Stereotyping: Improving Particle Swarm Performance With Cluster Analysis

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

Abstract – Individuals in the particle swarm population were "stereotyped" by cluster analysis of their previous best positions. The cluster centers then were substituted for the individuals' and neighbors' best previous positions in the algorithm. The experiments, which were inspired by the social-psychological metaphor of social stereotyping, found that performance could be generally improved by substituting individuals', but not neighbors', cluster centers for their previous bests.

1 Introduction

The particle swarm algorithm is based on a metaphor of human social interaction. Collaboration between individuals results in the rise of self-organizing "cultures" in the population and the optimization of individual problem-solution vectors. The metaphor actually gave rise to the discovery of the algorithm, through the route of social-psychological simulations, and since the optimization capabilities of the method were found, the metaphor has provided new ideas for improving its performance as well.

People who interact frequently tend to become more similar; their attitudes, beliefs, and behaviors change toward those of their companions. The process is accelerated in human society through categorization and stereotyping of group characteristics. Individuals identify themselves and one another with groups, and assign the qualities of groups to themselves and others. Research on reference groups (e.g., Sherif and Sherif, 1964; Newcomb, 1958) and social influence suggests that people choose their opinions and behaviors on the basis of their groups' norms, tending to converge on the group's average.

Swiss social psychologist Henri Tajfel (1981) has shown that *mere categorization* is sufficient for an aggregation of people to behave as a group. In one experimental paradigm, participants were randomly assigned to groups, and were told (falsely) that their group prefers the art of Kandinsky over that of Klee, or vice versa. Even though assignments were arbitrary, individuals rated the characteristics of the members of their group more positively and allocated rewards preferentially to their group. Turner and Oakes' (1989) *self categorization theory* further notes that individuals change their behaviors to become more similar to the group's perceived norms.

Group stereotyping can be compared to other forms of categorization. Cognitive research has shown that objects are assigned to a category by comparison to some central tendency of category members. Categories, in other words, are fuzzy: a robin is more of a bird than a penguin is. The debate continues however as to whether categories are defined by perfect exemplars, or by statistical prototypes, which might be a kind of average of members' characteristics. The present experiment compares implications of exemplar versus prototype implementations of social categories.

In the traditional particle swarm algorithm, individuals compare their neighbors' previous best performance on an objective function, and adjust their trajectory through the search space toward the best location found by any member of the neighborhood (social influence), and toward the location of their own previous best performance (learning from experience). Neighborhoods have been defined in various ways. The present paradigm used the neighborhood topology called *lbest*, with the neighborhood comprising the individual particle and its immediate neighbors on each side.

Randomly initialized individuals search according to the following formulas:

$$\vec{v}_i(t) = \chi(\vec{v}_i(t-1) + \varphi_1(\vec{p}_i - \vec{x}_i(t-1)) + \varphi_2(\vec{p}_g - \vec{x}_i(t-1)))$$
$$\vec{x}_i(t) = \vec{x}_i(t-1) + \vec{v}_i(t)$$

where $\vec{v}_i$ is individual i's velocity through the parameter space, $\vec{x}_i$ is the individual's current position, $\vec{p}_i$ is the best point found so far, $\vec{p}_g$ is the best point found by any member of i's neighborhood, and φ_1 and φ_2 are random numbers between zero and an upper limit that is a system parameter (in the current implementation the total $\varphi_{total} = \varphi_1 + \varphi_2$ is limited to 4.1). Though there are several variations on the particle swarm algorithm, the current versions implement a constriction coefficient χ defined by Maurice Clerc (Clerc and Kennedy, under review) to induce convergence and prevent the particles from exploding outside the desirable range of the search space. The coefficient is used with φ_{total}>4.0, and is calculated as:

$$\chi = \frac{2\kappa}{\left|2 - \varphi - \sqrt{\varphi(\varphi - 4)}\right|}$$

where κ is usually equal to 1.0. In the present experiments, velocities were further limited by *Vmax*, which was set to the dynamic range (*Xmax*) of the variables, as described in Eberhart amd Shi (2000).

