Novel and Effective Integer Optimization Approach for the NSF Panel-Assignment Problem: A Multiresource and Preference-Constrained Generalized Assignment Problem

Stacy L. Janak, Martin S. Taylor, and Christodoulos A. Floudas*

Department of Chemical Engineering, Princeton University, Princeton, New Jersey 08544-5263

Maria Burka and T. J. Mountziaris

Division of Chemical and Transport Systems, National Science Foundation, 4201 Wilson Boulevard, Arlington, Virginia 22230

The NSF panel-assignment problem studied in this work involves selecting an assignment of three or four reviewers to each proposal in a panel so as to optimize the sum of a set of preference criteria for each reviewer on each proposal while ensuring that each reviewer is assigned to approximately the same number of proposals. In addition, each proposal has three or four distinct positions that are assigned to reviewers based upon the preference criteria so that each reviewer holds each position approximately the same number of times. This multiresource and preference-constrained generalized assignment problem can be formulated as an integer linear programming problem and can be solved to optimality. In this work, a mathematical model is developed to address the NSF panel-assignment problem, and some representative example problems are solved to demonstrate the effectiveness of the proposed approach.

1. Introduction

The NSF panel-assignment problem posed in this work can be viewed as an enhanced version of the generalized assignment problem (GAP), which has been the subject of considerable research over the last twenty years. The GAP has many real-life applications including job scheduling, production planning, modeling of computer and communication networks, storage space allocation, vehicle routing, and facility location problems. The GAP seeks to determine the minimum cost assignment of *n* jobs to *m* agents so that each job (*j*) is assigned to exactly one agent (*i*), subject to resource restrictions on the agents. The GAP can be formulated as follows,

$$\begin{aligned} &\underset{x}{\text{Min}} \sum_{i\in I}\sum_{j\in J_i} c_{i,j}x_{i,j} \\ &\text{s.t.} \sum_{j\in J_i} a_{i,j}x_{i,j} \le b_i \quad \forall i \in I \\ &\sum_{i\in I_j} x_{i,j} = 1 \quad \forall j \in J \\ &x_{i,j} = \{0, 1\} \quad \forall i \in I, j \in J_i \end{aligned} \tag{1}$$

where $c_{i,j}$ is the cost of assigning job (*j*) to agent (*i*), $a_{i,j}$ is the amount of resource consumed by job (*j*) when assigned to agent (*i*), and b_i is the resource availability of agent (*i*). The binary assignment variable $x_{i,j}$ equals 1 if agent (*i*) is to perform job (*j*) and equals 0 otherwise.

Because the GAP is an NP-hard problem,[1−3] heuristic solutions are often necessary to solve large-scale problems. Most algorithms are based on branch-and-bound techniques and the relaxation of constraints, although other techniques have been employed. For an extended review of methods used to solve the GAP, the reader is referred to Cattrysse and Van Wassenhove[4] and Osman.[5]

Several exact solution algorithms for the GAP have been proposed that are able to solve to optimality problems that contain up to ∼200 tasks. These methods differ based on the way the bounds are computed in the branch-and-bound scheme, and different relaxations can be employed including linear relaxations, deletion of constraints, Lagrangian relaxations, surrogate relaxations, and Lagrangian decomposition. Ross and Soland[6] developed a depth-first branch-and-bound algorithm to solve the GAP. Fisher et al.[3] proposed a Lagrangian relaxation incorporated within a branch-and-bound formulation. This approach was extended by Guignard and Rosenwein[7] to utilize a Lagrangian dual-ascent procedure and exploit violations of the relaxed constraints. Savelsbergh[8] developed a branch-and-price scheme to solve a set partitioning reformulation of the GAP that incorporates column generation. Nauss[9] proposed a branch-and-bound algorithm that employs linear programming cuts, feasible-solution generators, Lagrangian relaxation, and subgradient optimization.

Various heuristic methods have also been proposed for the GAP. Martello and Toth[10,11] developed a greedy heuristic that incorporates local search techniques. Trick[12] used an LP relaxation to fix variables and then reassigns them using a swap neighborhood. Amini and Racer[13,14] developed a variable-depth search heuristic (VDSH) that is based on two-phase local search descent methods. Osman[5] developed hybrid simulated annealing and tabu search methods using various local search descent techniques. Cattrysse et al.[15] proposed a set partitioning heuristic that utilizes column generation and subgradient optimization that was later extended by Cattrysse et al.[16] to include lift-cover inequalities to improve the lower bound within a branch-and-bound procedure. Lorena and Narciso[17] used Lagrangian or surrogate relaxation within a subgradient search procedure and incorporated a variety of heuristics. Chu and Beasley[18] proposed a hybrid genetic algorithm heuristic which was extended by Feltl and Raidl[19] to include initialization heuristics and a modified handling of infeasible solutions. Yagiura et al.[20] developed an improved VDSH, which is a generalization of a local search and considers several branching rules.Yagiura et al.[21] proposed a tabu search algorithm that utilizes ejection chains.

* To whom all correspondence should be addressed. Tel.: (609) 258-4595. Fax: (609) 258-0211. E-mail: floudas@titan.princeton.edu.

10.1021/ie050478h CCC: $33.50 © 2006 American Chemical Society
Published on Web 11/12/2005

Several authors also considered variations of the GAP. Laguna et al.[22] considered the multilevel GAP where agents perform tasks at more than one efficiency level. They proposed a tabu search heuristic that employs ejection chains. French and Wilson[23] also considered the multilevel GAP and developed two different heuristics. Park et al.[24] studied the generalized multiassignment problem (GMAP) which allows multiple assignments for each job. They proposed a dual-based branch-and-bound algorithm that alternates between dual ascent and subgradient search. Gavish and Pirkul[25] considered the multi-resource GAP (MRGAP) in which each agent has a number of different potentially constraining resources. They analyzed several relaxations along with heuristics in a branch-and-bound procedure including subgradient optimization. Mazzola and Wilcox[26] also considered the MRGAP and developed heuristics which search for an initial feasible solution and then apply procedures to improve it. Toktas et al.[27] focused on the uncertainty in a variety of resource-constrained assignment problems where job-to-agent matching can be one-to-one or one-to-many and resources can be individually or collectively capacitated. They developed two deterministic solution strategies to address uncertainty in resource-constrained problems for which stochastic implementations are intricate or undesirable.

The NSF panel-assignment problem studied in this work is an enhanced version of the MRGAP. Each job has a specific number of agents assigned to it and each agent has a lower and upper bound on the number of jobs that can be assigned to it. In addition, preference criteria are incorporated to assign each agent to a specific position for a job, and this introduces preference-based constraints. The rest of the paper is organized as follows. Section 2 defines the problem under consideration. Section 3 develops the mathematical model for the NSF panel-assignment problem. In Section 4, several computational examples are presented to demonstrate the effectiveness of the formulation. Section 5 describes an online solver for the panel-assignment problem, and Section 6 provides conclusions.

