Simple Heuristics for the Generalized Quadratic Assignment Problem

Alan McKendall
Department of Industrial & Management Systems Engineering, West Virginia University
325A Mineral Resources Building, Morgantown, WV 26506
Email: Alan.McKendall@mail.wvu.edu

ABSTRACT

The generalized quadratic assignment problem (GQAP) can be stated as the problem of assigning *M* machines to *N* locations, where *M* > *N* and one or more machines can be assigned to each location without exceeding the capacities of the locations. Although the QAP is a well-researched problem, there are very few papers in the literature which focuses on the GQAP. In this paper, a construction algorithm and a local search heuristic are developed for the GQAP. Also, a mathematical model is presented for the problem, and a problem instance will be used to illustrate the solution techniques.

Keywords: Facility layout problem, Generalized quadratic assignment problem, Heuristics

INTRODUCTION

The problem of assigning *M* machines to *N* locations on the plant floor of a manufacturing facility such that the sum of material handling and installation costs is minimized is known as the machine layout problem. For this problem, the plant floor is represented as an array of *N* equal size grid units, each having enough capacity to store any of the *M* machines. Therefore, this problem can be modeled as a quadratic assignment problem (QAP). The QAP was introduced by **Koopmans and Beckmann (1957)**, and was proven to be NP Hard by **Sahni and Gonzales (1972)**. See **Burkard et al. (1998)** and **Loiola et al. (2007)** for an extensive review of the solution techniques for the QAP.

A generalization of the machine layout problem defined above is to assign one or more machines to each location on the plant floor such that the plant floor may be represented as an array of unequal-area grids. More specifically, *M* machines, which may have different space (area) requirements, are assigned to *N* locations of varying sizes (*M* > *N*) such that the capacities of the locations are not exceeded. This extended machine layout problem can be modeled as a generalized quadratic assignment problem (GQAP), which was introduced by **Lee and Ma (2004)**. Other applications of the GQAP are to assign sets of equipment to manufacturing sites as described in **Lee and Ma (2004)** and to assign sets of containers to storage locations in container yards as described in **Cordeau et al. (2006)**. In this paper, the GQAP is defined as the problem of assigning *M* machines to *N* locations (*M* > N) such that the capacity of the locations are not exceeded and the sum of material handling and installation costs is minimized.

The formulation of the GQAP is given below and is an adaptation of the model presented by **Lee and Ma (2004)**.

$$\text{Minimize } z = \sum_{i=1}^{M}\sum_{k=1}^{N} a_{ik} x_{ik} + \sum_{i=1}^{M}\sum_{j=1}^{M}\sum_{k=1}^{N}\sum_{l=1}^{N} c_{ijkl}\, f_{ij}\, d_{kl}\, x_{ik}\, x_{jl} \qquad (1)$$

$$\text{s.t.} \quad \sum_{k=1}^{N} x_{ik} = 1, \qquad i = 1, \ldots, M \qquad (2)$$

$$\sum_{i=1}^{M} r_i x_{ik} \le C_k, \quad k = 1, \ldots, N \qquad (3)$$

$$x_{ik} = \{0, 1\}, \qquad i = 1, \ldots, M,\ k = 1, \ldots, N \qquad (4)$$

where M is the number of machines, N is the number of locations, a_{ik} is the cost of assigning (installing) machine i to (at) location k, f_{ij} is the flow of materials from machine i to machine j, d_{kl} is the distance from location k to location l, c_{ijkl} is the unit cost per distance unit of moving materials from machine i (at location k) to machine j (at location l), r_i is the space requirement of machine i, and C_k is the amount of space available (capacity) at location k. The decision variables are defined as

$$x_{ik} = \begin{cases} 1, if\ machine\ i\ is\ assigned\ to\ location\ k, \\ 0, otherwise. \end{cases}$$

Objective function (1) minimizes the sum of the installation and material handling costs. Constraints (2) ensure that each machine is assigned to only one location. Constraints (3) ensure that the space capacity of each location is not exceeded, and the restrictions on the decision variables are given in (4).