Though the particle swarm attracts individuals' trajectories toward the previous best points found by individuals in their neighborhoods, humans often approximate the stereotypical behaviors and beliefs of groups, which are not necessarily the behaviors of any particular individuals. The current investigation modified the particle swarm algorithm by assigning individuals in the population to groups based on what region of the search space they were in; individuals were categorized on the basis of their search behavior just as people are stereotyped on the basis of their beliefs and behaviors. Average previous best positions found by the group could then be substituted for either the individual's previous best, the neighborhood's previous best, or both.

2 Stereotyping and Search

Population-based search algorithms, especially ones that feature interaction between individuals as in evolutionary recombination and crossover operators, often suffer from difficulties in multimodal problem spaces. Randomly paired individuals frequently come from regions of different local optima. As a result the offspring are likely to be relatively unfit. This problem has given rise to artificial solutions such as deme or island evolution, where populations evolve in isolation until they converge on a solution, and then individuals from various demes are brought together to compete.

It is possible however to use cluster-analytic methods to identify stereotypical populations, that is, groups of individuals that are working toward particular optima in the multimodal problem space. A number of *k*-means algorithms for accomplishing this are known. The one used here is vaguely derived from Warren Sarle's "fastclus" algorithm implemented in SAS (Sarle, 1985). It works like this:

1. Select *C* individuals' $\vec{p}_i$ vectors as proposed cluster centers (evenly distributed topologically in the population)
2. Calculate the distance of all *N* individuals' $\vec{p}_i$ from the *C* centers
3. Assign all *N* individuals to the nearest cluster center
4. Calculate the center, e.g., the mean point in vector space, for each cluster
5. Loop to 2 until centers stabilize

In the current implementation the program looped through steps 2-5 three times, as the desire was not necessarily to have perfect clusters, but "good enough" ones, balanced against computational cost. Each member of the population of proposed problem solutions was stereotypically assigned to a group on each iteration, and the central tendency of the group was identified. An assumption was that a group of individuals clustering in a region might indicate the location of an optimum.

3 The Modified Particle Swarm

Four variations were tried:

1. The formula used *i*'s and *g*'s individual previous bests
2. The individual best term was replaced with *i*'s cluster's center
3. The neighborhood best term was replaced with *g*'s cluster's center
4. Both best terms were replaced with cluster centers

The first of these versions is the "traditional" particle swarm. For version 1, $\vec{p}_i$ and $\vec{p}_g$ are simply the best points found by the individual and by its best-performing neighbor.

In version 2, the individual's previous best performance is stereotyped, that is, particle *i* is attracted toward its cluster's average previous best performance, as well as the performance of the best neighbor. This is the "self categorization" version, where the individual stereotypes itself, and substitutes the group norm for its own previous performance, while the neighborhood best is taken from an individual.

Version 3, replacing $\vec{p}_g$ with *g*'s cluster center, attracts particle *i* toward the stereotypical behavior of *g*'s group, while *i* still gravitates toward its own previous best performance. Social psychologically this is like conforming to a group norm rather than the actual behaviors of group members.

The fourth version uses cluster centers for both previous best terms. Here the individual stereotypes both self and other, and information from aggregate performance is used wholly.

4 Method: Experiment 1

An experiment was designed where particle swarms were run in each of the four conditions on five well-known test functions. The individual term of the particle swarm formula can be the individual particle's best performance or a cluster center, as can the social-influence term.

Table 1. Experimental design. Versions are implemented as two factors, CLUSI and CLUSG, each with two levels.

		CLUSG	
		individual g	cluster g
CLUSI	individual i	1	3
	cluster i	2	4

With this design it is possible to test the effects of each of the two independent variables and their interaction across the five test functions. Thus the design was conceptually a 2×2×5 factorial experiment. The dependent variable was best performance after some number of iterations. Because the test functions are not comparable, for instance, their ranges are quite different, it was not reasonable to compare performance on them using their unstandardized output. Instead, scores within each function were standardized for the analysis by subtracting the mean and dividing by the standard deviation. Using this method it is still not possible to test for a main effect of function, that is, we cannot identify if one function or another is harder or easier overall, but we can compare the effects of the experimental manipulations across the various functions, to see if using cluster centers in either term works better with some problems than with others. All conditions were run for twenty trials, with populations of 20 individuals and 5 clusters. Test functions are shown in the Appendix. Functions were implemented in 30 dimensions, except for f6, which is a 2-dimensional problem.

