The Multi-Story Space Assignment Problem

Peter Hahn*
hahn@seas.upenn.edu

J. MacGregor Smith†
jmsmith@ecs.umass.edu

Yi-Rong Zhu‡
yrzhu@seas.upenn.edu

May 29, 2008

Abstract — **The Multi-Story Space Assignment Problem (MSAP) is an innovative formulation of the multi-story facility assignment problem that allows one to model the location of departments of unequal size within multi-story facilities as a Generalized Quadratic 3-dimensional Assignment Problem (GQ3AP). Not only can the MSAP generate the design of the location of the departments in the facility, the MSAP also includes the evacuation planning for the facility. The formulation, background mathematical development, and computational experience with a branch and bound algorithm for the MSAP are also presented.**

Keywords — Multi-story; space assignment; evacuation

1 Motivation

Many facilities have certain functions or site conditions that require them to accommodate their occupants on multiple stories. Examples of these facilities are hospitals, office buildings, hotels, factories, warehouses, oil platforms, naval vessels, and so on. Multi-story facilities should be designed to achieve the following objectives: i) provide a safe and secure environment; ii) be structurally sound, environmentally efficient, and cost-effective; and iii) accommodate the functions and user needs of the people and equipment that occupy it.

The circulation network is crucial to the infrastructure and layout of the facility because it interconnects the department components of the facility and typically serves as the pathway of the building services (heating and ventilation, air conditioning, telecommunication, plumbing, electrical, and sanitary) as well as providing for emergency egress. In building evacuation problems, the design of the stairwell widths, configuration of the landings, number of stairwells, their location, exits and egress paths are all critically linked. It is a travesty that people are often trampled in a push to get out of the narrow exits of sports stadia or of burning buildings.

1.1 Purpose and Objectives

We seek to utilize the tools of combinatorial optimization to develop a methodology in which multi-story facilities are designed so that the previously mentioned objectives can be realized.

1.2 Outline of Paper

In §2, we discuss the background of the problem and the basic literature relevant to the problem, starting from the original Quadratic Assignment Problem papers. §3 presents the assumptions, definitions, and notation underlying the GQ3AP mathematical model of the problem. §4 discusses the GQ3AP dual ascent procedure for calculating a lower bound on the problem. §5 presents experimental results while §6 elaborates on open questions, future plans, and a summary and conclusions for the research.

2 Problem

In this section of the paper, we provide background on the MSAP problem, its complexity and the general literature related to the problem.

2.1 Background

To arrive at the MSAP formulation, we need to discuss the Quadratic Assignment Problem (QAP), as it represents the first combinatorial optimization formulation that gives rise to our model. Subsequent to that, we introduce the GQAP family of models and eventually the three-dimensional assignment models before presenting the MSAP.

2.2 Literature Review

The QAP covers a broad class of problems that involve the minimization of a total pair-wise interaction cost among N departments. The Koopmans-Beckman[34] version of the problem can be stated with reference to a practical situation where it is desired to locate N departments among N fixed locations, where for each pair of departments (i, k) a certain flow of commodities (or people) $f(i, k)$ is known and for each pair of locations

*Corresponding Author; Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, Pennsylvania. Mailing Address: 2127 Tryon St., Philadelphia, PA 19146-1228.

†Department of Mechanical and Industrial Engineering, University of Massachusetts Amherst, Massachusetts 01003

‡Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, Pennsylvania 19104

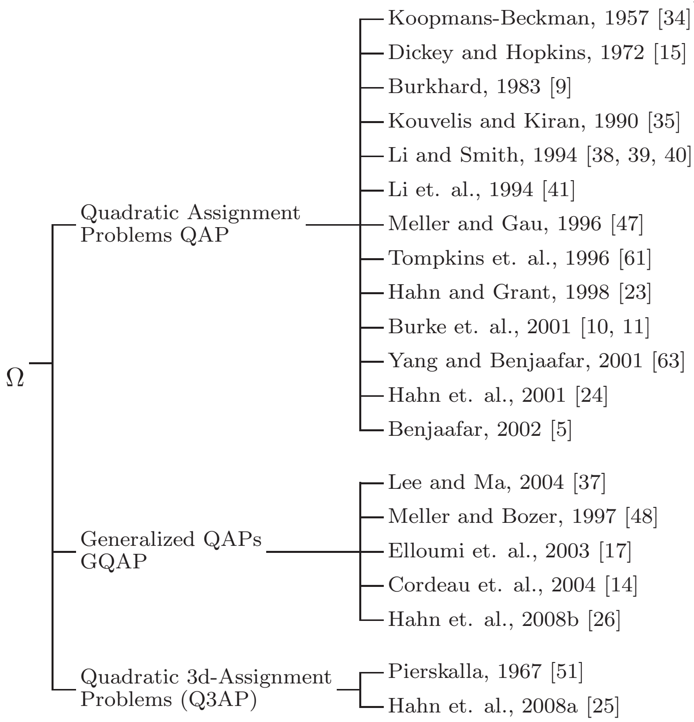

Figure 1: Morphology of QAP Models

(j, n) a corresponding distance $d(j, n)$ is known. The two-way transportation costs between departments i and k, given that i is assigned to location j and k is assigned to location n, are $f(i,k)\cdot d(j,n) + f(k,i)\cdot d(n,j)$. The objective is to find an assignment minimizing the sum of all such transportation costs.

Figure 1 arrays many of the associated references in this area. For example, Dickey and Hopkins [15] applied the QAP for the assignment of buildings in a University campus. Kouvelis and Kiran [35] modified the QAP formulation by explicitly considering the throughput requirements of automated manufacturing systems. Meller and Gau [47] presented a comprehensive review of early research on the single-floor layout problem, which received extensive interest in the literature before 1996. Burke et al. [10] suggest how local search, metaheuristics and evolutionary algorithms provide a promising way to make advances in the problem of allocating space within a university, a problem often modeled as a QAP. Burke, E.K., Cowling, P.I., Keuthen, R. [11] present effective new local and variable neighborhood search heuristics for the asymmetric Traveling Salesman Problem, which is a special case of the QAP. Yang and Benjaafar [63] used both implicit enumeration and a modified 2-opt heuristic in order to find good suboptimum solutions to the Quadratic Assignment Problem. Finally, Benjaafar [5] contrasts QAP solutions with those obtained by queueing network models and focusing on the objective of minimizing work-in-process. In the general case of the QAP, the cost of transportation between departments is known but is not decomposable into a product of a flow and a distance matrix.

The QAP has long been of interest to researchers in various fields, both because of its wide applicability and also its resistance to reliable computer solution. The QAP belongs to the class of $\mathcal{NP}$-complete problems and is considered one of the most difficult since it has no polynomial time approximation scheme[54].

Exact solution strategies for the QAP have been largely unsuccessful for any but the smallest of problem instances (approximately $N \leq 30$). As a result, a significant amount of effort has been put forth by researchers in developing inexact or heuristic methods, which obtain good suboptimal assignments using a reasonable amount of CPU time. For example, see Burkard [9], Li and Smith [38, 39, 40], Li et. al. [41].

To illustrate the success of heuristics on the QAP, an analysis was made of 27 of the 41 unsolved instances on QAPLIB, ranging from size 30 to size 256. Instances whose lower bound calculations were Gilmore Lawler (GL) were not included because the GL bound is poor. Average problem size is 73.1. The average gap between lower bound and best-known feasible solution is only 9%. Historically, it is the lower bound that is way off from the optimum solution. One can safely say that heuristic best-known solutions will generally be only a few percentage points from optimum. Furthermore, keep in mind that linear costs for QAPLIB instances are zero. With non-zero linear costs the gaps would be even smaller.

Stochastic local search (SLS) methods are among the most successful techniques for the approximate solution of quadratic assignment problems (see Hoos and Stützle [29]). These methods are also the best way to provide feasible solutions in large industrial problems with tight constraints. The most promising SLS methods are: the Simulated Annealing algorithm by Connolly [13], the Robust Tabu Search (RoTS) algorithm and Fast Ant System (FANT) by Taillard ([58] and [59]), and an iterated local search (ILS) algorithm by Stützle [57]. Computational experience has shown that these algorithms can find optimal solutions to a wide variety of QAP instances much faster than the best performing exact algorithms. For example, the ILS algorithm finds the best-known solution to the classically difficult Nugent 30 instance in about 1.3 CPU seconds, on average (measured on a 1.2 GHz MP Athlon CPU). These results on the QAP suggest that if one wants to solve the Multistory Space Assignment Problem for buildings having 100 or more stories, these methods are probably the only means to get good quality solutions. Keep in mind, heuristic methods cannot prove optimality of the solutions, but they can be used to provide good initial upper bounds for exact methods.

2.3 *The Generalized QAP (GQAP)*

The Generalized (GQAP) covers a broad class of problems that involve the minimization of a total pair-wise interaction cost among M departments, equipment, tasks or other entities, and where placement of these entities into N possible destinations is dependent upon existing resource capacities at each destination. These problems include finding the assignment of departments to fixed locations given limited area capacities at each possible location. The Lee and Ma[37] version of the problem can be stated with reference to a practical situation where it is desired to locate M departments among N fixed locations, where for each pair of departments (i, k) a certain traffic flow of commodities $f(i, k)$ is known and for each pair of locations (j, n) a corresponding distance $d(j, n)$ is known. The two-way transportation costs between departments i and k, given that i is assigned to location j and k is assigned to location n,are *f(i,k)·d(j,n) + f(k,i)·d(n,j).* The objective is to find an assignment minimizing the sum of all such transportation costs given that the capacity or resource constraints are met. In the general case of the GQAP, the cost of transportation between departments is known but is not decomposable into a product of a flow and a distance matrix.

Since the GQAP is relatively new, little work has been done to apply it to building design or facility location, in general. Notable is the work by Meller and Bozer[48] on the Multi-Floor Facility Layout Problem. They define the facility layout problem as finding a feasible, non-overlapping arrangement of departments (with given area requirements) within a facility (building) to minimize the interaction cost between departments. Interaction cost between two departments is typically expressed as the flow times the distance between departments. In general, departments have unequal area requirements, and some of them may be constrained *a priori* to certain locations in the building. Meller and Bozer appear to be the first to formulate the multi-floor facility layout problem (MFFLP) as a GQAP. They proceed to solve the MFFLP in two stages. Stage 1 assigns departments to floors in an attempt to minimize the inter-floor handling cost. Stage 2 determines the layout of each floor based on the assignments of stage 1. They argue that this approach not only comes up with near optimum solutions, but that it also has management appeal, since there are many management considerations that are not easy to quantify as costs.

