Procedia Computer Science

Volume 80, 2016, Pages 804–813

ICCS 2016. The International Conference on Computational Science

Enhancing Particle Swarm Optimization with Socio-cognitive Inspirations

Iwan Bugajski[1], Piotr Listkiewicz[1], Aleksander Byrski[1], Marek Kisiel-Dorohinicki[1], Wojciech Korczynski[1], Tom Lenaerts[2], Dana Samson[3], Bipin Indurkhya[4], and Ann Nowé[5]

[1] AGH University of Science and Technology, Faculty of Computer Science, Electronics and Telecommunications, Department of Computer Science, Al. Mickiewicza 30, 30-059 Krakow, Poland
{iwan.bugajski,piotr.listkiewicz}@gmail.com,{olekb,doroh,wojciech.korczynski}@agh.edu.pl
[2] Université Libre de Bruxelles Campus de la Plaine ULB CP212, Boulevard du Triomphe 1050 Bruxelles, Belgium tlenaert@ulb.ac.be
[3] Université Catholique de Louvain IPSY Pl. Cardinal Mercier 10 bte. L3.05.01 Louvain-la-Neuve, Belgium dana.samson@uclouvain.be
[4] Institute of Philosophy, Jagiellonian University ul. Grodzka 52 31-044 Krakow, Poland bipin@agh.edu.pl
[5] Vrije Universiteit Brussel Boulevard de la Plaine 2 1050 Ixelles, Belgium anowe@como.vub.ac.be

Abstract

We incorporate socio-cognitively inspired metaheuristics, which we have used successfully in the ACO algorithms in our past research, into the classical particle swarm optimization algorithms. The swarm is divided into species and the particles get inspired not only by the global and local optima, but share their knowledge of the optima with neighboring agents belonging to other species. Our experimental research gathered for common benchmark functions tackled in 100 dimensions show that the metaheuristics are effective and perform better than the classic PSO. We experimented with various proportions of different species in the swarm population to find the best mix of population.

Keywords: Metaheuristic computing, Swarm Intelligence, Nature-inspired computing

1 Introduction

Many algorithmic techniques in computer science are inspired by natural and biological processes. For example, decomposition of the population in evolutionary and similar computing techniques popularized such solutions as island model of evolution [2] that effectively enhanced the diversity of the computing metaheuristics. In the same vein, hybrid algorithms incorporating memetic, immunological, and hierarchical processes have been implemented [9]. In our research project, we are following the same approach by bringing different inspirations together

 Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2016

doi:10.1016/j.procs.2016.05.370

to yield interesting algorithms that are always needed (as stated by the famous No Free Lunch Theorem [11]).

In our earlier research, we experimented with socio-cognitive Ant Colony Optimization (ACO) metaheuristics [6, 8] by introducing into classic ACO a number of sub-species along with positing inter-relations among them, and by leveraging stigmergic relations among these artificial ants. These relations incorporated the ants' abilities to change perspective, to get inspired by the solutions found by other species, and so on. [6]. This approach produced efficient results in discrete optimization (solving different instances of TSP).

Our goal here is to extend the proposed socio-cognitive inspirations by applying it (to the possible extent) to Particle Swarm Optimization (PSO). Our motivation for attempting this extension is as follows. In the case of ACO, our goal was to build a common knowledge base about the solutions found by different species and share them. This was relatively easy to implement because the "derivation" of this knowledge was present in the pheromone table. In the case of PSO, this knowledge about global and local current optima is also quite natural, present and easy accessible in the system. Therefore, we plan to introduce various species that will take different inspirations from the current optima of the other species, and to test the resulting algorithms on the common benchmark functions.

In the next section, we provide the brief background, followed by a description of the socio-cognitive inspirations that are the basis of our approach. Then we introduce the classic PSO, and describe the socio-cognitive modification of this algorithm. Finally, we present the results of our experiments to evaluate the resulting algorithms, and then the conclusions.

