock, and showed significantly better performance on the Rotated Rastrigin benchmark.

The second comparison metric is the search effort or the number of function evaluations required to find the optimal solutions. The termination criterion of the search process is considered the sooner of reaching a maximum of 800,000 function evaluations per run, or reaching a 64-bit, double-precision value of zero, which is the global optimum on all tested benchmarks (except for the Shifted Rosenbrock function with a global optimum of 390). In our experiments, the proposed fitness-based method generally exhibited same or less search effort on most tested functions at different dimensionality. In particular, the proposed method remarkably exhibited less search effort on most tested benchmarks with n = 3 and 10, and fairly less search effort on Griewank and Classical Rastrigin with n =30. On the other hand, the commonly used distance-based method never achieved a better (i.e., fewer) number of function evaluations compared with the proposed fitness-based method (except for the low-dimensional Quadric function at n=3).

The third comparison metric is the algorithm's efficiency, represented as the CPU time needed to reach the solution (in seconds). All experiments were executed on a 2.20 GHz, 64-bit Core-i7 processor with 8 GB RAM. As shown in Figure 1, the CPU time is remarkably improved on the first 6 tested classical benchmarks at different dimensions. The improvement is generally stronger, however, in low-dimensional problems as opposed to high dimensional problems. The time improvement is particularly impressive for the Rastrigin, Quadric and Spherical functions. The average improvement on the first 6 classical benchmarks with low dimensionality (n = 3) is about 308%, which means PSO with Agile Restart is about 3 times faster than PSO with Distance-based Restart. It is also about 203% faster for n = 10 and 121% faster for n=30, on average.

As for the advanced (shifted/rotated) benchmarks [12; 24], we observed a slight time improvement, but not as substantial as the time improvement for the 6 tested classical benchmarks, as shown in Figure 1. The aggregated average time improvement on all 8 tested benchmarks became 238%, 159%, and 117% for n = 3, 10 and 30, respectively. This considerable CPU time improvement was expected, as discussed earlier, due to the fact that the proposed fitness-based method does not require the time consuming process of calculating pairwise distances between all particles and the global best. Instead, the stagnation is detected using the Average Swarm Fitness criterion, making use of the already calculated fitness values for each particle. Replacing distance-based criterion with the proposed fitness-based criterion can, therefore, be used to increase the computation speed for any Multi-Start PSO algorithm (MPSO) that uses Maximum Swarm Radius, such as RegPSO [7], which showed rather slow performance when recently applied to data clustering [15].

Table 1. Wilcoxon Rank-Sum Significance Test Results

Dimensions	*n = 3*		*n = 10*		*n = 30*	
Significance	P	Z (h)	P	Z (h)	P	Z (h)
Ackley	3.31E-20	9.21 (1)	4.73E-20	9.17 (1)	3.03E-01	1.03 (0)
Griewank	6.64E-04	3.40 (1)	6.72E-02	1.83 (0)	5.14E-09	**-5.84 (1)**
Quadric	1.69E-18	8.78 (1)	5.26E-19	8.91 (1)	7.07E-18	8.61 (1)
Rastrigin	1.85E-13	7.36 (1)	3.31E-20	9.21 (1)	1.33E-03	**-3.21 (1)**
Rosenbrock	2.75E-18	8.72 (1)	2.37E-06	4.72 (1)	7.49E-01	-0.32 (0)
Spherical	2.06E-17	8.49 (1)	3.31E-20	9.21 (1)	7.07E-18	8.61 (1)
Rotated Rastrigin	8.40E-01	0.20 (0)	5.06E-01	-0.67 (0)	1.58E-09	6.04 (1)
Shifted Rosenbrock	1.05E-06	**-4.88 (1)**	1.74E-15	**-7.96 (1)**	2.16E-01	1.24 (0)

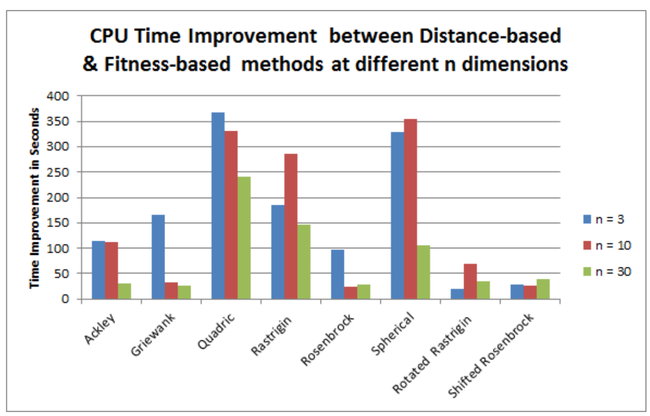

Figure 1. The run-time improvement between the proposed fitness-based (Agile) restart and the popular distance-based restart is up to three times better, on average.

5. CONCLUSIONS

This paper addresses the popular premature convergence issue of PSO by restarting particle positions and resetting their velocities and memories when an imminent stagnation situation is detected, using a novel stagnation detection method. The proposed stagnation detection method uses a more efficient fitness-based criterion to effectively trigger what we called the "Agile Restart" mechanism, as needed, by collectively incorporating the fitness performances of the swarm relative to the objective function, and comparing the Average Swarm Fitness with the global best fitness. The proposed fitness-based criterion has a computational advantage over the distance-based criterion of O(nk) for a swarm of n particles in k number of iterations. This is because it makes use of the already calculated fitness values of each particle to trigger the restart mechanism without the overhead of calculating all pairwise distances between particles and the global best at each iteration.

