Report on Consultant Effort for Virginia Space Grant Consortium

Rex K. Kincaid
Department of Mathematics
College of William & Mary
January 16, 2008

As a Virginia Space Grant Consortium initiative, I was tasked to assist Consortium Member NASA Langley Research Center's Geographic Information Systems Team with evaluation and development of optimization algorithms for facilities space utilization. My primary activity during has been to develop a tabu search code for space allocation problems at NASA-Langley Research Center. However, before developing a tabu search an appropriate test bed of networks was needed.

A C program was developed to randomly generate social networks with many of the same features exhibited at NASA-LaRC. The code generates a random set of points (e.g. Figure 1) in the unit square representing the initial allocation of individuals in the system. These points are the nodes of our social network. Next, a second, smaller (e.g. 5) set of random points on the unit square are generated to represent organization mass centers. Two method are used to assign individuals to organization. The first is random. The second probabilistically assigns each individual an organization in proportion to the inverse distances from each individual to all organization centers. It is convenient to think of the organization assignments for each individual as a color assignment. Computing these probabilities requires constructing the Euclidean distance between each individual and each organization mass center. Initially, we assume that the social network is completely dense. Every individual interacts with every other individual. Consequently, if n denotes the number of nodes (individuals) then there are $n(n-1)/2$ edges. The length of an edge is the Euclidean distance between a pair of nodes.

We now have a network in which each node is assigned a color and each edge has a designated number (the Euclidean distance). Next, we focus on the strength of the social connections between nodes (individuals). In particular we wish to estimate the interaction frequency between any pair of nodes. We define two indicators that are assumed to be independent. First, we calculate a spatial proximity indicator and assume interactions between individuals are proportional to the inverse square of the distance. That is, an edge twice as long as another will have $1/4$ the interactions. Next, we calculate the effect of organization membership on the interactions. If two edges are equal in length, but the first edge consists of nodes from the same organizations (same node color) while the second edge's nodes are from different organizations then we will assume that the interactions across the first edge are 10 times more frequent. The combined effect of these two indicators results in the social interaction frequency between any pair of nodes. This gives rise to a natural performance measure for our social network. Let the cost of a network equal the sum of the products of edge lengths and edge weights (interaction frequency) taken over all edges.