The term in objective function (1) used to obtain material handling cost has a quadratic term (i.e., product of two variables). As a result, the mathematical formulation (1) – (4) is nonlinear and is called a binary integer nonlinear programming model. The model is linearized by substituting w_{ijkl} for $x_{ik}x_{jl}$. Then, replace objective function (1) with

$$\text{Minimize } z = \sum_{i=1}^{M}\sum_{k=1}^{N} a_{ik}x_{ik} + \sum_{i=1}^{M}\sum_{j=1}^{M}\sum_{k=1}^{N}\sum_{l=1}^{N} c_{ijkl} f_{ij} d_{kl} w_{ijkl} \qquad (1')$$

and add the following constraints

$$x_{ik} + x_{jl} - 1 \leq w_{ijkl} \qquad \text{for } i, j = 1, \ldots, M,\ k, l = 1, \ldots, N \text{ where } j \neq i \text{ and } l \neq k \qquad (5)$$

$$w_{ijkl} = 0 \text{ or } 1 \qquad \text{for } i, j = 1, \ldots, M,\ k, l = 1, \ldots, N \text{ where } j \neq i \text{ and } l \neq k \qquad (6)$$

As a result, the linearized model (i.e., a binary integer linear programming model) for the GQAP consists of objective function (1') subject to constraints (2) – (6). This model will be used in the next section to solve a small GQAP instance.

The GQAP literature is very limited. **Lee and Ma (2004)** presented the first formulation for the GQAP. Also, the authors presented three methods for the linearization of the formulation, and a branch and bound algorithm to optimally solve the GQAP. **Hahn et al. (2008)** presented a new algorithm based on a reformulation linearization technique (RLT) dual ascent procedure to optimally solve the GQAP. Similarly, **Pessoa et al. (2008)** presented two exact algorithms for the GQAP which combine a previously proposed branch and bound scheme with a new Lagrangean relaxation procedure over a known RLT formulation. It is important to note that the exact algorithms presented above are unable to solve large-size problems in reasonable time. However, the following heuristics (or approximation algorithms) are able to obtain "good" solutions for large-size problems in reasonable time. **Cordeau et al. (2006)** presented a linearization of the GQAP formulation as well as a memetic heuristic for the GQAP, which combines genetic algorithms (**Holland, 1975**) and tabu search (**Glover, 1986**). **Mateus et al. (2011)** proposed several GRASP (greedy randomized adaptive search procedure) with path-relinking heuristics for the GQAP using different construction, local search, and path-relinking procedures.

In this paper, a construction algorithm and a local search heuristic are developed for solving large-size GQAP instances. The paper is organized as follows. Next, an illustrative example is presented and solved using the mathematical formulation presented above. Afterwards, construction algorithms and a local search technique are presented for the GQAP. Then concluding remarks are given.

ILLUSTRATIVE EXAMPLE

Consider a GQAP where 6 machines are assigned to 4 locations on the plant floor. First, the space requirements of each machine is determined by obtaining the footprints, personnel space needed, and material storage requirements for each machine. The space calculations are summarized in Table 1 below. Notice machine 2 requires 120 ft^2 of space.

		Area (square feet)			
Machine	Footprint	Equipment	Personnel	Material	Total (r_i)
1	5 ft x 10 ft	50	20	20	90
2	6 ft x 10 ft	60	20	40	120
3	5 ft x 6 ft	30	20	50	100
4	5 ft x 8 ft	40	20	50	110
5	5 ft x 10 ft	50	20	40	110
6	5 ft x 6 ft	30	20	20	70
			Total area required		600

Table 1. Calculations of space requirement (r_i) for each machine i.

Next, the plant floor configuration is given in Figure 1. The dimensions, area capacities, and centers of the locations (centroids) for the four locations (sites) are given in Table 2. For instance, location 2 is 20 feet long by 10 feet wide, and the centroid is at (25, 15). Therefore, location 2 has 200 square feet of capacity. Once the centroids are obtained, the distances between locations are calculated using the rectilinear distance measure. If (a_k, b_k) and (a_l, b_l) are the centroids for locations k and l, respectively, then the distance between the locations is $|a_k - a_l| + |b_k - b_l|$. See matrix d_{kl} for distances between locations.