Lee and Ma [37] coined the name Generalized Quadratic Assignment Problem. In their paper, they propose three Integer Linear Programming (ILP) formulations of the problem and a branch-and-bound algorithm to optimally solve the GQAP. They proceed to conduct computational experiments to demonstrate the performance of the proposed approaches. Their branch-and-bound algorithm works in two phases. In the first, they construct a feasible solution by means of two different greedy algorithms. They use the best feasible solution found as the initial upper bound for the branch-and-bound enumeration. In the second phase, at each node of the search tree, they calculated a lower bound using an approach akin to a method suggested by Adams and Johnson [2] for calculating lower bounds for the QAP. This method involves a reformulation-linearization of the GQAP and subsequent application of a Lagrangean dual approach. The resulting Lagrangean dual is solved by solving a series of N x M + 1 Generalized Assignment Problems (GAPs), where N is the number of available locations and M is the total number of departments to be located.

Hahn et al. [26] provide the most recent and most promising solution methods for the GQAP. They developed a branch-and-bound exact solution method and a solution heuristic. Their exact solution method is based on a first-level Reformulation Linearization Technique (RLT) dual ascent procedure that performs better than previously existing competing strategies. RLT is known for generating tight linear programming relaxations, not only for constructing exact solution algorithms, but also to design powerful heuristic procedures for large classes of discrete combinatorial and continuous non-convex programming problems [55]. Herein, we refer to first-level RLT as RLT1. Hahn et al.'s heuristic solution method is competitive with others.

In Table 1 of [26], several standard instances have been selected from a web site dedicated to the Task Assignment Problem (TAP) and the Constrained Task Assignment Problem (CTAP) [http://cedric.cnam.fr/oc/TAP/TAP.html]. Comparisons were also made with instances of the GQAP devised by Cordeau, et al. [14] and Lee and Ma [37].

The RLT1 branch and bound is generally faster than the other methods and is almost eleven times faster than the only other known result for the difficult Lee and Ma 16x7 instance. We are continuing to study the structure of the GQAP and are experimenting with various Lagrangian relaxations in an attempt to improve the tightness and speed of GQAP lower bound calculations.

2.4 The Quadratic 3-dimensional Assignment problem (Q3AP)

William P. Pierskalla [51] introduced the Quadratic 3-dimensional Assignment Problem (Q3AP) in a technical memorandum. The work was never published in the open literature. Since then, little on the subject has appeared. Hahn et al. [25] re-discovered the Q3AP while working on a problem arising in data transmission system design. The Q3AP is an extension of two NP-hard problems, the QAP and the 3AP (see [20]). Thus, it is easy to see that the Q3AP is also NP-hard. Our interest in the Q3AP stems from the fact that it is applied to problems where the objective is to minimize linear and quadratic costs associated with a pair of independent simultaneous one-to-one assignments. Such a problem arises in the design of wireless communication systems, wherein a digital message is repeated two times. During each of the repeats, the assignment of data word to transmitted symbols is modified. The Q3AP models the problem of optimizing the two assignments in such a way that the transmission errors are minimized. In the design of multi-story buildings, it is necessary to also simultaneously assign departments to locations and the same departments to escape exits in the face of quadratic costs. Whereas, the assignments in the Q3AP are one-to-one, the assignments in multi-story building design tend to be many-to-one. We will discuss this difference later on in this paper.

Hahn et al. [25] are the only ones to have solved Q3AP instances. They developed a branch-and- bound algorithm based upon the best techniques available for solving the QAP, as well as four different heuristic solution methods whose genesis also came from previous work applied to solving the QAP. Implementation of the four heuristic algorithms was done in a reasonably straightforward way. However, implementing the exact algorithm required the development of new lower bounds for the 3AP. Although the computational results are encouraging, they also illustrate the level of difficulty associated with the Q3AP. Presently, the exact solution method, based upon a reformulation linearization technique lower bound, is computationally feasible only for Q3AP instances of size 13 or smaller. Stochastic local search (SLS) techniques, which provide optimum or near optimum solutions for large and difficult QAP instances, are essential for reaching high quality solutions to reasonable size Q3AP instances. Among the four Q3AP heuristic algorithms, the overall best performance was given by an iterated local search algorithm. However, thus far, even this best performing SLS algorithm incurs, for small Q3AP instances, computation times for reaching optimal or best-known solutions that are orders of magnitudes larger than for similar size QAP instances. Clearly, much more work is needed on this challenging and yet important new combinatorial optimization problem. At this moment, work is being completed on a parallel implementation of the Q3AP branch-and-bound method. This parallel code is not only an instrument for solving exactly large instances, but it will also enable experimentation for improving the runtime of the exact solution algorithm.

3 Mathematical Model

We provide here the necessary assumptions, definitions and notation underlying the mathematical model, as well as the formulation of the MSAP and its associated family of QAP models.

3.1 Assumptions

a) *Distribution of the occupants on each floor is uniform and their travel time to and through the corridor and stairwell segments is a Poisson process λ_i from each department i to a stairwell. While a non-homogenous Poisson Process is probably more realistic, assuming a Poisson Process for the peak hour of the evacuation process is a reasonable assumption.*

b) *The footprint of the facility is a rectilinear polygon $\mathcal{P}$, see Figure 2, or can be closely approximated by a simple polygon. The polygon need not be convex but it is closed and bounded. The reason for the rectilinear polygon is that we also relate the optimization problem in this paper to the problem of stairwell location which has been treated in a separate paper [62].*

c) *We will assume the travel distance metric is rectilinear (Manhattan) and the area per floor is given. The structural system either is pre-defined or else is to be designed after this analysis.*

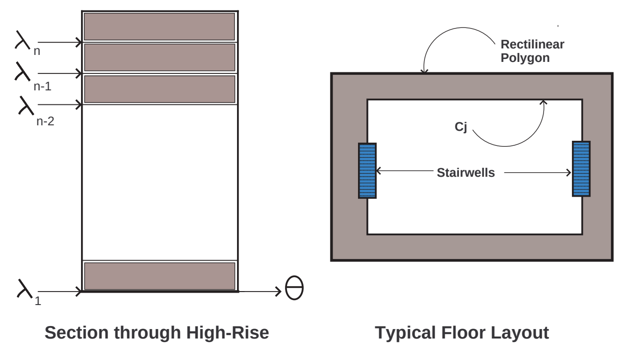

Figure 2: Multi-Story Facility Problem

3.2 Definitions

1. *Occupants are people who reside within the facility in the departments on the various floors and who must be evacuated in an emergency. There are a total of P occupants $P = \sum_{i=1}^{M} P_i$ where i is the department index and M is the total number of departments. The P and P_i helps us determine the magnitude of the peak hour λ_i per floor.*

2. *The circulation topology $\mathcal{C}_N$ is the collection of corridors and stairwells on each floor N. This is either known or has to be generated. This is important in the experimental results.*

3.3 GQAP/Q3AP Problem Formulation

Given M departments, where for each department i the number of people P_i per department and the amount of space required for the department a_i is known, how shall one allocate the M departments to N floors, such that the evacuation time is minimized?

The objective here is to find an assignment that minimizes the sum of all one-way evacuation costs. Certain constraints must be met: No department may be split between different floors. And, the available space on each floor may not be exceeded. This apparent limitation (namely that departments may not be split between floors) is not really a limitation of the model discussed in this paper. By subdividing departments into smaller entities (such as sub-departments) one is able to use the model to assign these smaller indivisible entities to floors. In fact we have implemented this feature of the model in two of the experimental results discussed in §5. In those experiments, we arbitrarily divided departments in half. In practical situations, there will be decisions to be made as to how to split departments and how to prevent them being split into small pieces that are spread out too much. Making these decisions intelligently would require additional analysis and optimization efforts that are beyond the scope of this paper.

A secondary, but nevertheless important, issue: Consider the same M departments and N floors. For each pair of departments (i, k) a certain flow of units (*i.e.*, occupants or materials) $f(i, k)$ is known and for each pair of floors (j, n) a corresponding cost to move one unit of flow between floors $d(j, n)$ is known. The secondary objective is to encourage assignments that minimize the sum of all transportation costs, given that the above constraints are met. Both primary and secondary issues are dealt with in the Multi-Story Assignment Problem (MSAP), defined for the first time here:

The following notation is suggested:

$M :=$ number of departments need to be placed

$N :=$ number of available floors

$S :=$ number of stairwells

$a_i :=$ floor area needed by department i

$A_j :=$ maximum usable floor area on floor j

$f_{ik} :=$ flow units from department i to department k

$d_{jn} :=$ cost per flow unit travel between floor j and floor n.

$e_{js} :=$ distance between floor j and stairwell exit s from building

$\lambda_i :=$ Poisson arrival rate of persons in the evacuation from department i

$\mu_s :=$ service rate capacity of stairwell s

$x_{ij} :=$ 1 if department i is assigned to floor j, and is 0 otherwise.

$y_{is} :=$ 1 if department i is assigned to stairwell s, and is 0 otherwise.

The Multi-story Space Assignment Problem is now stated as

$$Min\left\{ \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{s=1}^{S}\lambda_i e_{js} x_{ij} y_{is} + \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{\substack{k=1\\k\neq i}}^{M}\sum_{n=1}^{N} f_{ik} d_{jn} x_{ij} x_{kn} \right\} \quad (1)$$

subject to the following constraints on **X** and **Y**:

$$\sum_{i=1}^{M} a_i x_{ij} \leq A_j \; (j = 1, 2, \cdots, N) \quad (2)$$

$$\sum_{j=1}^{N} x_{ij} = 1 \; (i = 1, 2, \cdots, M) \quad (3)$$

$$\sum_{i=1}^{M} \lambda_i y_{is} \leq \mu_s \; (s = 1, 2, \cdots, S) \quad (4)$$

$$\sum_{s=1}^{S} y_{is} = 1 \; (i = 1, 2, \cdots, M) \quad (5)$$

$$x_{ij} = 0, 1 (i = 1, 2, \ldots, M; j = 1, 2, \ldots, N), \quad (6)$$

$$y_{is} = 0, 1 (i = 1, 2, \ldots, M; s = 1, 2, \ldots, S), \quad (7)$$

X and **Y** are said to be a 'solution'. In the last summation of (1), $k \neq i$ because permitting the equality would imply that a given department could be assigned to more than one floor.