5 Results, Experiment 1

Table 2 gives the mean best performance for all functions, for all experimental conditions, after 1,000 iterations. (Each entry represents 20 trials.)

Table 2. Mean best performance after 1,000 iterations, by experimental condition.

Mean Best Performance		
	individual-g	**cluster-g**
Sphere		
individual-i	0.0000	0.3178
cluster-i	0.0000	0.0002
Rosenbrock		
individual-i	39.6499	51.2138
cluster-i	25.2236	26.8999
Rastrigin		
individual-i	51.8001	9.2720
cluster-i	8.6626	10.3473
Griewank		
individual-i	0.0012	2.0669
cluster-i	0.0151	0.1580
Shaffer's f6		
individual-i	0.0000	0.0020
cluster-i	0.0017	0.0038

As seen in Table 3, analysis of variance on the results found that several effects were significant. The largest effect, by far, was the main effect for CLUSI: results were significantly better when $\vec{p}_i$ was a cluster center than when it was the individual particle's previous best position. Importantly, CLUSI interacted with FUNC, suggesting that the self-stereotyping manipulation improves performance more on some functions than on others. In fact, cluster-i performed somewhat less well than the standard version on the Griewank and Shaffer's f6 functions, though it was very much better on Rastrigin, and moderately better on Rosenbrock. The second-largest effect, in terms of the amount of variance explained, was CLUSG's main effect: performance was significantly worse when $\vec{p}_g$ was a cluster center. CLUSG interacted with FUNC, with the effect being that cluster-g conditions did worse than individual-g on all functions except Rastrigin, where they performed better than the average; this effect seems to be due largely to the poor performance of the standard Version 1 particle swarm on that function. CLUSI and CLUSG interacted significantly; while performance was better when $\vec{p}_i$ was a cluster center, algorithms with individual-i performed especially badly when $\vec{p}_g$ was a cluster center. Finally, the three-way interaction suggests that, in some sense, everything depends on everything else. While $\vec{p}_i$ cluster centers may result in better problem solutions in general, the improvement to be expected is moderated by whether $\vec{p}_g$ is an individual's best or a cluster center, depending on the function.

Table 3. Analysis of variance based on 400 trials, 20 in each condition, with 380 degrees of freedom for error.

Source	DF	Type III SS	Mean Square	F Value	Pr > F
CLUSI	1	48.29736	48.29736	107.6	0.0001
CLUSG	1	15.45021	15.45021	34.42	0.0001
FUNC×CLUSI	4	36.72209	4.590261	20.45	0.0001
FUNC×CLUSG	4	59.85455	14.96364	33.34	0.0001
CLUSI×CLUSG	1	4.31141	4.31141	9.6	0.0021
FUNC×CLUSI× CLUSG	4	59.79055	14.94764	33.3	0.0001

6 Method: Experiment 2

In human cognition, clustering or categorization of persons and things happens very fast. In fact, recognition seems to be largely a matter of sensing that information is processed faster for familiar stimuli than for unfamiliar ones; the term "perceptual fluency" is used in cognitive psychology to describe this effect (Jacoby and Dallas, 1981). In a von Neumann computer though clustering requires some work, which takes some time. The substitution of cluster centers resulted in better average results over a fixed number of iterations; another question is, how does the clustering affect length of time required to reach a criterion?

To test this question, the program was run with a timer until a criterion was met. Each experimental condition was run twenty times, as before; trials were terminated after 3,000 iterations if the criterion had not been reached, or after 500 iterations without improvement. As before, the constricted version with *Vmax=Xmax* was used, with populations of 20 particles.

Table 4. Functions used, their initialization ranges, and criteria, for Experiment 2.

Function	Xmax	Criterion
Sphere	100	0.01
Rosenbrock	10	100
Rastrigin	5.12	100
Griewank	300	0.05
Shaffer's f6	100	0.00001

As can be seen in Table 5, the cluster-g conditions struggled to meet the criteria, and the individual-g versions failed to meet them sometimes for two of the functions. The median amount of time required for Version 2, with cluster-i and individual-g, relative to the standard version 1 particle swarm ranged from about 74 per cent to to nearly 200 per cent. Thus, as expected, adding the clustering steps to the program generally made it take longer – but not always.

