Computers and Mathematics with Applications 60 (2010) 1115–1123

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

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

Lagrangian heuristic for a class of the generalized assignment problems

Igor Litvinchev [a,*], Miguel Mata [b], Socorro Rangel [c], Jania Saucedo [b]

[a] *Dorodnicyn Computing Center Russian Academy of Sciences, Moscow, Russia*

[b] *Faculty of Mechanical and Electrical Engineering, UANL - Nuevo Leon State University, Mexico*

[c] *Department of Computer Science and Statistics, UNESP - Sao Paulo State University, Brazil*

ARTICLE INFO

Keywords:
Lagrangian heuristic
Integer programming
Many-to-many assignment problem

ABSTRACT

A Lagrangian based heuristic is proposed for many-to-many assignment problems taking into account capacity limits for task and agents. A modified Lagrangian bound studied earlier by the authors is presented and a greedy heuristic is then applied to get a feasible Lagrangian-based solution. The latter is also used to speed up the subgradient scheme to solve the modified Lagrangian dual problem. A numerical study is presented to demonstrate the efficiency of the proposed approach.

1. Introduction

Lagrangian relaxation is a powerful tool to exploit structural properties of large optimization problems and to derive bounds for the optimal objective [1,2]. These bounds are widely used as a core of many numerical techniques and also provide a measure for the progress of the main algorithm. A Lagrangian solution is frequently used as a starting or reference point for various heuristics and approximate techniques. The literature on Lagrangian relaxation is quite extensive. We refer only to a few survey papers: [3–7].

To form a Lagrangian relaxation, "complicating" constraints are relaxed and a penalty term is added to the objective to discourage their violations. The classical measure of the deviations is a complementarity term—a linear combination of the constraint slacks with the coefficients called Lagrange multipliers. The optimal value of the Lagrangian problem, considered for fixed multipliers, provides a lower bound (for minimization problem) to the original optimal objective. This estimation is derived using the nonnegativity of the complementarity term.

An approach to tightening Lagrangian bounds by a more precise estimation of the penalty term arising in the Lagrangian problem was proposed in [8] and further developed in [9]. It is well known that under certain convexity and regularity conditions, the penalty turns to zero for the optimal primal–dual solution (complementarity condition). However, for nonconvex problems the complementarity condition is not necessarily fulfilled. An auxiliary optimization problem is used to estimate the penalty term and to construct the modified Lagrangian bound and the corresponding modified dual problem. The new bounds are numerically studied and compared with classical ones for a class of many-to-many assignment problem. The latter is a generalization of the assignment problem taking into account capacity limits for tasks and agents [10]. In contrast to the classical generalized assignment problem, taking into account the only agent's capacities, the many-to-many assignment problem is much less investigated [11]. For the instances used in the computational tests performed in [9], the modified Lagrangian problem provided high-quality bounds, typically within 0.5% of relative suboptimality. As a byproduct of the bound computation, an integer Lagrangian solution was obtained having a higher degree of primal feasibility and suboptimality than the standard Lagrangian solution.

[*] Corresponding author.
E-mail addresses: igorlitvinchev@gmail.com (I. Litvinchev), miguel@yalma.fime.uanl.mx (M. Mata), socorro@ibilce.unesp.br (S. Rangel), jania@yalma.fime.uanl.mx (J. Saucedo).

0898-1221/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2010.03.070

In this paper we apply a simple greedy heuristic to the (unfeasible) Lagrangian solution to get an approximate solution to the original problem. The numerical tests show that combining the modified Lagrangian solution with the greedy heuristic provides high quality approximate feasible solutions within 0.5% of the relative difference between dual and primal bounds. To solve a corresponding dual problem the subgradient algorithm is used. Similar to [12] the greedy solution was obtained in each iteration of the subgradient algorithm and then was used to update parameters of the subgradient technique. Incorporating a feasible greedy solution into the subgradient scheme results in a significant decrease in the number of iterations without dropping the quality of the bounds.

The reminder of the paper is organized as follows. In Section 2 we review the basic constructions to derive the modified Lagrangian bounds. The greedy and subgradient algorithm related to the many-to-many assignment problem are presented in Section 3. Numerical results are given in Section 4 and Section 5 concludes.

2. Deriving the modified bound

Let the original problem be stated in the form:

$$z^* = \max\{cx \mid Dx \le d, Ax \le b, x \in U\}, \tag{1}$$

where $x \in R^n$. The set U can be of general structure and may contain, for example, sign constraints on x and integrality constraints on some or all components of x. The condition $Dx \le d$ are m "complicating constraints", while the constraints $Ax \le b$, are "nice" in the sense that the optimization problem formed with only these constraints, together with $x \in U$, is easier than the original problem. Denote by x^* an optimal solution of (1). In what follows we denote $X = \{x \in U \mid Ax \le b\}$.