The objective function (OF) adds the costs associated with inter-departmental traffic to the costs associated with the evacuation of people from the building. One wishes, of course, to minimize both of these important quantities. Adding two so differently perceived quantities is both ambitious and arguable. However, if careful thought is applied when balancing the effects of these two contributors to objective function value, the correct balance may be made. Fortunately, there is a simple mechanism to explore the tradeoffs. One simply manipulates the average values of the first and second terms in (1). The ratio of these two average values (OF-ratio) then becomes a measure of the importance of one quantity (escape cost) versus the importance of the other quantity (inter-department cost). Choice of OF-ratio when actually designing a structure, is a problem best left to the architect, investors, government agencies, insurers, future occupants and other interested parties that have a vested interest in the cost and safety of the resulting structure. The job of the architect in utilizing this optimization tool, is to present alternatives to the interested parties, so that an intelligent decision can be made as to what value of OF-ratio to use in the optimization algorithm.

One might criticize the model of (1) through (7), by suggesting that the model does not take into account the cost of traffic flow within a department, which might be more expensive than the cost of flow between departments. This could be especially true for large departments that are spread out on a single floor. We argue, however, that the model we have developed is not confined to considering only traffic between floors. It is a simple matter to divide the building into "compartments" rather than floors, wherein several compartments can be located on a floor. Then, $[d_{jn}]$ would represent cost of moving between compartments whether they are on the same floor or on different floors.

Constraints (2) make sure that the capacity of each space is not exceeded, constraints (3) make sure that each department gets assigned only one space, constraints (4) make sure that the capacity of each escape route is not exceeded, and constraints (5) make sure that each department is assigned only one primary escape route.

We recognized that the MSAP, as given by Eqs. 1 - 5, is a generalization of both the GQAP and the Q3AP, thus making it an NP-hard problem. The codes we have developed for solving the GQAP and the Q3AP are an excellent basis for developing heuristics as well as exact methods for generating sub-optimal and optimal solutions.

We have therefore come up with the following strategy for solving the MSAP:

The current implementation is a branch-and-bound algorithm for solving the MSAP. This development is being done in the context of a Generalized Quadratic 3-dimensional Assignment Problem (GQ3AP), which is only slightly more general than the MSAP. The basis for this algorithm is the branch-and-bound code for the Q3AP. However, the objective function involves generalized (*i.e.*, many-to-one) assignments rather than the one-to-one assignments of the Q3AP.

The Q3AP lower bound calculation is based on the calculation of lower bounds for a series of 3-dimensional Assignment Problems (3APs) [32], and the calculation of lower bounds on the GQAP is based on calculation of a series of lower bounds on the Generalized Assignment Problem (GAP) [26]. Calculating the MSAP lower bound involves a reformulation-linearization of the GQ3AP and subsequent application of a Lagrangean dual approach. The resulting Lagrangean dual is solved by solving N x M x S plus one continuous Generalized 3-dimensional Assignment Problems (G3AP), where N is the number of available locations, M is the total number of departments to be located and S is the number of stairwells. Since the G3AP is an extension of the 3AP, it is easily seen that this new problem is also NP-hard. Rather than solve the individual continuous G3APs, we instead calculate efficient and tight lower bounds for these two problems.

In order to explain the basis for this approach we introduce here the formulation of the GQ3AP and show the relationships of our exact solution method to linear programming. What follows in the next section has a history in one of the several linearization techniques for the QAP listed in [50]. Specifically, we base our approach on the linearizations suggested by Adams and Sherali in [3] and [4]. For more details, we refer you to [23] and [1].

4 Dual Ascent Procedure for Calculating Lower Bounds

4.1 GQ3AP formulation

The GQ3AP is given by:

$$Min\Big\{ \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{s=1}^{S} C_{ijs}x_{ij}y_{is} + \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{s=1}^{S}\sum_{\substack{k=1\\k\neq i}}^{M}\sum_{n=1}^{N}\sum_{t=1}^{S} C_{ijsknt}x_{ij}y_{is}x_{kn}y_{kt}\Big\} \quad (8)$$

subject to constraints (2) through (7),
where, for purposes of solving the MSAP, we set

$$C_{ijs} = \lambda_i e_{js} \quad (9)$$

and

$$C_{ijsknt} = f_{ik}d_{jn} \quad (10)$$

where all the notations are defined just as before in the MSAP.
Then, the GQ3AP objective function becomes

$$\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{s=1}^{S} \lambda_i e_{js}x_{ij}y_{is} + \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{\substack{k=1\\k\neq i}}^{M}\sum_{n=1}^{N} f_{ik}d_{jn}x_{ij}x_{kn} \quad (11)$$

which is the same objective function as for the MASP.
An equivalent formulation of the GQ3AP can be stated as

$$Min\Big\{ \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{s=1}^{S} C_{ijs}x_{ijs} + \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{s=1}^{S}\sum_{\substack{k=1\\k\neq i}}^{M}\sum_{n=1}^{N}\sum_{t=1}^{S} C_{ijsknt}x_{ijs}x_{knt}\Big\} \quad (12)$$

where $x_{ijs} := 1$ if department i is assigned to floor j and to stairwell s, and is 0 otherwise. Eq.(12) is subject to the following constraints:

$$\sum_{j=1}^{N}\sum_{s=1}^{S} x_{ijs} = 1 \ (i = 1, 2, \cdots, M) \quad (13)$$

$$\sum_{i=1}^{M}\sum_{s=1}^{S} a_i x_{ijs} \leq A_j \ (j = 1, 2, \cdots, N) \quad (14)$$

$$\sum_{i=1}^{M}\sum_{j=1}^{N} \lambda_i x_{ijs} \leq \mu_s \ (s = 1, 2, \cdots, S) \quad (15)$$

$$x_{ijs} = 0, 1 (i = 1, 2, \ldots, M; j = 1, 2, \ldots, N; s = 1, 2, \ldots, S), \quad (16)$$

4.2 The RLT1 formulation for the GQ3AP

We proceed to transform the original model of the GQ3AP into an equivalent linearized mixed integer programming model G3RLT, similar to those models used successfully on the Q3AP and the GQAP. We follow the rules given in [2]. First, multiply each of equality constraints (13) and each of the inequality constraints (14) through (15), written as $\sum_{n=1}^{N}\sum_{t=1}^{S} x_{knt} = 1$, $\sum_{k=1}^{M}\sum_{t=1}^{S} a_k x_{knt} \le A_n$ and $\sum_{k=1}^{M}\sum_{n=1}^{N} \lambda_k x_{knt} \le \mu_t$, by each of N^3 binary variables x_{ijs}. Append all these new restrictions. Express the resulting product in the order $x_{ijs}x_{knt}$. Next, explicitly include the trivial restrictions $x_{ijs}x_{knt} = x_{knt}x_{ijs} \forall (i,j,s,k,n,t), k > i$. Then, linearize every occurrence of each product with a single continuous variable $z_{ijsknt} = x_{ijs}x_{knt}$, where $z_{ijsknt} :=$ 1 if department i is assigned to floor j and to stairwell s and department k is assigned to floor n and to stairwell t, and is 0 otherwise. Finally, consider the solution structure of the GQ3AP, which indicates that $z_{ijsint} = 0 \forall (i,j,s,n,t)$ with the exception that $z_{ijsijs} = x_{ijs} \forall (i,j,s)$. The RLT1 formulation G3RLT then results in

$$Min\Big\{\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{s=1}^{S} C_{ijs}x_{ijs} + \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{s=1}^{S}\sum_{\substack{k=1\\k\neq i}}^{M}\sum_{n=1}^{N}\sum_{t=1}^{S} C_{ijsknt}z_{ijsknt}\Big\} \quad (17)$$

such that:

$$\sum_{n=1}^{N}\sum_{t=1}^{S} z_{ijsknt} = x_{ijs} \quad (i,k = 1,2,\cdots,M; j = 1,2,\cdots,N), \; (s = 1,2,\cdots,S; k \neq i), \quad (18)$$

$$\sum_{k=1}^{M}\sum_{t=1}^{S} a_k z_{ijsknt} \le A_n x_{ijs} \quad (i = 1,2,\cdots,M; j = 1,2,\cdots,N, s = 1,2,\cdots,S), \quad (19)$$

$$\sum_{k=1}^{M}\sum_{n=1}^{N} \lambda_k z_{ijsknt} \le \mu_t x_{ijs} \quad (i = 1,2,\cdots,M; j = 1,2,\cdots,N, s = 1,2,\cdots,S), \quad (20)$$

$$z_{ijsknt} = z_{kntijs} \quad (i,k = 1,2,\cdots,M; j,n = 1,2,\cdots,N), \; (s,t = 1,2,\cdots,S; k \ge i), \quad (21)$$

$$\sum_{j=1}^{N}\sum_{s=1}^{S} x_{ijs} = 1 (i = 1,2,\cdots,M), \quad (22)$$

$$\sum_{i=1}^{M}\sum_{s=1}^{S} a_i x_{ijs} \le A_j (j = 1,2,\cdots,N), \quad (23)$$

$$\sum_{i=1}^{M}\sum_{j=1}^{N} \lambda_i x_{ijs} \le \mu_s (s = 1,2,\cdots,S), \quad (24)$$

$$z_{ijsknt} \ge 0 (i,k = 1,2,\cdots,M; j,n = 1,2,\cdots,N), \; (s,t = 1,2,\cdots,S; k \neq i), \quad (25)$$

$$x_{ijs} \in \{0,1\} \; (i = 1,2,\cdots,M; j = 1,2,\cdots,N), \; (s = 1,2,\cdots,S), \quad (26)$$

Problem GQ3AP and G3RLT are equivalent in the following sense. Given any feasible solution to the GQ3AP, there exists a y such that (x,y) is feasible to the G3RLT with the same objective value. Conversely, for any feasible solution (x,y) to the G3RLT, the corresponding y is feasible to the GQ3AP with the same objective value. This proof is given in [64]

The mathematical structure of G3RLT (expressions (17) through (26)) can be readily exploited via Lagrangian duality. An important step of establishing the Lagrangian dual of a combinatorial optimization problem is to partition the constraints into two sets: the constraints to be dualized into objective function using suitable Lagrangian multipliers and the remaining constraints which restrict the relatively easy subproblems.