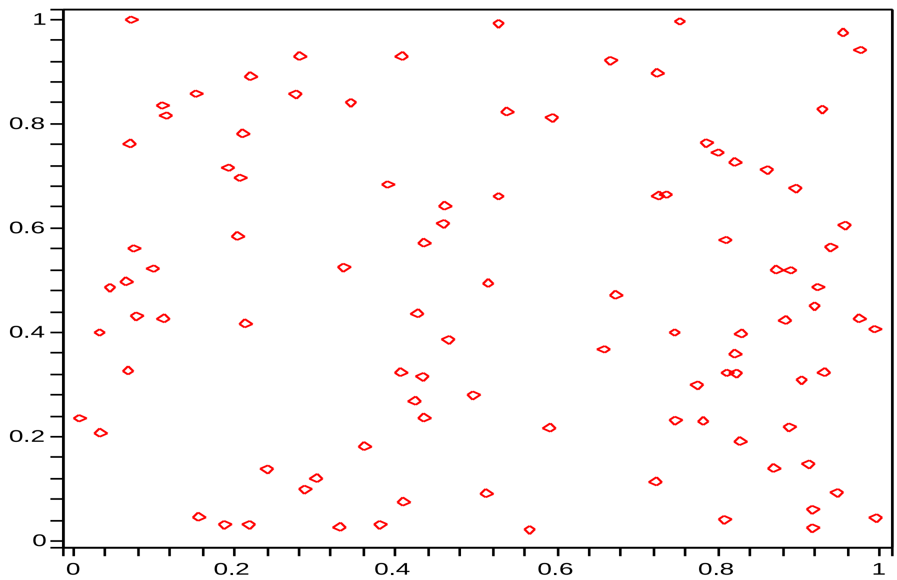

Figure 1. 100 randomly generated points in the unit square

The number of edges in a full network may be prohibitively large for many computations. Since many of the edge weights (strength of social connections) are quite small it should be possible to select a smaller set of edges from which decisions about network performance can be estimated with a reasonable level of accuracy. For example, if there are $n = 100$ nodes, then our network has $n(n-1)/2 = 4950$ edges. How accurately can we estimate the cost of a network by sampling 500 edges or 1000 edges? We construct a subset of the edges by sampling randomly (without replacement) via a roulette wheel in which each sector of the roulette wheel represents an edge weight. Table 1 records cost estimates for two 100 node networks. The first network (columns 3 and 4) used method 1 to assign organizations to individuals, while the second (columns 5 and 6) used method 2. Columns 3 and 5 record the percentage of the complete network cost (all 4,950 edges) captured by the selected subnetwork. Columns 4 and 6 record the rate of change in these cost estimates ad the number of edges increases. That is, $.181 = .596 - .415$. The patterns exhibited for both of these networks is quite similar. For example, when slightly more than fifty percent (2500) edges are selected for the subnetwork we achieve a cost estimate of 87 percent in the first case and 88 percent in the second. In addition, for both networks tested there is a similar decreasing rate of improvement with each additional 500 edges added to the subnetwork.

% Edges	# Edges	Cost Ratio	Ratio Δ		Cost Ratio	Ratio Δ
10	500	.415	—		.466	—
20	1000	.596	.181		.614	.148
30	1500	.715	.119		.749	.135
40	2000	.809	.094		.827	.078
50	2500	.869	.060		.882	.055
60	3000	.909	.044		.921	.039
70	3500	.942	.033		.949	.028
80	4000	.966	.024		.970	.021

Table 1. Cost estimates obtained by subnetworks

Given a particular (random) placement of nodes and an assignment of colors (organizations) can we improve upon the initial cost of the network by swapping node positions? To see what gains can me made we turn to a tabu search metaheuristic. Optimization can be done on the full network or on any subnetwork in which edges are selected as described above.

1. Tabu Search

Tabu search begins with an initial feasible solution either randomly generated or generated through some other technique. Next, the moves in the neighborhood of the initial solution are examined. (Here we focus on a pairwise exchange neighborhood, but the same description applies for neighborhoods of any ilk.) Tabu search evaluates all neighboring solutions and chooses the exchange which most improves the objective function value the most or, if there is no improving exchange, then it selects an exchange that diminishes the objective value the least. After a move is chosen, the two entries exchanged, i and j, are placed on a *tabu list*. A tabu list is a list of moves which cannot be performed again for a designated number of iterations. In doing so, j cannot be dropped and i added, avoiding the return to the previously visited solution. Thus, the tabu list discourages the greedy aspect of the search from cycling. When the tabu search evaluates the neighborhood of a subsequent solution, it will not choose moves that are tabu unless they meet a previously determined aspiration criteria. An aspiration criterion gives a lower limit on the amount of improvement of the objective function value, and if a move, although tabu, improves the objective function value more than the aspiration criteria, that move will be a candidate for selection. In the case of aspiration, cycling is clearly avoided since we are not returning to a previously observed solution value. Tabu search continues until an iteration limit is reached.

In the case of our social network a straightforward implementation of tabu search is possible. Exchanging the positions, (x,y) coordinates, of two nodes i and j is a move. Computing the cost of a move requires a bit more effort. If we are working with the full network there is no need to update the edge lengths, but each edge weight (interaction frequency) may need to be updated for any edge with either i or j as a node. When a sample of the edges from the full network are used then the adjacency list for nodes i and j must be swapped as well and both edge lengths and edge weights may need to be updated. We report on the results for tabu search when applied to the two 100 node networks associated with Figure 1 and Table 1. Examining all the neighbors of a particular solution is considered one iteration of tabu search.

