Proceedings of the International Joint Conference on Information Sciences, pp. 1557-1560, 2003

Dynamic Sociometry in Particle Swarm Optimization

Mark Richards and Dan Ventura
Computer Science Department
Brigham Young University
{mdr,ventura}@cs.byu.edu

Abstract. The performance of Particle Swarm Optimization is greatly affected by the size and sociometry of the swarm. This research proposes a dynamic sociometry, which is shown to be more effective on some problems than the standard star and ring sociometries. The performance of various combinations of swarm size and sociometry on six different test functions is qualitatively analyzed.

Introduction

Particle Swarm Optimization (PSO) is a relatively new computational learning algorithm, first introduced by James Kennedy and Russell Eberhart in 1995 [4, 8, 9]. It bears some resemblance to evolutionary computation [1]. The goal of PSO is to find the global optimum of some multidimensional (usually nonlinear) function. The algorithm has proven effective in solving many problems [3, 5, 6, 8].

In PSO, the search through the problem space can be thought of as the flight of a swarm of particles (points in the space). The goal is to have the particles converge on the optimum of the function, much like a flock of birds converges on some destination. The particles are initially distributed randomly through the problem space and given an initial velocity. Each particle keeps track of its location and fitness (the value of the function being optimized), as well as the best position (and corresponding fitness) it has encountered so far in its flight. Over time, the velocity of each particle is adjusted so that it moves stochastically toward its own best position and the best position found by another particle in its neighborhood. A particle's neighborhood is the subset of particles in the swarm with which it has direct communication. This network of connections between all of the particles is known as the *sociometry*, or topology of the swarm. The algorithm stops when some criterion is met—perhaps after a certain number of iterations, or when many iterations pass without significant improvement.

When PSO is applied to real-life problems, the function evaluations themselves are the most expensive part of the algorithm. Therefore, when comparing two PSO variations, it is helpful if they have both used the same number of function evaluations. In the experiments that follow, the swarm is allotted a certain number of function evaluations and terminates when that number is reached. Pseudocode for the algorithm is given in figure 1.

```
for each particle p_i in swarm S
   initialize position x_i, velocity v_i
   and neighborhood N_i

do
   for i = 1 to size of swarm
      n_b ← argmax_{p_j ∈ N_i} (bestFitnessValue(p_j))
      n_x ← bestFitnessLocation(n_b)
      b_x ← bestFitnessLocation(p_i)
      for d = 1 to numDimensions
         φ1, φ2 ← uniform random numbers ∈ [0,2]
         v_id ← χ [v_id + φ1 (x_id – b_xd) + φ2(x_id – n_xd)]
         x_id ← x_id + v_id
      next d
      update fitness, best fitness
   next i
until termination criterion met

χ = 0.729844 and is used to keep velocities from
exploding [2].
```

Figure 1. The basic Particle Swarm Optimization algorithm.

As with many other optimization algorithms, the user must define some parameters. One of these is the number of particles in the swarm. Kennedy and Eberhart have noted that PSO seems to work well with a smaller population than is generally used in genetic algorithms [8]. Most implementations of PSO have used a swarm size of 20. Here, we explore the effects of different population sizes on swarm performance.

Another parameter that the user must specify is the sociometry of the swarm network. Two popular sociometries are known as the ring and the star. In the ring sociometry, each particle p_i is connected to p_{i-1} and p_{i+1}. This topology tends to allow for broader exploration of the problem space. When one particle finds a promising region, only its immediate neighbors will initially be drawn to that area. No other particles in the swarm will know about that region unless their own immediate neighbors move there.

In the star sociometry, every particle is connected to every other particle. If one particle finds a superior region in the search space, all other members of the swarm are immediately drawn to it. As a result,

the swarm generally converges more quickly but sometimes to a suboptimal point in the space.

Kennedy and Mendes have explored several topologies, including random connections, and a "wheel" (where there is one central particle to which all others are connected and no other links) [10]. They have also used swarms with clusters. In these sociometries, the swarm is divided into three or four subgroups. Particles are connected to every other particle in their subgroup, but there are only a few connections between the subgroups. This is analogous to the "tribe" approach that is sometimes used in genetic algorithms. In all cases, the sociometry has been specified at the time of initialization and has been static through the run of the algorithm.