2. Problem Description

The panel-assignment problem can be defined as follows. Given (i) the number of available reviewers, (ii) the number of proposals in the panel, (iii) the number of reviews needed for each proposal, and (iv) a matrix of preferences for each reviewer on each proposal, the objective is to determine an assignment of reviewers to proposals on the panel so as to optimize the sum of the preferences for the reviewers to their assigned proposals. In addition, there are several requirements that must be met as follows:

(i) each reviewer should be assigned to approximately the same number of proposals,

(ii) each proposal should be assigned to the same number of reviewers,

(iii) reviewers that have a "conflict of interest" (COI) with a proposal must not be assigned to that proposal,

(iv) each proposal should have three or four distinct positions (LEAD, SCRIBE, REV1, REV2) and each position must be filled exactly once for each proposal and each reviewer should be assigned to each position approximately the same number of times, and

(v) the assignment of reviewers to positions should follow the priority $w_{i,j}^{\mathrm{LEAD}} \le w_{i,j}^{\mathrm{SCRIBE}} \le w_{i,j}^{\mathrm{REV1}} \le w_{i,j}^{\mathrm{REV2}}$ according to the preferences of the reviewers (j) for the proposals (i). Note that the smaller the value of $w_{i,j} > 0$, the higher the preference of a reviewer for a proposal. Typical values for $w_{i,j}$ are integers ranging from 1 for the highest preference to N for the lowest preference, where N is the total number of proposals in the panel.

3. Mathematical Model

The proposed formulation requires the following indices, sets, parameters, and variables.

Indices and Sets:

$i \in I$ proposals in the panel;

$j \in J$ reviewers in the panel;

$j \in J_i$ reviewers (j) which can be assigned to proposal (i) (i.e., preference $w_{i,j} > 0$);

$i \in I_j$ proposals (i) which can be assigned to reviewer (j) (i.e., preference $w_{i,j} > 0$).

Parameters:

N, total number of proposals (i) (i.e., cardinality of the set I);

M, total number of reviewers (j) (i.e., cardinality of the set J);

K, number of reviews needed for each proposal (i) (e.g., 3 or 4);

$w_{i,j}$, preference of each reviewer (j) for each proposal (i);

ϵ, a small positive constant (e.g., 0.001).

Variables:

$y(i, j)$ binary, whether or not reviewer (j) is assigned to proposal (i);

$L(i, j)$ binary, whether or not reviewer (j) is the LEAD for proposal (i);

$S(i, j)$ binary, whether or not reviewer (j) is the SCRIBE for proposal (i);

$R1(i, j)$ binary, whether or not reviewer (j) is the REV1 for proposal (i);

$R2(i, j)$ binary, whether or not reviewer (j) is the REV2 for proposal (i);

$x(i, j, jj)$ binary, whether or not reviewer (j) has an equal or higher preference for proposal (i) as reviewer (jj) (i.e., $w_{i,j} \le w_{i,jj}$);

$sl(i, j, jj)$ continuous, slack variable to relax the condition that reviewer (j) has as equal or higher preference for proposal (i) as reviewer (jj).

On the basis of this notation, the mathematical model for the panel-assignment problem involves the following constraints.

Resource Constraints for Each Reviewer.

$$\left\lfloor \frac{NK}{M} \right\rfloor \le \sum_{i \in I_j} y(i, j) \le \left\lceil \frac{NK}{M} \right\rceil, \forall j \in J \tag{2}$$

These constraints express the requirement that each reviewer (j) must be assigned to at least $\lfloor NK/M \rfloor$ proposals (i) and at most $\lceil NK/M \rceil$ proposals (i), where $\lfloor\ \rfloor$ indicates the floor function and $\lceil\ \rceil$ indicates the ceiling function. This ensures that each reviewer is assigned to approximately the same number of proposals.

$$\left\lfloor \frac{N}{M} \right\rfloor \le \sum_{i \in I_j} L(i, j) \le \left\lceil \frac{N}{M} \right\rceil, \quad \forall j \in J \tag{3}$$

$$\left\lfloor \frac{N}{M} \right\rfloor \le \sum_{i \in I_j} S(i, j) \le \left\lceil \frac{N}{M} \right\rceil, \quad \forall j \in J \tag{4}$$

$$\left\lfloor \frac{N}{M} \right\rfloor \le \sum_{i \in I_j} R1(i, j) \le \left\lceil \frac{N}{M} \right\rceil, \quad \forall j \in J \tag{5}$$

$$\left\lfloor \frac{N}{M} \right\rfloor \le \sum_{i \in I_j} R2(i, j) \le \left\lceil \frac{N}{M} \right\rceil, \quad \forall j \in J, K = 4 \tag{6}$$

Constraints 3−6 enforce that each reviewer (j) must be assigned to each position (i.e., LEAD, SCRIBE, REV1, and REV2) at least $\lfloor N/M \rfloor$ times and at most $\lceil N/M \rceil$ times. This ensures that each reviewer holds each position approximately the same number of times.

Resource Constraints for Each Proposal.

$$\sum_{j \in J_i} y(i, j) = K, \quad \forall i \in I \tag{7}$$

These constraints express the requirement that each proposal (i) must be reviewed by exactly K reviewers, or $y(i, j) = 1$ exactly K times.

$$\sum_{j \in J_i} L(i, j) = 1, \quad \forall i \in I \tag{8}$$

$$\sum_{j \in J_i} S(i, j) = 1, \quad \forall i \in I \tag{9}$$

$$\sum_{j \in J_i} R1(i, j) = 1, \quad \forall i \in I \tag{10}$$

$$\sum_{j \in J_i} R2(i, j) = 1, \quad \forall i \in I, K = 4 \tag{11}$$

Constraints 8−11 enforce that, for each proposal (i), all of the positions must be filled exactly once. Thus, one reviewer is assigned to be the LEAD (i.e., $L(i, j) = 1$), one reviewer is assigned to be the SCRIBE (i.e., $S(i, j) = 1$), one reviewer is assigned to be REV1 (i.e., $R1(i, j) = 1$), and if $K = 4$, then one reviewer is assigned to be REV2 (i.e., $R2(i, j) = 1$).

