An Analysis of Sub-swarms in Multi-swarm Systems

Article *in* Lecture Notes in Computer Science · December 2011

DOI: 10.1007/978-3-642-25832-9_28 · Source: DBLP

2 authors:


Antonio Bolufé-Röhler
University of Havana


Stephen Chen
York University

An Analysis of Sub-swarms in Multi-swarm Systems

Antonio Bolufé Röhler[1] and Stephen Chen[2]

[1] University of Havana, Havana, Cuba
bolufe@matcom.uh.cu
[2] School of Information Technology, York University 4700 Keele Street, Toronto, Ontario M3J 1P3
sychen@yorku.ca

Abstract. Particle swarm optimization cannot guarantee convergence to the global optimum on multi-modal functions, so multiple swarms can be useful. One means to coordinate these swarms is to use a separate search mechanism to identify different regions of the solution space for each swarm to explore. The expectation is that these independent sub-swarms can each perform an effective search around the region where it is initialized. This regional focus means that sub-swarms will have different goals and features when compared to standard (single) swarms. A comprehensive study of these differences leads to a new set of general guidelines for the configuration of sub-swarms in multi-swarm systems.

Keywords: Particle swarm optimization, exploration-exploitation, multi-swarm system, multi-modal search spaces.

1 Introduction

Particle Swarm Optimization (PSO) is an effective search technique for optimization problems in continuous domains [10]. Inspired by the principles that influence the flocking of birds and the schooling of fish, the main idea is the combination of personal experience from the individual and social experience from the group. In PSO, this experience is represented as an attraction to the best position found by a given individual (particle) and to the best one found by a set of individuals. Together with a particle's momentum, these attraction forces define the movement of each particle in the swarm.

The movement of particles in a swarm is naturally convergent. The convergence rate can be slower or faster depending on the communication topology (e.g. ring or star)[1], but eventually the swarm will focus its search efforts around the best-found solution(s). The convergent nature of this search process is not ideally suited to multi-modal search spaces where it is important to achieve an effective balance between exploration and exploitation. Compared to modifications which seek to improve the balance between exploration and exploitation in PSO (e.g. [4][7]), an alternative approach is to separate these processes into distinct phases which focus primarily on either exploration or exploitation. Multi-swarms systems (e.g. [2][9]) use multiple sub-swarms to search in different regions of the

D. Wang and M. Reynolds (Eds.): AI 2011, LNAI 7106, pp. 271–280, 2011.

solution space, and this two-phase organization supports an exploration around a diverse number of “best positions”.

A considerable amount of research (e.g. [5]) has been dedicated to study the optimal way to configure particle swarms, and a large amount of this research is summarized in the definition for standard PSO [1]. However, it is not expected that standard (single) swarms and smaller sub-swarms will have the same optimal configuration. For example, sub-swarms that use fewer iterations/function evaluations (FEs) can require more constriction to ensure convergence [3].

In this paper we investigate how different features of PSO – the number of particles, constriction factor, initial velocities, initial positions, and function evaluations – influence the behaviour of sub-swarms in comparison with standard swarms. As a reference point, all experiments in this paper use a novel method based on Estimation Distribution Algorithms (EDA) for selecting the initial positions of the sub-swarms. This initialization simulates the exploratory phase of a hypothetical multi-swarm system. A study of sub-swarms is then carried out with the purpose of providing general considerations and good design features for multi-swarm systems.

This analysis of sub-swarm behaviour begins in Section 2 with some background on different examples of multi-swarm systems. A brief description of the benchmark functions and the experimental design is given in Section 3. In Section 4, the influence of initial velocities on standard swarms in comparison to multi-swarm systems is analyzed. Sections 5 and 6 present some considerations about the constriction factor and the population size, respectively. In Section 7, results are combined into an overall recommendation for sub-swarm parameters. The discussion in Section 8 puts previous results in context, and a brief summary is presented in Section 9.

2 Multi-swarm Systems

The design of multi-swarm systems divides the processes of exploration and exploitation into two distinct phases. Each individual swarm focuses on exploitation in a specific region, and a separate mechanism which chooses these regions focuses on exploration. This exploratory mechanism can be considered as the essential part that differentiates one multi-swarm system from another.