2 PSO and its multi-species alternatives

Particle swarm optimization [5] is an iterative algorithm commonly used for mathematical optimization of certain problems. PSO belongs to a set of algorthims called metaheuristics - these algorithms do not guarantee finding the most optimum solution, but can yield a solution close to it. This fact makes PSO suitable for solving problems where either there is no known algorithm or execution of an exact algorithm consumes too much time or resources. Moreover, PSO does not require that the function being optimized be differentiable, regular or constant over time.

Particle swarm optimization was originally proposed for simulating social behavior, and was used for simulating the group movement of fish schools, bird flocks, and so on. But the algorithm was also found to be useful for performing mathematical optimization after some simplification.

In the past, there have been other attempts to decompose the PSO into sub-populations and sub-species, and here we would like to mention a few such notable approaches.

In [10], the authors demonstrated a cooperative PSO: they employed many swarms and made them optimize different parts of the solution vector in cooperation.

In another study[7], the authors modified PSO by introducing different swarms sharing information about their best solutions in order to escape the local extrema. Improved results were obtained in continuous optimization using this approach.

In [4], the author extended the original PSO by dividing the particle swarm spatially into multiple clusters called species in a multi-dimensional search space. Each species explored a different area of the search space to find the global or local optima along that dimension.

In [12], the number of particles in the sub-swarms were dynamically adjusted, yielding a competitive performance with respect to the examined benchmarks.

All these examples of multi-species PSO make a good starting point for further hybridization of this potent metaheuristic. The approach presented in this paper provides one such extension.

3 From perspective taking to enhancing PSO

Perspective taking is a core socio-cognitive ability which allows smooth social interactions between humans. There is substantial variability in human's perspective-taking performance. Part of the diversity can be explained by the relative weight that an individual gives to his/her own perspective relative to the perspective of other surrounding people [1]. Thus, some individuals give more weight to their own perspective (egocentric individuals) while other individuals give more weight to other people's perspective (altercentric individuals).

In [6], we have explored the usefulness of human perspective-taking characteristics on the search capabilities of ACO, enhancing it by introducing different sub-populations, e.g. *Egocentric individuals* (focused on their own knowledge) or *Altercentric individuals* (perceiving the actions of other ants and using them as inspirations). Such complex populations achieved interesting results in discrete optimization (namely solving TSP problem) increasing the efficiency in many of the studied instances.

PSO, as another swarm intelligence algorithm, seems a natural candidate for introducing socio-cognitive mechanism. In the case of ACO, the perspective was taken based on the different pheromone trails left by particular ant species, pushing the ants towards the sub-optimal solutions found by the pheromone-depositing ant species from the population. However, in the case of PSO, the mechanism of perspective taking may be implemented more easily. The act of following different current optima — best for the particular individual, best in the neighborhood, best globally — will be elaborated in the rest of this paper, resulting in a metaheuristic system that is conceptually similar to the socio-cognitive ACO.

4 Classic PSO

In the basic particle swarm optimization [5] implementation, the potential solutions are located in a subspace of n-dimensional Euclidean space R^n limited in each dimension (usually a n-dimensional hypercube). The search space is a domain of the optimized quality function $f : R^n \to R$.

A particle is a candidate solution described by three n-dimensional vectors: position $X = x_i, i \in [1...n]$; velocity $V = v_i, i \in [1...n]$; best known position $A_p = x_i, i \in [1...n]$. A swarm is a set of m particles. The swarm is associated with a n-dimensional vector $A_s = x_i, i \in [1...n]$ which is swarms best known position (the solution with the currently highest quality). Neighborhood is a subset of swarm. It is associated with a n-dimensional vector $A_n = x_i, i \in [1..n]$.

