INFORMATICA, 1993, Vol.4, No.1-2, 172-187

A COMPUTATIONAL COMPARISON OF LOCAL SEARCH HEURISTICS FOR SOLVING QUADRATIC ASSIGMENT PROBLEMS

Panos M. PARDALOS
University of Florida
Gainesville, FL 32611
Kowta A. MURTHY and Terry P. HARRISON
The Pennsylvania State University
University Park, PA 16802

Abstract. It is well known that, in general, exact algorithms for the Quadratic Assignment Problem (QAP) cannot solve problems of size $N > 15$. Therefore, it is necessary to use heuristic approaches for solving large-scale QAPs. In this paper, we consider a class of heuristic approaches based on local search criteria. In particular, we selected four algorithms; CRAFT, Simulated Annealing, TABU search and the Graph Partitioning (GP) approach and studied their relative performance in terms of the quality of solutions and CPU times. All of these algorithms performed roughly the same, based on the results of two sets of test problems executed on an IBM ES/3090-600S computer.

Key words: quadratic assignment, local search, tabu search, simulated annealing, graph partitioning, testing

1. Introduction. The Quadratic Assignment Problem (QAP) is defined as follows.

Given a set $N = \{1, 2, \ldots, n\}$ and two $(n \times n)$ matrices $F = (f_{ij})$ and $D = (d_{kl})$, find a permutation p of the set N that minimizes:

$$Q_{FD}(p) = \sum_i \sum_j f_{ij} d_{p(i)p(j)}.$$

The QAP was first introduced by Koopmans and Beckmann in 1957 [19] in the context of an analysis of the location of economic activity. In the framework of this classical facility-location problem, the set N describes the sites on which facilities are to be located.

The matrix $F = (f_{ij})$ is the flow matrix, where f_{ij} represents the flow of materials from facility i to facility j and $D = (d_{kl})$ is the distance matrix, where d_{kl} represents the distance from location k to location l. Note that, with this model, it is reasonable to assume that both matrices F and D are symmetric and have non-negative entries. In addition to its application in facility location and facility layout problems [19, 9,10], QAP has been found useful for such applications as problems in scheduling [12], the backboard wiring problem [36] in electronics, and even assignment of runners to a relay team [16]. Other applications may be found in [11, 20, 23].

The QAP has proved to be a very difficult problem computationally. This problem, of which the traveling salesman problem is a special case [21], is NP-hard. Furthermore, the problem of finding an ϵ-approximation solution is also NP-hard [31]. Problems of size $n > 15$ are not practically solvable in terms of obtaining exact solutions [27, 30].

Several heuristic methods have been developed to solve the quadratic assignment problem [5, 15, 33, 35, 37]. Most of these methods use a local search algorithm, where the local search depends on the formulation of the problem and the structure of its neighborhood. Such an algorithm starts with an initial feasible solution and successively moves to neighboring solutions until no further improvement is possible. In practice, these methods tend to converge quickly, but the complexity of finding a locally optimal solution has so far not been established. However, it has been proved that local search for the QAP is PLS-Complete [24].

In the case of QAP, most of these algorithms implement a restricted heuristic search based on a pair exchange or at most a triple exchange. In this paper, we present a comparative evaluation of four local search heuristic methods: CRAFT [1, 7, 32], SIMULATED ANNEALING [37], TABU search [35] and the GRAPH PARTITIONING approach [24], to solve a collection of well-known QAPs that have appeared in literature. In addition, we constructed a group of test problems (with known optimal solutions) using

the algorithm described in [25] and evaluated the above heuristics. In the next section, we briefly describe each algorithm. We then present implementation details of the algorithms and the selection of test problems. Finally, we present computational results on the performance of these algorithms in terms of accuracy of solutions and running times, together with our conclusions.

Many computational approaches based on local search, have been proposed for solving large-scale QAPs. However, to our knowledge, there has been no unified treatment and comparison of performance. Our study may prove useful not only to the theoretician, showing the relative performance of various local search heuristics, but also to the practitioner for guidance in solving large-scale QAPs.