For example, Waves of Swarm Particles (WoSP) [9] bases its diversification mechanism on the “collision” of particles. When particles get too close, a repulsive force expels the particles into new waves/sub-swarms, and this avoids a complete convergence. A key feature of the new sub-swarms is that their initial positions are not randomly selected as in normal swarms. Instead, they maintain some information from the previous trajectories of the particles. A similar relationship exists with initial velocities. In WoSP, the initial search direction after the ejection is based on the previous velocity of the particle.

The significance of initial positions and velocities is much clearer in locust swarms [2]. This multi-swarm system bases its diversification mechanism on a “devour and move on” strategy. Once a sub-swarm has devoured a region (intensive search) the swarm is ready to move on to another promising region. The initial positions of the new sub-swarm are selected using a scouting process around the best position found by the previous sub-swarm. The initial velocities are directed away from this previous optimum to further push the subsequent sub-swarms away from previously devoured parts of the search space.

Although the design of multi-swarm systems tends to focus on the selection of initial positions and initial velocities for the particles of a new sub-swarm, additional design considerations are also required. For example, a multi-swarm system that uses the same overall number of FEs as a standard swarm will require each sub-swarm to use a highly reduced number of iterations/function evaluations. With fewer iterations per particle, and considering each sub-swarm as an exploitation mechanism, it may be necessary to increase the convergence rate by decreasing the constriction factor. To increase the iterations for each particle (given a fixed number of total function evaluations), the swarm size can be reduced. These previously under-studied aspects of sub-swarm design are the focus of this paper.

3 Experimental Design

The experiments presented in this paper have been performed using set 3 (uni-modal functions with high conditioning – functions 10-14) and set 4 (multi-modal functions with adequate global structure – functions 15-19) of the Black-Box Optimization Benchmarking (BBOB) functions [8]. To provide some consistency with other results (e.g. [3][4]) five trials were run on the first five instances of each benchmark function for a total of 25 trials per function. This previous work also used a fixed number of function evaluations based on the dimensions D (i.e. $FEs = 5000 * D$), and they focused on a problem size of $D = 20$ dimensions.

The following experiments require a set of initial positions which are of high quality, but that are not completely converged. These initial positions can then be used to simulate the result of the exploratory phase in a multi-swarm system. Estimation Distribution Algorithms (EDAs) [11] are a promising candidate for selecting the initial positions. Among EDAs, the UMDA algorithm was chosen because it is a simple and methodical way to explore a search space.

The exploratory phase in a multi-swarm system doesn't use 100% of the available function evaluations. For each sub-swarm, a small number of FEs are used for exploration (to find initial positions), and the sub-swarm then tries to find the best possible solution from there (e.g. a nearby local optimum). The following analysis of sub-swarm behaviour uses initial positions selected by an UMDA algorithm after 20,000 FEs (UMDA 20).

The benchmark PSO for the current experiments is a constricted, ring topology version (i.e. standard PSO [1]) developed from the source code published in El-Abd and Kamel [6]. The published implementation uses a swarm size of $p = 40$ particles and a constriction factor $\chi = 0.792$. Together with random initial velocities and UMDA 20 initial positions (i.e. 20,000 FEs for a standard implementation as proposed in [11] with a population of 1,000 individuals and a selection coefficient of 0.2), these will be the default parameters for the experiments in this paper.

4 Effects of Initial Velocities

The effectiveness of a selection method for initial velocities based on Differential Evolution (DE) [12] has been reported by Chen and Montgomery in [3]. In Differential Evolution, a new solution is created by applying a difference vector to a base solution. This update equation (1) uses three unique solutions x_1, x_2 and x_3 drawn from the population, and a scaling factor F.

$$x = x_1 + F(x_2 - x_3) \tag{1}$$

To determine if this technique (DE-velocities) can improve any PSO algorithm regardless of the number of function evaluations, the following experiment compares the performance of swarms with random initial velocities and swarms with initial velocities selected with difference vectors (using F = 1.0). The initial positions are the p best UMDA 20 solutions, and the first results are for a high number of FEs (100,000, 80,000, 60,000, and 40,000). All other "sub-swarm" parameters are from the benchmark [10] (e.g. $\chi = 0.792$ and $p = 40$).