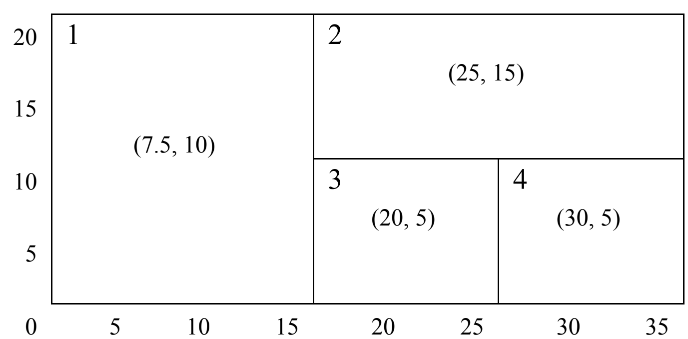

Figure 1. Plant floor configuration for Production area.

Location	Dimension	Area (C_k)	Centroid
1	15 ft x 20 ft	300 sq ft	(7.5, 10)
2	20 ft x 10 ft	200 sq ft	(25, 15)
3	10 ft x 10 ft	100 sq ft	(20, 5)
4	10 ft x 10 ft	100 sq ft	(30, 5)
Production Area	35 ft x 20 ft	700 sq ft	

$$d_{kl} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \left[\begin{array}{cccc} 0 & 22.5 & 17.5 & 27.5 \\ 22.5 & 0 & 15 & 15 \\ 17.5 & 15 & 0 & 10 \\ 27.5 & 15 & 10 & 0 \end{array}\right] \end{array}$$

Table 2. Calculations of capacity (C_k) of each location k, and distances (d_{kl}) between pairs of sites k and l.

The amount of materials (f_{ij}) flowing between machines i and j per month are obtained from route sheets and are given in Table 3. Also, the total cost of installing each machine i to each location k is calculated, and then the monthly equivalent cost (a_{ik}) is obtained and given in Table 3. Assume $c_{ijkl} = 1$ for all i, j, k, and l. Recall, c_{ijkl} is the unit cost per distance unit of moving materials from machine i (at location k) to machine j (at location l).

f_{ij} =

	1	2	3	4	5	6
1	0	33	41	9	16	56
2	0	0	49	91	78	23
3	0	0	0	41	38	44
4	0	0	0	0	6	17
5	0	0	0	0	0	68
6	0	0	0	0	0	0

a_{ik} =

	1	2	3	4
1	700	1,600	1,900	1,400
2	1,300	1,800	1,700	800
3	800	1,400	3,000	1,100
4	3,000	800	700	1,500
5	1,200	1,500	1,300	1,800
6	1,700	800	1,200	1,100

Table 3. Material flow and installation cost data.

Using the linearized mathematical formulation for the GQAP presented above which consist of objective function (1') subject to constraints (2) – (6), the optimal solution is obtained for the illustrative example using MPL modeling language (commercial software) and CPLEX 11.0 solver. Since the variables w_{ijkl} are used only to linearize the objective function, the values of these variables do not give any useful information and is not given here. However, $x_{13} = x_{21} = x_{34} = x_{42} = x_{51} = x_{61} = 1$, and all other decision variables are zero. The total cost of the solution (z^*) is \$17,165 which is the sum of \$8000 (total installation cost) and \$9165 (total material handling cost). More specifically, machines 2, 5, and 6 are assigned to location 1. Machine 4, 1, and 3 is assigned to location 2, 3, and 4, respectively. See optimal layout (assignment) in Figure 2. Since each machine is assigned to a location, and the capacity of the location is not exceeded, the optimal solution obtained is feasible. See Table 4. For instance, machines 2, 5, and 6 require 120, 110, and 70 square feet of area, respectively. Thus, a total of 300 square feet of area is required for location 1, which has 300 square feet of area. Therefore, the remaining capacity (unused area remaining) is zero.

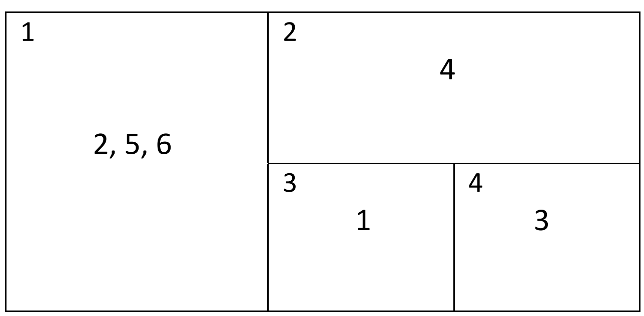

Figure 2. Layout of Production area.

