Discrete Applied Mathematics 157 (2009) 49–67

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

The generalized assignment problem with flexible jobs

Chase Rainwater *, Joseph Geunes, H. Edwin Romeijn

Department of Industrial and Systems Engineering, University of Florida, P.O. Box 116595, Gainesville, FL 32611-6595, United States

ARTICLE INFO

Article history:
Received 30 July 2007
Received in revised form 7 April 2008
Accepted 15 April 2008
Available online 29 May 2008

Keywords:
Generalized assignment problem Heuristics
Asymptotic feasibility
Asymptotic optimality

ABSTRACT

The Generalized Assignment Problem (GAP) seeks an allocation of jobs to capacitated resources at minimum total assignment cost, assuming a job cannot be split among multiple resources. We consider a generalization of this broadly applicable problem in which each job must not only be assigned to a resource, but its resource consumption must also be determined within job-specific limits. In this profit-maximizing version of the GAP, a higher degree of resource consumption increases the revenue associated with a job. Our model permits a job's revenue per unit resource consumption to decrease as a function of total resource consumption, which allows modeling quantity discounts. The objective is then to determine job assignments and resource consumption levels that maximize total profit. We develop a class of heuristic solution methods, and demonstrate the asymptotic optimality of this class of heuristics in a probabilistic sense.

1. Introduction

Over the past decade, the operations literature has increasingly emphasized demand and revenue management (see, e.g., Talluri and Van Ryzin [18]), as firms seek to exploit sources of supply and demand flexibility to increase profit margins. This has led to a number of new models that focus on profit maximization by accounting for both the costs and revenue implications associated with operations decisions. Recently, Chen and Hall [6] introduced several new "maximum profit scheduling" models that implicitly account for the fact that operations scheduling decisions can affect demand and therefore revenue. Another stream of literature considers optimal inventory management when demand levels (and hence revenues) depend on inventory levels (e.g., Baker and Urban [1,2], Gerchak and Wang [10], and Balakrishnan et al. [4]) and/or shelf space allocation (Wang and Gerchak [21]), both of which impact operations costs. Several papers have also considered maximizing profit in production planning contexts with price-dependent demand, where production and inventory costs are determined by solving an optimization problem containing a lot-sizing structure (e.g., Thomas [19], Kunreuther and Schrage [13], Gilbert [12], Biller et al. [5], Deng and Yano [7], Geunes, Romeijn, and Taaffe [11], and van den Heuvel and Wagelmans [20]). In this paper we take a profit-maximizing view of a general class of problems involving the assignment of requirements (or jobs) to available capacitated resources (agents). In particular, we introduce the *Generalized Assignment Problem with Flexible Jobs* (GAPFJ), a profit-maximizing extension of the Generalized Assignment Problem (GAP) in which both the assignments of jobs to available agents and the degree of resource consumption associated with each assignment must be determined. As we next discuss, this class of problems finds application in a wide range of practical settings.

In certain contexts, customer demands involve varying degrees of flexibility, allowing a supplier to better match these customer demands with operations resources. For example, for some construction materials, such as steel and wood, distributors of these materials will accept deliveries from suppliers in a range of sizes (Balakrishnan and Geunes [3]). The distributors permit this flexibility because they often perform further customized cutting and finishing operations for their own customers, whose exact size specifications are not known to the distributor in advance. Suppliers to such distributors

* Corresponding author. Tel: +1 352 281 7617; fax: +1 352 392 3537.
E-mail addresses: crainwa@ufl.edu (C. Rainwater), geunes@ise.ufl.edu (J. Geunes), romeijn@ise.ufl.edu (H. Edwin Romeijn).

0166-218X/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2008.04.017

are often compensated based on total weight delivered (within certain limits deemed acceptable to the distributor). Thus, these suppliers must match these flexible demands with available resources in order to maximize net profit. Clearly, if the supplier has unlimited resources, they can maximize profit by delivering at the upper limit of the distributor's stated acceptable size range. If, however, the supplier faces resource constraints (e.g., in terms of its quantity and sizes of raw materials) and must meet each element of a collection of customer demands, the problem of assigning these demands to available resources in order to maximize net profit is non-trivial.

Sales and advertising planning involves similar tradeoffs between revenue generation and resource constraints and costs. In sales force planning contexts, for example, the sales force serves as a set of resources, where each salesperson has a limited amount of time and/or effort that they can allocate to customers. It is often the case that the greater the amount of effort a salesperson allocates to a given customer, the greater the return from that customer in terms of sales. The planning phase therefore involves determining the assignment of sales force to customers and the degree of effort a salesperson should devote to each assigned customer in order to maximize the total return from customers (or expected return, when the relationship between effort and sales is not deterministic). This sales setting may be interpreted more generally as applying to a set of available marketing instruments, where an allocation of capacity-constrained marketing instruments to customers must be determined in order to maximize profit.

Since the GAPFJ generalizes the GAP, it is clearly NP-Hard. Furthermore, since the feasibility problem associated with the GAP is NP-Complete, it is clear that the feasibility problem associated with GAPFJ is NP-Complete as well. We therefore develop a customized family of heuristics, and show that this class of heuristics is asymptotically feasible and optimal with probability one as the number of jobs goes to infinity under a very broad probabilistic model for the problem parameters. Our heuristics are in the same spirit as certain heuristics that have been developed for the GAP by Martello and Toth [14] and Romeijn and Romero Morales [16]. In particular, given a vector of multipliers (each corresponding to an agent), a weight function is defined to measure the pseudo-profit of assigning a job to an agent. This weight function is then used to judiciously determine (i) the order in which to assign the jobs, (ii) the agent to which each job should be assigned, and (iii) an appropriate job size. In addition, these functions motivate improvement heuristics that are essential in order to be able to derive attractive performance guarantees for the heuristic. The main contribution of this paper is the development of a heuristic that is asymptotically feasible and optimal with probability one under a very general stochastic model of the problem parameters. Due to the nature of the GAPFJ, our approach for obtaining such guarantees is, particularly for the most general version of our model, significantly different from approaches used in the case of the GAP. Specifically, we rely heavily both on the solution to a suitable perturbation of the GAPFJ and on carefully designed solution improvement techniques. Thus, in addition to contributing to the literature on applied optimization in operations, we also provide new techniques for algorithm development and asymptotic analysis for combinatorial optimization problems. As our computational tests show, our heuristic solution approach is able to find optimal or near-optimal solutions with very limited computational effort for a broad range of problem dimensions.

The remainder of this paper is organized as follows. Section 2 formally defines the problem and provides a number of important structural results; these results both motivate a class of heuristics and enable us to derive associated performance guarantees in Section 3. Section 4 discusses approaches to further improve the heuristic methods we propose, and in Section 5 we present the results of our computational study, which validate the effectiveness of our proposed methods. We end the paper with some concluding remarks and directions of future research in Section 6.

2. Generalized assignment problem with flexible jobs

2.1. Model formulation and analysis

Consider a situation in which there are n jobs, each of which needs to be assigned to exactly one of m agents. The quantity of a single resource available to agent i is denoted by b_i ($i = 1, \ldots, m$). If job j is assigned to agent i, a fixed profit of p_{ij} is received and a fixed amount a_{ij} of the resource is consumed. Moreover, the corresponding job size must be set to a value within the nonempty interval $[\ell_{ij}, u_{ij}]$ (where $0 \le \ell_{ij} \le u_{ij} < \infty$). Finally, per unit of job size, a revenue of r_{ij} is accrued, and we assume that a single unit of resource is consumed. The objective is to determine an assignment of jobs to agents together with corresponding job sizes that together maximize total profit while satisfying the resource capacity constraints of the agents.

Observe that the total revenue received when job j is assigned to agent i at a level of $v_{ij} \in [\ell_{ij}, u_{ij}]$ is equal to $p_{ij} + r_{ij}v_{ij}$, which corresponds to a per-unit associated price of $\frac{p_{ij}}{v_{ij}} + r_{ij}$. Our model thus allows for a form of quantity discounts that may provide an incentive to customers to accept a flexible range of job sizes. Similarly, the total quantity of the resource that is consumed when job j is assigned to agent i at a level of $v_{ij} \in [\ell_{ij}, u_{ij}]$ is equal to $a_{ij} + v_{ij}$, which corresponds to a per-unit associated resource consumption of $\frac{a_{ij}}{v_{ij}} + 1$. Our model can thus account for the presence of, for example, fixed job setup times or other loss of resources at the start of a production run.

Using the preceding notation, the GAPFJ can be formulated as a mixed-integer linear programming problem as follows:

$$\text{maximize } \sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}x_{ij} + \sum_{i=1}^{m}\sum_{j=1}^{n} r_{ij}v_{ij} \tag{1}$$

subject to (P)

$$\sum_{j=1}^{n} a_{ij}x_{ij} + \sum_{j=1}^{n} v_{ij} \le b_i \quad i = 1, \ldots, m \tag{2}$$

$$\sum_{i=1}^{m} x_{ij} = 1 \quad j = 1, \ldots, n \tag{3}$$

$$v_{ij} \ge \ell_{ij}x_{ij} \quad i = 1, \ldots, m;\ j = 1, \ldots, n \tag{4}$$

$$v_{ij} \le u_{ij}x_{ij} \quad i = 1, \ldots, m;\ j = 1, \ldots, n \tag{5}$$

$$x_{ij} \in \{0, 1\} \quad i = 1, \ldots, m;\ j = 1, \ldots, n. \tag{6}$$

Note that the assumption that the unit resource consumption coefficients are equal to one can be made without loss of generality. Moreover, we will assume without loss of generality that the unit revenues are nonnegative. In principle, we allow the fixed profit and resource consumption coefficients to be either positive or negative. However, in most real-life applications we should expect these coefficients to be nonnegative. Finally, note that without loss of generality we could assume that $\ell_{ij} = 0$ for all $i = 1, \ldots, m$ and $j = 1, \ldots, n$ by appropriately modifying the fixed profit and resource consumption coefficients. However, for clarity of interpretation of our model, algorithms, and results we will allow for positive values of these lower bounds on the job sizes.

