

MAPPING THE CONVERGENCE OF GENETIC ALGORITHMS

ZVI DREZNER AND GEORGE A. MARCOULIDES

Received 29 August 2005; Revised 25 April 2006; Accepted 6 June 2006

This paper examines the convergence of genetic algorithms using a cluster-analytic-type procedure. The procedure is illustrated with a hybrid genetic algorithm applied to the quadratic assignment problem. Results provide valuable insight into how population members are selected as the number of generations increases and how genetic algorithms approach stagnation after many generations.

Copyright © 2006 Z. Drezner and G. A. Marcoulides. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

Hybrid genetic algorithms have recently become very popular metaheuristic methods (Beasley [6]). Most genetic algorithms produce offspring by mating parents and attempt to improve the population makeup by replacing existing population members with superior offspring. In contrast, hybrid genetic algorithms, sometimes called memetic algorithms (Moscato [28]), incorporate some heuristic improvement on every offspring before considering its inclusion into the population. For a plethora of introductory expositions published on the topic, see Salhi [33] or Beasley [6].

This paper examines the convergence of genetic algorithms using a cluster-analytic-type technique called the “MD procedure” (Marcoulides and Drezner [26]). The proposed procedure is illustrated with a hybrid genetic algorithm applied to the solution of the quadratic assignment problem (QAP). For a review of the QAP, see Rendl [31]. Because population members form clusters as progress is made to a solution, the clustering structure can provide a better implementation of genetic algorithms. For example, clustering structure can be used to develop better stopping criteria, for instance when the population clusters become stagnant.

In the next section we describe the MD procedure. In Section 3 we describe the quadratic assignment problem and the hybrid genetic algorithm used for its solution. In

2 Mapping the convergence of genetic algorithms

Section 4 we present an analysis of the procedures. Finally, Sections 5 and 6 present the results of some computational experiments and conclusions.

2. The MD procedure

The MD procedure is used to display k -dimensional data in two dimensions so that clusters can be easily observed. The procedure is successful in preserving distances between the various data points, thus retaining the structure of the set of points. It is based on the proposed solution for the layout problem (Drezner [12]), which is a variant of the DISCON (dispersion-concentration) procedure Drezner [11].

The layout problem is very similar to the QAP except that there are no specific locations for the facilities. While the QAP is concerned with finding the best permutation of the facilities among the *given* sites, the layout problem is concerned with the location of facilities of a given size anywhere in the plane. A set of weights $\{w_{ij}\}$, $w_{ij} = w_{ji}$ associated with facility pairs is given. As in the QAP, we wish that pairs of facilities with larger weights be closer to one another in the final configuration.

Drezner [12] proposed to minimize the function

$$\frac{\sum_{i \neq j=1}^n w_{ij} d_{ij}^2}{\sum_{i \neq j=1}^n d_{ij}^2} \quad (2.1)$$

which is equivalent to

$$\min \left\{ \sum_{i \neq j=1}^n w_{ij} d_{ij}^2 \right\} \quad \text{subject to} \quad \sum_{i \neq j=1}^n d_{ij}^2 = 1, \quad (2.2)$$

where d_{ij} is the Euclidean distance between the unknown locations of facilities i and j .

Since $d_{ij}^2 = (x_i - x_j)^2 + (y_i - y_j)^2$,

$$\begin{aligned} \sum_{i \neq j=1}^n w_{ij} d_{ij}^2 &= \sum_{i \neq j=1}^n w_{ij} (x_i - x_j)^2 + \sum_{i \neq j=1}^n w_{ij} (y_i - y_j)^2 \\ &= 2 \sum_{i=1}^n \left\{ \sum_{j=1}^n w_{ij} \right\} x_i^2 - 2 \sum_{i \neq j=1}^n w_{ij} x_i x_j + 2 \sum_{i=1}^n \left\{ \sum_{j=1}^n w_{ij} \right\} y_i^2 - 2 \sum_{i \neq j=1}^n w_{ij} y_i y_j. \end{aligned} \quad (2.3)$$

Define the matrix $S = \{s_{ij}\}$ by $S_{ii} = \sum_{j=1}^n w_{ij}$ and $S_{ij} = -w_{ij}$, for $i \neq j$. Our problem is equivalent to minimizing

$$\frac{x^T S x + y^T S y}{x^T E x + y^T E y}, \quad (2.4)$$

where E is the matrix S with all weights equal to 1.

The matrix S is singular and therefore one of its eigenvalues is 0 with an associated eigenvector of $(1, \dots, 1)$. Note that adding a constant to all weights does not change (2.1) or (2.2), and thus we can guarantee that all eigenvalues of S (except the zero eigenvalue) are positive. As is shown by Drezner in [12] and by Marcoulides and Drezner in [26],

the solution to (2.4) is $x = y$, where x or y is the eigenvector associated with the smallest positive eigenvalue. This solution is on a line. To get a two-dimensional solution, we select for the y -coordinates the best solution that is orthogonal to the first solution. This second solution is the eigenvector associated with the *second* smallest positive eigenvalue. These (x, y) coordinates provide a solution to the layout problem in the plane.

