ISSN 1744-1986

The Open University

Technical Report N^{o} 2008/09

DYNAMIC,HIERARCHICAL PARTICLE SWARM OPTIMISATION

Ian Kenny

26th June 2008

Department of Computing
Faculty of Mathematics and Computing
The Open University
Walton Hall,
Milton Keynes
MK7 6AA
United Kingdom

http://computing.open.ac.uk

DYNAMIC,HIERARCHICAL PARTICLE SWARM OPTIMISATION

Ian Kenny

Computing Department,
Faculty of Mathematics & Computing,
The Open University,
Walton Hall,
MILTON KEYNES.
MK7 6AA

Particle Swarm Optimisation (PSO) is an optimisation technique based on the principle of social influence. It has been applied successfully on a wide range of optimisation problems. This paper considers the possibility of a dynamic hierarchical extension to the particle swarm technique, allowing the swarm to consider several related datasets. This provides the advantage of being able to consider several data scans and aggregate the results into a master swarm model.

1 Introduction

Particle Swarm Optimisation (PSO) is an evolutionary optimisation technique originally developed by Kennedy and Eberhar t[8]. The original algorithm was not developed to optimise, but rather as a simple model of social behaviour. Once the potential for optimisation had been realised, the original algorithm underwent developments in order to improve its convergent and optimisation potential. Arguably the most important of these are due to Shi and Eberhart[12], the inertia weight variant improving the convergent behaviour of PSO, Clerc and Kennedy[1], the constriction factor variant, which defined the necessary relationships between the PSO parameters to affect convergence and van den Bergh[15] who developed the GCPSO a variant which, he argues, is guaranteed to converge on local optima. The inertia weight variant and the constriction factor variant were compared by Eberhart and Shi[3], the constriction factor variant is a special case of the inertia weight variant defining the optimal relationship for convergence between the parameters.

PSO can simply be described as follows: The basic idea is that potential solutions, called particles, 'fly' through the solution space and are influenced by their neighbours and their own past performance. Each particle represents a candidate solution, Figure 1 illustrates the overall principle.

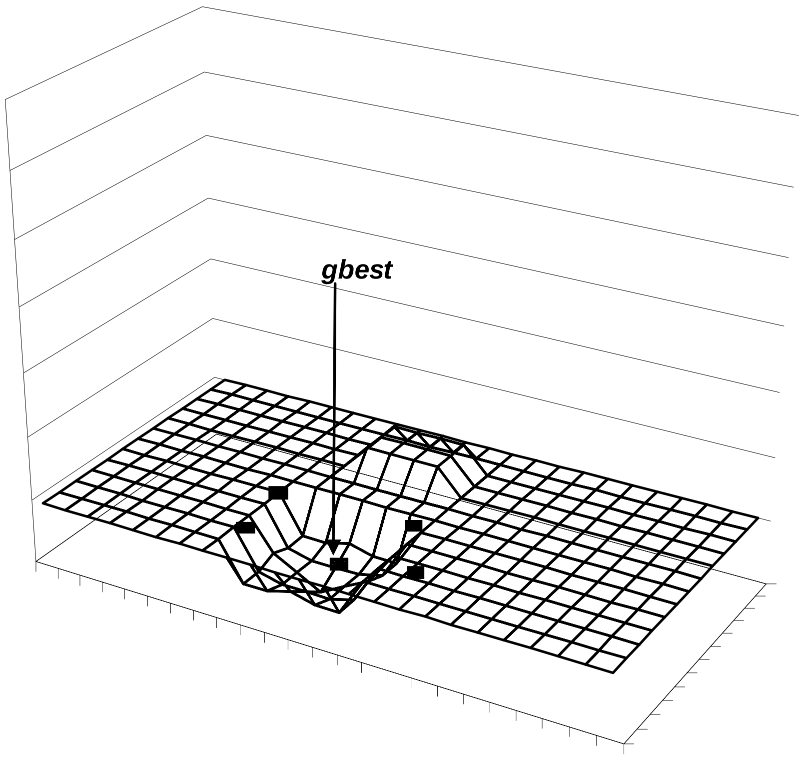

Figure 1: An idealised view of a swarm converging on a minimisation problem, gbest,the global best, is shown as the 'lowest' particle in the centre.

A typical population is between 20 and 40 particles which explore the solution space. The original form of the equations proposed by Kennedy & Eberhart[8] were developed by Shi & Eberhart[12] to give the conical form:

$$\vec{v}'_i = w \cdot \vec{v}_i + \varphi_{1i} \cdot (\vec{p}_i - \vec{x}_i) + \varphi_{2i} \cdot (\vec{p}_g - \vec{x}_i) \qquad 1a$$

$$\vec{x}'_i = \vec{x}_i + \vec{v}'_i \qquad 1b$$

Equation 1: conical PSO equations

In

Equation *1*a, the particle's previous velocity is added to the new velocity calculated via the appliance of the particle's previous best and the best of the population. It is apparent that this makes the velocity, a cumulative sum of the particle's previous influences. Therefore, it was necessary to implement a *vmax* parameter to restrict the maximum velocity. This was defined as follows:

$$(x_{id} > v_{max}) x_{id} : v_{max}$$
$$(x_{id} < -v_{max}) x_{id} : -v_{max}$$

Equation 2: velocity restriction, curtails the velocity to vmax

where *i* is the particle's index, *d* is the dimension and *vmax* was typically set to *xmax*; following investigations by Shi & Eberhart[11] *vmax* is implemented as the test in

Equation *2* after the update of the velocity for each dimension per iteration.

