What Makes a Successful Society? Experiments with Population Topologies in Particle Swarms

Rui Mendes* and José Neves

Departamento de Informática, Universidade do Minho, Portugal

Abstract. Previous studies in Particle Swarm Optimization (PSO) have emphasized the role of population topologies in particle swarms. These studies have shown that a relationship between the way individuals in a population are organized and their aptitude to find global optima exists. A study of what graph statistics are relevant is of paramount importance. This work presents such a study, which will provide guidelines that can be used by researchers in the field of PSO in particular and in the Evolutionary Computation arena in general.

Keywords: Particle Swarm Optimization, Swarm Intelligence, Evolutionary Computation

1 Introduction

The field of Particle Swarm Optimization (PSO) is evolving fast. Since its creation in 1995 [1, 2], researchers have proposed important contributions to the paradigm in the field of parameter selection [3, 4]. Lately, the field of population topologies has also been object of study, as its importance has been demonstrated [5, 6]. The study of topologies has also triggered the development of a very successful algorithm, Fully Informed Particle Swarm (FIPS), that has demonstrated to perform better than the canonical particle swarm, widely accepted by researchers as the state-of-the-art algorithm, in a well-known benchmark of hard functions [7, 8].

Due to the fact that FIPS has demonstrated superior results and its close relationship to the structure of the population, a study to understand the relationship between the population structure and the algorithm was conducted.

2 Canonical Particle Swarm

The standard algorithm is given in some form resembling the following:

$$\begin{aligned} \boldsymbol{v}_{t+1} &= \alpha\, \boldsymbol{v}_t + \mathbf{U}[0, \varphi_1] \otimes (\boldsymbol{P}_i - \boldsymbol{X}_t) \\ &\quad + \mathbf{U}[0, \varphi_2] \otimes (\boldsymbol{P}_g - \boldsymbol{X}_t) \qquad (1) \\ \boldsymbol{X}_{t+1} &= \boldsymbol{X}_t + \boldsymbol{v}_{t+1} \qquad (2) \end{aligned}$$

* The work of Rui Mendes is sponsored by the grant POSI/ROBO/43904/2002.

A.L.C. Bazzan and S. Labidi (Eds.): SBIA 2004, LNAI 3171, pp. 346–355, 2004.

where $\otimes$ denotes point-wise vector multiplication, $\mathbf{U}[min, max]$ is a function that returns a vector whose positions are randomly generated, following the uniform distribution between min and max, α is called the inertia weight and is less than 1, $\boldsymbol{v}_t$ and $\boldsymbol{X}_t$ represent the speed and position of the particle at time t, $\boldsymbol{P}_i$ refers to the best position found by the particle, and $\boldsymbol{P}_g$ refers to the position found by the member of its neighborhood that has had the best performance so far. The Type 1″ constriction coefficient is often used [4]:

$$\begin{aligned} \boldsymbol{v}_{t+1} &= \chi \left(\boldsymbol{v}_t + \mathbf{U}[0, \varphi_1] \otimes (\boldsymbol{P}_i - \boldsymbol{X}_t) \right. \\ &\qquad \left. + \mathbf{U}[0, \varphi_2] \otimes (\boldsymbol{P}_g - \boldsymbol{X}_t) \right) && (3) \\ \boldsymbol{X}_{t+1} &= \boldsymbol{X}_t + \boldsymbol{v}_{t+1} && (4) \end{aligned}$$

The two versions are equivalent, but are simply implemented differently. The second form is used in the present investigations. Other versions exist, but all are fairly close to the models given above.

A particle searches through its neighbors in order to identify the one with the best result so far, and uses information from that source to bias its search in a promising direction. There is no assumption, however, that the best neighbor at time t actually found a better region than the second or third-best neighbors. Important information about the search space may be neglected through overemphasis on the single best neighbor.

When constriction is implemented as in the second version above, lightening the right-hand side of the velocity formula, the constriction coefficient χ is calculated from the values of the acceleration coefficient limits, φ_1 and φ_2; importantly, it is the sum of these two coefficients that determines what χ to use. This fact implies that the particle's velocity can be adjusted by any number of terms, as long as the acceleration coefficients sum to an appropriate value. For instance, the algorithm given above is often used with $\chi = 0.7298$ and $\varphi_1 = \varphi_2 = 2.05$. The φ coefficients must sum, for that value of χ, to 4.1.

3 Fully Informed Particle Swarm

The idea behind FIPS is that social influence comes from the group norm, i.e., the center of gravity of the individual's neighborhood. Contrary to canonical particle swarm, there is no individualism. That is, the particle's previous best position takes no part in the velocity update.