4.3 An RLT1 lower bound for the GQ3AP

In order to apply the Lagrangian dual procedure to solve the G3RLT, we first obtain a smaller model via the substitution of variables based on constraints (21), thereby reducing the number of variables and constraints, in addition to halving the number of nonnegativity restrictions in (25). Then we use the fact that $x_{ijs}^2 = x_{ijs}$ and replace x_{ijs} by z_{ijsijs}, so the z_{ijsijs}s become 0-1 variables in the new model. Finally we replace the binary restrictions (26) on z_{ijsijs} by $0 \le z_{ijsijs} \le 1$, so that contrary to G3RLT being a mixed-integer programming problem, the new G3RLT1 is a continuous optimization problem. We obtain the following model

$$Min\Big\{\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{s=1}^{S} C_{ijs}z_{ijsijs} + \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{s=1}^{S}\sum_{\substack{k=1\\k>i}}^{M}\sum_{n=1}^{N}\sum_{t=1}^{S} \tilde{C}_{ijsknt}z_{ijsknt}\Big\} \quad (27)$$

where $\tilde{C}_{ijsknt} = C_{ijsknt} + C_{kntijs}$

such that:

$$\sum_{n=1}^{N}\sum_{t=1}^{S} z_{ijsknt} = z_{ijsijs} \quad (i,k=1,2,\cdots,M; j=1,2,\cdots,N), \quad (s=1,2,\cdots,S; k>i), \tag{28}$$

$$\sum_{n=1}^{N}\sum_{t=1}^{S} z_{kntijs} = z_{ijsijs} \quad (i,k=1,2,\cdots,M; j=1,2,\cdots,N), \quad (s=1,2,\cdots,S; k<i), \tag{29}$$

$$\sum_{\substack{k=1\\k>i}}^{M}\sum_{t=1}^{S} a_k z_{ijsknt} + \sum_{\substack{k=1\\k<i}}^{M}\sum_{t=1}^{S} a_k z_{kntijs} \le A_n z_{ijsijs} \quad (i=1,2,\cdots,M; j,n=1,2,\cdots,N, s=1,2,\cdots,S), \tag{30}$$

$$\sum_{\substack{k=1\\k>i}}^{M}\sum_{n=1}^{N} \lambda_k z_{ijsknt} + \sum_{\substack{k=1\\k<i}}^{M}\sum_{n=1}^{N} \lambda_k z_{kntijs} \le \mu_t z_{ijsijs} \quad (i=1,2,\cdots,M; j=1,2,\cdots,N, s,t=1,2,\cdots,S), \tag{31}$$

$$\sum_{j=1}^{N}\sum_{s=1}^{S} z_{ijsijs} = 1 \; (i=1,2,\cdots,M), \tag{32}$$

$$\sum_{i=1}^{M}\sum_{s=1}^{S} a_i z_{ijsijs} \le A_j \; (j=1,2,\cdots,N), \tag{33}$$

$$\sum_{i=1}^{M}\sum_{j=1}^{N} \lambda_i z_{ijsijs} \le \mu_s \; (s=1,2,\cdots,S), \tag{34}$$

$$0 \le z_{ijsknt} \le 1 \; (i,k=1,2,\cdots,M; j,n=1,2,\cdots,N), \quad (s,t=1,2,\cdots,S; k>i), \tag{35}$$

$$0 \le z_{ijsijs} \le 1 \; (i=1,2,\cdots,M; j=1,2,\cdots,N), \quad (s=1,2,\cdots,S), \tag{36}$$

The reduction in the number of variables is not done just for the purpose of reducing computational effort. It is essential to achieving much tighter lower bounds than would be achieved without this important step. Our prior experience with developing tight lower bounds for the QAP [23], GQAP [26] and Q3AP [25] convinced us of the efficacy of this procedure.

Now consider the following Lagrangian relaxation problem LR(ρ,ρ',γ) on the G3RLT1, whereby constraints (28) , (29) and (32) are placed into the objective function using multipliers ρ, ρ' and γ respectively. The development of this Lagrangian problem is given in [64].

$$Min\Big\{\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{s=1}^{S} \bar{C}_{ijs} z_{ijsijs} + \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{s=1}^{S}\sum_{\substack{k=1\\k>i}}^{M}\sum_{n=1}^{N}\sum_{t=1}^{S} \bar{C}_{ijsknt} z_{ijsknt} + \sum_{i=1}^{M}\gamma_i\Big\} \tag{37}$$

where

$$\bar{C}_{ijs} = C_{ijs} + \sum_{\substack{k=1\\k>i}}^{M}\rho_{ijsk} + \sum_{\substack{k=1\\k<i}}^{M}\rho'_{ijsk} - \gamma_i \quad (i=1,2,\cdots,M; j=1,2,\cdots,N; s=1,2,\cdots,S) \tag{38}$$

and

$$\bar{C}_{ijsknt} = \tilde{C}_{ijsknt} - \rho_{ijsk} - \rho'_{knti} \quad (i,k=1,2,\cdots,M; j,n=1,2,\cdots,N; s,t=1,2,\cdots,S; k>i) \tag{39}$$

Finally, the Lagrangian dual problem LD2 is as follows:

maximize $\theta(\rho,\rho',\gamma)$ where

$$\theta(\rho,\rho',\gamma) = \underset{z}{Min}\Big\{\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{s=1}^{S} \bar{C}_{ijs} z_{ijsijs} + \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{s=1}^{S}\sum_{\substack{k=1\\k>i}}^{M}\sum_{n=1}^{N}\sum_{t=1}^{S} \bar{C}_{ijsknt} z_{ijsknt}\Big\} + \sum_{i=1}^{M}\gamma_i \tag{40}$$

such that constraints (30), (31) and (33) through (36) hold.

LD2 may be written as a series of subproblems

maximize $\theta(\rho,\rho',\gamma)$ where

$$\theta(\rho,\rho',\gamma) = \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{s=1}^{S} P_{ijs}(\rho,\rho',\gamma) + \sum_{i=1}^{M}\gamma_i \tag{41}$$

where each subproblem $P_{ijs}(\rho,\rho',\gamma)$ is solved by

$$P_{ijs}(\rho,\rho',\gamma) = \underset{z}{Min}\Big\{(C_{ijs}+\sum_{\substack{k=1\\k>i}}^{M}\rho_{ijsk}+\sum_{\substack{k=1\\k<i}}^{M}\rho'_{ijsk}-\gamma_i)z_{ijsijs} + \sum_{\substack{k=1\\k>i}}^{M}\sum_{n=1}^{N}\sum_{t=1}^{S}(\tilde{C}_{ijsknt} - \rho_{ijsk} - \rho'_{knti})z_{ijsknt}\Big\} \tag{42}$$

under the constraints

$$\sum_{\substack{k=1\\k>i}}^{M}\sum_{t=1}^{S} a_k z_{ijsknt} + \sum_{\substack{k=1\\k<i}}^{M}\sum_{t=1}^{S} a_k z_{kntijs} \le A_n z_{ijsijs} \quad (n=1,2,\cdots,N), \tag{43}$$

$$\sum_{\substack{k=1\\k>i}}^{M}\sum_{n=1}^{N}\lambda_k z_{ijsknt} + \sum_{\substack{k=1\\k<i}}^{M}\sum_{n=1}^{N}\lambda_k z_{kntijs} \le \mu_t z_{ijsijs} \quad (t = 1, 2, \cdots, S), \quad (44)$$

$$\sum_{i=1}^{M}\sum_{s=1}^{S} a_i z_{ijsijs} \le A_j \ (j = 1, 2, \cdots, N), \quad (45)$$

$$\sum_{i=1}^{M}\sum_{j=1}^{N} \lambda_i z_{ijsijs} \le \mu_s \ (s = 1, 2, \cdots, S), \quad (46)$$

$$0 \le z_{ijsknt} \le 1 \ (i, k = 1, 2, \cdots, M; j, n = 1, 2, \cdots, N), \quad (s, t = 1, 2, \cdots, S; k > i), \quad (47)$$

$$0 \le z_{ijsijs} \le 1 \ (i = 1, 2, \cdots, M; j = 1, 2, \cdots, N), \quad (s = 1, 2, \cdots, S), \quad (48)$$

$$(i, k = 1, 2, \cdots, M; j, n = 1, 2, \cdots, N), \quad (s, t = 1, 2, \cdots, S; k > i). \quad (49)$$

Note in Eq.(42) that within each of the $M \times N \times S$ manifestations of $P_{ijs}(\rho, \rho', \gamma)$ there exists a 3-dimensional matrix of size $(M-1) \times N \times S$ whose elements are $\bar{C} = (\tilde{C}_{ijsknt} - \rho_{ijsk} - \rho'_{knti})$. These elements can be shown to be the cost coefficients of a modified objective function. Each of these elements is multiplied by a corresponding variable z_{ijsknt} of the RLT1 formulation in expressions (27) through (35). This 3-dimensional modified cost matrix of size $(M-1) \times N \times S$ is considered a submatrix of a much larger 6-dimensional matrix. When we use the term *submatrix* below, it is understood to be one of these 3-dimensional modified cost matrices.

Observe that in Eq.42, if the multipliers ρ and ρ' are positive quantities, then these amounts are subtracted from k-planes within the 3-dimensional modified cost submatrices and added to the corresponding linear cost element C_{ijs}. In [23] and in [26] similar operations of this nature are referred to as Class 1 operations. Observe next in Eq.41 and Eq.42 that, if the multipliers γ are also positive quantities, then those amounts are subtracted from i-planes within the 3-dimensional matrix of modified linear costs $\{\bar{C}_{ijs}\}$ and added to the value of $\theta(\rho, \rho', \gamma)$. In the above references, these latter operations are referred to as Class 2 operations. It can be shown that Class 1 operations (*i.e.*, the choice of values of multipliers ρ and ρ') have the property that the cost of all assignment $\{x_{ij}, y_{is}\}$ for the modified objective function are the same as they would be for the original objective function. It can be shown that Class 2 operations (*i.e.*, the choice of values of multipliers γ) will change the cost of all assignments $\{x_{ij}, y_{is}\}$ in such a way that all assignment costs are shifted by an identical amount, thus preserving their order with respect to cost. This is explained in Chapter 5 of [64].

4.4 *Dual Ascent Lower Bound Calculation*

Our RLT1 Dual Ascent Procedure algorithm is based on the concept that constant amounts are subtracted from the k-planes of each and every 3-dimensional modified cost coefficient submatrix $\{\bar{C}_{ijsknt}\}$ and added to the corresponding linear cost element $\bar{C}_{ijs}$ of the submatrix (Class 1 operations). Then, constant amounts are subtracted from the i-planes of the modified linear cost matrix $\{\bar{C}_{ijs}\}$ and added to $\theta(\rho, \rho', \gamma)$, which comprises a lower bound value (Class 2 operations). These are the steps:

Step 1: For each of the $M \times N \times S$ 3-dimensional modified cost submatrices, first collect into the submatrix those costs corresponding to complementary variables z_{ijsknt} and z_{kntijs}. Then, within the submatrix, subtract the minimum modified cost from each k-plane and add it to the corresponding linear cost element. Adding complementary costs first assures the largest possible transfer of quadratic to linear costs.