Marcoulides and Drezner [26] suggested the use of this layout algorithm to transform k -dimensional data to a two-dimensional scatter plot, while retaining the special structure of the data. They proposed to use the reciprocal of the distances between the points in the k -dimensional space as weights. In this way, points that are close to each other in the k -dimensional data will tend to be close in the solution of the layout problem. Let D_{ij} be the k -dimensional distances between points i and j . Marcoulides and Drezner [26] suggested to use $w_{ij} = D_{ij}^{-p}$ with a positive p as the weights in (2.1), and search for the best p using the golden section search. For each value of p , the correlation coefficient between the original distances D_{ij} and the calculated two-dimensional distances by the procedure d_{ij} is found, and the best p in $[0, 10]$ that maximizes this correlation coefficient is selected for implementation.

Marcoulides and Drezner [27] suggested the application of the MD procedure for cluster analysis with excellent results. They proposed to use the solution on a line (which is the projection of the scatter diagram on the x -axis). Clusters are identified as follows: the distances between successive points on the line are calculated and large distances between successive points constitute separators between clusters.

3. The quadratic assignment problem

The quadratic assignment problem (QAP) is considered to be one of the most difficult combinatorial optimization problems to solve. The problem is defined as follows. A set of n possible sites is given and n facilities are to be located on these sites, one facility at a site. Let c_{ij} be the cost per unit distance between facilities i and j and let d_{ij} be the distance between sites i and j . A high cost between two facilities means that we wish the two facilities to be close to one another. The cost f to be minimized over all possible permutations, calculated for an assignment of facility i to site $p(i)$ for $i = 1, \dots, n$, is

$$f = \sum_{i=1}^n \sum_{j=1}^n c_{ij} d_{p(i)p(j)}. \quad (3.1)$$

Optimal algorithms can solve relatively small problems. Recently, Anstreicher et al. [3], Hahn and Krarup [23], Nystrom [30], and Anstreicher and Brixius [2] report optimal solutions for problems with $n = 30$ to 36 facilities. Such optimal solutions are based on branch-and-bound algorithms which require “good” lower bounds. Gilmore [20] and Lawler [24] proposed the first lower bound based on the simple assignment problem. Anstreicher and Brixius [2] proposed a lower bound based on quadratic programming. Two lower bounds used by Hahn and Grant [21] and Hahn et al. [22] are based on a dual formulation. A dual formulation was suggested by Drezner in [13] and its implementation reported by Resende et al. in [32].

4 Mapping the convergence of genetic algorithms

Since optimal algorithms can solve only relatively small problems, considerable effort has been devoted to constructing heuristic algorithms. The first heuristic algorithm proposed for the solution of the QAP was CRAFT (Armour and Buffa [4]) which is a descent heuristic. More recent algorithms use metaheuristics such as Tabu search (Battiti and Tecchiolli [5]; Skorin-Kapov [34]; and Taillard [35]), simulated annealing (Burkard and Rendl [8]; Wilhelm and Ward [38]; and Connolly [10]), genetic algorithms (Ahuja et al. [1]; Fleurent and Ferland [18]; Tate and Smith [37]; and Drezner [15–17]), ant colony search (Gambardella et al. [19]), or specially designed heuristics (Drezner [14]; Li et al. [25]). For a complete discussion and list of references, see Burkard [7], Čela [9], Rendl [31], and Taillard [36].

3.1. The hybrid genetic algorithm. Genetic algorithms maintain a population of solutions. In order to create each generation, two parents are selected and merged to produce an offspring. If the produced offspring is better than the worst population member, the offspring replaces that member. The process continues for a prespecified number of generations. The best population member at the conclusion of the process is the solution of the genetic algorithm. Hybrid genetic algorithms apply an improving procedure on each offspring before considering it for inclusion in the population. Such an improvement procedure produces better offspring and the algorithm usually requires fewer generations to obtain quality solutions. The important components of a hybrid genetic algorithm are the merging process of two parents, and the postmerging improvement algorithm.

For the implementation of the genetic algorithm for the solution of the QAP, each solution (chromosome) is defined by the facilities assigned to sites #1, #2, ..., # n . The Hamming distance between two solutions is the *number* of facilities located at *different* sites. We use a population of 100 solutions. As the merging procedure, the “cohesive merging procedure” (Drezner [15]) is used. The idea behind the cohesive merging procedure is to select about half of the sites that are close to one another “(cohesive)” and assigning the facilities from the first parent to this cohesive set, and to assign the facilities from the second parent to the rest of the sites. For a complete description, the reader is referred to Drezner [15]. As the postmerging procedure, we use the “short” concentric Tabu search, modified by selecting a random number of levels. The concentric Tabu search was first presented by Drezner in [14] and was used as a postmerging procedure in hybrid genetic algorithms (Drezner [15–17]). The short version was used by Drezner in [16, 17] and gave excellent results.

The postmerging procedure is summarized below. The procedure starts with a solution termed the “center” solution and attempts to find a better solution by checking solutions at increasing Hamming distance from the center solution. This process can be viewed as searching in concentric circles centered at the center solution. The concentric Tabu search (Drezner [14]) stops once one application of the radial search fails to find a better solution. In the short concentric Tabu search, the maximum radius of the concentric searches is randomly generated at $[0.3n, 0.9n]$ which is less than the maximum possible Hamming distance between two solutions (n). The algorithm below applies between 3 and 9 “levels.” Each level is a concentric Tabu search, but if the search fails to produce a better solution, a new center solution is selected for the next level.

3.2. The postmerging procedure for the QAP. The procedure starts with a so-called “center” solution. The Hamming distance between permutation p and the center solution is Δp . The procedure proceeds by checking solutions with increasing Hamming distance.

