Unequal Area Facility Layout Using Genetic Search

David M. Tate, Senior Member IIE
and
Alice E. Smith, Senior Member IIE[1]

Department of Industrial Engineering
1031 Benedum Hall
University of Pittsburgh
Pittsburgh, PA 15261
412-624-5405
412-624-9831 (Fax)

Accepted to *IIE Transactions*, April 1994.

[1] Corresponding author.

Unequal Area Facility Layout Using Genetic Search

Why This Paper Is Important

Unequal area facility layout problems comprise a class of extremely difficult and widely-applicable optimization problems, arising in many diverse areas. There are many variations on the basic formulation, involving alternative objective functions, side constraints, distance metrics, cost measures, and facility shapes. Applications include industrial facility design, hospital layout, VLSI component placement, and warehouse organization.

In this paper, we present a heuristic search methodology, based on genetic algorithms (GA), for unequal area layout. We apply this methodology to several standard test problems from the literature, and show that the GA method gives solutions which are much better than the best previously reported solutions. We also show how, through the use of penalty-directed search, it is possible to find very good solutions to problems with difficult-to-satisfy side constraints, and to perform multi-criterion optimization with respect to cost measures that have been considered incommensurable in the past. The methodology presented is not intrinsically restricted to layout problems, but could be extended to other hard combinatorial problems.

The GA/penalty method's ability to find improved solutions to known problems, together with the ability to address problems with ill-behaved cost functions, multiple objectives, and/or side constraints, constitutes a significant contribution to the state of the art in facility layout. Furthermore, GA can be implemented to take advantage of parallel hardware to an extent not possible for other heuristic optimization methods.

Unequal Area Facility Layout Using Genetic Search

Abstract

This paper applies genetic optimization with an adaptive penalty function to the shape-constrained unequal area facility layout problem. We implement a genetic search for unequal area facility layout, and show how optimal solutions are affected by constraints on permitted department shapes, as specified by a maximum allowable aspect ratio for each department. We show how an adaptive penalty function can be used to find good feasible solutions to even the most highly constrained problems. We describe our genetic encoding, reproduction and mutation operators, and penalty evolution strategy. We provide results from several test problems that demonstrate the robustness of this approach across different problems and parameter settings.

Unequal Area Facility Layout Using Genetic Search

1. Introduction to the Unequal Area Facility Layout Problem

Facility Layout Problems (FLP) are a family of optimization problems involving the partition of a planar region of known dimensions (usually rectangular) into departments of known area, so as to minimize the costs associated with projected interactions between these departments. These costs may reflect transportation costs (including costs associated with the construction of a material-handling system), or preferences regarding adjacencies among departments. Such problems occur in many design environments, including manufacturing cell layout, hospital layout, and service center layout among others. Related problems arise in other engineering contexts such as VLSI placement and routing. All of these problems are known to be NP-hard [17].

The version of FLP we are concerned with was originally formulated by Armour and Buffa [1] as follows: we are given a rectangular region[2] with fixed dimensions H×W, and a collection of n required "departments" of specified area, whose total area is HW. To each ordered pair of departments (j,k) is associated a traffic flow F(j,k). The objective is to partition the region into n subregions, of appropriate area, so as to minimize the sum

$$\sum_{j \neq k}^{n} F(j,k) \cdot \delta(j,k,\Pi)$$

where $\delta(j,k,\Pi)$ is the distance (using a pre-specified metric) between the centroid of department j and the centroid of department k in the partition Π.

1.1. Equal area Departments

If all departments are of equal area, or can be physically interchanged without altering the overall adjacency or distance relationships among the remaining departments, it is easy to specify in advance a finite number of potential sites for these departments to occupy. Given this, the

[2]All of the techniques we will discuss can be extended to general polygonal regions with polygonal holes, by a straightforward but tedious process described in Tam [24].

layout problem can be modeled as a quadratic assignment problem (QAP) when the costs associated with flow between departments are assumed to be linear with respect to distance traveled and quantity of flow. Since Koopmans and Beckmann [13], more than fifty different articles have appeared in the literature, recommending heuristic procedures or approximate algorithms for finding good solutions to QAP formulations. Tate and Smith [19, 25] successfully applied genetic algorithms to the QAP test problems of Nugent [14], Scriabin and Vergin [18] and Steinberg [21].

