DISCRETE APPLIED MATHEMATICS

ELSEVIER Discrete Applied Mathematics 103 (2000) 209–235

# A class of greedy algorithms for the generalized assignment problem

H. Edwin Romeijn[a,*], Dolores Romero Morales[b]

[a]*Department of Industrial and Systems Engineering, University of Florida, 303 Weil Hall, P.O. Box 116595, Gainesville, FL 32611-6595, USA*

[b]*Rotterdam School of Management, Erasmus University of Rotterdam, P.O. Box 1738, NL-3000 DR Rotterdam, The Netherlands*

Received 23 October 1997; revised 24 August 1999; accepted 13 September 1999

**Abstract**

The Generalized Assignment Problem (GAP) is the problem of finding the minimal cost assignment of jobs to machines such that each job is assigned to exactly one machine, subject to capacity restrictions on the machines. We propose a class of greedy algorithms for the GAP. A family of weight functions is defined to measure a pseudo-cost of assigning a job to a machine. This weight function in turn is used to measure the desirability of assigning each job to each of the machines. The greedy algorithm then schedules jobs according to a decreasing order of desirability. A relationship with the partial solution given by the LP-relaxation of the GAP is found, and we derive conditions under which the algorithm is asymptotically optimal in a probabilistic sense. © 2000 Elsevier Science B.V. All rights reserved.

*Keywords*: Generalized Assignment Problem; Greedy heuristic; Asymptotic feasibility; Asymptotic optimality

## 1. Introduction

In the *Generalized Assignment Problem* (*GAP*) there are jobs which need to be processed and machines which can process these jobs. Each machine has a given capacity, and the processing time of each job depends on the machine that processes that job. The GAP is then the problem of assigning each job to exactly one machine, so that the total cost of processing the jobs is minimized and each machine does not exceed its available capacity. The problem can be formulated as an integer linear programming problem as follows:

$$\min \quad \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

* Corresponding author.

*E-mail addresses:* romeijn@ise.ufl.edu (H.E. Romeijn), d.romero@fac.fbk.eur.nl (D.R. Morales)


PII: S0166-218X(99)00224-3

s.t.

$$\sum_{j=1}^{n} a_{ij}x_{ij} \leqslant b_i, \quad i=1,\dots,m,$$

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

$$x_{ij} \in \{0,1\}, \quad i=1,\dots,m;\ j=1,\dots,n,$$

where the cost coefficients $c_{ij}$, the requirement coefficients $a_{ij}$, and the capacity parameters $b_i$ are all non-negative.

The GAP was defined by Ross and Soland [16], and is inspired by real-life problems such as assigning jobs to computer networks (see [1]), fixed charge plant location where customer requirements must be satisfied by a single plant (see [8]), and the Single Sourcing Problem (see [4]). Other applications that have been studied are routing problems (see [6]), and the $p$-median problem (see [17]). Various approaches can be found to solve this problem, some of which were summarized by Cattrysse and Van Wassenhove [3]. Due to its interest, this problem has been studied extensively from an algorithmic point of view. Nevertheless, these algorithms suffer from the *$\mathscr{NP}$-Hardness* of the GAP (see [7]). This means that computational requirements for solving this problem tend to increase very quickly with only a modest increase in the size of the problem. Actually, the GAP is *$\mathscr{NP}$-Hard* in the strong sense since the decision problem associated with the feasibility of the GAP is an *$\mathscr{NP}$-Complete* problem (see [11]).

Stochastic models for the GAP have been proposed by Dyer and Frieze [5], and Romeijn and Piersma [15]. In the latter paper a probabilistic analysis of the optimal solution of the GAP under these models was performed, studying the asymptotic behavior of the optimal solution value as the number of jobs $n$ (the parameter measuring the size of the problem) goes to infinity. Furthermore, a tight condition on the stochastic model under which the GAP is feasible with probability one when $n$ goes to infinity is derived.

In this paper we develop a class of greedy algorithms for the GAP, using a similar approach as for the Multi-Knapsack Problem (see [12,13]). As for the probabilistic analysis of the GAP, the fact that not all instances of the problem are feasible creates significant challenges.

The greedy algorithms proceed as follows: given a vector of multipliers (each corresponding to a machine), a weight function is defined to measure the pseudo-cost of assigning a job to a machine. This weight function is used to assign a desirability measure to each possible assignment of a job to a machine. The jobs are then assigned in decreasing order of the desirabilities. A similar idea was introduced by Martello and Toth [10], and in fact some of the weight functions proposed by them are elements of our family of weight functions.

In Section 2 of the paper we analyze the LP-relaxation of the GAP and its dual. In Section 3 we introduce the class of greedy algorithms, and show a relationship with the partial solution obtained by the LP-relaxation of the GAP when the multipliers are chosen equal to the optimal dual variables corresponding to the capacity constraints. We also give a geometrical interpretation of the algorithm, and show that, for a fixed number of machines, the best set of multipliers can be found in polynomial time. In Section 4 we show that, for large problem instances, the heuristic finds a feasible and optimal solution with probability one if the set of multipliers is chosen equal to the optimal dual variables corresponding to the capacity constraints. Moreover, conditions are given under which there exists a unique vector of multipliers, only depending on the number of machines and the probabilistic model for the parameters of the problem, so that the corresponding heuristic is asymptotically feasible and optimal. Finally, Section 5 contains a short summary.

## 2. LP-relaxation

The linear programming relaxation (LPR) of the GAP reads

$$\begin{aligned} \min \quad & \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \\ \text{s.t.} \quad & \\ & \sum_{j=1}^{n} a_{ij}x_{ij} \leqslant b_i, \quad i=1,\ldots,m, \qquad (1)\\ & \sum_{i=1}^{m} x_{ij} = 1, \quad j=1,\ldots,n, \\ & x_{ij} \geqslant 0, \quad i=1,\ldots,m,\ j=1,\ldots,n. \end{aligned} \tag{LPR}$$

Throughout this section we will assume that (LPR) has a feasible solution.

If the optimal solution for (LPR), say $x^{\text{LPR}}$, does not contain any fractional variable, then this clearly is the optimal solution for the GAP as well. In general, however, this will not be the case. We call a job $j$ a *non-split job* of (LPR) if there exists an index $i$ such that $x_{ij}^{\text{LPR}}=1$. The remaining jobs, called *split jobs*, are assigned to more than one machine. In the following, we show a relationship between the number of split jobs, the number of split assignments, and the number of machines used to full capacity. Let $F$ be the set of fractional variables in the optimal solution of (LPR), $x^{\text{LPR}}$, $S$ the set of split jobs in $x^{\text{LPR}}$, and $M$ the set of machines used to full capacity in $x^{\text{LPR}}$, i.e.

$$F = \{(i,j):\ 0 < x_{ij}^{\text{LPR}} < 1\},$$

$$S = \{j:\ \exists (i,j) \in F\},$$

$$M = \left\{ i:\ \sum_{j=1}^{n} a_{ij}x_{ij}^{\text{LPR}} = b_i \right\}.$$

**Lemma 2.1.** *If* (LPR) *is non-degenerate, then for the optimal solution* $x^{\mathrm{LPR}}$ *of* (LPR) *we have*

$$|F| = |M| + |S|.$$

**Proof.** Denote the surplus variables corresponding to the capacity constraints (1) by $s_i$ $(i = 1,\dots,m)$. (LPR) can then be reformulated as

$$\min \quad \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$$

$$\text{s.t.}$$

$$\sum_{j=1}^{n} a_{ij}x_{ij} + s_i = b_i, \quad i = 1,\dots,m,$$

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

$$x_{ij} \geqslant 0, \quad i = 1,\dots,m,\ j = 1,\dots,n,$$

$$s_i \geqslant 0, \quad i = 1,\dots,m.$$

Let $(x^{\mathrm{LPR}}, s^{\mathrm{LPR}})$ be the optimal solution of (LPR). Then, the set $M$ defined above is equal to the set of indices $i$ where $s_i^{\mathrm{LPR}} = 0$.

Under non-degeneracy, the number of non-zero variables in $(x^{\mathrm{LPR}}, s^{\mathrm{LPR}})$ is equal to $n+m$, the number of constraints in (LPR). The number of non-zero assignment variables is equal to $(n - |S|) + |F|$, where the first term corresponds to the variables satisfying $x_{ij}^{\mathrm{LPR}} = 1$, and the second term to the fractional assignment variables. Furthermore, there are $m - |M|$ non-zero surplus variables. Thus we obtain

$$n + m = (n - |S|) + |F| + (m - |M|)$$

which implies the desired result. □

Some properties will be derived for the dual programming problem corresponding to (LPR). Let (D) denote the dual problem of (LPR). Problem (D) can be formulated as

$$\max \quad \sum_{j=1}^{n} v_j - \sum_{i=1}^{m} b_i\lambda_i$$

$$\text{s.t.} \tag{D}$$

$$v_j \leqslant c_{ij} + a_{ij}\lambda_i, \quad i = 1,\dots,m;\ j = 1,\dots,n,$$

$$\lambda_i \geqslant 0, \quad i = 1,\dots,m,$$

$$v_j \quad \text{free} \quad j = 1,\dots,n.$$

Under non-degeneracy of (LPR), non-split jobs can be distinguished from split jobs using the dual optimal solution, as the following proposition shows.