Interestingly, the efficiency of the proposed fitness-based criterion did not come at the cost of the solution quality. The proposed stagnation detection method demonstrated superior solution quality (compared with the commonly used distance-based method) on most tested optimization benchmarks, verifying not only a significantly better, but also a remarkably more efficient performance. The significance of the performance comparison results over 50 independent runs was validated by the Wilcoxon rank-sum test at 95% confidence level.

The performance comparison between the distance-based method and the fitness-based stagnation detection methods was conducted at low-to-moderate dimensionality that ranges from 3 to 30, on a test set of 8 general optimization benchmarks. It is therefore planned for our future work to carry out further verification simulations on a more comprehensive set of benchmarks at even higher dimensions (i.e., greater than 30).

6. REFERENCES

[1] Ahmed, H.R. and Glasgow, J.I., 2014. An Improved Multi-Start Particle Swarm-based Algorithm for Protein Structure Comparison. In *Proceedings of the 2014 Genetic and Evolutionary Computation Conference (GECCO 2014)* (Vancouver, Canada 2014).

[2] Clerc, M., 1999. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on* IEEE, Washington, DC.

[3] Clerc, M., 2006. Stagnation analysis in particle swarm optimisation or what happens when nothing happens. *HAL : hal-00122031, version 1*.

[4] Clerc, M. and Kennedy, J., 2002. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *Evolutionary Computation, IEEE Transactions on 6*, 1, 58-73.

[5] Derrac, J., García, S., Molina, D., and Herrera, F., 2011. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation 1*, 1, 3-18.

[6] Eberhart, R.C. and Shi, Y., 2000. Comparing inertia weights and constriction factors in particle swarm optimization. In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on* IEEE, La Jolla, CA, 84-88.

[7] Evers, G.I. and Ben Ghalia, M., 2009. Regrouping particle swarm optimization: a new global optimization algorithm with improved performance consistency across benchmarks. In *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on* IEEE, 3901-3908.

[8] Hadley, G., 1964. *Nonlinear and Dynamic Programming*. Addison-Wesley.

[9] Kennedy, J., 1997. The particle swarm: social adaptation of knowledge. In *Evolutionary Computation, 1997., IEEE International Conference on* IEEE, 303-308.

[10] Kennedy, J. and Eberhart, R., 1995. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on* IEEE, 1942-1948.

[11] Li, X., 2004. Adaptively choosing neighbourhood bests using species in a particle swarm optimizer for multimodal function optimization. In *GECCO 2004* Springer, 105-116.

[12] Liang, J., Qu, B., Suganthan, P., and Hernández-Díaz, A.G., 2013. Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization. *Technical Report, Computational Intelligence Laboratory, Zhengzhou University*.

[13] Lu, H. and Chen, X., 2011. A new particle swarm optimization with a dynamic inertia weight for solving constrained optimization problems. *Information Technology Journal 10*, 8, 1536-1544.

[14] Mitchell, Y., Ardi, H., and Hung, N., 2011. Optimization of head movement recognition using augmented radial basis function neural network. In *the IEEE conf. on Engineering in Medicine and Biology Society*, Boston, MA, 2776-2779.

[15] Moulton, B., Yuwono, M., Nguyen, H., and Su, S., 2012. Method for increasing the computation speed of an unsupervised learning approach for data clustering. In *CEC 2012* IEEE, Brisbane, Australia.

[16] Poli, R., 2007. An analysis of publications on particle swarm optimization applications. *Technical Report, Department of Computer Science, University of Essex, UK*.

[17] Van Den Bergh, F., 2002. An analysis of particle swarm optimizers. PhD Thesis, Department of Computer Science, University of Pretoria, South Africa.

[18] Van Den Bergh, F. and Engelbrecht, A., 2002. A new locally convergent particle swarm optimizer. In *Proceedings of the IEEE Conf on systems, man, and cybernetics*, 96-101.

[19] Van Den Bergh, F. and Engelbrecht, A.P., 2010. A convergence proof for the particle swarm optimiser. *Fundamenta Informaticae 105*, 4, 341-374.

[20] Vargas, Y.N. and Chen, S., 2010. Particle swarm optimization with resets–improving the balance between exploration and exploitation. In *Advances in Soft Computing* Springer, 371-381.

[21] Worasucheep, C., 2008. A particle swarm optimization with stagnation detection and dispersion. In *Evolutionary Computation, CEC 2008*. IEEE, 424-429.

[22] Yang, X.S., 2010. Appendix A: Test Problems in Optimization. In *Engineering Optimization* John Wiley & Sons, 261-266.

[23] Yuwono, M., Su, S., and Moulton, B., 2011. Fall detection using a Gaussian distribution of clustered knowledge, augmented radial basis neural-network, and multilayer perceptron. In *Broadband and Biomedical Communications (IB2Com), 6th International Conference on* IEEE, 145-150.

[24] Zhan, Z.-H., Zhang, J., Li, Y., and Shi, Y.-H., 2011. Orthogonal learning particle swarm optimization. *Evolutionary Computation, IEEE Trans on 15*, 6, 832-847.