Location	Area (ft^2)	Machine	Area (ft^2) Required	Remaining Capacity (ft^2)
1	300	2, 5, 6	300	0
2	200	4	110	90
3	100	1	90	10
4	100	3	100	0

Table 4. Details of optimal solution.

SOLUTION TECHNIQUES

Since the mathematical model can only be used to solve small-size problems in reasonable time, heuristics are developed for the GQAP. As a result, additional notation is used to give another formulation of the GQAP. The solution is represented as

$$\boldsymbol{S} = (s(1), s(2), \ldots, s(M))$$

where $s(i) = k$, which is equivalent to saying location k is assigned to machine i. The solution is feasible if the following constraints hold,

$$\sum_{\forall i \ s.t.\ s(i)=k} r_i \le C_k \ \text{ for } k = 1, \ldots, N \tag{7}$$

The total cost of the solution is obtained using the following equation.

$$TC(\boldsymbol{S}) = \sum_{i=1}^{M} a_{is(i)} + \sum_{i=1}^{M} \sum_{\substack{j=1 \\ j \ne i}}^{M} c_{ijs(i)s(j)} f_{ij} d_{s(i)s(j)} \tag{8}$$

For example, consider the solution $\boldsymbol{S}$ = (3, 1, 4, 2, 1, 1) for the illustrative example given above (see Tables 1 – 3). That is, $s(1) = 3$, $s(2) = 1$, $s(3) = 4$, $s(4) = 2$, $s(5) = 1$, and $s(6) = 1$. More specifically, machines 2, 5, and 6 are assigned to location 1, machine 4 to location 2, machine 1 to location 3, and machine 3 to location 4. Notice, this is the same optimal solution obtained above using the mathematical formulation. For location 1 (i.e., $k = 1$), consider equation (7).

$$\sum_{\forall i \ s.t.\ s(i)=1} r_i \le C_1 \ \text{ or } \sum_{i=2,5,6} r_i \le C_1 .$$

Recall, $r_2 = 120$, $r_5 = 110$, $r_6 = 70$, and $C_1 = 300$. Therefore,

$$\sum_{i=2,5,6} r_i = 120 + 110 + 70 = 300 \le C_1 = 300.$$

Thus, the capacity constraint for location 1 holds. It is easy to validate that the constraints for locations 2 and 3 also hold. As a result, the solution $\boldsymbol{S}$ = (3, 1, 4, 2, 1, 1) is feasible. Now the total cost of the solution is obtained using equation (8), which is $17,165 which is the sum of $8000 (total installation cost) and $9165 (total material handling cost).

This formulation which minimizes objective function (8) subject to constraint (7) is called the combinatorial optimization problem (COP) formulation. It has fewer variables ($s(i)$), constraints, and solutions compared to the mathematical formulation given earlier. See comparison of models in Table 5 where M = 6 and N = 4 as in the illustrative example. Notice the number of constraints in the mathematical model is based on the number of constraints for constraints (2), (3), and (5) where the restrictions on the variables are not considered. Also, the number of solutions for both models considers all possible sets of values for each variable, whether the solutions are feasible or infeasible. More importantly, the solution space is much less for the COP model, only 4096 solutions compared to 16,777,216. Therefore, it is much more efficient using the COP formulation. Next, a construction algorithm for the GQAP is presented.

$M = 6, N = 4$	Math Model	COP Model
Number of Variables	$M(N) = 24$	$M = 6$
Number of Constraints	$M + N + M(M\text{-}1)(N)(N\text{-}1) = 370$	$N = 4$
Number of Solutions	$2^{M(N)} = 16{,}777{,}216$	$N^M = 4096$

Table 5. Comparison of the models.

The following construction algorithm is used to generate a solution for the GQAP.

Step 1: Initialize capacity of locations (i.e., $C(k) = \{C(1), C(2), \ldots, C(N)\}$).
Initialize space requirement of machines (i.e., $r(i) = \{r(1), r(2), \ldots, r(M)\}$).

Step 2: Sort machines in descending order with respect to $r(i)$ in the eligible machine set (EMS) and break ties by selecting machine with lower machine number.