Table 5. Median time to meet the criterion, in seconds, for all functions, for all experimental conditions. Numbers in parentheses give the percentage out of twenty trials that met the criterion. The right column gives the ratio of the median times for the two individual-g conditions, for each function.

	Median time in seconds		Ratio
	individual-g	cluster-g	
Sphere			
individual-i	1.479 (100)	∞ (0)	1.301
cluster-i	1.924 (100)	2.852 (100)	
Rosenbrock			
individual-i	1.334 (100)	∞ (30)	0.739
cluster-i	0.9902 (100)	1.260 (100)	
Rastrigin			
individual-i	0.631 (100)	1.283 (100)	1.994
cluster-i	1.258 (100)	0.990 (95)	
Griewank			
individual-i	1.895 (95)	∞ (0)	1.432
cluster-i	2.713 (90)	∞ (50)	
Shaffer's f6			
individual-i	0.574 (60)	∞ (30)	1.670
cluster-i	0.959 (70)	∞ (15)	

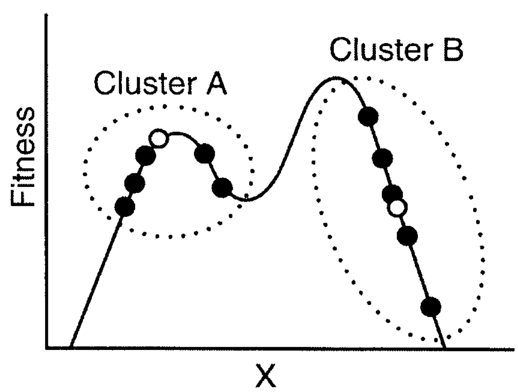
7 Discussion

A number of modifications to the particle swarm paradigm have been tested. Most, such as Angeline's (1998) hybridization of particle swarms and evolutionary optimization and the present author's (1999) manipulations of neighborhood configurations, affect performance differently depending on the function being optimized. Substitution of cluster centers for $\vec{p}_i$, though, seems to lend a general advantage across functions, as shown in the main effect for CLUSI. Even where the cluster-i version performed worse than the standard version, it was only slightly worse. The interaction of CLUSI with FUNC, though, suggests that further investigation is needed, as there may be functions where self-stereotyping performs significantly worse than the standard implementation. This sample of five test functions does not represent the entire universe of possibilities, but the results here are encouraging.

Clustering tends to add some computational cost. For instance, the present method required three iterations through every element of every vector in the entire population to calculate the cluster centers. It is anticipated though that some heuristics could be implemented to reduce this cost. For instance, new cluster centers need to be computed only when a new $\vec{p}_i$ is discovered; this is probably less often than every iteration. The clustering method used here is not intended to be optimal, either in its ability to identify clusters or in its computational efficiency. It is quite likely that a different clustering algorithm could reduce costs significantly.

The empirical evidence presented here suggests that particle swarm search is relatively effective when individuals are attracted toward the centers of their own clusters, and is not generally good when they are attracted to neighbors' cluster centers; it is helpful to stereotype oneself, but not to stereotype others.

Figure 1. Cluster A's center (white circle) performs better than any of the members of the cluster, while Cluster B's center performs better than some, and worse than others.



It is interesting to think about *how* substituting a cluster center for the individual's previous best can result in improvement. The explanation is to be found in consideration of the probability that any individual's performance will be better than its cluster's center. As seen in Figure 1, if the individuals comprising a cluster are distributed around a local optimum, as in Cluster *A*, then it is entirely possible that the evaluation of the cluster center will be better than that of any of the particles that make it up. If, on the other hand, the particles are approaching an optimum from one side, as exemplified by Cluster *B*, then the cluster center's fitness will be closer to the average of the particles' evaluations. It is not likely, since the clusters are defined by the best points found so far, that the cluster center will be worse than the average of the particles that make it up, unless cluster radii are so great relative to landscape correlation distance that clusters span optima. The average cluster center's fitness will be greater than the average individual's.

In the *lbest* version as used here, the neighborhood best is determined through comparison of individuals *i*-1, *i*, and *i*+1. Taking the best of the three obviously increases the chances that $\vec{p}_g$ will be better than a cluster center. It appears in the present data that the performance of the cluster center is usually better than that of an average individual, but worse than the best of a group of three.

