

Université Libre de Bruxelles

*Institut de Recherches Interdisciplinaires
et de Développements en Intelligence Artificielle*

Heterogeneous Particle Swarm Optimizers

Marco A. MONTES DE OCA, Jorge PEÑA, Thomas
STÜTZLE, Carlo PINCIROLI, and Marco DORIGO

IRIDIA – Technical Report Series

Technical Report No.
TR/IRIDIA/2009-001

January 2009

IRIDIA – Technical Report Series
ISSN 1781-3794

Published by:

IRIDIA, *Institut de Recherches Interdisciplinaires
et de Développements en Intelligence Artificielle*
UNIVERSITÉ LIBRE DE BRUXELLES
Av F. D. Roosevelt 50, CP 194/6
1050 Bruxelles, Belgium

Technical report number TR/IRIDIA/2009-001

Revision history:

TR/IRIDIA/2009-001.001 January 2009

The information provided is the sole responsibility of the authors and does not necessarily reflect the opinion of the members of IRIDIA. The authors take full responsibility for any copyright breaches that may result from publication of this paper in the IRIDIA – Technical Report Series. IRIDIA is not responsible for any use that might be made of data appearing in this publication.

Heterogeneous Particle Swarm Optimizers

Marco A. MONTES DE OCA^a

mmontes@ulb.ac.be

Jorge PEÑA^b

jorge.pena@unil.ch

Thomas STÜTZLE^a

stuetzle@ulb.ac.be

Carlo PINCIROLI^a

cpinciro@ulb.ac.be

Marco DORIGO^a

mdorigo@ulb.ac.be

^aIRIDIA, CoDE, Université Libre de Bruxelles, Brussels, Belgium

^bIMA, Université de Lausanne, Lausanne, Switzerland

January 2009

Abstract

Particle swarm optimization (PSO) is a swarm intelligence technique originally inspired by models of flocking and of social influence that assumed homogeneous individuals. During its evolution to become a practical optimization tool, some heterogeneous variants have been proposed. However, heterogeneity in PSO algorithms has never been explicitly studied and some of its potential effects have therefore been overlooked. In this paper, we identify some of the most relevant types of heterogeneity that can be ascribed to particle swarms. A number of particle swarms are classified according to the type of heterogeneity they exhibit, which allows us to identify some gaps in current knowledge about heterogeneity in PSO algorithms. Motivated by these observations, we carry out an experimental study of a couple of heterogeneous particle swarms each of which is composed of two kinds of particles. Directions for future developments on heterogeneous particle swarms are outlined.

1 Introduction

Artificial swarm intelligence systems are very often designed using elements inspired by (or directly taken from) models of natural systems composed of numerous entities that collectively exhibit complex behaviors [6, 12]. Examples of this are ant colony optimization [13] and particle swarm optimization (PSO) [20]. While the development of the former was influenced by models of ant foraging [9], the latter was inspired by behavioral models of bird flocking [37] and by models of social influence and culture dissemination [2, 18]. In the vast majority of these models, it is assumed that the group is composed of homogenous individuals.

Models that consider populations of homogeneous individuals are attractive because of their conceptual simplicity. However, heterogeneity is ubiquitous in nature [14]. It ranges from intra-species behavioral variations caused by morphologic or age differences, as in the remarkable case of social insects [10],

to examples of inter-species cooperation, such as symbiosis. These phenomena are more easily and accurately modeled with populations of heterogeneous individuals (cf. [15]). Heterogeneous systems have started to draw the attention of researchers working in different areas of swarm intelligence because designing task-specific agents is often easier than designing versatile, multipotent ones [27, 24].

In this paper, we are concerned with particle heterogeneity in PSO algorithms. Although some existing algorithms display some kind of heterogeneity, this design feature has never been explicitly dealt with. After briefly presenting some basic PSO algorithms in Section 2, we present our first contribution in Section 3, which is a taxonomy of heterogeneous PSO algorithms based on the mechanisms and heuristics that particles use to explore a problem’s search space. This taxonomy is based on the differences between particles at the level of their neighborhood size, their model of influence, and their update rule and its parameterization. A second contribution, also presented in Section 3, is the identification of some gaps in the current knowledge about the behavior and performance of heterogeneous PSO algorithms. This is done through the classification of several PSO variants according to the defined taxonomy. This analysis revealed that the study of heterogeneous PSO algorithms in which particles use different models of influence has not been addressed in the literature yet. Our third contribution, presented in Section 4, is an empirical study of two heterogeneous PSO algorithms composed of two different kinds of particles. This study is intended both to fill in the gap on model-of-influence heterogeneity and to improve our understanding of heterogeneous PSO algorithms in general. The results obtained show that the performance of a heterogeneous PSO algorithm of the kind studied here depends on the relative proportions of particles of different kinds in the swarm. Compared to swarms of homogeneous particles, a heterogeneous PSO algorithm typically performs better than the worst homogeneous swarm and, in some cases, it can have a better performance than the best homogeneous swarm. These results may serve as a baseline for future developments related to heterogeneity in particle swarm optimization, as discussed in Section 5.

2 Particle Swarm Optimization

PSO is a population-based stochastic optimization technique in which individuals (called *particles*) move in the solution space of an n -dimensional objective function f . There are three vectors associated to a particle i : its position vector \mathbf{x}_i , which represents a candidate solution, its velocity vector \mathbf{v}_i , representing the particle’s search direction, and its *personal best* vector \mathbf{p}_i , which denotes the particle’s best-so-far position in the solution space. The movement of a particle is socially mediated, that is, it depends on the search history of a set of particles called *informers* which are selected from its neighbors. Good points in the search space are communicated and used by other particles to guide their search. Thus, contrary to what happens in evolutionary algorithms, individuals in a particle swarm cooperate with each other in order to find better solutions in the fitness landscape.

The behavior of a particle swarm depends on at least three factors: (i) the *population topology*, which defines the neighborhood relations among particles,

(ii) the *model of influence*, which defines the mechanism to select, from each particle’s neighbors, the set of particles that act as informers, and (iii) the *update rule*, used to compute the next position of the particle using information from its informers. In the standard PSO algorithm [7], for example, the above-mentioned factors are instantiated as follows: (i) fully-connected graphs or rings (respectively known as *gbest* and *lbest* models in the PSO parlance) as population topologies, (ii) a *best-of-neighborhood* model of influence such that only the best particle in the neighborhood and the particle itself are taken as informers, and (iii) an update rule for the j th component of the i th particle’s position vector given by

$$v_{i,j}^{t+1} = wv_{i,j}^t + \phi_1\epsilon_1 \left(p_{g(i),j}^t - x_{i,j}^t \right) + \phi_2\epsilon_2 \left(p_{i,j}^t - x_{i,j}^t \right), \quad (1)$$

and

$$x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1}. \quad (2)$$

In Eq. 1, w is a parameter called *inertia weight*, ϕ_k are parameters called *acceleration coefficients*, ϵ_k are independent uniformly distributed pseudorandom numbers in the range $[0, 1)$, and $g(i)$ is particle i ’s best neighbor’s index.