The solution approach that we will develop and analyze in this paper is a class of heuristics that is inspired by the Lagrange relaxation of a reformulation of the LP-relaxation of (P). In particular, as we will show below, the optimization problem (LP) that is obtained by replacing the binary constraints (6) by nonnegativity constraints

$$x_{ij} \ge 0 \quad i = 1, \ldots, m;\ j = 1, \ldots, n \tag{6$'$}$$

is equivalent to the problem

$$\text{maximize } \sum_{i=1}^{m}\sum_{j=1}^{n} \left(p_{ij} + r_{ij}u_{ij}\right) s_{ij} + \sum_{i=1}^{m}\sum_{j=1}^{n} \left(p_{ij} + r_{ij}\ell_{ij}\right) t_{ij} \tag{7}$$

subject to (LP′)

$$\sum_{j=1}^{n} \left(a_{ij} + u_{ij}\right) s_{ij} + \sum_{j=1}^{n} \left(a_{ij} + \ell_{ij}\right) t_{ij} \le b_i \quad i = 1, \ldots, m \tag{8}$$

$$\sum_{i=1}^{m} \left(s_{ij} + t_{ij}\right) = 1 \quad j = 1, \ldots, n \tag{9}$$

$$s_{ij}, t_{ij} \ge 0 \quad i = 1, \ldots, m;\ j = 1, \ldots, n. \tag{10}$$

Theorem 2.1. *The optimization problems (LP) and (LP′) are equivalent.*

Proof. First note that we may modify (LP) by explicitly introducing (nonnegative) surplus and slack variables to constraints (4) and (5). For convenience, we will scale these so that they are expressed as a fraction of the width of the size range of the corresponding assignment. In other words, constraints (4) and (5) are replaced by

$$v_{ij} - (u_{ij} - \ell_{ij})s_{ij} = \ell_{ij}x_{ij} \quad i = 1, \ldots, m;\ j = 1, \ldots, n \tag{4$'$}$$

$$v_{ij} + (u_{ij} - \ell_{ij})t_{ij} = u_{ij}x_{ij} \quad i = 1, \ldots, m;\ j = 1, \ldots, n \tag{5$'$}$$

$$s_{ij}, t_{ij} \ge 0 \quad i = 1, \ldots, m;\ j = 1, \ldots, n. \tag{11}$$

It is easy to see that this reformulation is valid even if the width of the size range of an assignment is 0, i.e., if $\ell_{ij} = u_{ij}$. Now subtracting constraints (4′) from (5′) yields

$$\left(u_{ij} - \ell_{ij}\right) x_{ij} = \left(u_{ij} - \ell_{ij}\right) \left(s_{ij} + t_{ij}\right) \quad i = 1, \ldots, m;\ j = 1, \ldots, n$$

so that we can set, without loss of generality,

$$x_{ij} = s_{ij} + t_{ij} \quad i = 1, \ldots, m;\ j = 1, \ldots, n. \tag{12}$$

Moreover, multiplying constraints (4′) and (5′) by u_{ij} and ℓ_{ij} respectively yields

$$\ell_{ij}u_{ij}x_{ij} + u_{ij}\left(u_{ij} - \ell_{ij}\right) s_{ij} = u_{ij}v_{ij} \quad i = 1, \ldots, m;\ j = 1, \ldots, n \tag{4$''$}$$

$$\ell_{ij}u_{ij}x_{ij} - \ell_{ij}\left(u_{ij} - \ell_{ij}\right) t_{ij} = \ell_{ij}v_{ij} \quad i = 1, \ldots, m;\ j = 1, \ldots, n. \tag{5$''$}$$

Subtracting (5″) from (4″), we obtain

$$\left(u_{ij} - \ell_{ij}\right)\left(u_{ij}s_{ij} + \ell_{ij}t_{ij}\right) = \left(u_{ij} - \ell_{ij}\right) v_{ij} \quad i = 1, \ldots, m;\ j = 1, \ldots, n$$

so that we can set

$$v_{ij} = u_{ij}s_{ij} + \ell_{ij}t_{ij} \quad i = 1, \ldots, m;\ j = 1, \ldots, n. \tag{13}$$

Notice that the non-negativity of the slack and surplus variables by (11) along with (12) and (13) implies that (10) is sufficient to ensure all non-negativity conditions in (LP) are also satisfied in (LP′). Finally, substituting (12) and (13) into the objective (1) and constraints (2) and (3) of (P) yields the objective (7) as well as constraints (8) and (9) of (LP′). □

Next, let us denote the (nonnegative) dual multipliers of the capacity constraints (8) by λ_i ($i = 1, \ldots, m$) and the (free) dual multipliers of the assignment constraints (9) by μ_j ($j = 1, \ldots, n$). The dual (D′) of (LP′) is then given by

$$\text{minimize } \sum_{i=1}^{m} \lambda_i b_i + \sum_{j=1}^{n} \mu_j$$

$$\text{subject to} \tag{D′}$$

$$\mu_j \geq p_{ij} - \lambda_i a_{ij} + (r_{ij} - \lambda_i)\,\ell_{ij} \quad i = 1, \ldots, m;\ j = 1, \ldots, n \tag{14}$$

$$\mu_j \geq p_{ij} - \lambda_i a_{ij} + (r_{ij} - \lambda_i)\,u_{ij} \quad i = 1, \ldots, m;\ j = 1, \ldots, n \tag{15}$$

$$\lambda_i \geq 0 \quad i = 1, \ldots, m$$

$$\mu_j \text{ free} \quad j = 1, \ldots, n.$$

The following theorem derives a convenient and insightful expression for the optimal value to both (LP′) and (D′) as a function of the dual multipliers λ_i ($i = 1, \ldots, m$) of the capacity constraints (8) only.

Theorem 2.2. *The common optimal value of (LP′) and (D′) can be expressed as*

$$\min_{\lambda \geq 0} L(\lambda)$$

where

$$L(\lambda) = \sum_{j=1}^{n} \max_{i=1,\ldots,m} f_\lambda(i,j) + \sum_{i=1}^{m} \lambda_i b_i$$

and where

$$f_\lambda(i,j) = \begin{cases} p_{ij} - \lambda_i a_{ij} + (r_{ij} - \lambda_i)u_{ij} & \text{if } \lambda_i \leq r_{ij} \\ p_{ij} - \lambda_i a_{ij} + (r_{ij} - \lambda_i)\ell_{ij} & \text{if } \lambda_i > r_{ij}. \end{cases}$$

Proof. From constraints (14) and (15) we obtain that, without loss of optimality, the dual variables μ_j can be chosen as

$$\mu_j = \max_{i=1,\ldots,m} \max \left\{ p_{ij} - \lambda_i a_{ij} + (r_{ij} - \lambda_i)\,\ell_{ij},\, p_{ij} - \lambda_i a_{ij} + (r_{ij} - \lambda_i)\,u_{ij} \right\} \quad j = 1, \ldots, n. \tag{16}$$

Now note that the inner maximum in (16) is attained by the first argument if $\lambda_i \geq r_{ij}$ and by the second argument if $\lambda_i \leq r_{ij}$. This implies that we in fact have

$$\mu_j = \max_{i=1,\ldots,m} f_\lambda(i,j) \quad j = 1, \ldots, n \tag{17}$$

which yields the desired result. □

It is useful to introduce some terminology with respect to a feasible solution (s, t) to (LP′). Consider some job j. If $x_{ij} = s_{ij} + t_{ij} = 1$ for some agent i we say that job j is assigned to agent i and furthermore, job j is referred to as a *non-split* job. Similarly, if $0 < x_{ij} = s_{ij} + t_{ij} < 1$ for some agent i we say that the assignment of job j to agent i is *fractional*, and job j is referred to as a *split job*. More formally, we define the set

$$\mathcal{F} = \left\{(i,j) : 0 < x_{ij} < 1\right\} = \left\{(i,j) : 0 < s_{ij} + t_{ij} < 1\right\}$$

of fractional assignments, and the set

$$\mathcal{S} = \{j : \exists\, i \text{ such that } (i,j) \in \mathcal{F}\}$$

of split jobs. Furthermore, we will say that an assignment (i, j) such that $s_{ij} > 0$ and $t_{ij} = 0$ is executed at its upper bound, while an assignment (i, j) such that $s_{ij} = 0$ and $t_{ij} > 0$ is executed at its lower bound. The set

$$\mathcal{Q} = \left\{(i,j) : s_{ij} > 0 \text{ and } t_{ij} > 0\right\}$$

then consists of the agent/job pairs executed strictly between their bounds. Finally,

$$\mathcal{C} = \left\{ i : \sum_{j=1}^{n} a_{ij}x_{ij} + \sum_{j=1}^{n} v_{ij} = b_i \right\} = \left\{ i : \sum_{j=1}^{n} (a_{ij} + u_{ij})\, s_{ij} + \sum_{j=1}^{n} (a_{ij} + \ell_{ij})\, t_{ij} = b_i \right\}$$

is the set of agents that operate at full-capacity.

The following theorem establishes a close relationship between an optimal solution to (D′) and the corresponding primal optimal solution, provided that the latter is unique.

Theorem 2.3. *Suppose that* (LP′) *is feasible and that the optimal (basic) solution to* (LP′)*, say* (s^*, t^*)*, is unique. Furthermore, let* λ^* *be an associated complementary optimal solution to* (D′)*. The primal and dual solutions then satisfy the following properties.*

(i) *Let* $j \in \mathcal{S}$ *be a split job. Then there exists an agent* i' *such that*

$$f_{\lambda^*}(i', j) = \max_{\substack{i=1,\ldots,m \\ i \neq i'}} f_{\lambda^*}(i, j).$$

(ii) *Let* $j \notin \mathcal{S}$ *be a non-split job. Then it is assigned to agent* i' *if and only if*

$$f_{\lambda^*}(i', j) = \max_{i=1,\ldots,m} f_{\lambda^*}(i, j)$$

and

$$f_{\lambda^*}(i', j) > \max_{\substack{i=1,\ldots,m \\ i \neq i'}} f_{\lambda^*}(i, j).$$

(iii) *Let* $j \notin \mathcal{S}$ *be a non-split job that is assigned to agent* i'*. Then*

$$s^*_{i'j} = 1 - t^*_{i'j} \begin{cases} = 0 & \text{if } \lambda^*_{i'} > r_{i'j} \\ \in [0, 1] & \text{if } \lambda^*_{i'} = r_{i'j} \\ = 1 & \text{if } \lambda^*_{i'} < r_{i'j}. \end{cases}$$

Proof. First, note that uniqueness of the optimal solution (s^*, t^*) implies that (D′) is nondegenerate. To simplify notation, denote $x^* = s^* + t^*$ (recalling the relationship given in (12)).

(i) Let $j \in \mathcal{S}$ be a split job. This implies that there exist (at least) two agents, say i' and i'', such that $x^*_{i'j}, x^*_{i''j} > 0$. By the definition of x^*, complementary slackness now implies that for both i' and i'', at least one of the corresponding dual constraints (14) and (15) is binding. This, in turn, implies that for this job the maximum in Eq. (17) is attained for both agents i' and i'', yielding claim (i).

(ii) Let $j \notin \mathcal{S}$ be a non-split job. This implies that there exists only a single agent, say i', such that $x_{i'j} > 0$ (in fact, $x_{i'j} = 1$). By complementary slackness and the nondegeneracy of the dual solution, this means that for all agents except i' both corresponding dual constraints (14) and (15) are nonbinding. This, in turn, implies that for this job the maximum in Eq. (17) is attained for only agent i', yielding claim (ii).

(iii) Let $j \notin \mathcal{S}$ be a non-split job that is assigned to agent i', so that $x^*_{i'j} = 1$. First, recall that s_{ij} is the primal variable associated with (14) and t_{ij} is the primal variable associated with (15). Now by complementary slackness and dual nondegeneracy we have that $s^*_{i'j} > 0$ and $t^*_{i'j} = 0$ if and only if $\lambda^*_{i'} > r_{i'j}$, $s^*_{i'j} = 0$ and $t^*_{i'j} > 0$ if and only if $\lambda^*_{i'} < r_{i'j}$, and $s^*_{i'j} > 0$ and $t^*_{i'j} > 0$ if and only if $\lambda^*_{i'} = r_{i'j}$. Together with (12) this yields the desired result. □

It is interesting to see what Theorem 2.3 implies in terms of the (x, v) variables in our original (LP) formulation.

Corollary 2.4. *Suppose that* (LP′) *is feasible and that the optimal (basic) solution to* (LP′)*, say* (s^*, t^*)*, is unique. Furthermore, let* λ^* *be an associated complementary optimal solution to* (D′)*. Then there exists an optimal solution* (x^*, v^*) *to (LP) that satisfies the following property. If* $j \notin \mathcal{S}$ *is a non-split job that is assigned to agent i in the optimal solution to* (LP′)*, then*

$$v^*_{i'j} \begin{cases} = \ell_{i'j} & \text{if } \lambda^*_{i'} > r_{i'j} \\ \in [\ell_{i'j}, u_{i'j}] & \text{if } \lambda^*_{i'} = r_{i'j} \\ = u_{i'j} & \text{if } \lambda^*_{i'} < r_{i'j}. \end{cases}$$

Proof. The result follows immediately from Theorems 2.1 and 2.3. □

Remark 1. Note that if $\ell_{ij} = u_{ij}$ for at least one pair (i, j), the optimal solution to (LP′) cannot be unique. However, in this case we can arbitrarily set $t_{ij} \equiv 0$ for such pairs, which would allow uniqueness of the optimal solution to (LP′).