2. Computational approaches. We briefly describe the selected approaches for solving QAPs, namely, CRAFT, Simulated Annealing, TABU search and the Graph Partitioning (GP) approach for solving QAP.

2.1. CRAFT. CRAFT (Computerized Relative Allocation of Facilities Technique) [1, 7, 32] is a well-known heuristic for designing the layout of facilities that has been in use for over 25 years. Given a set of departments, locations, a matrix of flows between departments, and a matrix of costs to transport one item between two departments a unit distance, CRAFT iteratively improves an initial, user-supplied, layout by a series of department exchanges. At each step CRAFT considers either all possible 2-way, 3-way, or both 2-way and 3-way exchanges. It chooses the exchange that provides the most improvement in minimizing total cost, and then repeats the process until no improving exchange can be found.

2.2. Simulated annealing. Simulated annealing is a stochastic optimization technique which has been found effective in solving global optimization problems. Kilpatrick, Gelatt and Vecchi [18] tried this heuristic approach in 1983, based on the analogy between statistical mechanics and combinatorial optimization. The term "annealing" refers to the process of a thermal system by first

melting at high temperatures and then lowering the temperatures slowly based on an annealing schedule. The process is continued until the vicinity of the solidification temperature is reached, where the system is allowed to reach the "ground state" (the lowest energy state of the system). Simulated annealing is a Monte Carlo approach to simulate the behavior of this system to achieve thermal equilibrium at a given temperature in a given annealing schedule.

This analogy has been applied in solving combinatorial optimization problems. According to the above authors:

> Iterative improvement, commonly applied to such problems, is much like the microscopic rearrangement process modeled by statistical mechanics, with the cost function playing the role of energy. However, accepting only rearrangements that lower the cost function of the system is like extremely rapid quenching high temperatures to $T = 0$. So, it should not be surprising that resulting solutions are usually metastable. The Metropolis procedure from statistical mechanics provides a generalization of iterative improvement in which controlled uphill steps can also be incorporated in the search for a better solution.

In particular, the application of simulated annealing to the Traveling Salesman Problem was investigated extensively [18, 34]. Simulated Annealing is applied to the QAP [37] as follows.

- Given any feasible solution (a permutation of locations in relation to the facilities), randomly select two facilities, make a pair exchange and evaluate the consequent change (δE) in the total cost (E)
- Repeat the above step as long as $\delta E < 0$. Otherwise, select a random variable x from a uniform distribution $U(0,1)$. If $x < P(\delta E) = EXP(-\delta E/t_i)$ (where P represents the probability obtained from the exponential distribution (EXP)), then accept the pair exchange and repeat the process. Here t_i represents the annealing schedule temperature at stage i where $t_1 > t_2 > \ldots > t_r$ represents the annealing schedule. For example $t_i = 10 \times (0.9)^{(i-1)}$.

- The system remains at stage i until a predetermined number of pair exchanges have been considered before going to the next stage.
- If all the temperatures in the annealing schedule have been used, i.e. if $i > r$, then stop.

This approach was implemented by Burkard and Rendl in 1984 [8] and Wilhelm and Ward in 1987 [37].

2.3. TABU search. TABU search was first introduced by Glover [13, 14] as a meta-heuristic for solving optimization problems. The key idea is to avoid reaching a locally optimal point prematurely by taking certain paths even though they may not be currently advantageous (i.e. temporarily increases the current value of the objective function) hoping to eventually reach a better local optima. However, this may lead to cycling. To avoid the process of cycling, certain search directions are forbidden at each iteration.

This technique has been successfully applied to a variety of optimization problems including the Traveling Salesman Problem. Based on this approach, the TABU-Navigation algorithm for the QAP was developed by Skorin-Kapov in [35].

