European Journal of Operational Research 218 (2012) 366–376

Contents lists available at SciVerse ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

# A theoretical and empirical investigation on the Lagrangian capacities of the 0-1 multidimensional knapsack problem

Yourim Yoon [a], Yong-Hyuk Kim [b,*], Byung-Ro Moon [a]

[a] School of Computer Science and Engineering, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 151-744, Republic of Korea
[b] Department of Computer Science and Engineering, Kwangwoon University, 20 Kwangwoon-ro, Nowon-gu, Seoul 139-701, Republic of Korea

ARTICLE INFO

*Article history:*
Received 20 February 2010
Accepted 2 November 2011
Available online 15 November 2011

*Keywords:*
Integer programming
0-1 Multidimensional knapsack problem
Lagrangian method
Lagrangian capacity

ABSTRACT

We present a novel Lagrangian method to find good feasible solutions in theoretical and empirical aspects. After investigating the concept of *Lagrangian capacity*, which is the value of the capacity constraint that Lagrangian relaxation can find an optimal solution, we formally reintroduce Lagrangian capacity suitable to the 0-1 multidimensional knapsack problem and present its new geometric equivalent condition including a new associated property. Based on the property, we propose a new Lagrangian heuristic that finds high-quality feasible solutions of the 0-1 multidimensional knapsack problem. We verify the advantage of the proposed heuristic by experiments. We make comparisons with existing Lagrangian approaches on benchmark data and show that the proposed method performs well on large-scale data.



## 1. Introduction

The 0-1 multidimensional knapsack problem (0-1 MKP) is an NP-hard problem, but not strongly NP-hard (Garey and Johnson, 1979). It can be considered as an extended version of the well-known 0-1 knapsack problem (0-1 KP). In the 0-1 knapsack problem, given a set of *objects*, each object that can go into the *knapsack* has a *size* and a *profit*. The knapsack has a certain capacity for size. The objective is to find an assignment that maximizes the total profit not exceeding the given capacity. In the case of the multidimensional knapsack problem, the number of capacity constraints is more than one. For example, the constraints can be a *weight* besides a size. Naturally, 0-1 MKP is a generalized version of 0-1 KP.

For the knapsack problem with *only one* constraint, there have been a number of researches about efficient approximation algorithm to find a near-optimal solution (see, for instance, Caprara et al. (2000), Kellerer and Pferschy (1998), Lawler (1977) and Sahni (1975)). In this paper, we are interested in the problem with more than one constraint, i.e., the multidimensional knapsack problem. In Gavish and Pirkul (1985), Martello and Toth (2003) and Shih (1979) among others, the exact algorithms for 0-1 MKP have been introduced. Heuristic approaches for 0-1 MKP have also been extensively studied in the past (Alves and Almeida, 2007; Balev et al., 2008; Bektas and Oguz, 2007; Boyer et al., 2009; Chekuri and Khanna, 2000; Chu and Beasley, 1998; Crama and Mazzola, 1994; Fleszar and Hindi, 2009; Fréville and Plateau, 1994; Frieze and Clarke, 1984; Hanafi and Fréville, 1998; Kong et al., 2008; Li, 2005; Li and Curry, 2005; Magazine and Oguz, 1984; Osorio et al., 2002; Pirkul, 1987; Raidl, 1998; Thiel and Voss, 1994; Vasquez and Hao, 2001; Vasquez and Vimont, 2005; Wilbaut and Hanafi, 2009; Wilbaut et al., 2009). A number of methods for the 0-1 bi-knapsack problem, which is a particular case of 0-1 MKP, have also been proposed (Bagchi et al., 1996; Campello and Maculan, 1988; Fréville and Plateau, 1993, 1996; Martello and Toth, 2003; Thiongane et al., 2006; Weingartner and Ness, 1967). The reader is referred to Fréville (2004), Fréville and Hanafi (2005) and Kellerer et al. (2004) for deep surveys of 0-1 MKP.

However, most researches directly deal with the discrete search space. In this paper, we transform the search space of the problem into a real space instead of directly managing the original discrete space. The 0-1 MKP is the optimization problem with multiple constraints. We transform the problem using the multiple Lagrange multipliers (Luenberger, 1969). However, we have a lot of limitations since the domain is not continuous but discrete. Lagrangian heuristics have been mainly used to get good upper bounds of the integer problems by the *duality* (Nemhauser and Wolsey, 1999; Wolsey, 1998). To get good upper bounds, a number of researchers have studied dual solvings using Lagrangian duality (Fréville et al., 1990; Karwan and Rardin, 1979; Thiongane et al., 2006; Yu, 1990), surrogate duality (Boyer et al., 2009; Fréville et al., 1990; Fréville and Plateau, 1993; Glover, 1965, 1968; Greenberg and Pierskalla, 1970; Karwan and Rardin, 1979), and composite duality (Karwan and Rardin, 1980), and there was a recent study using cutting plane method for 0-1 MKP (Kaparis and

* Corresponding author.
*E-mail addresses:* yryoon@soar.snu.ac.kr (Y. Yoon), yhdfly@kw.ac.kr (Y.-H. Kim), moon@snu.ac.kr (B.-R. Moon).

0377-2217/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2011.11.011

Letchford, 2008). However, in this paper, we focus on finding good lower bounds, i.e., good feasible solutions.

There have been a number of papers that studied Lagrangian method for discrete problems (Beasley, 1990a; Chang and Wah, 1995; Everett, 1963; Gavish, 1978; Geoffrion, 1974; Schuurmans et al., 2001; Wah and Shang, 1997; Wah and Wu, 1999). There were also a few methods that used Lagrange multipliers for 0-1 MKP. Typically, most of them found just good upper bounds to be used in a *branch-and-bound* algorithm by dualizing constraints, and hence could not find a feasible solution directly (Martello and Toth, 1990). To the best of the author's knowledge, there has been only one Lagrangian method to find lower bounds (feasible solutions) that is proposed by Magazine and Oguz (1984) (MO-CONS). There was also a method called LM-GA that improved the performance by combining MO-CONS with genetic algorithms (Raidl, 1999). LM-GA used the real-valued weight-codings to make a variant of the original problem and then applied MO-CONS.[1] LM-GA provided a new viewpoint to solve 0-1 MKP, but it just used MO-CONS for fitness evaluation and did not give any contribution in the aspect of Lagrangian theory.

We observed that the hardness of Lagrangian search is closely related to the capacity value of the given problem instance. Inspired by this observation, we reintroduce *Lagrangian capacity* suitable to the 0-1 multidimensional knapsack problem and provide its new geometric equivalent condition including a new associated property. As an application, we also propose a new Lagrangian heuristic in relation to Lagrangian capacity.

The remainder of this paper is organized as follows. The definition of Lagrangian capacity and its geometric equivalent conditions are presented in Section 2. We also provide an example for one-dimensional case to help understanding of Lagrangian capacity in the same section. In Section 3, we describe the algorithm generating Lagrangian capacity from Lagrange multiplier. Our Lagrangian search and the constructive heuristic method using this generator are described in Section 4. We present empirical analysis in Section 5 and make conclusions in Section 6.

## 2. Lagrangian capacity

Let $n$ and $m$ be the numbers of objects and capacity constraints, respectively. Each object $j$ has a profit $c_j$ and, for each constraint $i$, a capacity consumption value $a_{ij}$. Each constraint $i$ has a capacity $b_i$. Then, we formally define 0-1 MKP as follows:

$$\begin{aligned} \text{maximize} \quad & \boldsymbol{c}^T\boldsymbol{x} \\ \text{subject to} \quad & A\boldsymbol{x} \leqslant \boldsymbol{b}, \\ & \boldsymbol{x} \in \{0,1\}^n, \end{aligned}$$

where $\boldsymbol{c} = (c_j)$ and $\boldsymbol{x} = (x_j)$ are $n$-dimensional column vectors, $A = (a_{ij})$ is an $m \times n$ matrix, $\boldsymbol{b} = (b_i)$ is an $m$-dimensional column vector, and $T$ means the transpose of a matrix or a column vector. $A$, $\boldsymbol{b}$, and $\boldsymbol{c}$ are given and each element of them is a nonnegative integer. In brief, the objective of 0-1 MKP is to find a binary vector $\boldsymbol{x}$ which maximizes the weighted sum $\boldsymbol{c}^T\boldsymbol{x}$ satisfying $m$ linear constraints $A\boldsymbol{x} \leqslant \boldsymbol{b}$.

We reintroduce the concept of *Lagrangian capacity*.[2] Since the domain of 0-1 MKP is discrete, Lagrangian method cannot solve the problem in all the cases.[3] Lagrangian capacities have the key whether or not given 0-1 MKP can be solved by Lagrangian method. In the above formal definition, the vector $\boldsymbol{b}$ corresponds to the capacity of the problem. We call this vector $\boldsymbol{b}$ the *capacity* in the following. Roughly speaking, Lagrangian capacity is a special capacity that the Lagrangian method can be successfully applied.

### 2.1. Definition of Lagrangian capacity

To discuss the Lagrangian capacity and its equivalent conditions, we need the following notations.

$\boldsymbol{u} \in \mathbb{R}^m, \quad \boldsymbol{u} \geqslant \boldsymbol{0}.$
$\Omega = \{0,1\}^n$
$\omega(\boldsymbol{b}) = \max\,\{\boldsymbol{c}^T\boldsymbol{x}\colon \boldsymbol{x} \in \Omega,\, A\boldsymbol{x} \leqslant \boldsymbol{b}\}$, where $\boldsymbol{b}$ is the capacity vector
$X(\boldsymbol{u}) = \{\boldsymbol{x} \in \Omega\colon \boldsymbol{c}^T\boldsymbol{x} - \boldsymbol{u}^TA\boldsymbol{x} \geqslant \boldsymbol{c}^T\boldsymbol{y} - \boldsymbol{u}^TA\boldsymbol{y},\ \forall \boldsymbol{y} \in \Omega\}$

$\omega(\boldsymbol{b})$ is the maximum of the objective function when the capacity is $\boldsymbol{b}$ and $X(\boldsymbol{u})$ is the set-valued function which maps a given real vector $\boldsymbol{u}$ into the maximizers of $\boldsymbol{c}^T\boldsymbol{x} - \boldsymbol{u}^TA\boldsymbol{x}$.

If the domain is convex and the problem satisfies several necessary conditions, there always exists a real vector $\boldsymbol{u} \geqslant \boldsymbol{0}$ such that $\omega(\boldsymbol{b}) = \boldsymbol{c}^T\boldsymbol{x} - \boldsymbol{u}^T(A\boldsymbol{x} - \boldsymbol{b})$. We can find such $\boldsymbol{u}$ and solve the unconstrained problem of maximizing $\boldsymbol{c}^T\boldsymbol{x} - \boldsymbol{u}^T(A\boldsymbol{x} - \boldsymbol{b})$ instead of the given constrained problem. Such method is called *Lagrangian method* and real vector $\boldsymbol{u}$ used in this method is called *Lagrange multiplier*. However, since the problem we are to solve has a discrete domain, there are a lot of limitations in using Lagrangian method. We will give conditions to be satisfied so that Lagrangian method is applied to discrete domain. In particular, we focus on the condition of the capacity. The value of the capacity which lets the equality $\omega(\boldsymbol{b}) = \boldsymbol{c}^T\boldsymbol{x} - \boldsymbol{u}^T(A\boldsymbol{x} - \boldsymbol{b})$ hold for some $\boldsymbol{u}$ is defined as *Lagrangian capacity*.

**Definition 1.** A capacity $\bar{\boldsymbol{b}}$ is called a *Lagrangian capacity* if there is $\bar{\boldsymbol{u}} \geqslant \boldsymbol{0}$ such that $\bar{\boldsymbol{u}}^T(A\bar{\boldsymbol{x}} - \bar{\boldsymbol{b}}) = 0$ and $A\bar{\boldsymbol{x}} \leqslant \bar{\boldsymbol{b}}$ for some $\bar{\boldsymbol{x}} \in X(\bar{\boldsymbol{u}})$.