The relative improvement (%-diff) achieved by DE-velocities versus random ones is reported in Table 1. The last row shows the mean improvement over all 10 functions for a given amount of FEs. In these results, the DE-velocities do not show any meaningful improvement compared to the random velocities. A possible explanation is that over the course of a long run, the swarm will conduct a thorough exploration of the search space regardless of the initial velocities.

Table 1. Comparison of DE Velocities vs Random Velocities

fn	100,000 FEs	80,000 FEs	60,000 FEs	40,000 FEs
10	2.1%	−7.8%	5.6%	−7.9%
11	−1.3%	−0.2%	−9.7%	−1.4%
12	0.4%	−6.6%	−5.3%	2.3%
13	−3.0%	−3.1%	2.9%	3.8%
14	1.3%	−1.8%	5.4%	17.6%
15	19.5%	−5.0%	−11.1%	6.0%
16	0.6%	−1.4%	−4.6%	9.1%
17	2.7%	9.0%	15.7%	8.3%
18	3.8%	−13.0%	−4.9%	−0.4%
19	3.2%	4.8%	−3.1%	0.8%
mean	2.9%	−2.5%	−0.9%	3.8%

In multi-swarm systems, the number of FEs is drastically reduced for each sub-swarm. To determine how this decrement influences the effects of initial velocities, the previous experiment is repeated with much fewer function evaluations. In the results shown in Table 2, each column corresponds to swarms with

10,000, 5,000, 2,000, and 1,000 FEs, i.e. 10, 5, 2, and 1 percent respectively of the total FEs of a standard swarm.

The relative improvement obtained by DE-velocities versus random velocities shows that the selection of the initial velocities can be beneficial for swarms which must converge in a small number of function evaluations. The total improvement presented in the last row demonstrates a clear trend – the importance of selected initial velocities increases as the number of FEs used by the sub-swarms decreases. These results replicate the benefit shown in [3] and demonstrate that initial velocities are an important design consideration in sub-swarms that does not exist for standard swarms.

Table 2. Comparison of DE Velocities vs Random Velocities

fn	10,000 FEs	5,000 FEs	2,000 FEs	1,000 FEs
10	7.6%	9.0%	32.2%	54.7%
11	−6.5%	−1.2%	9.9%	−7.1%
12	63.6%	63.8%	92.6%	99.2%
13	29.3%	28.9%	48.9%	79.7%
14	19.0%	35.0%	67.6%	86.8%
15	1.0%	5.5%	16.7%	18.7%
16	4.7%	7.4%	−7.3%	−2.9%
17	7.2%	15.4%	42.7%	69.9%
18	3.2%	4.3%	34.5%	55.1%
19	−0.9%	4.0%	3.1%	9.0%
mean	12.8%	17.2%	34.1%	46.3%

5 Effects of Smaller Constriction Factors

With fewer iterations per particle, it may be beneficial to increase the convergence rate of the sub-swarms (i.e. decrease the constriction factor χ). In standard PSO [1] the velocities of each particle are updated by

$$v_d = \chi(v_d + c_1\epsilon_1(pbest_d - x_d) + c_2\epsilon_2(gbest_d - x_d)) \tag{2}$$

In (2), v is the particle's velocity, x is the position of the particle, and d is a given dimension. The variables ϵ_1 and ϵ_2 are random values, which together with the weights c_1 and c_2 determine the contribution of attractions to the personal and global bests $pbest_d$ and $gbest_d$, respectively. The constriction factor is represented by χ, the specific value used for the constriction factor in [6] is $\chi = 0.792$. By changing the value of this parameter, it is possible to modify the particle's momentum, and therefore to either promote a more exploratory or a more exploitative behaviour.

The following experiments examine the effect of reducing the constriction factor on the performance of sub-swarms. The reported results (see Figure 1) are the relative improvement (%-diff) achieved with a reduced constriction factor versus the value used in the benchmark PSO (i.e. $\chi = 0.792$). The constriction χ was decreased by multiplying it by an additional reduction factor α ($\alpha \leq 1$). The initial positions were selected using UMDA 20, and all other sub-swarm parameters are from the benchmark (e.g. random initial velocities and a swarm size of $p = 40$).