The execution of the algorithm begins by initializing the start values. Each particle P belonging to the swarm S and in the neighborhood N is initialized with the following values:

1. position X of the particle P is initialized with a random vector belonging to the search space D,
2. best known position is initialized with current particles position: $A_p := X$
3. velocity V of the particle P is initialized with a random vector belonging to the search space D
4. swarm's best position is updated by the following rule: $iff(A_p) < f(A_s)thenA_s := A_p$

5. the neighborhood's best position is updated by following rule: $iff(A_p) < f(A_n) then A_n := A_p$

Once all the particles are initialized and uniformly distributed in the search space, the main part of the algorithm starts executing. During each iteration of the algorithm, the following steps are executed. These steps of the algorithm are executed until a termination criterion are

---

for `each particle P in swarm S` **do**
  `update particles position:`
    $X := X + V$
  `update particles velocity:`
    $V := a(A_s - X) + b(A_n - X) + c(A_p - X) + \omega V; a, b, c \in [0, 1]$
    where ω is the inertia factor
  `update global best positions:`
    $iff(A_p) < f(A_s) then A_s := A_p$
    $iff(A_p) < f(A_n) then A_n := A_p$
end for

---

met. The most common termination criteria for the particle swarm optimization are:

1. number of executed iterations reaches a specified value,
2. swarm's best position exceeds a specified value,
3. the algorithm find global optimum,
4. swarm's best positions in two subsequent iterations are the same.

5 Socio-cognitively inspired PSO

The classic implementation of the PSO presented above makes the algorithm very likely to become stuck in a local optima. Once each particle reaches its optimum state, the swarm stabilizes and it is impossible for it to leave this state. In such a situation, the best position of each particle, its neighborhood and the whole swarm will be located on the same point, and the calculated velocity will always be 0. This situation is caused by the low number of parameters affecting a particle's behavior. Each particle is influenced by three factors — it's own, the swarm's and the neighborhood's best positions — which affect the particle equally. This situation can be rectified by dividing the particles into different classes called the species.

Each species has its own algorithm for calculating the velocity. The differences between the algorithms for different species lies in different weight settings for each particle species. As described above, a particles velocity is determined by the following equation:

$$V := a(A_s - X) + b(A_n - X) + c(A_p - X) + \omega V; a, b, c \in \{0, 1\}$$

Changing values of the parameters a, b, and c modifies the impact of the three factors, while at the same time making the individuals perceive the results achieved by the other PSO sub-species. A further step is to incorporate the perspective-taking inspiration: namely the act of receiving inspiration from the results achieved by the other sub-species.

The species described in this paper were created by choosing different configurations of the parameters a, b, c (see Table 1):

Table 1: Configurations of the socio-cognitively inspired sub-species for different decision factors

No.	Species	Swarm [a]	Neighbor [b]	Particle [c]
1	Normal	1	1	1
2	Global and local	1	0	1
3	Global and neighborhood	1	1	1
4	Local and neighborhood	0	1	1
5	Global only	1	0	0
6	Local only	0	0	1
7	Neighborhood only	0	1	0
8	Random	random	random	random

- *Normal*: classic PSO where the particle's decisions are affected by the particle's optimal solution, the neighborhood's optimal solution and the global optimal solution, but each of these is given the same weight.
- *Global and local* This species is influenced only by its own best and global best positions.
- *Global and neighborhood* This species calculates its velocity by combining the influences of the swarm and the neighborhood best positions.
- *Local and neighborhood* This species ignores the global best solution and takes into account only the local bests - the neighborhood's and its own best positions.
- *Global only, Local only, Neighborhood only* These three species of particles are influenced only by a single factor: by it's own, the neighborhood's, or the swarms best position, respectively.
- *Random* The last species of particles introduces randomness into the algorithm. It takes into consideration all three parameters, but in each iteration the relative weights of the parameters are randomly reassigned. (The sum of the weights is always equal to 3.) The weight calculation is described by the following pseudocode:

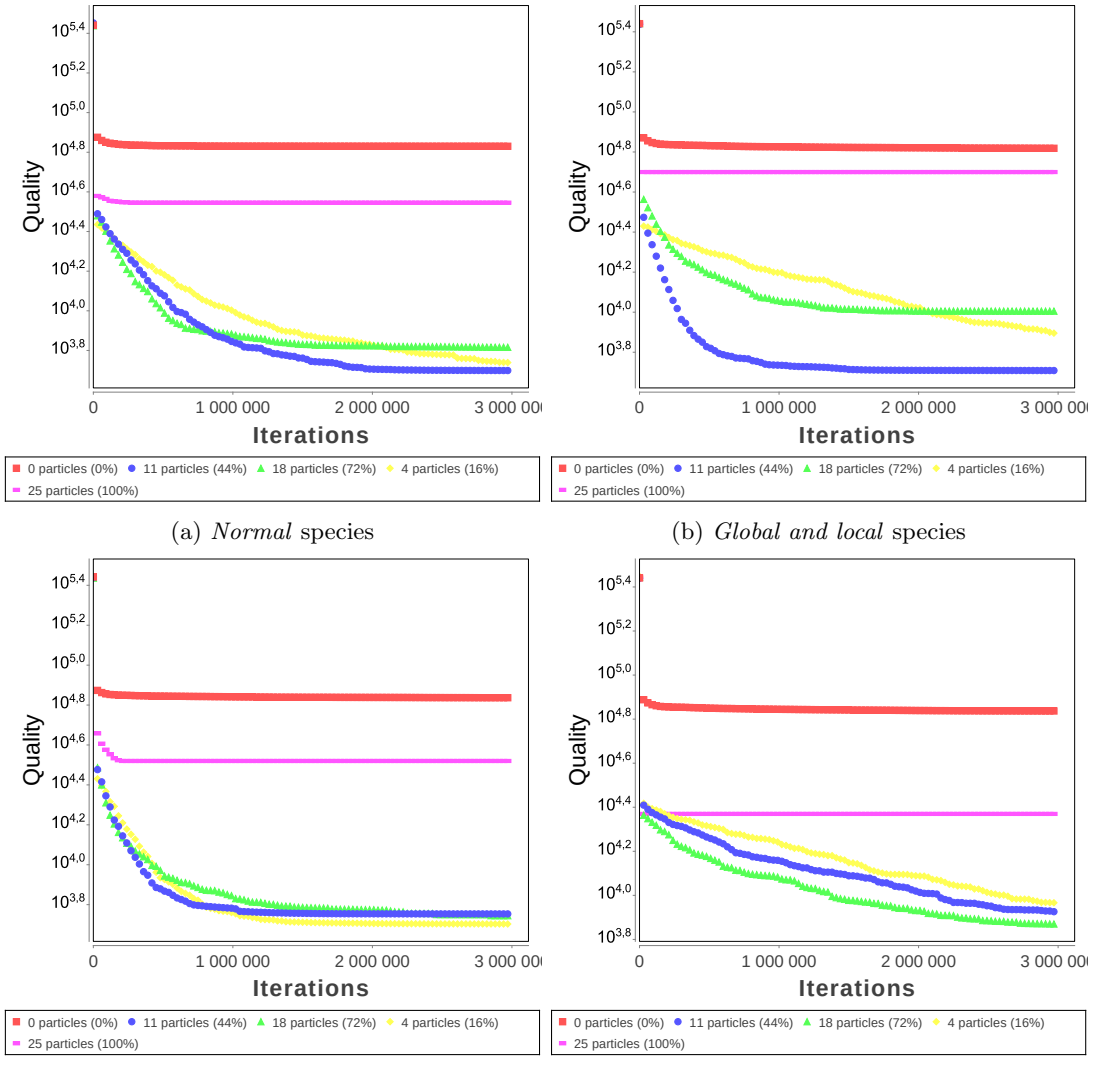
$a = random(0, 3)$
$b = random(0, 3 - a)$
$c = 3 - a - b$

While conducting experiments with different PSO configurations, we consider populations containing 'pure' species (only one kind of species) as well as those containing various proportions of different species. This allows us to study the optimization potential of the proposed metaheuristics.