Step 2: After **Step 1** has been done for all the submatrices, subtract the minimum cost from each i-plane of the linear cost matrix and add it to a reduction constant which constitutes a lower bound to the problem. If the resulting pattern of zeros satisfies the problem constraints, the procedure terminates with an optimum solution to the problem. If not, continue to **Step 3.** Stop if the bound increase is poor.

Step 3: Scan the matrix of modified linear costs for positive elements. Record locations of the non-positive (*i.e.*, zero) linear costs as this information will be needed in **Step 4.** Divide each positive linear cost element into $M-1$ approximately equal parts and add each part to a different (non linear-cost) k-plane of its submatrix. Positive linear costs are divided into $M-1$ approximately equal parts as follows: First of all, one must understand that it is essential that round-off errors must never be allowed in manipulating the **C** matrix. Thus, the **C** matrix in our algorithm is always integer. So, to divide the linear cost into $M-1$ almost equal parts, we divide by $M-1$ and round that down to the nearest integer. That provides $M-2$ equal parts. We then add those parts up and subtract from to get the remaining part. The motivation here is that by replacing costs into submatrices, there is a new opportunity to make those linear costs which had low or zero values after **Step 2** larger, permitting even more cost to be eventually moved to the lower bound. **Step 3** leaves the cost matrix **C** dramatically rearranged. Because of this rearrangement, the process can be repeated, *i.e.*, additional costs can be subtracted from within submatrices and moved to the linear cost element. The result is an iterative procedure that produces growth in the lower bound with each round. This growth, while significant, diminishes with each round, so at some point it does not pay to continue the process.

Step 4: Repeat **Step 1,** except this time be sure to do the collection of complementary costs and subtractions first for those submatrices whose linear costs were zero prior to **Step 3.** Go to **Step 2.**

The collecting of complementary costs prior to solving each submatrix is explained by the substitution of variables that eliminated constraints (21) in G3RLT. **Step 1** and **Step 4** are fully explained by formulation LD1. Since **Step 3** is comprised only of Class 1 operations, therefore, it is also consistent. **Step 2** is clarified by formulation LD2 which constitutes both Class 1 and Class 2 operations, as are all four steps of the level-1 dual ascent procedure.

Equality constraints (28), (29) and (32) were dualized and dealt with appropriately in the construction of the dual ascent procedure. The resource constraints (30), (31), (33) and (34) must be enforced in other ways. For the most part, this is done by calculating dual ascent procedure lower bounds on only those assignments that satisfy those four sets of constraints. However, a method similar to that described in Section 3.3 of [26] was implemented that improves the dual ascent procedure bound value, by taking into account the resource constraints. In this method, constraints (30) and (31) are enforced during the RLT1 lower bound calculation by raising to a very high cost value those submatrix elements whose selection would violate these two sets of constraints. Thus, it is possible to improve the lower bound beyond that available if only the equality constraints (28), (29) and (32) were enforced. We refer you to Section 5.4.3 of [64] for a more detailed explanation.

4.5 Advantages of the RLT1 dual ascent procedure

The RLT1 dual ascent procedure has a number of valuable properties that makes it ideal for use in a branch-and-bound algorithm for solving the GQ3AP.

1. The RLT1 dual ascent procedure is still one of the best among competing lower bounding techniques for the QAP, as demonstrated from experimental results, see Table 2 of [42] and Drezner et al. [16]. Moreover, it is the only method that is able to solve the GQAP of any size, see [26].

2. Similar to a number of other lower bounding techniques, the RLT1 dual ascent procedure generates a series of non-decreasing lower bounds for the GQ3AP. More importantly, with each such lower bound, it modifies the GQ3AP objective function so that a new GQ3AP is generated whose feasible solution set is identical to that of the original and whose objective function values are merely lessened by the amount of the lower bound.

3. The dual ascent procedure for a given partial assignment can be stopped as soon as the lower bound on the assumed partial assignment exceeds an upper bound on the original problem. It can also be stopped, in favor of making an additional partial assignment, when it becomes obvious that from its slow progress it is unlikely to ever reach the upper bound.

4. In a branch-and-bound algorithm, fathoming decisions can easily be recorded by arbitrarily increasing the costs of those elements in the cost matrix that correspond to partial assignments which have been eliminated as possibly being optimum. This modifies the GQ3AP, but is assured to fully enumerate all feasible solutions of the original problem.

4.6 The RLT1 dual ascent procedure in branch-and-bound

The RLT1 dual ascent procedure is utilized within a branch-and-bound algorithm as the auxiliary procedure for computing lower bounds. This method is similar to the branch-and-bound algorithm for the GQAP in [26], since the GQ3AP tree search must be limited to only those assignments that meet the resource constraints.

Branching follows the conventional technique of selecting a single 3-dimensional $i-j-s$ assignment at the first (highest) level, as well as at subsequent levels of partial assignment. In order to implement this selection, a linear cost is chosen to be involved in the 3-dimensional assignment. Based on the selection of linear cost C_{ijs}, the 3-dimensional submatrix $\mathbf{C_{ijs}}$ is involved in the assignment. The remaining submatrices of cost matrix $\mathbf{C}$ whose i-plane containing submatrix $\mathbf{C_{ijs}}$ disappear (as they cannot be involved in the assignment) and the problem is thus reduced to a GQ3AP of size $(M-1)\times N\times S$. It turns out that one k-plane likewise disappears from each submatrix of the remainders in the cost matrix C, which make the submatrices to be $(M-1)\times N\times S$ in size.

It is the application of the RLT1 dual ascent procedure lower bounding calculation on the $(M-1)\times N\times S$ size problem that attempts to fathom a partial assignment postulated by the selection of linear cost C_{ijs}. By fathoming, we calculate a lower bound and test it against the best-known upper bound. If the best-known upper bound is exceeded, the partial assignment is eliminated from the problem.

Recall in the previous sections, the dual ascent procedure moves costs out of the $\mathbf{C}$ matrix into a lower bound value, leaving a modified matrix $\mathbf{C}'$. For subsequent branch-and-bound operations along a given partial assignment path, the strategy is to take advantage of this fact and to use this reduced cost matrix $\mathbf{C}'$ for setting up subsequent subproblems deeper into the tree. Thus, lower bounds are calculated not from the original problem, but from the subproblems that the RLT1 dual ascent procedure already processed at earlier (higher) levels of partial assignment. Using the modified matrix $\mathbf{C}'$ rather than the original matrix $\mathbf{C}$ has the additional benefit that the problem is brought closer to dual solution, making it more likely that better feasible solutions will be found or that lower bounds will exceed upper bounds, thus cutting off a branch. The choice of the number of dual iterations at a given partial assignment is a dynamic decision, based on the progress of the lower bound achieved after a fixed number of iterations. If after a small number of iterations at the current partial assignment, sufficient progress is not reached, the lower bounding attempt stops and the algorithm proceeds to make an additional assignment.

Tree search is depth-first and based on single $i-j-s$ assignment. The order in which i assignments are made is very important and has profound influence on branch-and-bound runtime. Levels in the search tree are defined by how many partial assignments are made. The root is where no partial assignments have been made and is considered level 0. A single partial assignment is considered level 1, etc. Since the tree search involves one-to-one-to-one 3-dimensional assignments, at each level of the tree it is necessary only to examine a given i and its assignments to all possible (*i.e.*, feasible) j's and s's.

In the GQ3AP, more than one i can reside at a given j-s pair. Therefore, we have to exhaust all possible j's and s's at a given level, before we can determine if a better solution exists or if a branch of the tree at that level can be cut off. The exception to this rule is that no further consideration need be given to evaluating assignments if the resource constraints tell us that the locations can accept no further assignments.

The search strategy for selecting the next node is a simple depth-first strategy. If fathoming a given node is unsuccessful, we make an additional partial assignment, thus increasing the depth into the tree. If a node has been fathomed successfully, the next available node is selected. Partial assignments at a given level are selected according to a set of look-ahead bounding calculations that essentially determine the difficulty of eliminating the branch containing that assignment. Assignments selected in the order of decreasing difficulty are made when it is desired to find attractive feasible solutions to replace the upper bound, early in the search.

When the algorithm accumulates sufficient fathoming information, permanent decisions can be made on the assignment matrix that certain elements of the assignment matrix $Y = [y_{ijsknt}]$ are zero. These are recorded in the modified cost matrix $\mathbf{C}'$ by setting the corresponding element costs to be a very large value GREAT. This effectively bars the *decided-zero* elements from inclusion in a level-1 RLT dual ascent procedure solution, and focuses the reduction efforts on those elements still eligible for consideration. The communication of permanent decisions generally permits additional cost to be extracted from the matrix, resulting in an improved lower bound.

Our branch-and-bound code for the GQ3AP was developed in FORTRAN 77, using the techniques developed previously for the branch-and-bound algorithms for the Q3AP and the GQAP. Results of our experiments, using the RLT-1 branch-and-bound code, are presented in the next section.

5 Experimental Results

In this part of the paper, we present the algorithmic results for a set of problem instances with varying number of departments (M), floors (N), stairwells (S), and evacuation population flow characteristics.

5.1 Preliminaries

Figure 3 illustrates the general arrangement of the stairwells which represents the circulation topology of the evacuation process. We varied the number of stories in these experiments from seven to eight and the number of departments from ten to thirteen and the number of stairwells from two to three. Figure 4 shows a typical facility section configuration for the stairwells and stories and evacuation target at the ground level. For the first example, it is assumed that the occupants of the seven-story building comprise ten different departments and that there are two stairwells strategically located that can be accessed on every floor. In the eight story building, there are either two or three strategically located stairwells that can be accessed on every floor.

In the first version of this paper, several reviewers recognized the fact that we had examined only the situation where the entire departments were wholly assigned to single stairwells. Our choice of examples was unfortunate, since our model is not restricted to this situation. With precisely the same model, one can subdivide departments, so that their subdivisions can be assigned independently to stairwells. If desired, flow values can be chosen so that sub-departments will always be assigned to a single floor. Though, the model is general enough to allow sub-departments to be located one over another on different floors. We have therefore added experiments to show the simplicity and effectiveness of such an approach. In a real situation, the choice of how to subdivide departments and how to avoid spreading them too much would be a subject for future research.