Figure 1 shows the relationship between the improvement in performance and the reduction of the constriction factor for sub-swarms with different amounts of FEs (10,000, 5,000, 2,000 and 1,000). For example, the largest improvement of 67.4% is achieved for sub-swarms with 1,000 FEs by multiplying the original constriction factor with $\alpha = 0.55$ (i.e. $\chi = 0.401$) – see the highlighted tick-mark. The %-diff values are averages for all of the benchmark functions in BBOB sets 3 and 4 (e.g. the mean value in Tables 1 and 2).

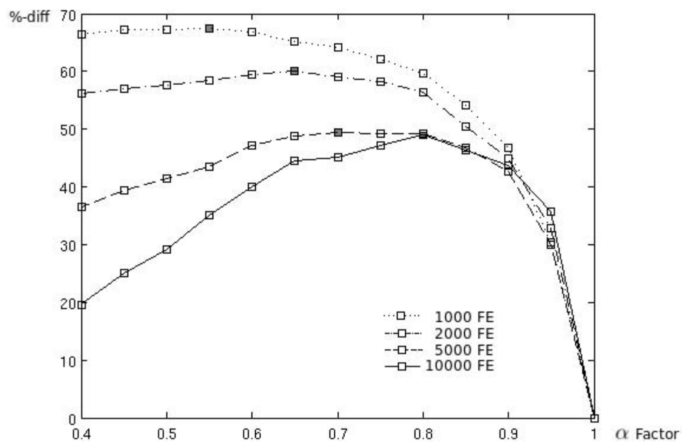

Fig. 1. Relationship between constriction factor and sub-swarm improvement

By analyzing Figure 1, two observations can be made. First, as the number of FEs decreases the (relative) improvement achieved by reducing the constriction factor increases. Second, the best value for the constriction factor gets smaller as the amount of function evaluations is reduced. A smaller constriction factor decreases the particles' momentum which gives the swarm a less exploratory (and more exploitative) behaviour. More exploitation allows sub-swarms to converge, but constriction values that are too small can cause premature convergence which can again decrease the performance of the sub-swarm.

6 Effects of Swarm Size

With a fixed quantity of function evaluations, the number of iterations can be altered by adjusting the swarm size: $FEs = popsize * iterations$. If the swarm size is decreased then it is possible to execute more iterations. On the other hand, a small population may affect the ability of the swarm to explore different regions of the solution space. Thus, the number of individuals causes a trade-off between exploration and exploitation.

In the recent definition for standard PSO [1], Bratton and Kennedy analyzed the influence that the number of particles can have on PSO performance. They report that "no swarm size between 20 – 100 particles produced results that were clearly superior or inferior to any other value for a majority of the tested problems". The best value for a swarm size may depend on problem-specific features like the number of dimensions, constraints, and other characteristics of the objective function. To provide some consistency with other results (e.g. [6]), a swarm size of $p = 40$ particles has been used in the previous experiments.

The purpose of the following set of experiments is to observe the effects of population size in sub-swarms which use a highly limited number of FEs. The reported results (see Figure 2) show the relative improvement (mean %-diff over the 10 functions in BBOB sets 3 and 4) achieved with smaller swarm sizes versus a swarm with $p = 40$ particles. The results in Figure 2 correspond to sub-swarms with different amounts of FEs (10,000, 5,000, 2,000 and 1,000). The initial positions were selected using UMDA 20, and all other sub-swarm parameters are from the benchmark (e.g. random initial velocities and a constriction factor of $\chi = 0.792$).