Given an initial permutation, the algorithm consists of maintaining a TABU list of a predetermined maximum size by constantly constructing and updating it. If a pair belongs to the TABU list for a given iteration, that pair is not allowed to be exchanged at that iteration unless the exchange yields an objective function value which is strictly better than the one obtained thus far. When a pair is selected for exchange, it is added at the bottom of the list which is maintained in FIFO order. The length of the list is a parameter to the algorithm. If it is too small, cycling may occur because there is insufficient information in the list to detect a cycle. If the length of the list is too large, it restricts the efficiency of the search because every entry in the list is ignored from consideration as a potential exchange leading to a better solution. It has been observed that the best value for the size of the list may be treated as an increasing function of the size of the problem.

2.4. Graph partitioning approach. The Graph Partitioning (GP) problem is defined as follows.

Given an undirected graph $G(V, E)$ with edge weights $w(e)$, the problem is to divide V (where $|V| = 2n$), into two equal subsets A and B, such that, the cost $C(A, B)$, which is the sum of the weights of all edges going from A to B, is the least.

This problem is known to be NP-Complete. A local search algorithm for this problem due to Kernighan-Lin (K-L) [17] is as follows.

A *swap* (exchange of a pair of vertices between partitions) of partition (A, B) is a partition (A', B'), where A and A' have a symmetric difference of 2, i.e., (A', B') is obtained from (A, B) by swapping one element of A with one element of B. (A', B') is a greedy swap if $C(A, B) - C(A', B')$ is maximized over all swaps of (A, B). Let (A_i, B_i) be a sequence of partitions, each of which is a greedy swap of the one preceding it, starting from (A_0, B_0). We call it monotonic if the differences of $A_i - A_0$ and $B_i - B_0$ are monotonically increasing (that is, no vertex is switched back to its original set (A_0, B_0)). Finally, we say that a partition (A', B') is a neighbor of (A, B) if it occurs in the unique maximal monotonic sequence of greedy swaps starting with (A, B). Note that such a sequence will consist of $n + 1$ partitions, with the last one equaling (B, A). Thus, each partition has n neighbors. The algorithm performs local search over this neighborhood structure.

Based on the K-L algorithm, a local search algorithm for the QAP as presented by Murthy and Pardalos [24] is as follows.

Graph partitioning local search algorithm for the QAP

Input. Two matrices F and D and a set $N = \{1, 2, \ldots, n\}$.
Output. A locally optimal permutation p for the QAP.

1. Given a permutation p_0 calculate its cost $C(p_0)$. $i = 0; g_i = 0$ and $G(i) = 0$, where g_i and $G(i)$ are step gain and cumulative gain respectively.
2. $i = i + 1$. For each pair of facilities not already selected, evaluate the step gain by exchanging their locations. Then,

select the pair with maximum gain. If no such pair exists, set $i = i - 1$ and go to 4. Let the maximum gain be $g_i = C(p_{i-1}) - C(p_i)$.

3. Compute the cumulative gain, $G(i) = \sum_{k=0}^{k=i} g_k$. If $G(i) > 0$; then go to 2.
4. Select k, such that $G(k)$ is maximum for $0 \leq k \leq i$.
5. If $k > 0$ then set $p_0 = p_k$ and $i = 0, g_i = 0, G(i) = 0$; then go to 2.
6. We have reached a local optimum for the QAP. Set $p = p_0$ and output p and $C(p)$.

Using the above approach, it has been proved that the local search problem for the QAP is PLS-Complete [24].

3. Computational results.

3.1. Programs and parameters. The source code for CRAFT was obtained via SHARE [32]. CRAFT requires very little guidance via control parameters. Essentially, the user need only choose whether to perform 2-way exchanges, 3-way exchanges, or the best of both 2-way and 3-way exchanges.