A typical set of flows for the ten departments are given in the flow matrix $\boldsymbol{F}$.

$$\mathbf{F} = \begin{pmatrix} - & 10 & 0 & 3 & 0 & 5 & 4 & 5 & 5 & 2 \\ & - & 5 & 4 & 5 & 0 & 0 & 0 & 4 & 4 \\ & & - & 1 & 0 & 3 & 4 & 7 & 4 & 7 \\ & & & - & 4 & 6 & 10 & 3 & 0 & 2 \\ & & & & - & 5 & 3 & 8 & 7 & 2 \\ & & & & & - & 7 & 8 & 3 & 9 \\ & & & & & & - & 9 & 1 & 0 \\ & & & & & & & - & 3 & 3 \\ & & & & & & & & - & 9 \\ & & & & & & & & & - \end{pmatrix}$$

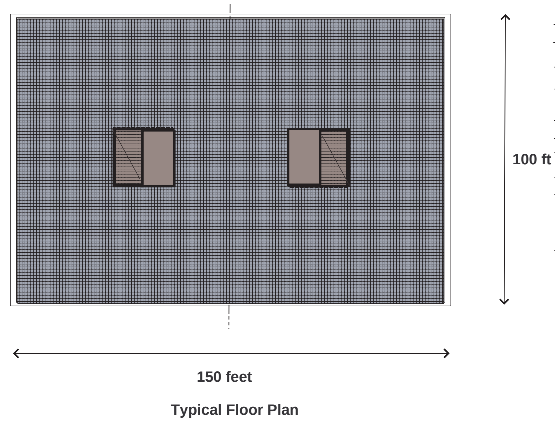

Figure 3: Multi-Story Plan

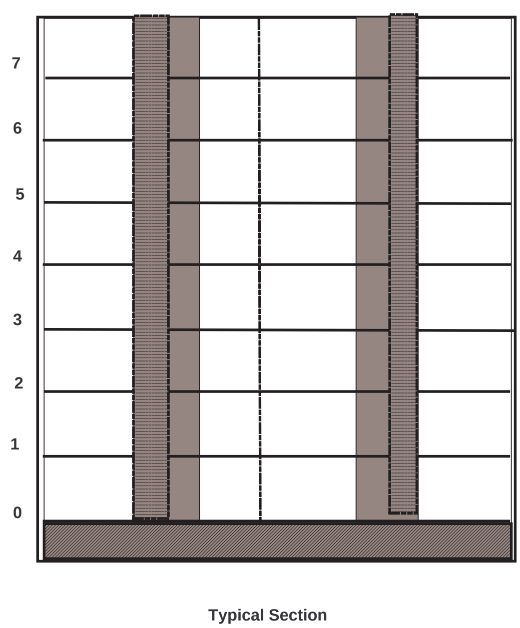

Figure 4: Multi-Story Facility

Flow Matrix

We assume the following area requirements $[a_i]$ for the ten departments.

$$[\mathbf{a_i}] = \mathbf{7.5(2, 1, 1, 2, 2, 1, 1, 2, 1, 1)}$$

Area Reqts. (sq units)

For the purpose of our test examples, we assume the following distances between floors.

$$d_{jn} = |j - n|$$

Distance (flow cost) Between Floors

This choice is just a convenience for testing our exact solution algorithm. In practice, the distance or flow cost can be whatever makes sense for the structure being optimized. In a realistic situation, flow costs between distant points a given floor may be larger than those between locations that are 'one on top of the other' on adjacent floors. As mentioned in §3.3, one can divide each floor into smaller pieces (called compartments) and let the matrix $[d_{jn}]$ represent travel distances between compartments, whether they are on the same floor or not. In this way, the topology/geometry of the location problem of a single floor is taken into account.

We assume the following evacuation flows $[\lambda_i]$ for the ten departments.

$$[\lambda_\mathbf{i}] = \begin{pmatrix} 100 \\ 50 \\ 40 \\ 120 \\ 100 \\ 65 \\ 70 \\ 150 \\ 50 \\ 70 \end{pmatrix}$$

Evac. Flows (people/min)

We assume the following floor-to-exit distances $[e_{js}]$ for the two departments.

$$[\mathbf{e_{js}}] = \begin{pmatrix} 0 & 0 \\ 1 & 1 \\ 2 & 2 \\ 3 & 3 \\ 4 & 5 \\ 5 & 5 \\ 6 & 6 \end{pmatrix}$$

Floor-to-exit dist. (stories)

In practical cases, it would be unusual for the floor-to-exit distances to be the same for different stairwells. We chose to make them identical for debugging purposes. When this is done, $[e_{js}]$ is replaced by $[e_j]$ and the model decomposes into two independent problems involving $[x]$ decisions and $[y]$ decisions. This made it possible for us to check all but one of our GQ3AP results using the tried and proven GQAP code. Making floor-to-exit distances equal, did not change the problem difficulty, nor did it in any way invalidate the test results or lessen their significance. In one experiment we made sure that the floor-to-exit distances from each floor were different for the two stairwells in that building. The results confirm that problem difficulty is essentially the same for the similar experiment in which the floor-to-exit distances are equal.

To complete the constraints for the ten departments, seven story, two stairwell example, we set the floor space on each floor to be 15 units of area, corresponding to 15,000 sq.ft.. We consider two values of stairwell capacity, the first is 'oversize' at 1400 occupants/minute and the second results in 'balanced' stairwell utilization, at 500 occupants per minute.

From our numerical experiments, the values of λ_i play an important role in the final outcomes. λ_i is defined as the Poisson arrival rate of persons in the evacuation from department i. Besides being the restricted parameter for the stairwell capacity constraints, λ_i in the objective function is the measurement of how important escape costs are compared to everyday inter-department travel costs. With larger λ_i values, the MSAP design puts more weight on the evacuation cost, which makes the runtime longer. In the results reported in the following section, we have chosen three different orders of magnitude of the $[\lambda_i]$ vector. The first choice is that the λ_i are all between $40 - 150$, resulting in a hard problem instance. The second choice is that the λ_i are all between $4 - 15$, resulting in a medium-hard problem instance and the third choice is that the λ_i are all between 0.4 and 1.5, resulting in an easy problem instance.

5.2 *Department Subdivision*

As mentioned earlier, our ten and thirteen department problem instances were hampered by the restriction that each department was wholly assigned to a single stairwell. While this situation might prove useful in some circumstances, it is not the only way in which our model might be used. We can artificially or justifiably subdivide departments into pieces that can each be assigned separately to individual stairwells. In order to assure that all the pieces end up on the same floor, it is simply a matter of adjusting the flow matrix to have an artificially large value of flow between pieces of the same department. Of course, if data were available regarding flows between sub-departments, these could then serve a more useful purpose. Two more problem instances were therefore generated from the 10 department, 7 floor, 2 stairwell example. In the first of these, the larger departments were divided into two pieces each. This resulted in 14 entities that have to be assigned to seven floors and to two stairwells. In the second of these, the larger departments were divided into four pieces each and the smaller departments into two pieces each. This resulted in 28 entities that have to be assigned to seven floors and to two stairwells. For the newly generated 14 entity, 7 floor, 2 stairwell problem instance, the flow matrix is given by:

$$\begin{pmatrix}
- & 100 & 5 & 0 & 1 & 1 & 0 & 0 & 3 & 2 & 2 & 1 & 2 & 1 \\
 & - & 5 & 0 & 1 & 0 & 0 & 0 & 2 & 2 & 1 & 1 & 3 & 1 \\
 & & - & 5 & 2 & 2 & 3 & 2 & 0 & 0 & 0 & 0 & 4 & 4 \\
 & & & - & 1 & 0 & 0 & 0 & 3 & 4 & 4 & 3 & 4 & 7 \\
 & & & & - & 100 & 1 & 1 & 3 & 5 & 1 & 1 & 0 & 1 \\
 & & & & & - & 1 & 1 & 3 & 5 & 1 & 0 & 0 & 1 \\
 & & & & & & - & 100 & 3 & 1 & 2 & 2 & 4 & 1 \\
 & & & & & & & - & 2 & 2 & 2 & 2 & 3 & 1 \\
 & & & & & & & & - & 7 & 4 & 4 & 3 & 9 \\
 & & & & & & & & & - & 5 & 4 & 1 & 0 \\
 & & & & & & & & & & - & 100 & 2 & 1 \\
 & & & & & & & & & & & - & 1 & 2 \\
 & & & & & & & & & & & & - & 9 \\
 & & & & & & & & & & & & & -
\end{pmatrix}$$

Sub-department Flow Matrix

We assume the following area requirements $[a_i]$ for the fourteen entities (departments or sub-departments).

$$[\mathbf{a_i}] = \mathbf{7.5(1,1,1,1,1,1,1,1,1,1,1,1,1,1)}$$

Sub-Department Area Reqts. (sq units)

We assume the following evacuation flows $[\lambda_i]$ for the fourteen entities.

$$[\lambda_\mathbf{i}] = \begin{pmatrix} 50 \\ 50 \\ 50 \\ 40 \\ 60 \\ 60 \\ 50 \\ 50 \\ 65 \\ 70 \\ 75 \\ 75 \\ 50 \\ 70 \end{pmatrix}$$

Sub-department Evac. Flows (people/min)

5.3 *Results*

Table 1 presents the branch-and-bound algorithm results for a small series of experiments. The first three experiments in this table refer to the seven story building, ten department, two stairwell examples described above. In the second experiment, the capacities of the stairwells are constrained to balance stairwell utilization. In the third experiment, in addition to balancing stairwell utilization, the large departments are cut in half and the fourteen entities (sub-departments and small departments) are handled as fourteen separate entities. The experiments in Table 1 were performed on a a single 740 MHz cpu of a Sun V880 server, a single 1.6 GHz cpu of a Sun Ultra 45 workstation or a single 770 MHz processor of a Dell 7150 server. Column one of the Table gives the number of entities to be assigned $M \in \{10, 13, 14\}$, the number of floors $N \in \{7, 8\}$ and the number of stairwells $S \in \{2, 3\}$. Column two gives the stairwell capacities ("oversize" or constrained for "balanced" utilization). In the case of the fourth row experiment, the entry in column 2 explains that the third row experiment has been repeated, this time with an e-matrix whose entries entries reflect a situation where escape distances from the each floor are different depending upon which stairwell is used. The important difference in the branch-and-bound search results is that there are far fewer optimum solutions. However, all the performance measures for this experiment (runtime, number of nodes and time to find the first optimum) are similar to those of the experiment of the third row. Column three gives the OF-ratio. This is the average magnitude of the $\lambda_i e_{js}$ term in the objective function to the average magnitude of the $f_{ik} d_{jn}$ term. The meaning of this ratio is discussed in §3.3 of this paper. Column four contains the root bound for the different problem instances. Column five gives the optimum objective function value of each example. The number of optima found in the

$\mathbf{M}, \mathbf{N}, \mathbf{S}$	Stair Cap.	OF-ratio	Root bound	Optimum	# optima	# Nodes	NormTime	Time to opt.
10, 7, 2	oversize	52.3	512.59	2,582	530	1,116,546	2,316	522
10, 7, 2	balanced	52.3	512.59	2,582	435	809,719	2,497	560
14, 7, 2	balanced	23.8	0.00	2,582	9,168	192,519,143	417,651	3,294
14, 7, 2	$e_{js} \neq e_{jj}$	25.8	0.00	2,793.5	294	215,342,567	509,271	3,893
13, 8, 2	oversize	0.7	88.65	380.2	288	10,463,230	22,014	1,445
13, 8, 2	oversize	7.59	130.65	689.5	512	20,284,374	68,543	37,037
13, 8, 2	balanced	7.59	130.67	689.5	1,086	19,140,059	47,640	25,192
13, 8, 3	oversize	0.75	107.45	351.05	19,683	1,062,543	14,066	5,037
13, 8, 3	oversize	7.51	159.00	630.5	48,493	14,107,944	111,699	4,876
13, 8, 3	balanced	7.51	158.42	630.5	14,772	7,568,612	88,218	38,988
13, 8, 3	balanced	75.14	538.79	3,086	11,640	141,585,318	745,079	9,165

Table 1: B&B Exact Algorithm Results

branch- and-bound search is given in column six. Column seven gives the number of nodes (partial assignments) evaluated in the search. Column eight gives the total runtime in seconds, normalized to the speed of the 1.6 GHz Ultra 45 processor, and column nine gives the run time in seconds to find the first optimum solution.