- (1) Set a counter $c = 0$. Randomly generate the number of levels L in $[3, 9]$ (with probability of $1/7$ for each level).
- (2) Select R (the radius of the search) randomly in $[0.3n, 0.9n]$.
- (3) Set $\Delta p = 0$. sol_0 is the center solution. Empty the solutions sol_1 and sol_2 (the best found solutions for $\Delta p + 1$ and $\Delta p + 2$, resp.).
- (4) All pair exchanges of sol_0 are evaluated.
- (5) If the exchanged solution is better than the best found solution, the best found solution is updated and the rest of the exchanges are evaluated.
- (6) If the distance of an exchanged solution is Δp or lower, it is in the Tabu list. Therefore, it is ignored and the rest of the exchanges are evaluated. (In this way, we force the search away from the center solution.)
- (7) If its distance is $\Delta p + 1$ or $\Delta p + 2$, sol_1 or sol_2 is updated, if necessary.
- (8) If a new best found solution is found by scanning all the exchanges of sol_0 , the starting (center) solution is set to the new best found solution. Go to step (1).
- (9) Otherwise, $\text{sol}_0 = \text{sol}_1$, $\text{sol}_1 = \text{sol}_2$, and sol_2 is emptied. Set $\Delta p = \Delta p + 1$.
- (10) If $\Delta p \leq R$, go to step (4).
- (11) If $\Delta p = R + 1$, advance the counter $c = c + 1$, and
 - (i) if $c \leq L$ and is odd, use the best solution with depth R as the new center solution and go to step (2);
 - (ii) if $c \leq L$ and is even, use the best solution found throughout the scan (the previous center solution is not considered) as the new center solution and go to step (2);
 - (iii) if $c = L + 1$ stop and report the best found solution.

4. Analysis

Hybrid genetic algorithms start with a population of random solutions (each improved by a postmerging procedure) and keep improving the population members by entering better offspring and removing poorer population members. As the number of generations increases, the population members tend to cluster into groups, such that group members are “close” to one another.

In order to analyze this phenomenon, we first define a distance between population members. The Hamming distance is used. The distance between two population members is the number of variables which are different in the two solutions. Thus, two population members are at distance zero from one another if they are identical. Note that this distance measure satisfies the triangle inequality.

Suppose we perform a cluster analysis on a given population. The distance between every pair of population members is calculated and a scatter plot is generated using the MD procedure. The distance matrix is given as input to the MD procedure, and the result is a two-dimensional scatter diagram of the population members. Pairs of population members that are “close” to one another tend to be close to one another in the scatter

6 Mapping the convergence of genetic algorithms

diagram. We employ the weights $[D_{ij} - D_{\min} + 1]^{-p}$, where D_{ij} is the Hamming distance between population members i and j , and $D_{\min} = \min_{i \neq j} \{D_{ij}\}$.

We implemented this idea in the analysis of a hybrid genetic (memetic) algorithm for the solution of the quadratic assignment problem. Each solution of the quadratic assignment problem is a permutation of n facilities. Therefore, the distance between two solutions (permutations) can be between 0 (when the permutations are identical) and n . Note that a distance of 1 is impossible.

4.1. Properties of the Hamming distance for the QAP

THEOREM 4.1. *The expected distance between two random permutations is equal to $n - 1$.*

Proof. Let $P_n(k)$ be the probability that two random permutations of n elements have k identical elements. The number of permutations that have k members identical to permutation #1 is $n!P_n(k)$. It is clear that $n!P_n(k) = \binom{n}{k}(n-k)!P_{n-k}(0)$ leading to

$$P_n(k) = \frac{P_{n-k}(0)}{k!}. \quad (4.1)$$

By (4.1),

$$P_{n-1}(k-1) = \frac{P_{n-k}(0)}{(k-1)!} = kP_n(k). \quad (4.2)$$

Therefore,

$$\sum_{k=1}^n kP_n(k) = \sum_{k=1}^n P_{n-1}(k-1) = 1. \quad (4.3)$$

We showed that the expected number of identical elements in two random permutations is 1, which proves the theorem. \square

Theorem 4.1 provides us with a reference for comparison between the average distance among all population members and the expected distance if the population members were random. Thus, if the average distance between all pairs of population members is lower than $n - 1$, then the population is not random.

5. Computational experiments

We selected three QAP problems for analysis: Nug30 (Nugent et al. [29]) of 30 facilities for which the optimum solution of 6124 is known (Anstreicher et al. [3]), Sko56 and Sko100a (Skorin-Kapov [34]) of 56 and 100 facilities, respectively, for which the best known solutions of 34458 and 152002 are not proven optimum yet. We used a population of 100 and therefore the scatter diagram consists of 100 points. Each problem was solved using $50n$ generations, and the results after multiples of $10n$ generations were recorded and analyzed.

In Table 5.1 we report for each problem the minimum and average distances among population members, and the minimum and average values of the objective function for

Table 5.1. Distances between population members and objective function values.

