A Comparison of Particle Swarm Optimization Algorithms Based on Run-Length Distributions

Marco A. MONTES DE OCA, Thomas STÜTZLE, Mauro BIRATTARI and Marco DORIGO
IRIDIA, CoDE, Université Libre de Bruxelles, Brussels, Belgium
{mmontes, stuetzle, mbiro, mdorigo}@ulb.ac.be

IRIDIA
Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle

ULB

Summary

We compare seven Particle Swarm Optimization (PSO) algorithms using Run-Length Distributions (RLDs). RLDs measure the probability of finding a solution of a certain quality after some specific number of function evaluations. Hence, RLDs completely characterize the performance of a stochastic optimization algorithm on a particular problem. The analysis of RLDs suggests ways of improving the performance of the studied variants.

Motivation and Goal

- Since the introduction of the first PSO algorithm, many algorithmic variants have been proposed.
- For years, authors have been comparing their variants only with one PSO algorithm (i.e., with the original one, or – more recently – with the so-called canonical PSO algorithm).
- To the best of our knowledge, there are no comparisons among variants reported in the literature.
- Consequently, the field lacks a clear definition of the set of algorithmic variants that can be considered the state-of-the-art.

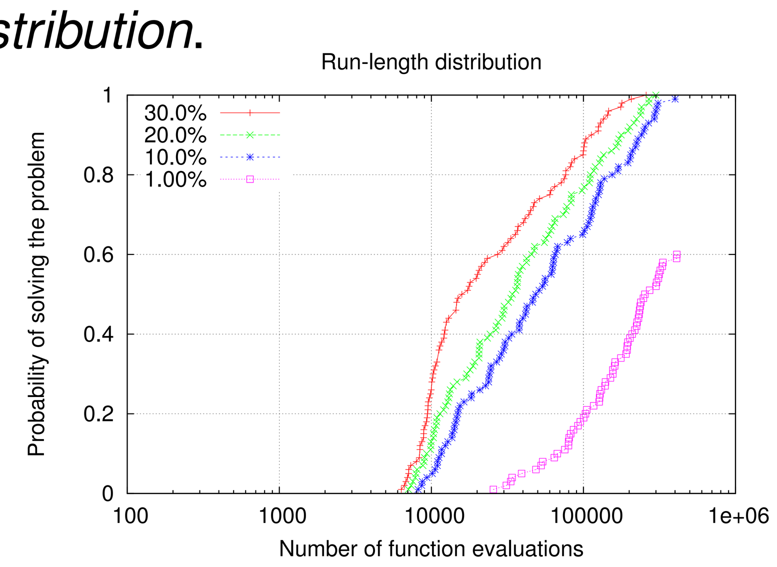
We compared some of the most influential PSO variants in order to identify the best performing ones among them. In this way, we contribute to the definition of the state-of-the-art in the field.

Run-Length Distributions

Let T_q be a random variable modeling the time needed by a stochastic optimization algorithm to find a solution of quality q. The *run-time distribution* of the algorithm is defined as

$$RT_q(t) = P(T_q \leq t)\,.$$

It is the cumulative distribution function of T_q. When run-times are measured in terms of objective function evaluations, a run-time distribution is called *run-length distribution*.



Experiments and Results

Experimental Setup

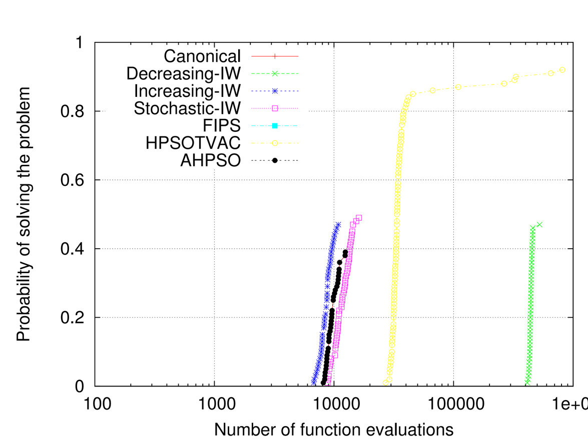
We compared seven PSO variants in 5 benchmark functions (Sphere, Rosenbrock, Rastrigin, Griewank, and Schaffer's F6 functions in 30 dimensions). The compared variants are:

Compared PSO variants
Canonical (Clerc and Kennedy, 2002)
Time-decreasing inertia weight (DEC-IW) (Shi and Eberhart, 1998)
Time-increasing inertia weight (INC-IW) (Zheng et al., 2003)
Stochastic inertia weight (Stochastic-IW) (Eberhart and Shi, 2001)
Fully informed PSO (FIPS) (Mendes et al., 2004)
Self-Organizing Hierarchical PSO with time-varying acceleration coefficients (HPSOTVAC) (Ratnaweera et al., 2004)
Adaptive Hierarchical PSO (AHPSO) (Janson and Middendorf, 2005)