Step 3a: Set $k = 1$;

Step 3b: If $k > N$, then go to step 5b. Else go to position 1 of the EMS (i.e., set $u = 1$).

Step 4: Set i = the machine in the uth position in EMS.

Step 5a: If $r(i) \leq C(k)$

Assign machine i to location k (i.e., set $s(i) = k$), and set $C(k) = C(k) - r(i)$;
Remove machine i from EMS. If EMS is empty, then go to step 5b;
If $C(k) < r(i)$ for $Last(i)$ in EMS, then set $k = k + 1$ and go to step 3b;
Else Go to step 4.

Else Set $u = u + 1$ and go to step 4.

Step 5b: Terminate algorithm. If EMS is empty, then display feasible solution $\boldsymbol{S}$. Else display "No feasible solution!"

To illustrate the construction algorithm, consider illustrative example presented earlier (see Tables 1 – 3).

Iteration 1: In step 1, let $C(k) = \{300, 200, 100, 100\}$ and $r(i) = \{90, 120, 100, 110, 110, 70\}$. In step 2, obtain **EMS** = {2, 4, 5, 3, 1, 6}, and set $k = 1$ (start with location 1) in step 3a. In step 3b, since $k = 1 < 4 = N$, set $u = 1$ (start at position 1 in EMS). Since machine 2 is in position 1, set $i = 2$ in step 4. In step 5a, since $r(2) = 120 < 300 = C(1)$, set $s(2) = 1$ (assign machine 2 to location 1), obtain $C(1) = C(1) - r(2) = 300 - 120 = 180$, and remove machine 2 from EMS. Thus, **EMS** = {4, 5, 3, 1, 6}. Since $C(1) = 180 > 70 = r(6)$, go to step 4.

Iteration 2: In step 4, set $i = 4$. In step 5a, since $r(4) = 110 < 180 = C(1)$, set $s(4) = 1$, obtain $C(1) = C(1) - r(4) = 180 - 110 = 70$, and obtain **EMS** = {5, 3, 1, 6}; Since $C(1) = 70 = r(6)$, go to step 4.

Iteration 3: In step 4, set $i = 5$. In step 5a, since $r(5) = 110 > 70 = C(1)$, set $u = 1 + 1 = 2$, and go to step 4.

Iteration 4: In step 4, set $i = 3$. In step 5a, since $r(3) = 100 > 70 = C(1)$, set $u = 2 + 1 = 3$, and go to step 4.

Iteration 5: In step 4, set $i = 1$. In step 5a, since $r(1) = 90 > 70 = C(1)$, set $u = 3 + 1 = 4$, and go to step 4.

Iteration 6: In step 4, set $i = 6$. In step 5a, since $r(6) = 70 = C(1)$, set $s(6) = 1$, obtain $C(1) = C(1) - r(6) = 70 - 70 = 0$, and obtain **EMS** = {5, 3, 1}. Since $C(1) = 0 < 90 = r(1)$, set $k = k + 1 = 2$, and go to step 3b.

Iteration 7: In step 3b, since $k = 2 < 4 = N$, set $u = 1$, and set $i = 5$ in step 4. In step 5a, since $r(5) = 110 < 200 = C(2)$, set $s(5) = 2$, obtain $C(2) = C(2) - r(5) = 200 - 110 = 90$, and obtain **EMS** = {3, 1}. Since $C(2) = 90 = r(1)$, go to step 4.

Iteration 8: In step 4, set $i = 3$. In step 5a, since $r(3) = 100 > 90 = C(2)$, set $u = 1 + 1 = 2$, and go to step 4.

Iteration 9: In step 4, set $i = 1$. In step 5a, since $r(1) = 90 = C(2)$, set $s(1) = 2$, obtain $C(2) = C(2) - r(1) = 90 - 90 = 0$, and obtain **EMS** = {3}. Since $C(2) = 0 < 100 = r(3)$, set $k = 2 + 1 = 3$, and go to step 3b.