6 Experimental results

The experimental results presented here were obtained using one node of Zeus supercomputer (Intel Xeon HP BL2x220c, 23 TB RAM, 169 TFlops total computing power). During the experiments, the well-known benchmarks, namely Rastrigin, Griewank, Rosenbrock and Ackley [3]

functions in 100 dimensions were executed. The search space was a 100-dimensional hypercube limited in each dimension to range [-100,100]. The experiments were repeated 30 times. Each experiment for Rastrigin and Rosenbrock benchmark consisted of conducting $3 \cdot 10^6$ iterations of the algorithm. Experiments for Griewank and Ackley consisted of $3 \cdot 10^6$ or of iterations.



(a) *Normal* species

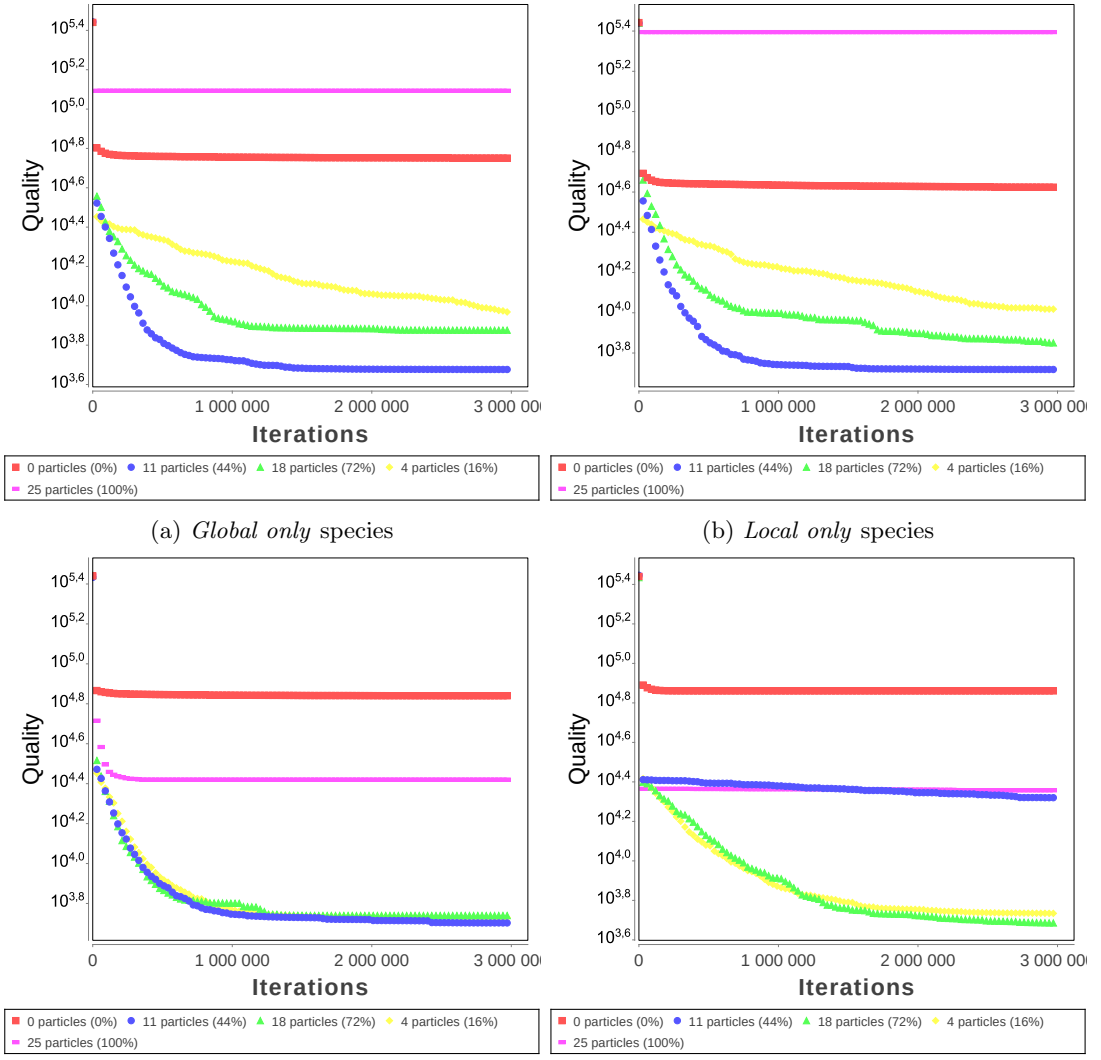
(b) *Global and local* species

(c) *Global and neighbourhood* species

(d) *Local and neighbourhood* species

Figure 1: Quality with different fractions of socio-cognitive subpopulations optimizing Rastrigin function (first 4 species)

All the experiments were conducted for 25-particle swarms consisting of different species. The particles were grouped in one-dimensional neighborhoods of size 5, so that every swarm was split into 5 neighborhoods each containing 5 consecutive particles. In each graph, presented in Fig. 1 and Fig. 2, the proportion of the examined species in the tested population is shown at the bottom, with all the other particles being divided equally among the remaining



(a) *Global only* species

(b) *Local only* species

(c) *Neighbourhood only* species

(d) *Random* species (remaining 4 species)

Figure 2: Quality with different fractions of socio-cognitive subpopulations optimizing Rastrigin function

species. Each species was examined by running simulations for 5 different swarms. Each swarm contained 0, 4, 11, 18 or 25 particles of the examined species and an equal share of the remaining species. For example, if *normal* species is being examined, 100% means that the population contains only classic PSO particles, while 44% means that the population consists of 11 classic PSO particles, and the remaining 14 were divided equally among the 7 remaining species (so two particles for each species.)

We searched for efficient configurations of the whole system, checking them against the executed benchmarks, being aware that, as suggested by the well-known *No Free Lunch Theorem* [11], no one single configuration can prevail for all possible optimization problems.