**Proposition 2.2.** *Suppose that* (LPR) *is non-degenerate. Let* $x^{\mathrm{LPR}}$ *be the optimal solution of* (LPR) *and* $(\lambda^{\mathrm{D}}, v^{\mathrm{D}})$ *be the optimal solution of* (D). *Then,*

(i) *For each* $j \notin S$, $x_{ij}^{\mathrm{LPR}} = 1$ *if and only if*

$$c_{ij} + \lambda_i^{\mathrm{D}} a_{ij} = \min_s (c_{sj} + \lambda_s^{\mathrm{D}} a_{sj}),$$

*and*

$$c_{ij} + \lambda_i^{\mathrm{D}} a_{ij} < \min_{s \neq i} (c_{sj} + \lambda_s^{\mathrm{D}} a_{sj}).$$

(ii) *For each* $j \in S$, there exists some $i = 1, \ldots, m$ such that

$$c_{ij} + \lambda_i^{\mathrm{D}} a_{ij} = \min_{s \neq i} (c_{sj} + \lambda_s^{\mathrm{D}} a_{sj}).$$

**Proof.** First, observe that $v_j^{\mathrm{D}} = \min_s (c_{sj} + \lambda_s^{\mathrm{D}} a_{sj}) \geqslant 0$ for each $j = 1, \ldots, n$. Thus, without loss of optimality, we can add to (D) non-negativity constraints for the variables $v_j$. By adding surplus variables $s_{ij}$ to the constraints in (D), we obtain the following alternative formulation of the dual problem.

$$\max \quad \sum_{j=1}^{n} v_j - \sum_{i=1}^{m} b_i \lambda_i$$

s.t.

$$\begin{aligned}
& v_j + s_{ij} = c_{ij} + a_{ij} \lambda_i, \quad i = 1, \ldots, m, \; j = 1, \ldots, n, \\
& \lambda_i \geqslant 0, \quad i = 1, \ldots, m, \\
& v_j \geqslant 0, \quad j = 1, \ldots, m, \\
& s_{ij} \geqslant 0, \quad i = 1, \ldots, m; \; j = 1, \ldots, n.
\end{aligned}$$

Let $(\lambda^{\mathrm{D}}, v^{\mathrm{D}}, s^{\mathrm{D}})$ be the optimal solution of (D). For each $j \in S$ there exists at least two variables $x_{ij}^{\mathrm{LPR}}$ that are strictly positive. Hence, by the complementary slackness conditions, there exists at least two variables $s_{ij}^{\mathrm{D}}$ that are equal to zero. This proves claim (ii).

To prove claim (i), it is enough to show that for each $j \notin S$ there exists exactly one variable $s_{ij}^{\mathrm{D}}$ that is equal to zero. By the complementary slackness conditions we know that there exists at least one such variable. It thus remains to show the uniqueness, which we will do by counting the variables that are zero in the vector $(\lambda^{\mathrm{D}}, v^{\mathrm{D}}, s^{\mathrm{D}})$. There are at least $m - |M|$ variables $\lambda_i^{\mathrm{D}}$, $|F|$ variables $s_{ij}^{\mathrm{D}}$ corresponding to $j \in S$, and $n - |S|$ variables $s_{ij}^{\mathrm{D}}$ corresponding to $j \notin S$ that are equal to zero. Thus, in total, there are at least $(m - |M|) + |F| + (n - |S|) = m + n$ zeroes in the dual solution, where the equality follows from Lemma 2.1. So, these are exactly all the variables at level zero in the vector $(\lambda^{\mathrm{D}}, v^{\mathrm{D}}, s^{\mathrm{D}})$. Then, for each $j \notin S$ there exists exactly one variable $s_{ij}^{\mathrm{D}} = 0$, and statement (i) follows. □

## 3. A class of greedy algorithms

### 3.1. Existing greedy algorithms

Martello and Toth [10] propose a heuristic for the GAP that is based on an ordering of the jobs. There, the assignment of job $j$ to machine $i$ is measured by a weight function $f(i,j)$. For each job, the difference between the second smallest and smallest values of $f(i,j)$ is computed, and the jobs are assigned in decreasing order of this difference. That is, for each job the *desirability* of assigning that job to its best machine is given by

$$\rho_j = \max_i \min_{s \neq i} (f(s,j) - f(i,j))$$

or

$$\rho_j = \min_{s \neq i_j} f(s,j) - f(i_j,j),$$

where

$$i_j = \arg\min_i f(i,j).$$

Due to capacity constraints on the machines, and given a job $j$, the index $i$ can assume the values of all feasible machines for that job, i.e., those machines that have sufficient capacity to process it.

Examples of the weight function $f(i,j)$ used by Martello and Toth [10] are
(i) $f(i,j) = c_{ij}$,
(ii) $f(i,j) = a_{ij}$,
(iii) $f(i,j) = a_{ij}/b_i$, and
(iv) $f(i,j) = -c_{ij}/a_{ij}$.

The motivation for choosing weight function (i) is that it is desirable to assign a job to a machine that can process it as cheaply as possible, and the motivation for weight functions (ii) and (iii) is that it is desirable to assign a job to a machine that can process it using the least (absolute or relative) capacity. Weight function (iv) tries to consider the effects of the previous weight functions jointly.

The greedy algorithm now reads:

**Greedy algorithm**

*Step* 0: Set $J = \{1,\ldots,n\}$, and $b'_i = b_i$ for $i = 1,\ldots,m$.

*Step* 1: Let $\mathscr{F}_j = \{i: a_{ij} \leq b'_i\}$ for $j \in J$. If $\mathscr{F}_j = \emptyset$ for some $j \in J$: STOP, the algorithm could not find a feasible solution. Otherwise, let

$$i_j = \arg\min_{i \in \mathscr{F}_j} f(i,j) \quad \text{for } j \in J,$$

$$\rho_j = \min_{\substack{s \in \mathscr{F}_j \\ s \neq i_j}} f(s,j) - f(i_j,j) \quad \text{for } j \in J.$$

*Step* 2: Let $\hat{\jmath} = \arg\max_{j\in J} \rho_j$, i.e., $\hat{\jmath}$ is the job to be assigned next, to machine $i_{\hat{\jmath}}$:

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

$$x^{\mathrm{G}}_{i\hat{\jmath}} = 0, \quad \text{for } i = 1,\dots,m;\ i \neq i_{\hat{\jmath}},$$

$$b'_{i_{\hat{\jmath}}} = b'_{i_{\hat{\jmath}}} - a_{i_{\hat{\jmath}}\hat{\jmath}},$$

$$J = J\setminus\{\hat{\jmath}\}.$$

*Step* 3: If $J = \emptyset$: STOP, $x^{\mathrm{G}}$ is a feasible solution to the GAP. Otherwise, go to Step 1.

In the next section we propose a new family of weight functions.

### 3.2. A new class of algorithms

As in weight function (iv) mentioned above, we would like to jointly take into account the fact that it is desirable to assign a job to a machine with minimal cost and minimal capacity. In order to achieve this, we define the family of weight functions

$$\{f_\lambda(i,j)\colon \lambda \in \mathbb{R}^m_+\},$$

where

$$f_\lambda(i,j) = c_{ij} + \lambda_i a_{ij}.$$

Note that if $\lambda_i = 0$ for all $i$, we obtain weight function (i). Furthermore, if $\lambda_i = M$ for all $i$, we approach weight function (ii) as $M$ grows large, whereas if $\lambda_i = M/b_i$ for all $i$ we approach weight function (iii) as $M$ increases.

For any non-negative vector $\lambda$, the weight function $f_\lambda$ defines a greedy algorithm, as described in the previous section. However, in order to be able to analyze the algorithm probabilistically, we modify it slightly as follows:

**Modified greedy algorithm**

*Step* 0: Set $J = \{1,\dots,n\}$, $b'_i = b_i$ for $i = 1,\dots,m$, and $\mathscr{F}_j = \{1,\dots,m\}$.

*Step* 1: If $\mathscr{F}_j = \emptyset$ for some $j \in J$: STOP, the algorithm could not find a feasible solution. Otherwise, let

$$i_j = \arg\min_{i\in\mathscr{F}_j} f(i,j) \quad \text{for } j \in J,$$

$$\rho_j = \min_{\substack{s\in\mathscr{F}_j\\ s\neq i_j}} f(s,j) - f(i_j,j) \quad \text{for } j \in J.$$

*Step* 2: Let $\hat{\jmath} = \arg\max_{j\in J} \rho_j$, i.e., $\hat{\jmath}$ is the job to be assigned next, to machine $i_{\hat{\jmath}}$. If $a_{i_{\hat{\jmath}}\hat{\jmath}} > b'_{i_{\hat{\jmath}}}$ then this assignment is not feasible; let $\mathscr{F}_j = \{i\colon a_{ij} \leq b'_i\}$ for $j \in J$

and go to Step 1. Otherwise,

$$x^{\mathrm{MG}}_{i_{\hat{\jmath}}\hat{\jmath}} = 1,$$
$$x^{\mathrm{MG}}_{i\hat{\jmath}} = 0, \quad \text{for } i = 1,\dots,m;\ i \neq i_{\hat{\jmath}},$$
$$b'_{i_{\hat{\jmath}}} = b'_{i_{\hat{\jmath}}} - a_{i_{\hat{\jmath}}\hat{\jmath}},$$
$$J = J \setminus \{\hat{\jmath}\}.$$

*Step* 3: If $J=\emptyset$: STOP, $x^{\mathrm{MG}}$ is a feasible solution to the GAP. Otherwise, go to Step 1.

The difference between this algorithm and the original greedy algorithm described in Section 3.1 is twofold. Firstly, in the initial stage of the modified greedy algorithm the capacity constraints are not taken into account when deciding which job to assign next. Secondly, there is a difference in the updating of the desirabilities $\rho_j$. In the original greedy algorithm, these are updated after each assignment of a job to a machine. In the modified greedy algorithm, the desirabilities are not updated as long as it is possible to assign the job with the largest desirability to its most desirable machine. In the next section we will discuss some properties of a specific choice for the vector $\lambda$.

### 3.3. Using the optimal dual vector

In the remainder of Section 3 we will derive some properties of the modified greedy algorithm, analogously to the properties of a class generalized greedy algorithms for the Multi-Knapsack Problem (see [13]).

The following proposition shows that the (partial) solution given by the modified greedy algorithm and the optimal solution of (LPR) coincide for all the non-split jobs in the optimal solution of (LPR), for a particular choice of the vector $\lambda$.

Let $x^{\mathrm{LPR}}$ be the optimal solution for (LPR). Let $(\lambda^{\mathrm{D}}, v^{\mathrm{D}})$ be the optimal dual vector, i.e., the optimal solution of (D) defined in Section 2. Let $NS$ be the set of non-split jobs of (LPR), i.e., $NS = \{1,\dots,n\}\setminus S$, where $S$ was defined in Section 2. Let $x^{\mathrm{MG}}$ be the (partial) solution of the GAP obtained by the modified greedy algorithm when $\lambda = \lambda^{\mathrm{D}}$.