Different settings for the population topology, the model of influence, or the update rule give rise to different PSO algorithms. As population topologies, two-dimensional lattices, small-world networks or random graphs are among the possible choices for replacing the standard fully-connected or ring graphs [19, 21]. Likewise, an alternative to the *best-of-neighborhood* model of influence can be implemented. The most salient example is Mendes’ *fully-informed* model [29], in which a particle is informed by all of its neighbors. The fully-informed model can be implemented as follows:

$$v_{i,j}^{t+1} = wv_{i,j}^t + \sum_{k=1}^{K_i} \phi_k \epsilon_k \left(p_{l(i,k),j}^t - x_{i,j}^t \right), \quad (3)$$

where K_i is particle i ’s total number of informers and $l(i, k)$ is the index of the k th informer of the i th particle. Other models of influence, such as choosing as informers some neighbors at random [17] or with a probability proportional to their “attractiveness” (defined in different ways) have also been proposed (cf. [29]).

Lastly, different update rules, which encode the mechanism for combining information from a particle’s informers, can be devised. For example, bare bones PSO algorithms replace the traditional update rules based on velocities by a mechanism whereby a particle’s next position is sampled from a probability distribution built on the informers’ personal bests. Kennedy’s Gaussian bare bones PSO [17] uses:

$$x_{i,j}^{t+1} = N \left(\frac{p_{i,j} + p_{g(i),j}}{2}, |p_{i,j} - p_{g(i),j}| \right), \quad (4)$$

where $N(\mu, \sigma)$ represents a number drawn from a normal distribution with mean μ and standard deviation σ .

Particle swarm optimization literature abounds with refinements, adjustments and sometimes completely different approaches to the basic algorithm. However, many of them follow the homogeneous population paradigm. In the next sections, we explore the concept of particle heterogeneity in PSO algorithms and its practical consequences.

3 Heterogeneous Particle Swarms

The standard PSO algorithm [7], whether in its *gbest* or *lbest* version, is a homogeneous swarm. Indeed, all particles have the same number of neighbors, choose informers according to the same model of influence and modify their velocities applying the same update rules. Furthermore, the parameters of the velocity update rule (w , ϕ_k) are the same for all particles.

Even though neighborhood sizes, models of influence, update rules and their parameters are properties of the swarm as a whole in the standard PSO, nothing prevents us from instantiating them at the individual level, thus introducing heterogeneity. We say that a particle swarm is heterogeneous if it has at least a pair of particles that differ in any of the above-mentioned aspects. For convenience, we call a particular instantiation of these elements, at the individual level, a particle's *configuration*. Depending on the nature of the differences between particles' configurations, different kinds of heterogeneity can be identified. If particles change configuration over time we qualify the resulting heterogeneity as *dynamic*, otherwise it is said to be *static*. *Adaptive* particle swarms at the individual level build on dynamic heterogeneity by triggering configuration changes as a response to some event caused by the behavior of the swarm, thus "guiding" the otherwise random dynamism.

Differences along any of the different aspects of a particle's configuration give rise to a taxonomy based on four types of heterogeneity. These differences are described in the following subsections.

3.1 Neighborhood heterogeneity

Neighborhood heterogeneity appears when particles have different neighborhood sizes. This kind of heterogeneity occurs when the population topology is not a regular graph, that is, when nodes may have different degrees.

This type of heterogeneity allows some particles to be potentially more influential in the collective search process than others. Neighborhood heterogeneity can be identified in several PSO algorithms, a selection of which is shown in chronological order in Table 1.

It is possible to quantify the levels of heterogeneity in swarms that belong to this class. Perhaps the simplest measure of neighborhood heterogeneity is the range of the particles' neighborhood size, which can go from 1 to the size of the swarm. A study on the relationship between neighborhood sizes and their effects on a swarm of 20 particles has been carried out by Kennedy and Mendes [21, 28, 22]. However, the size of a swarm has a strong effect on the level of neighborhood heterogeneity. The larger the swarm the larger the range of the particles' neighborhood size can be. A systematic study on the effects of different levels of neighborhood heterogeneity in swarms of various sizes is missing in the literature.

3.2 Model-of-influence heterogeneity

This type of heterogeneity occurs when particles in a swarm use different mechanisms for choosing their informers. An example of a swarm with model-of-influence heterogeneity could be one in which some of the particles are informed by the best particle of their neighborhood, while the others are fully-informed.

Table 1: Particle swarm optimizers with neighborhood heterogeneity

Main feature(s)	Type	Reference(s)
Wheels, small-world and random topologies	Static	Kennedy [19]
Interparticle distance-based neighborhoods	Adaptive	Suganthan [39]
Topologies with different average degree	Static	Kennedy & Mendes [21, 22]
Variable neighborhood size according to performance	Adaptive	Zhang et al. [40]
Directed tree-like topology	Dynamic	Janson & Middendorf [16]
Random neighborhood size	Dynamic	Mohais et al. [32, 31]
Neighborhoods based on Delaunay triangulation	Adaptive	Lane et al. [23]
Random and distance-based neighborhoods	Dynamic	Akat & Gazi [1]

To the best of the authors' knowledge, no particle swarm with model-of-influence heterogeneity has been proposed or studied before in the literature. In the next section, we investigate the behavior and performance of a swarm with this kind of heterogeneity.

3.3 Update-rule heterogeneity

If different particles use different rules for updating their position in the search space, we say that the swarm exhibits update rule heterogeneity. Conceptually, PSO algorithms of this class exhibit one of the most extreme cases of heterogeneity because particles can explore the search space in completely different ways. PSO algorithms that belong to this class have been proposed with the idea of having groups of specialized particles that perform different (but complementary) tasks. For example, one group explores the search space while another performs some form of local search. Some of these algorithms are listed in Table 2. The order of presentation is chronological.

3.4 Parameter heterogeneity

Two conditions must be met for parameter heterogeneity to exist: (i) a group of particles must use the same update rule, and (ii) at least a pair of these particles must differ in their update rules' parameter settings. A selection of works that feature PSO algorithms with parameter heterogeneity is shown in Table 3 in chronological order.