Examining the graphs that display the best qualities observed for different configurations of species (Fig. 1 and Fig. 2) for optimizing Rastrigin function, we can see that in almost all the experiments, the examined proportions of the species 44%, 16% and 72% yielded a performance close to the optimum. The best achieved results for Rastrigin function were around 4700. Of course, it was still far from the global optimum; however it reached much closer to the global optimum than the classic PSO which got stuck at 35129.23.

This experiment allowed us to select a number of promising configurations by combining the best swarms for which the results are shown in Figs. 1 and 2. The selected configurations are described in Tab. 2. Once the configurations were selected, the experiments were run again for each of the mentioned benchmarks. The new experiments were conducted for the same initial conditions: 25 particles per swarm, one-dimensional neighborhood of size 5, 100 dimensions and $3 \cdot 10^6$ iterations. The quality of the examined configurations is compared with the classic PSO in Fig. 3.

The result of this experiment show that the proposed metaheuristic still needs the tuning of parameters. However we can already see that Rastrigin and Ackley benchmarks made the system stuck in local extrema (after about 500000 iterations), while in the case of Rosenbrock and Griewank the system was still exploring, and might even go past the local optimum if the experiments were prolonged.

7 Conclusions

We presented a new vision of the PSO consisting of different, cooperating species. The species constituting the swarm leverage socio-cognitive ideas by getting inspired by the results presented by other species present at the same time in the population. Our results show that the proposed metaheuristics perform better than the classic PSO. We have examined a number of different swarm configurations, and have gathered data for further development of the algorithm to enhance their performance.

Our experiments are of course burdened with the curse of dimensionality, which makes searching for the optimum quite difficult. We would like to point out that Sugimoto et al. [7] achieved tangibly better results by getting to 258 for 100-dimensional Rastrigin function. We feel that our results show the efficacy of the proposed metaheuristics, and improving their overall efficiency will require a modification of the initial algorithm.

