Population Structure and Particle Swarm Performance

James Kennedy
Bureau of Labor Statistics
Washington, DC
Kennedy_Jim@bls.gov

Rui Mendes
Universidade do Minho
Braga, Portugal
rui@omega.di.uminho.pt

Abstract: The effects of various population topologies on the particle swarm algorithm were systematically investigated. Random graphs were generated to specifications, and their performance on several criteria was compared. What makes a good population structure? We discovered that previous assumptions may not have been correct.

I. INTRODUCTION

The trajectories of individual members of a particle swarm population have been analyzed in depth [3][[4][8], and those analyses have resulted in improvements in the performance of the algorithm. It has long been clear though that the uniqueness of the algorithm lies in the dynamic interactions of the particles. Even when changes are made to the formulas e.g., [1][2][4][8], the performance depends on an effect involving the entire population.

The particle swarm algorithm can be described generally as a population of vectors whose trajectories oscillate around a region which is defined by each individual's previous best success and the success of some other particle. Various methods have been used to identify "some other particle" to influence the individual. The two most commonly used methods are known as *gbest* and *lbest* (Figure 1). In the *gbest* population, the trajectory of each particle's search is influenced by the best point found by any member of the entire population. The *lbest* population allows each individual to be influenced by some smaller number of adjacent members of the population array. Typically *lbest* neighborhoods comprise exactly two neighbors, one on each side: a ring lattice.

A kind of lore has evolved regarding these sociometric structures. It has been thought that the *gbest* type converges quickly on problem solutions but has a weakness for becoming trapped in local optima, while *lbest* populations are able to "flow around" local optima, as subpopulations explore different regions [7]. The lore is based on experience and some data, but population topologies have not been systematically explored. The present research manipulates some sociometric variables that are hypothesized to affect performance.

There is not room in this forum to discuss the No Free Lunch implications of the present strategy. We are convinced that it is worthwhile to seek an optimization algorithm that performs well on a variety of standard test functions, even if it is average across the full range of possible functions.

II. CAUSAL FACTORS

The present study focused on population topologies where connections were undirected, unweighted, and did not vary over the course of a trial. The usual particle swarm rule was used, that an individual gravitated toward a stochastically weighted average of its own previous best point and the best point found by any member of its neighborhood.

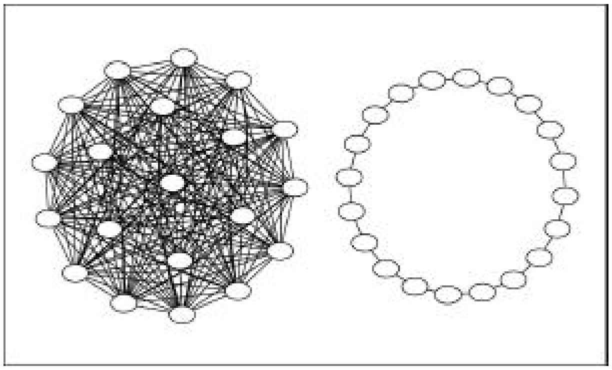

FIGURE 1. GBEST (LEFT) AND LBEST SOCIOMETRIC PATTERNS[i].

Watts [10][11] has shown that the flow of information through social networks is affected by several aspects of the networks. The first measure is the degree of connectivity among nodes in the net. Each individual in a particle swarm identifies the best point found by its *k* neighbors; *k*, then, is the variable that distinguishes *lbest* from *gbest* topologies, and is likely to affect performance.

A second factor identified by Watts was the amount of clustering, *C*. Clustering occurs when a node's neighbors are also neighbors to one another. The number of neighbors-in-common can be counted per node, and can be averaged over the graph.

Finally, Watts noted that the average shortest distance from one node to another was an important graph characteristic for determining the spread of information through the network. The present research did not manipulate this variable, which correlates very highly with both *k* and *C*.

Previous investigation within the particle swarm paradigm had found that the effect of population topology interacted with the function being optimized [6]. Some kinds of populations worked well on some functions, while other kinds worked better on other functions. Of course it would be best to find a methodology that worked well on a wide range of problems. Kennedy [6] theorized that populations with fewer connections might perform better on highly multimodal problems, while highly interconnected populations would be better for unimodal problems.