1.2. Unequal Areas

When departments have differing areas, we can no longer treat the problem as assigning departments to n distinct centroid locations. Instead, the locations of the centroids will depend on the exact configuration selected, making QAP formulations of the unequal area problem less tractable than their equal area counterparts. The best known large test problem for the unequal area facility layout problem is that of Armour and Buffa, who devised a 20 department problem with symmetrical flow matrix using the rectilinear distance metric [1]. They approached this problem by requiring all departments to be made up of contiguous rectangular "building blocks," and then applied departmental adjacent pairwise exchange. (Exchanging two adjacent departments does not affect the centroids of the other departments.) Tong approached this problem by assuming rectangular departmental shapes placed in bays [26]. Other unequal area test problems have been studied, including 12 and 14 department problems by Bazaraa [2], and a 10 department problem by van Camp et al. [27]. We illustrate our genetic approach on these test problems, as described in the following sections.

The primary difficulties associated with unequal area layout have to do with the vast number of possible physical layouts, and with the existence of many locally optimal layouts that are poor compared to the global optimum layout. For such a problem, one might expect parallel search methods to perform better than strictly serial searches, and randomized search methods to perform better than greedy or enumerative searches. Genetic algorithms combine both of these attributes in a parallel, stochastic heuristic.

2. Genetic Algorithms and Their Properties

Genetic algorithms (GAs) were described by Holland [10] and DeJong [5] as a family of global optimization heuristics motivated by the process of natural selection in biological systems. Under the GA paradigm, rather than generating a sequence of candidate solutions one at a time, a population of candidate solutions is maintained. The primary distinguishing features of GA are an encoding, an evaluation mechanism, a breeding mechanism, a mutation mechanism, and a culling mechanism. The encoding is a data structure which describes a unique feasible solution in a compact representation. To give a biological analogy, the encoding is like a chromosome, on which specific genes describe various aspects of the individual in question. The evaluation mechanism takes the encoding of an individual solution and computes the value of that solution with regard to the objective(s) of the optimization. The breeding mechanism is a procedure which produces a new encoding using two "parent" encodings from the current population as its source of "genes". The mutation mechanism is an algorithm for perturbing an encoding to produce a "nearby" encoding, and is used to help maintain a diverse gene pool from which to construct new solutions. The culling mechanism is a procedure for removing known solutions from the population and replacing them with new solutions.

To summarize the important aspects of genetic search:

1. The search is highly parallel, with each population member defining many different possible search directions. Potentially, GA search could be implemented extremely efficiently on massively parallel hardware.

2. No special information about the solution surface, such as gradient or local curvature, need be identified. The objective function need not be smooth, continuous or unimodal.

3. Genetic algorithms have proven to be fairly robust under varying parameter settings and problem particulars. As long as solutions with similar encodings do not have highly variant objective function values, genetic algorithms usually find near optimal solutions.

Genetic algorithms have been applied to many combinatorial optimization problems, including gas and oil pipeline flow [6, 8], job shop scheduling [3, 22], the traveling salesman problem [28], and sizing of communications networks [4]. The technique has also been applied to constrained combinatorial optimization problems such as bin packing [11] and scheduling [23]. While a GA approach requires the selection of parameters governing selection, breeding and mutation, published research has supported the claim that the effectiveness of the GA methodology is insensitive to the exact values of these parameters.

3. Using Genetic Algorithms for Unequal Facility Layout

3.1. Genetic Encoding and Operators

To effect search, departmental shapes must be restricted to some subset of possible shapes. We used the flexible bay structure developed by Tong [26]. The pre-specified rectangular area is divided in one direction into bays of varying width. Each bay is then divided into rectangular departments of equal width but different length. The bays are flexible in that their widths will vary with their number and contents. We chose to encode flexible bay solutions on two distinct chromosomes. The first chromosome carries a permutation of the integers 1 through n, where n is the number of departments. This sequence represents the sequence of departments, bay by bay, read from top to bottom, left to right. The second chromosome contains an encoding of the number of bays, and where in the sequence the breaks between bays occur. The encoding is shown in Fig. 1.