Assignment Constraints for Reviewers to Positions.

$$L(i, j) + S(i, j) + R1(i, j) + R2(i, j) = y(i, j), \quad \forall i \in I, j \in J_i \tag{12}$$

Constraints 12 represent the requirement that, if a reviewer (j) is assigned to a proposal (i) (i.e., $y(i, j) = 1$), then that reviewer must be assigned to exactly one of the four positions.

$$L(i, j) \le y(i, j), \quad \forall i \in I, j \in J_i \tag{13}$$

$$S(i, j) \le y(i, j), \quad \forall i \in I, j \in J_i \tag{14}$$

$$R1(i, j) \le y(i, j), \quad \forall i \in I, j \in J_i \tag{15}$$

$$R2(i, j) \le y(i, j), \quad \forall i \in I, j \in J_i, K = 4 \tag{16}$$

Constraints 13−16 enforce that a reviewer (j) can only be assigned to a specific position on a proposal (i) if the reviewer is actually assigned to the proposal (i.e., $y(i, j) = 1$).

Preference Constraints for Reviewers to Proposals. The following preference constraints developed to assign reviewers to proposals using the preference criteria matrix, $w_{i,j}$, are based on logic inference principles.[28]

$$L(i, j) + S(i, jj) - x(i, j, jj) \le 1, \quad \forall i \in I; j, jj \in J_i; j \ne jj \tag{17}$$

These constraints represent the requirement that, if reviewer (j) has a lower preference for proposal (i) than reviewer (jj) (i.e., $x(i, j, jj) = 0$), then if reviewer (j) is assigned to the proposal as LEAD, then reviewer (jj) cannot be assigned to the proposal as SCRIBE. Similarly, if reviewer (jj) is assigned to the proposal as SCRIBE, then reviewer (j) cannot assigned to the proposal as LEAD.

$$S(i, j) + R1(i, jj) - x(i, j, jj) \le 1, \quad \forall i \in I; j, jj \in J_i; j \ne jj \tag{18}$$

Constraints 18 represent the requirement that, if reviewer (j) has a lower preference for proposal (i) than reviewer (jj) (i.e., $x(i, j, jj) = 0$), then if reviewer (j) is assigned to the proposal as SCRIBE, then reviewer (jj) cannot be assigned to the proposal as REV1. Similarly, if reviewer (jj) is assigned to the proposal as REV1, then reviewer (j) cannot be assigned to the proposal as SCRIBE.

$$R1(i, j) + R2(i, jj) - x(i, j, jj) \le 1, \quad \forall i \in I; j, jj \in J_i; j \ne jj; K = 4 \tag{19}$$

Similar to constraints 17 and 18, constraints 19 represent the requirement that, if reviewer (j) has a lower preference for proposal (i) than reviewer (jj) (i.e., $x(i, j, jj) = 0$), then if reviewer (j) is assigned to the proposal as REV1, then reviewer (jj) cannot be assigned to the proposal as REV2. Or, if reviewer (jj) is assigned to the proposal as REV1, then reviewer (j) cannot be assigned to the proposal as SCRIBE.

$$-N(1 - x(i, j, jj)) \le w_{i,jj} - w_{i,j} + sl\,(i, j, jj), \quad \forall i \in I; j, jj \in J_i; j \ne jj \tag{20}$$

$$w_{i,jj} - w_{i,j} + sl(i, j, jj) \le Nx(i, j, jj) - \epsilon, \quad \forall i \in I; j, jj \in J_i; j \ne jj \tag{21}$$

Constraints 20 and 21 relate the value of $x(i, j, jj)$ to the relative values of the preference criteria $w_{i,j}$ and $w_{i,jj}$ so that, if $w_{i,j} \le w_{i,jj}$, then $x(i, j, jj) = 1$. Similarly, if $w_{i,j} > w_{i,jj}$, then $x(i, j, jj) = 0$. However, if no feasible solution is available where the preference constraints for each reviewer are maintained for each proposal, then the slack variable, $sl(i, j, jj)$ is activated allowing $w_{i,j} > w_{i,jj}$ and $x(i, j, jj) = 1$ where $sl(i, j, jj) = |w_{i,jj} - w_{i,j}|$. This slack variable is then penalized in the objective to avoid finding solutions which violate the preference constraints.

Objective Function.

$$\text{Min} \sum_{i \in I} \sum_{j \in J_i} w_{i,j} y(i, j) + \alpha \cdot \sum_{i \in I} \sum_{j \in J_i} \sum_{\substack{jj \in J_i 0 \\ j \ne jj}} sl(i, j, jj) \tag{22}$$

The objective function represents the minimization of the weighted sum of assignments of reviewers to proposals plus a term to minimize the infeasibilities introduced through the assignment of reviewers to positions. Thus, we want to select an assignment of reviewers (j) to proposals (i) so that the sum of all the preference criteria, $w_{i,j}$, for the chosen assignments (i.e., $y(i, j) = 1$) is minimized and the preference constraints for reviewers to proposals are violated only if necessary. In this case, the second term in the objective ensures that the preference constraints are violated by the smallest amount possible. Note that α balances the relative weight of both terms in the objective and is chosen to be large enough so that violations of the preference constraints are avoided if possible. Thus, the value of α is dependent on the relative value of the first term in the objective function. α should be chosen so that the second or penalty term in the objective, if it is active, will be at least one order of magnitude larger than the first term. The proposed mathematical model contains constraints 2 − 22 and is an integer linear optimization problem which can be solved to optimality using a linear solver. Also, a rank-ordered list of optimal solutions can be generated by introducing integer cuts and resolving the model.

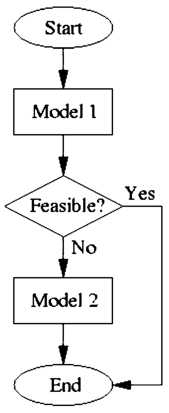


Figure 1. Flowchart of mathematical framework.

3.1. Remarks. 3.1.1. Mathematical Framework for the Panel-Assignment Problem. The panel-assignment problem can be posed using two different models, denoted as Model 1 and Model 2, which differ depending on the inclusion of the slack variables, $sl(i, j, jj)$, for the assignment of reviewers to positions. Note that constraints 20−22 all contain the positive slack variables, $sl(i, j, jj)$, which allow for the introduction of violations in the assignment of reviewers to specific positions (i.e., LEAD, SCRIBE, REV1, or REV2) for a proposal (i). Because the addition of the slack variables increases the size of the mathematical model, the problem is first solved without the slack variables, and then, if necessary, the slack variables are introduced. Specifically, Model 1 is formulated so that all the slack variables are fixed to zero in constraints 20−22, enforcing that no violations are allowed in the optimal solution. Thus, the objective function only involves one term to minimize the weighted sum of the assignment of reviewers to proposals. Model 2 is then formulated using the above equations and the objective function as defined, including the positive slack variables. Note that Model 1 and Model 2 contain the same number of constraints, however, Model 2 contains an additional set of continuous variables, making the problem a mixed-integer linear programming one instead of an integer programming problem. Thus, as outlined in Figure 1, Model 1 is solved first and, if it is feasible, we do not consider the more complicated Model 2. However, if Model 1 is infeasible, then Model 2 is solved to find a solution that allows an assignment of reviewers to positions which violates the preference constraints for reviewers to proposals based on the preference criteria matrix, $w_{i,j}$.