We hypothesized then that heterogeneous population structures, with some subsets of the population tightly connected and others relatively isolated, might provide the benefits of both *lbest* and *gbest* sociometries. It was noted that the heterogeneity could be of two types. Variance could be introduced into *k*, the number of neighbors for each node in the population, or it could be introduced into *C*, the number of neighbors in common. If *k* has a high variance, then some nodes will have numerous neighbors, while others have fewer. (In all cases, graphs are closed, meaning that at least one path exists from any node to any other). Variance in *C* means that cliques will be found in some parts of the population, where neighbors' neighbors are neighbors, while other parts of the population will be relatively isolated.

III. DEPENDENT VARIABLES

Five standard test functions were employed in the present research. These were the Sphere function, Rastrigin, Griewank, Rosenbrock, and Shaffer's f6 [9]. All except f6 were implemented in 30 dimensions; f6 is a two-dimensional problem.

There are two major measures of performance in an optimizer such as the particle swarm. The first is the best function result attained after some number of iterations. The present paper reports the best result found after 1,000 iterations of the algorithm.

It is possible however for the algorithm to rapidly attain a relatively good result while becoming trapped on a local optimum. Thus a second dependent measure used in the present investigation was the number of iterations required for the algorithm to meet a criterion. The criteria are given in Table 1. The algorithm was run for 10,000 iterations or until the criterion was met. If it was not met by that time, the measure was considered infinite, that is, it was reported as if the criterion would never be met. Thus medians rather than means are reported in iteration results.

A third dependent measure was derived from the second. That is a simple binary variable describing whether the version attains the criterion or not.

Table 1. Parameters and criteria for the five test functions.

Function	Dimensions	Initial range	Criterion
Sphere	30	±100	0.01
Rastrigin	30	±5.12	100
Griewank	30	±600	0.05
Rosenbrock	30	±30	100
f6	2	±100	0.00001

The performance at 1,000 iterations was standardized within functions for analysis, that is, data were linearly transformed so that results for each function had a mean of 0.0 and a standard deviation of 1.0. This simplified the statistical analysis by allowing the averaging of results from the five functions, rather than performing multivariate analyses. Since all functions were minimized, a negative standardized average means that a trial performed better than the mean. These are referred to below as "standardized performance results."

IV. RANDOM GRAPHS

The first experiments implemented the particle swarm algorithm in graphs that were randomly generated and optimized to meet some desired criteria. An optimization method inspired by simulated annealing, using a cooling mechanism where the temperature decreases exponentially, was used to alter a randomly initialized set of connections among a population of 20 nodes. Care was taken that each graph was unique. Each experimental condition was defined by a combination of *k*, *C*, standard deviation of *k* (*sdk*), and standard deviation of *C* (*sdC*). The standard deviation conditions were classified as high or low, with a threshold at 1.5.

Some configurations are impossible to create, for instance, it is not possible to create a closed graph with very low average *k* and high variance in *k*. Further, when sociometries are highly connected, it may not be possible to generate a sufficient number of unique graphs. The research design reflects these constraints, for instance there are many more graphs with *k*=10 than with *k*=3. While this affected the kinds of analyses we could perform, we decided it was more important to keep all the graphs that were created than to have trials equally assigned to conditions. 1,343 graphs were generated with the combinations of characteristics shown in the Appendix, plus 20 graphs with *k*=4.

Particle swarms with Type 1" constriction [3] were implemented. The form of the algorithm was:

$$\vec{v}_i = c\,(\vec{v}_i + U(0,1)j\,(\vec{p}_i - \vec{x}_i) + U(0,1)j\,(\vec{p}_g - \vec{x}_i))$$

$$\vec{x}_i = \vec{x}_i + \vec{v}_i$$

where $j = 2.01$ and χ=0.729844.

Results for all experimental conditions are shown in the Appendix. That table collapses together the results from conditions where the self is included in the neighborhood and where it is not. This factor was included in an analysis of variance of main effects. Other factors were the average *k*, *C*, *sdk*, and *sdC* for a graph. Dependent measures were standardized performance at 1,000 iterations averaged over the five functions (Perf.), the averaged binary measure of whether the criteria were met (Prop.), and the rank of the average number of iterations required to meet the criterion (Iter.). Three ANOVAs were run, one for each dependent measure.