For breeding, we used a variant of uniform crossover [15]. Each location in the offspring's sequence is occupied by the department in the corresponding location from one or the other parent with equal probability, so that all common locations in the parents are carried over to the child. Conflicts are then resolved to ensure that each department occurs exactly once in the offspring's encoding. The number and location of bay breaks in the solution is taken without change from one parent or the other, with equal probability. This breeding mechanism is shown in Fig. 2. Parents are selected based on the following: a uniform random number between 1 and $\sqrt{n}$ (where n is the population size) is chosen, then squared. The result is truncated and taken to be the rank of the parent to be selected (where string zero is the fittest string in the population). This selection gives reproductive preference to better ranked solutions, while allowing all solutions some probability of reproduction.

For mutation we defined three operators: one which splits an existing bay into two adjacent bays, one which merges two adjacent bays into one (by concatenation), and one which reverses a subsequence of the departments. Half of all mutations affect only the bay chromosome, and half affect only the sequence chromosome. Of those mutations altering the number of bays, half increase the number of bays and half decrease that number, so as to provide long-term stability in the tendencies of the population. Mutation and breeding were separate, independent processes.

3.2. Evolution Parameters

We ran several trial searches to determine a good set of evolution parameters, varying population size from 10 to 50, mutation rates from 1% to 50%, and examining replacement of mutated solutions with their mutants versus culling of the worse solutions in the population to make room for mutants. Although all parameter settings gave reasonable results, we settled on the overall best variation of a population size of 10, a mutation rate of 50%, and culling. We found that culling requires a substantially higher probability of mutation to ensure population diversity. One child per reproduction cycle was created[3], and always entered the population regardless of its solution quality. Generated solutions, i.e. mutants and children, were not guaranteed to be unique. Solutions in the initial population were randomly generated with $\sqrt{n}$ bays on average. Since genetic optimization is stochastic, we performed 10 runs for each combination of parameters, differing each only by its random number seed.

3.3 The Adaptive Penalty Function

The unequal area FLP as formulated by Armour and Buffa is not shape constrained; that is, no minimum side length or maximum aspect ratio is specified for any department. Armour and Buffa's interchange approach used a given initial layout, so that extreme configurations were

[3] A "steady state" or "incremental" GA, as opposed to a generational GA where multiple children are created and replace the current population.

not reachable from their initial configuration. Furthermore, their unit "building blocks" impose a *de facto* lower bound on side length. Constructive heuristics, however, can achieve much lower costs by aligning departmental centroids in stacks of long, narrow departments.

To ensure realistic layouts, we imposed constraints of varying severity on the maximum allowable aspect ratio (α) for each department. Aspect ratio is defined by:

$$\alpha = \max\{L_i, W_i\} / \min\{L_i, W_i\}.$$

(where L_i is the length and W_i is the width of department i).

As the maximum allowable aspect ratio becomes smaller, the problem becomes more constrained, and feasible solutions become harder to find. Table 1 shows the percentage of feasible solutions per 100,000 random solutions generated for the Armour and Buffa test problem for various maximum allowed aspect ratios. For high constraint levels, even finding a feasible solution is a difficult task for any heuristic.

To handle these additional constraints, we developed an adaptive penalty function for highly constrained genetic search [20]. (Static penalty functions for GAs were suggested by Goldberg [7] and elaborated upon by Richardson et al. [16]). The penalty function uses observed population data during evolution to adjust the severity of the penalty being applied to infeasible solutions.

Table 1. Percent Feasible of 100,000 Randomly Generated Solutions.