The source code for Simulated Annealing was developed as outlined in [37]. The parameters that control program execution are numerous. They are:

- $S = \{t_1, t_2, \ldots, t_r\}$, a set of annealing schedule temperatures, where $t_i = 10 \times (0.9)^{i-1} \quad \forall\, i = 1, 2, \ldots, r$. Therefore, $t_1 > t_2 > \ldots > t_r$.
- e = Epoch interval—an a priori number of pair-wise interchanges of facility locations at temperature t_i.
- ϵ = an error constant used to determine whether the system is in equilibrium at a specific temperature t_i.
- J = a constant representing the total number of interchanges attempted at the current temperature t_i.
- N' = a constant which when multiplied by the problem size, n, defines the maximum value J.

The source code for the Graph Partitioning type was taken from [24]. This program has no externally controlled parameters.

The source code for TABU search was obtained from Skorin-Kapov [35].

The parameters that control the running of this program are: TABU-SIZE (TABU table size) and MAXITER (maximum number of iterations).

Each of the above algorithms start with a given permutation and eventually reaches a local optimum by a series of exchanges.

The parameters for each program were chosen after conducting test runs with small as well as large problem sizes. The main considerations were to obtain comparable accuracy and CPU times. Based on the above considerations, the parameters selected were as follows.

1. CRAFT: Select the best of both 2-way and 3-way exchanges.
2. Simulated Annealing: $e = 50, \epsilon = .01$ and $N' = 100$.
3. Graph Partitioning approach: none.
4. TABU search: TABU-SIZE $= N/3$ and MAXITER $= 2N$.

3.2. Test problems. Data for F and D matrices are obtained in two ways.

The first set of problems are the well known NUGENT collection of quadratic assignment problems reported in [26] with problem sizes, $n = 6, 8, 12, 15, 20$ and 30.

The second set of problems are generated according to a test problem generator (with known optimal solutions) as reported in [25] with problem sizes $n = 10, 20, 30, 40, 50, 60, 70, 80$ and 90. For ready reference, the algorithm to generate test problems is described below. For details, see [25].

Input. w, a value to initialize the F matrix, and $z < w$, to obtain random values between $[0, z]$.
Output. Matrices F and D and an optimal permutation p^*.

1. Construct the matrix $D = (d_{ij})$, of which the elements are the distances between the knots of the two dimensional grid $r \times s$, where $rs = n$, using rectilinear distances. If (i, j) are neighboring knots, then $d_{ij} = 1$.

2. Set $F = (f_{ij})$ where $f_{ij} = w$ (an input parameter to the algorithm). Compute $g_{ij} = 2 - d_{ij}$.
3. **While** for any $i, j = 1, \ldots, n$ such that $g_{ij} \leq 0$.
4. Choose the pair l, m, such that $d_{lm} = \max\{d_{ij}\}$ where the max is taken over every (i, j) for which $g_{ij} \leq 0$. If no such pair exists, then go to 8.
5. Randomly select a grid point k on one of the shortest ways from l to m, such that, $| d_{lk} - d_{mk} | \leq 1$. Then, choose randomly, $\Delta \in [0, z < w]$ where w and z are the input parameters to the algorithm.
6. Set $f_{lm} := \Delta, f_{lk} := f_{lk} + (w - \Delta), f_{mk} := f_{mk} + (w - \Delta)$ and $g_{lm} := g_{lk} := g_{mk} := 1$.
7. **Endwhile.**
8. Finally, generate the random permutation $p^* = p^*(i), i = 1, 2, \ldots, n$, where P^* will be the optimal permutation. Form the matrix $F = (c_{ij})$ in which $f_{ij} = f_{uv}$ where $i = p^*(u)$ and $j = p^*(v)$.
9. Output F, D and p^* and optimal cost $w(\sum \sum d_{ij})$.

Using the test problem generator, 80 test problems were created (eight each for each $n = 10, 20, 30, 40, 50, 60, 70, 80$ and 90 with $w = 9$ and $z = 1, 2, 3, 4, 5, 6, 7$ and 8, respectively). The selected grid (r, s), (where $n = rs$), for these test cases is: for $n = 10, 20, 30$ and 40, $s = 5$ and $r = 2, 4, 6$ and 8, respectively; for $n = 50, 60, 70, 80$ and 90, $s = 10$ and $r = 5, 6, 7, 8, 9$ and 10, respectively).