The main effect for self/no-self was not significant for the performance measure or iterations required, but was for the binary variable, $F(12,2673)= 7.94$, $p< 0.0049$. The no-self conditions were significantly more likely to meet the criteria.

The only other significant effect was the effect of k on all three dependent measures. As can be seen in Table 2, the k=5 conditions had the best values at 1,000 iterations (Stand. Perf.), and required the fewest iterations to meet the criteria (Iter.), while k=3 had the highest success rate (Prop.).

Table 2. The three measures aggregated by levels of connectivity, *k*.

		Stand.		
k	**n**	**Perf.**	**Prop.**	**Iter.**
3	462	0.197	0.959	739.5
5	622	-0.055	0.949	670.6
10	1602	-0.035	0.912	792.8

The three dependent measures covaried reliably, as seen in Table 3.

Table 3. Spearman correlations among the three dependent measures.

	Prop.	**Iters.**
Perf.	-0.497	0.695
Prop.	.	-0.810

The proportion meeting the criterion correlates negatively with the other two measures because smaller values, that is, smaller minimization results and fewer iterations, are better for them. Thus Table 3 suggests that trials that performed well by one standard tended to perform well by the others, too.

V. SPECIAL GRAPHS

As noted above, the standard particle swarm configurations are not random graphs, but regular structures, versions of ring lattices. *Lbest* and *gbest* are but two possible ways to structure the population regularly. This section of the paper describes experiments with some topologies that were designed by the researchers.

Six sociometries were studied; four of these were implemented both with and without the self included, for a total of ten. These were

- gbest: which treated the entire population as the individual's neighborhood.
- lbest: where adjacent members of the population array comprised the neighborhood.
- pyramid: a three-dimensional wire-frame triangle.
- star: one central node influenced, and was influenced by, all other members of the population.
- "small:" a graph created with cliques and isolates, as an example of heterogeneity.
- von Neumann: neighbors above, below, and on each side on a two-dimensional lattice were connected.

The ten special sociometries were added to the random-graph data set, so their performance could be compared to the others. Conditions with fewer than 20 observations were removed to increase validity. Performance at 1,000 iterations was standardized, as before, and populations were grouped on the basis of the four independent variables, with the new graphs added as separate groups. The resulting dataset had 70 groups of sociometries to compare, with from 20 to 130 members in each group.

Some of the special graphs produced outstanding results – two were outstandingly good, and two outstandingly bad. A rough measure of goodness was created by ranking all groups on each of the three measures, performance, proportion, and iterations, and then taking the average rank.

The von Neumann neighborhood with self included ranked second in performance at 0.173 standard deviations below the mean, and third in proportion meeting the criterion, with 0.98 of trials meeting the criteria. It also ranked 37th out of 70 in terms of number of iterations to the criterion. Note that some configurations scoring well on this measure actually failed to meet the criteria a lot of time; fast to fail is not, in our eyes, as good as slow to succeed.

The von Neumann neighborhoods without self also did relatively well: they were fourth overall in performance, third in proportion, and 53rd in iterations.

Another special sociometry, called the "pyramid," was relatively good, too, ranking third in performance, eleventh in proportion, and 45th in iterations. It was ranked ninth overall by the combined measure.

The worst two graphs in the entire dataset were from the special subset. Next-to-worst, ranking 69th in all three measures, was the "star" configuration. Its function performance was a full 1.396 standard deviations above the mean. The star is the centralized topology where all information passes through one individual. There is some irony here, in that the star is the social configuration most used in business, government, and military organizations.

The absolute loser in the population was the *gbest* configuration without the self included. It met the criteria 0.80 of the time, and was dead last in that and the number of iterations to criteria. The selfless *gbest* rated 68th place in terms of its performance at 1,000 iterations, which was 0.234 standard deviations above the mean.

It is certainly worth noting that the *lbest* populations with self, one of the most common particle swarm topologies, ranked 64th out of 70 by the combined measure. They were slow and inaccurate. When the self was removed they performed somewhat better, earning a 61st place ranking.