**Proposition 3.1.** *Suppose that* (LPR) *is non-degenerate and feasible. If* $\lambda = \lambda^{\mathrm{D}}$, *then, for all*

$$x^{\mathrm{LPR}}_{ij} = 1 \Rightarrow x^{\mathrm{MG}}_{ij} = 1.$$

**Proof.** Consider the initial values of $\rho_j$ and $i_j$ ($j = 1,\dots,n$) in the modified greedy algorithm. The result then follows from the following claims:

(i) For all jobs $j \in NS$, we have that $x^{\mathrm{LPR}}_{i_j j} = 1$, i.e., the non-split jobs in the solution of (LPR) are assigned to the most desirable machine.

(ii) Capacity constraints are not violated for the partial solution given by the non-split jobs $j \in NS$.

(iii) The modified greedy algorithm considers all non-split jobs $j \in NS$ before the split jobs $j \in S$.

Claim (i) follows directly from Proposition 2.2(i) and the definition of the desirabilities $\rho_j$. Claim (ii) follows from the feasibility of $x^{\mathrm{LPR}}$ with respect to the capacity constraints. By again using Proposition 2.2, it follows that $\rho_j > 0$ for all $j \in NS$, and $\rho_j = 0$ for all $j \in S$, so that claim (iii) follows. Thus, all jobs $j \in NS$ are assigned in the same way by (LPR) and the modified greedy algorithm. □

### 3.4. Geometrical interpretation of the algorithm

In this section we will show how the modified greedy algorithm can be interpreted geometrically. To this end, define, for each job $j$, a set of $(m-1)m$ points $P^{jis} \in \mathbb{R}^{m+1}$ $(i, s = 1, \ldots, m,\ s \neq i)$ as follows:

$$(P^{jis})_\ell = \begin{cases} a_{ij} & \text{if } \ell = i, \\ -a_{sj} & \text{if } \ell = s, \\ c_{ij} - c_{sj} & \text{if } \ell = m+1, \\ 0 & \text{otherwise.} \end{cases}$$

Now consider $\lambda \in \mathbb{R}^m$ and the corresponding weight function $f_\lambda(i,j) = c_{ij} + \lambda_i a_{ij}$. Furthermore, define a hyperplane in $\mathbb{R}^{m+1}$ with normal vector $(\lambda, 1)$, i.e., a hyperplane of the form

$$\left\{ p \in \mathbb{R}^{m+1}\colon \sum_{\ell=1}^{m} \lambda_\ell p_\ell + p_{m+1} = C \right\}. \tag{2}$$

Observe that this hyperplane passes through the point $P^{jis}$ if

$$\begin{aligned} C &= \lambda_i a_{ij} - \lambda_s a_{sj} + c_{ij} - c_{sj} \\ &= f_\lambda(i,j) - f_\lambda(s,j). \end{aligned}$$

So, if machine $i$ is preferred over machine $s$ for processing job $j$ by the weight function $f_\lambda$ (i.e., $f_\lambda(i,j) < f_\lambda(s,j)$) then the point $P^{jis}$ lies *below* the hyperplane of the form (2) with $C = 0$, whereas otherwise the point $P^{jsi}$ lies *above* it.

Now let $C$ be a (negative) constant such that none of the points $P^{ijs}$ lie in the half-space

$$\left\{ p \in \mathbb{R}^{m+1}\colon \sum_{\ell=1}^{m} \lambda_\ell p_\ell + p_{m+1} \leq C \right\} \tag{3}$$

and for the moment disregard the capacity constraints of the machines. When $C$ is increased from this initial value, the corresponding half-space starts containing points $P^{jis}$. The interpretation of this is that whenever a point $P^{jis}$ is reached by the hyperplane defining the half-space, machine $i$ is preferred over machine $s$ for processing job $j$ with respect to the weight function $f_\lambda$. As soon as the half-space contains, for some $j$ and some $i$, all points $P^{jis}$ $(s \neq i)$, machine $i$ is preferred to all other machines, and job $j$ is assigned to machine $i$.

Now let us see in what order the jobs are assigned to machines. If for some job $j$ and some machine $i$ all points of the form $P^{jis}$ are contained in the half-space (3), then

$$C \geqslant \max_{s \neq i} (f_\lambda(i,j) - f_\lambda(s,j)).$$

The first time this occurs for some machine $i$ is if

$$C = \min_i \max_{s \neq i} (f_\lambda(i,j) - f_\lambda(s,j))$$

or, equivalently,

$$\begin{aligned} C &= -\max_i \min_{s \neq i} (f_\lambda(s,j) - f_\lambda(i,j)) \\ &= -\rho_j. \end{aligned}$$

Finally, the first job for which this occurs is the job for which the above value of $C$ is minimal, or for which $\rho_j$ is maximal. Thus, when capacity constraints are not considered, the movement of the hyperplane orders the jobs in the same way as the desirabilities $\rho_j$.

The modification of the geometrical version of the algorithm to include capacity constraints is straightforward. As soon as the geometrical algorithm would like to assign a job to a machine with insufficient remaining capacity, all points corresponding to this combination are removed, and the algorithm continues in the same way as before. If at some point all points corresponding to a job have been removed, this job cannot be scheduled feasibly and the algorithm terminates. In this way we precisely obtain the modified greedy algorithm.

### 3.5. Computational complexity of finding optimal multipliers

The performance of the modified greedy algorithm depends on the choice of a non-negative vector $\lambda \in \mathbb{R}^m$. Obviously, we would like to choose this vector $\lambda$ in such a way that the solution obtained is the one with the smallest objective function value attainable by the class of algorithms. Make the dependence on the solution found by the modified greedy algorithm on $\lambda$ explicit by denoting this solution by $x_{ij}^{\mathrm{MG}}(\lambda)$. Then define for all vectors $\lambda \in \mathbb{R}_+^m$

$$z^{\mathrm{MG}}(\lambda) = \begin{cases} \displaystyle\sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}^{\mathrm{MG}}(\lambda) & \text{if the modified greedy algorithm is feasible for } \lambda, \\ \infty & \text{otherwise.} \end{cases}$$

If there exists a vector $\lambda \geqslant 0$ with $z^{\mathrm{MG}}(\lambda) < \infty$ (in other words, the algorithm gives a feasible solution of the GAP for $\lambda$), we can define the best vector, $\tilde{\lambda}$, as the minimizer of $z^{\mathrm{MG}}(\lambda)$ over all the non-negative vectors $\lambda \in \mathbb{R}^m$ (if this minimum exists), i.e.,

$$z^{\mathrm{MG}}(\tilde{\lambda}) = \min_{\lambda \in \mathbb{R}_+^m} z^{\mathrm{MG}}(\lambda).$$

The following result shows how we can find the best set of multipliers (or decide that no choice of multipliers yields a feasible solution) in polynomial time (if the number of machines $m$ is fixed).

**Theorem 3.2.** *If the number of machines m in the GAP is fixed, there exists a polynomial time algorithm to determine an optimal set of multipliers, or to decide that no vector $\lambda \in \mathbb{R}^m_+$ exists such that the modified greedy algorithm finds a feasible solution of the GAP.*

**Proof.** Each vector $\lambda \in \mathbb{R}^m_+$ induces an ordering of the points $P^{jis}$, and thus an assignment of jobs to machines and an ordering of these assignments. Each of these orderings is given by a hyperplane in $\mathbb{R}^{m+1}$, and thus we need to count the number of hyperplanes giving different orderings. Those can be found by shifting hyperplanes in $\mathbb{R}^{m+1}$. The number of possible orderings is $O(n^{m+1}\log n)$ (see [9,13]). For each order obtained, the greedy algorithm requires $O(n^2)$ time to compute the solution for the GAP. Then, all the possible solutions can be found in $O(n^{m+3}\log n)$ time. In the best case, when at least there exists a vector $\lambda \in \mathbb{R}^m_+$ giving a feasible solution, we need $O(\log(n^{m+3}\log n)) = O(\log n)$ time to select the best set of multipliers. Thus, in $O(n^{m+3}\log n)$ we can find the best set of multipliers, or decide that the modified greedy algorithm is infeasible for each $\lambda \in \mathbb{R}^m_+$. □

## 4. Probabilistic analysis of the algorithm

### 4.1. A probabilistic model

In this section we will analyze the asymptotical behavior of modified greedy algorithm, when the number of jobs $n$ goes to infinity and the number of machines $m$ remains fixed. We impose a stochastic model on the parameters of the GAP, as proposed by Romeijn and Piersma [15].[1] Let $(A_j, C_j)$ be an i.i.d. absolutely continuous random vector in the bounded set $[0, \bar{A}]^m \times [\underline{C}, \bar{C}]^m$, where $A_j = (A_{1j}, \ldots, A_{mj})$ and $C_j = (C_{1j}, \ldots, C_{mj})$. Furthermore, let the capacities $b_i (i = 1, \ldots, m)$ depend linearly on $n$, i.e., $b_i = \beta_i n$, for positive constants $\beta_i$. Observe that the number of machines $m$ is fixed, thus the size of the instance of the GAP only depends on the number of jobs $n$.

As shown by Romeijn and Piersma [15], feasibility of the instances of the GAP is not guaranteed under the above stochastic model, even for the LP-relaxation (LPR) of the GAP. The following assumption ensures feasibility of the GAP with probability 1 as $n$ goes to infinity.

[1] Throughout this paper, random variables will be denoted by capital letters, and their realizations by the corresponding lowercase letters.

**Assumption 4.1.** The excess capacity

$$\Delta = \min_{\lambda\in\Omega} \left(\lambda^\top\beta - \mathscr{E}\left(\min_i(\lambda_i A_{i1})\right)\right)$$

(where $\Omega$ is the unit simplex) is strictly positive.

**Theorem 4.2** (cf. Romeijn and Piersma [15]). *As $n\to\infty$, the GAP is infeasible with probability one if $\Delta<0$, and feasible with probability if $\Delta>0$.*

Under feasibility of the GAP, some results on the convergence of the normalized optimal value of (LPR) and the GAP are derived in [15]. Let $Z_n$ be the random variable representing the optimal value of the GAP, and $Z_n^{\mathrm{LPR}}$ be the optimal value of (LPR). Let $X_n$ be the random vector representing the optimal solution of the GAP, and $X_n^{\mathrm{LPR}}$ be the optimal solution of (LPR).