Gen.	Nug30				Sko56				Sko100a			
	Distance		Objective		Distance		Objective		Distance		Objective	
	Min.	Aver.	Min.	Aver.	Min.	Aver.	Min.	Aver.	Min.	Aver.	Min.	Aver.
0	0	27.73	6134	6211.48	36	53.84	34512	34906.84	84	97.55	152368	153441.30
10n	0	26.98	6124	6160.00	2	45.35	34458	34501.96	2	41.95	152026	152097.70
20n	0	26.87	6124	6155.24	2	41.92	34458	34481.68	2	33.32	152026	152075.44
30n	0	26.83	6124	6152.54	2	41.36	34458	34479.04	2	30.32	152026	152072.46
40n	0	26.70	6124	6151.04	2	41.27	34458	34477.54	2	29.84	152026	152071.74
50n	0	26.72	6124	6149.94	2	41.55	34458	34477.24	2	29.77	152026	152071.58

all population members. The starting population of Nug30 (consisting of 100 population members) includes three pairs of identical population members (i.e., at a distance of zero from one another). These pairs of population members have an objective function values of 6146, 6172, and 6190, respectively. Since the hybrid genetic algorithm (Drezner [15–17]) does not allow into the population, the offspring that are identical to existing population members, no more identical population members are added to the population. After 50n generations, the worst population member has an objective function value of 6160, and thus two of the identical pairs were removed from the population, and only one of the three pairs is still a member of the population at the end of the process. The optimum solution of 6124 was obtained before 10n (300) generations are completed. It seems that the populations do not improve much after 300 iterations. The best known solution for Sko56 was also reached before 10n (560) iterations. The populations do not improve much after 20n generations. For Sko100a, the procedure obtained the value of 152026 which is slightly higher than the best known solution of 152002 after 10n generations as well. It should be noted that the best known solution for Sko100a was obtained frequently with other random seeds (see Section 5.2). The population also seems to have stabilized after 20n generations.

The average distance between population members generally declines as the number of generations increases. However it stabilizes after 20n – 40n generations. A random initial population is expected to have an average distance of $n - 1$ by Theorem 4.1. Since a postmerging procedure is applied on the initial population, the initial population is already somewhat clustered (average distance of 27.73 compared with expected of 29 for Nug30, 53.84 compared with 55 for Sko56, and 97.55 compared with 99 for Sko100a). In Figures 5.1, 5.2, and 5.3, we depict the scatter diagrams obtained by the MD procedure. The averages also confirm the scatter diagrams depicted in these figures. The scatter diagram of Nug30 (Figure 5.1) is the most scattered. Therefore, their average distance is not much lower than the expected distance for random populations. On the other hand, the scatter diagram of Sko100a (Figure 5.3) has one cluster of 97 population members. As expected, its average distance is the lowest when compared with the expected average of $n - 1$.

The starting population for Nug30 does not exhibit clear clustering. The successive scatter diagrams indicate “convergence” to five clusters. Note that the problem has exactly

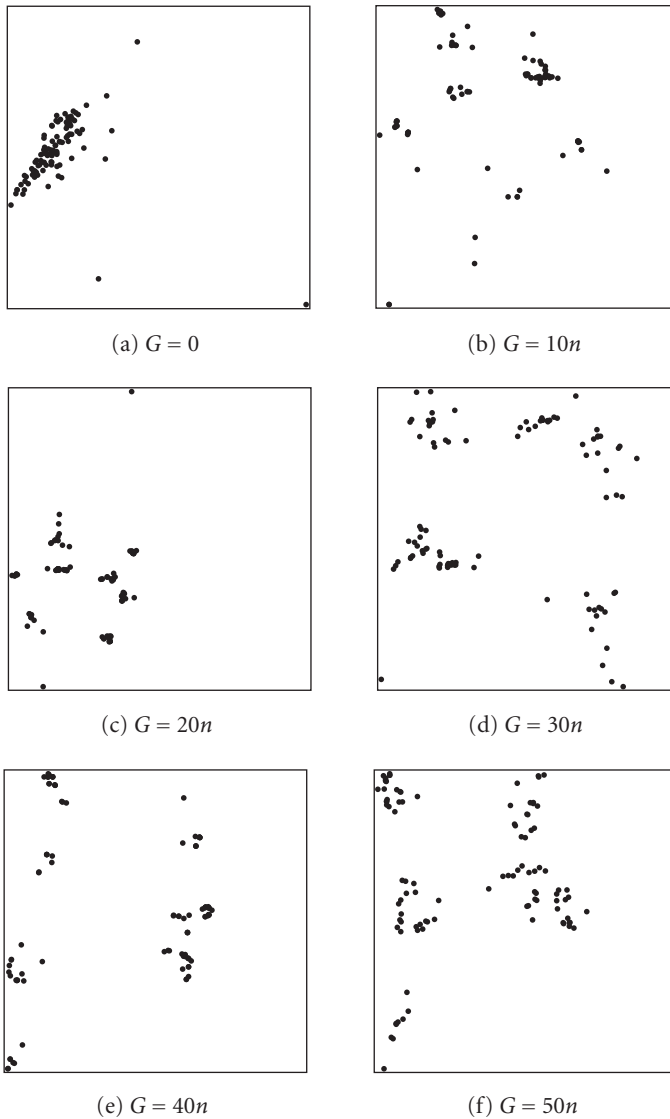


Figure 5.1. The scatter plots for Nug30.

four optimum solutions that are mirror images of one another and the Hamming distance between two optimum solutions is 30. It is important to note that if the scatter diagrams are projected on the x -axis, there are only two clusters. The separation between the two clusters is best for $G = 40n$.