Table 2: Particle swarm optimizers with update rule heterogeneity

Main feature(s)	Type	Reference(s)
Predator and prey particles	Static	Silva et al. [38]
Neutral and charged particles	Static	Blackwell and Bentley [4]
Fitness-distance-ratio and standard particles	Static	Baskar and Suganthan [3]
Neutral and quantum particles	Static	Blackwell and Branke [5]
Extra central particle	Static	Liu et al. [25]
Cooperator and defector particles	Adaptive	Di Chio et al. [11]

Table 3: Particle swarm optimizers with parameter heterogeneity

Main feature(s)	Type	Reference(s)
Interparticle distance-dependent inertia weight	Adaptive	Løvbjerg and Krink [26]
Parameter self-adaptation by ES	Adaptive	Miranda and Fonseca [30]
Adaptive acceleration coefficients	Adaptive	Zhang et al. [40]
Different maximum velocities after restart	Dynamic	Pongchairerks et al. [36]
Spatially extended particles with different properties	Adaptive	Monson and Seppi [33]

It is possible to calculate the level of parameter heterogeneity in a swarm by measuring the distance of the parameter vectors of the particles that compose it. This assumes update rules to show a smooth transition from one behavior to another by changing gradually a set of parameters.

4 Experimental Analysis

We empirically study two heterogeneous PSOs, namely, one with model-of-influence heterogeneity and one with update rule heterogeneity. We intend to explore (i) the intra-swarm interaction among particles with different configurations and (ii) the effects of such interactions on the algorithms' performance. These two experiments are considered in order to determine the extent to which the effects observed with one kind of heterogeneity hold if a different kind of heterogeneity is used.

4.1 Setup

The experimental design examines three main factors:

1. **Problem.** We used ten commonly used benchmark functions¹. In each replication, the functions' optima were shifted at random within the defined search range. In all cases, we used their 100-dimensional versions. All algorithms were run 100 times on each problem for up to 10^6 function evaluations.
2. **Particle configurations.** Two cases were considered: (i) swarms with update rule heterogeneity composed of particles using a classic velocity-based or a Gaussian bare bones update rule (see Section 2) and (ii) swarms with model-of-influence heterogeneity composed of particles using a best-of-neighborhood or a fully-informed strategy to select their informers. In this last case, the update rule is that of the standard PSO. We ran experiments with fully-connected and ring topologies. In our setup neighbourhoods are closed, i.e. particles belong to their own neighborhoods. The velocity-based best-of-neighborhood and fully-informed particle swarms were implemented using Clerc's constriction method [8], so that $w = 0.7298$ and $\sum_k \phi_k = 4.1w$ [35].
3. **Population size.** We used swarms of 10, 10^2 , and 10^3 particles. Each particle is initialized as being of one kind with probability $p \in \{0.2, 0.5, 0.8\}$. For instance, in our experiment with best-of neighborhood and fully-informed particles, 3 swarm compositions result: $p = 0.5$ produces unbiased swarms (50% best-of-neighborhood – 50% fully-informed), while $p = 0.2$ and $p = 0.8$ produce biased swarms (20%–80% and 80%–20%, respectively).

We look at the contribution of each kind of particle to the improvement of the best-so-far solution over time. An improvement occurs when a particle finds a better solution than the best-so-far, regardless of how much better it is. We compute, after t function evaluations, the proportion of improvements due to a specific particle type in the last k improvements. In our experiments, $k = 10 \lceil \log(t) \rceil$ in order to have more stable measurements, especially towards the maximum number of function evaluations. In the case of dual-particle-type swarms, computing this measure for one of the particle types completely describes the contributions of each type because the contribution of one type (c_1) can be computed from the contribution of the other (c_2) using the relation $c_1 = 1 - c_2$.

In the following subsections, we present some of the most important results obtained.

4.2 Different update rules: velocity-based & bare bones swarm

In these experiments, swarms are composed of particles that use different update rules. The neighborhood size and the model of influence (best-of-neighborhood) are the same for all the particles.

Figure 1 shows the contribution of velocity-based particles to the improvement of the best-so-far solution and the development of the solution quality

¹For their mathematical definition and the complete set of results, we refer the interested reader to <http://iridia.ulb.ac.be/supp/IridiaSupp2008-020/>

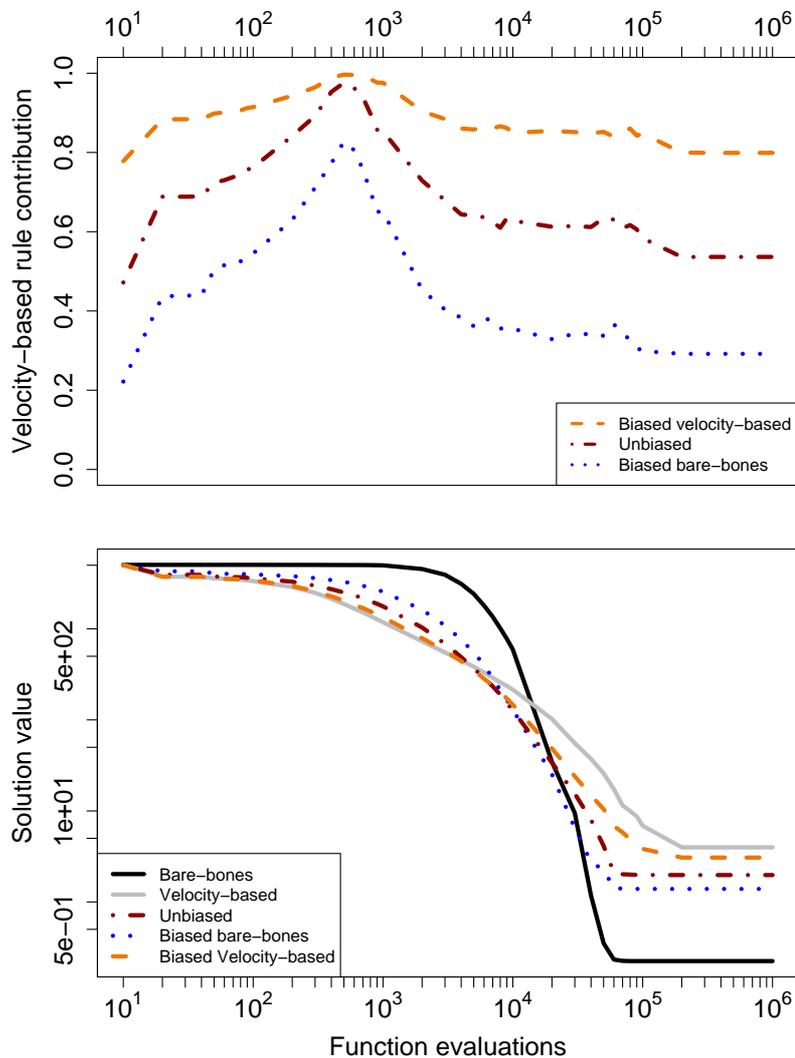


Figure 1: Statistic: Sample mean (100 runs). Problem: Griewank, 100 dimensions. Topology: Fully-connected. Swarm size: 10 particles. The contribution of the particles using the velocity-based update rule to the improvement of the best-so-far solution is shown in the upper part of the figure. The three dotted lines correspond to the three different swarm compositions used in the experiments (20%–80%, 50%–50%, 80%–20%). The development of the solution value over time is shown in the bottom part of the figure.

over time on Griewank’s function. The results were obtained using a 10-particle fully-connected particle swarm.