α	% Feasible	Best Feasible	α	% Feasible	Best Feasible
1000	100.00	2897.1	7	2.4	8705.6
50	87.3	5612.5	5	0.4	8705.6
25	53.1	6364.4	4	0.1	8705.6
15	22.7	7695.0	3	0.005	10190.3
10	9.6	7796.2	2 and Below	0.0	None Found

We noted that the degree of infeasibility of any one department is less important to the search process than the number of departments which are infeasible. For example, a solution in which more than half of all departments are slightly infeasible in shape might require extensive modifications in order to yield a feasible solution, while a solution with one extremely infeasible department might be made feasible simply by shifting that department into an adjoining bay. The object of the penalty function is to find feasible solutions; infeasible solutions are attractive only to the extent that they are likely to breed or mutate good feasible solutions. Accordingly, we scaled our penalty function so that a solution with one infeasible department and the best known objective function value would be considered equally promising with the best known feasible solution. This led to the form

$$p(n_i) = (n_i)^k (V_{feas} - V_{all})$$

where n_i is the number of infeasible departments, V_{feas} is the best feasible objective function value yet found, V_{all} is the best overall objective function value yet found, and the exponent k is a parameter which adjusts the "severity" of the penalty function. After experimentation, we choose k = 3, although performance of the adaptive penalty function was robust across k ranging from 1 to 5 [20]. We emphasize that the purpose of the penalty function is to guide the search to include near feasible solutions, not to replace the constraints of the original problem formulation. The value of our penalty alters over evolution, through the values V_{feas} and V_{all}, in a sense learning how highly-constrained this problem instance is without manual tuning. Since, for problems with difficult shape constraints finding any feasible solution can be nearly as difficult as finding near-optimal solutions, heuristics which always preserve feasibility at intermediate points in the search will be unable to operate effectively. Our adaptive penalty method, on the other hand, found feasible solutions to even the most highly-constrained problem instances.

4. Computational Results

4.1 The Smaller Test Problems

We selected the most widely known unequal area facility layout test problems for study. The smaller test problems have 10 through 14 departments, all with symmetric flows, as published in [2, 9, 27]. For the van Camp et al. [27] problem we used a minimum side length of 5 as specified in that paper. For the Bazaraa [2] problems we used a minimum side length of 1, as used by van Camp et al. in their study. We note that the Bazaraa problems had areas within the facility which were not assigned to departments, and the first test problem (12 departments) had a notch in the lower right hand corner which was unavailable for assignment to a department. We modeled the notch as one department with area = 2 and no interactions. The excess area was handled in two ways. First, the two smallest departments (each of unit area) were temporarily enlarged to area of 2 and minimum side length of 1. Second, three dummy departments of unit dimension with no interactions or shape restrictions were added. This brought the 12 department problem up to 16 departments for our formulation. The second Bazaraa problem of 14 departments specifies that department 14 be of unit area and have no interactions; we allowed this department to take any shape.

Table 2 shows our results for these three test problems, along with the results reported by previous researchers. Since the GA approach is stochastic, we show the best, mean and worst solution found over 10 runs. The GA approach dominates all other published results, even with the worst of 10 runs. Our best solutions improved upon the previously published best solutions by 16.3%, 16.2% and 19.9% respectively. Since we had ceased search after approximately 60,000 solutions had been generated, we decided to extend the search for only the best GA run for each of the three problems. These longer runs continued until approximately 300,000 solutions had been generated. While the 12 department Bazaraa problem was not improved upon, the other two problems were. Our new solution to the van Camp problem had a cost of 20320.5. Our new solution to the 14 department Bazaraa problem had a cost of 5077.0. When incremental improvement is desired, this strategy may be useful: begin with multiple short runs, then extend the run of the best solution. For the 12 department Bazaraa problem, we obtained further improvement by making obvious adjustments to the given layout in the course of restoring the 12 departments to their original areas. Our hand-modified solution had a cost of 8768.2. The Appendix shows our best solution to each of the three test problems as well as the adjusted solution for the first Bazaraa problem.

Table 2. Results for the van Camp and Bazaraa Test Problems.