3.3. Analysis of results. We implemented and tested the above algorithms on an IBM ES/3090-600S running CMS Version 5.6, using the VS/FORTRAN compiler Version 2.4.0 with OPT=3. With each of the test problems a set of twenty-five randomly generated initial permutations were used for each of the four algorithms. For consistency and comparability, the same set of initial permutations associated with each test problem were used with each algorithm. The results shown below include the averages over the 25 sample starting permutations for each test problem, where:

- MIN is the minimum value attained over the 25 runs for each problem,

- AVE is the average value obtained over the 25 runs and
- MAX is the maximum value attained over the same 25 runs.

Also included are the known optimal solution (OPT), when OPT is known (otherwise best known value (BKV)), and the ratios corresponding to MIN, AVE and MAX namely, MIN/(OPT or BKV), AVE/(OPT or BKV) and MAX/(OPT or BKV). All CPU times given below are in seconds.

The computational results suggest that with regard to the quality of solutions, the performance of all four heuristics is roughly the same. Using only 25 starting permutations, all algorithms produced solutions very close to the optimal independent of the problem size (mostly between 0 and 2 percent away from the optimal, except that CRAFT's solutions for NUGENT cases where $N = 12$ and $N = 15$ were 7 and 6 percent away from the optimal, respectively). It is surprising that even the maximum values obtained by these algorithms are at most only 15 percent away from the optimal. Regarding running times, Simulated Annealing took relatively less CPU time especially as the problem size increased. In comparison, CRAFT, GP and TABU required progressively more time in that order.

Conclusions. In this paper, we studied the computational performance of different local search heuristics for solving QAPs. We considered CRAFT, Simulated Annealing, TABU search and Graph Partitioning approach using two sets of test problems: i) the classic NUGENT set and ii) a set of large-scale test problems generated with known optimal solutions using the algorithm described in [25].

The computational results suggest that all of these approaches have almost the same performance. The quality of the solution suggests that local search algorithms are very promising, useful and reliable. Additionally, the running times of these algorithms in relation to the size of the problem is very small and consequently insignificant. Therefore, local search algorithms may be appealing, practical and inexpensive for solving large-scale practical QAPs. However, it may be of interest to study the relative performance

Table 1. NUGENT test problems

N and Heuristics	Best Known Value	MIN	MIN BKV	AVE	AVE BKV	MAX	MAX BKV	Ave. CPU Time
N = 6								
CRAFT	86	36	1.00	88	1.02	92	1.07	0.01
ANNEAL	86	86	1.00	90	1.05	94	1.09	0.05
GP	86	86	1.00	91	1.06	94	1.09	0.00
TABU	86	86	1.00	87	1.01	92	1.07	0.00
N = 8								
CRAFT	214	214	1.00	222	1.04	238	1.11	0.01
ANNEAL	214	214	1.00	220	1.03	238	1.11	0.07
GP	214	214	1.00	221	1.03	240	1.12	0.00
TABU	214	214	1.00	214	1.00	218	1.02	0.01
N = 12								
CRAFT	578	621	1.07	644	1.11	669	1.16	0.03
ANNEAL	578	578	1.00	610	1.06	634	1.10	0.14
GP	578	586	1.01	606	1.05	632	1.09	0.01
TABU	578	578	1.00	592	1.02	602	1.04	0.03
N = 15								
CRAFT	1,150	1,222	1.06	1,264	1.10	1,322	1.15	0.05
ANNEAL	1,150	1,156	1.01	1,211	1.05	1,288	1.12	0.21
GP	1,150	1,150	1.00	1,200	1.04	1,286	1.12	0.02
TABU	1,150	1,150	1.00	1,168	1.02	1,200	1.04	0.07
N = 20								
CRAFT	2,570	2,602	1.01	2,688	1.05	2,746	1.07	0.10
ANNEAL	2,570	2,624	1.02	2,680	1.04	2,780	1.08	0.36
GP	2,570	2,602	1.01	2,670	1.04	2,736	1.06	0.08
TABU	2,570	2,570	1.00	2,622	1.02	2,670	1.04	0.21
N = 30								
CRAFT	6,124	6,143	1.00	6,309	1.03	6,498	1.06	0.39
ANNEAL	6,124	6,226	1.02	6,351	1.04	6,536	1.07	0.76
GP	6,124	6,180	1.01	6,304	1.03	6,418	1.05	0.49
TABU	6,124	6,180	1.01	6,278	1.03	6,470	1.06	1.05