The research in enhancing the PSO algorithm with introducing socio-cognitive inspirations is continued. We plan to investigate the influence of parameters like inertia factor or velocity limit on particular species. A new version of algorithm in which different particles can have different values of mentioned parameters will be tested with selected benchmarks. The next phase of studies will be to prepare swarms consisted of particles dynamically changing their species. In such case the change will depend on the effectiveness of a given particle.

Acknowledgments

This research received partial support from Polish-Belgian Joint Research Project (2015–2017), under agreement between the PAN and FNRS: “Modelling the emergence of social inequality in multi-agent systems”. This research received partial support from AGH University of Science and Technology, Faculty of Computer Science, Electronics and Telecommunications Dean’s Grant for Young Scientists. This research was supported in part by PL-Grid Infrastructure.

Table 2: Number of particles of given species in selected configurations

Swarm name	Normal	Glob. and local	Global and neigh.	Local and neigh.	Global only	Local only	Neigh-bour only	Random weights
Classic	25	0	0	0	0	0	0	0
Swarm 1	5	10	5	0	0	0	0	5
Swarm 2	5	10	5	5	0	0	0	0
Swarm 3	0	0	0	0	7	7	7	4
Swarm 4	6	0	0	0	5	5	5	4
Swarm 5	3	6	4	2	2	2	2	4

References

[1] H.B. Bukowski. *What Influences Perspective Taking A dynamic and multidimensional approach*. PhD thesis, Université catholique de Louvain, 2014.

[2] Erick Cantu-Paz. *Efficient and Accurate Parallel Genetic Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 2000.

[3] J.G. Digalakis and K.G. Margaritis. An experimental study of benchmarking functions for genetic algorithms. In *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*, volume 5, pages 3810–3815 vol.5, 2000.

[4] Masao Iwamatsu. Multi-species particle swarm optimizer for multimodal function optimization. *IEICE TRANSACTIONS on Information and Systems*, E89-D(3):1181–1187, 2006.

[5] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948 vol.4, Nov 1995.

[6] Mateusz Sekara, Michal-Kowalski, Aleksander Byrski, Bipin Indurkhya, Marek Kisiel-Dorohinicki, Dana Samson, and Tom Lenaerts. Multi-pheromone ant colony optimization for socio-cognitive simulation purposes. *Procedia Computer Science*, 51:954 – 963, 2015. International Conference On Computational Science, ICCS 2015: Computational Science at the Gates of Nature.

[7] Masaki Sugimoto, Taku Haraguchi, Haruna Matsushita, and Yoshifumi Nishio. Particle swarm optimization containing plural swarms. In *In Proc. of 2009 International Workshop on Nonlinear Circuits and Signal Processing NCSP'09, Waikiki, Hawaii*. 2009.