Table 2 presents typical stairwell assignments found for the four different experiments that refer to the seven story building, ten department, two stairwell examples, three of which are listed in Table 1 and thus produced optimal assignments. In the first experiment, wherein no departments were subdivided and stairwells were oversized, all departments were assigned to stairwell No. 1. In this model, the evacuation flows added up to 815, but the stairwell capacities were so large (1400 for each stairwell) that each stairwell could accept the entire evacuation flow. The capacity constraints (4) should have evened the flow, but with such a large capacity, it was as if the constraints did not exist. The second experiment, wherein stairwell capacities were lowered to 500 each, partially balanced the stairwell flows (475 on stairwell No. 1 and 340 on stairwell No. 2). The third experiment, wherein the large departments were divided in twos, did a poorer job of balancing (325 on stairwell No. 1 and 490 on stairwell No. 2). The fourth experiment, wherein large departments were subdivided in twos and small departments divided in fours, was not completed due to lack of time, but produced assignments within 4.2 percent of optimum. Considering the small subdivisions of the 28, 7, 2 model, this fourth experiment is expected to give the most accurate estimates of how stairwell stairwells should be assigned. The balance achieved was 458 on stairwell No. 1 and 357 on stairwell No. 2.

Figure 5 illustrates the performance of the algorithm as a function of time for selected problem instances. The graph depicts the value of feasible solutions as they are encountered over time during the branch-and-bound search process. The vertical axis is the objective function value of each feasible solution encountered, normalized to the optimum objective function value of the four problem instances, where 1.0 represents the optimum value. The horizontal axis is the branch-and-bound runtime, normalized so that the first optimum solution of each instance is reached at time 1.0. The graph demonstrates that the algorithm encounters very good quality feasible solutions early in the branch-and-bound search process. Furthermore, the optimal solution is usually reached early in the branch-and-bound search, thus indicating the potential heuristic value of the B&B algorithm for generating good solutions quickly. While a performance guarantee of such a heuristic is not available, it still represents a viable approach, since search time for a good solution would be small in comparison to the total running time for the algorithm.

6 Open Questions, Future work, Summary and Conclusions

There are a number of open questions and future directions generated by this research. The first questions concerns the development of an heuristic approach over and above the one offered by the use of a cutoff-time period with the optimal seeking algorithm. The other questions concern improvements to the algorithm.

6.1 Heuristic algorithms for the MSAP

Heuristics and metaheuristics have been designed to address hard optimization problems. The quality of the solutions is generally very good and often optimal. These methods are also the best way to provide feasible solutions in large industrial problems with tight constraints. Moreover, such solutions become starting upper bounds for exact methods.

Metaheuristics are characterized by the definition of a priori strategies adapted to the problem structure. Several of these techniques are based on some form of simulation of a natural process, studied within another field of knowledge. The success of GQAP and Q3AP metaheuristics motivates the adaptation of those methods for the MSAP. Our first approach will be to combine the Simulated Annealing (SA) algorithms that were successful on the Q3AP and GQAP.

A common component of metaheuristics for the assignment problems is the "move" from a known so-

Dept.	10,7,2 oversize	10,7,2 balance	14,7,2 balance	28,7,2 balance
1	1	1	1,2	1,1,1,2
2	1	1	1	2,2
3	1	1	1	2,1
4	1	2	2,2	1,1,1,1
5	1	2	2,2	2,2,2,2
6	1	1	1	1,2
7	1	1	1	1,2
8	1	1	2,2	1,1,1,1
9	1	2	1	2,1
10	1	2	2	2,2

Table 2: Stairwell Assignments for four models of 10 Dept., 7 Story, 2 Staircase Problem

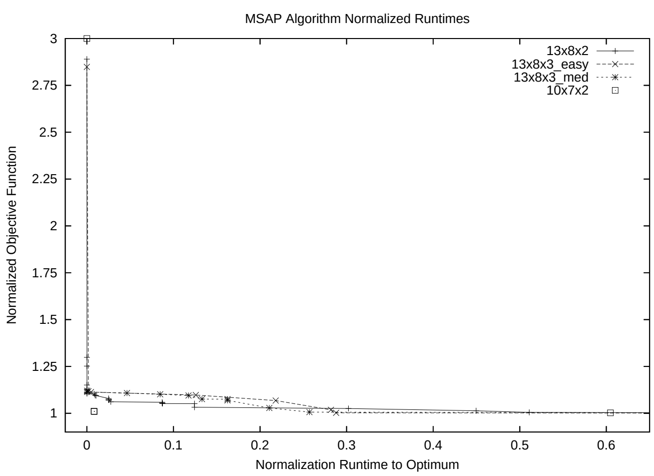

Figure 5: Algorithm solution graph

lution to another nearby solution. As part of the design process, in order to improve the speed of a Q3AP heuristic search, Kim [32] developed a new evaluation procedure for the cost implication of a move. In many ways, the GQ3AP heuristic algorithm Simulated Annealing would not require significant change from its GQAP counterpart. It merely requires that we combine that algorithm with the Q3AP procedure for evaluating the cost of a move.

6.2 *Algorithm Improvements*

It is evident from Table 1 that the growth in runtime with the number of floors in the building is a severe limitation to the usefulness of our exact solution method. Witness that for the most difficult problem instance, the runtime on a Sun Workstation computer is almost four weeks. The GQ3AP is a very difficult problem. Clearly, the quality of lower bounds is a critical factor in determining the runtime. We plan to investigate a number of methods for improving the tightness of the subproblem lower bounds and for improving the quality of the branching strategy. The current branching strategy is based on previously successful strategies developed for the QAP (see [24]). The search space for the GQ3AP is quite different from that of the QAP and thus new branching strategies need to be developed for this more difficult environment. We have learned quite a bit from the experiments directed to balance the flows of evacuees from the building on the various escape routes (i.e., stairwells). This experience teaches us that more work needs to be done to better deal with this important issue. Finally, the existence of large numbers of multiple optima lends a great deal of ambiguity to the results. There is a need to develop a post-optimization process that would sift through the various optimum solutions and bring forward for consideration only those solutions that merit further consideration.

6.3 *Summary and Conclusions*

This paper has presented an innovative formulation of the Multi-Story Assignment Problem (MSAP), that not only assigns unequal area departments to the different floors of the facility, but also accounts for the evacuation of the occupants. Besides the detailed mathematical development of the formulation and its ancestral evolution from quadratic assignment problems (QAP), a detailed branch-and-bound procedure based upon an RLT1 dual ascent procedure has been presented. The effective implementation of the algorithm was also demonstrated on a number of test problems. While the run times of the algorithm are exponential in the problem size, the effectiveness of the algorithm for achieving good solutions quickly was also shown. All in all, the paper describes a new type of quadratic assignment problem and provides an exact solution method and experimental results for this new problem type.

Acknowledgements

We wish to thank Bum-Jin Kim, Ph.D., Catholic University, Washington, D.C. for his assistance in testing and debugging our exact solution algorithm for the Multi-story Space Assignment Problem. This material is based upon work supported by the U.S. National Science Foundation under grant No. DMI-0400155.

References

[1] Adams, W.P., Guignard, M., Hahn, P.M., Hightower, W.L. (2007). A Level-2 Reformulation-Linearization technique bound for the quadratic assignment problem, *European Journal of Operational Research*, **180(3)**, 983-996.

[2] Adams, W.P. and Johnson, T.A. (1994). Improved linear programming-based lower bounds for the quadratic assignment problem. In: *Quadratic Assignment and Related Problems* [edited by P.M. Pardalos and H. Wolkowicz], *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, **16**, 43-75, AMS, Rhode Island.

[3] Adams, W.P., and Sherali H.D. (1986). A tight linearization and an algorithm for zero-one quadratic programming problems, *Management Science* **32**, 1274-1290.

[4] Adams, W.P., and Sherali H.D. (1990). Linearization strategies for a class of zero-one mixed integer programming problems, *Operations Research* **38**, 217-226.

[5] Benjaafar, S. (2002). Modeling and analysis of congestion in the design of facility layouts. *Management Science*, **48(5)**, 679-704.

[6] Billionnet, A. and Elloumi, S. (1995). An Algorithm for Finding the K-Best Allocations of a Tree Structured Program, *Journal of Parallel and Distributed Computing*, **26**: 225-232.

[7] Billionnet, A. and Elloumi, S. (2001) Best Reduction of the Quadratic Semi-Assignment Problem, *Discrete Applied Mathematics (DAM)*, **109(3)** 197-213.

[8] Bland, J.A. and Dawson, G.P. (1994). Large-scale layout of facilities using a heuristic hybrid algorithm. *Applied Mathematical Modeling*, **18(9)**, 500-503.

[9] Burkard, R. E. (1983). A Heuristic for Quadratic Boolean Programs with Applications to Quadratic Assignment Problems, *European Journal of Operational Research*, **13**, 374-386.

[10] Burke, E.K., Cowling, P., Landa, J.D. and McCollum, B. (2000). Three Methods to Automate the Space Allocation Process in UK Universities, Practice, and Theory of Automated Timetabling III, PATAT 2000, The Third International Conference on the Practice and Theory of Automated Timetabling, Konstanz, Germany, 16-18 August 2000, 374-393. Selected for post conference publication in Springer Lecture Notes in Computer Science, **2079**, ISBN 3-540-42421-0, 254-273.