Different diagrams are obtained for Sko56 and Sko100a. In Figure 5.2, we observe no clusters at the starting population (with two outliers). The amorphous “cloud” is no longer observed even for $G = 10n$. A projection on the x -axis for $G = 10n$ indicates that

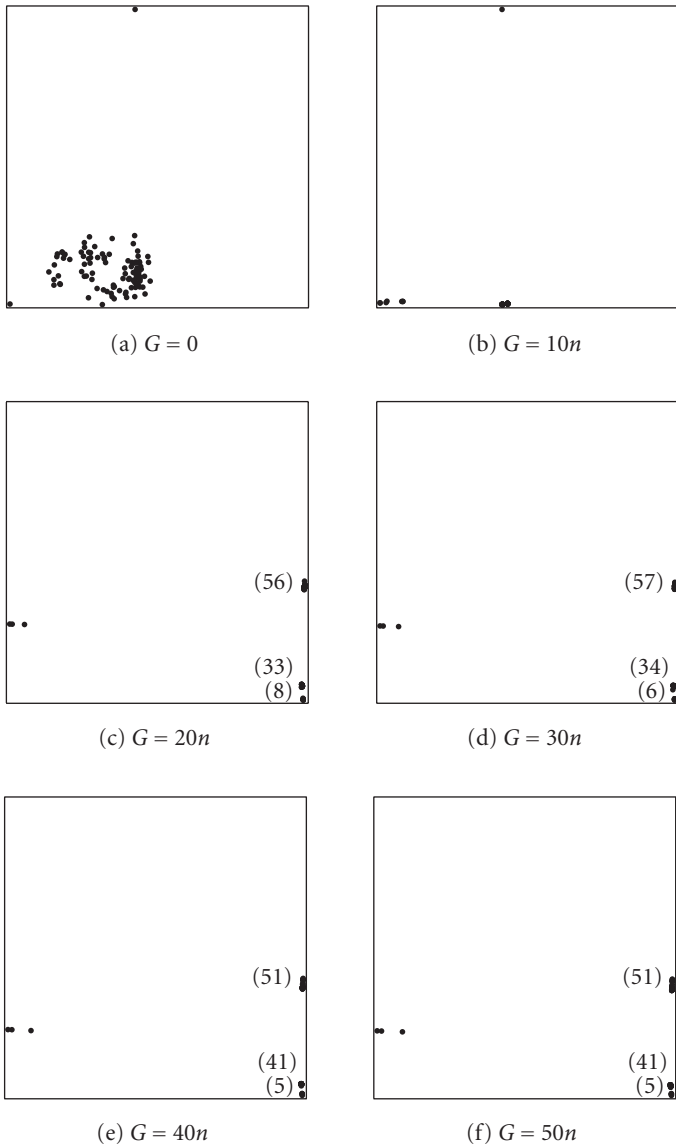


Figure 5.2. The scatter plots for Sko56.

there are two clusters. The general structure obtained for $G = 20n$ remains almost unchanged until $G = 50n$ generations are reached. The projections on the x -axis indicate two distinct clusters of 3 and 97 population members, respectively. The second cluster of 97 population members is divided into three clusters in the second dimension.

In Figure 5.3, we depict the scatter diagrams for Sko100a. The starting population does not exhibit any clusters. The projection on the x -axis indicates two clusters of 99 and 1

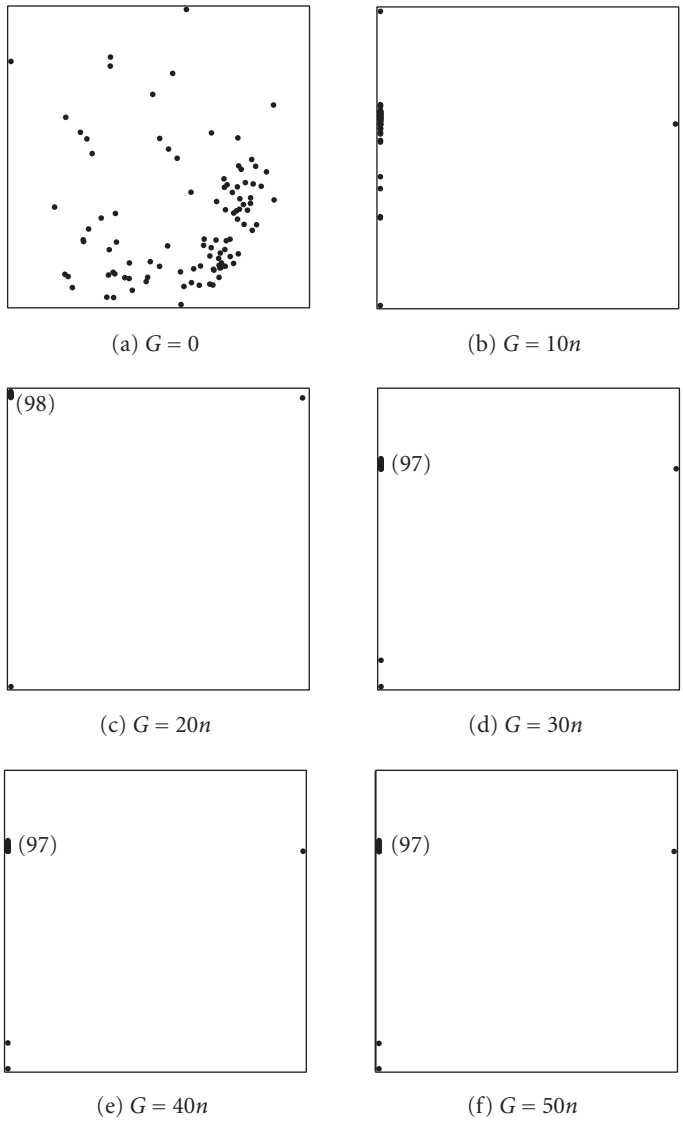


Figure 5.3. The scatter plots for Sko100a.

members each starting with $G = 10n$. From $G = 30n$ and upward, a cluster of 97 with 3 outliers is evident.