Let $u = \{u_i\} \ge 0$ be an m-vector of Lagrangian multipliers. The standard Lagrangian problem is defined as:

$$z(u) = \max\{cx + u(d - Dx) \mid x \in X\}.$$

We assume for simplicity that it has an optimal solution for all $u \ge 0$. This problem yields the well known Lagrangian bound:

$$z^* \le z(u), \quad \forall u \ge 0, \tag{2}$$

while the best Lagrangian bound and the associated Lagrangian multipliers u^* are defined from the Lagrangian dual problem:

$$z(u^*) = \min_{u \ge 0} z(u) \equiv w_D. \tag{3}$$

In what follows we consider constructions leading to dual bounds, which are at least as good as the standard Lagrangian bounds. We assume that a certain information about an optimal solution to (1), x^*, is known:

Assumption. A set $W \subseteq R^n$ is known, such that $x^* \in W$.

We will refer to W as the *localization of* x^*, or simply the *localization*. The set W can be defined by manipulating the constraints of the original problem, by querying a decision maker, etc.

The feasibility of x^* to the standard Lagrangian problem implies that $cx^* + u(d - Dx^*) \le z(u)$. Since x^* is feasible to (1) and $u \ge 0$, then the condition $u(d - Dx^*) \ge 0$ yields immediately that $z^* \le z(u)$ as in (2). For the convex problem (1), under certain regularity assumptions, the complementarity condition $u^*(d - Dx^*) = 0$ is fulfilled. However, for $u \ne u^*$, the complementary term, $u(d - Dx^*)$, can be strictly positive. For the nonconvex case the complementarity term, $u(d - Dx^*)$, can be strictly positive for $u = u^*$. Thus, we may try to strengthen the standard Lagrangian bound, $z(u)$, using tighter estimations of the complementarity term instead of $u(d - Dx^*) \ge 0$.

Since $x^* \in W$, then for any localization W we have:

$$z(u) \ge cx^* + u(d - Dx^*) \ge cx^* + \min_{y \in W} u(d - Dy),$$

which gives immediately:

$$z^* \le \max_{x \in X}\{cx + u(d - Dx)\} - \min_{y \in W} u(d - Dy). \tag{4}$$

To simplify further notations, we use $\theta(u)$ for the optimal objective value of the second maximization problem in (4). We may expect that, for those integer programs where constraints $Dx \le d$ are not fulfilled as equalities for all feasible solutions, a reasonable choice of W may result in $\theta(u) = \min_{y \in W} u(d - Dy) > 0$, thus improving the standard Lagrangian bound.

Calculating $\theta(u)$ so far was solely intended to estimate the complementarity term associated with the $Dx \le d$, without taking into account the original objective function. We may "balance" $z(u)$ and $\theta(u)$ by introducing the original objective function when calculating $\theta(u)$ (see [9] for more details). Finally we get the modified Lagrangian bound:

$$z^* \le z_M(u, \pi) = \max_{x \in X}\{(1 - \pi)cx - uDx\} + \max_{y \in W}\{\pi cy + uDy\}. \tag{5}$$

To simplify further notations, we use $\eta(\pi, u)$ for the optimal objective value of the first maximization problem in (5), while $\xi(\pi, u)$ stands for the optimal objective of the second. Note that, by our assumption, calculating $\eta(\pi, u)$ is easy since the corresponding problem contains only "nice" constraints. The modified dual problem corresponding to the bound (5) is then stated as:

$$w_{MD} = \min_{u\geq 0,\pi} \{\eta(\pi, u) + \xi(\pi, u)\}. \tag{6}$$

From the definition of $z_M(\pi, u)$ it follows that, in general, the tighter the localization W is, the smaller $\xi(\pi, u)$ is and the better the modified upper bound $z_M(\pi, u)$ is. From this point of view it is worth to retain in the definition of W as many original constraints as possible, which can complicate calculation of $\xi(\pi, u)$. Instead, we may estimate $\xi(\pi, u)$ using Lagrangian relaxation for the corresponding maximization problem. We will refer to this case as the nested Lagrangian relaxation. Suppose that the localization has the form $W_1 = \{y \mid Py \leq p, y \in Y \cap U\}$, where $Y \subseteq R^n$, $p \in R^q$, and the matrix P is dimensioned accordingly. We assume that the set Y has a favorable structure (for example, decomposable) and we will handle the constraints $y \in Y$ explicitly, while the constraints $Py \leq p$ will be dualized using a q-vector of multipliers $v \geq 0$. Estimating $\xi(\pi, u)$ by the standard Lagrangian bound yields:

$$\begin{aligned} \xi(\pi, u) &\equiv \max_{y\in W_1}\{\pi cy + uDy\} \\ &\leq \max_{y\in Y\cap U}\{\pi cy + uDy + v(p - Py)\} \\ &\equiv \xi^L(\pi, u, v) \quad \forall v \geq 0. \end{aligned}$$

Based on the estimations of $\xi(\pi, u)$ we get the modified dual problem associated with the nested Lagrangian relaxation for localization W_1:

$$w^L_{MD}(W_1) = \min_{u,v\geq 0,\pi} \{\eta(\pi, u) + \xi^L(\pi, u, v)\}. \tag{7}$$

The bound (7) involves $m + q + 1$ multipliers: a m-vector u, a q-vector v and a scalar π.