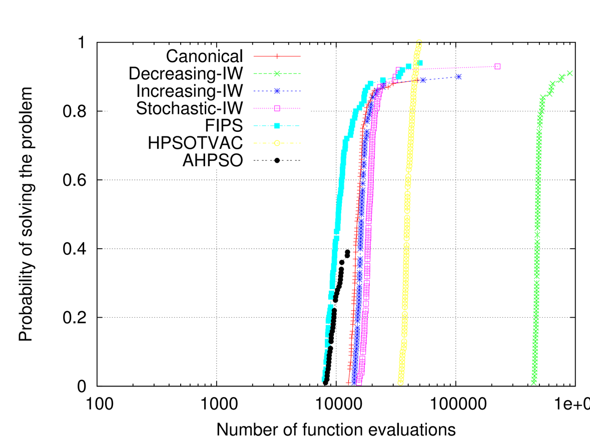
We ran 100 independent trials of 1 000 000 function evaluations. We used swarms of 20 particles using fully connected (FC) and ring (R) topologies.

We tested all the algorithms under the assumption that no *a priori* knowledge about the structure of the problem was available. Thus, each algorithm used the same parameterization across all benchmark problems.

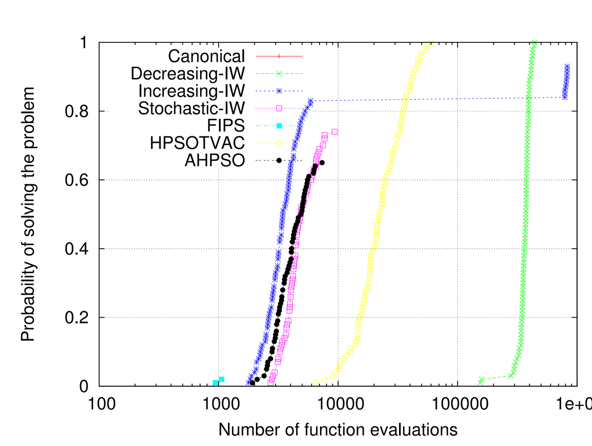
Results



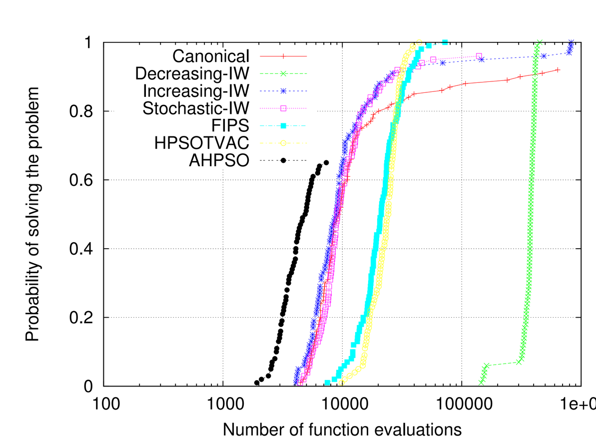
Griewank FC



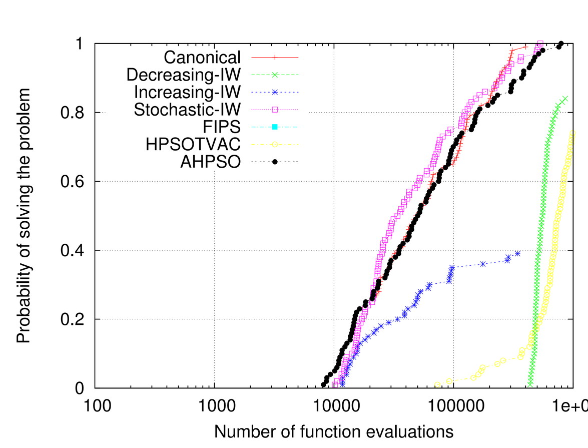
Griewank R



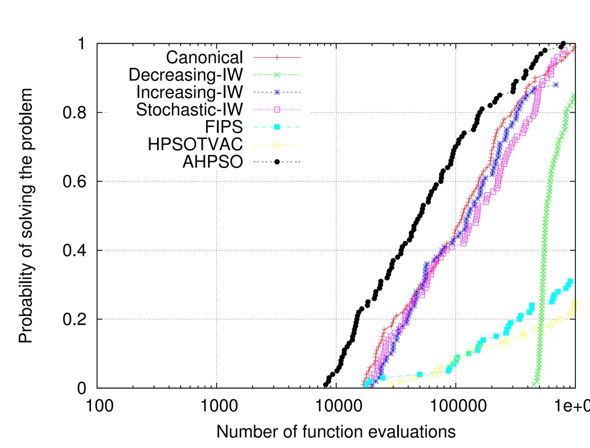
Rastrigin FC



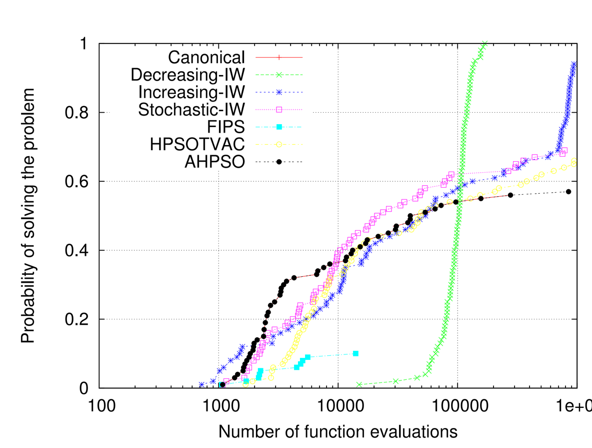
Rastrigin R



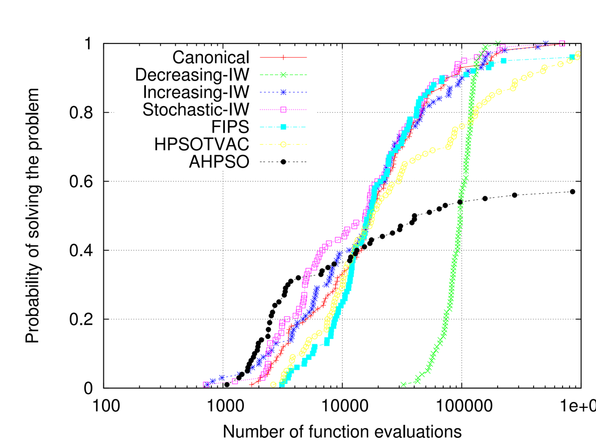
Rosenbrock FC



Rosenbrock R



Schaffer's F6 FC



Schaffer's F6 R

Conclusions

- No algorithm dominates all the others.
- Different algorithms exhibit a stagnating behavior with different degrees of severity, which can be alleviated by changing the population topology to a loosely connected one.
- Different algorithms are sensitive to a change in the population topology in different degrees.
- For short runs, PSOs with convergent properties get the best results. For medium-long and long runs, PSOs with explorative properties are the best performing.

We hope that in the future authors consider more than just one reference PSO to assess the significance of their results. For different conditions (e.g. the maximum number of function evaluations), this reference set should consider different PSO variants.

References

Clerc, M., Kennedy, J.: The particle swarm–explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation 6(1) (2002) 58–73