Solution	10 Dept. van Camp	12 Dept. Bazaraa	14 Dept. Bazaraa
GA Best	20472.2	8861.0 (8768.2)	5080.1
GA Mean	21745.7	9509.9	5318.9
GA Worst	23612.6	9894.3	5506.8
van Camp's NLT [27]	24445	11910	6875
Bazaraa [2]	-	14079	8170.5
Hassan's PLANET [9]	-	11664-11808	6399-6480
Hassan's SHAPE [9]	-	10578-11140	6339-6462

4.2 The Armour and Buffa Problem

The largest well known unequal area test problem is that of Armour and Buffa [1], a 20 department problem with symmetrical flows (note that we have corrected the flow matrix as originally published in [1] following Scriabin and Vergin [18] and Huntley and Brown [12]). As mentioned earlier, Armour and Buffa did not fix a minimum side length or maximum allowable aspect ratio; however from their solutions it can be seen that their maximum aspect ratio was approximately 3 (department M). The maximum allowable aspect ratios we tested for this problem were, in ascending constrainedness, α = 1000 (essentially unconstrained), 50, 25, 15, 10, 7, 5, 4, 3, 2, and 1.75. We also altered our objective function to search for minimum feasible aspect ratio, and this search yielded α = 1.70667. Using α = 1.70667, we resumed the search for best layout, as reported in Table 3.

Table 3 lists the mean, best and worst solutions found over the 10 runs for each degree of constrainedness for the Armour and Buffa problem. As degree of constrainedness increased, the mean best solution found increased because departments had to assume a squarer shape, forcing centroids apart. To compare results, Armour and Buffa reported a best solution found of 7862.09 [1]; our worst solution for α = 3 improves upon Armour and Buffa's best by 15.7%, while our best solution improves upon their's by 25.8%. Tong reported a best solution of 5204.5 using the flow matrix as originally published, which omitted one direction of the heaviest single flow [26]. She used minimum side lengths roughly equivalent to an α of 10. Our best solution for α = 10 improved upon Tong by 11.0%. The researchers cited in section 4.1 did not report results for the Armour and Buffa problem. The Appendix shows the layout for our best solutions for selected values of α. Note that, in general, square shapes for large departments must be forced by tight constraints.

Table 3. Solutions Over 10 Runs of the Armour and Buffa Test Problem.

α	Mean	Best	Worst
1000	1639.6	1638.5	1648.7
50	3062.2	3009.5	3299.1
25	4029.4	3635.9	4627.2
15	4762.4	4296.1	5352.2
10	5094.5	4633.3	5930.1
7	5479.6	5255.0	6144.5
5	6045.6	5524.7	6539.3
4	6001.0	5743.1	6492.3
3	6285.3	5832.6	6625.3
2	6669.8	6171.1	7226.5
1.75*	7638.8	7205.4	8085.8
1.70667#	7110.8	6662.9	7585.8

* For α = 1.75, only 3 runs found feasible solutions.
\# For α = 1.70667, only 8 runs found feasible solutions.

4.3 Computational Effort

To compare computational effort, we list the number of solutions generated before finding the best solution and the number of solutions generated per CPU second. All CPU times reported were using a DEC 3000-400 AXP workstation. The number of solutions generated before finding the best solution had an upper bound of 60,000 for the three smaller test problems, and 500,000 for the Armour and Buffa problem with α = 5. These were the limits we imposed on each run. Table 4 shows the best, mean and worst number of solutions required to find the best in each run, and the average number of solutions generated per CPU second. CPU effort grows roughly proportional to the square of the number of departments, and the Euclidean distance metric (van Camp problem) is much more CPU intensive due to the square root evaluation. Number of solutions needed before the best is found is highly variable, however, later-found best solutions are generally small improvements over some solution found much earlier.

Table 4. Computational Effort.

Problem	Least # of Soln.	Mean # of Soln.	Most # of Soln.	Mean # Soln./CPU Sec.
van Camp	440	11202	28112	3068
Bazaraa 1	13517	33940	55821	3628
Bazaraa 2	2868	25642	55704	4431
Armour & Buffa	406	69525	144842	2640

4.4 Discussion

We have shown the robustness of the GA approach over various test problems, parameter settings, random number seeds, and degrees of constrainedness. We were able to substantially improve upon previously published results for all the unequal area flow matrix test problems considered. The primary reason genetic algorithms have not been used often in constrained optimization has been the lack of a satisfactory method for directing search to near-feasible regions. Our adaptive penalty method solves this problem, opening the door to convenient GA implementation for many classical NP hard problems.