**Theorem 4.3** (cf. Romeijn and Piersma [15]). *The normalized optimal value of (LPR), $(1/n)Z_n^{\mathrm{LPR}}$, tends to*

$$\theta \equiv \max_{\lambda\geqslant 0}\left(\mathscr{E}\left(\min_i(C_{i1}+\lambda_i A_{i1})\right) - \lambda^\top\beta\right)$$

*with probability when n goes to infinity.*

Assumption 4.4 ensures that the normalized optimal value of the GAP converges to the same constant $\theta$. Denote by $e$ the vector in $\mathbb{R}^m$ whose components are all equal to one.

**Assumption 4.4.** $\psi'_+(0)$, the right derivative of $\psi:\mathbb{R}\to\mathbb{R}$ is strictly positive, where

$$\psi(x) = \min_{\lambda\geqslant xe}\left(\lambda^\top\beta - \mathscr{E}\left(\min_i(C_{i1}+\lambda_i A_{i1})\right)\right).$$

**Theorem 4.5** (cf. Romeijn and Piersma [15]). *Under Assumption* 4.4,

$$Z_n \leqslant Z_n^{\mathrm{LPR}} + (\bar{C}-\underline{C})\cdot m$$

*with probability as $n\to\infty$, and $(1/n)Z_n$ tends to $\theta$ with probability as $n\to\infty$.*

The proof of this result is based on showing that, under Assumption 4.4, the normalized sum of the slacks of the capacity constraints of the optimal solution of (LPR) is eventually strictly positive. Since we will make explicit use of this result, we will state it as a theorem.

**Theorem 4.6** (cf. Romeijn and Piersma [15]). *Under Assumption* 4.4,

$$\sum_{i=1}^m \beta_i - \frac{1}{n}\sum_{i=1}^m\sum_{j=1}^n A_{ij}X_{ij}^{\mathrm{LPR}} > 0 \quad \textit{with probability one as } n\to\infty.$$

Finally, the following proposition ensures that (LPR) is non-degenerate with probability one, which will enable us to use Proposition 3.1.

**Proposition 4.7.** (LPR) *is non-degenerate with probability, under the proposed stochastic model.*

**Proof.** The proof of Proposition 2.2 shows that the feasible region of the dual of (LPR) can be expressed as

$$v_j + s_{ij} = c_{ij} + a_{ij}\lambda_i \quad i = 1,\dots,m;\ j = 1,\dots,n. \tag{4}$$

Any basic solution to this system can be characterized by choosing a subset of $m+n$ variables to be equal to zero. Degeneracy now means that one of the remaining variables needs to be zero as well. Since each of the hyperplanes in (4) has a random coefficient and/or right-hand side, this happens with probability zero. □

From now on, we will assume that Assumptions 4.1 and 4.4 are satisfied. In the remainder of this section we will then show that the modified greedy algorithm is asymptotically feasible and optimal for two different choices of $\lambda$.

In Section 4.2, we consider the choice $\lambda = \lambda_n^*$, where $\lambda_n^*$ represents the optimal dual multipliers of the capacity constraints of (LPR) when (LPR) is feasible and an arbitrary non-negative vector when (LPR) is infeasible. (Clearly, if (LPR) is infeasible, so is the GAP.) Note that this choice depends on the problem instance. Therefore, in Section 4.3 we give conditions under which the sequence of random variables $\{\Lambda_n^*\}$ converges with probability one to a vector $\lambda^* \in \mathbb{R}_+^m$, only depending on the probabilistic model. Hence, the choice of $\lambda$ will be equal for all problem instances (and problem sizes, as measured by $n$) corresponding to that model. Again, asymptotic feasibility and optimality will be shown.

In the remainder of this paper, let $X_n^{\mathrm{MG}}$ denote the solution of the GAP given by the modified greedy algorithm, and $Z_n^{\mathrm{MG}}$ be its objective value. Note that $X_n^{\mathrm{MG}}$ and $Z_n^{\mathrm{MG}}$ depend on the choice of $\lambda$. This dependence will be suppressed for notational convenience, but at any time the particular value of $\lambda$ considered will be clear from the context.

### *4.2. The optimal dual multipliers*

In this section we will choose the vector of optimal dual multipliers of the capacity constraints of (LPR), say $\lambda_n^*$, as the multipliers to use in the modified greedy algorithm. (As mentioned above, if (LPR) is infeasible we let $\lambda_n^*$ be any non-negative vector.)

In Theorem 4.8, we show that the modified greedy algorithm is asymptotically feasible with probability one. This proof combines the results of Proposition 3.1, where it is shown that $X_n^{\mathrm{LPR}}$ and $X_n^{\mathrm{MG}}$ coincide for the non-split jobs of the solution of (LPR), and Theorem 4.6. For notational simplicity, we suppress the dependence of the vectors $X_n^{\mathrm{LPR}}$ and $X_n^{\mathrm{MG}}$ on $n$.

**Theorem 4.8.** *The modified greedy algorithm is asymptotically feasible with probability one, when $\lambda = \lambda_n^*$.*

**Proof.** Note that the result follows if

$$\sum_{i=1}^{m} \beta_i - \frac{1}{n} \sum_{i=1}^{m} \sum_{j \in NS} A_{ij} X_{ij}^{\mathrm{MG}} > 0 \tag{5}$$

with probability when $n$ goes to infinity, since this implies that the capacity remaining for the jobs in $S$ grows linearly in $n$, while $|S| \leqslant m$ (see [2], and Lemma 2.1).

To show this result, recall that by Theorem 4.2 and Proposition 4.7, (LPR) is feasible and non-degenerate with probability one. For any feasible and non-degenerate instance, Proposition 3.1 now says that $x^{\mathrm{LPR}}$ and $x^{\mathrm{MG}}$ coincide for each job $j \in NS$, the set of non-split jobs of (LPR). In other words, for each problem instance,

$$x_{ij}^{\mathrm{LPR}} = x_{ij}^{\mathrm{HS}} \quad \text{for all } j \in NS, i = 1, \ldots, m.$$

Thus,

$$\begin{aligned} \sum_{i=1}^{m} \beta_i - \frac{1}{n} \sum_{i=1}^{m} \sum_{j \in NS} A_{ij} X_{ij}^{\mathrm{MG}} &= \sum_{i=1}^{m} \beta_i - \frac{1}{n} \sum_{i=1}^{m} \sum_{j \in NS} A_{ij} X_{ij}^{\mathrm{LPR}} \\ &\geqslant \sum_{i=1}^{m} \beta_i - \frac{1}{n} \sum_{i=1}^{m} \sum_{j=1}^{n} A_{ij} X_{ij}^{\mathrm{LPR}} \\ &> 0 \end{aligned} \tag{6}$$

with probability as $n \to \infty$, where the strict inequality (6) follows from Theorem 4.6. □

In Theorem 4.9, we show that the modified greedy algorithm is asymptotically optimal with probability. The proof is similar to the proof of Theorem 4.8.

**Theorem 4.9.** *The modified greedy algorithm is asymptotically optimal with probability one, when $\lambda = \lambda_n^*$.*

**Proof.** From Theorem 4.8 we know that the modified greedy algorithm is asymptotically feasible with probability, for $\lambda = \lambda_n^*$. Moreover, Theorems 4.3 and 4.5 imply that $|(1/n)Z_n - (1/n)Z_n^{\mathrm{LPR}}| \to 0$ with probability. It thus suffices to show that $|(1/n)Z_n^{\mathrm{LPR}} - (1/n)Z_n^{\mathrm{MG}}| \to 0$ with probability. By definition,

$$\begin{aligned} \left| \frac{1}{n} Z_n^{\mathrm{LPR}} - \frac{1}{n} Z_n^{\mathrm{MG}} \right| &= \frac{1}{n} Z_n^{\mathrm{MG}} - \frac{1}{n} Z_n^{\mathrm{LPR}} \\ &= \frac{1}{n} \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij} X_{ij}^{\mathrm{MG}} - \frac{1}{n} \sum_{i=1}^{m} \sum_{j=1}^{n} C_{ij} X_{ij}^{\mathrm{LPR}} \end{aligned}$$

$$= \frac{1}{n}\sum_{i=1}^{m}\sum_{j\in S} C_{ij}X_{ij}^{\mathrm{MG}} - \frac{1}{n}\sum_{i=1}^{m}\sum_{j\in S} C_{ij}X_{ij}^{\mathrm{LPR}}$$

$$\leqslant \bar{C}\frac{m}{n} - \underline{C}\frac{m}{n} \to 0 \quad \text{with probability one,} \tag{7}$$

where Eq. (7) follows from Proposition 3.1, since (LPR) is feasible and non-degenerate with probability one (see Theorem 4.2 and Proposition 4.7). Thus, the result follows. □

### *4.3. A unique vector of multipliers*

The asymptotic optimality of the modified greedy algorithm has been proved by choosing $\lambda = \lambda_n^*$. However, using this choice the vector of multipliers depends on the problem instance. In this section, we will derive conditions under which a single vector of multipliers suffices for all instances and problem sizes (as measured by the number of jobs) under a given probabilistic model. (See Rinnooy et al. [13] for an analogous result for a class of generalized greedy algorithms for the Multi-Knapsack Problem.)

Let $L : \mathbb{R}^m \to \mathbb{R}$ be the function defined by

$$L(\lambda) = \mathscr{E}\left(\min_{i=1,\dots,m}(C_{i1} + \lambda_i A_{i1})\right) - \lambda^\top \beta. \tag{8}$$

Recall from Theorem 4.3 that the maximum value of the function $L$ on the set $\mathbb{R}_+^m$ is equal to $\theta$. We will first show that, under some regularity conditions, the function $L$ has a unique maximizer, say $\lambda^*$, over the non-negative orthant. Next we prove that the modified greedy algorithm, with $\lambda = \lambda^*$, is asymptotically feasible and optimal.