A critical issue is constructing a suitable localization which is tight enough and results in "easy" calculation of ξ or ξ^L. A detailed discussion on this subject, as well as a comparison among the modified bound and other Lagrangian type bounds can be found in [9]. Here we mention only that if the original problem has two groups of interesting constraints, then we may define X by one group of constraints, while using the other group to define the localization W. In this case both subproblems in (7) has an attractive (e.g., decomposable) property. Such a structure of the original problem, called "double decomposable" can be found in the generalized assignments problem, the multiple knapsack problem and the facility location problem, to mention a few.

3. A Lagrangian based greedy heuristic for the many-to-many assignment problem

The Assignment Problems (AP) involve optimally matching the elements of two or more sets. When there are only two sets, they may be referred as "tasks" and "agents". In the generalized assignment problem (GAP) each task is assigned to one agent, as in the classic AP, but it allows that an agent may be assigned to more than one task, while recognizing that a task may use only part of an agent's capacity rather than all of it (see e.g. [11,10] and the references therein). The many-to-many assignment problem is a further generalization of AP which takes into account capacity limits of both tasks and agents. Such a situation arises, for example, in a medical center, where doctors (agents) have to attend their patients (tasks) in a limited time period, while patients cannot also spend a lot of time in the center.

The many-to-many assignment problem (MMAP) can be stated as:

$$z_{ip} = \max \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} a_{ij}x_{ij} \leq b_i, \quad i = 1, 2, \ldots, m, \tag{8}$$

$$\sum_{i=1}^{m} d_{ij}x_{ij} \leq d_j, \quad j = 1, 2, \ldots, n, \tag{9}$$

$$x_{ij} \in \{0, 1\},$$

where $x_{ij} = 1$ if agent i is assigned to task j, and 0 otherwise, c_{ij} is the profit (utility) of assigning agent i to task j, a_{ij} is the amount of agent i's capacity used to execute task j, and b_i is the available capacity of agent i. We also assume that each task has its own capacity (time) limit, such that d_{ij} is the amount of task j's capacity used when executed by agent i, and d_j is the available capacity of task j.

Note that (MMAP) has a double-decomposable structure: if we dualize the first m constraints, then the relaxed problem decomposes into n independent subproblems, each having a single knapsack-type constraint $\sum_{i=1}^{m} d_{ij}x_{ij} \leq d_j$, while relaxing the second group of constraints we get m single knapsack constrained subproblems.

To derive the modified Lagrangian bounds (see Section 2) for the (MMAP) we include the constraints (8) in the set X and will treat them as "easy", while constraints (9) are included in Y and will be treated as "complicating". A set $\{Px \leq p\}$ used in the localization W_1 for calculating $w^L_{MD}(W_1)$ is defined by (8). The constraints $y \in Y$ will be handled explicitly, while $Py \le p$ will be dualized in the estimation $\xi^L(\pi, u, v)$. We define:

$$X = \left\{x_{ij} \in \{0,1\} \mid \sum_{j=1}^{n} a_{ij}x_{ij} \le b_i,\ i = 1, 2, \dots, m\right\} = \prod_{i=1}^{m} X_i,$$
$$X_i = \left\{x_{ij} \in \{0,1\} \mid \sum_{j=1}^{n} a_{ij}x_{ij} \le b_i\right\},$$
$$W_1 = \{y_{ij} \in \{0,1\} \mid y \in Y,\ Py \le p\},$$
$$Y = \left\{y_{ij} \in \{0,1\} \mid \sum_{i=1}^{m} d_{ij}y_{ij} \le d_j,\ j = 1, 2, \dots, n\right\} = \prod_{j=1}^{n} Y_j,$$
$$Y_j = \left\{y_{ij} \in \{0,1\} \mid \sum_{i=1}^{m} d_{ij}y_{ij} \le d_j\right\},$$
$$\{Py \le p\} \equiv \left\{\sum_{j=1}^{n} a_{ij}y_{ij} \le b_i,\ i = 1, 2, \dots, m\right\}.$$

Let $u = \{u_j, j = 1, 2, \dots, n\}$ and $v = \{v_i, i = 1, 2, \dots, m\}$ be the Lagrangian multipliers. Then modified Lagrangian dual for (MMAP) is:

$$\bar{w}_{MD}(W_1) = \min_{u,v\ge 0,\pi} \varphi(\pi, u, v) \tag{10}$$

where:

$$\varphi(\pi, u, v) = \eta(\pi, u) + \xi^L(\pi, u, v),$$
$$\eta(\pi, u) = \sum_i \max_{x\in X_i} \left\{\sum_j [(1-\pi)c_{ij} - u_j d_{ij}]x_{ij}\right\},$$
$$\xi^L(\pi, u, v) = \sum_j \max_{y\in Y_j} \left\{\sum_i (\pi c_{ij} + u_j d_{ij} - v_i a_{ij})y_{ij}\right\} + \sum_i v_i b_i.$$

The modified Lagrangian dual (10) provided high-quality bounds for the (MMAP) for the instances testes in [9]. It turned out that the corresponding integer Lagrangian solutions (x, y) had a higher degree of primal feasibility and suboptimality than the standard Lagrangian solution. In the next section we consider a greedy algorithm to recover primal feasibility. The feasible solution is then used in a subgradient algorithm to obtain the modified bounds.