Dynamic Sociometry

When exploring large problem spaces, optimization algorithms must effectively balance exploration and exploitation. Generally, it is wise to first make a broad survey of the space, and then focus effort on the regions of the space that look most promising. This has motivated the dynamic sociometry for PSO. The swarm is initialized with a ring-type sociometry. Each particle is connected to just one other member of the swarm. Over time, additional links are added. Eventually, the network is fully connected in a star sociometry. The strategy implemented here is to have the swarm fully connected after 4/5ths of the allotted function evaluations have been used. Before that time, one new connection is added to each particle at regular intervals. In this implementation, connections are not symmetric.

For example, suppose a swarm has 12 particles and is allotted 9600 function evaluations. Each particle p_i is initially connected to p_{i+1} (with p_{11} connected to p_0). The swarm gradually adds connections, so that after 9600*0.8 = 7680 function evaluations have been made, all of the particles are inter-connected. After the initial ring setup, each particle must be connected to 10 more particles, so one new connection is added after each 768 evaluations (64 iterations of the algorithm on a 12-particle swarm). After iteration 64, each p_i is connected to p_{i+2}; after 128 iterations, each p_i is connected to p_{i+3}; and so forth.

Experiments and Results

Experiments were run on six test functions. The dynamic sociometry was compared to the ring and star sociometries. Experiments were also run to test the effect of different swarm sizes. Population sizes of 5, 10, 15, 20, 25, 30, 40, 50, and 60 were used. For each combination of problem, size, and sociometry, the median performance over 200 runs is reported. All problems were run with 30 dimensions. Table 1 shows the definition of each function, its global minimum, and the range of values (along each dimension) used for the initialization of the particles' positions. Table 2 shows a snapshot of each function in two dimensions. These graphs give some indication of the topographical features of each problem, although the "noise" in the problems is not visible at this resolution. In all experiments, the particles were initialized with zero velocity. The algorithm made 60,000 evaluations on each run. Thus, for a swarm size of five, there were 12,000 iterations; but with a population of 60, there were only 1,000 iterations. Results of the experiments are shown in Table 3. Some qualitative analysis follows.

De Jong and Sphere. These functions are similar. The sphere function is the same on all axes, while De Jong becomes steeper in each dimension. Both are smooth, and the global optimum in both cases can be found simply by following the gradient. The results for these two functions were, not surprisingly, quite similar. On both problems, the most superior performance was achieved by the ring sociometry on a ten-particle swarm. Performance improved drastically when the population size increased from five to ten, but then deteriorated rapidly as the population continued to grow. The poor performance of the five-member swarms may be due to the fact that with such a small population, the probability of one of the particles stumbling across a promising region is small. The degradation of performance observed as the population size increases is likely because of the smaller number of iterations that are allowed. There is not enough time to track the gradient.

Rosenbrock. The same significant improvement is observed as the size of the swarm increases from five to ten. However, the degradation of performance as swarm size increases from 10 to 60 is much less severe (less than one order of magnitude). There does not appear to be much variability between the three sociometries.

Rastrigin. This function has hundreds of steep local optima. In contrast to the sphere and de Jong functions, performance actually increases as the size of the population is increased. With the larger population, there is a better chance that one of the particles will stumble into the best "valley," or at least a very good one. The dynamic sociometry performs much better than the other two at all population sizes. The strategy used in the dynamic sociometry to balance exploration and exploitation appears to pay off in this case.

Griewank. At a macroscopic level, this function appears very similar to the sphere and de Jong functions. It does, however, have a very significant amount of noise, so there are many deceiving local optima. Performance is again worst with the smallest swarms but improves by several orders of magnitude for the ring and dynamic sociometries as the population size increases from 10 to 20. From that point, degradation of performance occurs more rapidly for the

ring topology than it does for the dynamic one. The most surprising result here is how poorly the star sociometry performs. Because of the high connectivity present at the beginning in the star sociometry, the swarm is drawn early to a sub-optimal region. It fails to thoroughly explore the whole space.

Giunta. As with the Rastrigin function, performance increases as the size of the population increases. This function also has many steep local optima. With a population size of five, the ring sociometry gives the best performance. However, the ring fails to improve as much as the other two as size increases. The star sociometry is much stronger than the dynamic with smaller swarm sizes, but that gap is much narrower when 60 particles are used. The trend suggests that the optimal number of particles is even greater than 60, but the performance improvement is leveling out by that point.

Conclusions and Future Work

None of the sociometries studied here completely dominates the others. The ring sociometry seems to work best when the function is smooth. The dynamic and star sociometries tend to excel when there are many local optima. Larger swarms tend to be more effective on functions that have more numerous and dramatic local optima. Statistical analysis will provide a more quantitative comparison of the performance of PSO with different sociometries and population sizes.