The proportion of solution improvements due to velocity-based particles grows over time until reaching a maximum after around 500 function evalu-

ations. Afterwards, their contribution starts decreasing until reaching a value close to the initial one. The initial level reflects the distribution of particles in the swarm. In a dual-particle-type swarm with a 20%–80% particle distribution, a contribution level of the first kind of particle equal to 0.2 means that none of the particle types dominates the other. This behavior is observed in biased and unbiased swarms; however, the magnitude of the changes depends on the swarm composition. With 80% bare bones swarms, the growth of the contribution of the velocity-based component is more pronounced than in 50% or 20% bare bones swarms. The lower part of the figure shows the solution quality improvement over time. The results obtained with homogeneous particle swarms of each kind are included for reference. During approximately 15,000 function evaluations, the bare bones particle swarm performs worse than the standard velocity-based PSO algorithm. However, the situation is the opposite afterwards, until the maximum number of functions evaluations is reached. The tested heterogeneous particle swarms show an intermediate performance that is biased according to the swarm’s composition. For example, the performance of the swarm with more velocity-based particles than bare bones particles is closer to that of the particle swarm with velocity-based particles only.

Another example of the behavior of a particle swarm with static update rule heterogeneity is shown in Figure 2. It shows the same data as Figure 1 but on Rastrigin’s function. The results shown were obtained using 10 particles and a fully-connected topology. As before, the contribution of velocity-based particles grows over time up to a maximum that occurs, in this case, around 1,000 function evaluations to later decrease and reach a level similar to the initial one. In terms of solution quality improvement over time, the homogeneous bare bones particle swarm is outperformed by the velocity-based one. The performance of the heterogeneous particle swarms is intermediate between the two homogeneous extremes.

Table 4 shows the ranking of the solution quality after 10^6 function evaluations obtained by all the algorithms tested in this experiment. The results are grouped by problem, population topology and population size. Of the 60 conditions tested, in 13 cases a heterogeneous particle swarm was the best ranked and in 6 cases it was the worst. Heterogeneous swarms have, in most cases, intermediate ranks. From these data and the summary statistics shown at the bottom of the table, it can be seen that the relative ranking of the homogeneous swarms is highly dependent on the population topology to changes in the population size. The rankings of heterogeneous swarms are quite stable to both kind of changes, which suggests that they are more robust.

4.3 Different models of influence: best-of-neighborhood & fully-informed swarm

The experiments presented in this subsection feature swarms composed of particles that use different models of influence. The neighborhood size and the update rules are the same for all the particles.

Figure 3 shows the contribution of particles using the best-of-neighborhood model of influence to the improvement of the best-so-far solution and the development of the solution quality over time on Ackley’s function. These results were obtained using 10 particles and a fully-connected topology.

Table 4: Ranking of the solution quality obtained by swarms with and without update rule heterogeneity after 10^6 function evaluations¹

Problem	Particles	Fully-connected					Ring				
		BB	BBB	U	BV	V	BB	BBB	U	BV	V
Ackley	10^1	5	4	3	2	1	5	4	3	2	1
	10^2	5	4	2	1	3	5	4	3	2	1
	10^3	5	4	2	3	1	5	4	3	2	1
Griewank	10^1	1	2	3	4	5	1	2	4	3	5
	10^2	1	2	3	4	5	5	2	4	3	1
	10^3	5	1	4	2	3	5	4	3	2	1
Rastrigin	10^1	5	4	2	3	1	5	4	3	2	1
	10^2	1	2	3	4	5	3	5	4	2	1
	10^3	3	1	2	4	5	5	4	3	2	1
Rosenbrock	10^1	2	1	5	4	3	5	4	3	2	1
	10^2	5	4	2	3	1	5	4	3	2	1
	10^3	5	4	3	1	2	5	4	3	2	1
Salomon	10^1	1	2	3	4	5	1	2	3	4	5
	10^2	1	2	3	4	5	5	4	3	2	1
	10^3	5	4	3	2	1	5	4	3	2	1
Schaffer	10^1	1	2	3	4	5	1	2	5	4	3
	10^2	2	3	1	4	5	5	4	3	2	1
	10^3	5	4	3	2	1	5	4	3	2	1
Schwefel	10^1	5	4	3	2	1	5	4	3	2	1
	10^2	5	4	3	2	1	5	4	3	2	1
	10^3	5	4	3	2	1	5	4	3	2	1
Sphere	10^1	1	2	3	4	5	4	5	3	1	2
	10^2	5	2	3.5	1	3.5	5	2	1	4	3
	10^3	5	4	3	2	1	5	4	3	2	1
Step	10^1	5	4	3	2	1	5	4	3	2	1
	10^2	5	4	3	2	1	5	4	3	2	1
	10^3	5	4	3	1	2	5	4	3	2	1
Weierstrass	10^1	3	1	2	4	5	4	5	3	2	1
	10^2	2	3	5	4	1	2	1	3	5	4
	10^3	5	4	3	1.5	1.5	5	4	3	2	1
Summary	Median	5	4	3	2.5	2	5	4	3	2	1
	IQR	3	2	0	2	4	0	0	0	0	0

¹ BB, BBB, U, BV, and V stand for bare bones, biased bare bones, unbiased, biased velocity-based, and velocity-based respectively. IQR stands for interquartile range.