To indicate such $\bar{\boldsymbol{u}}$ and $\bar{\boldsymbol{x}}$, we will use the terms "$\bar{\boldsymbol{b}}$'s corresponding Lagrange multiplier" and "$\bar{\boldsymbol{b}}$'s corresponding solution," respectively.

### 2.2. Properties of Lagrangian capacity

As mentioned above, Lagrangian capacity for general mathematical programs is a well known concept and many properties of it are also known. In this subsection, we focus on 0-1 MKP and provide a unified view about Lagrangian capacity and its properties for 0-1 MKP. This subsection surveys previous literature and also presents a new theorem so that it gives readers a unified theory about Lagrangian capacity and its geometric properties for 0-1 MKP.

First, we remark the fact that the condition for the Lagrangian capacity is equivalent to the saddle point condition. It is known that the saddle point condition offers a convenient compact description of the essential elements of the Lagrangian results for convex programming (Luenberger, 1969). Its general version for arbitrary mathematical programs is well known (Brooks and Geoffrion, 1966).

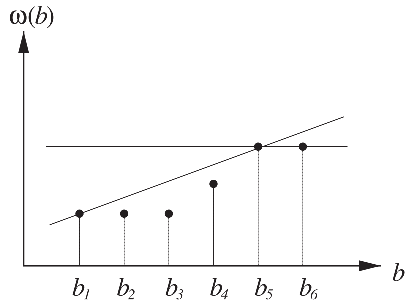
**Definition 2.** If there exists $\bar{\boldsymbol{u}} \geqslant \boldsymbol{0}$ such that $\boldsymbol{c}^T\boldsymbol{x} - \bar{\boldsymbol{u}}^T(A\boldsymbol{x} - \bar{\boldsymbol{b}}) \leqslant \boldsymbol{c}^T\bar{\boldsymbol{x}} - \bar{\boldsymbol{u}}^T(A\bar{\boldsymbol{x}} - \bar{\boldsymbol{b}}) \leqslant \boldsymbol{c}^T\bar{\boldsymbol{x}} - \boldsymbol{u}^T(A\bar{\boldsymbol{x}} - \bar{\boldsymbol{b}})$ for all $x \in \Omega$ and $u \geqslant \boldsymbol{0}$, we say that $\bar{\boldsymbol{b}}$ with $\bar{\boldsymbol{x}}$ satisfies the *saddle point condition.*

**Proposition 1.** *A capacity $\bar{\boldsymbol{b}}$ is a Lagrangian capacity if and only if $\bar{\boldsymbol{b}}$ with some $\bar{\boldsymbol{x}}$ satisfies the saddle point condition.*

[1] See Section 4.3.

[2] The *gap* is well known as the contrary concept of Lagrangian capacity of this paper (Bellmore et al., 1970; Everett, 1963). That is, a gap is a capacity at which it is impossible for Lagrangian method to find an optimal solution. However, the gap is for arbitrary mathematical programs. Lagrangian capacity of this paper is defined specially for 0-1 MKP.

[3] When the domain is convex, it is possible to find an optimal solution using Lagrangian method (Luenberger, 1969).




[Capacities $b_1$, $b_5$, and $b_6$ are Lagrangian since supporting hyperplanes at these points exist. But, $b_2$, $b_3$, and $b_4$ are not Lagrangian capacities since such hyperplanes do not exist.]

**Fig. 1.** The relationship between Lagrangian capacity and supporting hyperplane.

It is well known that if the capacity of the given problem instance is the Lagrangian capacity (or not *gap*) and the corresponding solution is found, we can confirm that the solution is optimal (Everett, 1963; Nemhauser and Wolsey, 1999). The next proposition shows this fact.

**Proposition 2.** *If $\bar{\boldsymbol{b}}$ is a Lagrangian capacity, there exist $\bar{\boldsymbol{u}} \geqslant \boldsymbol{0}$ and $\bar{\boldsymbol{x}} \in X(\bar{\boldsymbol{u}})$ satisfying $\bar{\boldsymbol{u}}^T(A\bar{\boldsymbol{x}} - \bar{\boldsymbol{b}}) = 0$ and $A\bar{\boldsymbol{x}} \leqslant \bar{\boldsymbol{b}}$ by the definition. Then, $\omega(\bar{\boldsymbol{b}}) = \boldsymbol{c}^T\bar{\boldsymbol{x}}$.*

The next corollary will be used practically in the next section.

**Corollary 1.** *If $\bar{\boldsymbol{x}} \in X(\bar{\boldsymbol{u}})$, then $A\bar{\boldsymbol{x}}$ is a Lagrangian capacity and $\omega(A\bar{\boldsymbol{x}}) = \boldsymbol{c}^T\bar{\boldsymbol{x}}$.*

By Propositions 1 and 2, the next corollary is implied naturally and this form is given in Brooks and Geoffrion (1966).

**Corollary 2.** *If $\bar{\boldsymbol{b}}$ with $\bar{\boldsymbol{x}}$ satisfies the saddle point condition, then $\omega(\bar{\boldsymbol{b}}) = \boldsymbol{c}^T\bar{\boldsymbol{x}}$.*

Next, we state a lemma about the sensitivity. Its generalized version for convex programming is well known from Luenberger (1969). The lemma will be used to prove a theorem about supporting hyperplane which will be mentioned in the following.

**Lemma 1.** *Suppose that $\bar{\boldsymbol{x}} \in X(\bar{\boldsymbol{u}})$ and $\tilde{\boldsymbol{x}} \in X(\tilde{\boldsymbol{u}})$. Then,*

$$\bar{\boldsymbol{u}}^T(A\bar{\boldsymbol{x}} - A\tilde{\boldsymbol{x}}) \leqslant \boldsymbol{c}^T\bar{\boldsymbol{x}} - \boldsymbol{c}^T\tilde{\boldsymbol{x}} \leqslant \tilde{\boldsymbol{u}}^T(A\bar{\boldsymbol{x}} - A\tilde{\boldsymbol{x}}).$$

*In particular, for any $\boldsymbol{x} \in \Omega$, $\boldsymbol{c}^T\boldsymbol{x} - \boldsymbol{c}^T\bar{\boldsymbol{x}} \leqslant \bar{\boldsymbol{u}}^T(A\boldsymbol{x} - A\bar{\boldsymbol{x}})$.*

Finally, we give a new geometric interpretation of Lagrangian capacity for 0-1 MKP. It is a new adaptation of the observation (Luenberger, 1969) for convex programming. Some geometric analyses for nonlinear programming can be found in Gould (1969).

**Definition 3.** Let $K$ be a nonempty set in $\mathbb{R}^n$, and let $\bar{\boldsymbol{x}} \in K$. When $p \neq \boldsymbol{0}$, a hyperplane $H = \{\boldsymbol{x} : \boldsymbol{p}^T(\boldsymbol{x} - \bar{\boldsymbol{x}}) = 0\}$ is called a *supporting hyperplane of K at $\bar{\boldsymbol{x}}$* if either $\boldsymbol{p}^T(\boldsymbol{x} - \bar{\boldsymbol{x}}) \geqslant 0$ for each $x \in K$, or else, $\boldsymbol{p}^T(\boldsymbol{x} - \bar{\boldsymbol{x}}) \leqslant 0$ for each $x \in K$.

We will try to understand Lagrangian capacity geometrically by analyzing it on the graph of $(\boldsymbol{b}, \omega(\boldsymbol{b}))$. If $\bar{\boldsymbol{b}}$ is Lagrangian capacity, there exists a supporting hyperplane at $(\bar{\boldsymbol{b}}, \omega(\bar{\boldsymbol{b}}))$.[4] The next theorem explains this fact. This will be helpful to understand Lagrangian capacity and the structure of the problem space. A similar theorem has been proved on real domain, however, to the best of the authors' knowledge, its expansion to integer domain of 0-1 MKP is a new result.

**Theorem 1.** *Let $K = \left\{ \begin{pmatrix} \boldsymbol{b} \\ \omega(\boldsymbol{b}) \end{pmatrix} : \boldsymbol{b}$ is a nonnegative integer vector in $\mathbb{R}^m \right\} \subset \mathbb{R}^{m+1}$. Then, $\bar{\boldsymbol{b}}$ is a Lagrangian capacity if and only if there exists a supporting hyperplane of K at $\begin{pmatrix} \bar{\boldsymbol{b}} \\ \omega(\bar{\boldsymbol{b}}) \end{pmatrix}$.*

A simple but typical example for one-dimensional capacity vector is shown in Fig. 1 to help understand the above theorem. In this figure, the capacities $b_2$, $b_3$, and $b_4$ are not Lagrangian because supporting hyperplanes at these points do not exist. However, we can draw supporting hyperplanes at $b_1$, $b_5$, and $b_6$, and hence they are Lagrangian capacities. These supporting hyperplanes may not be unique as we see in the case of capacity $b_5$. Two hyperplanes in the figure are both supporting hyperplanes at $b_5$. For more examples for one-dimensional case, see the next subsection.

### 2.3. Characterization of Lagrangian capacity for 0-1 KP

In this subsection, we clarify Lagrangian capacities of 0-1 KP, i.e., 0-1 MKP with one constraint. We are to identify all the Lagrangian capacities exactly in this case.

Consider the following 0-1 KP.

$$\begin{aligned} &\text{maximize} && \boldsymbol{c}^T\boldsymbol{x} \\ &\text{subject to} && \boldsymbol{a}^T\boldsymbol{x} \leqslant b, \\ &&& \boldsymbol{x} \in \{0,1\}^n, \end{aligned}$$

where $\boldsymbol{c} = (c_j)$ and $\boldsymbol{a} = (a_j)$ are $n$-dimensional column vectors and $b$ is a nonnegative integer. Each element of $\boldsymbol{c}$ and $\boldsymbol{a}$ is a positive integer.

Let $p_j = c_j/a_j$ be the profit density of object $j$. Let $I_\lambda := \{j: p_j > \lambda\}$ and $J_\lambda := \{j: p_j = \lambda\}$ for $\lambda \in \mathbb{R}$. Let $LC_1 := \left\{ \sum_{j \in I_\lambda} a_j + \sum_{j \in K} a_j : \lambda \geqslant 0,\ K \subseteq J_\lambda \right\}$ and $LC_2 := \left\{ k \in \mathbb{N}\ :\ k \geqslant \sum_{j=1}^n a_j \right\}$. $LC_1$ is a finite set. We can know all the Lagrangian capacities of 0-1 KP from the following fact.

**Fact 1.** A capacity $b$ is a Lagrangian capacity for 0-1 KP if and only if $b \in LC_1 \cup LC_2$.

Table 1 shows an example of problem instance. In this example, $LC_1 = \{0,3,4,13,20,25\}$ and $LC_2 = \{k \in \mathbb{N}\ :\ k \geqslant 25\}$. Also, we can check the relation between Lagrangian capacities and the existence of the supporting hyperplanes in Fig. 2.

**Table 1**
An example instance of 0-1 KP.

| $j$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $c_j$ | 2 | 12 | 9 | 3 | 5 |
| $a_j$ | 5 | 3 | 9 | 1 | 7 |
| $p_j$ | 0.4 | 4.0 | 1.0 | 3.0 | 0.7 |

[4] In fact, it is clear to write as $\begin{pmatrix} \bar{\boldsymbol{b}} \\ \omega(\bar{\boldsymbol{b}}) \end{pmatrix}$ since $\bar{\boldsymbol{b}}$ is a column vector.

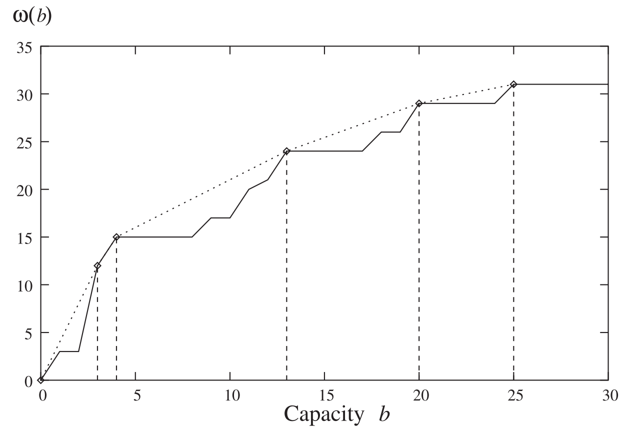



**Fig. 2.** Visualization of Lagrangian capacities for a 0-1 KP instance.

## 3. Generating Lagrangian capacities

If the capacity of the given problem is a Lagrangian capacity, we know that there exist a corresponding Lagrange multiplier and a corresponding solution, i.e., an optimal solution. However, although their existence is true, it is hard to find them. In fact, it is not easy even to know whether the given capacity is a Lagrangian capacity.