Iteration 10: In step 3b, since $k = 3 < 4 = N$, set $u = 1$, and set $i = 3$ in step 4. In step 5a, since $r(3) = 100 = C(3)$, set $s(3) = 3$, obtain $C(3) = C(3) - r(3) = 100 - 100 = 0$, and obtain **EMS** = {}. Since EMS is empty, go to step 5b. In step 5b, the algorithm is terminated, and the solution $\boldsymbol{S} = \{2, 1, 3, 1, 2, 1\}$ is displayed. In other words, machines 2, 4, and 6 are assigned to location 1, machines 1 and 5 are in location 2, machine 3 is in location 3, and location 4 is not assigned a machine.

The algorithm presented above either yields a feasible solution or no solution. A solution is not obtained when the difference between the total machine requirements (total area required by machines) and total capacity of locations (total area available) is relatively small, and when machines are assigned to locations such that some of the unused capacities of the locations are relatively large. If this is the case, the above algorithm can be modified such that in step 2 the machines can be ordered in ascending order, instead of descending order. Also, instead of starting at the first location, the algorithm can start at the last location ($k = 4$), and reduce k until all machines are assigned to locations. The different combinations of these modifications result in four different algorithms, which can be used such that a solution can always be generated. Next, an algorithm used to improve the constructed solution is presented next.

The following improvement algorithm, called the steepest descent local search heuristic, is used to improve the solution obtained from the construction algorithm presented above.

Step 1: Construct a solution, $S_0 = (s(1), s(2), \ldots, s(M))$, using the above construction algorithm, and obtain its cost, $TC(S_0)$ using objective function (8).
Step 2: Evaluate all feasible solutions obtained from all possible drop/add operations on S_0 and all possible pairwise exchange operations on S_0.
Step 3: Pick best solution, S, with respect to cost, $TC(S)$. If $TC(S) < TC(S_0)$, set $S_0 = S$, $TC(S_0) = TC(S)$, and go to step 2. Else, terminate heuristic and display solution S_0.

The drop/add operation ($u, v; v'$) represents exchanging location v assigned to machine u with location v' (drop v and add v'). For example, if $S_0 = \{\mathbf{2}, 1, 3, 1, 2, 1\}$, then the drop/add operation (1, 2; 4) produces the solution $\{\mathbf{4}, 1, 3, 1, 2, 1\}$. In other words, machine 1 assigned to location 2 is reassigned to location 4. Constraint (7) where $k = 4$ can be used to check for feasibility of solution. Since solution is feasible, the objective function value (OFV) of the solution is obtained using (8). All possible drop/add operations for each machine is considered. Since there are $N = 4$ locations, and each machine is already assigned to one location, there are $N - 1 = 3$ possible operations for each machine. Since there are $M = 6$ machines, there are $M(N - 1) = 6(3)$ possible drop/add operations.

The pairwise exchange operation ($u, v; u', v'$) represents exchanging location v assigned to machine u with location v' assigned to machine u'. For example, if $S_0 = \{\mathbf{2}, 1, \mathbf{3}, 1, 2, 1\}$, then the pairwise exchange operation (1, 2; 3, 3) produces the solution $\{\mathbf{3}, 1, \mathbf{2}, 1, 2, 1\}$. In other words, machine 1 assigned to location 2 exchanges locations with machine 3 assigned to location 3. Constraints (7) where $k = 2$ and 3 can be used to check for feasibility of solution. Since solution is infeasible for $k = 2$, the objective function value (OFV) of the solution is not calculated using (8). All possible pairwise exchange operations are considered. Since two machines are swapping locations and there are $M = 6$ machines, there are a combination of $M = 6$ pick two ($M(M - 1)/2 = 15$) possible pairwise exchange operations. However, if $u' = v'$, the solution does not change and is not considered. For instance, if $S_0 = \{\mathbf{2}, 1, 3, 1, \mathbf{2}, 1\}$, then the operation (1, 2; 5, 2) produces the same solution $\{\mathbf{2}, 1, 3, 1, \mathbf{2}, 1\}$; therefore there are always less than $M(M - 1)/2$ pairwise exchange operations when a location has more than one machine assigned to it.