In the canonical particle swarm, each particle explores around a region defined by its previous best success and the success of the best particle in its neighborhood. The difference in FIPS is that the individual should gather information about the whole neighborhood. For that, let us define $\mathcal{N}$ as the set of neighbors of i and p_k as the best position found by individual k.

$$\begin{aligned} v_{t+1} &= \chi \left(v_t + \frac{\sum_{k \in \mathcal{N}} \mathrm{U}[0, \varphi_{max}](p_k - x_t)}{|\mathcal{N}|} \right) \\ x_{t+1} &= x_t + v_{t+1} \end{aligned} \qquad (5)$$

This formula is a generalization of the canonical version. In fact, if $\mathcal{N}$ is defined to contain only i itself and its best neighbor, this formula is equivalent to the one presented in equation 4. Thus, in FIPS the velocity update is performed according to a stochastically weighted average of the difference between the particle's current position and each of its neighbors' previous best.

As can be concluded from equation 5, the algorithm uses neither information about the relative quality of each of the solutions found by its neighbors nor about the particle's previous best position. The particle simply oscillates around the stochastic center of gravity of its neighbors' previous findings.

4 Population Structures and Graph Statistics

In particle swarms, individuals strive to improve themselves by imitating traits found in their successful peers. Thus, "social norms" emerge because individuals are influenced by their neighbors. The definition of the social neighborhood of an individual, i.e., which individuals influence it, is very important. As practice demonstrates, the topology that is most widely used – *gbest*, where all individuals influence one another – is vulnerable to local optima.

Social influence is dictated by the information found in the neighborhood of each individual, which is only a subset of the population. The relationship of influence is defined by a social network – represented as a graph – that we call *population topology* or *sociometry*.

The goal of sociometries is to control how soon the algorithm converges. The goal is find which aspects of the graph structure are responsible for the information "spread". It does not make sense to study topologies where there are isolated subgroups, as they would not communicate among themselves. Therefore, all graphs studied are connected, i.e., there is a path between any two vertices. Results reported by researchers confirm that PSO performs well with small populations of 20 individuals.

4.1 Degree and Distribution Sequence

Degree determines the scale of socialization: An individual without neighbors is an outsider; an individual with few neighbors cannot gather information from nor influence others in the population; an individual with many neighbors is both well informed and i possesses a large sphere of influence.

One of the most interesting measures of the spread of information seems to be the distribution sequence. In fact it can be seen as an extension of the degree. In short, this sequence, named Λ_l, gives the number of individuals that can only be reached through a path of l edges.

$\mathbf{\Lambda_1(v)}$ This is the degree of vertex v. It represents the number of individuals immediately influenced by v.

$\mathbf{\Lambda_2(v)}$ This is the number of v's neighbor's neighbors. To influence these individuals, v must influence its neighbors for a sufficiently long period of time.

$\Lambda_3(v)$ This is the number of individuals three steps away from v. To influence these individuals, v has to transitively influence its neighbors and its neighbors' neighbors.

Besides the degree, this study also investigates the effects of Λ_2. Λ_3 is not used because it is not defined on most of the graphs used.

4.2 Average Distance, Radius and Diameter

In a sparsely connected population, information takes a long time to travel. The spreading of information is an important object of study. Scientists study this effect in many different fields, from social sciences to epidemiology. A measure of this is path length. Path length presents a compromise between exploration and exploitation: If it is too small, it means that information spreads too fast, which implies a higher probability of premature convergence. If it is large, it means that information takes a long time to travel through the graph and thus the population is more resilient and not so eager to exploit earlier on.

However, robustness comes at a price: speed of convergence. It seems important to find an equilibrium. This statistic correlates highly with degree: a high degree means a low path length and vice-versa. The radius of a graph is the smallest maximal difference of a vertex to any other. The diameter of a graph is the largest distance between any two vertices.

4.3 Clustering

Clustering measures the percentage of a vertex's neighbors that are neighbors to one another. It measures the degree of "cliquishness" of a graph. Overlapping plays an important part in social networks. We move in several circles of friends. In these, almost everyone knows each other. In fact we act as bridges or shortcuts between the various circles we frequent. Clustering influences the information spread in a graph. However, its influence is more subtle. The degree of homogenization forces the cluster to follow a social norm. If most of the connections are inside the cluster; all individuals in it will tend to share their knowledge fairly quickly. Good regions discovered by one of them are quickly passed on to the other members of the group. Even a partial degree of clustering helps to disseminate information. It is easier to influence an individual if we influence most of its neighbors.