But, if $\bar{\boldsymbol{x}} \in X(\bar{\boldsymbol{u}})$ for some $\bar{\boldsymbol{u}} \in \mathbb{R}^m$, we can easily know that $A\bar{\boldsymbol{x}}$ is a Lagrangian capacity whose corresponding Lagrange multiplier is $\bar{\boldsymbol{u}}$ by Corollary 1. In this case, $\omega(A\bar{\boldsymbol{x}})$ is $\boldsymbol{c}^T\bar{\boldsymbol{x}}$. Based on this fact, we devise an alternative that does not guarantee to find an optimal solution but is practical.

One reason why this method is practical is that it is easy to find an assignment $\boldsymbol{x}$ such that $\boldsymbol{x} \in X(\boldsymbol{u})$ for given $\boldsymbol{u}$. To find such $\boldsymbol{x}$, we have to maximize the value of $\boldsymbol{c}^T\boldsymbol{x} - \boldsymbol{u}^T A\boldsymbol{x}$ by the definition of $X(\boldsymbol{u})$.

$$\boldsymbol{c}^T\boldsymbol{x} - \boldsymbol{u}^T A\boldsymbol{x} = \sum_{j=1}^{n} c_j x_j - \sum_{i=1}^{m} u_i \left( \sum_{j=1}^{n} a_{ij} x_j \right) = \sum_{j=1}^{n} x_j \left( c_j - \sum_{i=1}^{m} u_i a_{ij} \right).$$

To maximize $\boldsymbol{c}^T\boldsymbol{x} - \boldsymbol{u}^T A\boldsymbol{x}$ for fixed $\boldsymbol{u}$, we have to set $x_j$ to be 1 only if $c_j > \sum_{i=1}^m u_i a_{ij}$ for each $j = 1,2,\ldots,n$. In fact, in the case that $c_j = \sum_{i=1}^m u_i a_{ij}$, it does not matter if we set $x_j$ to be 0 or 1. But, in practice, $u_i$ is a *double* precision real number and this situation rarely happens. For that reason, we do not consider such case in this paper.

Since each $c_j$ does not have an effect on the others, getting the maximum is fairly easy. This algorithm computes just $\sum_{i=1}^m u_i a_{ij}$ for each $j = 1,2,\ldots,n$, its time complexity becomes $O(mn)$. We call this algorithm *Lagrangian method for the 0-1 multidimensional knapsack problem (LMMKP)* and Fig. 3 shows the pseudo-code of this algorithm. Variants of this procedure have already been used as a subroutine in solving Lagrangian dual problem, i.e., Lagrangian relaxation (Fréville et al., 1990; Magazine and Oguz, 1984). However, we briefly described our adaptation for completeness.

```
LMMKP(u, c, A)
{
    for j ← 1 to n
        if c_j > Σ_{i=1}^{m} u_i a_ij then x*_j ← 1;
        else x*_j ← 0;
    b* ← Ax*;
    μ* ← c^T x*;
    return (μ*, x*, b*);
}
```

$\boldsymbol{u} = (u_1, u_2, \ldots, u_m) \in \mathbb{R}^m$, $\boldsymbol{u} \geq \boldsymbol{0}$.

**Fig. 3.** Lagrangian method for 0-1 MKP.

LMMKP can be applied to various values of $\boldsymbol{u} \geqslant \boldsymbol{0}$, and we can obtain $\boldsymbol{x} \in X(\boldsymbol{u})$ for each $\boldsymbol{u}$. As results of applying LMMKP several times, we obtain the numbers of Lagrangian capacities and their corresponding optimal solutions. Among them, we can choose the *nearest* one to $\boldsymbol{b}$, the capacity of the given problem instance. The selected capacity $\boldsymbol{b}^*(=A\boldsymbol{x}^*)$ has to be less than or equal to $\boldsymbol{b}$ since the solution $\boldsymbol{x}^*$ must satisfy the constraint of the given problem instance, i.e., $A\boldsymbol{x}^* \leqslant \boldsymbol{b}$. If the capacity $A\boldsymbol{x}^*$ is equal to $\boldsymbol{b}$, $A\boldsymbol{x}^*$ is a Lagrangian capacity and hence $\boldsymbol{x}^*$ becomes an optimal solution by Corollary 1. But it is a rare case.

In this method, it is the key to find $\boldsymbol{u}$ which corresponds to the Lagrangian capacity as close to the capacity of the given problem instance as possible. New strategies for searching the Lagrange multiplier space are given in the next section.

## 4. Lagrangian heuristics

We suggest two heuristics to find good Lagrange multipliers for LMMKP effectively. We devise a novel iterative method using properties of Lagrangian capacity. Also, in terms of Lagrangian capacity, we present a constructive heuristic, which may be considered as a variant of the method proposed by Magazine and Oguz (1984).

Then, comparing with our Lagrangian heuristics, we review existing Lagrangian ones. We describe the *subgradient algorithm* which is typically used in the branch-and-bound method and other Lagrangian heuristics which showed good performance.

### *4.1. Feasibility-pursuing Lagrangian search*

Our goal is to find a real vector $\boldsymbol{u}$ whose corresponding Lagrangian capacity is quite close to the capacity of the given problem instance. We could observe that the value of $\boldsymbol{u}$ and the value of its corresponding Lagrangian capacity are closely related by the following theorem. We found the property in the following theorem, which can be practically used to solve 0-1 MKP.

**Theorem 2.** *Let $u = (u_1, u_2, \ldots, u_m)$ and $\boldsymbol{u}' = (u'_1, u'_2, \ldots, u'_m)$, where $u_i = u'_i$ for $i \neq k$ and $u_k \neq u'_k$. Suppose that $x \in X(u)$ and $x' \in X(u')$. Then, if $u_k < u'_k$, $b_k \geqslant b'_k$ and if $u_k > u'_k$, $b_k \leqslant b'_k$ where $b = Ax$ and $b' = Ax'$.*

Let $\boldsymbol{b}$ be the capacity of the given problem instance and let $\boldsymbol{b}^*$ be the Lagrangian capacity obtained by LMMKP with $\boldsymbol{u}$. By the above theorem, if $b^*_k > b_k$, choosing $\boldsymbol{u}' = (u_1, \ldots, u'_k, \ldots, u_m)$ such that $u'_k > u_k$ and applying LMMKP with $\boldsymbol{u}'$ makes the value of $b^*_k$ smaller. It makes the $k$th constraint satisfied or the exceeded amount for the $k$th capacity decreased. Of course, another constraint may become violated by this operation. Also, which $u_k$ to be changed is at issue in the case that several constraints are not satisfied. Hence, it is necessary to set efficient rules about which $u_k$ to be changed and how much to change it. If good rules are made, we can find out better Lagrange multipliers than randomly generated ones quickly.

We selected the method that chooses a random number $k(\leqslant m)$ and increases or decreases the value of $u_k$ by the above theorem iteratively. In each iteration, a Lagrangian capacity $\boldsymbol{b}^*$ is obtained by applying LMMKP with $\boldsymbol{u}$. If $\boldsymbol{b}^* \leqslant \boldsymbol{b}$, all constraints are satisfied and hence the best solution is updated. Since the possibility to find a better Lagrangian capacity exists, the algorithm does not stop here. Instead, it chooses a random number $k$ and decreases the value of $u_k$. If $\boldsymbol{b}^* \nleqslant \boldsymbol{b}$, we focus on satisfying constraints preferentially. For this, we choose a random number $k$ among the numbers such that their constraints are not satisfied and increase $u_k$ hoping the $k$th value of corresponding Lagrangian capacity to be decreased and then the $k$th constraint to be satisfied. We set the amount of $u_k$'s change in the $t$th iteration to be $1/(t + \gamma - 1)$. The change be-

```
// N and γ are parameters of FPLS.
FPLS(A, b, c)
{
        u ← 0;
        for t ← 1 to N
                δ ← 1/(t + γ − 1);
                (μ*, x*, b*) ← LMMKP(u, c, A);
                I ← {i : b*_i ≤ b_i} and J ← {i : b*_i > b_i};
                if I = {1, 2, ..., m}     // If all the constraints are satisfied, i.e., b* ≤ b
                        Update the best solution;
                        Choose a random element k among I;
                        u_k ← u_k − δ;
                else
                        Choose a random element k among J;
                        u_k ← u_k + δ;
        return the best solution;
}
```

**Fig. 4.** Feasibility-pursuing Lagrangian search.

comes smaller as the number of iterations grows. The value of $\gamma$ is tunable for the given problem instance.

In general, Lagrangian heuristics for discrete problems have focused on obtaining good upper bounds, but this algorithm is distinguished in that it primarily pursues finding feasible solutions. Thus we call this algorithm *Feasibility-Pursuing Lagrangian Search (FPLS)* in the following. Fig. 4 shows the pseudo-code of this iterative algorithm. It takes $O(nmN)$ time, where $N$ is the number of iterations.

### 4.2. Constructive heuristic

Magazine and Oguz (1984) proposed a constructive method using Lagrange multipliers. We devise a method similar to it focusing on the goal of finding a good Lagrangian capacity and its corresponding Lagrange multiplier.

First, $\boldsymbol{u}$ is set to be $\boldsymbol{0}$. Then, $(1,1,\ldots,1)^T$ is its corresponding solution since $X(\boldsymbol{u}) = \{\boldsymbol{x} \in \Omega\colon\ \boldsymbol{c}^T\boldsymbol{x} \geqslant \boldsymbol{c}^T\boldsymbol{y},\ \forall \boldsymbol{y} \in \Omega\}$. It means that all the objects are put in the knapsack and so almost all constraints are violated. If $\boldsymbol{u}$ is increased, some objects become taken out, i.e., for some $j$, $x_j$ becomes 0. We increase $\boldsymbol{u}$ adequately for only one object to be taken out. We change only one Lagrange multiplier at a time. We randomly choose one number $k$ and change $u_k$.

Reconsider

$$\boldsymbol{c}^T\boldsymbol{x} - \boldsymbol{u}^T A\boldsymbol{x} = \sum_{j=1}^{n} x_j\left(c_j - \sum_{i=1}^{m} u_i a_{ij}\right).$$

```
CH(A, b, c)
{
        u ← 0;
        J ← {1, 2, ..., n};
        do
                Choose a random element k among {1, 2, ..., m};
                for each j ∈ J
                        α_j ← (c_j − Σ_{i=1}^{m} u_i a_ij)/a_kj;
                u_k ← u_k + min_{j∈J} α_j;
                J ← J \ {argmin_{j∈J} α_j};
                (μ*, x*, b*) ← LMMKP(u, c, A);
        until b* ≤ b;
        return (μ*, x*, b*);
}
```

**Fig. 5.** Constructive heuristic (a variant of MO-CONS (Magazine and Oguz, 1984)).

Making $\left(c_j - \sum_{i=1}^{m} u_i a_{ij}\right)$ be negative by increasing $u_k$ lets $x_j = 0$ by LMMKP. For each $j$ such that $x_j = 1$, let $\alpha_j$ be the increment of $u_k$ to make $x_j$ be 0. Then $\left(c_j - \sum_{i=1}^{m} u_i a_{ij} - \alpha_j a_{kj}\right)$ has to be negative. That is, if we increase $u_k$ by $\alpha_j$ such that $\alpha_j > \left(c_j - \sum_{i=1}^{m} u_i a_{ij}\right)/a_{kj}$, the $j$th object is taken out. So, if we just change $u_k$ to $u_k + \min_j \alpha_j$, leave $u_i$ as it is for $i \neq k$, and apply LMMKP again, exactly one object is taken out. We take out objects one by one in this way and stop this procedure when every constraint is satisfied.