Remark 2. Clearly, uniqueness of the optimal solution to (LP′), even with the modification given in Remark 1, cannot be guaranteed for all instances. However, in Section 3 we will introduce a stochastic model for problem instances of (P) under which this can be guaranteed with probability one.

In Section 3 we will use the insights of Theorems 2.2 and 2.3 to develop a heuristic for solving the GAPFJ. We will, however, first derive an important result characterizing the nature of basic feasible solutions to (LP′) that will later prove significant in the average case performance analysis of that heuristic. To arrive at this result we note that the number of nonzero variables in a basic feasible solution is bounded by the number of equality constraints in (LP′). We then count the number of basic variables using the set descriptions defined previously. This analysis provides an important inequality which is used in Corollary 2.6 to arrive at a bound on the number of split jobs and jobs assigned between their bounds in an optimal solution to (LP′).

Theorem 2.5. *Let (s, t) be a basic feasible solution to (LP′). Then the total number of assignments that are either fractional or strictly between their bounds is bounded by the total number of split jobs plus the total number of agents operating at full capacity, i.e.,*

$$|\mathcal{F}| + |\mathcal{Q}| \leq |\mathcal{S}| + |\mathcal{C}|.$$

Proof. The optimization problem (LP′) has $2mn$ variables and $m+n$ equality constraints. The total number of variables which are nonzero in a basic feasible solution is therefore no larger than $m + n$. Now observe that there are

- $n - |\mathcal{S}| + |\mathcal{F}| + |\mathcal{Q}|$ non-zero components in (s, t);
- $m - |\mathcal{C}|$ non-zero slack variables associated with constraints (8);

This yields

$$m + n \geq (n - |\mathcal{S}| + |\mathcal{F}| + |\mathcal{Q}|) + (m - |\mathcal{C}|)$$

which yields the desired result. □

Corollary 2.6. *Let (s, t) be a basic feasible solution to (LP′). Then the total number of jobs that are either split or executed strictly between their bounds is bounded by the number of agents, i.e.,*

$$|\mathcal{S}| + |\mathcal{Q}| \leq m.$$

Proof. Note that each split job has at least two corresponding fractional assignment variables, so that $|\mathcal{F}| \geq 2|\mathcal{S}|$. The result then follows directly from Theorem 2.5 and the fact that $|\mathcal{C}| \leq m$. □

3. An asymptotically optimal heuristic

3.1. Development of the heuristic

There is an attractive intuitive interpretation of the result of Theorem 2.2 by noting that we can interpret the value of the dual variable λ_i as a unit cost of capacity of agent i. Then, note that we can view $p_{ij} - \lambda_i a_{ij}$ as a fixed pseudo-profit that is received if job j is assigned to agent i, regardless of its level. Next, we can view the difference between the actual corresponding unit revenue r_{ij} and the cost λ_i of using a unit of capacity of agent i as a unit pseudo-profit that is received if job j is assigned to agent i. The sign of the pseudo-profit then indicates the level at which a job should be assigned: if the unit pseudo-profit is positive, job j (if assigned to agent i) is executed at its upper bound u_{ij}, yielding a total pseudo-profit of $p_{ij} - \lambda_i a_{ij} + (r_{ij} - \lambda_i)u_{ij}$. Similarly, if the unit pseudo-profit is negative, job j (if assigned to agent i) is executed at its lower bound ℓ_{ij}, yielding a total pseudo-profit of $p_{ij} - \lambda_i a_{ij} + (r_{ij} - \lambda_i)\ell_{ij}$. In summary, the function $f_\lambda(i, j)$ can be viewed as a *pseudo-profit* associated with the assignment of job j to agent i for a given vector of dual prices λ.

We will use this interpretation to propose a heuristic for the GAPFJ. That is, our heuristic will, to a large extent, assign jobs according to the pseudo-profit function. (Note that any nonnegative vector λ defines a distinct pseudo-profit function, so that we will in fact obtain a family of heuristics. However, we later show that the heuristic enjoys an attractive performance guarantee if we use an optimal dual solution to either the original problem or a perturbation thereof.) In particular, we will attempt to assign each job to the agent that maximizes its pseudo-profit function and select the corresponding job size accordingly. More specifically, the most profitable agent for job j is given by

$$i_j = \arg\max_{i \in \mathcal{I}_j} f_\lambda(i, j)$$

where $\mathcal{I}_j \subseteq \{1, \ldots, m\}$ is the set of agents currently under consideration for job j. It is easy to see that, in general, assigning all jobs j to their most profitable agent i_j at a size as described in the preceding paragraph cannot be expected to yield a feasible solution to the GAPFJ. We therefore select the order in which the jobs are assigned by considering not only the maximum pseudo-profit but also the second largest pseudo-profit for each job. We define the difference between these two values:

$$\rho_j = f_\lambda(i_j, j) - \max_{i' \in \mathcal{I}_j \setminus \{i_j\}} f_\lambda(i', j)$$

to be the *desirability* of assigning job j to its most profitable agent. We then assign the jobs to their most profitable agent in nonincreasing order of desirability, as long as it is feasible to do so.

Our heuristic proceeds in two phases. In the first, greedy phase of the heuristic, the set $\mathcal{J}$ keeps track of the set of jobs that remain to be assigned. During the course of this phase (Steps 1–3), jobs may be identified that can no longer be feasibly assigned. The set $\tilde{\mathcal{J}}$ of such jobs will be handled in the second, improvement phase of the algorithm (Steps 4–7). Throughout the heuristic, b'_i denotes the capacity remaining for agent i ($i = 1, \ldots, m$).

We now formally present our heuristic as follows:

Heuristic – Greedy phase

Step 0. Set $\mathcal{J} = \{1, \ldots, n\}$, $\tilde{\mathcal{J}} = \varnothing$, $b'_i = b_i$ for $i = 1, \ldots, m$, and $\mathcal{I}_j = \{1, \ldots, m\}$. Set $x^{\mathrm{G}}_{ij} = v^{\mathrm{G}}_{ij} = 0$ for $i = 1, \ldots, m$; $j = 1, \ldots, n$.

Step 1. Let

$$i_j \in \arg\max_{i \in \mathcal{I}_j} f_\lambda(i, j) \quad \text{for } j \in \mathcal{J}$$

$$\rho_j = f_\lambda(i_j, j) - \max_{i' \in \mathcal{I}_j \setminus \{i_j\}} f_\lambda(i', j) \quad \text{for } j \in \mathcal{J}.$$

Step 2. Select $\hat{\jmath} \in \arg\max_{j \in \mathcal{J}} \rho_j$, i.e., $\hat{\jmath}$ is the job to be assigned next (to agent $i_{\hat{\jmath}}$). If $a_{i_{\hat{\jmath}}j} + \ell_{i_{\hat{\jmath}}} \leq b'_{i_{\hat{\jmath}}}$ continue to Step 3. Otherwise, $a_{i_{\hat{\jmath}}j} + \ell_{i_{\hat{\jmath}}} > b'_{i_{\hat{\jmath}}}$, which means this assignment is not feasible; let $\mathcal{I}_{\hat{\jmath}} = \{i : a_{i\hat{\jmath}} + \ell_{i\hat{\jmath}} \leq b'_i\}$. If $\mathcal{I}_{\hat{\jmath}} = \varnothing$, set $\tilde{\mathcal{J}} = \mathcal{J}$ and STOP.

Step 3. Set

$$x^{\mathrm{G}}_{i_{\hat{\jmath}}\hat{\jmath}} = 1$$

$$v^{\mathrm{G}}_{i_{\hat{\jmath}}\hat{\jmath}} = \begin{cases} \min\{u_{i_{\hat{\jmath}}\hat{\jmath}}, b'_{i_{\hat{\jmath}}} - a_{i_{\hat{\jmath}}\hat{\jmath}}\} & \text{if } r_{i_{\hat{\jmath}}\hat{\jmath}} > \lambda_{i_{\hat{\jmath}}} \\ \ell_{i_{\hat{\jmath}}\hat{\jmath}} & \text{if } r_{i_{\hat{\jmath}}\hat{\jmath}} \leq \lambda_{i_{\hat{\jmath}}} \end{cases}$$

$$b'_{i_{\hat{\jmath}}} = b'_{i_{\hat{\jmath}}} - (v^{\mathrm{G}}_{i_{\hat{\jmath}}\hat{\jmath}} + a_{i_{\hat{\jmath}}\hat{\jmath}}).$$

Let $\mathcal{J} = \mathcal{J} \setminus \{\hat{\jmath}\}$. If $\mathcal{J} \neq \varnothing$, return to Step 1; otherwise, STOP.

If the greedy phase of the heuristic ends with $\tilde{\mathcal{J}} = \varnothing$, then $(x^{\mathrm{G}}, v^{\mathrm{G}})$ is a feasible solution to the GAPFJ. Otherwise, we will continue the heuristic with an improvement phase. To distinguish the (partial) solution obtained at the end of the greedy phase from the solution delivered by the improvement phase we set $(x^{\mathrm{H}}, v^{\mathrm{H}}) = (x^{\mathrm{G}}, v^{\mathrm{G}})$.

In the improvement phase, we reduce the size of some previously assigned jobs to their corresponding lower bounds to free up capacity that can be used to assign any of the jobs in $\tilde{\mathcal{J}}$. Realizing that the minimum amount of capacity that is required to assign job j to agent i is $a'_{ij} \equiv a_{ij} + \ell_{ij}$, let $\bar{a}'$ be an upper bound on this value among all unassigned jobs. Then, at least

$$\sum_{i=1}^{m} \left\lfloor \frac{b'_i}{\bar{a}'} \right\rfloor \geq \frac{1}{\bar{a}'} \sum_{i=1}^{m} b'_i - m$$

jobs can be accommodated within the remaining agent capacities. Thus, all jobs in $\tilde{\mathcal{J}}$ can be assigned if the agents have cumulative available capacity $\sum_{i=1}^{m} b'_i \geq (|\tilde{\mathcal{J}}| + m)\bar{a}'$. Note that such an assignment can be found by arbitrarily assigning the jobs in $\tilde{\mathcal{J}}$ to any agent that can feasibly accommodate it.

Heuristic – Improvement phase

Step 4. Let $\mathcal{A} = \{(i, j) : x^{\mathrm{H}}_{ij} = 1 \text{ and } v^{\mathrm{H}}_{ij} > \ell_{ij}\}$ and set $\bar{a}' = \max_{(i,j) \in \{1,\ldots,m\} \times \tilde{\mathcal{J}}} (\ell_{ij} + a_{ij})$.

Step 5. Identify a set $\mathcal{A}' \subseteq \mathcal{A}$ with the property that

$$\sum_{i=1}^{m} b'_i + \sum_{(i,j) \in \mathcal{A}'} (v^{\mathrm{H}}_{ij} - \ell_{ij}) \geq (|\tilde{\mathcal{J}}| + m)\bar{a}' \tag{18}$$

and, in addition, $\mathcal{A}'$ is minimal in the sense that removing any element from it causes (18) to be violated. If such a set does not exist, set $\mathcal{A}' = \mathcal{A}$.