Table 2. Random test problems with known optimal solutions Part 1

N and Heuristics	Best Known Value	MIN	MIN BKV	AVE	AVE BKV	MAX	MAX BKV	Ave. CPU Time
N = 10								
CRAFT	1,890	1,890	1.00	2,003	1.06	2,175	1.15	0.02
ANNEAL	1,890	1,890	1.00	2,008	1.06	2,165	1.15	0.11
GP	1,890	1,890	1.00	1,989	1.05	2,157	1.14	0.01
TABU	1,890	1,890	1.00	1,922	1.02	2,067	1.09	0.02
N = 20								
CRAFT	10,260	10,430	1.02	11,048	1.08	11,645	1.13	0.11
ANNEAL	10,260	10,284	1.00	11,021	1.07	11,628	1.13	0.36
GP	10,260	10,260	1.00	10,904	1.06	11,415	1.11	0.10
TABU	10,260	10,260	1.00	10,737	1.05	11,239	1.10	0.21
N = 30								
CRAFT	28,710	28,838	1.00	30,420	1.06	31,959	1.11	0.47
ANNEAL	28,710	28,803	1.00	30,340	1.06	31,478	1.10	0.77
GP	28,710	28,710	1.00	30,303	1.06	31,427	1.09	0.52
TABU	28,710	28,710	1.00	30,160	1.05	31,308	1.09	1.05
N = 40								
CRAFT	60,840	61,012	1.00	64,016	1.05	67,619	1.11	1.38
ANNEAL	60,840	60,840	1.00	64,030	1.05	66,360	1.09	1.32
GP	60,840	60,840	1.00	64,041	1.05	66,259	1.09	1.70
TABU	60,840	60,840	1.00	63,570	1.04	66,160	1.09	3.31
N = 50								
CRAFT	110,250	110,785	1.00	115,661	1.05	120,799	1.10	3.24
ANNEAL	110,250	110,343	1.00	115,797	1.05	120,538	1.09	2.04
GP	110,250	110,564	1.00	115,358	1.05	120,027	1.09	4.30
TABU	110,250	110,250	1.00	114,578	1.04	120,022	1.09	7.97
N = 60								
CRAFT	169,920	170,990	1.01	178,330	1.05	185,556	1.09	6.60
ANNEAL	169,920	171,033	1.01	178,636	1.05	185,687	1.09	2.90
GP	169,920	171,116	1.01	177,742	1.05	184,129	1.08	9.28
TABU	169,920	170,355	1.00	177,631	1.05	184,032	1.08	17.09

Table 2. Random test problems with known optimal solutions Part 2

N and Heuristics	Best Known Value	MIN	MIN BKV	AVE	AVE BKV	MAX	MAX BKV	Ave. CPU Time
N = 70								
CRAFT	246,330	248,766	1.01	258,983	1.05	267,577	1.09	12.32
ANNEAL	246,330	250,354	1.02	259,212	1.05	267,125	1.08	4.04
GP	246,330	249,435	1.01	258,437	1.05	266,781	1.08	18.26
TABU	246,330	247,677	1.01	257,643	1.05	266,819	1.08	31.94
N = 80								
CRAFT	341,280	344,565	1.01	357,521	1.05	370,172	1.08	21.06
ANNEAL	341,280	345,710	1.01	358,918	1.05	370,214	1.08	5.31
GP	341,280	343,239	1.01	358,232	1.05	369,627	1.08	35.38
TABU	341,280	343,995	1.01	356,774	1.05	369,694	1.08	54.77
N = 90								
CRAFT	456,570	461,851	1.01	477,344	1.05	494,306	1.08	33.10
ANNEAL	456,570	463,950	1.02	479,303	1.05	494,125	1.08	6.77
GP	456,570	460,114	1.01	479,826	1.05	493,792	1.08	56.89
TABU	456,570	458,601	1.00	476,842	1.04	494,119	1.08	88.64