Fig. 5 shows this constructive algorithm. The number of operations to take out the object is at most $n$, and, for each $j$, computing $\alpha_j$ takes $O(m)$ time. Hence, the total time complexity becomes $O(n^2m)$.

### 4.3. Comparison with previous Lagrangian heuristics

In this subsection, we briefly examine existing Lagrangian heuristics for discrete optimization problems to compare with our new heuristics.

Coping with non-differentiability of the Lagrangian led to the last technical development: *subgradient algorithm*. Subgradient algorithm is a fundamentally simple procedure. Typically, the subgradient algorithm has been used as a technique for generating good upper bounds for branch-and-bound methods, where it is known as *Lagrangian relaxation* (Geoffrion, 1974; Martin, 1999). The reader is referred to Fisher (1981) and Shapiro (1979) for the deep survey of Lagrangian relaxation. At each iteration of the subgradient algorithm, one takes a step from the present Lagrange multiplier in the direction opposite to a subgradient, which is the direction of $(\boldsymbol{b}^* - \boldsymbol{b})$, where $\boldsymbol{b}^*$ is Lagrangian capacity obtained by LMMKP and $\boldsymbol{b}$ is the original capacity. It looks similar to the direction by FPLS, but the main difference lies in that, in each iteration, the subgradient algorithm changes all coordinate values by the subgradient direction but FPLS changes only one coordinate value. Consequently, this makes FPLS find lower bounds more easily than subgradient algorithm usually producing upper bounds.

The only previous attempt to find lower bounds using Lagrangian method is MO-CONS (Magazine and Oguz, 1984), however, MO-CONS without hybridization with other metaheuristics could not show satisfactory results. LM-GA by Raidl (1999) obtained better results by the hybridization of weight-coded genetic algorithm and MO-CONS. In LM-GA, a candidate solution is represented by a vector $(w_1, w_2, \ldots, w_n)$ of weights. Weight $w_j$ is associated with item $j$. Each profit $c_j$ is modified by applying several biasing techniques with these weights, i.e., we can obtain a modified problem instance $P'$ which has the same constraints as those of the original problem instance but has a different objective function. And then, solutions for this modified problem instance are obtained by applying a *decoding heuristic*. In particular, LM-GA used the constructive heuristic MO-CONS by Magazine and Oguz as a decoding heuristic. The feasible solutions for the modified problem instance are also feasible for the original problem instance since they satisfy the same constraints. So, weight-coding does not need an explicit repairing algorithm.

CH is different from MO-CONS in that CH has a random factor, so different solution is produced for each run. As a result, we can obtain better solution than MO-CONS by choosing the best solution from many independent runs of CH. FPLS, which is a more advanced method of CH, is different from LM-GA in that FPLS improves MO-CONS itself by using a relationship between Lagrange multiplier and Lagrangian capacity by Theorem 2, but LM-GA just uses MO-CONS itself. Lagrange multipliers in FPLS can move to more diverse directions than MO-CONS because of its random factor. MO-CONS can be considered as a special case of FPLS.

## 5. Experiments

We made experiments on well-known benchmark data publicly available from the OR-Library (Beasley, 1990b, 1996), which are the same as those used in Chu and Beasley (1998). They are composed of 270 instances with 5, 10, and 30 constraints. They have different numbers of objects and different tightness ratios. The tightness ratio means the value $\alpha$ such that $b_i = \alpha\sum_{j=1}^{n} a_{ij}$ for each $i = 1,2,\ldots,m$. The class of instances are briefly described below.

- $m.n$-$\alpha$: $m$ constraints, $n$ objects, and tightness ratio $\alpha$. Each class has 10 instances.

### 5.1. Distribution of Lagrangian capacities

Since FPLS tries to find Lagrangian capacity close to the original capacity, FPLS cannot perform well if Lagrangian capacities sparsely exist. From this fact, we can infer that the performance of FPLS depends on the distribution of Lagrangian capacities. To investigate the distribution of Lagrangian capacities, we performed a simple experiment. We generated 10,000,000 random Lagrange multipliers from a normal distribution $N(0,\sigma^2)$, computed Lagrangian capacities which correspond to them, and counted the number of distinct Lagrangian capacities. The standard deviation $\sigma$ is set to be $\frac{1}{3}\max_j c_j/a_{ij}$. If $u_i > \max_j c_j/a_{ij}$, then $X(\boldsymbol{u}) = \{\mathbf{0}\}$ regardless of the values of the other variables. This trivial feasible solution is meaningless, so we adjusted $\sigma$ not to generate these trivial Lagrange multipliers in most cases.

The larger the ratio of distinct Lagrangian capacities is, the more we can expect that FPLS finds good Lagrangian capacities and hence good feasible solutions. Table 2 gives the results from this experiment and Fig. 6 depicts them. The larger the number of constraints and objects is, the larger the ratio of distinct Lagrangian capacities is. Thus we expect that the proposed heuristic will perform well on large scale problems with respect to $m$ and $n$.

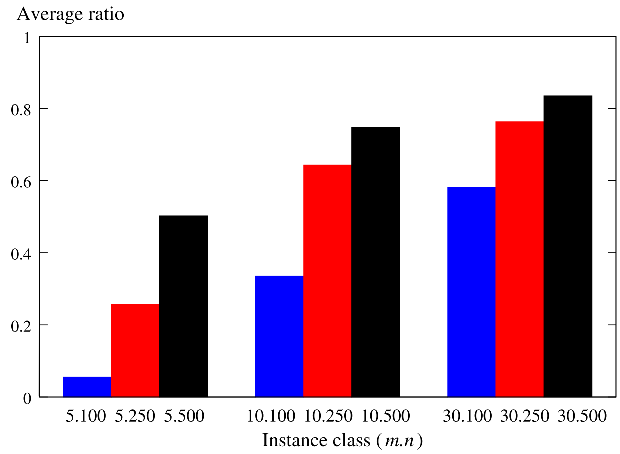



**Fig. 6.** Ratio of distinct Lagrangian capacities.

**Table 2**
Ratio of distinct Lagrangian capacities.

| Instance class ($m \cdot n$) | Average ratio[a] |
|---|---|
| 5.100 | 0.055 |
| 5.250 | 0.257 |
| 5.500 | 0.502 |
| 10.100 | 0.335 |
| 10.250 | 0.643 |
| 10.500 | 0.748 |
| 30.100 | 0.581 |
| 30.250 | 0.763 |
| 30.500 | 0.835 |

[a] Average from 30 instances.

### 5.2. Test on benchmark data

We tested the proposed algorithm on benchmark data by Chu and Beasley (1998). As a measure of performance, we used the percentage difference-ratio $100 \times (LP\_optimum - output)/LP\_optimum$ which was used in recent studies (Chu and Beasley, 1998; Raidl, 1998, 1999).[5] It has a value in the range [0,100]. The smaller the value is, the closer it is to the optimum.[6] We compared our iterative Lagrangian search method (FPLS) with the constructive Lagrangian method by Magazine and Oguz (1984) (MO-CONS), the constructive heuristic in Section 4.2 (CH), and Raidl's genetic algorithm using weight-codings based on Lagrangian relaxation (LM-GA) (Raidl, 1999). FPLS and CH were performed 1000 times and the best solutions among them were chosen for results. For FPLS, we set $N$ to be 30,000 and $\gamma$ to be 10.[7]

Table 3 shows the results with 30 constraints for which FPLS outperformed LM-GA. CH is a variant of MO-CONS showing better results than MO-CONS. The reason comes from the random factor of CH algorithm. CH can produce various solutions from different runs so we can take the best among them, whereas MO-CONS is a deterministic algorithm so that it always produces the same solution.

FPLS outperformed LM-GA for the classes with a large number of constraints ($m$ = 30). However, FPLS could not perform well for the instance classes with smaller numbers of constraints ($m$ = 5 or 10). These results were expected by the prior experiment for investigating the distribution of Lagrangian capacities. We can also observe that the relative performance of FPLS over LM-GA becomes more attractive as the number of objects increases. The last column in Table 3 corresponds to the numbers of improved and equalled results compared with the best results of Chu and Beasley (1998) (CB-GA). Surprisingly, FPLS could find better results than one of the state-of-the-art methods (Chu and Beasley, 1998) for some instances with 30 constraints (8 best over 90 instances). In sum, as an application of Lagrangian capacity, FPLS showed the best perfor-

[5] *LP_optimum* is the optimal value of the linear programming (LP) relaxation over $\mathbb{R}$.

[6] More precisely speaking, upper bound by LP relaxation.

[7] LM-GA performed MO-CONS (Magazine and Oguz, 1984) about 200,000 times for evaluation. From the fact that the time complexity of MO-CONS is $O(n^2m)$ and that of FPLS is $O(nmN)$, we estimate FPLS with 1000 runs and $N$ = 30,000 to take comparable time to LM-GA.

**Table 3**
Results from large benchmark data ($m$ = 30).

| Instance | MO-CONS | CH | | LM-GA | FPLS | | #Improved/#Equalled |
|---|---|---|---|---|---|---|---|
| | Ave. | Ave. | (CPU)[a] | Ave. | Ave. | (CPUa) | |
| 30.100-0.25 | 17.39 | 11.75 | (0.88) | 3.08 | 3.10 | (296) | 0/4 |
| 30.100-0.50 | 11.82 | 7.51 | (0.63) | 1.48 | 1.39 | (311) | 1/5 |
| 30.100-0.75 | 6.58 | 4.20 | (0.35) | 0.94 | 0.86 | (299) | 0/5 |
| Average | 11.93 | 7.82 | (0.62) | 1.83 | 1.78 | (302) | Total 1/14 |
| 30.250-0.25 | 13.54 | 11.36 | (5.35) | 1.62 | 1.27 | (735) | 2/2 |
| 30.250-0.50 | 8.64 | 6.71 | (3.36) | 0.71 | 0.57 | (763) | 1/0 |
| 30.250-0.75 | 4.49 | 3.56 | (1.94) | 0.45 | 0.32 | (738) | 1/1 |
| Average | 8.89 | 7.21 | (3.55) | 0.93 | 0.72 | (745) | Total 4/3 |
| 30.500-0.25 | 9.84 | 9.39 | (19.24) | 1.00 | 0.69 | (1464) | 1/0 |
| 30.500-0.50 | 7.10 | 6.17 | (13.39) | 0.45 | 0.30 | (1527) | 1/0 |
| 30.500-0.75 | 3.72 | 3.34 | (6.95) | 0.33 | 0.19 | (1463) | 1/0 |
| Average | 6.89 | 6.30 | (13.19) | 0.59 | 0.39 | (1485) | Total 3/0 |
| Total average | 9.24 | 7.11 | (5.79) | 1.12 | 0.96 | (844) | Total 8/17 |

• LM-GA (Raidl, 1999) is the genetic algorithm combined with MO-CONS (Magazine and Oguz, 1984).
• CH means the constructive heuristic of Section 4.2.
• FPLS means our iterative Lagrangian heuristic pursuing feasibility of Section 4.1.
• The results are given as the average percentage difference-ratio over 10 instances, where percentage difference-ratio means 100 × ($LP\_optimum - output$)/$LP\_optimum$.
[a] Average of overall CPU seconds for 1000 runs on Intel Core2 Duo CPU 2.66 GHz.

mance for large scale instances among existing Lagrangian heuristics. Hence, FPLS will be a good choice in real-world problems.

For more analysis about the performance of FPLS, we conducted additional experiments showing results according to the number of iterations ($N$) and the number of runs ($R$). Table 4 shows the results with various values of $N$ and $R$. Each value in the table means the average percentage difference-ratio over all instances with 30 constraints. The performance of FPLS consistently improves as $N$ or $R$ increases, but even FPLS with $N$ = 5000 and $R$ = 500 showed practically good quality (1.02) to outperform LM-GA (1.12). It only takes a 12th of the CPU times reported in Table 3.