Step 6. Set

$$b'_i = b'_i + \sum_{j:(i,j) \in \mathcal{A}'} (v^{\mathrm{H}}_{ij} - \ell_{ij}) \quad \text{for } i = 1, \ldots, m$$

$$v^{\mathrm{H}}_{ij} = \ell_{ij} \quad \text{for } (i, j) \in \mathcal{A}'.$$

Step 7. Attempt to identify a feasible solution to the GAPFJ by (i) assigning and determining jobs and job levels for $j \in \tilde{\mathcal{J}}$, and (ii) increasing job levels for assignments $(i, j) \in \mathcal{A}'$. If this is successful, return the solution as $(x^{\mathrm{H}}, v^{\mathrm{H}})$. Otherwise, the heuristic is unable to find a feasible solution to the GAPFJ.

In the remainder of this section we analyze a basic implementation of the improvement phase of the heuristic where we identify an arbitrary set $\mathcal{A}'$ in Step 5, and try to assign jobs in $\tilde{\mathcal{J}}$ in arbitrary order to agents that can accommodate them in Step 7. In Section 5 we will propose a more sophisticated implementation with guaranteed superior behavior.

The following theorem establishes a close relationship between the solution that is obtained by the greedy phase of the heuristic and a basic optimal solution to (LP′).

Theorem 3.1. *Suppose that (LP′) is feasible and that the optimal (basic) solution to (LP′), say (s^*, t^*), is unique. If we choose λ in the heuristic equal to an associated optimal dual vector λ^*, we have for all non-split jobs $j \notin \mathcal{S}$ that the greedy phase of the heuristic*

(i) *assigns this job to the same agent, say i_j, as* (LP′) *;*
(ii) *executes this job at the same level as* (LP′) *provided* $(i_j, j) \notin \mathcal{Q}$.

Proof. Theorem 2.3(i)–(ii) implies that, in the greedy phase of the heuristic, $\rho_j > 0$ for all $j \notin \mathcal{S}$ and $\rho_j = 0$ for $j \in \mathcal{S}$. Thus, Step 2 guarantees that all non-split jobs are considered before any split jobs as long as the sets $\mathcal{I}_j$ remain unchanged. Claim (i) then immediately follows from the fact that the preferred assignments of the non-split jobs are all feasible. Next, Theorem 2.3(iii) implies that, in Step 3 of the greedy phase of the heuristic, all jobs j for which $(i, j) \notin \mathcal{Q}$ are executed at the same level as in (LP′). (Note that the greedy phase does not necessarily execute jobs for which $(i, j) \in \mathcal{Q}$ at the same level as in (LP′). This follows since, by complementary slackness, $(i, j) \in \mathcal{Q}$ is equivalent to $\mu_j = f_\lambda(i, j) = p_{ij} - \lambda_i a_{ij}$, i.e. the unit revenue $r_{ij} - \lambda_i = 0$, which does not necessarily imply that the solution to (LP′) makes this assignment at its lower bound, as the heuristic does.) □

Theorem 3.1 states that, if we choose λ in the heuristic equal to an optimal dual vector λ^* of (LP′), the greedy phase of the heuristic starts by making the assignments of non-split jobs in the solution to (LP′), with the only possible deviation being the size of jobs that are strictly between their bounds in that solution. This, together with Corollary 2.6, then implies that the total number of jobs that are unassigned in the greedy phase or for which the assignment or size differs from the solution to (LP′) is no larger than the number of agents, m. The improvement phase of the heuristic is aimed at creating sufficient space to allow the assignment of any unassigned jobs. Our goal in the next section is to use this result to derive strategies for choosing λ for two very general stochastic models on the problem parameters such that the heuristic is asymptotically feasible and optimal with probability one as the number of jobs increases. That is, as the number of jobs increases, it is very unlikely that the heuristic is unable to find a feasible solution and, moreover, if it finds a feasible solution, its relative error will decline as the number of jobs increases.

3.2. Average case analysis of the heuristic

This section provides an analysis of the asymptotic behavior of our heuristic under a probabilistic model on the problem parameters that keeps the number of agents, m, fixed and lets the number of jobs, n, approach infinity. We propose a stochastic model for the GAPFJ that is similar to the ones commonly used for the GAP and its extensions (see, e.g., Dyer and Frieze [9], Romeijn and Piersma [15], and Romeijn and Romero Morales [17]).[1] In particular, we assume that each job is characterized by a random vector of parameters $(P_j, R_j, A_j, L_j, D_j)$, where $P_j = (P_{1j}, \ldots, P_{mj})$, $R_j = (R_{1j}, \ldots, R_{mj})$, $A_j = (A_{1j}, \ldots, A_{mj})$, $L_j = (L_{1j}, \ldots, L_{mj})$, and $D_j = (D_{1j}, \ldots, D_{mj})$. Here P_j and R_j are the vectors of fixed profits and unit revenues for job j, respectively. Furthermore, A_j is the vector of fixed resource consumptions for job j, while L_j is the vector of lower bounds and $U_j \equiv L_j + D_j$ is the vector of upper bounds on the size of job j. The vectors $(P_j, R_j, A_j, L_j, D_j)$ are assumed to be i.i.d. on the compact set $[\underline{P}, \bar{P}]^m \times [\underline{R}, \bar{R}]^m \times [\underline{A}, \bar{A}]^m \times [\underline{L}, \bar{L}]^m \times [\underline{D}, \bar{D}]^m$, where the conditional distributions of $(P_j, R_j | A_j, L_j, D_j)$ are absolutely continuous. In addition, we have $\underline{R}, \underline{L} \geq 0$ so that both the size of a job and the unit revenue accrued are non-negative. Furthermore, the difference between the upper and lower bound parameters is taken to be strictly positive, $\underline{D} > 0$, to ensure there is a decision to be made with regard to the size of a job. For convenience, we assume that $\underline{R}\,\underline{L} > -\underline{P}$ so that the total profit associated with *any* feasible assignment is nonnegative. (Note that this assumption is mild since it is automatically satisfied if, for example, $\underline{P} > 0$. Moreover, it can be made without loss of generality since we may add or subtract a constant value from all fixed profit coefficients without impacting the profit ranking of the solutions.) Furthermore, as is common in probabilistic models of this type, we allow for the accommodation of an increasing number of jobs while the number of agents remains constant by letting the capacity of agent i grow linearly with the number of jobs, i.e., we let $b_i = \beta_i n$ where β_i is a positive constant $(i = 1, \ldots, m)$.

Finally, we wish to focus on problem instances that admit a feasible solution. Note that an instance of the GAPFJ has a feasible solution if and only if the associated GAP with all requirements set to their minimum value $a_{ij} + \ell_{ij}$ $(i = 1, \ldots, m;\ j = 1, \ldots, n)$ is feasible. This leads to the following assumption that we will impose on our probabilistic model:

Assumption 3.2.

$$\Delta \equiv \min_{\lambda \geq 0;\ \lambda^\top e = 1} \left(\sum_{i=1}^{m} \lambda_i \beta_i - E\left(\min_{i=1,\ldots,m} \lambda_i (A_{i1} + L_{i1}) \right) \right) > 0.$$

By Romeijn and Piersma [15, Theorem 3.2], this assumption ensures that an instance randomly generated according to our stochastic model is feasible with probability one as $n \to \infty$. Note that this assumption is mild, since they also show that instances generated are asymptotically infeasible with probability one as $n \to \infty$ if $\Delta < 0$.

In the remainder of this section, we will show that, for a suitably chosen strategy for the parameter λ, the heuristic will provide a feasible and optimal solution to the GAPFJ with probability one as $n \to \infty$. In particular, consider an instance of the GAPFJ generated from the probabilistic model described above, and let Z_n^*, Z_n^{LP}, Z_n^{G}, and Z_n^{H} denote its optimal solution value, the value of its LP-relaxation, and the value of the solution obtained by the greedy and the improvement phases of the heuristic, respectively. (Note that these values are random variables, and that we have explicitly recognized that they

[1] As a convention, we will denote parameters and solutions that are random variables by capital letters, while realizations will be denoted by lowercase letters.

are a function of the number of jobs, n.) We then say that the heuristic is *asymptotically feasible and optimal* if

(i) the solution $(X^{\mathrm{H}}, V^{\mathrm{H}})$ produced by the heuristic is asymptotically feasible with probability one;
(ii) $\lim_{n\to\infty} \left(Z_n^* - Z_n^{\mathrm{H}}\right)/Z_n^* = 0$ with probability one.

Since, under our assumptions on the problem parameters, we have that, for any feasible instance of the GAPFJ, $Z_n^* \geq (\underline{R}\,\underline{L} + \underline{P})n$ with $\underline{R}\,\underline{L} + \underline{P} > 0$, the latter is equivalent to

(ii′) $\lim_{n\to\infty} \frac{1}{n}\left(Z_n^* - Z_n^{\mathrm{H}}\right) = 0$ with probability one

which is the characterization of asymptotic optimality that we will use in the remainder of this paper.

We will distinguish between two classes of instances of the GAPFJ. The first class of instances is characterized by job requirements that are *agent-independent*, i.e., $A_{ij} = A_{1j}$, $L_{ij} = L_{1j}$, and $D_{ij} = D_{1j}$ ($i = 1, \ldots, m$; $j = 1, \ldots, n$). For this class, we will show that the heuristic is asymptotically feasible and optimal with probability one if we choose λ equal to an associated optimal dual vector λ^*. The second class of instances follows the general probabilistic model discussed earlier in this section. For this class, we will show that the heuristic is asymptotically feasible and optimal with probability one if we choose λ equal to an optimal dual vector of an appropriately perturbed instance of the GAPFJ.

3.2.1. Agent-independent requirements

Recall that Theorem 3.1 establishes a strong connection between an optimal solution to (LP′) and the solution obtained by the greedy phase of the heuristic if the former is unique with respect to non-split jobs and if we choose λ equal to an associated optimal dual vector λ^* of (LP′). This connection is employed to obtain asymptotic feasibility and optimality. Before we formally prove this result, however, we will first derive two useful preliminary results. The first result provides an intuitive characterization of an optimal solution to (LP′) under the proposed stochastic model. This result is common for models in which parameters are generated from absolutely continuous distributions (see [8,9,16]). It is presented here formally due to its significance in our asymptotic analysis.

Lemma 3.3. *Under our stochastic model, if (LP′) is feasible, its optimal solution is unique with probability one.*

Proof. A non-unique optimal solution to (LP′) exists only if the hyperplane representative of solutions with optimal profit

$$\sum_{i=1}^{m}\sum_{j=1}^{n}\left(p_{ij} + r_{ij}u_{ij}\right)s_{ij}^* + \sum_{i=1}^{m}\sum_{j=1}^{n}\left(p_{ij} + r_{ij}\ell_{ij}\right)t_{ij}^*$$

intersects the feasible region at multiple points. Recall that the unit revenues r_{ij} ($i = 1, \ldots, m$; $j = 1, \ldots, n$) and the fixed profits p_{ij} ($i = 1, \ldots, m$; $j = 1, \ldots, n$) are generated from a joint distribution that, *conditional* on the values of the requirements parameters in the constraints, is absolutely continuous. Thus the probability that data generated by our stochastic model allows for multiple optimal solutions to (LP′) is zero, so that the optimal solution to (LP′) is unique with probability one. □

Lemma 3.4. *When job requirements are agent-independent, the aggregate capacity that is either unused or used for job levels in excess of their lower bound increases linearly in n with probability one as $n \to \infty$, in any feasible solution to (LP).*

Proof. For convenience, denote the effective lower bound on an assignment by $A'_{ij} = A_{ij} + L_{ij}$. Since the job requirements are agent-independent we have that $A'_{ij} = A'_{1j}$ ($i = 1, \ldots, m$). Then, for any assignment vector x that is feasible to the LP-relaxation of the corresponding GAP we have

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{m}\left(b_i - \sum_{j=1}^{n}A'_{ij}x_{ij}\right) &= \sum_{i=1}^{m}\beta_i - \frac{1}{n}\sum_{i=1}^{m}\sum_{j=1}^{n}A'_{1j}x_{ij}\\ &= \sum_{i=1}^{m}\beta_i - \frac{1}{n}\sum_{j=1}^{n}A'_{1j}\left(\sum_{i=1}^{m}x_{ij}\right)\\ &= \sum_{i=1}^{m}\beta_i - \frac{1}{n}\sum_{j=1}^{n}A'_{1j}.\end{aligned}$$

For the case of agent-independent requirements, Romeijn and Piersma [15] show that Assumption 3.2 is equivalent to the condition

$$E(A'_{1j}) < \sum_{i=1}^{m}\beta_i$$

so that, by the Central Limit Theorem,

$$\sum_{i=1}^{m}\beta_i - \frac{1}{n}\sum_{i=1}^{m}\sum_{j=1}^{n}A'_{ij}x_{ij} > 0 \quad \text{with probability one if } n \to \infty$$

for any feasible relaxed assignment vector x. This yields the desired result. □

We are now ready to formally prove our first asymptotic feasibility and optimality result.