Shy, Y., Eberhart, R.: A modified particle swarm optimizer. In: Proceedings of the 1998 IEEE World Congress on Computational Intelligence, Piscataway, NJ, IEEE Press (1998) 69–73

Zheng, Y.L., Ma, L.H., Zhang, L.Y., Qian, J.X.: Empirical study of particle swarm optimizer with an increasing inertia weight. In: Proceedings of the 2003 IEEE Congress on Evolutionary Computation, Piscataway, NJ, IEEE Press (2003) 221–226

Eberhart, R., Shi, Y.: Tracking and optimizing dynamic systems with particle swarms. In: Proceedings of the 2001 IEEE Congress on Evolutionary Computation, Piscataway, NJ, IEEE Press (2001) 94–100

Mendes, R., Kennedy, J., Neves, J.: The fully informed particle swarm: Simpler, maybe better. IEEE Transactions on Evolutionary Computation 8(3) (2004) 204–210

Ratnaweera, A., Halgamuge, S.K.,Watson, H.C.: Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coeffcients. IEEE Transactions on Evolutionary Computation 8(3) (2004) 240–255

Janson, S., Middendorf, M.: A hierarchical particle swarm optimizer and its adaptive variant. IEEE Transactions on Systems, Man and Cybernetics–Part B 35(6)(2005) 1272–1282

This work was supported by the ANTS project, an *Action de Recherche Concertée* funded by the Scientific Research Directorate of the French Community of Belgium. Marco Montes de Oca acknowledges support from the Programme Alβan, the European Union Programme of High Level Scholarships for Latin America, scholarship No. E05D054889MX. Thomas Stützle and Marco Dorigo acknowledge suport from the Belgian National Fund for Scientific Research (FNRS), of which they are a Research Associate and a Research Director, respectively.