Recently, there have been researches showing better performance than (Chu and Beasley, 1998) on the benchmark data of 0-1 MKP (Vasquez and Vimont, 2005; Vimont et al., 2008; Hanafi and Wilbaut, 2008, 2011; Wilbaut and Hanafi, 2009; Boussier et al., 2010). Although FPLS did not dominate the state-of-the-art methods in this study, we believe that there is room for further improvement when FPLS is combined with metaheuristics, e.g., genetic algorithms (Goldberg, 1989) and tabu search (Glover, 1989), as in Chu and Beasley (1998) and Vasquez and Vimont (2005) since FPLS is a stand-alone method without hybridization with any other heuristic and its computational cost can be adjusted according to given metaheuristic through the number of iterations ($N$) or the number of runs ($R$).

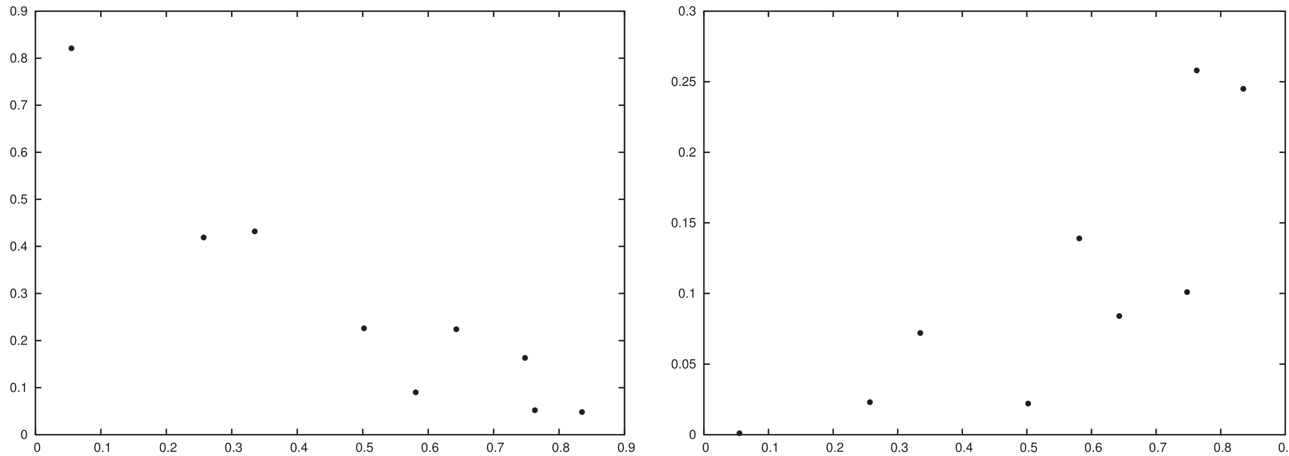
We expect that FPLS will show competitive performance with the state-of-the-art methods mentioned above, in the case that the problem size is larger than those of benchmark data. To support our expectation, we also investigated the performance of FPLS compared with the genetic algorithm in Chu and Beasley (1998) (CB-GA).

The performance of FPLS is very relevant to the distribution of Lagrangian capacities. Let $r_{m,n}$ be the ratio of distinct Lagrangian capacities with $m$ constraints and $n$ objects. As shown in Table 2 and Fig. 6, the larger $m$ and $n$ are, the higher $r_{m,n}$ is. Intuitively, FPLS performs well when the ratio is high. To verify whether or not the intuition is valid in practical, we compared the performances of FPLS and CB-GA in relation with the value of $r_{m,n}$. We calculated the percentage difference-ratio $100 \times (C - X)/C$, where $C$ and $X$ are the output values of CB-GA and target method, respectively. And then we plotted the values with the corresponding $r_{m,n}$s on $x - y$ plane, where the $x$-coordinates indicate the value of $r_{m,n}$ and the $y$-coordinates indicate the percentage difference-ratio to CB-GA. Small percentage difference-ratio means that the output value of target method is close to that of CB-GA, i.e., the output of target method has good quality.

In Fig. 7(a), we can see that the difference of output values between FPLS and CB-GA decreases as $r_{m,n}$ increases, i.e., as the problem size increases. This is not a trivial phenomenon since it is not the case for LM-GA as shown in Fig. 7(b). The difference tends to rather increase for LM-GA. This fact also supports that FPLS did not outperform LM-GA for $m$ = 5 and $m$ = 10 although the values of $r_{5,500}$, $r_{10,250}$, and $r_{10,500}$ are near to those for $m$ = 30. LM-GA tends to perform better in small-sized instances than in large-sized ones in contrast with FPLS. Based on these observations, it is quite reasonable to conclude that FPLS will perform better as the problem size increases.

**Table 4**
Performance of FPLS according to the values of $N$ and $R$.

| $R\backslash N$ | 5000 | 10,000 | 15,000 | 20,000 | 25,000 | 30,000 | 35,000 | 40,00 | 45,000 | 50,000 |
|---|---|---|---|---|---|---|---|---|---|---|
| 500 | 1.02 | 1.00 | 0.99 | 0.98 | 0.98 | 0.98 | 0.98 | 0.98 | 0.98 | 0.98 |
| 1000 | 1.01 | 0.98 | 0.98 | 0.97 | 0.97 | 0.97 | 0.97 | 0.97 | 0.97 | 0.97 |
| 1500 | 1.00 | 0.98 | 0.97 | 0.97 | 0.97 | 0.96 | 0.96 | 0.96 | 0.96 | 0.96 |
| 2000 | 0.99 | 0.97 | 0.97 | 0.96 | 0.96 | 0.96 | 0.96 | 0.96 | 0.96 | 0.96 |
| 2500 | 0.98 | 0.96 | 0.96 | 0.96 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 |
| 3000 | 0.98 | 0.96 | 0.96 | 0.96 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 |
| 3500 | 0.98 | 0.96 | 0.96 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 |
| 4000 | 0.97 | 0.96 | 0.96 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 |
| 4500 | 0.97 | 0.96 | 0.95 | 0.95 | 0.95 | 0.94 | 0.94 | 0.94 | 0.94 | 0.94 |
| 5000 | 0.97 | 0.96 | 0.95 | 0.95 | 0.95 | 0.94 | 0.94 | 0.94 | 0.94 | 0.94 |




(a) FPLS (b) LM-GA

- These figures present the performances of two methods in relation with the distribution of Lagrangian capacities.
- $x$-coordinates indicate $r_{m,n}$ in Table 2 and $y$-coordinates indicate the percentage difference-ratio $100 \times (C - X)/C$, where $C$ and $X$ are the output values of CB-GA and target method, respectively.

**Fig. 7.** Relative performance of FPLS and LM-GA compared to CB-GA.

### 5.3. Test on instances with Lagrangian capacities

Recently, other test data set different from those in OR-library have been developed for 0-1 MKP (Hill et al., 2010). As another application of Lagrangian capacity, we also generated a new class of problem instances whose optimal solutions are known. By testing on these instances, we can measure the suboptimality of heuristics. We also tested FPLS and CH on newly-generated problem instances whose capacities are Lagrangian.

The purpose of FPLS is to find the Lagrangian capacity closest to the original one and then solve the new instance with the obtained Lagrangian capacity instead of the original one. So if the capacity of a given instance is Lagrangian, the performance of FPLS can be maximized. The probability of the capacities of practical instances, such as benchmark data, being Lagrangian may follow the ratios given in Table 2. Since the ratio is high for large-sized instances, FPLS shows good performance on the instances, but not the case for small-sized ones. However, we can guess that FPLS would perform well regardless of the instance size if the capacities of given instances are Lagrangian. To verify this, we used these new data with Lagrangian capacities.

We generated 30 instances for each class with different number of objects and constraints (totally 270 instances). All the capacities of these instances are adjusted to be Lagrangian capacities as follows. We generated random real vector $\boldsymbol{u}$s in a proper range and then applied LMMKP using the original values of the benchmark data except only the capacity vector. By LMMKP, we got Lagrangian capacities for each problem. This operation was repeated until the tightness ratio of each constraint was laid between 0.2 and 0.8. The new instances were generated by substituting these Lagrangian

**Table 5**
Results from new instances with Lagrangian capacities.

| Instance | CT-GA | | S-GA | | CH | | FPLS | |
|---|---|---|---|---|---|---|---|---|
| | Ave. | (CPU)[a] | Ave. | (CPU)a | Ave. | (CPU)[a] | Ave. | (CPU)[a] |
| N.5.100 | 0.50 | (7 s) | 0.03 | (1.00 s) | 23.66 | (0.20 ms) | 0.00 | (0.06 s) |
| N.5.250 | 0.54 | (38 s) | 0.22 | (2.56 s) | 19.05 | (1.35 ms) | 0.00 | (0.15 s) |
| N.5.500 | 1.25 | (144 s) | 0.09 | (5.56 s) | 9.48 | (3.90 ms) | 0.00 | (0.29 s) |
| Average | 0.76 | (63 s) | 0.11 | (3.04 s) | 17.39 | (1.82 ms) | 0.00 | (0.17 s) |
| N.10.100 | 1.16 | (8 s) | 0.00 | (1.48 s) | 27.11 | (0.35 ms) | 0.00 | (0.12 s) |
| N.10.250 | 0.92 | (40 s) | 0.17 | (3.81 s) | 20.77 | (2.30 ms) | 0.00 | (0.27 s) |
| N.10.500 | 2.41 | (150 s) | 0.25 | (9.29 s) | 22.20 | (6.58 ms) | 0.00 | (0.57 s) |
| Average | 1.50 | (66 s) | 0.14 | (4.86 s) | 23.36 | (3.08 ms) | 0.00 | (0.32 s) |
| N.30.100 | 0.48 | (13 s) | 0.00 | (3.87 s) | 15.98 | (0.55 ms) | 0.00 | (0.27 s) |
| N.30.250 | 1.22 | (48 s) | 0.07 | (9.40 s) | 12.18 | (4.64 ms) | 0.00 | (0.64 s) |
| N.30.500 | 3.26 | (166 s) | 0.56 | (22.27 s) | 11.42 | (18.02 ms) | 0.04 | (1.29 s) |
| Average | 1.65 | (76 s) | 0.21 | (11.85 s) | 13.19 | (7.74 ms) | 0.01 | (0.73 s) |
| Total average | 1.30 | (68 s) | 0.15 | (6.58 s) | 17.98 | (4.21 ms) | 0.00 | (0.41 s) |

- CT-GA is a hybrid genetic algorithm in Cotta and Troya (1997).
- S-GA is a genetic algorithm implemented to be similar to (Chu and Beasley, 1998) by Shah.
- CH means the constructive heuristic of Section 4.2.
- FPLS means our iterative Lagrangian heuristic pursuing feasibility of Section 4.1.
- The results are given as the average percentage difference-ratio over 30 instances, where percentage difference-ratio means $100 \times$ (*optimum* − *output*)/*optimum*.

[a] Average CPU time for a single run on Intel Core2 Duo CPU 2.66 GHz.

capacities for the original capacities. Since we know the corresponding solutions of these Lagrangian capacities by LMMKP, we obtain optimal solutions of these new instances.

Table 5 shows the results on the new test data. To check the difficulty of the new instances, we obtained the executable file of a hybrid genetic algorithm in Cotta and Troya (1997) (CT-GA) and the free source code of a genetic algorithm implemented to be similar to CB-GA (Chu and Beasley, 1998) (S-GA)[8]. As a measure of performance, we used the percentage difference-ratio 100 × (*optimum* −*output*)/*optimum*. The only difference from the one used in previous experiments is that we used *optimum* instead of *LP_optimum* since we know the optimal values in these cases. CT-GA, S-GA, CH, and FPLS were performed once for each instance, and the average percentage difference-ratios over 30 instances in the same class are given in the table.

In our experiments, FPLS found the optima for almost all the instances except only one instance in N.30.500. On the other hand, the results of CT-GA and S-GA were 1.30 and 0.15, respectively. From this fact, we can know that newly generated data are not so easy to solve by other methods. CH also showed poor performance when compared with its performance on benchmark data, although it uses Lagrangian properties as FPLS does. FPLS tends to perform well on large-scale instances as stated in Sections 5.1 and 5.2. However, the results suggest that FPLS easily finds an optimal solution when the capacity of the given problem instance is Lagrangian, even for small-sized instances.