**Lemma 4.10.** *The following statements hold*:
(i) *The function L is concave.*
(ii) $L(\Lambda_n^*) \to \theta$ *with probability when n goes to infinity.*

**Proof.** See the appendix. □

In the remainder of this paper we will impose the following regularity conditions.

**Assumption 4.11.** For each $i = 1,\dots,m$,
(i) $\mathscr{E}(A_{i1}) > \beta_i$.
(ii) $\mathscr{E}(A_{i1}I_i) > 0$, where $I_i$ is a random variable taking the value 1 if $i = \arg\min_s C_{s1}$, and 0 otherwise.

Assumption 4.11(i) says that there should be no machine that can process all jobs with probability one (if $n$ goes to infinity). Assumption 4.11(ii) says that every machine should be desirable for a significant (i.e., increasing linearly with $n$ with probability one) number of jobs when processing costs are taken into account.

We are now able to show the first main results of this section.

**Theorem 4.12.** *If the density of* $(C_1, A_1)$ *is strictly positive over a convex open set, and if Assumption* 4.11 *holds, then L has a unique maximizer on the set* $\mathbb{R}^m_+$.

**Proof.** See the appendix. □

**Proposition 4.13.** *If the density of* $(C_1, A_1)$ *is strictly positive on a convex open set, and if Assumption* 4.11 *holds, there exists a unique vector* $\lambda^* \in \mathbb{R}^m_+$ *such that*

$$\Lambda_n^* \to \lambda^*$$

*with probability one when n goes to infinity.*

**Proof.** This result follows immediately by using Corollary 27.2.2 in Rockafellar [14], Lemma 4.10, Theorem 4.12, and the remark following Eq. (8) at the beginning of this section. □

The asymptotic results are based on showing that the algorithm assigns most of the jobs in the same way when using $\lambda^*$ or $\lambda_n^*$, when $n$ is large enough. Recall that $NS_n$ represents the set of non-split jobs of the optimal solution for (LPR) and $\rho_j$ is calculated with vector $\lambda^*$.

First, we will define a barrier $\varepsilon_n$ such that the best machine with $\lambda^*$ and $\lambda_n^*$ is the same for each job $j$ satisfying $\rho_j > \varepsilon_n$. The barrier $\varepsilon_n$ is defined as

$$\varepsilon_n = \sup_{j=1,\dots,n} \max_{\ell \neq i}((\lambda_\ell^* - (\lambda_n^*)_\ell)a_{\ell j} - (\lambda_i^* - (\lambda_n^*)_i)a_{ij}),$$

where $(\lambda_n^*)_\ell$ represents the $\ell$th component of vector $\lambda_n^* \in \mathbb{R}^m_+$. Note that $\varepsilon_n \geqslant 0$.

**Proposition 4.14.** *If* $\rho_j > \varepsilon_n$, *then*

$$\arg\min_s(c_{sj} + \lambda_s^* a_{sj}) = \arg\min_s(c_{sj} + (\lambda_n^*)_s a_{sj}).$$

**Proof.** Let $j$ be a job with $\rho_j > \varepsilon_n$. Since $\varepsilon_n$ is non-negative, the desirability of job $j$ is strictly positive, so that $i_j = \arg\min_s(c_{sj} + \lambda_s^* a_{sj})$ is unique.

Using the definition of $\varepsilon_n$, $\rho_j > \varepsilon_n$ implies that

$$\rho_j > \max_{\ell \neq i}((\lambda_\ell^* - (\lambda_n^*)_\ell)a_{\ell j} - (\lambda_i^* - (\lambda_n^*)_i)a_{ij}).$$

Since $\rho_j = \min_{s \neq i_j}((c_{sj} + \lambda_s^* a_{sj}) - (c_{i_j j} + \lambda_{i_j}^* a_{i_j j}))$, we thus have that

$$\min_{s \neq i_j}((c_{sj} + \lambda_s^* a_{sj}) - (c_{i_j j} + \lambda_{i_j}^* a_{i_j j})) > \max_{\ell \neq i_j}((\lambda_\ell^* - (\lambda_n^*)_\ell)a_{\ell j} - (\lambda_{i_j}^* - (\lambda_n^*)_{i_j})a_{i_j j}).$$

This implies that, for $s \neq i_j$,

$$(c_{sj} + \lambda_s^* a_{sj}) - (c_{i_j j} + \lambda_{i_j}^* a_{i_j j}) > (\lambda_s^* - (\lambda_n^*)_s)a_{sj} - (\lambda_{i_j}^* - (\lambda_n^*)_{i_j})a_{i_j j}$$

and thus

$$c_{sj} + (\lambda_n^*)_s a_{sj} > c_{i_j j} + (\lambda_n^*)_{i_j} a_{i_j j}$$

so, $i_j = \arg\min_s(c_{sj} + (\lambda_n^*)_s a_{sj})$. □

**Corollary 4.15.** *If* (LPR) *is feasible and non-degenerate, each* $j$ *with* $\rho_j > \varepsilon_n$ *is a non-split job of* (LPR).

**Proof.** From Proposition 4.14, if $\rho_j > \varepsilon_n$

$$\min_s (c_{sj} + (\lambda_n^*)_s a_{sj})$$

is reached at exactly one component. Since (LPR) is feasible and non-degenerate, result follows from Proposition 2.2(i). □

We may observe that the modified greedy algorithm with $\lambda = \lambda^*$ can assign all jobs with desirability $\rho_j > \varepsilon_n$, since all those jobs are non-split jobs of the optimal solution of (LPR) by Corollary 4.15, and they are assigned to the same machine as in the solution to (LPR) by Proposition 4.14. We will now study the behaviour of $\varepsilon_n$ as $n \to \infty$.

**Lemma 4.16.** $\varepsilon_n$ *tends to* 0 *with probability one as* $n$ *goes to infinity.*

**Proof.** This result follows immediately from Proposition 4.13. □

We already know that the modified greedy algorithm schedules all the jobs with desirability $\rho_j > \varepsilon_n$ without violating any capacity constraint. What remains to be shown is that there is enough space to assign the remaining jobs. In the next result, we study the number of jobs that has not been assigned yet. We will denote this set by $\mathcal{N}_n$, i.e.,

$$\mathcal{N}_n = \{j = 1, \ldots, n \colon \rho_j \leqslant \varepsilon_n\}.$$

**Proposition 4.17.** *We have that* $|\mathcal{N}_n|/n \to 0$ *with probability one when* $n$ *goes to infinity.*

**Proof.** Let $F_{\rho_1}$ be the distribution function of the random variable $\rho_1$. Given a number of jobs $n$, we define a Boolean random variable $Y_{jn}$ which takes value 1 if $\rho_j \leqslant \varepsilon_n$, and 0 otherwise, for each $j = 1, \ldots, n$. So,

$$\frac{|\mathcal{N}_n|}{n} = \frac{\sum_{j=1}^n Y_{jn}}{n}.$$

For fixed $n$, the variables $Y_{jn}$ are identically distributed as a Bernoulli random variable with parameter $P(\rho_j \leqslant \varepsilon_n) = P(\rho_1 \leqslant \varepsilon_n) = F_{\rho_1}(\varepsilon_n)$.

Now assume that the desired result is not true. Then, there exists a subsequence $\{|\mathcal{N}_{n_k}|/n_k\}$ which tends to $\ell > 0$, since the original sequence lies completely in the compact set $[0,1]$. Now, consider the sequence of variables $\bar{Y}_j$ taking the value 1 if $\rho_j \leqslant F_{\rho_1}^{-1}(\ell/2)$, and 0 otherwise. The variables $\bar{Y}_j$ are i.i.d. as a Bernoulli random variable with parameter $P(\rho_j \leqslant F_{\rho_1}^{-1}(\ell/2)) = F_{\rho_1}(F_{\rho_1}^{-1}(l/2)) = \ell/2$. Using Lemma 4.16 and the absolute continuity of the variables $C_1$ and $A_1$, there exists a constant $n_0 \in \mathbb{N}$ such that for all $n \geqslant n_0$, $F_{\rho_1}(\varepsilon_n) < \ell/2$, which implies that for each $n_k \geqslant n_0$ $Y_{jn_k} \leqslant \bar{Y}_j$, and then

$$\frac{|\mathcal{N}_{n_k}|}{n_k} = \frac{\sum_{j=1}^{n_k} Y_{jn_k}}{n_k} \leqslant \frac{\sum_{j=1}^{n_k} \bar{Y}_j}{n_k} \to \frac{\ell}{2},$$

where the convergence follows by the strong law of the large numbers. But this contradicts the fact that $|\mathcal{N}_{n_k}|/n_k$ tends to $\ell$. □

Now, we are able to prove that the modified greedy algorithm is asymptotically feasible when $\lambda = \lambda^*$, with probability one.

**Theorem 4.18.** *The modified greedy algorithm is asymptotically feasible with probability one for $\lambda = \lambda^*$.*

**Proof.** Since (LPR) is feasible and non-degenerate with probability one (see Theorem 4.2 and Proposition 4.7), from Corollary 4.15 we have that $x^{\mathrm{LPR}}$ and $x^{\mathrm{MG}}$ coincide for each job $j \notin \mathcal{N}_n$, that is

$$x_{ij}^{\mathrm{LPR}} = x_{ij}^{\mathrm{MG}} \quad \text{for all } j \notin \mathcal{N}_n;\ i = 1, \ldots, m.$$

Thus,

$$\begin{aligned} \frac{1}{n}\left(\sum_{i=1}^m b_i - \sum_{i=1}^m \sum_{j \notin \mathcal{N}_n} A_{ij} X_{ij}^{\mathrm{MG}}\right) &= \sum_{i=1}^m \beta_i - \frac{1}{n}\sum_{i=1}^m \sum_{j \notin \mathcal{N}_n} A_{ij} X_{ij}^{\mathrm{MG}} \\ &= \sum_{i=1}^m \beta_i - \frac{1}{n}\sum_{i=1}^m \sum_{j \notin \mathcal{N}_n} A_{ij} X_{ij}^{\mathrm{LPR}} \\ &\geqslant \sum_{i=1}^m \beta_i - \frac{1}{n}\sum_{i=1}^m \sum_{j=1}^n A_{ij} X_{ij}^{\mathrm{LPR}} \\ &> 0 \quad \text{with probability one as } n \to \infty, \end{aligned} \tag{9}$$

where inequality (9) follows from Theorem 4.6. To assign the remaining jobs it is enough to show that

$$\sum_{i=1}^m \left\lfloor \frac{b_i - \sum_{j=1}^n A_{ij} X_{ij}^{\mathrm{LPR}}}{\bar{A}} \right\rfloor \geqslant |\mathcal{N}_n|$$

which is true if

$$\sum_{i=1}^m \left( \frac{b_i - \sum_{j=1}^n A_{ij} X_{ij}^{\mathrm{LPR}}}{\bar{A}} \right) \geqslant m + |\mathcal{N}_n|$$

or

$$\frac{1}{n}\sum_{i=1}^m \left( b_i - \sum_{j=1}^n A_{ij} X_{ij}^{\mathrm{LPR}} \right) \geqslant \left( \frac{m + |\mathcal{N}_n|}{n} \right) \bar{A}.$$

From Proposition 4.16, $|\mathcal{N}_n|/n$ tends to zero with probability one when $n$ goes to infinity, so together with inequality (9) the result follows. $\square$

Finally, we can prove asymptotic optimality with probability one of the modified greedy algorithm when $\lambda = \lambda^*$.