Table 2 and Figures 2 through 5 summarize the performance of tabu search for two 100 node networks. In both cases 500 iterations (neighborhoods) were completed and a move was forced to remain tabu for the next 40 iterations unless it led to a solution with the best observed cost (aspiration criteria). Columns 1 lists the initial cost value for the two networks. Column 2 records the best cost value found when tabu search is allowed to search 500 neighborhoods (iterations). Column 3 lists the iteration at which the best solution is found, while column 4 records the first iteration for which an examined move was tabu. Column 6 records the average number of tabu moves found per iteration until the best solution is uncovered. The number after the backward slash computes the same average but ignores the iterations for which no tabu moves were found. This second number is nearly identical for both networks. Column 5 records the number of times a tabu move met the aspiration criteria. The general trend (observed in many other cases as well) is that the network for which the organization assignments is random–method 1–(row 1 of Table 2) is much bumpier with many local optima. High quality local optima for networks in which the organizations are assigned probabilistically–method 2–(row 2 of Table 2) are much easier to find. For example, 44 of the 116 iterations needed to uncover the best observed solution (row 2 Table 2) required no tabu moves. In addition, tabu search required required less than a third of the iterations needed to find the best observed solution for method 2 network than for the method 1 network.

Initial Cost	Best Cost	Best Itr	1st STM	Total Asp	Avg # STM
39,070.71	34,737.32	359	itr 14	55	10.8/11.3
44,414.94	36,153.39	116	itr 44	9	7.1/11.5

Table 2. Tabu search results for two 100 node complete networks

Figures 2–5 display the pattern associated with the costs of the solutions found at each iteration of tabu search. The tabu search results for the 100 node network (row 1 of Table 2) in which the organization codes are assigned randomly are displayed in Figures 2 and 3. The tabu search results for the 100 node network (row 2 of Table 2) in which the organization codes are assigned probabilistically are displayed in Figures 4 and 5. Both Figures 2 and 4 show the steady initial downhill progress (the greedy search phase) while Figure 3 and 5 show the movement of tabu search across various local optima.

An obvious question to consider is if optimizing any of the subnetworks (Table 1) leads to an improved complete network. Optimizing a network with fewer edges requires less computational effort. Thus, if the cost approximation is unaffected by tabu search, we would have an improved optimization procedure. To test this hypothesis, we applied tabu search to the same two 100 node networks but used 2500 edge cost approximations rather than the complete set of 4950 edges. From Table 1 we know that by using 2500 edges we achieve a initial cost approximation of 87 percent for the method 1 (randomly assigned organizations) generated network and 88 percent for the method 2 (probabilistically assigned organizations) network. As before, we allow 500 iterations and a move is tabu for 40 iterations. Table 3 records the cost comparisons. Columns 1 and 2 duplicate the information found in Table 2 for the complete set of edges (initial versus optimized). Columns 3 lists the initial 2500 edge solution cost and column 4 records the best solution found by tabu search. Column 5 gives the cost of the complete network using the optimized node locations from the 2500 edge tabu search solution associated with column 4. First, we see that the quality of the cost approximation for the optimized 2500 edge network has degraded. The initial values of the cost approximations were 87 and 88 percent. After tabu search is applied to the 2500 edge network the cost approximations are 62 and 63 percent respectively. In addition, the quality of the solutions in column 5 are significantly inferior to those given in column 2. In fact, a simple greedy search applied to the original complete network would arrive at better cost values than those given in column 5. Consequently, it seems likely that it will not be possible to focus the optimization procedure on the reduced network.

Complete Cost	Complete TS	2500 Cost	2500 TS Cost	Complete TS
39,070.71	34,737.32	33,956.29	22,794.52	36,536.57
44,414.94	36,153.39	39,180.57	24,009.02	38,208.13

Table 3. Tabu search results for two 100 node 2500 edge networks

2. Conclusions and Recommendations

The experimental results indicate that tabu search is an improvement over a greedy local search. For the complete networks (Table 2) the improvement in the cost of the final solution of tabu search over a greedy search is small (less than 0.5%) while for the 2500 edge networks (Table 3) the improvement in the final cost is 3 – 5%. In addition, both the short term memory function (tabu moves) and aspiration criteria were critical in breaking out of local optima. It appears that there is a fairly flat basin of solutions with similar costs for the problems studied here. Of course, these problems were generated at random and real data may lead to an entirely different set of conclusions.