## 6. Conclusions

In this paper, we reintroduced the concept of Lagrangian capacity, which was introduced in the 1960s, and provided its new geometric equivalent condition. Lagrangian relaxation has been widely used for constrained optimization problems. In 0-1 MKP, it has usually been used to obtain good upper bounds. In this paper, we analyzed Lagrangian method in discrete problem through Lagrangian capacity and its properties. Also, we tried to find good lower bounds by using a novel simple heuristic based on these properties. The definition of Lagrangian capacity and the descriptions of its properties are somewhat theoretical and may not give intuition. So we provided a visualized example for one-dimensional case to help understanding.

In theoretical views, we found the new geometric properties of Lagrangian capacities by presenting Theorems 1 and 2, analyzed new intuitive properties for the one-dimensional case, and linked these theoretical results to practical application beyond just stating theory.

In experiments, we showed the best performance using only the iterative method for instances with Lagrangian capacities. We expect that there is room for further improvement through combining it with other approaches. For example, the solutions obtained by our Lagrangian heuristic can be used as initial solutions for other metaheuristics, e.g., evolutionary algorithms (Chu and Beasley, 1998; Raidl, 1998, 1999; Yoon et al., 2005) or tabu search (Glover and Kochenberger, 1996; Hanafi and Fréville, 1998; Løkketangen and Glover, 1996; Vasquez and Vimont, 2005). Research of such hybridization is promising and left for future study.

The proposed Lagrangian heuristic showed a good tendency to perform better as the number of the constraints increases. Considering the general fact that the problem becomes harder as the problem size becomes larger, this is an unusual behavior. We conjecture that this behavior is highly related to the distribution of Lagrangian capacities according to the number of the constraints. To find out more properties and applications of Lagrangian capacity are left for future work and we believe that such studies will help us understand the structure of the problem space better.

## Acknowledgments

The authors thank the anonymous referees, Prof. Yves Crama, and the late Prof. Peter L. Hammer for their encouragement and valuable suggestions in improving this paper. The present Research has been conducted by the Research Grant of Kwangwoon University in 2011. The ICT at Seoul National University provides research facilities for this study. This work was supported by the Engineering Research Center of Excellence Program (Grant 2011-0000966), Basic Science Research Program (Grant 2011-0004215), and Mid-career Researcher Program (Grant 2011-0018006) of Korea Ministry of Education, Science and Technology (MEST)/ National Research Foundation of Korea (NRF).

## Appendix A. Proofs

### A.1. Proof of Proposition 1

If $\bar{\boldsymbol{b}}$ is a Lagrangian capacity, there exist $\bar{\boldsymbol{u}} \geqslant \boldsymbol{0}$ and $\bar{\boldsymbol{x}} \in X(\bar{\boldsymbol{u}})$ satisfying $\bar{\boldsymbol{u}}^T(A\bar{\boldsymbol{x}} - \bar{\boldsymbol{b}}) = 0$ and $A\bar{\boldsymbol{x}} \leqslant \bar{\boldsymbol{b}}$.[9] Since $\bar{\boldsymbol{x}} \in X(\bar{\boldsymbol{u}}), \boldsymbol{c}^T\bar{\boldsymbol{x}} - \bar{\boldsymbol{u}}^T A\bar{\boldsymbol{x}} \geqslant \boldsymbol{c}^T\boldsymbol{x} - \bar{\boldsymbol{u}}^T A\boldsymbol{x}$ for all $\boldsymbol{x} \in \Omega$. Hence, $\boldsymbol{c}^T\bar{\boldsymbol{x}} - \bar{\boldsymbol{u}}^T(A\bar{\boldsymbol{x}} - \bar{\boldsymbol{b}}) = \boldsymbol{c}^T\bar{\boldsymbol{x}} - \bar{\boldsymbol{u}}^T A\bar{\boldsymbol{x}} + \bar{\boldsymbol{u}}^T\bar{\boldsymbol{b}} \geqslant \boldsymbol{c}^T\boldsymbol{x} - \bar{\boldsymbol{u}}A\boldsymbol{x} + \bar{\boldsymbol{u}}^T\bar{\boldsymbol{b}} = \boldsymbol{c}^T\boldsymbol{x} - \bar{\boldsymbol{u}}^T\ (A\boldsymbol{x} - \bar{\boldsymbol{b}})$ for all $\boldsymbol{x} \in \Omega$. And, since $\boldsymbol{u}^T(A\bar{\boldsymbol{x}} - \bar{\boldsymbol{b}}) \leqslant 0$ for all $\boldsymbol{u} \geqslant \boldsymbol{0}$, $\boldsymbol{c}^T\bar{\boldsymbol{x}} - \bar{\boldsymbol{u}}^T(A\bar{\boldsymbol{x}} - \bar{\boldsymbol{b}}) = \boldsymbol{c}^T\bar{\boldsymbol{x}} \leqslant \boldsymbol{c}^T\bar{\boldsymbol{x}} - \boldsymbol{u}^T(A\bar{\boldsymbol{x}} - \bar{\boldsymbol{b}})$.

To show the other direction, suppose that $\bar{\boldsymbol{b}}$ with $\bar{\boldsymbol{x}}$ satisfies the saddle point condition. Then, there exists $\bar{\boldsymbol{u}} \geqslant \boldsymbol{0}$ such that $\boldsymbol{c}^T\bar{\boldsymbol{x}} - \bar{\boldsymbol{u}}^T A\bar{\boldsymbol{x}} \geqslant \boldsymbol{c}^T\boldsymbol{x} - \bar{\boldsymbol{u}}^T A\boldsymbol{x}$ for all $\boldsymbol{x} \in \Omega$. Hence, $\bar{\boldsymbol{x}} \in X(\bar{\boldsymbol{u}})$. Also,

$$
\begin{aligned}
&\boldsymbol{u}^T(A\bar{\boldsymbol{x}} - \bar{\boldsymbol{b}}) \leqslant \bar{\boldsymbol{u}}^T(A\bar{\boldsymbol{x}} - \bar{\boldsymbol{b}}) \quad \text{for all } \boldsymbol{u} \geqslant \boldsymbol{0},\\
&(\boldsymbol{v} + \bar{\boldsymbol{u}})^T(A\bar{\boldsymbol{x}} - \bar{\boldsymbol{b}}) \leqslant \bar{\boldsymbol{u}}^T(A\bar{\boldsymbol{x}} - \bar{\boldsymbol{b}}) \quad \text{for all } \boldsymbol{v} \geqslant \boldsymbol{0},\\
&\boldsymbol{v}^T(A\bar{\boldsymbol{x}} - \bar{\boldsymbol{b}}) \leqslant \boldsymbol{0} \quad \text{for all } \boldsymbol{v} \geqslant \boldsymbol{0},\\
&A\bar{\boldsymbol{x}} - \bar{\boldsymbol{b}} \leqslant \boldsymbol{0}.
\end{aligned}
$$

From the above statement, we showed that $A\bar{\boldsymbol{x}} \leqslant \bar{\boldsymbol{b}}$. $\bar{\boldsymbol{u}}^T(A\bar{\boldsymbol{x}} - \bar{\boldsymbol{b}}) \leqslant 0$ since $\bar{\boldsymbol{u}} \geqslant \boldsymbol{0}$ and $A\bar{\boldsymbol{x}} - \bar{\boldsymbol{b}} \leqslant \boldsymbol{0}$. On the other hand, $\bar{\boldsymbol{u}}^T(A\bar{\boldsymbol{x}} - \bar{\boldsymbol{b}}) \geqslant 0$ since the inequality $\bar{\boldsymbol{u}}^T(A\bar{\boldsymbol{x}} - \bar{\boldsymbol{b}}) \geqslant \boldsymbol{u}^T(A\bar{\boldsymbol{x}} - \bar{\boldsymbol{b}})$ holds for $\boldsymbol{u} = \boldsymbol{0}$, Hence, $\bar{\boldsymbol{u}}^T(A\bar{\boldsymbol{x}} - \bar{\boldsymbol{b}}) = 0$. □

### A.2. Proof of Proposition 2

Let $\boldsymbol{x}$ be an arbitrary element in $\Omega$ satisfying $A\boldsymbol{x} \leqslant \bar{\boldsymbol{b}}$. Since $\bar{\boldsymbol{x}} \in X(\bar{\boldsymbol{u}})$, $\boldsymbol{c}^T\bar{\boldsymbol{x}} - \bar{\boldsymbol{u}}^T A\bar{\boldsymbol{x}} \geqslant \boldsymbol{c}^T\boldsymbol{x} - \bar{\boldsymbol{u}}^T A\boldsymbol{x}$. Then,

$$
\begin{aligned}
\boldsymbol{c}^T\bar{\boldsymbol{x}} &\geqslant \boldsymbol{c}^T\boldsymbol{x} + \bar{\boldsymbol{u}}^T(A\bar{\boldsymbol{x}} - A\boldsymbol{x})\\
&= \boldsymbol{c}^T\boldsymbol{x} + \bar{\boldsymbol{u}}^T(A\bar{\boldsymbol{x}} - \bar{\boldsymbol{b}} + \bar{\boldsymbol{b}} - A\boldsymbol{x})\\
&= \boldsymbol{c}^T\boldsymbol{x} + \bar{\boldsymbol{u}}^T(\bar{\boldsymbol{b}} - A\boldsymbol{x})\\
&\geqslant \boldsymbol{c}^T\boldsymbol{x}. \qquad \square
\end{aligned}
$$

### A.3. Proof of Lemma 1

Since $\bar{\boldsymbol{x}} \in X(\bar{\boldsymbol{u}})$, $\boldsymbol{c}^T\bar{\boldsymbol{x}} - \bar{\boldsymbol{u}}^T A\bar{\boldsymbol{x}} \geqslant \boldsymbol{c}^T\tilde{\boldsymbol{x}} - \bar{\boldsymbol{u}}^T A\tilde{\boldsymbol{x}}$. Hence, $\boldsymbol{c}^T\bar{\boldsymbol{x}} - \boldsymbol{c}^T\tilde{\boldsymbol{x}} \geqslant \bar{\boldsymbol{u}}^T(A\bar{\boldsymbol{x}} - A\tilde{\boldsymbol{x}})$. The other inequality is proved similarly. □

### A.4. Proof of Theorem 1

Suppose that $\bar{\boldsymbol{b}}$ is a Lagrangian capacity. Then, there exist $\bar{\boldsymbol{u}} \geqslant \boldsymbol{0}$ and $\bar{\boldsymbol{x}} \in X(\bar{\boldsymbol{u}})$ satisfying $\bar{\boldsymbol{u}}^T(A\bar{\boldsymbol{x}} - \bar{\boldsymbol{b}}) = 0$ and $A\bar{\boldsymbol{x}} \leqslant \bar{\boldsymbol{b}}$ by the definition. By Proposition 2, $\omega(\bar{\boldsymbol{b}}) = \boldsymbol{c}^T\bar{\boldsymbol{x}}$.

[8] http://shah.freeshell.org/gamultiknapsack/.

[9] In linear algebra, a matrix $A^T$ means the *transpose* of a matrix $A$.