**Theorem 4.19.** *The modified greedy algorithm is asymptotically optimal with probability one.*

**Proof.** From Theorem 4.18, the modified greedy algorithm is asymptotically feasible with probability one.

In a similar fashion as for Theorem 4.9, we have that $(1/n)Z_n - (1/n)Z_n^{\mathrm{LPR}} \to 0$. It then remains to show that $(1/n)Z_n^{\mathrm{LPR}} - (1/n)Z_n^{\mathrm{MG}} \to 0$. By definition,

$$\begin{aligned}\left|\frac{1}{n}Z_n^{\mathrm{LPR}} - \frac{1}{n}Z_n^{\mathrm{MG}}\right| &= \frac{1}{n}Z_n^{\mathrm{MG}} - \frac{1}{n}Z_n^{\mathrm{LPR}} \\ &= \frac{1}{n}\sum_{i=1}^{m}\sum_{j=1}^{n} C_{ij}X_{ij}^{\mathrm{MG}} - \frac{1}{n}\sum_{i=1}^{m}\sum_{j=1}^{n} C_{ij}X_{ij}^{\mathrm{LPR}} \\ &= \frac{1}{n}\sum_{i=1}^{m}\sum_{j\in\mathcal{N}_n} C_{ij}X_{ij}^{\mathrm{MG}} - \frac{1}{n}\sum_{i=1}^{m}\sum_{j\in\mathcal{N}_n} C_{ij}X_{ij}^{\mathrm{LPR}} \qquad (10) \\ &\leqslant \bar{C}\frac{|\mathcal{N}_n|}{n} - \underline{C}\frac{|\mathcal{N}_n|}{n}. \qquad (11)\end{aligned}$$

Equality (10) follows from Proposition 4.14, since (LPR) is feasible and non-degenerate with probability one (see Theorem 4.2 and Proposition 4.7). Then, using Proposition 4.16, both of the terms in (11) tend to zero with probability one when $n$ goes to infinity. $\square$

## 5. Summary

In this paper we have considered the Generalized Assignment Problem (GAP) of finding a minimum-cost assignment of jobs to machines. From a probabilistic analysis of the optimal value function of this problem, we have constructed a new class of greedy algorithms. Although we cannot guarantee that this algorithm finds a feasible, let alone optimal solution, we are able to show that, under a stochastic model of the problem parameters, a member of the class (that only depends on this stochastic model) is asymptotically feasible and optimal with probability one. Moreover, we have shown that the best solution obtainable by any member of the class can be found in polynomial time, when the number of machines is considered fixed.

## Appendix

Let $L_n : \mathbb{R}^m \to \mathbb{R}$ be a real-valued function defined as

$$L_n(\lambda) = \frac{1}{n}\sum_{j=1}^{n} \min_i (c_{ij} + \lambda_i a_{ij}) - \lambda^\top \beta.$$

To prove Lemma 4.10 we first need to prove the following auxiliary result. Recall that $\lambda = \lambda_n^*$ is defined as the vector of optimal dual multipliers of the capacity constraints of (LPR) when (LPR) is feasible and an arbitrary non-negative vector when (LPR) is infeasible.

**Proposition A.1.** *The following statements hold*:

(i) *If* (LPR) *is feasible*, $\lambda_n^*$ *is the maximizer of function* $L_n$ *on the set of non-negative vectors* $\lambda \in \mathbb{R}^m$.

(ii) $L_n(\Lambda_n^*) \to \theta$, *with probability one when n goes to infinity*.

(iii) *For n large enough*, $\lambda_n^*$ *has at least one component equal to zero with probability one*.

**Proof.** From the formulation of the dual problem (D) we can deduce $v_j = \min_i(c_{ij} + \lambda_i a_{ij})$ and the optimal value of (D) can be written as

$$\begin{aligned}\max_{\lambda \geqslant 0}\left(\sum_{j=1}^{n} \min_i (c_{ij} + \lambda_i a_{ij}) - \lambda^\top b\right) &= n \max_{\lambda \geqslant 0}\left(\frac{1}{n}\sum_{j=1}^{n} \min_i (c_{ij} + \lambda_i a_{ij}) - \lambda^\top \beta\right)\\ &= n \max_{\lambda \geqslant 0} L_n(\lambda),\end{aligned}$$

and statement (i) follows.

By strong duality, Theorem 4.2 and Proposition 4.7, $(1/n)Z_n^{\mathrm{LPR}} = L_n(\Lambda_n^*)$. Statement (ii) now follows by using Theorem 4.3.

In the proof of Theorem 4.5, functions $\Psi_n$ are defined as

$$\begin{aligned}\Psi_n(x) &= \min_{\lambda \geqslant xe}\left(\lambda^{\mathrm{T}}\beta - \frac{1}{n}\sum_{j=1}^{n} \min_i (c_{ij} + \lambda_i a_{ij})\right)\\ &= -\max_{\lambda \geqslant xe} L_n(\lambda).\end{aligned}$$

In this proof it is shown that the sequence $\{\Psi_n\}$ converges pointwise to function $\Psi$. Moreover, under Assumption 4.4, it is deduced that,

$$\liminf_{n\to\infty} (\Psi_n)'_+(0) > 0 \quad \text{with probability one.} \tag{A.1}$$

In particular, $\Psi_n(0) = -\max_{\lambda \in \mathbb{R}_+^m} L_n(\lambda)$. From (A.1), eventually, $\Psi_n(\epsilon) \geqslant \Psi_n(0)$ (where $\epsilon > 0$). Thus, the maximum of function $L_n$ on $\mathbb{R}_+^m$ cannot be reached in a vector with all components strictly positive. $\square$

Now we are able to prove Lemma 4.10.

**Lemma 4.10.** *The following statements hold*:
(i) *The function $L$ is concave.*
(ii) $L(\Lambda_n^*) \to \theta$ *with probability one when $n$ goes to infinity.*

**Proof.** Using the strong law of large numbers, it is easy to see that the sequence of functions $L_n$ converges pointwise to the function $L$, with probability one. Each of the functions $L_n$ is concave on $\mathbb{R}^m_+$, since it is expressed as the algebraic sum of a linear function and the minimum of linear functions. Thus, statement (i) follows by using pointwise convergence of $L_n$ to $L$ on $\mathbb{R}^m_+$.

To prove statement (ii), we first show uniform convergence of the functions $L_n$ to $L$ on a compact set containing the maximizers of the functions $L_n$ and $L$. Let $B$ be the compact set on $\mathbb{R}^m_+$ defined as

$$B = \left\{\lambda \in \mathbb{R}^m \colon \lambda \geqslant 0,\ \mathscr{E}\left(\max_s(C_{s1}) - \min_i(C_{i1})\right) - \lambda^\top \beta \geqslant 0\right\}. \tag{A.2}$$

Using the strong law of large numbers, we have

$$\Pr\left(\exists n_1 \colon \forall n \geqslant n_1,\ \frac{1}{n}\sum_{j=1}^{n}\left(\max_s(c_{sj}) - \min_i(c_{ij})\right)\right.$$
$$\left.\leqslant 1 + \mathscr{E}\left(\max_s(C_{s1}) - \min_i(C_{i1})\right)\right) = 1. \tag{A.3}$$

Since Assumption 4.4 is satisfied, Proposition A.1(iii) assures that if $n$ is large enough $L_n$ reaches its maximum in a vector with at least one component equal to zero, with probability one. By increasing $n_1$ in (A.3) if necessary, we can assume that for each $n \geqslant n_1$, $\Lambda_n^*$ has at least one component equal to zero with probability one. We will show that, for a fixed $n \geqslant n_1$, each vector $\lambda \in \mathbb{R}^m_+$, with $\lambda \ngtr 0$ and $\lambda \notin B$ is no better than the origin, that is, $L_n(\lambda) \leqslant L_n(0)$.

$$\begin{aligned} L_n(\lambda) &= \frac{1}{n}\sum_{j=1}^{n}\min_i(c_{ij} + \lambda_i a_{ij}) - \lambda^\top \beta \\ &\leqslant \frac{1}{n}\sum_{j=1}^{n}\max_i(c_{ij}) - \lambda^\top \beta \qquad \text{(A.4)} \\ &< \frac{1}{n}\sum_{j=1}^{n}\min_i(c_{ij}) \qquad \text{(A.5)} \\ &= L_n(0). \end{aligned}$$

Inequality (A.4) follows from the fact that $L_n$ reaches its maximum in a vector with at least one component equal to zero, and strict inequality (A.5) follows since $\lambda \notin B$ and $\lambda \in \mathbb{R}^m_+$. This means that, for each $\lambda \notin B$, $\lambda \in \mathbb{R}^m_+$

$$\Pr(\exists n_1 \colon \forall n \geqslant n_1,\ L_n(\lambda) \leqslant L_n(0)) = 1.$$

Since the origin belongs to $B$, this implies that $\Lambda_n^* \in B$ for each $n \geqslant n_1$, with probability one. In a similar fashion we can prove that each maximizer of the function $L$ belongs to $B$. Note that the set $B$ is compact since $\mathscr{E}(\max_s(C_{s1}) - \min_i(C_{i1}))$ is finite.