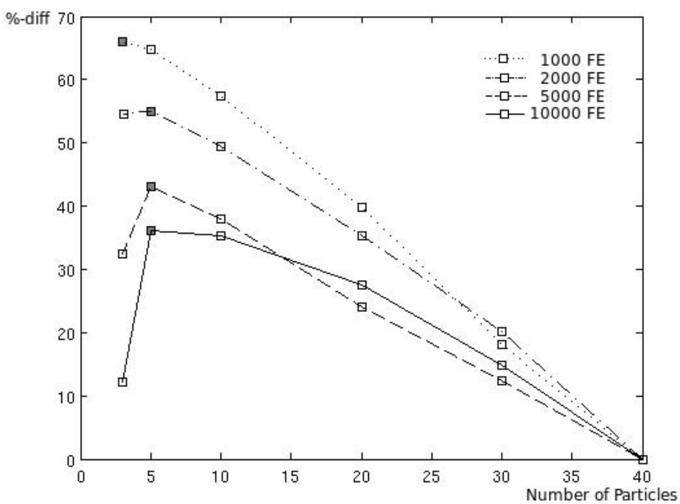

Fig. 2. Relation between number of particles and sub-swarm improvement

In Figure 2, a considerable improvement in performance can be observed when the swarm size is decreased to values far below those suggested by Bratton and Kennedy for a standard swarm [1]. With fewer function evaluations, sub-swarms benefit from smaller populations which allow more iterations and a subsequent increase in their ability to adapt to the function's landscape.

7 Recommended Parameters

So far, the different parameters have been analysed separately with the aim of better understanding the effects that each of them has on sub-swarm behaviour. In this section, the best found combination of parameters for sub-swarms is reported. The focus is on swarm parameters given a set of initial positions (i.e. UMDA 20). Similar to the experiments in Sections 4–6, all of the swarms start with the same initial positions.

The selection of initial velocities is a binary decision, whether to use random or non-random velocities, and the results reported in Section 4 support the use of (non-random) DE-based initial velocities. The selection of the two other parameters, i.e. constriction factor and sub-swarm size, depends on different characteristics of the sub-swarm. Extensive tests (partially shown in Sections 4–6) have led to suggested values of $\alpha = 0.8$ ($\chi = 0.634$) and a swarm size of $p = 15$ particles. In Table 3, the total improvement achieved with these parameters is presented. The results represent the relative improvement (%-diff) achieved with the recommended parameters for sub-swarms versus standard parameters (i.e. random initial velocities, a constriction factor of $\chi = 0.792$, and $p = 40$ particles).

Table 3. Improvement of well parametrized sub-swarms vs. standard parameters

fn	10,000 FEs	5,000 FEs	2,000 FEs	1,000 FEs
10	62.4%	61.3%	66.4%	78.0%
11	23.8%	22.7%	99.9%	99.9%
12	100%	100%	100%	100%
13	88.9%	92.3%	100%	100%
14	92.4%	94.2%	100%	100%
15	36.8%	26.8%	99.9%	100%
16	−0.68%	−1.82%	100%	100%
17	55.9%	67.6%	100%	100%
18	47.8%	50.6%	100%	100%
19	0.26%	25.8%	100%	100%
mean	50.7%	53.9%	96.6%	97.8%

The first two columns in Table 4 show the difference between the initial positions (UMDA 20) and the final results that can be achieved by UMDA in 100,000 FEs (i.e. UMDA 100). This difference represents an initial target for the performance of a multi-swarm system. Starting from the UMDA 20 positions, a sub-swarm using the recommended parameters (i.e. column 1 from Table 3) can achieve results comparable with those from UMDA 100. The UMDA 20 + PSO 10 system uses a total of 30,000 FEs, and it is already more effective than UMDA on 6 of the 10 functions. On the remaining 4 functions, a large amount of the gap between UMDA 20 and UMDA 100 has been covered. Future work will attempt to use the remaining 70,000 FEs to build a multi-swarm system that is more effective than either UMDA or PSO alone.

Table 4. UMDA vs. PSO Sub-swarm

fn	UMDA 100	UMDA 20	UMDA 20+PSO 10
10	$1.68e+04$	$4.24e+04$	$\mathbf{8.81e+03}$
11	$7.80e+01$	$9.03e+01$	$\mathbf{5.86e+01}$
12	$7.47e-01$	$9.61e+03$	$\mathbf{5.70e-01}$
13	$6.44e+00$	$4.84e+01$	$\mathbf{2.19e+00}$
14	$2.63e-03$	$9.77e-02$	$\mathbf{1.39e-03}$
15	$3.06e+00$	$1.03e+02$	$5.39e+01$
16	$1.42e+01$	$2.02e+01$	$\mathbf{1.41e+01}$
17	$2.38e-03$	$3.24e-01$	$7.65e-02$
18	$1.53e-01$	$2.33e+00$	$8.08e-01$
19	$2.84e+00$	$4.26e+00$	$3.56e+00$

8 Discussion

Multi-swarm systems do not base their search process on standard swarms, but on sub-swarms which have a more regional search focus. Two main issues differentiate sub-swarms from standard (single) swarms: the considerable difference in FEs and their non-random initial positions (previously selected by a separate search mechanism that guides the multi-swarm system). Both conditions are reflected in the tests performed in this paper – the later is recreated through the initialization of sub-swarms at the best solutions provided by a relatively short UMDA search.