Let $\boldsymbol{p} := \begin{pmatrix} \bar{\boldsymbol{u}} \\ -1 \end{pmatrix}$. Then, $\boldsymbol{p}^T\left(\begin{pmatrix} \boldsymbol{b} \\ \omega(\boldsymbol{b}) \end{pmatrix} - \begin{pmatrix} \bar{\boldsymbol{b}} \\ \omega(\bar{\boldsymbol{b}}) \end{pmatrix}\right) = \bar{\boldsymbol{u}}^T(\boldsymbol{b}-\bar{\boldsymbol{b}}) - (\omega(\boldsymbol{b}) - \omega(\bar{\boldsymbol{b}}))$. We will show that $\bar{\boldsymbol{u}}^T(\boldsymbol{b}-\bar{\boldsymbol{b}}) \geqslant \omega(\boldsymbol{b}) - \omega(\bar{\boldsymbol{b}})$ for every integer vector $\boldsymbol{b} \geqslant \boldsymbol{0}$. Let $\boldsymbol{x_b} \in \operatorname{argmax}_{A\boldsymbol{x}\leqslant \boldsymbol{b}} \boldsymbol{c}^T\ \boldsymbol{x}$. Then, $\omega(\boldsymbol{b}) = \boldsymbol{c}^T\ \boldsymbol{x_b}$ and $A\boldsymbol{x_b} \leqslant \boldsymbol{b}$

$$\begin{aligned}
\bar{\boldsymbol{u}}^T(\boldsymbol{b}-\bar{\boldsymbol{b}}) &= \bar{\boldsymbol{u}}^T(\boldsymbol{b} - A\boldsymbol{x_b} + A\boldsymbol{x_b} - A\bar{\boldsymbol{x}} + A\bar{\boldsymbol{x}} - \bar{\boldsymbol{b}}) \\
&= \bar{\boldsymbol{u}}^T(\boldsymbol{b} - A\boldsymbol{x_b}) + \bar{\boldsymbol{u}}^T(A\boldsymbol{x_b} - A\bar{\boldsymbol{x}}) + \bar{\boldsymbol{u}}^T(A\bar{\boldsymbol{x}} - \bar{\boldsymbol{b}}) \\
&= \bar{\boldsymbol{u}}^T(\boldsymbol{b} - A\boldsymbol{x_b}) + \bar{\boldsymbol{u}}^T(A\boldsymbol{x_b} - A\bar{\boldsymbol{x}}) \\
&\geqslant \bar{\boldsymbol{u}}^T(A\boldsymbol{x_b} - A\bar{\boldsymbol{x}}) \\
&\geqslant \boldsymbol{c}^T\boldsymbol{x_b} - \boldsymbol{c}^T\bar{\boldsymbol{x}}\ (\because \text{Lemma 1}) \\
&= \omega(\boldsymbol{b}) - \omega(\bar{\boldsymbol{b}}).
\end{aligned}$$

Now, suppose that there exists a supporting hyperplane at $\begin{pmatrix} \bar{\boldsymbol{b}} \\ \omega(\bar{\boldsymbol{b}}) \end{pmatrix}$. Without loss of generality, we may assume that there exists $\boldsymbol{p} = (p_1, p_2, \ldots, p_{m+1})^T$ such that $\boldsymbol{p}^T\left(\begin{pmatrix} \boldsymbol{b} \\ \omega(\boldsymbol{b}) \end{pmatrix} - \begin{pmatrix} \bar{\boldsymbol{b}} \\ \omega(\bar{\boldsymbol{b}}) \end{pmatrix}\right) \leqslant 0$. If $p_{m+1} = 0$, $(p_1, p_2, \ldots, p_m)(\boldsymbol{b} - \bar{\boldsymbol{b}}) \leqslant 0$ for all $\boldsymbol{b} \geqslant \boldsymbol{0}$ so that $(p_1, p_2, \ldots, p_m) = \boldsymbol{0}$. This contradicts the assumption that $\boldsymbol{p} \neq \boldsymbol{0}$. Hence, $p_{m+1} \neq 0$. Let $\bar{\boldsymbol{p}} = -\frac{1}{p_{m+1}}\boldsymbol{p}$ and $\bar{\boldsymbol{u}} = (\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_m)$. Then,

$$\bar{\boldsymbol{p}}^T\left(\begin{pmatrix} \boldsymbol{b} \\ \omega(\boldsymbol{b}) \end{pmatrix} - \begin{pmatrix} \bar{\boldsymbol{b}} \\ \omega(\bar{\boldsymbol{b}}) \end{pmatrix}\right) = \bar{\boldsymbol{u}}^T(\boldsymbol{b}-\bar{\boldsymbol{b}}) - (\omega(\boldsymbol{b}) - \omega(\bar{\boldsymbol{b}})) \geqslant 0. \tag{1}$$

We will show that $\bar{\boldsymbol{u}} \geqslant \boldsymbol{0}$. Suppose not. Then, there exists $\bar{u}_k < 0$ for some $k$. Let $\tilde{\boldsymbol{b}} = (\bar{b}_1, \ldots, \bar{b}_k + 1, \ldots, \bar{b}_m)$. Since $\tilde{\boldsymbol{b}} \geqslant \bar{\boldsymbol{b}}$, $\omega(\tilde{\boldsymbol{b}}) \geqslant \omega(\bar{\boldsymbol{b}})$. By Eq. (1), $\bar{\boldsymbol{u}}^T(\boldsymbol{b} - \bar{\boldsymbol{b}}) = \bar{u}_k \geqslant \omega(\tilde{\boldsymbol{b}}) - \omega(\bar{\boldsymbol{b}}) \geqslant 0$. This is a contradiction. Hence, $\bar{\boldsymbol{u}} \geqslant \boldsymbol{0}$.

Choose $\bar{\boldsymbol{x}} \in \{\boldsymbol{x} : \boldsymbol{x} \in \Omega, A\boldsymbol{x} \leqslant \bar{\boldsymbol{b}}\}$ such that $\boldsymbol{c}^T\bar{\boldsymbol{x}} = \omega(\bar{\boldsymbol{b}})$. Then, for all $\boldsymbol{x} \in \Omega$,

$$\begin{aligned}
&(\boldsymbol{c}^T\bar{\boldsymbol{x}} - \bar{\boldsymbol{u}}^T A\bar{\boldsymbol{x}}) - (\boldsymbol{c}^T\boldsymbol{x} - \bar{\boldsymbol{u}}^T A\boldsymbol{x}) \\
&= \boldsymbol{c}^T\bar{\boldsymbol{x}} - \boldsymbol{c}^T\boldsymbol{x} - \bar{\boldsymbol{u}}^T(A\bar{\boldsymbol{x}} - A\boldsymbol{x}) \\
&\geqslant \omega(\bar{\boldsymbol{b}}) - \boldsymbol{c}^T\boldsymbol{x} - \bar{\boldsymbol{u}}^T(\bar{\boldsymbol{b}} - A\boldsymbol{x}) \quad (\because A\bar{\boldsymbol{x}} \leqslant \bar{\boldsymbol{b}}) \\
&\geqslant \omega(\bar{\boldsymbol{b}}) - \omega(A\boldsymbol{x}) - \bar{\boldsymbol{u}}^T(\bar{\boldsymbol{b}} - A\boldsymbol{x}) \quad (\because \omega(A\boldsymbol{x}) \geqslant \boldsymbol{c}^T\boldsymbol{x}) \\
&\geqslant 0 \quad (\because \text{Eq. (1)}).
\end{aligned}$$

Hence, $\bar{\boldsymbol{x}} \in X(\bar{\boldsymbol{u}})$. Finally, we will show that $\bar{\boldsymbol{u}}^T(A\bar{\boldsymbol{x}} - \bar{\boldsymbol{b}}) = 0$ to finalize that $\bar{\boldsymbol{b}}$ is a Lagrangian capacity. In fact, $\bar{\boldsymbol{u}}^T(A\bar{\boldsymbol{x}} - \bar{\boldsymbol{b}}) \geqslant \omega(A\bar{\boldsymbol{x}}) - \omega(\bar{\boldsymbol{b}}) = \boldsymbol{c}^T\bar{\boldsymbol{x}} - \boldsymbol{c}^T\bar{\boldsymbol{x}} = 0$ and $\bar{\boldsymbol{u}}^T(A\bar{\boldsymbol{x}} - \bar{\boldsymbol{b}}) \leqslant 0$ ($\because \bar{\boldsymbol{u}} \geqslant \boldsymbol{0}$ and $A\bar{\boldsymbol{x}} - \bar{\boldsymbol{b}} \leqslant \boldsymbol{0}$). □

*A.5. Proof of Fact 1*

If $\bar{b}$ is a Lagrangian capacity, then there exists $\bar{u}$ such that $\bar{u}(\boldsymbol{a}^T\bar{\boldsymbol{x}} - \bar{b}) = 0$ and $\boldsymbol{a}^T\bar{\boldsymbol{x}} \leqslant \bar{b}$ for some $\bar{\boldsymbol{x}} \in X(\bar{u})$. If $\bar{u} = 0$, $\bar{\boldsymbol{x}} = (1, 1, \ldots, 1)^T$ and $\bar{b} \geqslant \sum_{j=1}^n a_j$. Then, $\bar{b} \in LC_2$. If $\bar{u} \neq 0$, $\boldsymbol{a}^T\bar{\boldsymbol{x}} - \bar{b} = 0$. Since $\boldsymbol{c}^T\boldsymbol{x} - \bar{u}\boldsymbol{a}^T\boldsymbol{x} = \sum_{j=1}^n x_j(c_j - \bar{u}a_j) = \sum_{j=1}^n x_j a_j (p_j - \bar{u})$, if $\bar{\boldsymbol{x}} \in X(\bar{u})$, $\bar{x}_j = 1$ whenever $j \in I_{\bar{u}}$ and $\bar{x}_j = 0$ whenever $j \notin I_{\bar{u}} \cup J_{\bar{u}}$. If $j \in J_{\bar{u}}$, the value of $\bar{x}_j$ does not care. Hence, $\bar{b} = \boldsymbol{a}^T\bar{\boldsymbol{x}} \in LC_1$.

Conversely, if $\bar{b} \in LC_1$, $\bar{b} = \sum_{j \in I_\lambda} a_j + \sum_{j \in K} a_j$ for some $\lambda \geqslant 0$ and $K \subseteq J_\lambda$. Let $\bar{x}_j = 1$ if $j \in I_\lambda \cup K$ and $\bar{x}_j = 0$ otherwise. Then, $\bar{\boldsymbol{x}} \in X(\lambda)$ and $\bar{b} = \boldsymbol{a}^T\bar{\boldsymbol{x}}$. Hence, $\bar{b}$ is a Lagrangian capacity. If $\bar{b} \in LC_2$, let $\bar{\boldsymbol{x}} = (1, 1, \ldots, 1)$. Then, $\bar{\boldsymbol{x}} \in X(0)$, $\bar{b} \geqslant \boldsymbol{a}^T\bar{\boldsymbol{x}}$, and $0(\boldsymbol{a}^T\bar{\boldsymbol{x}} - \bar{b}) = 0$. □

*A.6. Proof of Theorem 2*

Since $\boldsymbol{x} \in X(\boldsymbol{u})$,

$$\sum_{j=1}^n x_j\left(c_j - \sum_{i=1}^m u_i a_{ij}\right) \geqslant \sum_{j=1}^n x'_j\left(c_j - \sum_{i=1}^m u_i a_{ij}\right).$$

Similarly, $\boldsymbol{x}' \in X(\boldsymbol{u}')$,

$$\sum_{j=1}^n x'_j\left(c_j - \sum_{i=1}^m u'_i a_{ij}\right) \geqslant \sum_{j=1}^n x_j\left(c_j - \sum_{i=1}^m u'_i a_{ij}\right).$$

By summing the above two inequalities,

$$\begin{aligned}
&\sum_{j=1}^n x_j\left(c_j - \sum_{i=1}^m u_i a_{ij}\right) + \sum_{j=1}^n x'_j\left(c_j - \sum_{i=1}^m u'_i a_{ij}\right) \\
&\quad \geqslant \sum_{j=1}^n x'_j\left(c_j - \sum_{i=1}^m u_i a_{ij}\right) + \sum_{j=1}^n x_j\left(c_j - \sum_{i=1}^m u'_i a_{ij}\right).
\end{aligned}$$

By cancellation and multiplying both sides by $-1$,

$$\sum_{j=1}^n x_j \sum_{i=1}^m u_i a_{ij} + \sum_{j=1}^n x'_j \sum_{i=1}^d u'_i a_{ij} \leqslant \sum_{j=1}^n x'_j \sum_{i=1}^m u_i a_{ij} + \sum_{j=1}^n x_j \sum_{i=1}^m u'_i a_{ij}.$$

Since $u_k \neq u'_k$ and $u_i = u'_i$ for $i \neq k$,

$$\sum_{j=1}^n x_j u_k a_{kj} + \sum_{j=1}^n x'_j u'_k a_{kj} \leqslant \sum_{j=1}^n x'_j u_k a_{kj} + \sum_{j=1}^n x_j u'_k a_{kj}.$$

Substitute $b_k$ and $b'_k$ for $\sum_{j=1}^n x_j a_{kj}$ and $\sum_{j=1}^n x'_j a_{kj}$, respectively. Then,

$$u_k b_k + u'_k b'_k \leqslant u_k b'_k + u'_k b_k.$$

Now we obtain the following inequality:

$$(u_k - u'_k)(b_k - b'_k) \leqslant 0. \qquad \square$$

## References


Alves, M.J., Almeida, M., 2007. MOTGA: A multiobjective Tchebycheff based genetic algorithm for the multidimensional knapsack problem. Computers & Operations Research 34 (11), 3458–3470.