Dynamic sociometry proved to be effective in some situations but not all. This research proposed only one possible dynamic sociometry. Other variants need to be explored.

References

[1] Angeline, P., "Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences." *Proceedings of the Seventh Annual Conference on Evolutionary Programming*, March 1998, pp. 601-610.

[2] Clerc, M. and Kennedy, J., "The particle swarm – explosion, stability, and convergence in a multidimensional complex space." *IEEE Transactions on Evolutionary Computation,* Volume 6, Issue 1, February 2002, pp. 58-73.

[3] Eberhart, R. and Hu. X., "Human tremor analysis using particle swarm optimization." *Proceedings of the Congress on Evolutionary Computation,* 1999, Washington, DC, pp. 1927-1930.

[4] Eberhart, R. and Kennedy, J., "A new optimizer using particle swarm theory." *Proceedings of Sixth International Symposium on Micro Machine and Human Science,* Nagoya, Japan, October 1995.

[5] Eberhart, R. and Shi, Y., "Evolving artificial neural networks." *Proceedings of the International Conference on Neural Networks and Brain,* 1998, Beijing, P.R.C. PL5-PL13.

[6] Eberhart, R. and Shi. Y., "Tracking and optimizing dynamic systems with particle swarms." *Proceedings of the Congress on Evolutionary Computation,* 2001, Seoul, Korea.

[7] Kennedy, J., "Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance." *Proceedings of the Congress on Evolutionary Computation,* 1999, pp. 1931-1938.

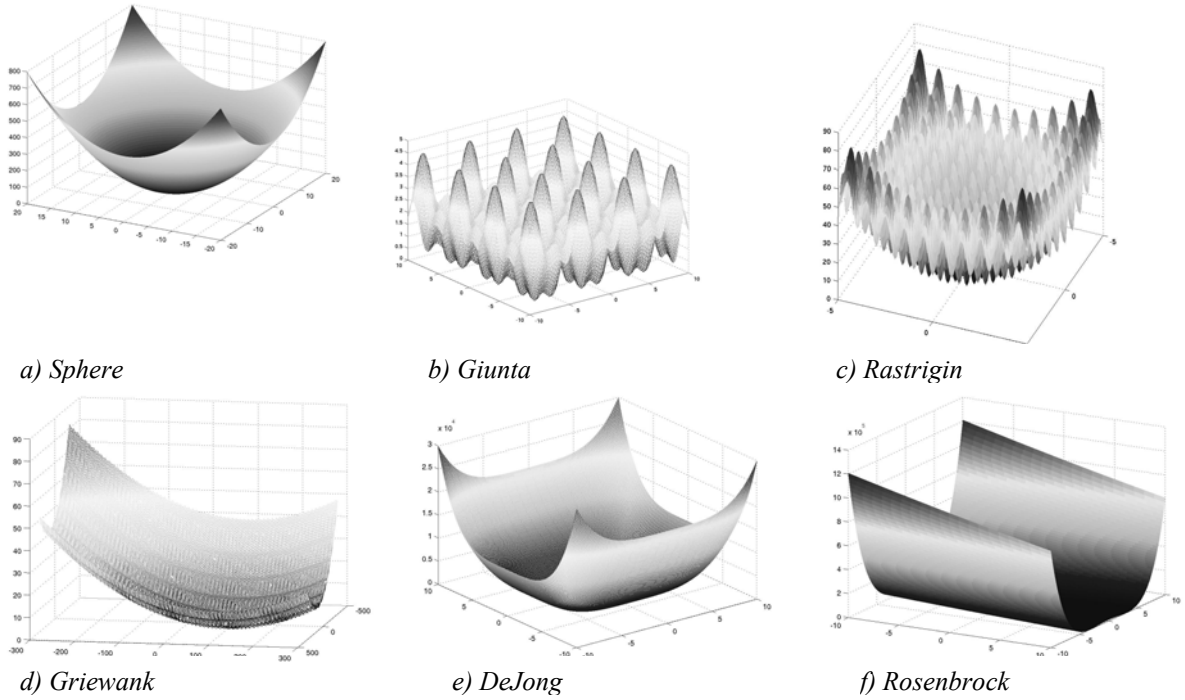
[8] Kennedy, J. and Eberhart, R., *Swarm Intelligence*, Morgan Kaufmann Academic Press, 2001.

[9] Kennedy, J. and Eberhart, R., "Particle Swarm Optimization." *Proceedings of IEEE Conference on Neural Networks,* Perth, Australia, 1995, pp. 1942-1948.