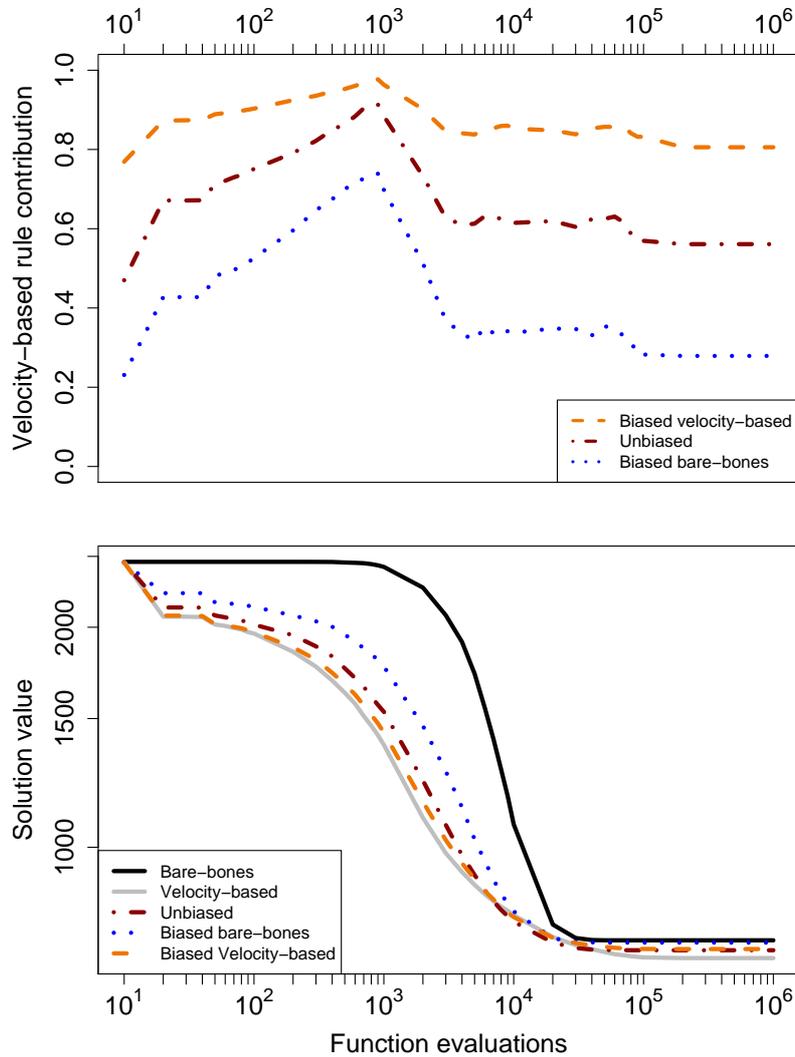


Figure 2: Statistic: Sample mean (100 runs). Problem: Rastrigin, 100 dimensions. Topology: Fully-connected. Swarm size: 10 particles. The contribution of the particles using the velocity-based update rule to the improvement of the best-so-far solution is shown in the upper part of the figure. The three dotted lines correspond to the three different swarm compositions used in the experiments (20%–80%, 50%–50%, 80%–20%). The development of the solution value over time is shown in the bottom part of the figure.

The contribution of the particles using the best-of-neighborhood model of influence starts decreasing, reaches a minimum at around 400 function evaluations and starts increasing again, more aggressively, to reach a maximum at around 3000 function evaluations. It then decreases again but stabilizes at a

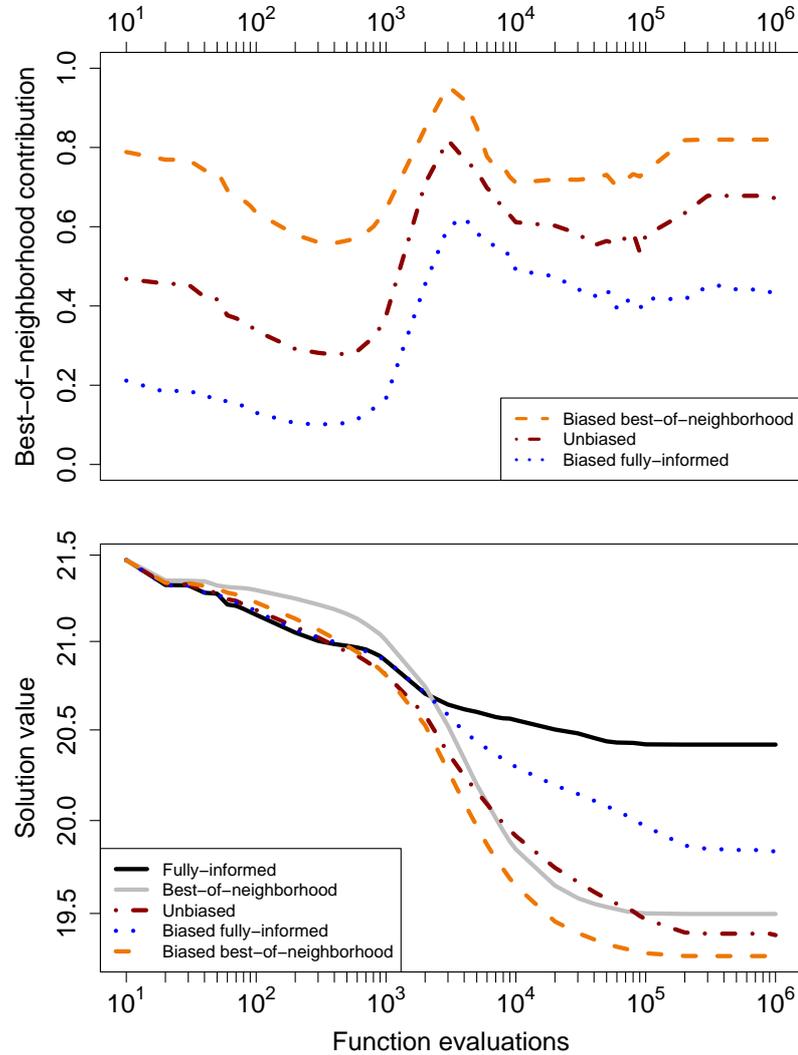


Figure 3: Statistic: Sample mean (100 runs). Problem: Ackley, 100 dimensions. Topology: Fully-connected. Swarm size: 10 particles. The contribution of the particles using the best-of-neighborhood model of influence to the improvement of the best-so-far solution is shown in the upper part of the figure. The three dotted lines correspond to the three different swarm compositions used in the experiments (20%–80%, 50%–50%, 80%–20%). The development of the solution value over time is shown in the bottom part of the figure.

value higher than the initial one. This behavior is richer than the one observed for update rule heterogeneity. In fact, these richer interactions lead, in the case of the swarm biased toward the best-of-neighborhood model, to an overall performance that is better than the performance of the best of the two homogeneous

particle swarms. However, this behavior is not typical for the whole set of functions and dimensionalities. Just as in the previous example, the typical result is to have an intermediate performance between those obtained by purely homogeneous swarms. This can be seen in Figure 4, which shows the contribution of particles using the best-of-neighborhood model of influence to the improvement of the best-so-far solution and the development of the solution quality over time on Salomon’s function. The results shown were obtained using 10 particles and a fully-connected topology. In this case, the contribution of best-of-neighborhood particles decreases over time until it reaches a minimum at around 300 function evaluations and then increases again until stabilization at around 5,000 function evaluations. The solution quality reached by any heterogeneous particle swarm is clearly in between the one reached by the homogeneous swarms, except during the first function evaluations in which the performance of the fully-informed particle swarm is undistinguishable from the one of the heterogeneous swarms.