3.1.2. Preference Criteria Matrix, $w_{i,j}$. The preference criteria matrix, $w_{i,j}$, gives the preference of each reviewer (j) for each proposal (i) where the smaller the value of $w_{i,j} > 0$, the larger is the preference for the proposal. If there is a conflict of interest (COI) for a reviewer on a specific proposal, then $w_{i,j}$ should be set to zero. In turn, the condition that $w_{i,j} = 0$ ensures that the sets I_j and J_i are empty for that combination of reviewer (j) and proposal (i); thus, $y(i, j)$ never appears in any of the constraints in the model and, consequently, takes on a value of zero. Similarly, we can write the following equivalent constraints to enforce this condition.

$$y(i, j) = 0, \quad \forall i \in I, j \notin J_i \tag{23}$$

In addition, for the cases when a reviewer (j) does not give a preference for a proposal (i), or $w_{i,j}$ is left blank, we assign $w_{i,j}$ a large value or the worst-case value of N, the total number of proposals. Also, note that each proposal (i) must have at least K nonzero entries in the preference criteria matrix, $w_{i,j}$ (i.e., at least K reviewers who do not have a conflict of interest), where K is the number of reviews needed per proposal. Otherwise, there are not enough available reviewers for proposal (i) and the problem will not have a feasible solution.

3.1.3. Number of Reviews Per Proposal, K. The proposed mathematical model is able to solve the panel-assignment problem to optimality and has been tested for the cases when there are either three or four reviews required for each proposal. Note that the model can be extended to the general case in which the number of reviews is proposal specific, or K_i. However, the current formulation only addresses the cases when $K = 3$ or $K = 4$. For the case when $K = 3$, we include the following constraint in the model.

$$R2(i, j) = 0, \quad \forall i \in I, j \in J \tag{24}$$

This constraint ensures that no reviewers (j) are assigned to the position REV2. Similarly, constraints 6, 11, 16, and 19 are not included in the mathematical model when $K = 3$.

3.1.4. Removal of $y(i, j)$ Variables. Note that, since the assignment constraints for reviewers to positions, or constraint 12, correspond to the definitions of the $y(i, j)$ variables, we can substitute this expression into the appropriate constraints and, hence, eliminate the $y(i, j)$ variables along with some sets of constraints. This results in a more compact model with fewer binary variables and constraints. Thus, constraints 2, 7, and 22 can be redefined without the $y(i, j)$ variables as follows.

$$\left\lfloor \frac{NK}{M} \right\rfloor \le \sum_{i \in I_j} (L(i, j) + S(i, j) + R1(i, j) + R2(i, j)) \le \left\lceil \frac{NK}{M} \right\rceil, \quad \forall j \in J$$

$$\sum_{j \in J_i} (L(i, j) + S(i, j) + R1(i, j) + R2(i, j)) = K, \quad \forall i \in I$$

$$\text{Min} \sum_{i \in I} \sum_{j \in J_i} w_{i,j} (L(i, j) + S(i, j) + R1(i, j) + R2(i, j)) + \alpha \cdot \sum_{i \in I} \sum_{j \in J_i} \sum_{\substack{jj \in J_i \\ j \neq jj}} sl(i, j, jj)$$

Also, constraints 13−16 can be removed from the model and constraints 12 need to be rewritten in order to ensure that no reviewer (j) is assigned to more than one position for a single proposal (i). Using this form of the model, we remove (IJ_i)

$$L(i, j) + S(i, j) + R1(i, j) + R2(i, j) \le 1, \quad \forall i \in I, j \in J_i$$

binary variables and ($4IJ_i$) constraints. In the examples that follow, we will refer to this model as the reformulated version of the model.

4. Computational Examples

In this section, several panel-assignment problems are solved to demonstrate the effectiveness of the proposed approach. Each example is implemented with GAMS 2.50[29] on a 3.20 GHz Linux workstation and is solved with the linear solver CPLEX 9.0.[30]

4.1. Example 1. In the first example, a simple panel-assignment problem is solved which involves 10 reviewers that must be assigned to 24 proposals so that each proposal is reviewed exactly 3 times. Thus, $N = 24$, $M = 10$, and $K = 3$. The preference criteria matrix for each reviewer on each proposal can be found in Table 1, where entries are assigned as described in Section 3.1.2.

Table 1. Preference Criteria Matrix for Example 1

	rev1	rev2	rev3	rev4	rev5	rev6	rev7	rev8	rev9	rev10
prop1	1	2	24	24	24	24	10	2	3	24
prop2	2	0	24	24	24	24	11	4	1	24
prop3	12	0	6	24	8	2	12	24	5	1
prop4	10	3	3	1	5	14	0	5	12	24
prop5	3	12	11	24	24	16	24	6	5	24
prop6	24	6	8	24	0	13	7	7	1	8
prop7	6	4	24	24	24	24	6	1	1	24
prop8	24	24	24	24	10	8	24	24	12	1
prop9	24	24	24	24	4	9	24	24	12	4
prop10	12	10	2	1	24	3	5	24	3	1
prop11	10	24	24	1	11	24	24	24	12	24
prop12	24	24	5	1	7	7	24	24	12	8
prop13	24	5	1	24	24	15	4	24	3	4
prop14	6	24	12	1	24	0	3	24	12	24
prop15	24	9	4	24	24	12	2	9	5	24
prop16	24	1	7	1	3	6	24	24	12	4
prop17	24	24	24	24	9	1	24	24	12	4
prop18	24	24	24	24	2	4	24	24	12	4
prop19	8	8	24	24	24	24	1	8	9	24
prop20	24	24	9	7	6	5	24	24	12	6
prop21	24	24	10	7	12	11	8	10	7	24
prop22	8	11	24	24	24	0	24	24	12	24
prop23	6	24	24	24	24	10	9	3	9	24
prop24	24	7	24	7	1	0	24	24	12	8

Table 2. First Optimal Assignment for Example 1, Obj = 329

	rev1	rev2	rev3	rev4	rev5	rev6	rev7	rev8	rev9	rev10
prop1		L						S	R1	
prop2	S							R1	L	
prop3			R1			S				L
prop4		S	L					R1		
prop5	L							R1	S	
prop6			R1				S		L	
prop7		R1						L	S	
prop8					R1	S				L
prop9					L	R1				S
prop10			R1	S						L
prop11	S			L	R1					
prop12			S	L		R1				
prop13		R1	L				S			
prop14	R1			L			S			
prop15			S				L		R1	
prop16		L			S					R1
prop17					R1	L				S
prop18					L	S				R1
prop19	R1						L	S		
prop20				R1	S	L				
prop21				S			R1		L	
prop22	L	S							R1	
prop23	S						R1	L		
prop24		S		R1	L					
LEAD	2	2	2	3	3	2	2	2	3	3
SCRIBE	3	3	2	2	2	3	3	2	2	2
REV1	2	2	3	2	3	2	2	3	3	2
TOTAL	7	7	7	7	8	7	7	7	8	7

The optimal solution has an objective function value of 329 and the assignments can be found in Table 2. Note that each reviewer is assigned to 7 or 8 proposals and is assigned to be the LEAD (L), the SCRIBE (S), and the REV1 (R1) either 2 or 3 times each. In addition, although the problem involves a large number of binary variables (2165 for the original model and 1932 for the reformulated model), it is solved to optimality at the root node in a fraction of a second. The next four optimal integer solutions, in a rank-ordered list, are provided in the Appendix.