Theorem 3.5. *Consider problem instances generated according to our stochastic model with, in addition, agent-independent job requirements. Moreover, choose λ in the heuristic equal to an associated optimal dual vector λ^* of (LP′). Then the heuristic is asymptotically feasible and optimal.*

Proof. Since we are only interested in a probabilistic and asymptotic feasibility guarantee, we may by Lemma 3.3 assume that the solution to (LP′) is unique with respect to non-split jobs. Then Theorem 3.1 says that the greedy phase of the heuristic assigns no more than $|\mathcal{S}|$ jobs to a different agent or to no agent at all, and no more than $|\mathcal{Q}|$ jobs to the same agent but at a different level than the optimal solution to (LP′). Again denoting the effective lower bound of a job by $A' = A + L$ with corresponding upper bound $\bar{A}' = \bar{A} + \bar{L}$, we have that each of the unassigned jobs requires no more than $\bar{A}'$ units of capacity, so that it would suffice if an *aggregate* of $(|\mathcal{S}| + m)\bar{A}'$ units of capacity among all agents were available. Now let $b' = \sum_{i=1}^m b_i' < (|\mathcal{S}| + m)\bar{A}'$ denote the aggregate remaining capacity at the end of the greedy phase of the heuristic, and recall that, by Corollary 2.6, we know that $|\mathcal{S}| + |\mathcal{Q}| \le m$. Sufficient capacity can therefore be made available if, in the improvement phase, we are able to reduce to their lower bounds the level of $\left\lceil \left((|\mathcal{S}| + m)\bar{A}' - b'\right)/\underline{D}\right\rceil$ jobs that the greedy phase of the heuristic assigned at their upper bounds. Since this number is independent of n, Lemma 3.4 implies that, for large enough n, this can indeed be done, implying that the heuristic is asymptotically feasible.

Finally, note that the objective function value of a feasible solution that is obtained by the heuristic satisfies

$$Z^{\mathrm{H}} \ge Z^{\mathrm{LP}} - \left(\left\lceil \left((|\mathcal{S}| + m)\bar{A}' - b'\right)/\underline{D}\right\rceil + |\mathcal{S}| + |\mathcal{Q}|\right)\bar{D}\bar{R} - |\mathcal{S}|(\bar{P} - \underline{P} + (\bar{R} - \underline{R})\bar{L}). \tag{19}$$

In the right hand side of this inequality, $\left((|\mathcal{S}| + m)\bar{A}' - b'\right)\bar{D}\bar{R}$ is the lost revenue from reducing jobs to their lower bounds in order to assure enough aggregate capacity for unassigned jobs in the greedy phase of the heuristic. The term $(|\mathcal{S}| + |\mathcal{Q}|)\bar{D}\bar{R}$ is lost revenue from jobs either not assigned to the same agent as in (LP′), or jobs executed at a different level than (LP′). The quantity $|\mathcal{S}|(\bar{P} - \underline{P} + (\bar{R} - \underline{R})\bar{L})$ accounts for the loss of fixed profit resulting from jobs not assigned to the same agent as in (LP′).

Clearly $Z^{\mathrm{LP}} \le Z_n^* \le Z^{\mathrm{H}}$ implies

$$\lim_{n\to\infty} \frac{1}{n}\left(Z_n^* - Z_n^{\mathrm{H}}\right) \le \lim_{n\to\infty} \frac{1}{n}\left(Z_n^{\mathrm{LP}} - Z_n^{\mathrm{H}}\right).$$

Provided that the heuristic solution is feasible we then have

$$\begin{aligned}
\lim_{n\to\infty} \frac{1}{n}\left(Z_n^{\mathrm{LP}} - Z_n^{\mathrm{H}}\right) &\le \lim_{n\to\infty} \frac{1}{n}\left(\left(\left\lceil \left((|\mathcal{S}| + m)\bar{A}' - b'\right)\underline{D}\right\rceil + |\mathcal{S}| + |\mathcal{Q}|\right)\bar{D}\bar{R} + |\mathcal{S}|\left(\bar{P} - \underline{P} + (\bar{R} - \underline{R})\bar{L}\right)\right) \quad \text{(by (19))}\\
&\le \lim_{n\to\infty} \frac{1}{n}\left(\left(\left\lceil (2m\bar{A}' - b')/\underline{D}\right\rceil + m\right)\bar{D}\bar{R} + m\left(\bar{P} - \underline{P} + (\bar{R} - \underline{R})\bar{L}\right)\right) \quad \text{(by Corollary 2.6)}\\
&= 0.
\end{aligned}$$

Since the heuristic is asymptotically feasible, this implies that the heuristic is asymptotically optimal as well. □

3.2.2. Agent-dependent requirements

The result of Lemma 3.4 can unfortunately not be extended to the general case where job requirements are agent-dependent, preventing us from extending the approach in the previous section to show asymptotic feasibility of the heuristic if we choose λ equal to an optimal dual vector λ^* of (LP′). In general, we therefore take a different approach: we choose λ equal to the optimal dual vector of an instance of (LP′) in which the capacities have been reduced by an appropriately chosen small amount. Note, however, that we will still apply the two phases of the heuristic using the original capacities. By ensuring that the (temporary) capacity reductions are large enough to ensure that the jobs that are fractionally assigned in the LP-relaxation can be assigned feasibly but, at the same time, small enough for the corresponding solution to be close to optimal, we will be able to show that the greedy phase of the heuristic alone is asymptotically feasible and optimal.

More formally, consider an instance of the GAPFJ with n jobs ($n = 1, 2, \ldots$). Then associate with this instance a perturbed instance of (LP′) in which all of the normalized capacities β_i ($i = 1, \ldots, m$) are reduced by δ_n, where

$$\lim_{n\to\infty} \delta_n = 0 \tag{20}$$

$$\lim_{n\to\infty} n\delta_n = \infty \tag{21}$$

and

$$0 < \delta_n \le \delta < \Delta \le \min_{i=1,\ldots,m} \beta_i.$$

We will denote the perturbed problem by $(\mathrm{LP}'(\delta_n))$ and its optimal value by $Z_n^{\mathrm{LP}'}(\delta_n)$. The following preliminary result shows that the optimal values of the original and perturbed problems are very close.

Lemma 3.6. *The optimal values of* (LP′) *and* (LP′(δ_n)) *are close in the sense that, with probability one,*

$$\lim_{n\to\infty} \frac{1}{n} Z_n^{\mathrm{LP}'}(\delta_n) = \lim_{n\to\infty} \frac{1}{n} Z_n^{\mathrm{LP}'}.$$

Proof. See the Appendix. □

The following theorem employs this result to prove that we have a heuristic for the GAPFJ that is asymptotically feasible and optimal.

Theorem 3.7. *Consider problem instances generated according to our general stochastic model. Moreover, choose λ in the heuristic equal to an optimal dual vector to* (LP′(δ_n)). *Then the heuristic is asymptotically feasible and optimal.*

Proof. As in the proof of Theorem 3.5, we may assume that the solution to the perturbed instance of (LP′) is unique with respect to non-split jobs, since we are only interested in a probabilistic and asymptotic feasibility guarantee. Then Theorem 3.1 says that the greedy phase of the heuristic assigns no more than $|\mathcal{S}|$ jobs to a different agent or to no agent at all, and no more than $|\mathcal{Q}|$ jobs to the same agent but at a different level than the optimal solution to (LP′). Since, by Corollary 2.6, we know that $|\mathcal{S}| + |\mathcal{Q}| \le m$, it is easy to see that the additional amount of aggregate capacity over the amount used in the perturbed instance of (LP′) required for these jobs is independent of n. By (21) we can therefore conclude that, with probability one, the greedy phase of the heuristic yields a feasible solution to the GAPFJ. Moreover, the objective function value of a feasible solution that is obtained by the greedy phase of the heuristic satisfies

$$Z^{\mathrm{G}} \ge Z^{\mathrm{LP}'}(\delta_n) - |\mathcal{S}|\,(\bar{R}(\bar{L}+\bar{D}) - \underline{RL}) - |\mathcal{Q}|(\bar{D}\bar{R}) - |\mathcal{S}|(\bar{P} - \underline{P} + (\bar{R} - \underline{R})\bar{L}) \tag{22}$$

so that, in that case,

$$\begin{aligned}
\lim_{n\to\infty} \frac{1}{n}\left(Z_n^* - Z_n^{\mathrm{H}}\right) &\le \lim_{n\to\infty} \frac{1}{n}\left(Z_n^* - Z_n^{\mathrm{G}}\right) \quad (\text{since } Z^{\mathrm{G}} \le Z^{\mathrm{H}})\\
&\le \lim_{n\to\infty} \frac{1}{n}\left(Z_n^{\mathrm{LP}'} - Z_n^{\mathrm{G}}\right) \quad (\text{since } Z^* \le Z^{\mathrm{LP}'})\\
&= \lim_{n\to\infty} \frac{1}{n}\left(Z_n^{\mathrm{LP}'}(\delta_n) - Z_n^{\mathrm{G}}\right) \quad (\text{with probability one as } n\to\infty, \text{by Lemma 3.6})\\
&\le \lim_{n\to\infty} \frac{1}{n}\left(|\mathcal{S}|\,(\bar{R}(\bar{L}+\bar{D}) - \underline{RL}) + |\mathcal{Q}|(\bar{D}\bar{R}) + |\mathcal{S}|\,(\bar{P} - \underline{P} + (\bar{R} - \underline{R})\bar{L})\right) \quad (\text{by (22)})\\
&\le \lim_{n\to\infty} \frac{1}{n}\left((|\mathcal{S}| + |\mathcal{Q}|)\,(\bar{R}(\bar{L}+\bar{D}) - \underline{RL}) + |\mathcal{S}|\,(\bar{P} - \underline{P} + (\bar{R} - \underline{R})\bar{L})\right) \quad (\text{by Corollary 2.6})\\
&\le \lim_{n\to\infty} \frac{m}{n}\left(\bar{R}(\bar{L}+\bar{D}) - \underline{RL} + \bar{P} - \underline{P} + (\bar{R} - \underline{R})\,\bar{L}\right)\\
&= 0.
\end{aligned}$$

Since the heuristic is asymptotically feasible, this implies that the heuristic is asymptotically optimal as well. □

It is interesting to note that Theorem 3.7 actually shows that the greedy phase of the heuristic alone is asymptotically feasible and optimal.