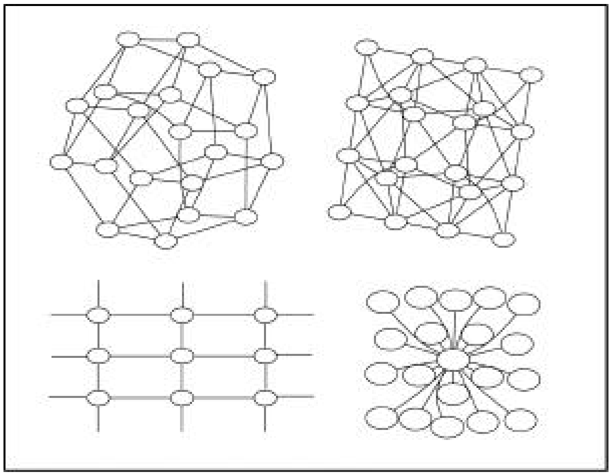

FIGURE 2. THE VON NEUMANN NEIGHBORHOOD IS SHOWN WRAPPED (TOP LEFT) AND ONE SECTION IS FLATTENED OUT (LOWER LEFT). THE PYRAMID SOCIOMETRY IS AT THE TOP RIGHT, AND THE STAR AT LOWER RIGHT.

The lore mentioned above was somewhat borne out. The *gbest* version with self was actually ranked second in the population for number of iterations to the criteria, requiring a median of only 404.8 iterations. Unfortunately it met the criteria only 0.85 of the time. The *lbest* version needed 755.5 iterations, but met the standard 0.94 of the time. Neither of these approaches seemed especially good in comparison.

VI. SELECTING WINNERS

We had now built up a data set of results from 2,777 particle swarm trials on each of the test functions. The special populations were run twenty times each, but the random graphs were run once each and grouped by characteristics.

For the next stage of the study, we ranked each of the random graphs based on its one trial in self and no-self conditions. Then we chose several hundred of the top performers and ran each of them twenty times on the five functions, to learn whether their excellent performance would be consistent, or was simply due to chance.

The 306 “best” graphs were run 20 times each. Three measures were used for each configuration corresponding to those taken in the previous experiment.

It was found that aggregated scores correlated very little with the individual scores for the same graphs, as measured in the previous experiment. Inspection of those previous results showed that a large proportion of the top performers in the individual trials were from versions with *k*=10, but none of these were among the top performers when scores were aggregated. The explanation for this is that topologies with a large number of connections have the potential to converge rapidly if they are initialized in a good region. If however they start out badly, they will quickly converge to an inferior solution, resulting in a mediocre average.

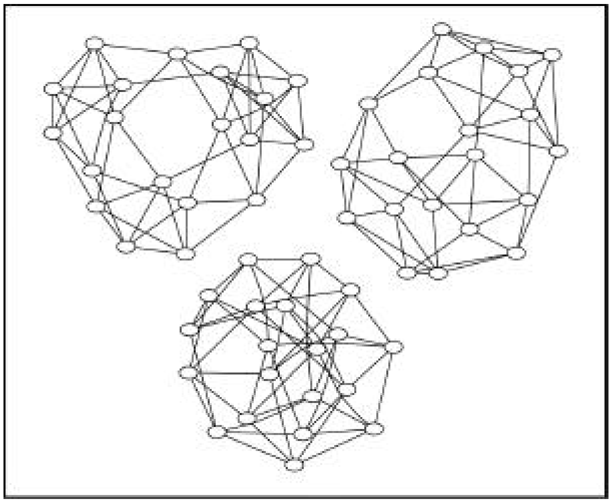

FIGURE 3. THESE SOCIOMETRIES MET THE CRITERIA 100 PER CENT OF THE TIME.

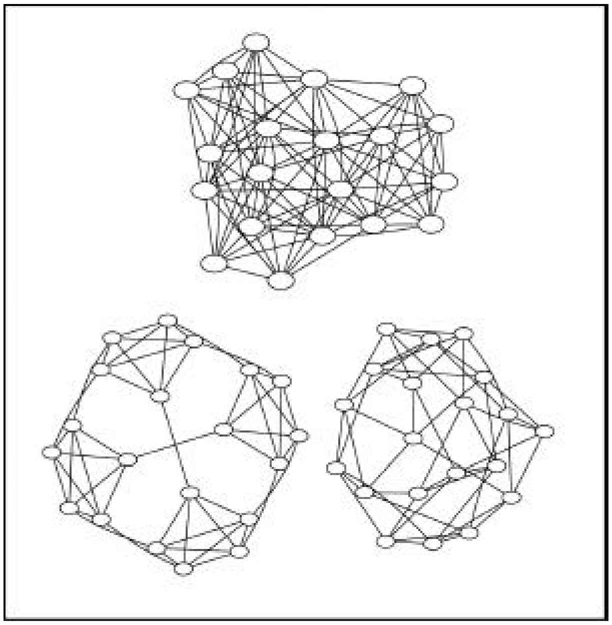

FIGURE 4. THE THREE BEST POPULATIONS, RANKED ON A SCORE BASED ON PROPORTION AND PERFORMANCE.