3.1. The Lagrangian based greedy heuristic

To get a feasible Lagrangian based solution we use a simple greedy approach. First we try to decrease to zero some components currently equal to 1 to obtain a feasible solution. The choice of the candidate component is based on the smallest decrease of a rounding indicator (e.g. minimal cost component). After a feasible solution is obtained we try to increase to 1 some zero components based on the largest increase of another rounding indicator (e.g., maximal cost component) while maintaining feasibility.

Let x^0 be a current binary point not necessary feasible to (MMAP). Let Ω_0 be a set of all pairs (i, j) with $x^0_{ij} = 0$ and Ω_1 be a set of all pairs (i, j) with $x^0_{ij} = 1$. Denote

$$\delta_i = b_i - \sum_{j=1}^{n} a_{ij}x^0_{ij}, \qquad \sigma_j = d_j - \sum_{i=1}^{m} d_{ij}x^0_{ij}.$$

If $\min_{i,j}\{\delta_i, \sigma_j\} \ge 0$, then x^0 is feasible to (MMAP). Otherwise, we first decrease, in a greedy manner, some positive x^0_{ij} to 0 to get a feasible solution (x^{gr}). Then we try to improve this feasible solution by increasing, in a greedy fashion, some zero components to 1. A summary of the Greedy Algorithm is given by Algorithm 1.

Algorithm 1

```
1. **Let** $x^0$ be a Lagrangian solution ($x$ or $y$). Set $x^{gr} = x^0$.
2. **Set** $t_{ij}$, $r_{ij}$ as the rounding indicators (*e.g.* $t_{ij} = c_{ij}$, $r_{ij} = c_{ij}$,
   see **Comment 1** below).
3. **Feasibility** test.
```

```
    **For** $x^0$ compute $\delta_i, \sigma_j$.
    **while** $\min_{i,j}\{\delta_i, \sigma_j\} \le 0$ **do** (rounding down)
        Compute: $\min_{(i,j)\in\Omega_1} t_{ij}$. Let this min be attained for $(i,j)'$.
        **Set**:
            $x_{ij}^{gr} = 0$ for $(i,j) = (i,j)'$,
            $\Omega_1 = \Omega_1 \setminus (i,j)'$ and $\Omega_0 = \Omega_0 \cup (i,j)'$,
            $\delta_i = \delta_i + a_{(ij)'}, \sigma_j = \sigma_j^k + d_{(ij)'}$.
    **end_while**
    **Let** $x^{gr}$ a feasible solution obtained in the rounding down step,
    $S_0 \subseteq \Omega_0$ be a set of $(i,j) \in \Omega_0$ with both $a_{ij} \le \delta_i$ and $d_{ij} \le \sigma_j$.
    **while** $S_0 \neq \emptyset$ **do** (rounding up)
        **Compute** $\max_{(i,j)\in S_0} r_{ij}$. Let this max be attained for $(i,j)'$.
        **Set**:
            $\delta_i = \delta_i - a_{(ij)'}, \sigma_j = \sigma_j - d_{(ij)'}$,
            $\Omega_1 = \Omega_1 \cup (i,j)'$ and $\Omega_0 = \Omega_0 \setminus (i,j)'$,
            $x_{ij}^{gr} = 1$ for $(i,j) = (i,j)'$.
        **update** $S_0$.
    **end_while**
  **Return** greedy solution $x^{gr}$.
**end Algorithm 1**
```

Comment 1. The *rounding down* part of Algorithm 1 may be based on pure cost criterion $\min_{(i,j)\in\Omega_1} t_{ij}$ for $t_{ij} = c_{ij}$. It is possible to use another indicator, setting for example, $t_{ij} = c_{ij}/\max(a_{ij}, d_{ij})$. In this way we can take into account the impact of the component (i,j) in violating the constraints (the larger values of a_{ij}, d_{ij} the faster we get feasibility). Similarly, we can try $t_{ij} = c_{ij}/\max\{(a_{ij}/b_i), (d_{ij}/d_j)\}$, since relative values $(a_{ij}/b_i), (d_{ij}/d_j)$ also give a measure of feasibility. In the *rounding up* part of the Algorithm 1 we may use $t_{ij} = r_{ij}$. Alternatively, we can use $r_{ij} = c_{ij}/\min(a_{ij}, d_{ij})$. Small values of a_{ij}, d_{ij} help towards a small degradation in the solution's feasibility. Another possibility is to set $r_{ij} = c_{ij}/\min\{(a_{ij}/b_i), (d_{ij}/d_j)\}$.