3.3. Model extension

It is interesting to note that our heuristic can still be applied (and retain the associated theoretical properties) if the variable revenue function is convex rather than linear, i.e., if the term $r_{ij}v_{ij}$ in the objective function is replaced by $\tilde{r}_{ij}(v_{ij})$ where $\tilde{r}_{ij}$ is a convex and nondecreasing function. Such a revenue function may be relevant from a practical point of view by realizing that a customer may be willing to pay an increasing amount per unit of product supplied within the acceptable range. In fact, in light of the discussion in Section 2.1, our model could accommodate a situation that exhibits both economies of scale for the supplier (through the fixed profit term) and a larger marginal value to customers who receive additional units of product. To apply our heuristics to this model generalization, we can simply linearize the revenue function by defining $r_{ij} = (\tilde{r}_{ij}(u_{ij}) - \tilde{r}_{ij}(\ell_{ij}))/(u_{ij} - \ell_{ij})$. The asymptotic performance guarantees then follow in a relatively straightforward manner by realizing that the solution to the LP-relaxation overestimates the job revenue for no more than m jobs.

4. Heuristic improvement issues

4.1. Solution improvement

Recall that our statement of the heuristic approach left some flexibility, in particular in Steps 5 and 7 of the improvement phase. Although a basic implementation was sufficient to obtain asymptotic performance guarantees, we will in this section discuss a more sophisticated implementation which is guaranteed to yield superior results in practice.

4.1.1. Improvement phase

First, consider the selection of a set $\mathcal{A}'$ of jobs whose job levels will be reduced to their lower bounds in Step 5. Rather than identifying an arbitrary set that satisfies the properties specified in the heuristic, we sequentially add assignments from $\mathcal{A}$ to $\mathcal{A}'$ in the reverse of the order in which they were assigned in the greedy phase of the heuristic, until the set $\mathcal{A}'$ satisfies the desired properties. Next, rather than attempting to arbitrarily assign jobs in $\tilde{\mathcal{J}}$ to agents (in Step 7) we use the modified greedy algorithm proposed by Romeijn and Romero Morales [16] to solve the following instance of the GAP:

$$\begin{aligned}
&\text{maximize } \sum_{i=1}^{m}\sum_{j\in\tilde{\mathcal{J}}} p'_{ij}x_{ij}\\
&\text{subject to}\\
&\sum_{j\in\tilde{\mathcal{J}}} a'_{ij}x_{ij} \le b'_i \quad i=1,\ldots,m\\
&\sum_{i=1}^{m} x_{ij} = 1 \quad j\in\tilde{\mathcal{J}}\\
&x_{ij}\in\{0,1\} \quad i=1,\ldots,m;\ j\in\tilde{\mathcal{J}}
\end{aligned}\tag{I}$$

where $p'_{ij} = p_{ij} + r_{ij}\ell_{ij}$ and $a'_{ij} = a_{ij} + \ell_{ij}$ ($i = 1, \ldots, m$; $j = 1, \ldots, n$). Denote the optimal solution to (I) by x^{I}. The heuristic solution is updated by setting

$$x^{\mathrm{H}}_{ij} = x^{\mathrm{I}}_{ij} \quad \text{and} \quad v^{\mathrm{H}}_{ij} = x^{\mathrm{I}}_{ij}\ell_{ij} \quad i=1,\ldots,m;\ j\in\tilde{\mathcal{J}}. \tag{23}$$

4.1.2. Post-processing phase

Both the greedy phase of the heuristic in Section 3.1 and the improvement phase described in Section 4.1.1 are designed to provide high quality feasible solutions. However, it may still be possible to improve the quality of the solution. In particular, *given* that a feasible assignment x^{H} has been obtained (in either the greedy or improvement phase), we propose to *optimally* determine corresponding job sizes v^{P} (where the superscript P denotes the solution after the post-processing phase). In fact, if $\mathcal{B}_i = \{j : x^{\mathrm{H}}_{ij} = 1\}$ is the set of jobs assigned to agent i, the optimal job sizes can be determined as follows. First, solve the following continuous knapsack problems for $i = 1, \ldots, m$:

$$\begin{aligned}
&\text{maximize } \sum_{j\in\mathcal{B}_i} r_{ij}w_{ij}\\
&\text{subject to}\\
&\sum_{j\in\mathcal{B}_i} w_{ij} \le b_i - \sum_{j\in\mathcal{B}_i} a_{ij}\\
&0 \le w_{ij} \le u_{ij} - \ell_{ij} \quad j\in\mathcal{B}_i.
\end{aligned}\tag{KP$_i$}$$

Then, if w^* denotes the optimal solutions to these problems, set $v^{\mathrm{P}}_{ij} = w^*_{ij}$. It is easy to see that the problems (KP$_i$) can be solved by choosing the flexible component of the sizes of the jobs in $\mathcal{B}_i$ as large as possible in nonincreasing order of r_{ij} as long as agent capacity allows.

4.2. Capacity perturbation scheme

Recall that, for the general case where requirements are allowed to be agent-dependent, we reduce the normalized agent capacities β_i by some amount δ_n within the framework of Section 3.2.2. Despite the asymptotic feasibility result of Theorem 3.7, it is of course still possible that the heuristic fails to find a feasible solution. In particular, an inappropriate perturbation may lead to infeasibility for one of the following two reasons:

(i) If δ_n is too large, the resulting perturbed capacities may be such that the instance of (LP$'(\delta_n)$) is infeasible so that we cannot perform the greedy phase of the heuristic.

(ii) If δ_n is too small, then we fail to reserve enough capacity to accommodate the jobs that remain unassigned in the greedy phase.

We propose to use this information to iteratively modify the capacity perturbation as needed. Note that, to ensure that no perturbed agent capacities are nonpositive, we should initially have $0 \equiv \underline{\delta} < \delta_n < \bar{\delta} \equiv \min_{i=1\ldots,m} \beta_i$. In case the heuristic is unsuccessful due to (i), we update $\bar{\delta} = \delta_n$ and decrease δ_n. If, on the other hand, it is unsuccessful due to (ii), we update $\underline{\delta} = \delta_n$ and increase δ_n. In either case, we set the new capacity perturbation to

$$\delta_n = \underline{\delta} + \omega \times (\bar{\delta} - \underline{\delta})$$

(where $\omega \in (0, 1)$) and reapply the heuristic.

For instances with agent-independent requirements, no perturbation is required to obtain asymptotic performance guarantees. However, as for instances with agent-dependent requirements, it is of course possible that the heuristic does not find a feasible solution. In that case, we can apply the same iterative scheme, recognizing that we initially have $\delta_n = 0$.

5. Computational results

In this section we test the performance of our heuristics on a large set of randomly generated test problems. Following the theoretical results, we separately consider problems with agent-independent and agent-dependent requirements.

5.1. Experimental design

We use the stochastic model given in Section 3.2 as the basis for generating problem instances. We consider instances with $m = 15$ and $m = 30$ agents, and $n = 5m, 10m, 25m, 50m$, and $100m$ jobs. For each job, we generate the vectors of revenue parameters R_j and P_j independently from uniform distributions on [1, 2] and [30, 50], respectively. The job requirements A_j, L_j and D_j are generated from uniform distributions on [10, 20], [75, 125], and [15, 35], respectively. Note that, for instances with agent-independent requirements, only a single value for each of these parameters is generated, while for instances with agent-dependent requirements, we generate m values (one for each agent) independently from the specified distributions. We focused on instances in which the agent-capacities were identical, that is, we set $b_i = \beta n$ $(i = 1, \dots, m)$. For instances with agent-independent requirements, the value Δ in Assumption 3.2 then reduces to

$$\Delta = \beta - \frac{E(A_{1j} + L_{1j})}{m} \tag{24}$$

and we therefore consider capacities of the form

$$\beta = \tau \cdot \frac{E(A_{1j} + L_{1j})}{m}. \tag{25}$$

For instances with agent-dependent requirements, Romeijn and Romero Morales [17] showed that the value Δ in Assumption 3.2 reduces to

$$\Delta = \beta - \frac{E\left(\min_{i=1,\dots,m} (A_{i1} + L_{i1})\right)}{m} \tag{26}$$

provided that the (lower bounds of the) job requirements $A_{i1} + L_{i1}$ are independent and have an increasing failure rate distribution, as is the case in our experiments. We therefore consider capacities of the form

$$\beta = \tau \cdot \frac{E\left(\min_{i=1,\dots,m} (A_{i1} + L_{i1})\right)}{m}. \tag{27}$$

It is easy to see that in both (25) and (27), $\tau > 1$ is equivalent to Assumption 3.2 being satisfied. Moreover, these choices ensure that the tightness of the instances across different values of m is comparable for a given value of τ. In our experiments, we have considered values of $\tau = 1.1, 1.2$, and 1.3. The former two values correspond to cases where the capacity constraints are expected to have a strong limiting effect on the job sizes that can be accommodated. The third value corresponds to loosely capacitated instances since they yield that the expressions in (24) and (26) are positive even when the ranges D_{i1} are added to the job requirements.

We apply both the greedy phase and the improvement phase to each problem instance. We use the iterative capacity perturbation scheme described in Section 4.2 to improve the ability of the heuristic to find feasible solutions for smaller values of n; here we simply set the update parameter equal to $\omega = \frac{1}{2}$. Moreover, unless otherwise noted, the post-processing phase described in Section 4.1.2 is applied as well. Finally, for a meaningful assessment of the heuristic performance we run CPLEX until a solution is obtained whose objective function value is at least as good as the one found by the heuristic or until 15 min of CPU time have been used. For each problem class, we present average results of 25 randomly generated instances. All experiments were performed on a PC with a 3.40 GHz Pentium IV processor and 2 GB of RAM, and all mixed-integer and linear programming problems were solved using CPLEX 10.1. The tables report

(i) the number of instances in which the heuristic found a feasible solution,
(ii) an upper bound on the relative solution error as measured by

$$\text{error} = \frac{z^{\mathrm{LP}'} - z^{\mathrm{H}}}{z^{\mathrm{H}}} \times 100\%,$$

(where, since the error is meaningless if no feasible solution is found, the average error is determined with respect to the instances for which the heuristic is able to find a feasible solution only),
(iii) the average CPU time used by the heuristic, for both the greedy phase and the greedy phase followed by the improvement phase over all iterations,
(iv) the average number of capacity perturbation iterations performed,
(v) the CPU time required by CPLEX, and
(vi) the number of instances for which CPLEX failed to obtain a solution of the desired quality within the allotted time (indicated by a superscript).