The conclusion of this study depends on which dependent measure one consults. When groups were ranked by the proportion meeting criteria, sixteen of the best twenty graphs had *k*=5, with three tens and two fours. When they were ranked by performance at 1,000 iterations, 12 of the top twenty had *k*=5. But when ranking by the number of iterations needed to meet the criteria, nineteen of the twenty best samples had *k*=10 – and the remaining one had *k*=19, the *gbest* version. Thus, as expected, greater connectivity speeds up convergence, though it does not tend to improve the population's ability to discover global optima.

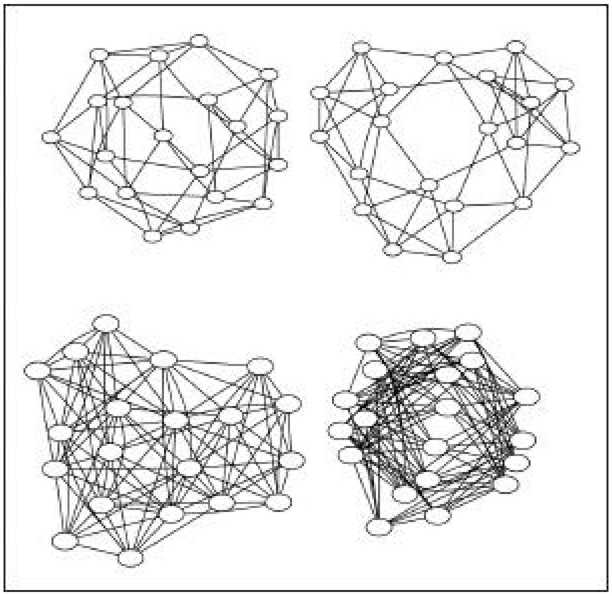

FIGURE 5. THE FOUR BEST-PERFORMING POPULATIONS AT 1,000 ITERATIONS.

The von Neumann neighborhoods fared well in this comparison. They ranked 7th in the proportion of trials meeting the criteria (0.98), and 15th in performance, at more than 0.12 standard deviations below the mean.

VII. SUMMARY

We conceptualize influence within a particle swarm population as flowing information which moves fastest between connected pairs of individuals, but is buffered or slowed by the presence of intermediaries. Thus, if individual *i* finds a good solution, this may be passed to its adjacent neighbor *j*, but not immediately to *k*, which is not connected to *i*. If the solution is indeed a good one, though, *j*'s performance will improve, until *j* (which is connected to *k*) is the best in *k*'s neighborhood. When this happens, the solution found initially by *i* may be communicated to *k*.

The traditional particle swarm topology known as "*gbest*" instantiates the most immediate communication possible; all particles are directly connected to the best solution in the population. On the other hand, the ring lattice known as "*lbest*" is the slowest, most indirect communication pattern. Where *i* is opposite *z* on the lattice, a good solution found by *i* has to pass through *i*'s immediate neighbor, that particle's immediate neighbor, and so on, until it reaches *z*. Thus a solution found by *i* moves very slowly around the ring.

As usual, the issue is the "optimal allocation of trials" [5]. When distances between nodes are too short, and communication passes too quickly, there is a tendency for the population to move rapidly toward the best solution found in the early iterations. On complex function landscapes, this is not necessarily a good thing; the population will fail to explore outside of locally optimal regions. On the other hand, inhibiting communication too much results in inefficient allocation of trials, as individual particles wander cluelessly through the search space.

The research presented here has identified some superior population configurations, but has not precisely named the topological factors that result in best performance on a range of functions. We have eliminated some bad solutions, and raised doubts about the sociometries that have been most widely used since particle swarms were first introduced in 1995.

Our recommendation for the present is that particle swarm researchers try the von Neumann configuration, which performed more consistently in our experiments than the topologies commonly found in current practice. Research not reported here supports the recommendation.