An interesting consequence of the standard particle swarm's implementation is that it is unable to consider datasets which have related areas of data by which I mean datasets, which have multiple readings for a given subject or example, geological or medical data, whether subject is the site or the patient, respectively.

2 The Need for a Hierarchical Approach?

The described process above of convergence results of a particle swarm optimisation, implemented in its generic form, can only consider a single dataset at a given time. For example, if one considers medical data, such as EEG scans, PSO can only accurately consider a single scan per patient at a given time. Generally this can be seen as a limitation in terms of being able to analyse multiple readings in a single dataset. It is difficult to envisage a generic or accurate method of aggregating data gained from multiple scans without the data from each scan, and potentially contributing to that of another scan and therefore distorting the score of the other scan. At the same time, it is clear that in order to process medical data and produce a valid result, multiple scans per patient need to be aggaregated.

The concept of a dynamic hierarchical particle swarm is proposed here. In this model child swarms are allowed to explore the scans of a given patient. The master, or hierarchical swarm, is unable to aggregate the data scores from the child swarms and cluster them appropriately. For example, if one was trying to determine epileptic and non-epileptic episodes the child swarms would be given the EEG scans the process with the *gbest* of each swarm being passed to the master swarm for each iteration. A dynamic clustering process could then be used to determine the emerging clusters from within the swarms. This has the advantage of not needing to predetermine the number of clusters, expected to be found in the data. It also gives particle swarm and advantage over other techniques since it can dynamically adapt to the scores around it.

As mentioned previously, however, the major benefit of this technique is seen as giving PSO the ability to consider more generic forms of data space, where there are discreet areas of space within the overall problem space. It also, by its nature, allows different objective functions to be applied by the swarms within the solution space. This might depend on their level within the hierarchy; alternatively, each child swarm might operate with a different objective function. Conceptually, this provides a very flexible optimisation technique and extends the strength of PSO in being able to consider the same dataset simultaneously in several different ways, and then aggregating the results of the swarm's processing within the master swarm.

Taking the process that little further, one needs to consider the actions of the swarm at higher levels of a hierarchy. Clearly, it is more sensible for them to act as a repellents rather than attractors. Since, if they were merely to act as an attractor, they would hasten the child swarms convergence on a point within their sub-swarm solution space, which was closest in value to the global best (one might consider it the universal best), represented by the master swarm. Through the repulsion implementation in the master swarm, it forces the *gbest* particle in each child swarms to explore other areas. This directly addresses, the stagnation problem tackled through the van den Burgh[14][15] however it does so without needing any additional feedback from the nature of the solution space.

2.1 Dynamic hierarchical swarm implementation

The previous section outlines the concept of a dynamic hierarchical swarm. I shall now consider how this might be implemented.

Conceptually, this is visualised as a swarm of *gbests* from the child swarms with the master swarm applying a repellents force on the *gbests* from its child swarms. By taking the *gbests* from each child the repellent equation given below is applied to push the *gbest* farther afield.

Clearly, the particle swarm equations need to be the formulated to act as a repellents rather than attractors. A fairly straightforward modification to the equations, is given below:

$$\vec{v}'_i = w \cdot \vec{v}_i + \varphi_{1i} \cdot (\vec{p}_i + \vec{x}_i) + \varphi_{2i} \cdot (\vec{p}_g + \vec{x}_i) \qquad \text{1a}$$
$$\vec{x}'_i = \vec{x}_i + \vec{v}'_i \qquad \text{1b}$$

Equation 3: a modified, repellent, form of the particle swarm equations.

Equation 3 shows a modified particle equation where the navigation elements in 1a are replaced with additions. This is one possible to change and has the disadvantage that it will lead to large increments in the particle step size. However, it should be remembered that this is a modification applied to the master swarm and the particle involved will be influenced in the next iteration, through the conical form of the equations. Obviously, an alternative to change would be to reverse all the signs in the equations. However, this would lead to greater change in the swarm's behaviour. Further study needs to be carried out as to the appropriate modification.

As mentioned before one other possible applications of this extended version of the particle swarm optimisation technique is processing, medical data, where each patient has multiple scans to be considered simultaneously, or as a whole. In addition to medical data, new number of application areas such as geological data, oceanographic astronomical observations, processing can be attacked. Indeed, any application where multiple observations can be generated and need to be considered simultaneously or as a whole may be appropriate to this application of particle swarm.