Table 1
Agent-independent requirements: $m = 15, \tau = 1.1$

n	Greedy phase			Improvement phase			it.	CPLEX
	Feas.	Error (%)	Time (s)	Feas.	Error (%)	Time (s)		Time (s)
75	0	–	0.04	24	1.12	0.33	5.84	0.24
150	0	–	0.09	25	0.38	0.13	1	0.29
375	0	–	0.20	25	0.09	0.29	1	0.85
750	0	–	0.35	25	0.05	0.51	1	1.74
1500	0	–	0.72	25	0.03	1.03	1	4.18

Table 2
Agent-independent requirements: $m = 15, \tau = 1.2$

n	Greedy phase			Improvement phase			it.	CPLEX
	Feas.	Error (%)	Time (s)	Feas.	Error (%)	Time (s)		Time (s)
75	0	–	0.04	25	3.20	0.07	1	0.20
150	0	–	0.07	25	1.82	0.11	1	0.41
375	0	–	0.18	25	0.47	0.27	1	0.94
750	0	–	0.33	25	0.11	0.49	1	1.93
1500	0	–	0.68	25	0.02	0.98	1	4.85

Table 3
Agent-independent requirements: $m = 15$ $\tau = 1.3$

n	Greedy phase			Improvement phase			it.	CPLEX
	Feas.	Error (%)	Time (s)	Feas.	Error (%)	Time (s)		Time (s)
75	25	0.79	0.06	25	0.79	0.06	1	1.30
150	25	0.24	0.10	25	0.24	0.10	1	2.00
375	25	0.06	0.24	25	0.06	0.24	1	4.93
750	25	0.02	0.45	25	0.02	0.45	1	6.28
1500	25	0.00	0.91	25	0.00	0.91	1	10.82

Table 4
Agent-independent requirements: $m = 30, \tau = 1.1$

n	Greedy phase			Improvement phase			it.	CPLEX
	Feas.	Error (%)	Time (s)	Feas.	Error (%)	Time (s)		Time (s)
150	0	–	0.18	25	1.05	0.35	1.56	1.74
300	0	–	0.27	25	0.36	0.39	1	1.51
750	0	–	0.68	25	0.09	0.95	1	4.26
1500	0	–	1.40	25	0.04	1.90	1	6.76
3000	0	–	2.95	25	0.02	3.95	1	13.34

5.2. Agent-independent requirements

Tables 1–6 summarize the results obtained with our heuristics when applied to instances generated according to the model described in the previous section with agent-independent requirements. Recall that Theorem 3.5 says that the heuristic formed by the greedy and improvement phases is asymptotically feasible and optimal. However, a similar guarantee cannot be given for the greedy phase alone; in particular, applying the greedy phase alone does not guarantee asymptotic feasibility. The computational results confirm this: for the two classes in which the capacity constraints are tightest ($\tau = 1.1$ and $\tau = 1.2$), the greedy phase is not able to find a feasible solution in any of the instances generated. In contrast, when the greedy phase is considered in conjunction with the improvement phase a feasible solution is obtained for almost all instances, with the exception of instances with $n = 5m$ and $\tau = 1.1$ (for both $m = 15$ and $m = 30$). However, our results show that, for such instances, performing the iterative capacity perturbation scheme of Section 4.2 yields a feasible solution in all but a single instance (with $m = 15$, $n = 75$, and $\tau = 1.1$). Note that, when $\tau = 1.3$, the greedy phase alone is able to find a feasible solution for all instances. This can partly be explained by the fact that, for large n, it can be expected that all jobs can be performed at their upper bounds, making the improvement phase unnecessary even from a theoretical point of view. It is noteworthy, however, that the greedy phase alone still performs very well for instances with smaller values of n when $\tau = 1.3$, despite the lack of any theoretical feasibility guarantee.

The results clearly show that the average error approaches zero as the number of jobs increases. It is interesting to note that, for both values of m, the average errors are largest when $\tau = 1.2$. This behavior is a consequence of the nature of the improvement phase, in which the heuristic creates capacity for unassigned jobs by decreasing the size of already assigned jobs to their lower bounds. When $\tau = 1.2$, jobs can generally be performed at higher levels than when $\tau = 1.1$, so that the net effect of the improvement phase on solution quality is understandably larger for $\tau = 1.2$ than for $\tau = 1.1$ However,

Table 5
Agent-independent requirements: $m = 30$, $\tau = 1.2$

n	Greedy phase			Improvement phase			it.	CPLEX
	Feas.	Error (%)	Time (s)	Feas.	Error (%)	Time (s)		Time (s)
150	0	–	0.15	25	3.56	0.21	1	0.83
300	0	–	0.26	25	1.60	0.38	1	2.11
750	0	–	0.66	25	0.38	0.92	1	5.82
1500	0	–	1.34	25	0.12	1.83	1	9.79
3000	0	–	2.73	25	0.02	3.71	1	19.44

Table 6
Agent-independent requirements: $m = 30$, $\tau = 1.3$

n	Greedy phase			Improvement phase			it.	CPLEX
	Feas.	Error (%)	Time (s)	Feas.	Error (%)	Time (s)		Time (s)
150	25	0.67	0.18	25	0.67	0.18	1	8.21
300	25	0.29	0.34	25	0.29	0.34	1	17.62
750	25	0.04	0.84	25	0.04	0.84	1	28.24
1500	25	0.01	1.63	25	0.01	1.63	1	50.19
3000	25	0.00	3.25	25	0.00	3.25	1	163.96

Table 7
Agent-dependent requirements: $m = 15$, $\tau = 1.1$

n	Greedy phase			Improvement phase			it.	CPLEX
	Feas.	Error (%)	Time (s)	Feas.	Error (%)	Time (s)		Time (s)
75	10	5.06	0.08	25	9.32	0.39	4.84	3.04
150	6	2.65	0.18	25	9.32	0.92	4.88	$144.55^{(3)}$
375	25	1.28	0.36	25	1.28	0.36	1	$-^{(25)}$
750	25	0.80	0.78	25	0.80	0.78	1	$-^{(25)}$
1500	25	0.52	1.89	25	0.52	1.89	1	$-^{(25)}$

this pattern does not continue for $\tau = 1.3$ since, as we concluded above, the improvement phase is not required for these instances. We also remark that the solution errors depend mainly on the ratio n/m between the number of jobs and the number of agents.

The heuristic is computationally very efficient, on average taking only slightly more than 1 s of CPU time when $m = 15$ and about 4 s of CPU time when $m = 30$. For smaller problem instances, CPLEX is able to find solutions of the same quality very rapidly as well. However, for larger instances and as the capacity increases CPLEX is up to approximately 10 times slower for $m = 15$ and up to approximately 50 times slower for $m = 30$. Our heuristic is therefore especially promising for large instances and cases where the GAPFJ needs to be solved repeatedly, for example under different scenarios or when it is a subproblem in a more complex strategic optimization problem. Perhaps surprisingly, despite the fact that an instance of (LP′) has to be solved, the time required by the heuristic increases only modestly (approximately linearly) in the size of the problem.

5.3. Agent-dependent requirements

Recall that, for instances with agent-dependent requirements, the heuristic employs the (dual) solution to the LP-relaxation of an instance of the GAPFJ in which the normalized capacities β are reduced by a quantity δ_n satisfying (20) and (21) to ensure asymptotic feasibility and optimality. To ensure that no perturbed agent capacities are nonpositive for any value of n, we propose to choose

$$\delta_n = \alpha \times \frac{\beta}{\sqrt{n}} \tag{28}$$

where $0 < \alpha < 1$, and where the magnitude of α represents a tradeoff between feasibility and solution quality. In our computational experiments we simply use $\alpha = \frac{1}{2}$.

Tables 7–12 summarize the results obtained with our heuristics when applied to instances generated according to the model described in Section 5.1 with agent-dependent requirements. Recall that Theorem 3.7 says that the greedy phase of the heuristic is asymptotically feasible and optimal. Although the greedy phase alone fails to find a feasible solution to a substantial number of problem instances for smaller ratios n/m, the greedy phase is uniformly successful for larger ratios. Moreover, the pattern of average errors after the greedy phase alone shows (conditionally on finding a feasible solution) a decreasing trend, illustrating the theoretical asymptotic optimality result.

Table 8
Agent-dependent requirements: $m = 15$, $\tau = 1.2$

n	Greedy phase			Improvement phase			it.	CPLEX
	Feas.	Error (%)	Time (s)	Feas.	Error (%)	Time (s)		Time (s)
75	4	1.55	0.07	25	7.17	0.34	4.36	1.56
150	15	0.61	0.17	25	2.68	0.48	2.60	50.30
375	25	0.21	0.31	25	0.21	0.31	1	$341.90^{(10)}$
750	25	0.10	0.65	25	0.10	0.65	1	$478.38^{(20)}$
1500	25	0.05	1.41	25	0.05	1.41	1	$333.57^{(23)}$

Table 9
Agent-dependent requirements: $m = 15$, $\tau = 1.3$

n	Greedy phase			Improvement phase			it.	CPLEX
	Feas.	Error (%)	Time (s)	Feas.	Error (%)	Time (s)		Time (s)
75	9	0.97	0.07	25	8.90	0.22	2.68	0.38
150	19	0.29	0.16	25	0.66	0.20	1.12	0.47
375	25	0.02	0.29	25	0.02	0.29	1	0.81
750	25	0.01	0.57	25	0.01	0.57	1	1.72
1500	25	0.00	1.17	25	0.00	1.17	1	4.50

Table 10
Agent-dependent requirements: $m = 30$, $\tau = 1.1$

n	Greedy phase			Improvement phase			it.	CPLEX
	Feas.	Error (%)	Time (s)	Feas.	Error (%)	Time (s)		Time (s)
150	1	2.25	0.40	25	13.38	3.10	7.72	139.74
300	0	–	0.50	25	8.03	5.37	7.00	$440.51^{(23)}$
750	14	0.79	1.76	25	2.60	10.83	3.64	$-^{(25)}$
1500	25	0.48	4.86	25	0.48	4.86	1	$-^{(25)}$
3000	25	0.31	14.11	25	0.31	14.11	1	$-^{(25)}$

Table 11
Agent-dependent requirements: $m = 30$, $\tau = 1.2$

n	Greedy phase			Improvement phase			it.	CPLEX
	Feas.	Error (%)	Time (s)	Feas.	Error (%)	Time (s)		Time (s)
150	0	–	0.34	25	5.01	2.21	6	55.25
300	0	–	0.42	25	3.95	4.21	6	$374.52^{(24)}$
750	20	0.22	1.44	25	0.67	4.68	2	$-^{(25)}$
1500	25	0.09	3.54	25	0.09	3.54	1	$-^{(25)}$
3000	25	0.06	8.93	25	0.06	8.93	1	$-^{(25)}$

Table 12
Agent-dependent requirements: $m = 30$, $\tau = 1.3$

n	Greedy phase			Improvement phase			it.	CPLEX
	Feas.	Error (%)	Time (s)	Feas.	Error (%)	Time (s)		Time (s)
150	0	–	0.32	25	11.48	1.69	4.84	2.97
300	4	0.35	0.38	25	1.91	1.15	1.96	4.02
750	21	0.04	1.20	25	0.04	1.25	1	6.16
1500	25	0.01	2.75	25	0.01	2.75	1	8.53
3000	25	0.00	6.38	25	0.00	6.38	1	13.70

The contribution to feasibility of the iterative capacity perturbation scheme is apparent for instances with $n \leq 25m$: a feasible solution was obtained in all instances. It should not be surprising that this results in an increase in average error, since the instances for which a single iteration of the greedy phase is not able to find a feasible solution are clearly the harder ones. However, the pattern of average errors still exhibits a strongly decreasing trend, again clearly illustrating the theoretical asymptotic optimality result.

The average CPU time required by the heuristic is substantially larger for instances with agent-dependent requirements than for instances with agent-independent requirements, particularly when $m = 30$. CPLEX is able to solve instances with loose capacities ($\tau = 1.3$) in about twice the time required by the heuristic (and even in only about 50% more time for the smallest instances). The instances with tighter capacities clearly illustrate the strength of the heuristic. When $\tau = 1.1$ and

Table 13
Post-processing effect on heuristic with greedy and improvement phase; agent-independent requirements: $\tau = 1.2$

n	$m = 15$ Error (%)		$m = 30$ Error (%)	
	Before	After	Before	After
$5m$	19.42	3.20	19.47	3.56
$10m$	10.62	1.82	10.90	1.60
$25m$	4.05	0.47	4.15	0.38
$50m$	2.00	0.11	2.03	0.12
$100m$	0.98	0.02	1.00	0.02

Table 14
Post-processing effect; agent-dependent requirements: $\tau = 1.2$

n	$m = 15$				$m = 30$			
	Greedy phase Error (%)		Improvement phase Error (%)		Greedy phase Error (%)		Improvement phase Error (%)	
	Before	After	Before	After	Before	After	Before	After
$5m$	7.92	1.55	21.30	7.17	–	–	18.80	5.01
$10m$	4.91	0.61	10.48	2.68	–	–	15.84	3.95
$25m$	2.78	0.21	2.78	0.21	2.07	0.22	4.40	0.67
$50m$	1.91	0.10	1.91	0.10	1.36	0.09	1.36	0.09
$100m$	1.32	0.05	1.32	0.05	0.93	0.06	0.93	0.06

$n \geq 25m$, CPLEX was not able to find a feasible solution within 15 min of CPU time for any instance, and was successful for only 2 instances with $n = 10m$ and $m = 30$. When $\tau = 1.2$ we see a similar behavior for $n \geq 10m$ and $m = 30$. For $m = 15$ and $n = 10m$ the time required by CPLEX exceeds that of the heuristic by a factor of 100, while for $n \geq 25m$ CPLEX is again not successful in a substantial number of problem instances. (Note that the computation times for CPLEX are averaged over only those instances in which it was successful and therefore do not include the instances for which CPLEX was unsuccessful within 15 min. Moreover, for each instance in which the CPU time limit expired, CPLEX had not yet found a feasible solution.)