5. References

[1] Armour, G. C. and Buffa, E. S., "A Heuristic Algorithm and Simulation Approach to Relative Location of Facilities," *Management Science* 9, 294-309 (1963).

[2] Bazaraa, M. S., "Computerized Layout Design: A Branch and Bound Approach," *AIIE Transactions* 7, 432-438.

[3] Davis, L., "Job Shop Scheduling With Genetic Algorithms," *Proceedings of an International Conference on Genetic Algorithms*, 136-140 (1985).

[4] Davis, L. and Coombs, S., "Genetic Algorithms and Communication Link Speed Design: Theoretical Considerations," *Proceedings of the Second International Conference on Genetic Algorithms*, 252-256 (1987).

[5] DeJong, K. A., *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*, Doctoral Dissertation, University of Michigan (1975).

[6] Goldberg, D. E., *Computer-aided Gas Pipeline Operation Using Genetic Algorithms and Rule Learning*, Doctoral Dissertation, University of Michigan (1983).

[7] Goldberg, D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading (1989).

[8] Goldberg, D. E. and Kuo, C. H., "Genetic Algorithms in Pipeline Optimization," *Journal of Computers in Civil Engineering* 1, 128-141 (1987).

[9] Hassan, M. M. D., Hogg, G. L., and Smith, D. R., "SHAPE: A Construction Algorithm for Area Placement Evaluation," *International Journal of Production Research* 24, 1283-1295 (1986).

[10] Holland, J. H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor (1975).

[11] House, R. L. and Dagli, C. H., "An Approach to Three-Dimensional Packing Using Genetic Algorithms," in *Intelligent Engineering Systems Through Artificial Neural Networks, Volume 2* (C. H. Dagli, L. I. Burke and Y. C. Shin, editors), ASME Press, New York, 937-942 (1992) .

[12] Huntley, C. L. and Brown, D. E., "A Parallel Heuristic for Quadratic Assignment Problems," *Computers and Operations Research* 18, 275-289 (1991).

[13] Koopmans, T. C. and Beckmann, M., "Assignment Problems and the Location of Economic Activities," *Econometrica* 25, 53-76 (1957).

[14] Nugent, C. E., Vollmann, T. E. and Ruml, J., "An Experimental Comparison of Techniques for the Assignment of Facilities to Locations," *Operations Research* 16, 150-173 (1968).

[15] Radcliffe, N. J., "Equivalence Class Analysis of Genetic Algorithms, *Complex Systems* 5, 183-205 (1991).

[16] Richardson, J. T., Palmer, M. R., Liepins, G. and Hilliard, M., "Some Guidelines for Genetic Algorithms With Penalty Functions," *Proceedings of the Third International Conference on Genetic Algorithms and Their Applications*, 191-197 (1989).

[17] Sahni, S. and Gonzalez, T., "P-complete Approximation Problems," *Journal of the Association for Computing Machinery* 23, 555-565 (1976).

[18] Scriabin, M. and Vergin, R. C., "Comparison of Computer Algorithms and Visual Based Methods for Plant Layout," *Management Science* 22, 172-181 (1975).

[19] Smith, A. E. and Tate, D. M., "Using Genetic Algorithms for Equal Area Facilities Layout Design," *Robotics and Manufacturing: Recent Trends in Research, Education, and*

Applications, Volume 4 (M. Jamshidi, R. Lumia, J. Mullins and M. Shahinpoor, editors), ASME Press, 923-928 (1992).

[20] Smith, A. E. and Tate, D. M., "Genetic Optimization Using a Penalty Function," *Proceedings of the Fifth International Conference on Genetic Algorithms and Their Applications*, 499-505 (1993).

[21] Steinberg, L., "The Backboard Wiring Problem: A Placement Algorithm," *SIAM Review* 3, 37-50 (1961).

[22] Storer, R. H., Wu, S. D. and Vaccari, R., "New Search Spaces for Sequencing Problems With Application to Job Shop Scheduling," *Management Science* 38, 1495-1509 (1992).