The Lagrangian solution is always feasible either to the first or to the second group of constraints of problem (MMAP). So we can simplify Algorithm 1 by considering only $\delta_i(\sigma_j)$ when rounding down, depending on whether x or y is used for rounding the modified Lagrangian solution.

3.2. Solving the modified dual problem

A popular approach to solve the dual problem is by subgradient optimization, first used in the Lagrangian context in [13]. Here we present the basic steps of the subgradient technique used in [9] to calculate $\bar{w}_{MD}(W_1)$ and modified to use the greedy solution obtained by Algorithm 1. A more detailed discussion of subgradient optimization can be found in [2,14].

Let π^k, u^k, v^k be the values of the Lagrangian multipliers in the kth iteration, $\varphi^k = \varphi(\pi^k, u^k, v^k)$, and x_{ij}^k, y_{ij}^k be the associated subproblems solutions:

$$x_{ij}^k = \arg\max_{x\in X_i}\left\{\sum_j [(1-\pi^k)c_{ij} - u_j^k d_{ij}]x_{ij}\right\},$$

$$y_{ij}^k = \arg\max_{y\in Y_j}\left\{\sum_i [\pi^k c_{ij} + u_j^k d_{ij} - v_i^k a_{ij}]y_{ij}\right\}.$$

A subgradient is directly identified after solving the subproblems as:

$$\gamma^k = [\partial\varphi/\partial\pi]^k = -\sum_i\sum_j c_{ij}x_{ij}^k + \sum_j\sum_i c_{ij}y_{ij}^k,$$

$$\alpha_i^k = [\partial\varphi/\partial v_i]^k = b_i - \sum_j a_{ij}y_{ij}^k,$$

$$\beta_j^k = [\partial\varphi/\partial u_j]^k = -\sum_i d_{ij}x_{ij}^k + \sum_i d_{ij}y_{ij}^k.$$

Denote by s^k a vector composed of all $\{\gamma^k, \alpha_i^k, \beta_j^k\}$, let $\lambda^k = \{\pi^k, v^k, u^k\}$ and set:

$$\bar{\lambda}^{k+1} = \lambda^k - \varepsilon_k(\varphi^k - \varphi_{lb})\frac{s^k}{\left\|s^k\right\|^2}, \tag{11}$$

where $\varepsilon_k \in (0, 2]$, φ_{lb} is a lower bound on $\varphi^* = \bar{w}_{MD}(W_1)$. Since $z_{ip} \le \bar{w}_{MD}(W_1)$ we may set φ_{lb} equal to the objective function value of (MMAP) associated to a given feasible solution. In what follows we will apply the greedy algorithm in each iteration to get a feasible solution and update φ_{lb} accordingly.

The multipliers for the next iteration are defined as the projection of $\bar{v}^{k+1}, \bar{u}^{k+1}$ onto the nonnegative orthant since $u, v \geq 0$, while π has no sign restrictions:

$$\begin{aligned}
&\pi^{k+1} = \bar{\pi}^{k+1},\\
&v^{k+1} = \max\{0; \bar{v}^{k+1}\} \text{ where } \max \text{ is taken componentwise,}\\
&u^{k+1} = \max\{0; \bar{u}^{k+1}\} \text{ where } \max \text{ is taken componentwise.}
\end{aligned}$$

A summary of the subgradient algorithm to compute the modified bound for the (MMAP) is given in Algorithm 2.

Algorithm 2

```
Given initial values for $\pi^0, u^0, v^0$.
Set $\varphi_{ub}^0 = \infty$ and $\varphi_{lb}^0 = -\infty$.
While (not stop) do
    Compute
        $x_{ij}^k = \arg\max_{x \in X_i} \left\{ \sum_j [(1-\pi^k)c_{ij} - u_j^k d_{ij}]x_{ij} \right\}$,
        $y_{ij}^k = \arg\max_{y \in Y_j} \left\{ \sum_i (\pi^k c_{ij} + u_j^k d_{ij} - v_i^k a_{ij}) y_{ij} \right\}$,
        $\varphi^k$.
    Use Algorithm 1 to obtain feasible solutions $x^{gr}$ and $y^{gr}$
    with objective values $cx^{gr}$ and $cy^{gr}$ respectively.
    Let: $\varphi_{lb}^{k+1} = \max\{\varphi_{lb}^k, cx^{gr}, cy^{gr}\}$, $\varphi_{ub}^{k+1} = \min\{\varphi_{ub}^k, \varphi^k\}$.
    Update
        $\gamma^k = [\partial\varphi/\partial\pi]^k = -\sum_i \sum_j c_{ij} x_{ij}^k + \sum_j \sum_i c_{ij} y_{ij}^k$,
        $\alpha_i^k = [\partial\varphi/\partial v_i]^k = b_i - \sum_j a_{ij} y_{ij}^k$,
        $\beta_j^k = [\partial\varphi/\partial u_j]^k = -\sum_i d_{ij} x_{ij}^k + \sum_i d_{ij} y_{ij}^k$.
    Compute $\bar{\lambda}^{k+1}$ according to (11) with $\varphi_{lb} = \varphi_{lb}^{k+1}$.
    Project $\bar{v}^{k+1}, \bar{u}^{k+1}$ onto the nonnegative orthant.
    Make stop tests.
end_while
end Algorithm 2
```