4.1.1. Comparison with Manual Solution. A manual solution was also previously determined for this example problem by the NSF, allowing a direct comparison with the proposed mathematical approach. The manual solution corresponds to an objective function value of 342, and the assignment of reviewers to proposals can be seen in Table 3. Note that the manual solution violates several of the constraints in the proposed formulation. First, reviewer 2 is assigned to 4 proposals as REV1. Also, proposals 6, 11, 20, and 21 assign the position REV1 to a reviewer who has a higher preference (or lower $w_{i,j}$) for the proposal than the reviewer assigned to the position SCRIBE. In addition, proposal 10 assigns the position SCRIBE to a reviewer who has a higher preference for the proposal than the reviewer assigned to the position LEAD. Thus, the proposed mathematical model is able to determine an assignment of reviewers to proposals with an optimal objective function value and satisfies all the required conditions. In contrast, the manual solution has an inferior objective function value and does not meet all the necessary requirements.

Table 3. Manual Assignment for Example 1, Obj = 342

	rev1	rev2	rev3	rev4	rev5	rev6	rev7	rev8	rev9	rev10
prop1	L	R1						S		
prop2	S							R1	L	
prop3			R1			S				L
prop4		L	S					R1		
prop5	L							R1	S	
prop6		R1						S	L	
prop7		R1						L	S	
prop8					R1	S				L
prop9					S	R1				L
prop10			L	S					R1	
prop11	R1			L	S					
prop12			S	L		R1				
prop13			L				S			R1
prop14	R1			L			S			
prop15			S				L		R1	
prop16		L	R1							S
prop17					R1	L				S
prop18					L	S				R1
prop19	S	R1					L			
prop20				S	R1	L				
prop21				R1			S		L	
prop22	L	S					R1			
prop23	S						R1	L		
prop24		S		R1	L					
LEAD	3	2	2	3	2	2	2	2	3	3
SCRIBE	3	2	3	2	2	3	3	2	2	2
REV1	2	4	2	2	3	2	2	3	2	2
TOTAL	8	8	7	7	7	7	7	7	7	7

Table 4. Preference Criteria Matrix for Example 2

	rev1	rev2	rev3	rev4	rev5
prop1	3	1	10	3	0
prop2	6	2	9	6	1
prop3	4	3	8	4	4
prop4	0	4	7	8	6
prop5	1	5	6	1	2
prop6	2	6	5	2	3
prop7	8	7	4	8	5
prop8	5	8	3	5	0
prop9	10	9	2	8	7
prop10	3	10	1	3	10

4.2. Example 2. The second example considers an instance of the panel-assignment problem which cannot satisfy the preference constraints for reviewers to proposals and, thus, must introduce infeasibilities (i.e., $sl(i, j, jj)$) into the solution. Thus, if the slack variables are not included, then the problem is infeasible. This example involves 5 reviewers that must be assigned to 10 proposals so that each proposal is reviewed exactly 4 times. Thus, $N = 10$, $M = 5$, and $K = 4$. The preference criteria matrix for each reviewer on each proposal can be seen in Table 4, where entries are assigned as described in Section 3.1.2.

The optimal solution using $\alpha = 1000$ has an objective function value of 5188, where the first term in the objective has a value of 188. The assignments and infeasibilities can be found in Table 5. Note that each reviewer is assigned to 8 proposals and is assigned to be the LEAD (L), the SCRIBE

Table 5. Optimal Assignment for Example 2, Obj = 188

	rev1	rev2	rev3	rev4	rev5	infeasibilities
prop1	S	L	R2	R1		
prop2	R1	S		R2	L	
prop3	L	S		R1	R2	S ↔ L by 1
prop4		L	S	R2	R1	R1 ↔ S by 1
prop5	S		R2	L	R1	
prop6		R2	R1	L	S	
prop7	R2	R1	S		L	S ↔ L by 1
prop8	L	R2	R1	S		R1 ↔ S by 2
prop9	R2	R1	L		S	
prop10	R1		L	S	R2	
LEAD	2	2	2	2	2	
SCRIBE	2	2	2	2	2	
REV1	2	2	2	2	2	
TOTAL	8	8	8	8	8	

Table 6. Model and Solution Statistics for Example 3

	random 1	random 2	random 3	random 4
best objective	810	814	899	888
binary variables (orig.)	14697	14722	14732	14741
binary variables (reform.)	13736	13761	13772	13781
avg. CPU time (s) (orig.)	1.34	1.12	3.84	7.12
avg. CPU time (s) (reform.)	0.84	0.71	1.38	2.91

(S), the REV1 (R1), and the REV2 (R2) exactly 2 times each. However, these assignments are made by introducing infeasibilities into the preference constraints for reviewers to proposals. For instance, for proposal 3, reviewer 1 is assigned to be the LEAD and reviewer 2 is assigned to be the SCRIBE. However, reviewer 2 has a preference of 3 for proposal 3 and reviewer 1 has a preference of 4. Thus, the assignment of positions does not follow the preference constraints for reviewers to proposals. This assignment, though, corresponds to the minimum violation of the preference constraints for reviewers to proposals. Note that there are additional integer solutions with an objective function value of 188, each which corresponds to a different set of violations of the preference constraints.