For the problems studied here, changes in the cost from one solution to another (on fully connected networks) are driven by how the organization code weights are distributed. When two individuals i and j (nodes) are exchanged the effect of the Euclidean distance measure is unchanged. What changes is whether the interaction frequency measure will be scaled by a factor of 10 or not for each node adjacent to i and j. A more sophisticated weighting scheme between individuals will likely result in a cost landscape with more pronounced peaks and valleys. Consequently, we expect that the benefits of tabu search over a greedy local search to increase. We recommend that tabu search be implemented in the VB.NET code.

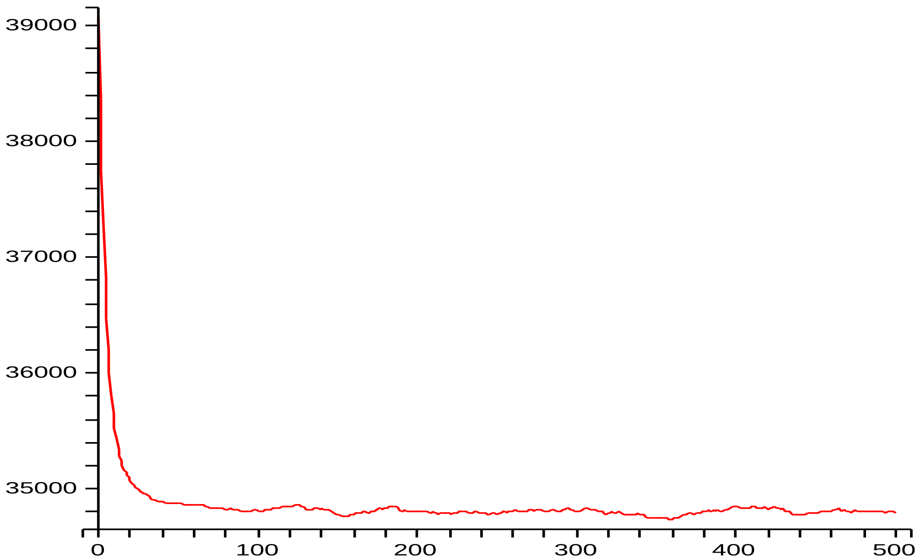

Figure 2. Complete cost minimization trajectory for method 1 network

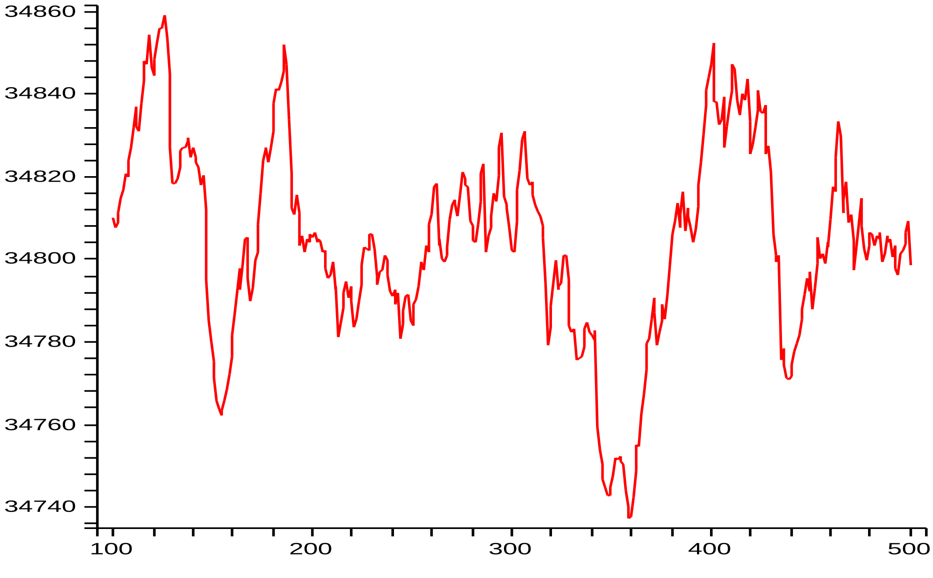

Figure 3. Partial cost minimization trajectory for method 1 network

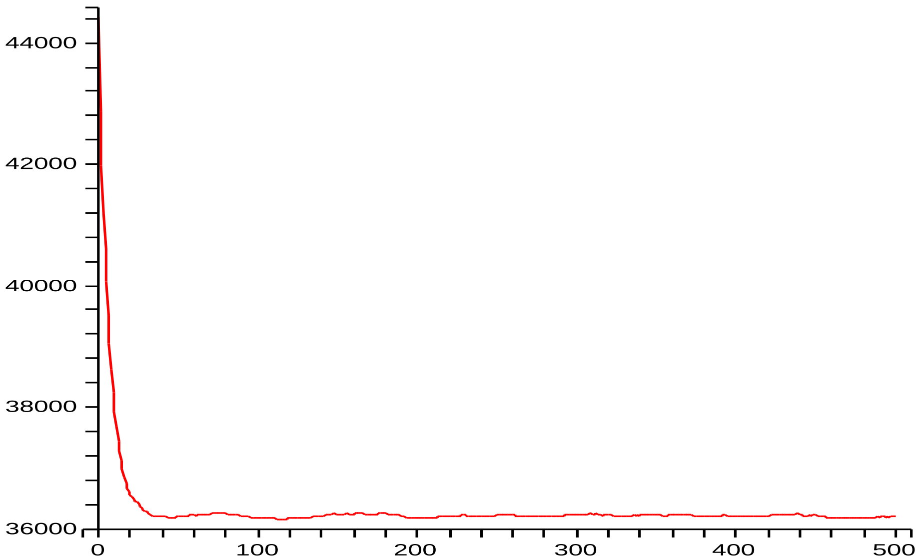

Figure 4. Complete cost minimization trajectory for method 2 network

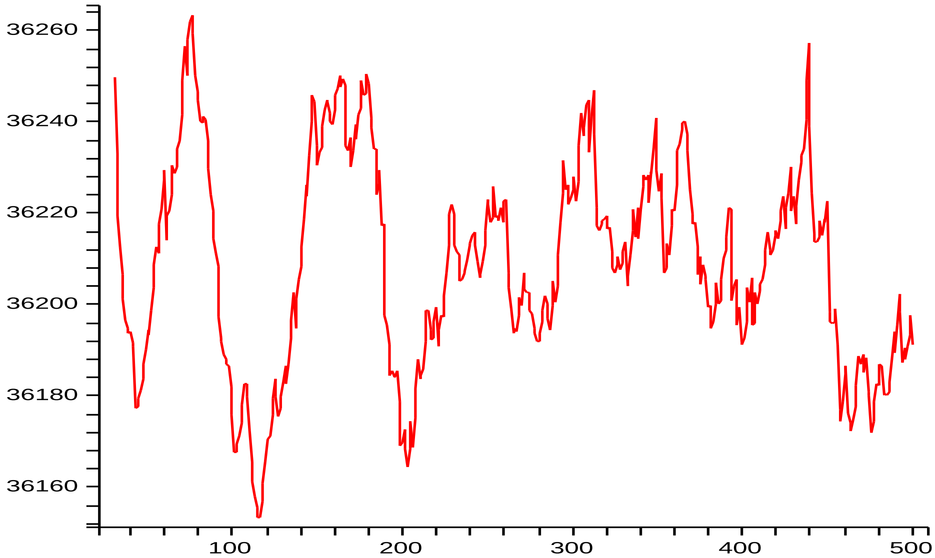

Figure 5. Partial cost minimization trajectory for method 2 network