REFERENCES

[1] P. J. Angeline, Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences. *Evolutionary Programming VII*, 601-610, 1998.

[2] A. Carlisle and G. Dozier, An off-the-shelf PSO. In *Proceedings of the Particle Swarm Optimization Workshop*, Indianapolis, IN, April, 2001.

[3] M. Clerc and J. Kennedy, The Particle Swarm: Explosion, Stability, and Convergence in a Multi-Dimensional Complex Space. *IEEE Transactions on Evolutionary Computation*, in press.

[4] Y. Shi and R. Eberhart. (1998). Parameter Selection in Particle Swarm Optimization. *Proceedings of the Seventh Annual Conf. on Evolutionary Programming*, March 1998.

[5] J. Holland, *Adaptation in Natural and Artificial Systems*, MIT Press, 1975/1992.

[6] J. Kennedy, Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance. *Proceedings of the 1999 Conference on Evolutionary Computation*, 1931-1938.

[7] J. Kennedy and R. Eberhart, *Swarm Intelligence*, Morgan Kaufmann Academic Press, 2001.

[8] E. Ozcan and C. K. Mohan, Particle Swarm Optimization: Surfing the Waves, *Proceedings of the 1999 Congress on Evolutionary Computation*, Washington, DC, USA, July 6-9.

[9] R. G. Reynolds and C. Chung, "Knowledge-based self-adaptation in evolutionary programming using cultural algorithms," in the *Proceedings of IEEE International Conference on Evolutionary Computation (ICEC'97)*, 1997, pp. 71-76

[10] D. J. Watts, *Small Worlds: The Dynamics of Networks Between Order and Randomness*, Princeton University Press, 1999.

[11] D. J. Watts and S. H. Strogatz (1998). Collective dynamics of 'small-world' networks. *Nature, 393*, 440-442.

APPENDIX

Random topologies implemented in the experiments and their results on three measures.

k	C	SDk	SDC	n	Stand. Perf.	Prop.	Iter.
3	0	H	L	40	0.263	0.97	683.9
3	0	L	L	164	0.056	0.976	691.3
3	1	H	L	80	0.194	0.930	871.4
3	1	L	L	178	0.313	0.954	785.9
5	0	H	L	40	-0.101	0.960	647.5
5	0	L	L	78	-0.103	0.956	633
5	1	H	L	40	-0.005	0.950	692
5	1	L	H	66	-0.077	0.945	607.1
5	1	L	L	150	-0.106	0.948	569.7
5	2	H	L	40	-0.05	0.955	753.4
5	2	L	H	40	0.008	0.945	822.9
5	2	L	L	40	-0.114	0.960	618.1
5	3	H	L	36	0.073	0.928	912.8
5	3	L	H	2	0.159	1.000	1789
5	3	L	L	90	0.027	0.942	842.9
10	0	L	L	42	0.022	0.890	1654.5
10	1	H	H	24	-0.146	0.917	731.1
10	1	H	L	44	-0.089	0.936	529.3
10	1	L	L	80	-0.040	0.902	1077.5
10	2	H	H	258	-0.034	0.902	1104.8
10	2	H	L	22	-0.118	0.900	1314
10	2	L	H	44	0.013	0.886	∞
10	2	L	L	100	-0.078	0.912	674.9
10	3	H	L	120	-0.056	0.912	706.8
10	3	L	L	124	-0.097	0.93871	574.8
10	4	H	L	40	-0.016	0.905	814.6
10	4	L	H	44	0.199	0.90909	1111.7
10	4	L	L	260	-0.048	0.90923	941.5
10	6	H	L	42	-0.009	0.91429	1239.3
10	6	L	L	26	-0.006	0.92308	783.3
10	7	H	H	44	-0.047	0.91364	741.7
10	7	H	L	74	-0.033	0.91892	761.3
10	7	L	L	94	-0.011	0.92766	706.8
10	8	H	H	76	-0.012	0.91579	647.5
10	8	H	L	44	0.037	0.91818	643

Note: "Stand. Perf."= mean standardized best function result after 1,000 iterations; "Prop."= average proportion meeting criteria by 10,000 iterations; "Iter."= median number of iterations required to meet the criteria.

[1] Graphs were made with AT&T open-source software called "neato," available from http://www.research.att.com/sw/tools/graphviz/