Theorem 10.8 in Rockafellar [14] shows that $L_n$ converges uniformly to $L$ with probability one on $B$. Now consider the following inequality

$$|L(\Lambda_n^*) - \theta| \leqslant |L(\Lambda_n^*) - L_n(\Lambda_n^*)| + |L_n(\Lambda_n^*) - \theta|.$$

From the uniform convergence the first term of the right-hand side tends to zero and from Proposition A.1(ii) the second term tends to zero, and statement (ii) follows. □

To prove Theorem 4.12, we first derive the Hessian of the function $L$. Before we do this, we will first introduce some simplifying notation. If $c \in \mathbb{R}^m$, then we define $c_{(k)} = (c_1, \ldots, c_{k-1}, c_{k+1}, \ldots, c_m) \in \mathbb{R}^{m-1}$. Moreover, we interpret $(c_{(k)}, z)$ to be equivalent to $(c_1, \ldots, c_{k-1}, z, c_{k+1}, \ldots, c_m)$, where the usage of either notation is dictated by convenience, and where the meaning should be clear from the context. Similarly, we define $c_{(k,i)}$ to be the vector in $\mathbb{R}^{m-2}$ which can be obtained from $c$ by removing both $c_k$ and $c_i$.

In the next result we will suppress, for notational convenience, the index 1 in the vector $(C_1, A_1)$.

**Lemma A.2.** *Let $(C, A)$ be a random vector with absolutely continuous distributions in $[\underline{C}, \bar{C}]^m \times [0, \bar{A}]^m$. Then the function L is twice differentiable, and for each $k = 1, \ldots, m$*

$$\frac{\partial L(\lambda)}{\partial \lambda_k} = \mathscr{E}(A_k X_k(\lambda)) - \beta_k,$$

$$\frac{\partial^2 L(\lambda)}{\partial \lambda_i \partial \lambda_k} = \begin{cases} \mathscr{E}_A\left(A_k A_i \int_{\underline{C}}^{\bar{C}} \cdots \int_{\underline{C}}^{\bar{C}} X_{ki}(\lambda) \right. \\ \qquad \left. f|_A\left(c_{(k)}, \min_{s \neq k}(c_s + \lambda_s A_s) - \lambda_k A_k\right) \mathrm{d}c_{(k)}\right) & \text{if } i \neq k, \\ \mathscr{E}_A\left(-A_k^2 \int_{\underline{C}}^{\bar{C}} \cdots \int_{\underline{C}}^{\bar{C}} \right. \\ \qquad \left. f|_A\left(c_{(k)}, \min_{s \neq k}(c_s + \lambda_s A_s) - \lambda_k A_k\right) \mathrm{d}c_{(k)}\right) & \text{if } i = k, \end{cases}$$

*where $X_k(\lambda)$ is a Boolean random variable taking the value 1 if $k = \arg\min_s(C_s + \lambda_s A_s)$ and 0 otherwise, $X_{ki}(\lambda)$ is a Boolean random variable taking the value 1 if $i = \arg\min_{s \neq k}(C_s + \lambda_s A_s)$ and 0 otherwise, and $f|_A$ is the density function of the vector $C$ conditional upon $A$.*

**Proof.** For notational simplicity, define

$$\tilde{L}(\lambda) = \mathscr{E}\left(\min_i(C_i + \lambda_i A_i)\right). \tag{A.6}$$

We will determine the first and second order partial derivatives of $\tilde{L}$. The first and second order partial derivatives of $L$ can then be determined from the following relationship between $L$ and $\tilde{L}$.

$$\frac{\partial L(\lambda)}{\partial \lambda_k} = \frac{\partial \tilde{L}(\lambda)}{\partial \lambda_k} - \beta_k,$$

$$\frac{\partial^2 L(\lambda)}{\partial \lambda_i \partial \lambda_k} = \frac{\partial^2 \tilde{L}(\lambda)}{\partial \lambda_i \partial \lambda_k}.$$

Function $\tilde{L}$ can be written as

$$\tilde{L}(\lambda) = \mathscr{E}_A \left( \sum_{i=1}^{m} \int_{\underline{C}}^{\bar{C}} \cdots \int_{\underline{C}}^{\bar{C}} \int_{\underline{C}}^{\min_{s\neq i}(c_s+\lambda_s A_s)-\lambda_i A_i} (c_i + \lambda_i A_i) f(c)\, \mathrm{d}c_i\, \mathrm{d}c_{(i)} \right),$$

where $f$ is the density function of vector $C$. Here we have assumed without loss of generality that the vectors $C$ and $A$ are independent. If they are not, then the density function $f$ should be replaced by $f|_A$, the density function of $C$ conditioned by $A$, throughout this proof.

By the Dominated Convergence Theorem, the first partial derivatives of $\tilde{L}$ with respect to $\lambda_k$ ($k = 1, \ldots, m$) are equal to

$$\begin{aligned}
&\frac{\partial \tilde{L}(\lambda)}{\partial \lambda_k} \\
&\quad = \frac{\partial}{\partial \lambda_k} \left( \mathscr{E}_A \left[ \int_{\underline{C}}^{\bar{C}} \cdots \int_{\underline{C}}^{\bar{C}} \int_{\underline{C}}^{\min_{s\neq k}(c_s+\lambda_s A_s)-\lambda_k A_k} (c_k + \lambda_k A_k) f(c)\, \mathrm{d}c_i\, \mathrm{d}c_{(i)} \right] \right. \\
&\qquad \left. + \mathscr{E}_A \left[ \sum_{i\neq k} \int_{\underline{C}}^{\bar{C}} \cdots \int_{\underline{C}}^{\bar{C}} \int_{\underline{C}}^{\min_{s\neq i}(c_s+\lambda_s A_s)-\lambda_i A_i} (c_i + \lambda_i A_i) f(c)\, \mathrm{d}c_i\, \mathrm{d}c_{(i)} \right] \right) \\
&\quad = \mathscr{E}_A \left[ \int_{\underline{C}}^{\bar{C}} \cdots \int_{\underline{C}}^{\bar{C}} \int_{\underline{C}}^{\min_{s\neq k}(c_s+\lambda_s A_s)-\lambda_k A_k} A_k f(c)\, \mathrm{d}c_i\, \mathrm{d}c_{(i)} \right] \\
&\qquad - \mathscr{E}_A \left[ \int_{\underline{C}}^{\bar{C}} \cdots \int_{\underline{C}}^{\bar{C}} A_k \min_{s\neq k}(c_s + \lambda_s A_s) f\left( c_{(k)}, \min_{s\neq k}(c_s + \lambda_s A_s) - \lambda_k A_k \right) \mathrm{d}c_{(k)} \right]
\end{aligned} \tag{A.7}$$

$$\begin{aligned}
&+ \sum_{i\neq k} \mathscr{E}_A \left[ \int_{\underline{C}}^{\bar{C}} \cdots \int_{\underline{C}}^{\bar{C}} \min_{s\neq i}(c_s + \lambda_s A_s) \frac{\partial}{\partial \lambda_k} \left( \min_{s\neq i}(c_s + \lambda_s A_s) \right) \right. \\
&\qquad \left. \times f\left( c_{(i)}, \min_{s\neq i}(c_s + \lambda_s A_s) - \lambda_i A_i \right) \mathrm{d}c_{(i)} \right].
\end{aligned} \tag{A.8}$$

We will show that terms (A.7) and (A.8) are equal, and thus their difference vanishes. We observe that (A.7) can be written as follows:

$$\mathscr{E}_A\left[\int_{\underline{C}}^{\bar{C}}\cdots\int_{\underline{C}}^{\bar{C}} A_k \min_{s\neq k}(c_s+\lambda_s A_s) f\left(c_{(k)}, \min_{s\neq k}(c_s+\lambda_s A_s)-\lambda_k A_k\right) \mathrm{d}c_{(k)}\right]$$

$$=\mathscr{E}_A\left[\sum_{i\neq k}\int_{\underline{C}}^{\bar{C}}\cdots\int_{\underline{C}}^{\min_{s\neq k,i}(c_s+\lambda_s A_s)-\lambda_i A_i} A_k(c_i+\lambda_i A_i)\right.$$

$$\left. f(c_{(k)}, c_i+\lambda_i A_i-\lambda_k A_k)\,\mathrm{d}c_i\,\mathrm{d}c_{(k,i)}\right]$$

$$=\mathscr{E}_A\left[\sum_{i\neq k}\int_{\underline{C}}^{\bar{C}}\cdots\int_{\underline{C}+\lambda_i A_i-\lambda_k A_k}^{\min_{s\neq k,i}(c_s+\lambda_s A_s)-\lambda_k A_k} A_k(c_k+\lambda_k A_k)\right.$$

$$\left. f(c_{(i)}, c_k+\lambda_k A_k-\lambda_i A_i)\,\mathrm{d}c_k\,\mathrm{d}c_{(i,k)}\right].$$

The first equality has been obtained by varying the index $i$ where $\min_{s\neq k}(c_s+\lambda_s A_s)$ is reached, and the second one by making a change of variables. With respect to (A.8), the partial derivative $(\partial/\partial\lambda_k)(\min_{s\neq i}(c_s+\lambda_s A_s))$ has value different from zero only when $\min_{s\neq i}(c_s+\lambda_s A_s)$ is reached at $s=k$. Thus, we have that

$$\sum_{i\neq k}\mathscr{E}_A\left[\int_{\underline{C}}^{\bar{C}}\cdots\int_{\underline{C}}^{\bar{C}}\left(\min_{s\neq i}(c_s+\lambda_s A_s)\right)\frac{\partial}{\partial\lambda_k}\left(\min_{s\neq i}(c_s+\lambda_s A_s)\right)\right.$$

$$\left. f\left(c_{(i)}, \min_{s\neq i}(c_s+\lambda_s A_s)-\lambda_i A_i\right)\mathrm{d}c_{(i)}\right]$$

$$=\mathscr{E}_A\left[\sum_{i\neq k}\int_{\underline{C}}^{\bar{C}}\cdots\int_{\underline{C}}^{\min_{s\neq k,i}(c_s+\lambda_s A_s)-\lambda_k A_k} A_k(c_k+\lambda_k A_k)\right.$$

$$\left. f(c_{(i)}, c_k+\lambda_k A_k-\lambda_i A_i)\,\mathrm{d}c_k\,\mathrm{d}c_{(i,k)}\right].$$