5 Parallel Coordinates and Visual Data Analysis

Parallel coordinates provide an effective representation tool to perform hyperdimensional data analysis [9]. Parallel coordinates were proposed by Inselberg [10] as a new way to represent multi-dimensional information. Since the original proposal, much subsequent work has been accomplished, e.g., [11]. In traditional Cartesian coordinates, all axes are mutually perpendicular. In parallel coordinates, all axes are parallel to one another and equally spaced. By drawing the axes parallel to one another, one can represent points, lines and planes in hyper-dimensional spaces. Points are represented by connecting the coordinates on each of the axes by a line.

Parallel coordinates are a very useful tool in visual analysis. It is very easy to identify clusters visually in high dimensional data by using color transparency. Color transparency is used to darken less clustered areas and brighten highly clustered ones. By using brushing techniques, it is possible to examine subsets of the data and to identify relationships between variables.

In this study, parallel coordinates were used to identify the graph statistics present in all highly successful population topologies. By using brushing, it is possible to identify highly successful groups and identify what characteristics are shared by all topologies belonging to them.

6 Parameter Selection and Test Procedure

The present experiments extracted two kinds of measures of performance on a standard suite of test functions. The functions were the sphere or parabolic function in 30 dimensions, Rastrigin's function in 30 dimensions, Griewank's function in 10 and 30 dimensions (the importance of the local minima is much higher in 10 dimensions, due to the product of co-sinuses, making it much harder to find the global minimum), Rosenbrock's function in 30 dimensions, Ackley's function in 30 dimensions, and Schaffer's f6, which is in 2 dimensions. Formulas can be found in the literature (e.g., in [12]).

The experiments conducted compare several conditions among themselves. A condition is an algorithm paired with a topology. To have a certain degree of precision as to the value of a certain measure pertaining to a given condition, 50 runs were performed per condition.

6.1 Mean Performance

One of the measures used is the best function result attained after a fixed number of function evaluations. This measure reports the expected performance an algorithm will have on a specific function. The mean performance is a measure of sloppy speed. It does not necessarily indicate whether the algorithm is close to the global optimum. A relatively high score can be obtained on some of these multi-modal functions simply by finding the best part of a locally optimal region.

When using many functions, results are usually presented independently on each of the functions used and there is no methodology to conclude which of the approaches has a good performance over all the functions. However, this considerably complicates the task of evaluating which approach is the best. It is not possible to combine raw results from different functions, as they are all scaled differently. To provide an easier way of combining the results from different functions, uniform fitness is used, instead of raw fitness. A uniform fitness can simply be regarded as a proportion: a uniform fitness of less than 0.1 can be interpreted as being one of the top 10% solutions. In this study, the number of iterations elapsed before performance is recorded is of 1,000.

6.2 Proportion of Successes

While the measure of mean performance gives an indication of the quality of the solution found, an algorithm can achieve a good result while getting stuck in a local optimum. The proportion of successes shows the percentage of times that the algorithm was able to reach the globally optimal region. The proportion of successes validates the results of the average performance. It may be possible for good results to be achieved by combining an extremely good result in a function (e.g. the Sphere, with an average result in a more difficult function). The algorithm is left to run until 3,000 iterations have elapsed and then its success is recorded.

6.3 Parameter Selection

As the goal of this study is to verify the impact of the choice of social topologies in the behavior of the algorithm, the tuning parameters are fixed. They are set to the values that are widely used by the community and that are deemed to be the most appropriate ones, as demonstrated in [4]. The value of φ was set to 4.1, which is one of the most used in the community of particle swarms. This value is split equally between φ_1 and φ_2. The value of χ was set to 0.729. All the population topologies used in this study comprise 20 individuals.

6.4 Topology Generation

The graphs representing the social topologies were generated according to a given set of constraints. These were representative of several parameters deemed important in the graph structure. Preliminary studies of the graph statistics indicated that by manipulating the average degree and average clustering, along with the corresponding standard deviations, it was possible to manipulate the other statistics over the entire range of possible values. These parameters were used to create a database of graphs with average degrees ranging from 3 to 10 and clustering from 0 to 1. A database of graph statistics of these topologies was collected, to be used in the analysis. The total number of population topologies used amounts to 3,289.

7 Analysis of the Results

The results obtained are analyzed visually using Parvis, a tool for parallel coordinates visualization of multidimensional data sets. To allow for an easier interpretation of the figures, the name of each of the axes is explained:

Alg 1 for Canonical Particle Swarm, 2 for FIPS.
Prop Proportion of successes.
Perf Average performance.
Degree Average degree of the population topology.
ClusteringCoefficient Clustering coefficient of the population topology.