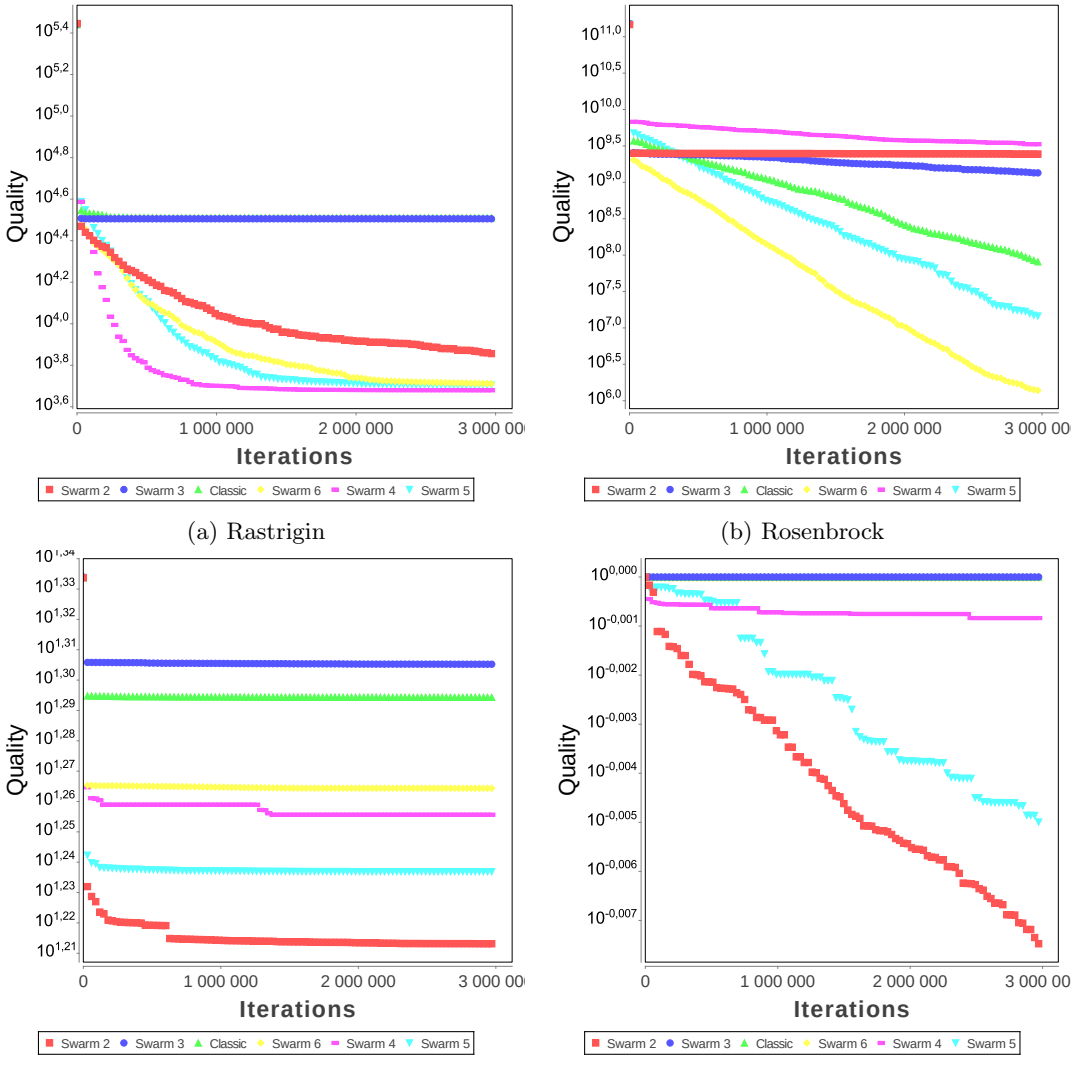
[8] E. Swiderska, J. Lasisz, A. Byrski, T. Lenaerts, D. Samson, B. Indurkhya, A. Nowé, and M. Kisiel-Dorohinicki. Measuring diversity of socio-cognitively inspired aco search. In *Applications of Evolutionary Computation, Proc. of 19th European Conference, EvoApplications 2016, Porto, Portugal, March 30–April 1*, Lecture Notes in Computer Science 9597. Springer, 2016.

[9] El-Ghazali Talbi. *Metaheuristics: From Design to Implementation*. Wiley, 2009.

[10] F. van den Bergh and A.P. Engelbrecht. A cooperative approach to particle swarm optimization. *Evolutionary Computation, IEEE Transactions on*, 8(3):225–239, June 2004.

[11] David H. Wolpert and William G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

[12] G.G. Yen and Wen Fung Leong. Dynamic multiple swarms in multiobjective particle swarm optimization. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 39(4):890–911, July 2009.



(a) Rastrigin

(b) Rosenbrock

(c) Ackley

(d) Griewank

Figure 3: Quality of selected configurations for each mentioned benchmark