Thus (A.8) and (A.7) can be written as

$$\mathscr{E}_A\left[\sum_{i\neq k}\int_{\underline{C}}^{\bar{C}}\cdots\int_{\underline{C}}^{\underline{C}+\lambda_i A_i-\lambda_k A_k} A_k(c_k+\lambda_k A_k) f(c_{(i)}, c_k+\lambda_k A_k-\lambda_i A_i)\,\mathrm{d}c_k\,\mathrm{d}c_{(i,k)}\right]. \tag{A.9}$$

Now note that, for all $c_k\in[\underline{C},\underline{C}+\lambda_i A_i-\lambda_k A_k]$, $c_k+\lambda_k A_k-\lambda_i A_i\leqslant\underline{C}$, so that expression (A.9) equals 0. Thus the first partial derivatives of $\tilde{L}$ can be written as

$$\frac{\partial\tilde{L}(\lambda)}{\partial\lambda_k}=\mathscr{E}_A\left[\int_{\underline{C}}^{\bar{C}}\cdots\int_{\underline{C}}^{\bar{C}}\int_{\underline{C}}^{\min_{s\neq k}(c_s+\lambda_s A_s)-\lambda_k A_k} A_k f(c)\,\mathrm{d}c\right]=\mathscr{E}(A_k X_k(\lambda)).$$

The expression of the second order partial derivatives follow in a similar way. □

**Theorem 4.12.** *If the density of* $(C_1, A_1)$ *is strictly positive over a convex open set, and if Assumption* 4.11 *holds, then* $L$ *has a unique maximizer on the set* $\mathbb{R}_+^m$.

**Proof.** For notational convenience, we suppress the index 1 in the vector $(C_1, A_1)$. From the proof of Lemma A.1, we know that

$$\sup_{\lambda \in \mathbb{R}_+^m} L(\lambda) = \max_{\lambda \in B} L(\lambda),$$

where $B$ is a compact set defined by (A.2). Thus, the function $L$ has at least one maximizer $\lambda^*$ on $\mathbb{R}_+^m$. In the following, we will show uniqueness of this maximizer.

Denote by $I$ the set of non-active capacity constraints for $\lambda^*$ with dual multiplier equal to zero, that is

$$I = \{i\colon \lambda_i^* = 0,\ E[A_i X_i(\lambda^*)] < \beta_i\}.$$

From the sufficient second-order condition, it is enough to show that $H(\lambda^*)$, the Hessian of $L$ at $\lambda^*$, is negative definite on the subspace

$$M = \{y \in \mathbb{R}^m\colon\ y_l = 0,\ \forall l \in I\}.$$

Now let $y \in M$, $y \neq 0$, and evaluate the quadratic form associated with the Hessian of $L$ in $\lambda^*$:

$$\begin{aligned}
& y^\top H(\lambda^*) y \\
&= \sum_{\substack{k,i \notin I \\ i>k}} 2 y_k y_i \mathscr{E}_A \Bigg[ A_k A_i \int_{\underline{C}}^{\bar{C}} \cdots \int_{\underline{C}}^{\bar{C}} X_{ki}(\lambda^*) \\
&\quad \times f|_A \left( c_{(k)}, \min_{s \neq k}(c_s + \lambda_s^* A_s) - \lambda_k^* A_k \right) \mathrm{d}c_{(k)} \Bigg] \\
&\quad + \sum_{k \notin I} y_k^2 \mathscr{E}_A \left[ -A_k^2 \int_{\underline{C}}^{\bar{C}} \cdots \int_{\underline{C}}^{\bar{C}} f|_A \left( c_{(k)}, \min_{s \neq k}(c_s + \lambda_s^* A_s) - \lambda_k^* A_k \right) \mathrm{d}c_{(k)} \right] \\
&= -\sum_{\substack{k,i \notin I \\ i>k}} \mathscr{E}_A \Bigg[ (y_k A_k - y_i A_i)^2 \int_{\underline{C}}^{\bar{C}} \cdots \int_{\underline{C}}^{\bar{C}} X_{ki}(\lambda^*) \\
&\quad \times f|_A \left( c_{(k)}, \min_{s \neq k}(c_s + \lambda_s^* A_s) - \lambda_k^* A_k \right) \mathrm{d}c_{(k)} \Bigg] \\
&\quad - \sum_{k \notin I} y_k^2 \sum_{l \in I} \mathscr{E}_A \Bigg[ A_k^2 \int_{\underline{C}}^{\bar{C}} \cdots \int_{\underline{C}}^{\bar{C}} X_{kl}(\lambda^*) \\
&\quad \times f|_A \left( c_{(k)}, \min_{s \neq k}(c_s + \lambda_s^* A_s) - \lambda_k^* A_k \right) \mathrm{d}c_{(k)} \Bigg].
\end{aligned}$$

Since the vector $(C,A)$ has positive density on an open set, so does $A$, and then,

$$\mathscr{E}_A[(y_k A_k - y_i A_i)^2] > 0 \quad \text{if } (y_k, y_i) \neq (0,0).$$

To prove that $y^\top H(\lambda^*) y > 0$, it is enough to show that for each $k \notin I$ there exists a vector $(c_{(k)}, a)$ such that

$$f|_{A=a}\left(c_{(k)}, \min_{s \neq k}(c_s + \lambda_s^* a_s) - \lambda_k^* a_k\right) > 0$$

or, equivalently, there exists a vector $(c_{(k)}, a)$ such that

$$\mu_k(c_{(k)}, a) + \lambda_k^* a_k < \min_{s \neq k}(c_s + \lambda_s^* a_s) < v_k(c_{(k)}, a) + \lambda_k^* a_k,$$

where $(\mu_k(c_{(k)}, a), v_k(c_{(k)}, a))$ is the interval where $C_k$ has positive density when $(C_{(k)}, A) = (c_{(k)}, a)$. Now suppose that this vector does not exist. Then $\min_{s \neq k}(c_s + \lambda_s^* a_s) \leqslant \mu_k(c_{(k)}, a) + \lambda_k^* a_k$ or $\min_{s \neq k}(c_s + \lambda_s^* a_s) \geqslant v_k(c_{(k)}, a) + \lambda_k^* a_k$, for all vectors $(c_{(k)}, a)$ with positive density (since this set is convex and open). In the first case

$$\min_{s \neq k} c_s \leqslant \min_{s \neq k}(c_s + \lambda_s^* a_s) \leqslant \mu_k(c_{(k)}, a) + \lambda_k^* a_k = \mu_k(c_{(k)}, a).$$

But then $\mathscr{E}(A_k X_k(0)) = 0$, which contradicts Assumption 4.11(ii). In the second case, it can be deduced that $\mathscr{E}(A_k) = \mathscr{E}(A_k X_k(\lambda^*)) < \beta_k$, which contradicts Assumption 4.11(i). □

## References


[1] V. Balachandran, An integer generalized transportation model for optimal job assignment in computer networks, Oper. Res. 24 (4) (1976) 742–759.

[2] J.F. Benders, J.A.E.E. van Nunen, A property of assignment type mixed integer linear programming problems, Oper. Res. Lett. 2 (2) (1983) 47–52.

[3] D.G. Cattrysse, L.N. Van Wassenhove, A survey of algorithms for the generalized assignment problem, European J. Oper. Res. 60 (1992) 260–272.

[4] A. De Maio, C. Roveda, An all zero-one algorithm for a certain class of transportation problems, Oper. Res. 19 (6) (1971) 1406–1418.

[5] M. Dyer, A. Frieze, Probabilistic analysis of the generalised assignment problem, Math. Programming 55 (1992) 169–181.

[6] M.L. Fisher, R. Jaikumar, A generalized assignment heuristic for vehicle routing, Networks 11 (1981) 109–124.

[7] M.L. Fisher, R. Jaikumar, L.N. Van Wassenhove, A multiplier adjustment method for the generalized assignment problem, Management Sci. 32 (9) (1986) 1095–1103.

[8] A. Geoffrion, G.W. Graves, Multicommodity distribution system design by Benders decomposition, Management Sci. 20 (5) (1974) 822–844.

[9] A.K. Lenstra, J.K. Lenstra, A.H.G. Rinnooy Kan, T.J. Wansbeek, Two lines least squares, Ann. Discrete Math. 16 (1982) 201–211.

[10] S. Martello, P. Toth, An algorithm for the generalized assignment problem, in: J.P. Brans (Ed.), Operational Research, IFORS, North-Holland, Amsterdam, 1981, pp. 589–603.

[11] S. Martello, P. Toth, Knapsack Problems, Algorithms and Computer Implementations, Wiley, New York, 1990.

[12] M. Meanti, A.H.G. Rinnooy Kan, L. Stougie, C. Vercellis, A probabilistic analysis of the multi-knapsack value function, Math. Programming 46 (1990) 237–247.

[13] A.H.G. Rinnooy Kan, L. Stougie, C. Vercellis, A class of generalized greedy algorithms for the multi-knapsack problem, Discrete Appl. Math. 42 (1993) 279–290.

[14] R.T. Rockafellar, Convex Analysis, Princeton University Press, Princeton, NJ, 1970.
[15] H.E. Romeijn, N. Piersma, A probabilistic feasibility and value analysis of the generalized assignment problem, ERASM Management Report Series No. 293, Rotterdam School of Management, Erasmus University Rotterdam, 1996.
[16] G.T. Ross, R.M. Soland, A Branch and Bound Algorithm for the generalized assignment problem, Math. Programming 8 (1975) 91–103.
[17] G.T. Ross, R.M. Soland, Modeling facility location problems as generalized assignment problems, Management Sci. 24 (3) (1977) 345–357.