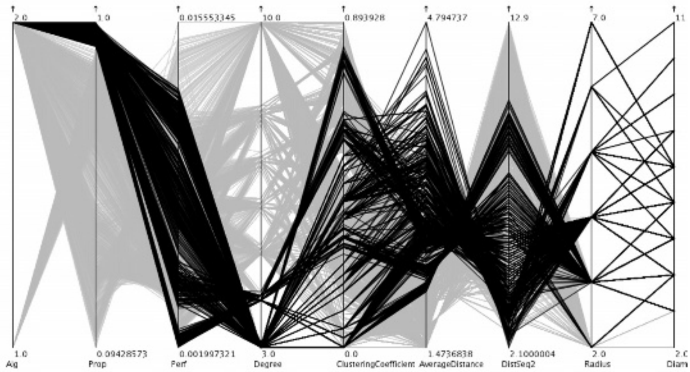

Fig. 1. Experiments with a proportion of successes higher than 93%. All the experiments belong to the FIPS algorithm.

AverageDistance Average distance between two nodes in the graph.
DistSeq2 The distribution sequence of order 2.
Radius The radius of the graph.
Diameter The diameter of the graph.

First, the experiments responsible for a proportion of successes higher than 93% are isolated (Figure 1). All the results belong to the FIPS algorithm. None of the canonical experiments was this successful. However, some of the experiments have low quality average performance. The next step is to isolate the topologies with both a high proportion of successes and a high quality average performance (Figure 2). Fortunately, all of these have some characteristics in common:

- the average degree is always 4;
- the clustering coefficient is low;
- the average distance is always similar.

As most of the graph statistics are related to some degree, it seems interesting to display the graph statistics of all graphs with degree 4 (Figure 3). This shows that the average distance is similar for graphs with a somewhat low clustering coefficient. Thus, it makes sense to concentrate the efforts in just the average degree and clustering coefficient.

Figure 4 shows the experiments of FIPS, using topologies with average degree 4 and clustering lower than $0,5$. This figure is similar to Figure 2. As a further exercise, Figure 5 shows what happens when the clustering is restricted to values lower than $0,0075$. This set identifies very high quality solutions, according to both measures.

8 Conclusions and Further Work

This study corroborates the results reported in [7, 8] that FIPS shows superior results to the ones of the canonical particle swarm. It showed that the successful

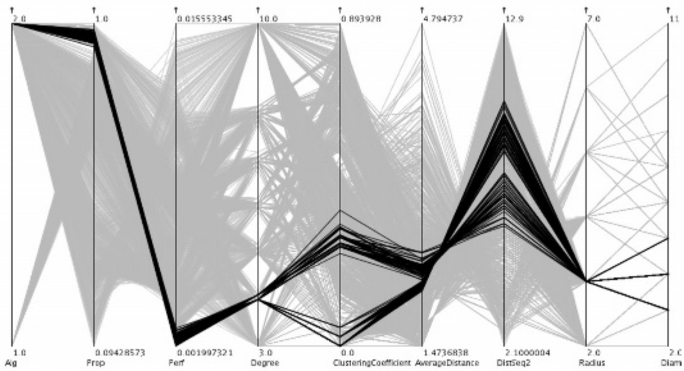

Fig. 2. Experiments with a high proportion of successes and a high quality average performance. The following conclusions can be drawn: the average degree is always 4; the clustering coefficient is low; the average distance is always similar.

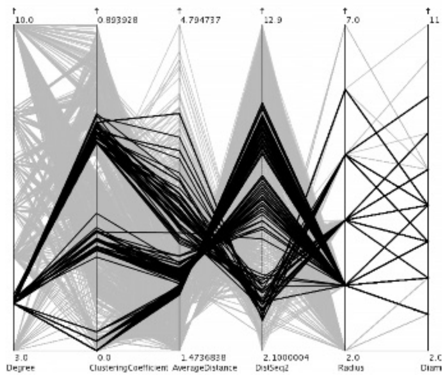

Fig. 3. Graph statistics of all topologies with average degree 4.

topologies had an average of four neighbors. This result can be easily rationalized: The use of more particles triggers the possibility of crosstalk effects encountered in neural network learning algorithms. In other words, the pulls experienced in the directions of multiple particles will mostly cancel each other and reduce the possible benefits of considering their knowledge.

Parallel coordinates proved to be a powerful tool to analyze the results. The capabilities of the tool used allowed for a very straightforward test of different hypothesis. The visual analysis of the results was able to find a set of graph statistics that explains what makes a good social topology.