The subgradient method is not monotone, that is, it is not necessary that $\varphi^k \geq \varphi^{k+1}$. In practice, the parameter ε_k is varying in $(0, 2]$, beginning with $\varepsilon_k = 2$. If after K consecutive iterations with a fixed value for ε_k the function φ is not improved "sufficiently", then a smaller value of ε_k is used, say, a half of ε_k. The stopping criteria used in Algorithm 2 are: (a) maximum iteration number is reached; (b) ε_k is already small enough; or (c) the relative difference between the best integer feasible solution found so far and the Lagrangian bound is within a given threshold.

4. Numerical results

In this section we numerically compare the Lagrangian bounds, standard and modified, and corresponding greedy solution for the same two sets of instances of (MMAP) as in [9]: small instances with sizes $m \times n$ for $m \in \{5, 8, 10\}$ and $n = 50$, and large instances with $m \in \{5, 10, 20\}$ and $n = 100$. The data were random integers with:

$$b_i = \alpha \left(\sum_j a_{ij} - 1 \right), \qquad d_j = \alpha \left(\sum_j d_{ij} - 1 \right), \quad 0 < \alpha \leq 1,$$

and divided in three classes (a, b, and c) with respect to the values of α: a $(\alpha = 1)$, b $(\alpha = 0.9)$, c $(\alpha = 0.8)$. More details of the data generation can be found in [9].

All Lagrangian-type bounds, standard and modified, were calculated by the subgradient method given by Algorithm 2 presented in Section 3.2. We used $K = 5$, and if $(\varphi^k - \varphi^{k+1})/\varphi_{ub}^{k+1} \leq 0.005$ for 5 consecutive iterations with fixed ε_k, this parameter was updated to $\varepsilon_{k+1} = \varepsilon_k/2$. The stopping criteria for the Algorithm 2 were specified as follows: (a) at most 250 iterations were permitted; (b) the runs stop if $\varepsilon_{k+1} \leq 0.005$; or (c) $(\varphi_{ub}^{k+1} - \varphi_{lb}^{k+1})/\varphi_{lb}^{k+1} \leq 0.0001$. All optimization subproblems associated with a subgradient algorithm were solved by the system CPLEX 10.0 [15]. The runs were executed on a machine AMD Athlon 64X2 Dual Core, 2.8 GHz and 2048 MB RAM.

Table 1
Relative quality of the bounds (results for small problems).

m	n	cl	gap_{md}^{gr}	iter	gap_{md}	iter	gap_{lag}^{gr}	iter	gap_{lag}	iter	gap_{lp}
5	50	a	0.00	5	0.00	165	10.91	55	11.09	100	11.01
5	50	b	0.00	5	0.00	80	7.22	74	7.50	125	7.18
5	50	c	0.12	55	0.36	110	3.29	80	3.61	85	3.30
8	50	a	0.00	5	0.00	80	5.62	55	5.76	110	6.00
8	50	b	0.09	75	0.12	125	2.19	95	2.49	200	2.21
8	50	c	0.11	55	0.32	75	3.17	80	3.42	145	3.22
10	50	a	0.00	5	0.00	130	3.96	55	4.08	65	4.39
10	50	b	0.07	65	0.28	70	1.42	74	1.62	115	1.49
10	50	c	0.20	55	0.51	170	1.63	67	1.86	115	1.73

For all problem instances we have calculated:

z_{ip} optimal objective of the original integer problem,

z_{lp} optimal objective of the LP relaxation,

z_{lag} classical Lagrangian bound w_D,

z_{lag}^{gr} classical Lagrangian bound w_D computed by Algorithm 2,

z_{md} modified Lagrangian bound $\bar{w}_{MD}(W_1)$ computed as in [9],

z_{md}^{gr} modified Lagrangian bound $\bar{w}_{MD}(W_1)$ computed by Algorithm 2,

thus obtaining five upper bounds for z_{ip}. The bounds z_{lag}, z_{md} were computed similar to Algorithm 2, but without applying greedy Algorithm 1 and updating φ_{lb} at each iteration. For z_{lag}^{gr} the greedy solution obtained by Algorithm 1 was used in the subgradient method in a way similar to Algorithm 2.