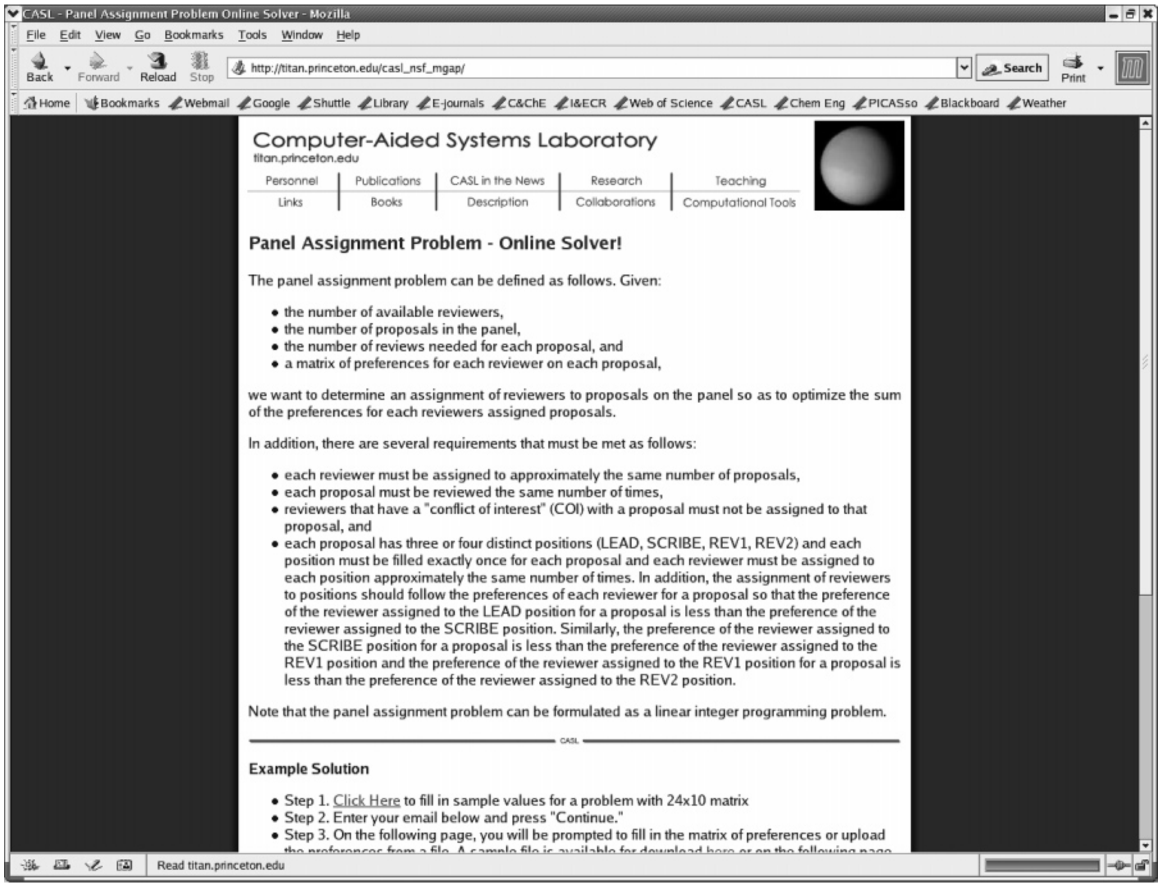
Table 7. Model and Solution Statistics for Example 4

	random 1	random 2	random 3	random 4
best objective	1674	1712	1809	1663
binary variables (orig.)	104186	104228	104318	104438
binary variables (reform.)	100344	100388	100476	100593
avg. CPU time (s) (orig.)	51.41	21.68	8.77	9.79
avg. CPU time (s) (reform.)	18.73	7.52	5.72	5.36

4.3. Example 3. The third example considers a larger panel-assignment problem which involves 20 reviewers that must be assigned to 50 proposals so that each proposal is reviewed exactly 3 times. Thus, $N = 50$, $M = 20$, and $K = 3$. The preference criteria matrix for each reviewer on each proposal is generated randomly so that each reviewer has 2 conflicts of interest and gives preferences for 25 proposals from 1 to 25. Then, the other 23 proposals are given a worst-case value of 50. Four different randomly generated preference criteria were considered, and each was solved with both the original model and the reformulated model to get 10 integer solutions. The model and solution statistics for this example can be seen in Table 6.

The optimal objective function varies between each randomly generated preference criteria, and the required computational effort can be seen to be situation and model dependent. However, the problem is still solvable to optimality in a reasonable amount of time.

4.4. Example 4. The fourth example considers an even larger panel-assignment problem which involves 40 reviewers that must be assigned to 100 proposals so that each proposal is


CASL - Panel Assignment Problem Online Solver - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://titan.princeton.edu/casl_nsf_mgap/
Search Print
Home Bookmarks Webmail Google Shuttle Library E-journals C&ChE I&ECR Web of Science CASL Chem Eng PICASso Blackboard Weather
Computer-Aided Systems Laboratory
titan.princeton.edu
Personnel Publications CASL in the News Research Teaching
Links Books Description Collaborations Computational Tools
Read titan.princeton.edu

Panel Assignment Problem - Online Solver!

The panel assignment problem can be defined as follows. Given:

- the number of available reviewers,
- the number of proposals in the panel,
- the number of reviews needed for each proposal, and
- a matrix of preferences for each reviewer on each proposal,

we want to determine an assignment of reviewers to proposals on the panel so as to optimize the sum of the preferences for each reviewers assigned proposals.

In addition, there are several requirements that must be met as follows:

- each reviewer must be assigned to approximately the same number of proposals,
- each proposal must be reviewed the same number of times,
- reviewers that have a "conflict of interest" (COI) with a proposal must not be assigned to that proposal, and
- each proposal has three or four distinct positions (LEAD, SCRIBE, REV1, REV2) and each position must be filled exactly once for each proposal and each reviewer must be assigned to each position approximately the same number of times. In addition, the assignment of reviewers to positions should follow the preferences of each reviewer for a proposal so that the preference of the reviewer assigned to the LEAD position for a proposal is less than the preference of the reviewer assigned to the SCRIBE position. Similarly, the preference of the reviewer assigned to the SCRIBE position for a proposal is less than the preference of the reviewer assigned to the REV1 position and the preference of the reviewer assigned to the REV1 position for a proposal is less than the preference of the reviewer assigned to the REV2 position.

Note that the panel assignment problem can be formulated as a linear integer programming problem.

CASL

Example Solution

- Step 1. Click Here to fill in sample values for a problem with 24x10 matrix
- Step 2. Enter your email below and press "Continue."
- Step 3. On the following page, you will be prompted to fill in the matrix of preferences or upload the preferences from a file. A sample file is available for download here or on the following page

Figure 2. Panel-assignment problem online solver.

reviewed exactly 3 times. Thus, $N = 100$, $M = 40$, and $K = 3$. The preference criteria matrix for each reviewer on each proposal is again generated randomly. However, in this problem, each reviewer is given 4 conflicts of interest and gives preferences for 50 proposals from 1 to 50. Then, the other 46 proposals are given a worst-case value of 100. Four different randomly generated preference criteria were considered, and each was solved with both the original model and the reformulated model to get three integer solutions. The model and solution statistics for this example can be seen in Table 7.