To illustrate the steepest descent local search heuristic, consider illustrative example presented earlier (see Tables 1 – 3). In iteration 1, the solution $\boldsymbol{S_0} = \{2, 1, 3, 1, 2, 1\}$ is obtained using the proposed construction algorithm in step 1. Also, the total cost (OFV) of the solution S_0 (i.e., $TC(\boldsymbol{S_0})$ = \$21,255) is obtained using (8). In step 2, all possible solutions are obtained for $\boldsymbol{S_0}$. See Tables 6 and 7. Notice most of the solutions are infeasible, since the problem is tightly constraint (i.e., difference between the total machine requirements (total area required by machines = 600 ft^2) and total capacity of locations (total area available = 700 ft^2) is relatively small (100 ft^2)). Nevertheless, operation (4, 1; 5, 2) produces the solution $\boldsymbol{S} = \{2, 1, 3, \mathbf{2}, \mathbf{1}, 1\}$, which has the lowest cost (i.e., $TC(\boldsymbol{S})$ = \$18,050) in step 3. Since $TC(\boldsymbol{S})$ = \$18,050 < \$21,255 = $TC(\boldsymbol{S_0})$, set $\boldsymbol{S_0} = \boldsymbol{S} = \{2, 1, 3, 2, 1, 1\}$, $TC(\boldsymbol{S_0})$ = \$18,050, and go to step 2. In step 3 of iteration 2, operation (3, 3; 4) produces the solution $\boldsymbol{S} = \{2, 1, \mathbf{4}, 2, 1, 1\}$, which has the lowest cost (i.e., $TC(\boldsymbol{S})$ = \$17,460). Since $TC(\boldsymbol{S})$ = \$17,460 < \$18,050 = $TC(\boldsymbol{S_0})$, set $\boldsymbol{S_0} = \boldsymbol{S} = \{2, 1, 4, 2, 1, 1\}$, $TC(\boldsymbol{S_0})$ = \$17,460, and go to step 2. In step 3 of iteration 3, operation (1, 2; 3) produces the solution $\boldsymbol{S} = \{\mathbf{3}, 1, 4, 2, 1, 1\}$, which has the lowest cost (i.e., $TC(\boldsymbol{S})$ = \$17,165). Since $TC(\boldsymbol{S})$ = \$17,165 < \$17,460 = $TC(\boldsymbol{S_0})$, set $\boldsymbol{S_0} = \boldsymbol{S} = \{3, 1, 4, 2, 1, 1\}$, $TC(\boldsymbol{S_0})$ = \$17,165, and go to step 2. In step 3 of iteration 4, operation (6, 1; 2) produces the solution $\boldsymbol{S} = \{3, 1, 4, 2, 1, \mathbf{2}\}$, which has the lowest cost (i.e., $TC(\boldsymbol{S})$ = \$17,240). Since $TC(\boldsymbol{S})$ = \$17,240 > \$17,165 = $TC(\boldsymbol{S_0})$, terminate heuristic and display solution $\boldsymbol{S_0} = \boldsymbol{S} = \{3, 1, 4, 2, 1, 1\}$, $TC(\boldsymbol{S_0})$ = \$17,165. This solution is called a local optimum; however, since the solution is equivalent to the solution obtained using the mathematical model, it is also a global optimum.

#	Operation	Solution	OFV
1	(1, 2; 1)	{1, 1, 3, 1, 2, 1}	--
2	(1, 2; 3)	{3, 1, 3, 1, 2, 1}	--
3	(1, 2; 4)	{4, 1, 3, 1, 2, 1}	$21,580
4	(2, 1; 2)	{2, 2, 3, 1, 2, 1}	--
5	(2, 1; 3)	{2, 3, 3, 1, 2, 1}	--
6	(2, 1; 4)	{2, 4, 3, 1, 2, 1}	--
7	(3, 3; 1)	{2, 1, 1, 1, 2, 1}	--
8	(3, 3; 2)	{2, 1, 2, 1, 2, 1}	--
9	(3, 3; 4)	{2, 1, 4, 1, 2, 1}	$20,695

#	Operation	Solution	OFV
10	(4, 1; 2)	{2, 1, 3, 2, 2, 1}	--
11	(4, 1; 3)	{2, 1, 3, 3, 2, 1}	--
12	(4, 1; 4)	{2, 1, 3, 4, 2, 1}	--
13	(5, 2; 1)	{2, 1, 3, 1, 1, 1}	--
14	(5, 2; 3)	{2, 1, 3, 1, 3, 1}	--
15	(5, 2; 4)	{2, 1, 3, 1, 4, 1}	--
16	(6, 1; 2)	{2, 1, 3, 1, 2, 2}	--
17	(6, 1; 3)	{2, 1, 3, 1, 2, 3}	--
18	(6, 1; 4)	{2, 1, 3, 1, 2, 4}	$20,495

Table 6. Solutions obtained from add/drop operation.