The ranking of the algorithms’ solution quality after 10^6 function evaluations is shown in Table 5. In this case, in 25 cases a heterogeneous particle swarm was the best ranked and in 4 cases it was the worst. In this experiment, the ranking of the homogeneous swarms is highly dependent on the population topology used. This is already a well-known effect, especially for the fully-informed PSO algorithm [28, 34]. The value of the interquartile range of the distribution of ranks across problems and population sizes reveals that heterogeneous swarms, specially the unbiased ones, are more stable to changes in the population topology and size.

5 Discussion and Outlook

We say that a PSO algorithm is heterogeneous if at least two of its constituent particles differ in their configurations, that is, if they differ in at least one of four factors: (i) their neighborhood sizes, (ii) their models of influence, (iii) their update rules or (iv) their update rules’ parameters. Particles with different configurations will behave differently, but such difference may not be significant from a statistical point of view. In this work we are more interested in identifying the most relevant types of heterogeneity that a PSO algorithm can exhibit than identifying the differences that produce statistically significant behavioral dissimilarities. Future research should be oriented towards this goal.

One of the results of our experimental analysis is that the nature of the differences between particles is not the only factor that determines the performance of a heterogeneous PSO algorithm. In particular, the relative composition of the swarm plays a major role in this respect. Our results show that, independently of the swarm size, the performance of heterogeneous PSO algorithms depends on the proportion of the swarm that is of a particular particle type. As a consequence of this, the typical performance of a heterogeneous PSO algorithm is intermediate with respect to the homogeneous cases. This is not necessarily negative because this means that heterogeneous swarms are more robust than homogeneous swarms. According to our results, a heterogeneous PSO algorithm will usually exhibit a better performance than the worst performing homogeneous PSO algorithm. Using a heterogeneous swarm thus reduces the risk of using a homogeneous swarm of the “wrong” type for the problem at hand.

Another result of the experimental analysis is the identification of some

Table 5: Ranking of the solution quality obtained by swarms with and without model of influence heterogeneity after 10^6 function evaluations¹

Problem	Particles	Fully-connected					Ring				
		F	BF	U	BB	B	F	BF	U	BB	B
Ackley	10^1	5	4	2	1	3	4	2	1	3	5
	10^2	5	4	3	2	1	2	1	3	4	5
	10^3	5	4	3	2	1	1	2	3	4	5
Griewank	10^1	5	4	3	2	1	5	4	3	2	1
	10^2	5	4	3	2	1	5	2.5	4	2.5	1
	10^3	5	4	3	2	1	1	2	3	4	5
Rastrigin	10^1	5	4	3	1	2	2	1	3	4	5
	10^2	5	4	3	1	2	3	1	2	4	5
	10^3	5	1	3	4	2	5	2	1	3	4
Rosenbrock	10^1	5	4	3	2	1	5	4	3	2	1
	10^2	5	4	3	1	2	5	1	3	2	4
	10^3	5	4	2	1	3	1	2	3	4	5
Salomon	10^1	5	4	3	2	1	5	4	3	2	1
	10^2	5	4	3	2	1	5	3	1	2	4
	10^3	5	4	2	3	1	2	1	3	4	5
Schaffer	10^1	5	4	3	1	2	5	4	3	2	1
	10^2	5	4	3	2	1	5	4	3	2	1
	10^3	5	4	3	1	2	5	4	3	2	1
Schwefel	10^1	5	4	3	2	1	2	1	4	5	3
	10^2	5	4	3	2	1	5	4	3	2	1
	10^3	5	4	3	2	1	5	4	3	2	1
Sphere	10^1	5	4	3	2	1	4	3	5	1	2
	10^2	5	4	3	1	2	4	2.5	1	2.5	5
	10^3	5	4	2	1	3	1	2	3	4	5
Step	10^1	5	4	3	2	1	2	1	3	5	4
	10^2	5	4	3	2	1	1	2	3	4	5
	10^3	4	5	3	2	1	1	2	3	4	5
Weierstrass	10^1	5	4	3	1	2	5	4	3	1.5	1.5
	10^2	5	4	2	3	1	1	5	3.5	3.5	2
	10^3	5	1.5	1.5	4	3	1	2	3	4	5
Summary	Median	5	4	3	2	1	4	2	3	3	4
	IQR	0	0	0	1	1	3.75	2	0	2	4

¹ F, BF, U, BB, and B stand for fully-informed, biased fully-informed, unbiased, biased best-of-neighborhood, and best-of-neighborhood respectively. IQR stands for interquartile range.