Even though the optimal objective function value and the required computational effort varies between each randomly generated preference criteria, the problem is still solvable to optimality in a reasonable amount of time.

5. Panel-Assignment Problem Online Solver

A web-based interface has been created to allow users to solve instances of the above-defined panel-assignment problem using the proposed mathematical formulation (see Figure 2). The interface can be found at http://titan.princeton.edu/casl_nsf_mgap/ or under the "Links" tab at http://titan.princeton.edu. The interface allows the user to define and then solve a panel-assignment problem to determine a small subset of optimal integer solutions. The user must enter the number of proposals to consider (N), the number of reviewers to consider (M), the number of reviews needed per proposal (K) where $K = 3$ or $K = 4$, and the number of desired integer solutions. Note that, the more integer solutions to be determined, the longer the total computational time required to generate the desired output. The interface also allows the preference criteria matrix, $w_{i,j}$, to be entered manually in a provided table or to be uploaded from a text file or an Excel file, each with a predefined format. The interface automatically fills in blank entries in the preference criteria matrix according to the rules set forth in Section 3.1.2. After the user submits the required input data, a file is generated using the GAMS[29] modeling language, and the problem is solved on a 3.2GHz Linux workstation using the linear solver CPLEX.[30] The solver's progress is reported onscreen to the user, and once finished, an output file is generated that can be downloaded for the user to save or print and/or it can be e-mailed directly to the user's e-mail address.

6. Conclusions

In this paper, we have introduced a novel mathematical framework to address the NSF panel-assignment problem which is formulated as a multiresource, preference-constrained generalized assignment problem. The proposed model allows for the identification of an assignment of three or four reviewers to each proposal in a panel so as to optimize the sum of a set of preference criteria for each reviewer on each proposal in the panel, while ensuring that each reviewer is assigned to approximately the same number of proposals. In addition, the problem is defined so that each proposal has three or four distinct positions that are assigned to reviewers based upon preference criteria so that each reviewer holds each position approximately the same number of times. The resulting mathematical formulation results in an integer or a mixed-integer linear programming problem and can be solved to optimality. Several computational examples are presented to demonstrate the effectiveness of the proposed approach. Also, an online solver for the NSF panel-assignment problem has been developed and will be made available to the scientific community.

Finally, the mathematical formulation proposed for the preference-constrained generalized assignment problem can be used for a variety of additional applications. For instance, management in any corporation could use the generalized assignment problem to assign small groups or even individuals to different company projects. Hospitals could use the assignment problem to assign doctors and nurses to shifts or to patients. Defense agencies could use the proposed approach to assign combat units to different military tasks. Also, manufacturing facilities could use the mathematical framework to assign operators or maintenance workers to shifts. Although we have named only a few applications, it is clear that the mathematical framework proposed in this work can be used on a wide variety of problems, providing a systematic and efficient method to solve the preference-constrained generalized assignment problem.

Table 8. Second Optimal Assignment for Example 1, Obj = 329

	rev1	rev2	rev3	rev4	rev5	rev6	rev7	rev8	rev9	rev10
prop1	L	R1						S		
prop2	S							R1	L	
prop3			R1			S				L
prop4		L	S					R1		
prop5	L							R1	S	
prop6		S	R1						L	
prop7		R1						L	S	
prop8					R1	S				L
prop9					S	R1				L
prop10			L			S	R1			
prop11	S			L	R1					
prop12			S	L		R1				
prop13			L				S			R1
prop14	R1			L			S			
prop15			S				L		R1	
prop16		L		S						R1
prop17					R1	L				S
prop18					L	R1				S
prop19	R1						L	S		
prop20				R1	S	L				
prop21				S			R1		L	
prop22	L	S							R1	
prop23	S						R1	L		
prop24		S		R1	L					
LEAD	3	2	2	3	2	2	2	2	3	3
SCRIBE	3	3	3	2	2	3	2	2	2	2
REV1	2	2	2	2	3	3	3	3	2	2
TOTAL	8	7	7	7	7	8	7	7	7	7

Table 9. Third Optimal Assignment for Example 1, Obj = 329

	rev1	rev2	rev3	rev4	rev5	rev6	rev7	rev8	rev9	rev10
prop1	L	R1						S		
prop2	S							R1	L	
prop3			R1			S				L
prop4		L	S					R1		
prop5	L							R1	S	
prop6		S	R1						L	
prop7		R1						L	S	
prop8					R1	S				L
prop9					S	R1				L
prop10			L				R1		S	
prop11	S			L	R1					
prop12			S	L		R1				
prop13			L				S			R1
prop14	R1			L			S			
prop15			S				L		R1	
prop16		L		S						R1
prop17					R1	L				S
prop18					L	R1				S
prop19	R1						L	S		
prop20				R1	S	L				
prop21				S			R1		L	
prop22	L	S							R1	
prop23	S						R1	L		
prop24		S		R1	L					
LEAD	3	2	2	3	2	2	2	2	3	3
SCRIBE	3	3	3	2	2	2	2	2	3	2
REV1	2	2	2	2	3	3	3	3	2	2
TOTAL	8	7	7	7	7	7	7	7	8	7

Table 10. Fourth Optimal Assignment for Example 1, Obj = 330

	rev1	rev2	rev3	rev4	rev5	rev6	rev7	rev8	rev9	rev10
prop1	L	R1						S		
prop2	S							R1	L	
prop3			R1			S				L
prop4		L	S					R1		
prop5	L							R1	S	
prop6			R1				S		L	
prop7		R1						L	S	
prop8					R1	S				L
prop9					S	R1				L
prop10			L			R1			S	
prop11	S			L	R1					
prop12			S	L						R1
prop13		R1	L				S			
prop14	R1			L			S			
prop15			S				L		R1	
prop16		L		S						R1
prop17					R1	L				S
prop18					L	R1				S
prop19	R1						L	S		
prop20				R1	S	L				
prop21				S			R1		L	
prop22	L	S							R1	
prop23	S						R1	L		
prop24		S		R1	L					
LEAD	3	2	2	3	2	2	2	2	3	3
SCRIBE	3	2	3	2	2	2	3	2	3	2
REV1	2	3	2	2	3	3	2	3	2	2
TOTAL	8	7	7	7	7	7	7	7	8	7

Table 11. Fifth Optimal Assignment for Example 1, Obj = 330

	rev1	rev2	rev3	rev4	rev5	rev6	rev7	rev8	rev9	rev10
prop1	L	R1						S		
prop2	S							R1	L	
prop3			R1			S				L
prop4		L	S					R1		
prop5	L							R1	S	
prop6		S					R1		L	
prop7		R1						L	S	
prop8					R1	S				L
prop9					S	R1				L
prop10			L			S	R1			
prop11	S			L	R1					
prop12			S	L		R1				
prop13			L				S			R1
prop14	R1			L			S			
prop15			S				L		R1	
prop16		L		S						R1
prop17					R1	L				S
prop18					L	R1				S
prop19	R1						L	S		
prop20				R1	S	L				
prop21			R1	S					L	
prop22	L	S							R1	
prop23	S						R1	L		
prop24		S		R1	L					
LEAD	3	2	2	3	2	2	2	2	3	3
SCRIBE	3	3	3	2	2	3	2	2	2	2
REV1	2	2	2	2	3	3	3	3	2	2
TOTAL	8	7	7	7	7	8	7	7	7	7

Acknowledgment

The authors acknowledge the comments of the reviewers which helped in improving the submitted manuscript, particularly for the elimination of the $y(i,j)$ variables.