of these local search algorithms against other heuristics, such as, cutting plane and integer programming methods [3, 4, 5, 6, 2, 29] for solving large-scale QAPs.

Acknowledgments. We would like to thank Professor J. Skorin-Kapov for providing us the TABU search code.

REFERENCES

[1] Armour, G.C., and E.S. Buffa (1963). Heuristic algorithm and simulation approach to relative location of facilities. *Management Science*, **9**, 294–309.

[2] Balas, E. and J.B. Mazzola (1980). Quadratic 0-1 programming by a new linearization. In *Proceedings of the TIMS/ORSA*, Washington D.C., May

1980. TIMS/ORSA.

[3] Bazaraa, M.S., and A.N. Elshafei (1979). Exact and heuristic procedures for the quadratic assignment problem. *Naval Research Logistics Quarterly*, **26**, 109–120.

[4] Bazaraa, M.S., and H.D. Sherali (1979). New approaches for solving the quadratic assignment problem. *Operations Research Verfahren*, **32**, 29–46.

[5] Bazaraa, M.S., and H.D. Sherali (1980). Bender's partitioning scheme applied to a new formulation of the quadratic assignment problem. *Naval Research Logistics Quarterly*, **27**, 29–41.

[6] Bazaraa, M.S., and H.D. Sherali (1982). On the use of exact and heuristic cutting plane methods for the quadratic assignment problem. *Journal of Operations Research Society*, **33**, 991–1003.

[7] Buffa, E.S., G.C. Armour and T.E. Vollman (1962). Allocating facilities with craft. *Harvard Business Review*, **42**, 136–158.

[8] Burkard, R.E., and F. Rendl (1984). A thermodynamically motivated simulation procedure for combinatorial optimization problems. *European Journal of Operations Research*, **17**, 169–174.

[9] Dicky, J.W., and J.W. Hopkins (1972). Campus building arrangement using TOPAZ. *Transportation Research*, **6**, 59–68.

[10] Elshafei, A.W.(1977) Hospital layout as a quadratic assignment problem. *Operations Research Quarterly*, **28**, 167–179.

[11] Francis, R.L., and J.A. White (1974). *Facility Layout and Location*. Prentice-Hall, Englewood Cliffs, N.J.

[12] Geoffrion, A.M., and G.W. Graves (1976). Scheduling parallel production lines with changeover costs: Practical applications of a quadratic assignment/LP approach . *Operations Research*, **24**, 595–610.

[13] Glover, F. (1989). Tabu search - Part 1. *ORSA Journal on Computing*, 1(3), 190–206.

[14] Glover, F. (1989). Tabu search - Part 2. *ORSA Journal on Computing*, 2(1), 4–32.

[15] Hadley, S. (1989). *Continuous optimization approaches for the quadratic assignment problem*. Ph.D. thesis, University of Waterloo.

[16] Heffley, D.R. (1977). Assigning runners to a relay team. In S.P. Ladany and R.E. Machol, editors, *Optimal Strategies in Sports*, North-Holland, Amsterdam.

[17] Kernighan, B., and S. Lin (1972). An efficient heuristic procedure for partitioning graphs. *Bell Systems Journal*, **49**, 291–307.

[18] Kirpatrick, S.D., C.D. Gelatti and M.P. Vecchi (1983). Optimization by simulated annealing. *Science*, **220**, 671–680.

[19] Koopmans, T.C., and M.J. Beckmann (1957). Assignment problems and the location of economic activities. *Econometrica*, **25**, 53–76.