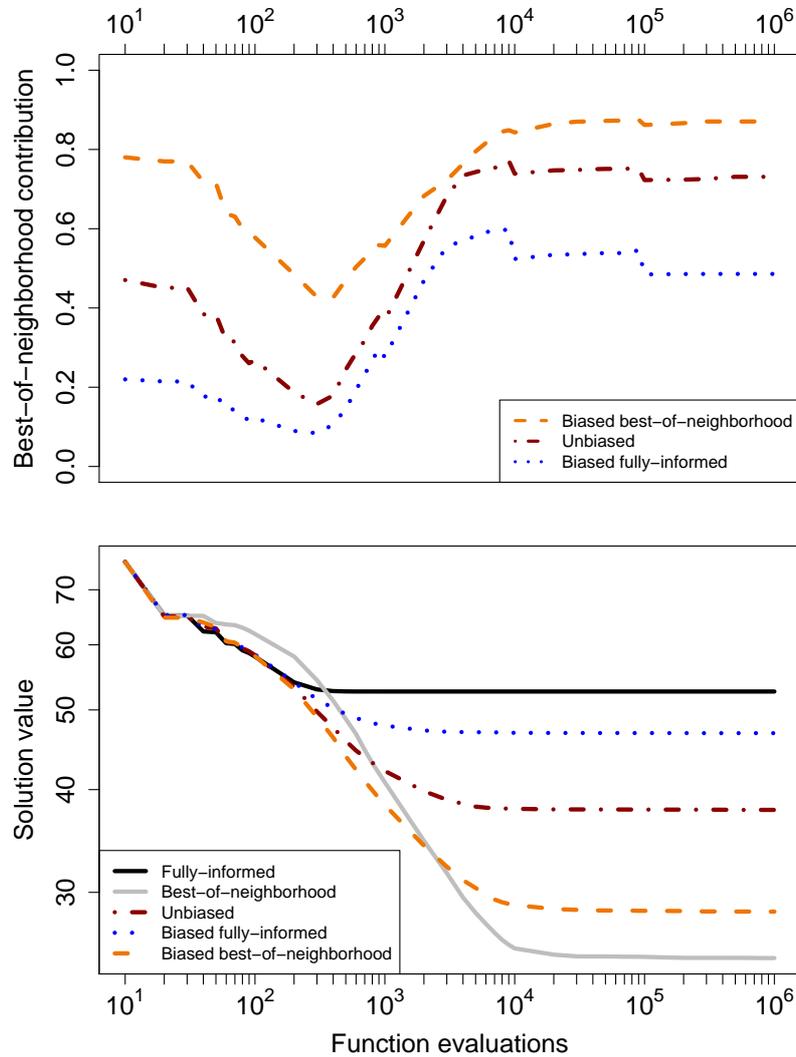


Figure 4: Statistic: Sample mean (100 runs). Problem: Salomon, 100 dimensions. Topology: Fully-connected. Swarm size: 10 particles. The contribution of the particles using the best-of-neighborhood model of influence to the improvement of the best-so-far solution is shown in the upper part of the figure. The three dotted lines correspond to the three different swarm compositions used in the experiments (20%–80%, 50%–50%, 80%–20%). The development of the solution value over time is shown in the bottom part of the figure.

interesting interactions between particles of different kind. The contribution profile of each kind of particle revealed that particles of different kind indeed contribute differently to the improvement of the best-so-far solution. This means that the interaction between particles of different kind is not at all random; on

the contrary, some particles contribute to the solution improvement differently during the optimization process. These results point towards the idea that mechanisms allowing particles to change configuration during the optimization process could be beneficial. Future work is aimed at studying this topic.

From a practical point of view, heterogeneity has a number of consequences. Perhaps the most important is that it seems easier to design PSO algorithms with the right balance of exploration and exploitation by changing the number of exploratory particles, than designing sophisticated update rules to make particles capable of doing both things. This means that heterogeneity enables designers to move design complications from the individual level to the swarm level.

6 Conclusions

The design of artificial swarm intelligence systems consisting of numerous dynamically interacting entities is a difficult task. In an effort to simplify the problem, researchers have assumed that the constituent entities of a system are all identical. While dealing with homogeneous populations can indeed simplify the design task in some cases, it can also make it more difficult in others. The alternative to the homogeneous approach is to consider populations of heterogeneous agents in which some of them can be specialized in some particular task or exhibit a particular behavior.

In this paper, we examined the PSO literature and classified several works according to a taxonomy that considers the most relevant types of heterogeneity that can be found in this kind of algorithms. Our analysis showed that different types of heterogeneity has been used before in PSO. However, heterogeneity at the level of the mechanism used by particles to select the neighbors that effectively influence their movement had remained unexplored until now. Moreover, the study of heterogeneity in PSO has not been done systematically and therefore there are still gaps in our understanding of the effects of heterogeneity in PSO algorithms.

A series of experiments with static dual-particle-type particle swarm optimization algorithms were carried out. The results show that, typically, heterogeneous swarms perform better than the worst homogeneous swarm and that, in some cases, they can outperform the best performing homogeneous swarm.

Acknowledgments. This work was supported by the META-X project, an *Action de Recherche Concertée* funded by the Scientific Research Directorate of the French Community of Belgium. M. A. Montes de Oca acknowledges support from the Programme Alβan, the European Union Programme of High Level Scholarships for Latin America, scholarship No. E05D054889MX. J. Peña acknowledges support from the *PERPLEXUS* project funded by the Future and Emerging Technologies programme (IST-STREP) of the European Commission (grant IST-034632). T. Stützle and M. Dorigo acknowledge support from the F.R.S-FNRS of the French Community of Belgium of which they are a Research Associate and a Research Director, respectively.

References

- [1] S. Burak Akat and Veysel Gazi. Particle swarm optimization with dynamic neighborhood topology: Three neighborhood strategies and preliminary results. In *Proceedings of the 2008 IEEE Swarm Intelligence Symposium*, Piscataway, NJ, 2008. IEEE Press.
- [2] Robert Axelrod. The dissemination of culture: A model with local convergence and global polarization. *Journal of Conflict Resolution*, 41(2):203–226, 1997.
- [3] A. Baskar and Ponnuthurai N. Suganthan. A novel concurrent particle swarm optimization. In *Proceedings of the 2004 IEEE Congress on Evolutionary Computation*, pages 792–796, Piscataway, NJ, 2004. IEEE Press.
- [4] Tim. M. Blackwell and Peter J. Bentley. Dynamic search with charged swarms. In *GECCO '02: Proceedings of the Conference on Genetic and Evolutionary Computation*, pages 19–26, San Francisco, CA, 2002. Morgan Kaufmann.
- [5] Tim M. Blackwell and Jurgen Branke. Multi-swarm optimization in dynamic environments. In *LNCS 3005. Proc. of Applications of Evolutionary Computing: EvoWorkshops 2004: EvoBIO, EvoCOMNET, EvoHOT, EvoISAP, EvoMUSART, and EvoSTOC*, pages 489–500, Berlin, Germany, 2004. Springer.
- [6] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, New York, NY, 1999.
- [7] D. Bratton and J. Kennedy. Defining a standard for particle swarm optimization. In *Proceedings of 2007 IEEE Swarm Intelligence Symposium*, pages 120–127, Piscataway, NJ, 2007. IEEE Press.
- [8] Maurice Clerc and James Kennedy. The particle swarm–explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1):58–73, 2002.
- [9] J.-L. Deneubourg, S. Aron, S. Goss, and J.-M. Pasteels. The self-organizing exploratory pattern of the argentine ant. *Journal of Insect Behavior*, 3(2):159–168, 1990.
- [10] C. Detrain and J.-M. Pasteels. Caste differences in behavioural thresholds as a basis for polyethism during food recruitment in the ant, *pheidole pallidula* (nyl.) (hymenoptera: Myrmicinae). *Insect Behavior*, 4(2):157–176, 1991.
- [11] Cecilia Di Chio, Paolo Di Chio, and Mario Giacobini. An evolutionary game-theoretical approach to particle swarm optimisation. In *LNCS 4974. Proc. of Applications of Evolutionary Computing: EvoWorkshops 2008: EvoCOMNET, EvoFIN, EvoHOT, EvoIASP, EvoMUSART, EvoNUM, EvoSTOC, and EvoTransLog*, pages 575–584, Berlin, 2008. Springer.