The proximity of the bounds to the optimal integer solution was represented by:

$$\text{gap}_{md}^{gr} = \frac{z_{md}^{gr} - z_{ip}}{z_{ip}} 100\%, \qquad \text{gap}_{md} = \frac{z_{md} - z_{ip}}{z_{ip}} 100\%,$$

$$\text{gap}_{lag}^{gr} = \frac{z_{lag}^{gr} - z_{ip}}{z_{ip}} 100\%, \qquad \text{gap}_{lag} = \frac{z_{lag} - z_{ip}}{z_{ip}} 100\%, \quad \text{and} \quad \text{gap}_{lp} = \frac{z_{lp} - z_{ip}}{z_{ip}} 100\%.$$

The results for the small instances are reported in Table 1, while Table 2 presents results for the larger instances. The column iter, next to the column of the corresponding proximity indicator, presents the number of iterations of the subgradient technique necessary to meet a stopping criterion. For all instances with $\text{gap}_{md}^{gr} = \text{gap}_{md} = 0.00$ the stopping criterion (c) was fulfilled (see also $\text{gap}_{\text{bound}}$ in Tables 3 and 4). For all other instances runs were terminated by the stopping criterion (b). For the problem instance 20 × 100c we were not be able to find the optimal solution, CPLEX aborted due to insufficient memory. The best integer solution found after examining 316,560 nodes in the branch and cut tree (in AMPL/CPLEX notations, absmipgap = 27.0845, relmipgap = 0.000494234) was used then to calculate the indicators.

As can be seen from Tables 1 and 2, incorporating greedy solution in the subgradient scheme slightly improves the quality (of the approximate values) of the bounds obtained by the subgradient method, but the effect is rather modest. For all problem instances the modified bound (gap_{md}^{gr}) is significantly tighter than the classical Lagrangian bounds (gap_{lag}^{gr}, gap_{lag}). Moreover, the number of iterations of the subgradient method reduces significantly by using a greedy solution. This takes place for all problem instances. Note that the computational cost for one iteration of the subgradient technique (solving integer Lagrangian problems) is much higher than the one for obtaining a greedy solution (simply reordering data). Thus combining the subgradient method with a "cheap feasibility recovering" technique is favorable for this class of problem. A similar effect was mentioned in the papers on Lagrangian heuristics for the generalized assignment problems [16,17] and, for more general settings, in [12].

A feasible Lagrangian-based solution was derived by the greedy technique given by Algorithm 1 (see Section 3.1). In this way there are two Lagrangian-based solutions: one for the classical bound and one for the modified bound. The quality of the (best over all subgradient iterations) feasible solution σ is presented in Tables 3 and 4 and was measured according to:

$$\text{gap}_{ip}(\sigma) = \frac{z_{ip} - \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}\sigma_{ij}}{z_{ip}} 100\%.$$

Along with $\text{gap}_{ip}(\sigma)$, the following indicator was used for each bound:

$$\text{gap}_{\text{bound}}(\sigma) = \frac{z_{\text{bound}} - \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}\sigma_{ij}}{z_{\text{bound}}} 100\%,$$

Table 2
Relative quality of the bounds (results for large problems).

m	n	cl	gap_{md}^{gr}	iter	gap_{md}	iter	gap_{lag}^{gr}	iter	gap_{lag}	iter	gap_{lp}
5	100	a	0.00	5	0.00	190	11.28	75	11.16	95	11.10
5	100	b	0.00	5	0.00	100	6.53	49	6.84	110	6.49
5	100	c	0.02	70	0.03	60	4.48	75	4.76	110	4.45
10	100	a	0.00	5	0.00	175	4.31	60	4.41	105	4.38
10	100	b	0.04	65	0.07	55	1.21	70	1.53	175	1.22
10	100	c	0.11	70	0.17	85	1.99	79	2.19	65	1.99
20	100	a	0.00	5	0.00	85	1.61	49	1.65	90	1.83
20	100	b	0.07	70	0.08	105	0.76	65	0.91	135	0.78
20	100	c	0.11	80	0.79	180	0.59	49	0.73	130	0.62

Table 3
Quality of the greedy solution (results for small problems).

Type			Greedy + Lagr		Greedy + ModLagr	
m	n	cl	gap_{ip}	$\text{gap}_{\text{bound}}$	gap_{ip}	$\text{gap}_{\text{bound}}$
5	50	a	2.33	11.94	0.00	0.00
5	50	b	1.31	7.95	0.00	0.00
5	50	c	0.44	3.61	0.17	0.30
8	50	a	0.82	6.09	0.00	0.00
8	50	b	0.19	2.33	0.11	0.20
8	50	c	0.77	3.83	0.02	0.13
10	50	a	0.60	4.39	0.00	0.00
10	50	b	0.33	1.72	0.08	0.15
10	50	c	0.61	2.20	0.51	0.71

Table 4
Quality of the greedy solution (results for large problems).

Type			Greedy + Lagr		Greedy + ModLagr	
m	n	cl	gap_{ip}	$\text{gap}_{\text{bound}}$	gap_{ip}	$\text{gap}_{\text{bound}}$
5	100	a	1.97	11.91	0.00	0.00
5	100	b	0.61	6.70	0.00	0.00
5	100	c	0.42	4.69	0.07	0.09
10	100	a	0.63	4.74	0.00	0.00
10	100	b	0.07	1.27	0.08	0.12
10	100	c	0.84	2.78	0.21	0.32
20	100	a	0.20	1.78	0.00	0.00
20	100	b	0.31	1.06	0.10	0.17
20	100	c	0.42	1.00	0.19	0.30

where z_{bound} stands for the associated bound, z_{lag}^{gr} or z_{md}^{gr}. To get $\text{gap}_{\text{bound}}$ we do not need to know the optimal value of the original problem, only values of the primal and dual bounds calculated by Algorithm 2 are used for estimating the quality of the feasible solutions.