5.1. Further investigation of Sko56. The scatter plot for Sko56 (Figure 5.2) shows three main clusters and three outliers. We further investigated the Sko56 problem by recreating the scatter diagram by removing the 3 outliers and running the MD procedure on the remaining 97 population members for $G = 50n$ so that the internal structure of the three

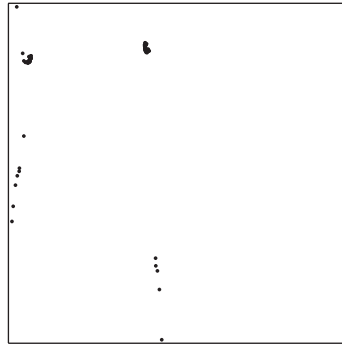


Figure 5.4. The final scatter of Sko56 without the 3 outliers.

Table 5.2. Objective function values for various clusters.

$G = 10n$					
Cluster size	1	3	4	36	56
Average	34524.0	34497.3	34513.5	34502.4	34500.7
Minimum	34524	34472	34508	34462	34458
Maximum	34524	34512	34518	34526	34526
$G = 20n$					
Cluster size	3	8	33	56	—
Average	34466.0	34478.0	34483.7	34481.9	—
Minimum	34462	34462	34458	34458	—
Maximum	34472	34490	34494	34494	—
$G = 50n$					
Cluster size	3	5	41	51	—
Average	34466.0	34471.2	34478.2	34477.7	—
Minimum	34462	34462	34458	34458	—
Maximum	34472	34480	34486	34486	—

main clusters can be observed. The resulting scatter diagram is depicted in Figure 5.4. The projection on the x -axis indicates two distinct clusters. However, the clusters of 41 and 5 members appear as two clusters in the second dimension and the cluster of 51 members reveals a “core” of 42 population members and 9 members in its vicinity with 7 of them possibly defining another cluster.

Another interesting experiment is the analysis of the values of the objective functions for the different clusters. In Table 5.2, we report the statistics for the members of each cluster for $G = 10n, 20n, 50n$. The clusters are depicted in Figure 5.2. For $G = 10n$, the cluster of one is at the top of the scatter diagram. The cluster of 56 is depicted as two or three close points at the bottom-left corner, and the close-by cluster is the cluster of 36, followed by clusters of 3 and 4.

It is clear that the cluster of one has almost the worst value of the objective function in the population (34524 compared with the worst value of 34526, see Table 5.2). It is removed from the population in a few generations. It is interesting that the three outliers have the best average of the value of the objective function of all clusters. Many more iterations are required before any of the members of this cluster are removed from the population. Fortunately, the best known solution is in the bigger clusters. However, it is conceivable that the best solution could fall near the cluster of three. If so, the algorithm will miss it, because it is unlikely (probability of 0.0006 per generation) that both parents will be selected from this cluster to augment it. Once the structure of the clusters is known, one can modify the parent selection accordingly in order to generate better offspring.

5.2. Is avoiding identical population members helpful? Most genetic algorithms do not check whether newly generated offsprings are identical to existing population members before considering them for inclusion in the population. Such a provision is proposed and applied by Drezner in [15–17] with good results.

An offspring generated by two identical population members is identical to its parents (regardless of the merging process). The postmerging procedure cannot improve it (it could not further improve its parents at the time they were generated), and therefore the offspring joins the population and more identical members are added to the population. As the group of identical members increases in number, the likelihood of merging two identical parents increases. After some generations, more and more identical parents are merged and the population may consist of all identical members and no improvement is possible. We believe that the reason other genetic algorithms do not employ this provision is that researchers are under the impression that generating an identical offspring is very unlikely and one can ignore such a possibility.

The new tool proposed in this paper can be used to analyze the effect of such a provision. We ran the hybrid genetic 10 times each for Nug30, Sko56, and Sko100a with and without the provision (of not adding offspring identical to existing population members). With this provision in place, the optimum solution for Nug30 and the best known solution for Sko56 were found in all 10 runs. The best known solution for Sko100a was found 3 times out of 10 with the average solution being just 0.015% over the best known solution. The same hybrid genetic algorithm without the provision also found the optimum solution to Nug30 in all 10 runs, but found it only 4 times out of 10 for Sko56 with the average solution being only 0.008% above the best known solution. The best known solution of Sko100a was found four times out of ten but with the average solution being 0.022% over the best known solution.

In many of these runs, the population after $50n$ generations consisted of 100 identical members. In many of these cases, all population members are inferior to the best known solution. This clearly indicates an early convergence to an inferior local minimum. In Figure 5.5 we depict the clusters for Nug30. For $G = 50n$, all population members are optimal. The average Hamming distances are 27.73 for $G = 0$, 25.32 for $G = 10n$, 19.03, 19.54, 21.55, and 3.42 for the next checkpoints, respectively. Contrary to the scatter diagrams in Figure 5.1, convergence is observed to four clear clusters, each with identical optimal members. The cluster on the left consists of 94 members, the cluster in the