- [12] Marco Dorigo and Mauro Birattari. Swarm intelligence. *Scholarpedia*, 2(9):1462, 2007.
- [13] Marco Dorigo and Thomas Stützle. *Ant Colony Optimization*. The MIT Press, Cambridge, MA, 2004.
- [14] Mike Hansell. *Built by Animals*. Oxford University Press, New York, NY, 2007.
- [15] Lu Hong and Scott E. Page. Problem solving by heterogeneous agents. *Journal of Economic Theory*, 97(1):123–163, 2001.
- [16] S. Janson and M. Middendorf. A hierarchical particle swarm optimizer and its adaptive variant. *IEEE Transactions on Systems, Man and Cybernetics B*, 35(6):1272–1282, 2004.
- [17] J. Kennedy. Bare bones particle swarms. In *Proceedings of the 2003 IEEE Swarm Intelligence Symposium*, pages 80–87, Piscataway, NJ, 2003. IEEE Press.
- [18] James Kennedy. Thinking is social: Experiments with the adaptive culture model. *Journal of Conflict Resolution*, 42(1):56–76, 1998.
- [19] James Kennedy. Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance. In *Proceedings of the 1999 IEEE Congress on Evolutionary Computation*, pages 1931–1938, Piscataway, NJ, 1999. IEEE Press.
- [20] James Kennedy, Russell Eberhart, and Yuhui Shi. *Swarm Intelligence*. Morgan Kaufmann, San Francisco, CA, 2001.
- [21] James Kennedy and Rui Mendes. Population structure and particle swarm performance. In *Proceedings of the 2002 IEEE Congress on Evolutionary Computation*, pages 1671–1676, Los Alamitos, CA, 2002. IEEE Computer Society.
- [22] James Kennedy and Rui Mendes. Neighborhood topologies in fully informed and best-of-neighborhood particle swarms. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 36(4):515–519, 2006.
- [23] James Lane, Andries Engelbrecht, and James Gain. Particle swarm optimization with spatially meaningful neighbours. In *Proceedings of the 2008 IEEE Swarm Intelligence Symposium*, Piscataway, NJ, 2008. IEEE Press.
- [24] Ling Li, Alcherio Martinoli, and Yaser S. Abu-Mostafa. Learning and measuring specialization in collaborative swarm systems. *Adaptive Behavior*, 12(3–4):199–212, 2004.
- [25] Yu Liu, Zheng Qin, Zhewen Shi, and Jiang Lu. Center particle swarm optimization. *Neurocomputing*, 70(4–6):672–679, 2007.
- [26] M. Løvbjerg and T. Krink. Extending particle swarm optimisers with self-organized criticality. In *Proceedings of the 2002 IEEE Congress on Evolutionary Computation*, pages 1588–1593, Los Alamitos, CA, 2002. IEEE Computer Society.

- [27] Sven Magg and Renè te Boekhorst. Pattern formation in homogeneous and heterogeneous swarms: Differences between versatile and specialized robots. In *Proceedings of the 2007 IEEE Symposium on Artificial Life*, pages 311–316, Piscataway, NJ, 2007. IEEE Press.
- [28] Rui Mendes. *Population topologies and their influence in particle swarm performance*. PhD thesis, Escola de Engenharia, Universidade do Minho, Portugal, May 2004.
- [29] Rui Mendes, James Kennedy, and Jose Neves. The fully informed particle swarm: Simpler, maybe better. *IEEE Trans. Evolutionary Computation*, 8(3):204–210, 2004.
- [30] V. Miranda and N. Fonseca. EPSO - best-of-two-worlds meta-heuristic applied to power system problems. In *Proceedings of the 2002 IEEE Congress on Evolutionary Computation*, pages 1080–1085, Piscataway, NJ, 2002. IEEE Press.
- [31] Arvind Mohais, Rui Mendes, Christopher Ward, and Christian Posthoff. Neighborhood re-structuring in particle swarm optimization. In *LNCS 3809. AI 2005: Advances in Artificial Intelligence*, pages 776–785, Berlin, Germany, 2005. Springer.
- [32] Arvind Mohais, Christopher Ward, and Christian Posthoff. Randomized directed neighborhoods with edge migration in particle swarm optimization. In *Proceedings of the 2004 IEEE Congress on Evolutionary Computation*, pages 548–555, Piscataway, NJ, 2004. IEEE Press.
- [33] Christopher K. Monson and Kevin D. Seppi. Adaptive diversity in PSO. In *GECCO '06: Proceedings of the Conference on Genetic and Evolutionary Computation*, pages 59–66, New York, NY, 2006. ACM Press.
- [34] Marco A. Montes de Oca and Thomas Stützle. Convergence behavior of the fully informed particle swarm optimization algorithm. In M. Keijzer et al., editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2008)*, pages 71–78. ACM, 2008.
- [35] Riccardo Poli, James Kennedy, and Tim Blackwell. Particle swarm optimization. An overview. *Swarm Intelligence*, 1(1):33–57, 2007.
- [36] Pisut Pongchairerks and Voratas Kachitvichyanakul. A non-homogeneous particle swarm optimization with multiple social structures. In Voratas Kachitvichyanakul et al., editors, *Proceedings of the International Conference on Simulation and Modeling*, pages 137–144, Bangkok, Thailand, 2005. Asian Institute of Technology.
- [37] Craig W. Reynolds. Flocks, herds and schools: A distributed behavioral model. *ACM Computer Graphics*, 21(4):25–34, 1987.
- [38] Arlindo Silva, Ana Neves, and Ernesto Costa. An empirical comparison of particle swarm and predator prey optimisation. In *LNAI 2464. AICS '02: Proceedings of the 13th Irish International Conference on Artificial Intelligence and Cognitive Science*, pages 103–110, Berlin, Germany, 2002. Springer.

- [39] Ponnuthurai N. Suganthan. Particle swarm optimiser with neighbourhood operator. In *Proceedings of the 1999 IEEE Congress on Evolutionary Computation*, pages 1958–1962, Piscataway, NJ, 1999. IEEE Press.
- [40] Wen Zhang, Yutian Liu, and Maurice Clerc. An adaptive PSO algorithm for reactive power optimization. In *Proceedings of the Sixth International Conference on Advances in Power System Control, Operation and Management (APSOM)*, pages 302–307, London, UK, 2003. IEE.