5.4. *Effect of post-processing phase*

The asymptotic feasibility and optimality guarantees of the heuristic for instances with agent-independent as well as instances with agent-dependent requirements hold even in the absence of the post-processing procedure described in Section 4.1.2. However, as mentioned above, the results presented thus far pertain to solutions that have been improved by this post-processing phase. Therefore, we will in this section study the effect of the post-processing phase on the quality of the solutions, illustrating simultaneously the asymptotic performance guarantees without the post-processing phase as well as the practical importance of applying this phase.

Tables 13 and 14 summarize these results. For brevity, we have only focused on instances with intermediate capacities ($\tau = 1.2$); however, the results for other values of τ are qualitatively similar. For the case of agent-independent requirements we have omitted the results of the greedy phase alone, since this phase was never able to find a feasible solution and thus the post-processing phase is irrelevant. In both tables, the columns labeled "before" contain the solution errors without post-processing, while the columns labeled "after" contain the solution errors with post-processing. We see that, in all cases and for both types of problem instances, the results without the post-processing phase are consistent with the asymptotic optimality guarantees. However, we also see that the post-processing phase substantially reduces solution error, particularly for problem instances with a small ratio of n/m.

6. Summary and concluding remarks

In this paper we considered the GAPFJ, which generalizes the classical GAP. Our extension applies to situations in which, along with the assignment of jobs to agents, a flexible degree of resource consumption must be determined for each of these assignments. To solve the GAPFJ, we propose a class of heuristics motivated by attractive properties of the optimal solution of the LP-relaxation to the GAPFJ and its corresponding dual. For two classes of job requirements we show that an implementation of the heuristic exists that is asymptotically feasible and optimal with probability one under a very broad stochastic model on the problem parameters. Our computational study demonstrates that the heuristic performs very well, particularly for large ratios of the number of jobs to the number of agents. When additional improvement strategies that we propose in this paper are also considered, the heuristic is successful on instances with smaller ratios as well. We observe that the time required to obtain solutions of comparable quality is considerably less for our heuristic than for the commercial solver CPLEX. The fact that our heuristic obtains quality solutions so quickly is encouraging for further research directions. Specifically, we believe that the heuristic may be very valuable when solving more general related optimization problems for which the GAPFJ arises as a subproblem that needs to be solved repeatedly.

Acknowledgement

The work of the second and third authors was supported by the National Science Foundation under Grant No. DMI-0355533.

Appendix

Lemma 3.6. *The optimal values of* (LP′) *and* $(LP'(\delta_n))$ *are close in the sense that, with probability one,*

$$\lim_{n\to\infty}\frac{1}{n}Z_n^{\mathrm{LP}}(\delta_n)=\lim_{n\to\infty}\frac{1}{n}Z_n^{\mathrm{LP}}.$$

Proof. The normalized optimal value of $(\mathrm{LP}'(\delta_n))$ can be expressed as

$$\frac{1}{n}Z_n^{\mathrm{LP}}(\delta_n)=\min_{\lambda\geq 0}\Phi_n(\lambda;\delta_n)$$

where

$$\begin{aligned}\Phi_n(\lambda;\delta_n)&=\frac{1}{n}\sum_{j=1}^{n}\max_{i=1,\ldots,m}f_\lambda(i,j)+\sum_{i=1}^{m}\lambda_i(\beta_i-\delta_n)\\&=\Phi_n(\lambda;0)-\left(\sum_{i=1}^{m}\lambda_i\right)\delta_n.\end{aligned}\tag{29}$$

We will show that we may restrict ourselves to vectors λ in a compact set. First, note that

$$\min_{\lambda\geq 0}\Phi_n(\lambda;\delta)\leq\Phi_n(0;\delta)\leq\bar{R}(\bar{L}+\bar{D})+\bar{P}.$$

Furthermore, for any λ we have

$$\begin{aligned}\Phi_n(\lambda;\delta)&=\frac{1}{n}\sum_{j=1}^{n}\max_{i=1,\ldots,m}\left((r_{ij}-\lambda_i)\ell_{ij}+(r_{ij}-\lambda_i)^+(u_{ij}-\ell_{ij})+p_{ij}-\lambda_i a_{ij}\right)+\sum_{i=1}^{m}\lambda_i\beta_i-\delta\cdot\sum_{i=1}^{m}\lambda_i\\&\geq\frac{1}{n}\sum_{j=1}^{n}\max_{i=1,\ldots,m}\left(p_{ij}-\lambda_i a_{ij}+(r_{ij}-\lambda_i)\ell_{ij}\right)+\sum_{i=1}^{m}\lambda_i\beta_i-\delta\cdot\sum_{i=1}^{m}\lambda_i\\&\geq\underline{R}\,\underline{L}+\underline{P}+\left(\sum_{i=1}^{m}\lambda_i\beta_i-\frac{1}{n}\sum_{j=1}^{n}\min_{i=1,\ldots,m}\lambda_i(a_{ij}+\ell_{ij})\right)-\delta\cdot\sum_{i=1}^{m}\lambda_i\\&\geq\underline{R}\,\underline{L}+\underline{P}+\min_{\lambda'\geq 0,\,\lambda'^\top e=1}\left(\sum_{i=1}^{m}\lambda'_i\beta_i-\frac{1}{n}\sum_{j=1}^{n}\min_{i=1,\ldots,m}\lambda'_i(a_{ij}+\ell_{ij})\right)\cdot\sum_{i=1}^{m}\lambda_i-\delta\cdot\sum_{i=1}^{m}\lambda_i\\&\to\underline{R}\,\underline{L}+\underline{P}+(\Delta-\delta)\cdot\sum_{i=1}^{m}\lambda_i\end{aligned}$$

with probability one as $n\to\infty$, by Romeijn and Piersma [15, Theorem 3.1]. Since the value $Z_n^{\mathrm{LP}}(\delta_n)$ is nonnegative, the function $\Phi_n(\lambda;\delta)$ thus attains its minimum on the compact set $\Lambda=\left\{\lambda\geq 0:\sum_{i=1}^{m}\lambda_i\leq\Gamma\right\}$ (where $\Gamma=\left(\bar{R}(\bar{L}+\bar{D})+\bar{P}-\underline{R}\,\underline{L}-\underline{P}\right)/(\Delta-\delta)$) with probability one as $n\to\infty$.

Now note that $\Phi_n(\lambda;\delta_n)\geq\Phi_n(\lambda;\delta)$. By the convexity of $\Phi_n(\lambda;0)$ in λ and Eq. (29) it then follows that the function $\Phi_n(\lambda;\delta_n)$ also attains its minimum on Λ with probability one as $n\to\infty$. This means that

$$\Phi_n(\lambda;\delta_n)\geq\Phi_n(\lambda;0)-\Gamma\delta_n\quad\text{with probability one as } n\to\infty$$

so that

$$\begin{aligned}\frac{1}{n}Z_n^{\mathrm{LP}}&\geq\min_{\lambda\geq 0}\Phi_n(\lambda;\delta_n)\\&\geq\min_{\lambda\geq 0}\Phi_n(\lambda;0)-\Gamma\delta_n\\&=\frac{1}{n}Z_n^{\mathrm{LP}}-\Gamma\delta_n\quad\text{with probability one as } n\to\infty.\end{aligned}$$

The desired result now follows by using (20). □

References

[1] R.C. Baker, T.L. Urban, A deterministic inventory model with an inventory-level dependent demand, Naval Research Logistics 35 (1988) 99–116.
[2] R.C. Baker, T.L. Urban, A deterministic inventory system with an inventory-level-dependent demand rate, Journal of the Operational Research Society 39 (9) (1988) 823–831.
[3] A. Balakrishnan, J. Geunes, Production planning with flexible product specifications: An application to specialty steel manufacturing, Operations Research 51 (1) (2003) 94–112.
[4] A. Balakrishnan, M.S. Pangburn, E. Stavrulaki, Stack them high, let 'em fly: Lot sizing policies when inventories stimulate demand, Management Science 50 (5) (2004) 630–644.
[5] S. Biller, L.M.A. Chan, D. Simchi-Levi, J. Swann, Dynamic pricing and the direct-to-customer model in the automotive industry, Electronic Commerce Journal 122 (2005) 584–601.
[6] Z.-L. Chen, N.G. Hall, The coordination of pricing and scheduling decisions, Technical Report, 2007.
[7] S. Deng, C.A. Yano, Joint production and pricing decisions with setup costs and capacity constraints, Management Science 52 (5) (2006) 741–756.
[8] M. Dyer, A. Frieze, Probabilistic analysis of the multidimensional knapsack problem, Mathematics of Operations Research 14 (1) (1989) 162–176.
[9] M. Dyer, A. Frieze, Probabilistic analysis of the generalised assignment problem, Mathematical Programming 55 (1992) 169–181.
[10] Y. Gerchak, Y. Wang, Periodic-review inventory models with inventory-level-dependent demand, Naval Research Logistics 41 (1994) 99–116.
[11] J. Geunes, H.E. Romeijn, K. Taaffe, Requirements planning with pricing and order selection flexibility, Operations Research 54 (2) (2005) 394–401.
[12] S.M. Gilbert, Coordination of pricing and multi-period procurement for constant priced goods, European Journal of Operational Research 114 (2) (1999) 330–337.
[13] H. Kunreuther, L. Schrage, Joint pricing and inventory decisions for constant priced items, Management Science 19 (7) (1973) 732–738.
[14] S. Martello, P. Toth, An algorithm for the generalized assignment problem, in: J.P. Brans (Ed.), Operational Research, North-Holland, 1981, pp. 589–603.
[15] H.E. Romeijn, N. Piersma, A probabilistic feasibility and value analysis of the generalized assignment problem, Journal of Combinatorial Optimization 3 (2000) 325–355.
[16] H.E. Romeijn, D. Romero Morales, A class of greedy algorithms for the generalized assignment problem, Discrete Applied Mathematics 103 (2000) 209–235.
[17] H.E. Romeijn, D. Romero Morales, Generating experimental data for the generalized assignment problem, Operations Research 49 (2001) 866–878.
[18] K.T. Talluri, G.J. Van Ryzin, The Theory and Practice of Revenue Management, Springer, 2004.
[19] J. Thomas, Price-production decisions with deterministic demand, Management Science 16 (11) (1970) 747–750.
[20] W. van den Heuvel, A.P.M. Wagelmans, A polynomial time algorithm for a deterministic joint pricing and inventory model, European Journal of Operational Research 170 (2) (2006) 463–480.
[21] Y. Wang, Y. Gerchak, Supply chain coordination when demand is shelf-spacedependent, Manufacturing & Service Operations Management 3 (2001) 82–87.