Appendix A: Additional Four Rank-Ordered Integer Solutions for Example 1

Tables 8−11 show four additional rank-ordered integer solutions for Example 1.

Literature Cited

(1) Sahni, S.; Gonzalez, T. P−Complete Approximation Problems. *J. Assoc. Comput. Mach.* **1976**, *23*, 555.

(2) Garey, M. R.; Johnson, D. S. *Computers and Intractability: A Guide to the Theory of NP−Completeness*; Freeman: San Francisco, CA, 1979.

(3) Fisher, M. L.; Jaikumar, R.; Van Wassenhove, L. N. A Multiplier Adjustment Method for the Generalized Assignment Problem. *Manage. Sci.* **1986**, *32*, 1095.

(4) Cattrysse, D. G.; Van Wassenhove, L. N. A Survey of Algorithms for the Generalized Assignment Problem. *Eur. J. Oper. Res.* **1992**, *60*, 260.

(5) Osman, I. H. Heuristics for the Generalized Assignment Problem: Simulated Annealing and Tabu Search Approaches. *OR Spektrum* **1995**, *17*, 211.

(6) Ross, G. T.; Soland, P. M. Branch and Bound Algorithm for a Generalized Assignment Problem. *Math. Program.* **1975**, *8*, 91.

(7) Guignard, M.; Rosenwein, M. An Improved Dual-Based Algorithm for the Generalized Assignment Problem. *Oper. Res.* **1989**, *37*, 658.

(8) Savelsbergh, M. A Branch-and-Price Algorithm for the Generalized Assignment Problem. *Oper. Res.* **1997**, *45*, 831.

(9) Nauss, R. M. Solving the Generalized Assignment Problem: An Optimizing and Heuristic Approach. *INFORMS J. Comput.* **2003**, *15*, 249.

(10) Martello, S.; Toth, P. An Algorithm for the Generalized Assignment Problem. In *Proceedings of the Ninth International Conference on Operational Research*; Brans, J. P., Ed.; North-Holland: Amsterdam, The Netherlands, 1981; pp 589−603.

(11) Martello, S.; Toth, P. *Knapsack Problems: Algorithms and Computer Implementations*; John Wiley & Sons: Chichester, U.K., 1990.

(12) Trick, M. A. A Linear Relaxation Heuristic for the Generalized Assignment Problem. *Nav. Reg. Log.* **1992**, *39*, 137.

(13) Amini, M. M.; Racer, M. A Rigorous Computational Comparison of Alternative Solution Methods for the Generalized Assignment Problem. *Manage. Sci.* **1994**, *40*, 868.

(14) Amini, M. M.; Racer, M. A Hybrid Heuristic for the Generalized Assignment Problem. *Eur. J. Oper. Res.* **1995**, *87*, 343.

(15) Cattrysse, D. G.; Salomon, M.; Van Wassenhove, L. N. A Set Partitioning Heuristic for the Generalized Assignment Problem. *Eur. J. Oper. Res.* **1994**, *72*, 167.

(16) Cattrysse, D. G.; Degraeve, Z.; Tistaert, J. Solving the Generalized Assignment Problem using Polyhedral Results. *Eur. J. Oper. Res.* **1998**, *108*, 618.

(17) Lorena, L. A. N.; Narciso, M. G. Relaxation Heuristics for a Generalized Assignment Problem. *Eur. J. Oper. Res.* **1996**, *91*, 600.

(18) Chu, P. C.; Beasley, J. E. A Genetic Algorithm for the Generalized Assignment Problem. *Comput. Oper. Res.* **1997**, *24*, 17.

(19) Feltl, H.; Raidl, G. R. An Improved Hybrid Genetic Algorithm for the Generalized Assignment Problem. In *Proceedings of the Nineteenth Annual ACM Symposium on Applied Computing*, Nicosia, Cyprus, 2004; ACM Press: New York; pp 990−995.

(20) Yagiura, M.; Yamaguchi, T.; Ibaraki, T. A Variable Depth Search Algorithm with Branching Search for the Generalized Assignment Problem. *Optim. Method. Soft.* **1998**, *10*, 419.

(21) Yagiura, M.; Ibaraki, T.; Glover, F. An Ejection Chain Approach for the Generalized Assignment Problem. *INFORMS J. Comput.* **2004**, *16*, 133.

(22) Laguna, M.; Kelly, J. P.; Gonzalez-Velarde, J. L.; Glover, F. Tabu Search for the Multilevel Generalized Assignment Problem. *Eur. J. Oper. Res.* **1995**, *82*, 176.

(23) French, A. P.; Wilson, J. M. Heuristic Solution Methods for the Multilevel Generalized Assignment Problem. *J. Heuristics* **2002**, *8*, 143.

(24) Park, J. S.; Lim, B. H.; Lee, Y. A Lagrangian Dual-Based Branch-and-Bound Algorithm for the Generalized Multi-Assignment Problem. *Manage. Sci.* **1998**, *44*, S271.

(25) Gavish, B.; Pirkul, H. Algorithms for the Multi-Resource Generalized Assignment Problem. *Manage. Sci.* **1991**, *37*, 695.

(26) Mazzola, J. B.; Wilcox, S. P. Heuristics for the Multi-Resource Generalized Assignment Problem. *Nav. Reg. Log.* **2001**, *48*, 468.

(27) Toktas, B.; Yen, J. W.; Zabinsky, Z. B. Addressing Capacity Uncertainty in Resource-Constrained Assignment Problems. *Comput. Oper. Res.* **2006**, *33*, 724.

(28) Floudas, C. A. *Nonlinear and Mixed-Integer Optimization*; Oxford University Press: New York, 1995.

(29) Brooke, A.; Kendrick, D.; Meeraus, A.; Raman, R. *GAMS: A User's Guide*; South San Franciso, CA, 2003.

(30) CPLEX. *ILOG CPLEX 9.0 User's Manual*; 2005.

Received for review April 21, 2005
Revised manuscript received September 16, 2005
Accepted October 12, 2005

IE050478H