[10] Kennedy, J. and Mendes R., "Population structure and particle swarm performance." *Proceedings of the Congress on Evolutionary Computation,* 2002. CEC '02., Volume 2, pp. 1671-1676.

Name	Formula	Initial Range	Dim	Min.
De Jong f4	$f(\mathbf{x}) = \sum_{i=1}^{n} i \cdot x_i^4$	±20	30	0
Rosenbrock	$f(\mathbf{x}) = \sum_{i=1}^{n} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	±10	30	0
Rastrigin	$f(\mathbf{x}) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	±5.12	30	0
Griewank	$f(\mathbf{x}) = \frac{1}{4000}\sum_{i=1}^{n}(x_i - 100)^2 - \prod_{i=1}^{n}\cos[\frac{(x_i - 100)}{\sqrt{i}}] + 1$	±300	30	0
Sphere	$f(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$	±50	30	0
Giunta	$f(\mathbf{x}) = \sum_{i=1}^{n} \sin(\frac{16}{15}x_i - 1) + \sin^2(\frac{16}{15}x_i - 1) + \frac{1}{50}\sin[4(\frac{16}{15}x_i - 1)] + \frac{3}{100}$	±10	30	≈0.9

Table 1. Test problems used in experiments.



a) Sphere *b) Giunta* *c) Rastrigin*

d) Griewank *e) DeJong* *f) Rosenbrock*

Table 2. Graphs of functions used in experiments.

Function	Sociometry	Size of Swarm								
		5	10	15	20	25	30	40	50	60
DeJong f4	Ring	2.16734E-19	**1.234E-58**	1.68056E-47	1.17649E-35	5.2478E-28	1.67188E-22	1.21236E-15	1.56348E-11	9.27951E-09
	Star	9.78098E-05	2.29074E-44	2.00455E-57	6.39962E-58	2.93725E-53	1.20653E-48	1.5256E-40	1.99755E-34	8.49304E-30
	Dynamic	1.26527E-17	3.567E-52	2.0248E-55	1.31758E-50	1.82739E-45	2.28922E-40	2.64199E-33	8.02155E-28	9.11459E-24
Rosenbrock	Ring	70.9131	22.4409	23.8687	24.4061	24.5818	24.909	25.5905	26.4521	27.1614
	Star	210.282	20.4808	19.7255	**17.089**	19.4842	21.0972	21.8026	22.5597	22.8654
	Dynamic	71.3968	21.3428	20.9648	21.1223	21.5867	22.091	22.8626	23.5524	24.0095
Rastrigin	Ring	115.415	88.5512	82.5814	74.6217	75.6209	70.6461	68.6804	67.6638	66.9437
	Star	122.381	95.5158	84.5714	80.5915	77.6067	69.647	66.6622	60.6924	60.6924
	Dynamic	78.6017	58.7025	55.7176	52.7328	51.7378	47.758	45.7681	43.7782	**41.7883**
Griewank	Ring	0.280244	0.01969	0.00739604	1.89735E-19	**2.168E-19**	1.07433E-14	1.81456E-10	8.54105E-08	7.00149E-06
	Star	1.18337	0.131911	0.0683882	0.0405032	0.0172312	0.0172386	0.012321	0.0123161	0.00739604
	Dynamic	0.224826	0.00985728	3.6169E-15	5.5999E-17	**2.168E-19**	2.1684E-19	1.6263E-19	1.0842E-18	0.00739604
Sphere	Ring	2.77919E-13	**1.423E-41**	1.09195E-34	3.83608E-26	1.95995E-20	1.30252E-16	1.06001E-11	9.81586E-09	8.85562E-07
	Star	0.381799	8.49702E-22	2.92319E-31	9.96222E-33	1.51968E-32	8.40708E-31	2.5101E-26	2.35023E-22	1.13674E-19
	Dynamic	4.0946E-12	2.92291E-33	1.52035E-35	1.91279E-33	2.61298E-30	2.92738E-27	1.51762E-22	7.26256E-19	3.96684E-16
Giunta	Ring	2.00494	1.9275	1.78866	1.74837	1.66934	1.71671	1.67678	1.71688	1.71558
	Star	2.20904	1.77877	1.54911	1.44731	1.32925	1.25231	1.20106	1.13912	**1.11479**
	Dynamic	2.41654	2.21339	2.00446	1.92534	1.73633	1.64487	1.55523	1.43401	1.35078

Table 3. Median performance over 200 runs of PSO on six functions using various combinations of sociometry and swarm size. The best combination of swarm size and sociometry for each problem is highlighted.