To validate the conjectures concluded by this work, a large number of graphs with the characteristics found should be generated and tested to see if all the graphs in the set have similar characteristics when interpreted as a population topology. Further tests with other problems should also be performed, especially with real-life problems, to validate the results found.

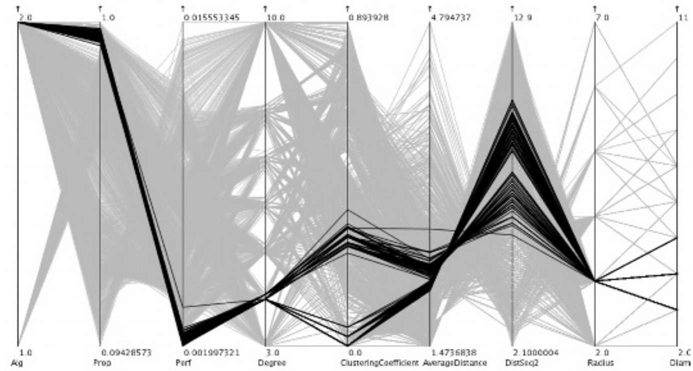

Fig. 4. Experiments of FIPS with topologies with average degree 4 and clustering lower than 0, 5.

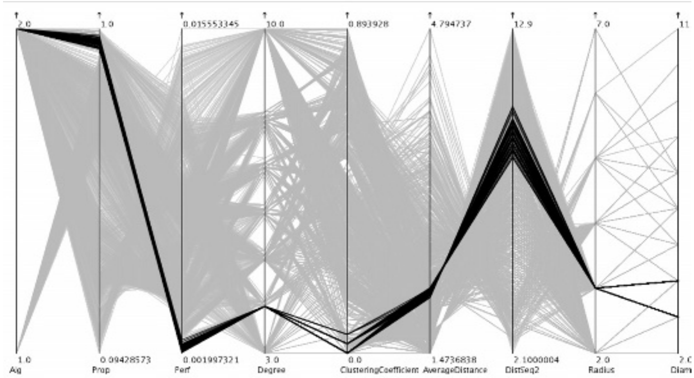

Fig. 5. Experiments of FIPS with topologies with average degree 4 and clustering lower than 0, 0075.

References

1. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, IEEE Service Center (1995) 39–43
2. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN'95 - International Conference on Neural Networks. Volume 4., Perth, Western Australia (1995) 1942–1948
3. Shi, Y., Eberhart, R.C.: Parameter selection in particle swarm optimization. In Porto, V.W., Saravanan, N., Waagen, D., Eiben, A.E., eds.: Evolutionary Programming VII, Berlin, Springer (1998) 591–600 Lecture Notes in Computer Science 1447.
4. Clerc, M., Kennedy, J.: The particle swarm: Explosion, stability, and convergence in a multi-dimensional complex space. IEEE Transactions on Evolutionary Computation **6** (2002) 58–73

5. Kennedy, J.: Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance. In: Proceedings of the 1999 Conference on Evolutionary Computation, IEEE Computer Society (1999) 1931–1938
6. Kennedy, J., Mendes, R.: Topological structure and particle swarm performance. In Fogel, D.B., Yao, X., Greenwood, G., Iba, H., Marrow, P., Shackleton, M., eds.: Proceedings of the Fourth Congress on Evolutionary Computation (CEC-2002), Honolulu, Hawaii, IEEE Computer Society (2002)
7. Mendes, R., Kennedy, J., Neves, J.: Watch thy neighbor or how the swarm can learn from its environment. In: Proceedings of the Swarm Intelligence Symposium (SIS-2003), Indianapolis, IN, Purdue School of Engineering and Technology, IUPUI, IEEE Computer Society (2003)
8. Mendes, R., Kennedy, J., Neves, J.: The fully informed particle swarm: Simpler, maybe better. IEEE Transactions of Evolutionary Computation (in press 2004)
9. Wegman, E.: Hyperdimensional data analysis using parallel coordinates. Journal of the American Statistical Association **85** (1990) 664–675
10. Inselberg, A.: n-dimensional graphics, part I–lines and hyperplanes. Technical Report G320-2711, IBM Los Angeles Scientific Center, IBM Scientific Center, 9045 Lincoln Boulevard, Los Angeles (CA), 900435 (1981)
11. Inselberg, A.: The plane with parallel coordinates. The Visual Computer **1** (1985) 69–91
12. Reynolds, R.G., Chung, C.: Knowledge-based self-adaptation in evolutionary programming using cultural algorithms. In: Proceedings of IEEE International Conference on Evolutionary Computation (ICEC'97). (1997) 71–76