As can be seen from Tables 3 and 4, for all problem instances and both indicators, gap_{ip} and $\text{gap}_{\text{bound}}$, the greedy solutions derived by the modified bound are better than the ones resulting from the standard Lagrangian bound. The values of the indicator $\text{gap}_{\text{bound}}$ for the modified bound are significantly (typically, in 10 times) smaller then corresponding values for the standard bound.

5. Conclusions

The procedure to tighten the Lagrangian bound was applied to the many-to-many assignment problem. The corresponding modified Lagrangian dual problem was solved approximately by the subgradient method. For the instances used in the computational tests the approach provided high-quality bounds, typically within less than 0.5% of relative suboptimality. A simple greedy technique was used to derive a feasible Lagrangian-based solution. Incorporating a feasible greedy solution into the subgradient scheme results in a significant decrease in the number of iterations without dropping the quality of the bounds. Using the modified bound provides high-quality approximate feasible solutions, typically within 0.5% of the suboptimality measured by the relative difference between dual and primal bounds.

The main focus of this paper was to study the effect of introducing a heuristic solution on the calculation and quality of the dual bounds. Much less attention was paid to designing the heuristic used to get feasible solutions. An interesting area for future research is to use the modified bounds in combination with other greedy approaches, e.g., using different choices for rounding up/down components, as well as with more sophisticated heuristic techniques (see, e.g., [18] and the references therein). An alternative to the subgradient technique used in this paper would be using more stable approaches such as center-based or bundle algorithms [19].

Acknowledgements

This work was partially supported by RFBR, Russia (09-01-00592), Mexican foundations CONACyT (61343), PAICYT (CE008-09), and PROMEP (103.5/09/3905), and Brazilian agencies CNPq and FAPESP.

References

[1] L.S. Lasdon, Optimization Theory for Large Systems, 2nd ed., Dover, 2002.
[2] R.K. Martin, Large Scale Linear and Integer Programming: A Unified Approach, Kluwer, Boston, 1999.
[3] J.E. Beasley, Lagrangian relaxation, in: C.R. Reeves (Ed.), Modern Heuristic Techniques for Combinatorial Problems, Blackwell Scientific Publications, 1993, pp. 243–303.
[4] M.L. Fisher, An application oriented guide to Lagrangian relaxation, Interfaces 15 (1985) 10–21.
[5] A. Frangioni, About Lagrangian methods in integer optimization, Annals of Operations Research 139 (2005) 163–169.
[6] C. Lemaréchal, Lagrangian relaxation, in: M. Jünger, D. Naddef (Eds.), Computational Combinatorial Optimization, Springer Verlag, 2001, pp. 115–160.
[7] J.F. Shapiro, A survey of Lagrangian techniques for discrete optimization, Annals of Discrete Mathematics 5 (1974) 113–138.
[8] I. Litvinchev, Refinement of Lagrangian bounds in optimization problems, Computational Mathematics and Mathematical Physics 47 (7) (2007) 1101–1108.
[9] I. Litvinchev, S. Rangel, J. Saucedo, A Lagrangian bound for many-to-many assignment problems, Journal of Combinatorial Optimization 19 (3) (2010) 241–257.
[10] D.W. Pentico, Assignment problems: a golden anniversary survey, European Journal of Operational Research 176 (2007) 774–793.
[11] R. Burkard, M. Dell'Amico, S. Martello, Assignment Problems, Springer, 2009.
[12] M. Boschetti, V. Maniezzo, Benders decomposition, Lagrangian relaxation and metaheuristic design, Journal of Heuristics 15 (3) (2009) 283–312.
[13] M. Held, R.M. Karp, The traveling salesman problem and minimum spanning trees, Operations Research 18 (1970) 1138–1162.
[14] L.A. Wolsey, Integer Programming, John Wiley & Sons, Inc., New York, NY, 1998.
[15] ILOG, ILOG CPLEX 10.0 (User's Manual), in: Mathematical Programming Optimizers, 2006.
[16] L.A.N. Lorena, M.J. Narciso, Relaxation heuristics for generalized assignment problem, European Journal of Operational Research 91 (1996) 600–610.
[17] M.J. Narciso, L. Lorena, Lagrangian/surrogate relaxation for generalized assignment problems, European Journal of Operational Research 114 (1) (1999) 165–177.
[18] V. Jeet, E. Kutanoglu, Lagrangian relaxation guided problem space search heuristics for generalized assignment problems, European Journal of Operational Research 182 (2007) 1039–1056.
[19] L. Bahiense, N. Maculan, C. Sagastizábal, The volume algorithm revisited: relation with bundle methods, Mathematical Programming 94 (2002) 41–69.