[11] Burke, E. K., Cowling, P. and Keuthen, R. (2001). Effective local and guided variable neighbourhood search methods for the asymmetric travelling salesman problem. In Applications of Evolutionary Computing, EvoWorkshops 2001, **2037** of Lecture Notes in Computer Science, pages 203-212. Springer Verlag, Berlin, Germany.

[12] Ciriani, V., Pisanti, N. and Bernasconi, A. (2004). Room allocation: a polynomial subcase of the quadratic assignment problem. *Discrete Applied Mathematics*, **144(3)**, 263-269.

[13] Connolly, D.T. (1990), An improved annealing scheme for the QAP, *European Journal of Operational Research*, **46(1)**, 93-100.

[14] Cordeau, J-F., Gaudioso, M., Laporte, G., and Moccia, L. (2006), A Memetic Heuristic for the Generalized Quadratic Assignment Problem, *INFORMS Journal on Computing*, **18(4)**, 433-443.

[15] Dickey, J.W., Hopkins, J.W. (1972). Campus building arrangement using Topaz. *Transportation Research* **6**, 59-68.

[16] Drezner, Z., Hahn, P.M. and Taillard, E. (2005). Recent advances for the quadratic assignment problem with special emphasis on instances that are difficult for meta-heuristic methods, *Annals of Operations Research*, **139**, 65-94.

[17] Elloumi, S., Roupin, F. and Soutif, E. (2003). Comparison of Different Lower Bounds for the Constrained Module Allocation Problem, - TR CEDRIC No 473.

[18] Elshafei, A.N. (1977). Hospital layout as a quadratic assignment problem. *Operations Research Quarterly*, **28(1)**, 167-179.

[19] Felkel. P. and Obdrzalek, S. (1998). Straight skeleton implementation. *Spring Conference on Computer Graphics - SCCG'98*, Budmerice, Slovakia, April 23-25, 210–218.

[20] Garey, M.R. and Johnson, D.S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman, San Francisco.

[21] Haghani, A. and Chen, M.C. (1998). Optimizing gate assignments at airport terminals. *Transportation Research Part A: Policy and Practice*, **32(6)**, 437-454.

[22] Hahn, P.M. and Krarup, J. (2001). A hospital facility layout problem finally solved. *Journal of Intelligent Manufacturing*, **12**, 487-496.

[23] Hahn, P.M. and Grant T. (1998). Lower Bounds for the Quadratic Assignment Problem Based on a Dual Formulation, *Operations Research*, **46**, 912-922.

[24] Hahn, P.M., Hightower, W.L., Johnson, T.A., Guignard-Spielberg, M. and Roucairol, C. (2001). Tree elaboration strategies in branch and bound algorithms for solving the quadratic assignment problem, *Yugoslavian Journal of Operational Research*, **11(1)**, 41-60.

[25] Hahn, P.M., Kim,B-J., Hightower, W. L., Stützle, T., Kanthak, S., Samra, H., Ding, Z. and Guignard, M. (2008a), The quadratic three-dimensional assignment problem: exact and heuristic solution methods, *European Journal of Operational Research*, **184(2)**, 416-428.

[26] Hahn, P.M., Kim, B-J., Guignard, M., Smith, J.M., and Zhu, Y-R. (2008b). An Algorithm for the Generalized Quadratic Assignment Problem, to appear in *Computational Optimization and Applications*.

[27] Heffley, D.R. (1980). Decomposition of the Koopmans-Beckmann problem. *Regional Science and Urban Economics*, **10(4)**, 571-580.

[28] Ho, Y.C. and Moodie, C.L. (2000). A hybrid approach for concurrent layout design of cells and their flow paths in a tree configuration. *International Journal of Production Research*, **38(4)**, 895-928.

[29] Hoos, H. H. and Stützle, T. (2004). *Stochastic Local Search: Foundations and Applications*, Morgan Kaufmann Publishers, San Francisco, CA, USA. .

[30] Khare, V.K., Khare, M.K. and Neema, M.L. (1988a). Estimation of distribution parameters associated with facilities design problems involving forward and backtracking of materials. *Computers and Industrial Engineering*, **14(1)**, 63-75.

[31] Khare, V.K., Khare, M.K. and Neema, M.L. (1988b). Combined computer-aided approach for the facilities design problem and estimation of the distribution parameter in the case of multigoal optimization. *Computers and Industrial Engineering*, **14(4)**, 465-476.

[32] Kim, B-J. (2006), Investigation of Methods for Solving New Classes of Quadratic Assignment Problems (QAPs), Dissertation in Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA.

[33] Kochhar, J.S., Foster, B.T. and Heragu, S.S. (1998). Hope: A genetic algorithm for the unequal area facility layout problem. *Computers and Operations Research*, **25(7-8)**, 583-594.

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

[35] Kouvelis, P. and Kiran A. S., (1990). The plant layout problem in automated manufacturing systems, *Annals of Operations Research*, **26**, 397-412.

[36] Lacksonen, T.A. and Enscore, Jr. E.E. (1993). Quadratic assignment algorithms for the dynamic layout. *International Journal of Production Research*, **31(3)**, 503-517.

[37] Lee, C-G, and Ma, Z. (2004). The Generalized Quadratic Assignment Problem, *Research Report, Department of Mechanical and Industrial Engineering, University of Toronto*, Toronto, Ontario, M5S 3G8, Canada.

[38] Li, Wu-ji and MacGregor Smith, J. (1995). An Algorithm for Quadratic Assignment Problems, *European Journal of Operations Research*, **81** 205-216.

[39] Li, Wu-ji and MacGregor Smith, J. (1998). Evaluation of Star, Grid, and Ring Topologies in Facility Layout and Network Design, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, **40**, 219-246.

[40] Li, Wu-ji and MacGregor Smith, J. (1994). Stochastic Quadratic Assignment Problems, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, **16**, 221-236.

[41] Li, Y., Pardalos, P., Resende, M. (1994), A Greedy Randomized Adaptive Search Procedure for the Quadratic Assignment Problem, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, **16**, 237-261.

[42] Loiola, E., Maia de Abreu, N.M., Boaventura-Netto, P.O., Hahn, P.M. and Querido, T. (2007). An Analytical Survey for the Quadratic Assignment Problem, *European Journal of Operational Research*, **176**, 657-690.

[43] Magirou, V. F. and Milis, J. Z. (1989), An Algorithm for the Multiprocessor Assignment Problem, *Operations Research Letters,* **8**, 351-356.

[44] Magirou, V. F. (1992). An Improved Partial Solution to the Task Assignment and Multiway Cut Problems, *Operations Research Letters,* **12**, 3-10.

[45] Martin, W. (1998). Fast equi-partitioning of rectangular domains using stripe decomposition. *Discrete Applied Mathematics*, **82(1-3)**, 193-207.

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

[47] Meller, R.D. and Gau, K. Y. (1996). The facility layout problem: Recent and emerging trends and perspectives. *J. Manufacturing Systems*, **15**, 351-366.

[48] Meller, R.D. and Bozer, Y.A. (1997) Alternative approaches to solve the multi-floor facility layout problem, *Journal of Manufacturing Systems,* **16(3)**, ABI/INFORM Global, 192.

[49] Oliva, J.M., Perrin, M., and Coquillart,S. (1996). 3D Reconstruction of Complex Polyhedral Shapes from Contours using a Simplied Generalized Voronoi Diagram, *Computer Graphics Forum,* **15(3)**, C-397-C-408.

[50] Pardalos, P.M., Rendl, F. and Wolkowicz, H. (1994). The Quadratic Assignment Problem: A Survey and Recent Developments, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, **16**

[51] Pierskalla, W.P. (1967). The Multi-Dimensional Assignment Problem, *Technical Memorandum No. 93*, Operations Research Department, CASE Institute of Technology.

[52] Rossin, D.F., Springer, M.C. and Klein, B.D. (1999). New complexity measures for the facility layout problem: an empirical study using traditional and neural network analysis. *Computers and Industrial Engineering*, **36(3)**, 585-602.

[53] Roupin, F. (2003). From Linear to Semi-definite Programming: an Algorithm to obtain Semidefinite Relaxations for Bivalent Quadratic Problems - TR CEDRIC No 388, 2003.

[54] Sahni, S. and Gonzalez,T. (1976). P-Complete Approximation Problems, *Journal Assoc. Computing Machinery*, **23(3)**, 555-565.

[55] Sherali, H.D. and Adams, W.P. (1999). *A reformulation-linearization technique for solving discrete and continuous non-convex problems*, 1st edition, Kluwer Academic Publishers, Norwell, MA 02061.

[56] Sofianopoulous, S. (1992). The Process Allocation Problem: A Survey of the Application of Graph-Theoretic and Integer Programming Approaches, *Journal of the Operational Research Society*, **43(5)**, 407-413.

[57] Stützle, T. (2006). Iterated local search for the quadratic assignment problem. *European Journal of Operational Research*, **174(3)**, 1519-1539.

[58] Taillard, E. (1991). Robust taboo search for the quadratic assignment problem, *Parallel Computing*, **17**, 443-455.

[59] Taillard, E. D. (1998). FANT: Fast ant system, Technical report IDSIA-46-98, IDSIA, Lugano, 1998.

[60] Tavakkoli-Moghaddain, R. and Shayan, E. (1998). Facilities layout design by genetic algorithms. *Computers and Industrial Engineering*, **35(3-4)**, 527-530.

[61] Tompkins, J. A., White, J. A., Bozer, Y. A., Frazelle, E. H., Tanchoco, J. M. A. and Trevino, J. (1996). *Facilities Planning*, 2nd ed. John Wiley, New York.

[62] Weng, Y. and MacGregor Smith, J. (2006). On the Equi-area Partitioning Problem for Rectilinear Simple Polygons, Department of Mechanical and Industrial Engineering, University of Massachusetts, Amherst Campus *in review*.

[63] Yang, T. and Benjaafar, S. (2001). FLQ: A software for facility layout with queueing. Technical report 0015, Division of Industrial Engineering, Department of Mechanical Engineering, University of Minnesota, Minneapolis, MN.

[64] Zhu, Y-R. (2007). Recent Advances and Challenges in the Quadratic Assignment and Related Problems, *A Dissertation in Electrical and Systems Engineering. University of Pennsylvania*, August. Available on the Internet at `http://www.seas.upenn.edu/~hahn/Dissertation_yrz.pdf`.