[20] Krarup, J., and P.M. Pruzan (1978). Computer-aided layout design. *Mathematical Programming Study*, **9**, 75–94.

[21] Lawler, E.L. (1963). The quadratic assignment problem. *Management Science*, **9**, 586–599.

[22] Majone, G., and E.S. Quade (1980). *Pitfalls of Analysis*. Wiley & Sons, New York.

[23] McCormik, E.J. (1970). *Human Factors Engineering*. McGraw-Hill, New York.

[24] Murthy, K.A., P.M. Pardalos and Y. Li (1992). A local search algorithm for the quadratic assignment problem. *Informatica* Vol. 3, No. 4.

[25] Murthy, K.A., and P.M. Pardalos (1990). A polynomial-time approximation algorithm for the quadratic assignment problem. Technical Report CS-33-90, The Pennsylvania State University.

[26] Nugent, C.E., T.E. Vollmann and J. Ruml (1969). An experimental comparison of techniques for the assignment of facilities to locations. *Journal of Operations Research*, **16**, 150–173.

[27] Pardalos, P.M., and J. Crouse (1989). A parallel algorithm for the quadratic assignment problem. In *Proceedings of the Supercomputing 1989 Conference*, ACM Press. pp. 351–360.

[28] Pardalos, P.M., K.A. Murthy and Y. Li (1992). Computational experience with parallel algorithms for solving the quadratic assignment problem. In *Computer Science and Operations Research: New Developments in their Interface*, O. Balci et al (Editors), Pergamon Press. pp. 267–278.

[29] Pardalos, P.M., and G.P. Rodgers (1990). Computational aspects of a branch and bound algorithm for quadratic zero-one programming. *Computing*, **45**, 131–144.

[30] Roucairol, C. (1987). A parallel branch and bound algorithm for the quadratic assignment problem. *Discrete Applied Mathematics*, **18**, 211–225.

[31] Sahni, S., and T. Gonzalez (1976). P-complete approximation problems. *Journal of the Association of Computing Machinery*, **23**, 555–565.

[32] SHARE (1965). Computerized relative allocation of facilities technique. Share Program Library. SDA3391.

[33] Sherali, H.D. P., and Rajgopal (1986). A flexible polynomial time construction and improvement heuristic for the quadratic assignment problem. *Computers and Operations Research*, **13**(5), 587–600.

[34] Skiscim, C.C., and B.L. Golden (1983). Optimization by simulated annealing: A preliminary computational study for the TSP. In *Proceedings of the 1983 Winter Simulation Conference*.

[35] Skorin-Kapov J. (1990). Tabu search applied to the quadratic assignment problem. *ORSA Journal on Computing*, **2**(1), 33–45.

[36] Steinberg, L. (1961). The backboard wiring problem: A placement algorithm. *SIAM Review*, **3**, 37–50.

[37] Wilhelm, M.R., and T.L. Ward (1987). Solving quadratic assignment problems by simulated annealing. *IEEE Transactions*, **19**(1), 107–119.

Received January 1993

P. M. Pardalos was born in Greece in 1954. He received his B. S. degree in Mathematics from Athens University and his Ph. D. degree in Computer Science from the University of Minnesota. Currently he has a visiting associate professor at the University of Florida and a professor at the Technical University of Crete, Greece. His research interests include global and combinatorial optimization, parallel algorithms and software development. He is the author and editor of several books in global optimization and serves as a managing editor of the Journal of Global Optimization.

K. A. Murthy received his Ph. D. degree in Computer Science from the Pennsylvania State University in 1991. His research interests include software development and combinatorial algorithms. He works as an independent consultant.

T. P. Harrison is Associate Professor of Management Science and Information Systems at the Smeal College of Buisiness Administretion, Penn State University. He received his Ph.D. in Management Science from the University of Tennessee. His research interests are in the theory and practice of large scale mathematical programming, especially in modelling productuion/distribution systems, manufacturing, and the management of natural resources.