These differences between standard swarms and sub-swarms imply that different design decisions and parameter values are necessary in multi-swarm systems. In particular, the use of non-random initial velocities leads to large improvements in sub-swarm performance, but they provide no benefits in standard swarms. The optimal swarm size also changes in sub-swarms with the reported results showing that sub-swarms benefit from smaller populations. When the overall number of function evaluations is greatly reduced, fewer particles lead to more iterations, and this allows the sub-swarm to better adapt to the function's landscape.

Sub-swarms usually start in good positions of a specific sub-region, so there is less need to boost exploration as in standard swarms. Subsequently, sub-swarms also benefit from a reduced constriction factor that promotes a more exploitative behaviour. However, it should be noted that the recommended constriction factor has a direct relation with the initial magnitude of the particle velocities, and thus the method used to select the initial velocities [3].

9 Summary

Standard PSO recommends a set of parameters and design decisions such as random initial velocities, a constriction factor of $\chi = 0.729$, and swarms with 20 – 100 particles. These values lead to optimal performance in standard (single) swarms. Sub-swarms have been shown to perform better with features like (non-random) DE-based initial velocities, a constriction factor of $\chi = 0.634$, and $p = 15$ particles. Future work will use these new design recommendations in the development of a multi-swarm system that uses UMDA during the exploratory phase.

References

1. Bratton, D., Kennedy, J.: Defining a Standard for Particle Swarm Optimization. In: Proceedings of the 2007 IEEE Swarm Intelligence Symposium, pp. 120–127 (2007)
2. Chen, S.: Locust Swarms – A New Multi–Optima Search Technique. In: Proceedings of the 2009 IEEE Congress on Evolutionary Computation, pp. 1942–1948 (2009)
3. Chen., S., Montgomery, J.: Selection Strategies for Initial Positions and Initial Velocities in Multi–optima Particle Swarms. In: Proceedings of the 2011 Genetic and Evolutionary Computation Conference, pp. 53–60 (2011)
4. Chen., S., Noa Vargas, Y.: Improving the Performance of Particle Swarms through Dimension Reductions – A Case Study with Locust Swarms. In: Proceedings of the 2010 IEEE Congress on Evolutionary Computation, pp. 211–220 (2010)
5. Clerc, M., Kennedy, J.: The particle swarm – explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation 6(1), 58–73 (2002)
6. El–Abd, M., Kamel, M.S.: Black–Box Optimization Benchmarking for Noiseless Function Testbed using Particle Swarm Optimization. In: Proceedings of the 2009 Genetic and Evolutionary Computation Conference, pp. 2269–2273 (2009)
7. Engelbrecht, A.P.: Heterogeneous particle swarm optimization. In: Proceedings of 7th International Conference on Swarm Intelligence, pp. 191–202 (2010)
8. Hansen, N., Fick, S., Ros, R., Auger, A.: Real–Parameter Black–Box Optimization Benchmarking 2009: Noiseless Functions Definitions. INRIA Technical Report, RR–6829 (2009)
9. Hendtlass, T.: WoSP: A Multi–Optima Particle Swarm Algorithm. In: Proceedings of the 2005 IEEE Congress on Evolutionary Computation, pp. 727–734 (2005)
10. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: Proceedings of the 1995 IEEE International Conference on Neural Networks, pp. 1942–1948 (1995)
11. Larrañaga, P., Lozano, J.A.: Estimation of Distribution Algorithms, A New Tool for Evolutionary Computation. Kluwer Academic Publishers (2011)
12. Storn, R., Price, K.: Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization 11, 341–359 (1995)