How many clusters should there be? The answer to this question probably depends on problem dimensionality and the number of local optima. Further, there cannot be more clusters than population members. Since the point is to identify areas where several individuals are aggregating around a local optimum, the number of clusters should be considerably less than the number of population members. Selection of the number five in the present research was a purely arbitrary starting point for investigation.

Finally, increasing the number of clusters, at least by the present method, increases computational cost. From this view it is better to have fewer clusters, rather than more.

8 In Sum

The particle swarm algorithm was developed as a simplified model of social behavior, in particular human social behavior. As in human societies, individuals gravitate toward one another's positions in a (cognitive) space, especially emulating the successes of their neighbors. Human societies provide another tendency, as well, which is the tendency to stereotype self and others, to use the norms of one's own group as a point of reference toward which one tends to gravitate, and the perceived norms of others' groups to form assumptions about them.

The present results surprisingly suggest that self-categorization is more important, in terms of cognitive optimization, than the tendency to categorize others. Individuals are better off to refer to their own group or cluster's central tendencies, that is, to stereotype themselves, and perform more poorly when they try to emulate the central tendencies of their neighbors' groups, rather than the neighbors themselves.

In terms of the particle swarm as a computational intelligence paradigm, the present results indicate that performance can be improved – with some small but nontrivial computation cost – by substituting cluster centers for $\vec{p}_i$, but not for $\vec{p}_g$.

9 References

Angeline. (1998). Using selection to improve particle swarm optimization. *Proceedings of the 1998 International Conference on Evolutionary Computation*, Anchorage, Alaska, 84-89. Piscataway NJ: IEEE Press.

Clerc, M., and Kennedy, J. (under review). The particle swarm: Explosion, stability, and convergence in a multi-dimensional complex space.

Eberhart, R. C., and Shi, Y. (2000, in press). Comparing inertia weights and constriction factors in particle swarm optimization. *Proceedings of the 2000 Congress on Evolutionary Computation.*

Jacoby, L. L., and Dallas, M. (1981). On the relationship between autobiographical memory and perceptual learning. *Journal of Experimental Psychology: General, 110,* 306-340.

Kennedy, J. (1999). Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance. *Proceedings of the 1999 Congress on Evolutionary Computation*, 1931-1938. Piscataway NJ: IEEE Press.

Newcomb, T. M. (1958). Attitude development as a function of reference groups: The Bennington study. In Maccoby, E. E., T. M. Newcomb, and E. L. Hartley (Eds.), *Readings in Social Psychology.* New York: Henry Holt and Company.

Sarle, W. S. (1985). The FASTCLUS procedure. *SAS User's Guide Statistics, Version 5 Ed.*, pp. 377 - 401.

Sherif, M., and Sherif, C. W. (1964). *Reference Groups: Exploration into Conformity and Deviation of Adolescents.* Chicago: Henry Regnery Company.

Tajfel, H. (1981). *Human Groups and Social Categories.* Cambridge: Cambridge University Press.

Turner, J. C., and Oakes, P. J. (1989). Self-categorization theory and social influence. In P. B. Paulus (Ed.), *The Psychology of Group Influence* (pp. 233-275). Hillsdale NJ: Erlbaum.

Appendix: Functions Used

Sphere function (De Jong's f1)	$f_1(\vec{x}) = \sum_{i=1}^{n} x_i^2$
Schaffer's f6	$f_6(\vec{x}) = 0.5 + \frac{(\sin\sqrt{x^2+y^2}) - 0.5}{(1.0+0.001(x^2+y^2))^2}$
Griewank function	$f_7(\vec{x}) = \frac{1}{4000}\sum_{i=1}^{n}(x_i-100)^2 - \prod_{i=1}^{n}\cos\left(\frac{(x_i-100)}{\sqrt{i}}\right) + 1$
Rosenbrock function	$f_9(\vec{x}) = \sum_{i=1}^{n}(100(x_{i+1}-x^2)^2 + (x_i-1)^2)$
Rastrigin function	$f_{10}(\vec{x}) = \sum_{i=1}^{n}\left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$

Acknowledgment

Thanks to Russ Eberhart and Yuhui Shi for commenting on an earlier draft of this paper and suggesting the second experiment.