

Scalability of Particle Swarm Algorithms

Sébastien Piccand
Department of Computer
Science & Information
Systems
University of Limerick
Limerick, Ireland
sebastien.piccand@ul.ie

Michael O'Neill
Natural Computing Research
& Applications
School of Computer Science &
Informatics
University College Dublin
Dublin, Ireland
m.oneill@ucd.ie

Jacqueline Walker
Department of Electronic and
Computer Engineering
University of Limerick
Limerick, Ireland
jacqueline.walker@ul.ie

ABSTRACT

When dealing with complex optimisations problems, evolutionary computation techniques have proven to be very helpful. Amongst optimisation algorithms driven by evolutionary computation techniques, particle swarm algorithms have proven to be a very good alternative to genetic algorithms because of their faster convergence. However they can still suffer from premature convergence to local optima. Premature convergence occurs when the particles of the swarm are too close to each other to enable further exploration of the space. To put it another way, the dispersion or distribution of the swarm throughout the search space has been localised to a small region with a consequent stagnation of the search process. Many strategies have been used to try to prevent convergence to a local optimum. However little work has been done on problems of high dimensions. By high dimensions, we mean dimensions of 100 and above. This paper focuses, therefore, on the limitations of classical particle swarm algorithms when dealing with high dimensional problems. We compare different versions of Particle Swarm Algorithms: GBest, LBest with ring or random neighbourhood of size 3 [2], and GCPSO [4]. Those algorithms were run twice, with a linearly decreasing inertia weight, and with the use of a constriction factor. We also used two repulsive-based algorithms: Charged PSO [1] and Predator Prey [3].

Our main focus is problems of high dimensionality. In particular, we applied the different algorithms to the benchmarks functions Ackley and Rosenbrock, in the following dimensions: 100, 250, 500. Even though these represent problems of relatively small dimensionality in a real-world context, experiments on higher dimensions have not been necessary to show the limits of the algorithms studied. Each experiment was run 20 times. The swarm size was chosen from [30 50 100 250 500] and so that it is not greater than the problem size. Each experiment is 10000 iterations long. A one-way analysis of variance (ANOVA) was used to compare the performance of each algorithms.

We found that the LBest algorithms perform significantly better when used with the constriction factor. GBest and GCPSO perform better with linearly decreasing inertia with a small swarm size, but better with the constriction factor with a big swarm size. The improvement of GCPSO on GBest is not statistically significant in our experiments.

The LBest algorithms with the constriction factor seem to be the best algorithms to handle problems of high dimensionality. The LBest algorithm with fixed neighbourhood seems to be less sensitive to the size of the swarm than the LBest algorithm with random neighbourhood. Especially, in the case of the Rosenbrock function of size 500, increasing the size of the swarm does not improve the performance of LBest with constricted factor and fixed neighbourhood.

The algorithms based on repulsion between particles, i.e. Charged Swarm and Predator Prey, do not perform very well. Indeed, even if the predator prey algorithm gives quite good results, it is trapped in a local optimum, as the fitness value stagnates on a constant value for the last 50% of iterations. This may come from a too low level of repulsion. Tuning the parameters used for repulsion seems to be very important for high dimensionality problems.

Experiments show that almost all the algorithms managed to solve the problems for dimension 100 but none of the algorithms managed to solve the problem in the case of problems of dimension 250 and 500. The LBest algorithm with random neighbourhood and constriction factor performed the best.

Further work will be done on modelling the size of the swarm required to be able to solve the problems. Other particle swarm algorithms will also be included.

Categories and Subject Descriptors: I.2.8 [Artificial Intelligence: Problem Solving, Control Methods, and Search]: Heuristic methods; G.1.6 [Numerical Analysis: Optimization]: Global optimization; [Track]: Ant Colony Optimization, Swarm Intelligence, and Artificial Immune Systems

General Terms: Algorithms, Performance, Experimentation

Keywords: swarm intelligence, optimization, high dimensionality

REFERENCES

- [1] T. M. Blackwell and P. J. Bentley. Dynamic search with charged swarms. In *GECCO '02*, 2002.
- [2] M. Clerc. *Particle Swarm Optimization*. 2006.
- [3] A. Silva, A. Neves, and E. Costa. An empirical comparison of particle swarm and predator prey optimisation. In *AICS '02*, 2002.
- [4] F. van den Bergh. *An analysis of particle swarm optimizers*. PhD thesis, University of Pretoria, 2006.