[23] Syswerda, G., "Schedule Optimization Using Genetic Algorithms," *Handbook of Genetic Algorithms* (L. Davis, editor), Van Nostrand Reinhold, New York, 332-349 (1991).

[24] Tam, K. Y., "A Simulated Annealing Algorithm for Allocating Space to Manufacturing Cells, *International Journal of Production Research* 30, 63-87 (1991).

[25] Tate, D. M. and Smith, A. E., "A Genetic Approach to the Quadratic Assignment Problem," *Computers and Operations Research,* in print.

[26] Tong, X., *SECOT: A Sequential Construction Technique for Facility Design*, Doctoral Dissertation, University of Pittsburgh (1991).

[27] van Camp, D. J., Carter, M. W. and Vannelli, A., "A Nonlinear Optimization Approach for Solving Facility Layout Problems," *European Journal of Operational Research* 57, 174-189 (1991).

[28] Whitley, D., Starkweather, T. and Shaner, D., "The Traveling Salesman and Sequence Scheduling: Quality Solutions Using Genetic Edge Recombination," *Handbook of Genetic Algorithms* (L. Davis, editor), Van Nostrand Reinhold, New York, 350-372 (1991).

Figure 1. The GA Encoding.

Figure 2. Breeding Mechanism.

Appendix

Encoding of Single Best Solution for the Smaller Test Problems

van Camp
Permutation = (1,2,6,9,10,7,8,4,5,3)
Breakpoints = (1,3,4,6,8)
Cost = 20320.5

Bazarra 1 (12 Departments)
Permutation = (12,6,7,1,2,3,8,9,4,14,5,16,15,11,10,13)
where departments 13-16 are dummy departments
Breakpoints = (3,4,5,6,8,9,12,13)
Cost = 8861.0

Bazaraa 1 (12 Departments) - Adjusted Solution
Permutation = (12,6,7,1,2,3,8,9,4,5,11,10)
where there are no dummy departments
Breakpoints = (3,4,5,6,8,9,10)
Cost = 8768.2

Bazaraa 2 (14 Departments)
Permutation = (10,6,14,11,5,13,1,3,2,4,7,8,9,12)
Breakpoints = (3,6,7,8,9,10)
Cost = 5077.0

Layout of Single Best Solution for the Armour and Buffa Problem

A. Armour and Buffa for $\alpha = 1000$.
Cost = 1638.5.

B. Armour and Buffa for $\alpha = 25$.
Cost = 3538.6.

C. Armour and Buffa for $\alpha = 10$.
Cost = 4633.3.

D. Armour and Buffa for $\alpha = 5$.
Cost = 5524.7.

E. Armour and Buffa for $\alpha = 3$.
Cost = 5832.6.

F. Armour and Buffa for $\alpha = 1.70667$.
Cost = 6662.9.

Biographical Sketches

David M. Tate

David M. Tate was an Assistant Professor in the Department of Industrial Engineering at the University of Pittsburgh, where he worked for five years after receiving his Ph.D. in Operations Research from Cornell University in 1989. His research interests include heuristic combinatorial optimization, characterization and control of stochastic systems with due-dates, and discrete-event simulation methodology. His past employment experience includes several years working at the former National Bureau of Standards (now NIST), in the Center for Manufacturing Engineering (Factory Automation Systems Division), and in the Center for Applied Mathematics (Operations Research Division).

Dr Tate is a Senior Member of IIE and a member of the Operations Research Society of America.

Alice E. Smith

Alice E. Smith is an Assistant Professor in the Department of Industrial Engineering at the University of Pittsburgh. She had ten years of industrial experience with Southwestern Bell Corporation. Her research interests are computational intelligence in manufacturing, including neural networks, genetic algorithms and fuzzy systems. Her articles have appeared in *International Journal of Production Research*, *Journal of Intelligent Manufacturing*, *International Journal of Advanced Manufacturing Technology*, and *The Engineering Economist.* She has served as Principal Investigator on research funded by the National Science Foundation, Lockheed Corporation and the Ben Franklin Technology Center of Western Pennsylvania.

She is a senior member of IIE and SWE, and a member of IEEE.