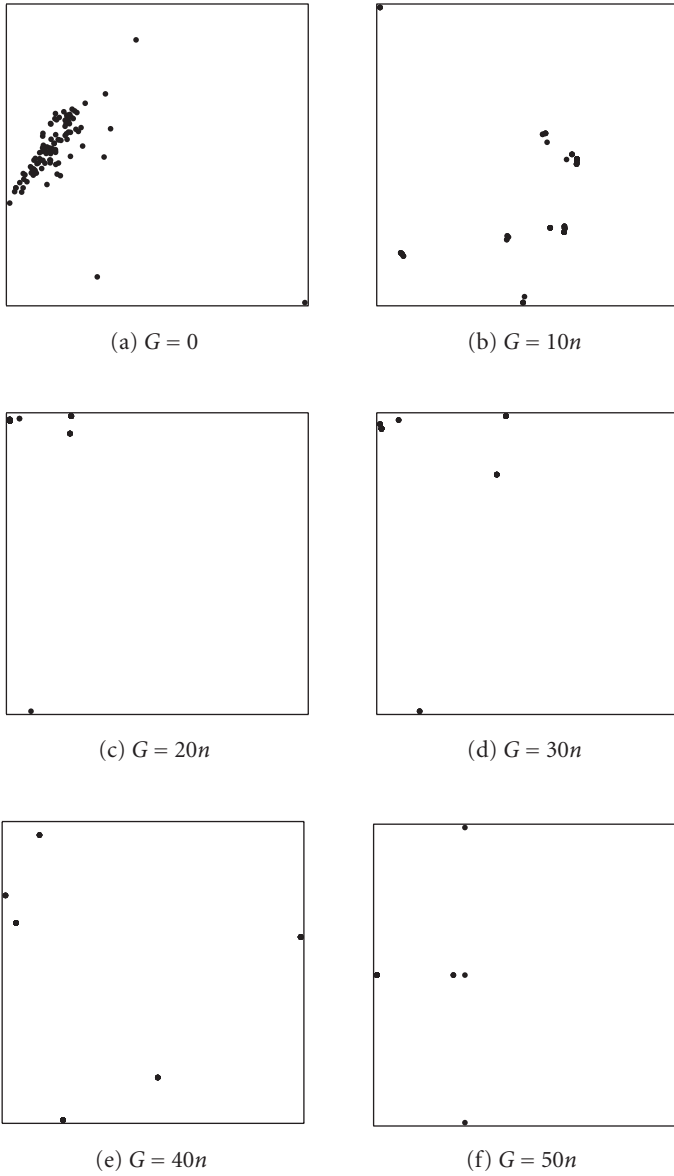


Figure 5.5. The scatter plots for Nug30 without the provision.

middle-right consists of 4 members, and the two clusters (one on top and one at the bottom) have one member each. We are “lucky” in this case that early convergence was to the optimum and not to an inferior local optimum. However, we were not that lucky in six runs for Sko56 and six runs of the Sko100a problem.

6. Conclusions

In this paper we proposed to investigate the structure of the population in genetic algorithms by applying the MD cluster-analytic-type procedure. By analyzing the results, valuable information and insight can be gained into the behavior and characteristics in the population as the genetic algorithm progresses. As an illustration, we analyzed the inclusion of the provision that an offspring identical to an existing population member is ignored rather than being added to the population. The resulting scatter plots show the early convergence of the algorithm, when this provision is not implemented. We also observed that the population becomes stagnant after about $20n$ generations and there is no need to perform $50n$ generations for these test problems.

In future research, we advocate use of this tool in order to construct better and more efficient genetic algorithms. Since the calculations involved in this procedure are very quick, the parameters controlling the genetic procedures can be modified during the progression of the genetic algorithm according to the results of such analyses. As we observed in our test problems, a stopping criterion based on the scatter diagrams can be established for genetic algorithms.

References

- [1] R. K. Ahuja, J. B. Orlin, and A. Tiwari, *A greedy genetic algorithm for the quadratic assignment problem*, Computers & Operations Research **27** (2000), no. 10, 917–934.
- [2] K. M. Anstreicher and N. W. Brixius, *A new bound for the quadratic assignment problem based on convex quadratic programming*, Mathematical Programming **89** (2001), no. 3, 341–357.
- [3] K. M. Anstreicher, N. W. Brixius, J.-P. Goux, and J. Linderoth, *Solving large quadratic assignment problems on computational grids*, Mathematical Programming **91** (2002), no. 3, 563–588.
- [4] G. C. Armour and E. S. Buffa, *A heuristic algorithm and simulation approach to relative location of facilities*, Management Science **9** (1963), 294–309.
- [5] R. Battiti and G. Tecchiolli, *The reactive tabu search*, ORSA Journal on Computing **6** (1994), 126–140.
- [6] J. E. Beasley, *Population heuristics*, Handbook of Applied Optimization (P. M. Pardalos and M. G. C. Resende, eds.), Oxford University Press, Oxford, 2002, pp. 138–157.
- [7] R. E. Burkard, *Locations with spatial interactions: the quadratic assignment problem*, Discrete Location Theory (P. B. Mirchandani and R. L. Francis, eds.), Wiley-Intersci. Ser. Discrete Math. Optim., Wiley, New York, 1990, pp. 387–437.
- [8] R. E. Burkard and F. Rendl, *A thermodynamically motivated simulation procedure for combinatorial optimization problems*, European Journal of Operational Research **17** (1984), no. 2, 169–174.
- [9] E. Çela, *The Quadratic Assignment Problem: Theory and Algorithms*, Combinatorial Optimization, vol. 1, Kluwer Academic, Dordrecht, 1998.
- [10] D. T. Connolly, *An improved annealing scheme for the QAP*, European Journal of Operational Research **46** (1990), no. 1, 93–100.
- [11] Z. Drezner, *DISCON: a new method for the layout problem*, Operations Research **28** (1980), 1375–1384.
- [12] ———, *A heuristic procedure for the layout of a large number of facilities*, Management Science **33** (1987), 909–915.
- [13] ———, *Lower bounds based on linear programming for the quadratic assignment problem*, Computational Optimization & Applications **4** (1995), no. 2, 159–165.
- [14] ———, *Heuristic algorithms for the solution of the quadratic assignment problem*, Journal of Applied Mathematics and Decision Sciences **6** (2002), no. 3, 163–173.