3 Conclusion

This paper has considered the implementation of a dynamic hierarchical particle swarm with the intention of providing a more flexible integrated way of considering multiple datasets, while keeping the separate identity of each dataset. I believe that this will produce an extension of particle swarm, which will allow it to be considered as an optimisation technique in its own right, rather than supporting another technique. The reasoning behind this is that the hierarchical level provides an effective aggregation of the more discreet results from the lower level swarms. In the same way that a neural network aggregates the values in the preceding layers

Further research is needed, to explore the appropriate configuration and usage of hierarchical particle swarm. However, it is hoped that the intrinsic strengths of PSO can be utilised effectively to produce dynamic optimisation heuristic. The research should, in particular, concentrate on how the relative influences of the swarm is that different levels can be utilised, whilst preserving the information gathered at the lower levels.

[1] Clerc M. & Kennedy J., (2002), “The Particle Swarm - Explosion, Stability, And Convergence In A Multidimensional Complex Space”, Evolutionary Computation, IEEE Transactions On , Volume: 6 , Issue: 1, pp.58-73

[2] Eberhart R & Kennedy J, (2004), “Human Tremor Analysis With Particle Swarm Optimisation”, Proceedings Of The 2003 IEEE Swarm Intelligence Symposium. SIS'03 (cat. No.03ex706), pp.1927-1930

[3] Eberhart, R.C. & Shi Y, (2000), “Comparing Inertia Weights And Constriction Factors In Particle Swarm Optimization”, Evolutionary Computation, 2000. Proceedings Of The 2000 Congress On volume 1, 16-19 July 2000, pp.84 - 88

[4] Engelbrecht A. P. et al, (2005), "Niching Ability Of Basic Particle Swarm Optimization Algorithms", Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE , vol., no.pp. 397- 400, 8-10 June 2005, pp. 397- 400

[5] Higashi N & Iba H, (2003), “Particle Swarm Optimisation With Gaussian Mutation”, Proceedings Of The IEEE Swarm Intelligence Symposium 2003 (SIS 2003), Indianapolis, Indiana, USA., pp.72-79

[6] Kennedy J. & Mendes R., (2003), “Neighborhood Topologies In Fully-Informed And Best-of-Neighborhood Particle Swarms”, Proceedings Of The 2003 IEEE International Workshop On Soft Computing In Industrial Applications 2003 (SMCIA/03), pp.445-450

[7] Kennedy J. & Mendes R., (2003), “Neighborhood Topologies In Fully-Informed And Best-of-Neighborhood Particle Swarms”, Proceedings Of The 2003 IEEE International Workshop On Soft Computing In Industrial Applications 2003 (SMCIA/03), pp.445-450

[8] Kennedy J. & Eberhart R., (1995), “Particle Swarm Optimisation”, Neural Networks, 1995. Proceedings. IEEE International Conference On , Volume: 4, 27 Nov.-1 Dec. 1995, pp.1942-1948

[9] Kennedy J., (2005), "Why Does It Need Velocity?", Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE , vol., no.pp. 38- 44, 8-10 June 2005, pp. 38- 44

[10] Lei Xie & Liying Jiang, (2005), "Global Optimal ICA And Its Application In Brain MEG Data Analysis", Neural Networks and Brain, 2005. ICNN&B '05. International Conference on , vol.1, no.pp. 353- 357, 13-15 Oct. 2005, pp. 353-357

[11] Shi Y. & Eberhart R. C., (1998), “Parameter Selection In Particle Swarm Optimization”, Evolutionary Programming VII: Proceedings Of The Seventh Annual Conference On Evolutionary Programming, New York., pp. 591-600

[12] Shi Y. & Eberhart, R., (1998), “A Modified Particle Swarm Optimizer”, Proceedings Of The IEEE Congress On Evolutionary Computation (CEC 1998), Piscataway, NJ, pp. 69-73

[13] Slade W. H. et al, (2004), "Inversion Of Ocean Color Observations Using Particle Swarm Optimization", Geoscience and Remote Sensing, IEEE Transactions on Volume 42, Issue 9, Sept. 2004, pp.1915 - 1923

[14] van den Bergh F. & Engelbrecht A. P., (2002), "A New Locally Convergent Particle Swarm Optimizer", Proceedings of IEEE International Conference on Systems, Man, and Cybernetics 2002 (SMC 2002), pp. 96-101

[15] van den Bergh F., (2002), “An Analysis Of Particle Swarm Optimizers” , Ph.D. Dissertation, Dept. Comput. Sci., Univ. Pretoria, Pretoria, South Africa, 2002.

[16] Xu Jun-Jie & Xin Zhan-Hong, (2005), "An Extended Particle Swarm Optimizer", Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International , vol., no.pp. 5 pp.-, 4-8 April 2005, pp. 5 pp.-,

[17] Yong-Ling Z. et al, (2003), "On The Convergence Analysis And Parameter Selection In Particle Swarm Optimization", Machine Learning and Cybernetics, 2003 International Conference on Volume 3, 2-5 Nov. 2003, pp.1802 – 1807