#	Operation	Solution	OFV
1	(1, 2; 2, 1)	{1, 2, 3, 1, 2, 1}	--
2	(1, 2; 3, 3)	{3, 1, 2, 1, 2, 1}	--
3	(1, 2; 4, 1)	{1, 1, 3, 2, 2, 1}	--
4	(1, 2; 5, 2)	Same as S_0	--
5	(1, 2; 6, 1)	{1, 1, 3, 1, 2, 2}	--
6	(2, 1; 3, 3)	{2, 3, 1, 1, 2, 1}	--
7	(2, 1; 4, 1)	Same as S_0	--
8	(2, 1; 5, 2)	{2, 2, 3, 1, 1, 1}	--

#	Operation	Solution	OFV
9	(2, 1; 6, 1)	Same as S_0	--
10	(3, 3; 4, 1)	{2, 1, 1, 3, 2, 1}	--
11	(3, 3; 5, 2)	{2, 1, 2, 1, 3, 1}	--
12	(3, 3; 6, 1)	{2, 1, 1, 1, 2, 3}	--
13	(4, 1; 5, 2)	{2, 1, 3, 2, 1, 1}	$18,050
14	(4, 1; 6, 1)	Same as S_0	--
15	(5, 2; 6, 1)	{2, 1, 3, 1, 1, 2}	--

Table 7. Solutions obtained from pairwise exchange operation.

CONCLUSION

The proposed construction algorithm and steepest descent local search heuristic were coded using the Visual Basic programming language, and the illustrative example was solved on a Pentium IV 1.5GHz PC. It required only 0.17 seconds of run time. Although the optimal solution was obtained for the illustrative example using the proposed heuristics, they may not always produce the optimal solution, especially for large-size problems. Therefore, for future research, a more powerful heuristic, such as tabu search heuristic, which obtains many local optima in search of the global optimum, will be developed for the GQAP.

REFERENCES

Burkard, R. E., Cela, E., Pardalos, P. M., and Pitsoulis, L.S. The Quadratic Assignment Problem. In *Handbook of Combinatorial Optimization*, D.-Z. Du and P.M. Pardalos (eds). Boston, MA: Kluwer, 1998.

Cordeau, J.-F., Gaudioso, M., Laporte, G., and Moccia, L. “A memetic heuristic for the generalized assignment problem.” *INFORMS Journal on Computing*, 2006, 18, 433-443.

Glover F. “Future paths for integer programming and links to artificial intelligence.” *Computers & Operations Research* 1986; 13(5), 533-549.

Hahn, P. M., Kim, B.-J., Guignard, M., MacGregor Smith, J., and Zhu, Y.-R. "An Algorithm for the Generalized Quadratic Assignment Problem." *Computational Optimization and Applications*, 2008, 40(3), 351-372.

Holland, J. H. *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press, 1975.

Koopmans, T. C. and Beckmann, M. J. "Assignment Problems and the Location of Economic Activities." *Econometrica*, 1957, 25, 53-76.

Lee, C.-G. and Ma, Z. "The generalized quadratic assignment problem," Research Report, Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Ontario M5S 3G8, Canada, 2004.

Loiola, E. M., de Abreu, N. M. M., Boaventura-Netto, P. O., Hahn, P., and Querido, T. "A Survey for the Quadratic Assignment Problem." *European Journal of Operational Research*, 2007, 176(2), 657-690.

Mateus, G. R., Resende, M. G. C., and Silva, R. M. A. "GRASP with Path-relinking for the Generalized Quadratic Assignment Problem." *Journal of Heuristics*, 2011, 17, 527-565.

Pessoa, A. A., Hahn, P. M., Guignard, M., and Zhu, Y.-R. "Algorithms for the Generalized Quadratic Assignment Problem Combining Lagrangean Decomposition and the Reformulation-Linearization Technique." *European Journal of Operational Research*, 2010, 206(1), 54-63.

Sahni, S. and Gonzales, T. "P-complete Approximation Problems." Journal of the ACM, 1976, 23, 555-565.