- [15] ———, *A new genetic algorithm for the quadratic assignment problem*, *INFORMS Journal on Computing* **15** (2003), no. 3, 320–330.
- [16] ———, *Compounded genetic algorithms for the quadratic assignment problem*, *Operations Research Letters* **33** (2005), no. 5, 475–480.
- [17] ———, *The extended concentric tabu for the quadratic assignment problem*, *European Journal of Operational Research* **160** (2005), no. 2, 416–422.
- [18] C. Fleurent and J. Ferland, *Genetic hybrids for the quadratic assignment problem*, *Quadratic Assignment and Related Problems* (New Brunswick, NJ, 1993), DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 16, American Mathematical Society, Rhode Island, 1994, pp. 173–187.
- [19] L. M. Gambardella, E. D. Taillard, and M. Dorigo, *Ant colonies for the quadratic assignment problem*, *Journal of the Operational Research Society* **50** (1999), no. 2, 167–176.
- [20] P. C. Gilmore, *Optimal and suboptimal algorithms for the quadratic assignment problem*, *Journal of the Society of Industrial and Applied Mathematics* **10** (1962), no. 2, 305–313.
- [21] P. M. Hahn and T. L. Grant, *Lower bounds for the quadratic assignment problem based upon a dual formulation*, *Operations Research* **46** (1998), no. 6, 912–922.
- [22] P. M. Hahn, W. L. Hightower, W. P. Adams, and M. Guignard-Spielberg, *A level-2 reformulation-linearization technique bound for the quadratic assignment problem*, to appear in *European Journal of Operational Research*.
- [23] P. M. Hahn and J. Krarup, *A hospital facility problem finally solved*, *Journal of Intelligent Manufacturing* **12** (2001), no. 5–6, 487–496.
- [24] E. L. Lawler, *The quadratic assignment problem*, *Management Science* **9** (1963), 586–599.
- [25] Y. Li, P. M. Pardalos, and M. G. C. Resende, *A greedy randomized adaptive search procedure for the quadratic assignment problem*, *Quadratic Assignment and Related Problems* (P. M. Pardalos and H. Wolkowicz, eds.), DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 16, American Mathematical Society, Rhode Island, 1994, pp. 237–261.
- [26] G. A. Marcoulides and Z. Drezner, *A procedure for transforming points in multi-dimensional space to two-dimensional*, *Educational and Psychological Measurement* **53** (1993), 933–940.
- [27] ———, *A procedure for detecting pattern clustering in measurement designs*, *Objective Measurement: Theory into Practice* (M. Wilson, K. Draney, and G. Engelhard Jr., eds.), vol. 5, Ablex, New Jersey, 2000, pp. 287–302.
- [28] P. Moscato, *Memetic algorithms*, *Handbook of Applied Optimization* (P. M. Pardalos and M. G. C. Resende, eds.), Oxford University Press, Oxford, 2002.
- [29] C. E. Nugent, T. E. Vollman, and J. Ruml, *An experimental comparison of techniques for the assignment of facilities to locations*, *Operations Research* **16** (1968), 150–173.
- [30] M. Nystrom, *Solving certain large instances of the quadratic assignment problem: Steinberg's examples*, Working paper, California Institute of Technology, California, 1999.
- [31] F. Rendl, *The quadratic assignment problem*, *Facility Location: Applications and Theory* (Z. Drezner and H. Hamacher, eds.), Springer, Berlin, 2002, pp. 439–457.
- [32] M. G. C. Resende, K. G. Ramakrishnan, and Z. Drezner, *Computing lower bounds for the quadratic assignment problem with an interior point algorithm for linear programming*, *Operations Research* **43** (1995), no. 5, 781–791.
- [33] S. Salhi, *Heuristic search methods*, *Modern Methods for Business Research* (G. A. Marcoulides, ed.), Lawrence Erlbaum Associates, New Jersey, 1998.
- [34] J. Skorin-Kapov, *Tabu search applied to the quadratic assignment problem*, *ORSA Journal on Computing* **2** (1990), no. 1, 33–45.
- [35] E. D. Taillard, *Robust taboo search for the quadratic assignment problem*, *Parallel Computing* **17** (1991), no. 4–5, 443–455.
- [36] ———, *Comparison of iterative searches for the quadratic assignment problem*, *Location Science* **3** (1995), no. 2, 87–105.

16 Mapping the convergence of genetic algorithms

- [37] D. M. Tate and A. E. Smith, *A genetic approach to the quadratic assignment problem*, Computers & Operations Research **22** (1995), no. 1, 73–83.
- [38] M. R. Wilhelm and T. L. Ward, *Solving quadratic assignment problems by simulated annealing*, IIE Transactions **19** (1987), 107–119.

Zvi Drezner: College of Business and Economics, California State University-Fullerton, Fullerton, CA 92834, USA

E-mail address: zdrezner@fullerton.edu

George A. Marcoulides: College of Business and Economics, California State University-Fullerton, Fullerton, CA 92834, USA

E-mail address: gmarcoulides@fullerton.edu