Bagchi, A., Bhattacharyya, N., Chakravarti, N., 1996. LP relaxation of the two dimensional knapsack problem with box and GUB constraints. European Journal of Operational Research 89 (3), 609–617.

Balev, S., Yanev, N., Fréville, A., Andonov, R., 2008. A dynamic programming based reduction procedure for the multidimensional 0-1 knapsack problem. European Journal of Operational Research 186 (1), 63–76.

Beasley, J.E., 1990a. A Lagrangean heuristic for set-covering problems. Naval Research Logistics 37 (1), 151–164.

Beasley, J.E., 1990b. OR-library: Distributing test problems by electronic mail. Journal of the Operational Research Society 41 (11), 1069–1072.

Beasley, J.E., 1996. Obtaining test problems via Internet. Journal of Global Optimization 8 (4), 429–433.

Bektas, T., Oguz, O., 2007. On separating cover inequalities for the multidimensional knapsack problem. Computers & Operations Research 34 (6), 1771–1776.

Bellmore, M., Greenberg, H.J., Jarvis, J.J., 1970. Generalized penalty function concepts in mathematical optimization. Operations Research 18 (2), 229–252.

Boussier, S., Vasquez, M., Vimont, Y., Hanafi, S., Michelon, P., 2010. A multi-level search strategy for the 0-1 multidimensional knapsack problem. Discrete Applied Mathematics 158 (2), 97–109.

Boyer, V., Elkihel, M., Baz, D.E., 2009. Heuristics for the 0-1 multidimensional knapsack problem. European Journal of Operational Research 199 (3), 658–664.

Brooks, R., Geoffrion, A., 1966. Finding Everett's Lagrange multipliers by linear programming. Operations Research 14 (6), 1149–1153.

Campello, R.E., Maculan, N., 1988. An $O(n^3)$ worst case bounded special LP knapsack (0-1) with two constraints. RAIRO. Recherche Opérationnelle 22 (1), 27–32.

Caprara, A., Kellerer, H., Pferschy, U., Pisinger, D., 2000. Approximation algorithms for knapsack problems with cardinality constraints. European Journal of Operational Research 123 (2), 333–345.

Chang, Y.-J., Wah, B.W., 1995. Lagrangian techniques for solving a class of zero-one integer linear programs. In: Proceedings of the 19th International Computer Software and Applications Conference, pp. 156–161.

Chekuri, C., Khanna, S., 2000. A PTAS for the multiple knapsack problem. In: Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 213–222.

Chu, P.C., Beasley, J.E., 1998. A genetic algorithm for the multidimensional knapsack problem. Journal of Heuristics 4 (1), 63–86.

Cotta, C., Troya, J.M., 1997. A hybrid genetic algorithm for the 0-1 multiple knapsack problem. In: Smith, G.D., Steele, N.C., Albrecht, R.F. (Eds.), Artificial Neural Networks and Genetic Algorithms 3. Springer-Verlag, pp. 251–255.

Crama, Y., Mazzola, J.B., 1994. On the strength of relaxations of multidimensional knapsack problems. INFOR 32 (4), 219–225.

Everett III, H., 1963. Generalized Lagrange multiplier method for solving problems of optimum allocation of resources. Operations Research 11 (3), 399–417.

Fisher, M.L., 1981. The Lagrangian relaxation method for solving integer programming problems. Management Science 27 (1), 1–18.

Fleszar, K., Hindi, K.S., 2009. Fast, effective heuristics for the 0-1 multi-dimensional knapsack problem. Computers & Operations Research 36 (5), 1602–1607.

Fréville, A., 2004. The multidimensional 0-1 knapsack problem: An overview. European Journal of Operational Research 155 (1), 1–21.

Fréville, A., Hanafi, S., 2005. The multidimensional 0-1 knapsack problem – bounds and computational aspects. Annals of Operations Research 139 (1), 195–227.

Fréville, A., Lorena, L.A.N., Plateau, G., 1990. Efficient subgradient algorithms for the 0-1 multiknapsack Lagrangean and surrogate duals. Tech. Rep. Research Report LIPN, University of Paris 13.

Fréville, A., Plateau, G., 1993. An exact search for the solution of the surrogate dual of the 0-1 bidimensional knapsack problem. European Journal of Operational Research 68 (3), 413–421.

Fréville, A., Plateau, G., 1994. An efficient preprocessing procedure for the multidimensional 0-1 knapsack problem. Discrete Applied Mathematics 49 (1–3), 189–212.

Fréville, A., Plateau, G., 1996. The 0-1 bidimensional knapsack problem: Toward an efficient high-level primitive tool. Journal of Heuristics 2 (2), 147–167.

Frieze, A.M., Clarke, M.R.B., 1984. Approximation algorithms for the $m$-dimensional 0-1 knapsack problem: Worst-case and probabilistic analyses. European Journal of Operational Research 15 (1), 100–109.

Garey, M., Johnson, D.S., 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman and Company.

Gavish, B., 1978. On obtaining the 'best' multipliers for a Lagrangean relaxation for integer programming. Computers & Operations Research 5 (1), 55–71.

Gavish, B., Pirkul, H., 1985. Efficient algorithms for solving multiconstraint zero-one knapsack problems to optimality. Mathematical Programming 31 (1), 78–105.

Geoffrion, A.M., 1974. Lagrangian relaxation for integer programming. Mathematical Programming Study 2, 82–114.

Glover, F., 1965. A multiphase-dual algorithm for the zero-one integer programming problem. Operations Research 13 (6), 879–919.

Glover, F., 1968. Surrogate constraints. Operations Research 16 (4), 741–749.

Glover, F., 1989. Tabu search – Part I. ORSA Journal on Computing 1 (3), 190–206.

Glover, F., Kochenberger, G., 1996. Critical event tabu search for multidimensional knapsack problems. In: Metaheuristics: Theory and Applications. Kluwer Academic Publishers, pp. 407–427.

Goldberg, D., 1989. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley.

Gould, F.J., 1969. Extensions of Lagrange multipliers in nonlinear programming. SIAM Journal on Applied Mathematics 17 (6), 1280–1297.

Greenberg, H.J., Pierskalla, W.P., 1970. Surrogate mathematical programming. Operations Research 18 (5), 924–939.

Hanafi, S., Fréville, A., 1998. An efficient tabu search approach for the 0-1 multidimensional knapsack problem. European Journal of Operational Research 106 (2–3), 659–675.

Hanafi, S., Wilbaut, C., 2008. Scatter search for the 0-1 multidimensional knapsack problem. Journal of Mathematical Modelling and Algorithms 7 (2), 143–159.

Hanafi, S., Wilbaut, C., 2011. Improved convergent heuristic for the 0-1 multidimensional knapsack problem. Annals of Operations Research 183 (1), 125–142.

Hill, R.R., Cho, Y.K., Moore, J.T., 2012. Problem reduction heuristic for the 0-1 multidimensional knapsack problem. Computers & Operations Research 39 (1), 19–26.

Kaparis, K., Letchford, A.N., 2008. Local and global lifted cover inequalities for the 0-1 multidimensional knapsack problem. European Journal of Operational Research 186 (1), 91–103.

Karwan, M.H., Rardin, R.L., 1979. Some relationships between Lagrangian and surrogate duality in integer programming. Mathematical Programming 17 (1), 320–334.

Karwan, M.H., Rardin, R.L., 1980. Searchability of the composite and multiple surrogate dual functions. Operations Research 28 (5), 1251–1257.

Kellerer, H., Pferschy, U., 1998. A new fully polynomial approximation scheme for the knapsack problem. In: Proceedings of the First International Workshop on Approximation Algorithms for Combinatorial Optimization, pp. 123–134.

Kellerer, H., Pferschy, U., Pisinger, D., 2004. Knapsack Problems. Springer-Verlag.

Kong, M., Tian, P., Kao, Y., 2008. A new ant colony optimization algorithm for the multidimensional knapsack problem. Computers & Operations Research 35 (8), 2672–2683.

Lawler, E.L., 1977. Fast approximation algorithms for knapsack problems. In: Proceedings of the 18th Annual Symposium on Foundations of Computer Science, pp. 206–213.

Li, V.C., 2005. Tight oscillations tabu search for multidimensional knapsack problems with generalized upper bound constraints. Computers & Operations Research 32 (11), 2843–2852.

Li, V.C., Curry, G.L., 2005. Solving multidimensional knapsack problems with generalized upper bound constraints using critical event tabu search. Computers & Operations Research 32 (4), 825–848.

Løkketangen, A., Glover, F., 1996. Probabilistic move selection in tabu search for zero-one mixed integer programming problems. In: Metaheuristics: Theory and Applications. Kluwer Academic Publishers, pp. 467–487.

Luenberger, D.G., 1969. Optimization by Vector Space Methods. John Wiley & Sons, Inc.

Magazine, M.J., Oguz, O., 1984. A heuristic algorithm for the multidimensional zero-one knapsack problem. European Journal of Operational Research 16 (3), 319–326.

Martello, S., Toth, P., 1990. Knapsack Problems: Algorithms and Computer Implementations. John Wiley & Sons, Inc.

Martello, S., Toth, P., 2003. An exact algorithm for the two-constraint 0-1 knapsack problem. Operations Research 51 (5), 826–835.

Martin, R.K., 1999. Large Scale Linear and Integer Optimization: A Unified Approach. Kluwer Academic Publishers.

Nemhauser, G.L., Wolsey, L.A., 1999. Integer and Combinatorial Optimization. John Wiley & Sons, Inc.

Osorio, M.A., Glover, F., Hammer, P., 2002. Cutting and surrogate constraint analysis for improved multidimensional knapsack solutions. Annals of Operations Research 117 (1–4), 71–93.

Pirkul, H., 1987. A heuristic solution procedure for the multiconstraint 0-1 knapsack problem. Naval Research Logistics 34 (2), 161–172.

Raidl, G.R., 1998. An improved genetic algorithm for the multiconstrained 0-1 knapsack problem. In: Proceedings of the IEEE International Conference on Evolutionary Computation. pp. 207–211.

Raidl, G.R., 1999. Weight-codings in a genetic algorithm for the multiconstraint knapsack problem. In: Proceedings of the Congress on Evolutionary Computation. vol. 1, pp. 596–603.

Sahni, S., 1975. Approximate algorithms for the 0/1 knapsack problem. Journal of the ACM 22 (1), 115–124.

Schuurmans, D., Southey, F., Holte, R.C., 2001. The exponentiated subgradient algorithm for heuristic boolean programming. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence. vol. 1, pp. 334–341.

Shapiro, J.F., 1979. A survey of Lagrangian techniques for discrete optimization. Annals of Discrete Mathematics 5, 113–138.

Shih, W., 1979. A branch and bound method for the multiconstraint zero-one knapsack problem. Journal of the Operational Research Society 30 (4), 369–378.

Thiel, J., Voss, S., 1994. Some experiences on solving multiconstraint zero-one knapsack problems with genetic algorithms. INFOR 32 (4), 226–242.

Thiongane, B., Nagih, A., Plateau, G., 2006. Lagrangean heuristics combined with reoptimization for the 0-1 bidimensional knapsack problem. Discrete Applied Mathematics 154 (15), 2200–2211.

Vasquez, M., Hao, J.-K., 2001. A hybrid approach for the 0-1 multidimensional knapsack problem. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence. vol. 1, pp. 328–333.

Vasquez, M., Vimont, Y., 2005. Improved results on the 0-1 multidimensional knapsack problem. European Journal of Operational Research 165 (1), 70–81.

Vimont, Y., Boussier, S., Vasquez, M., 2008. Reduced costs propagation in an efficient implicit enumeration for the 0-1 multidimensional knapsack problem. Journal of Combinatorial Optimization 15 (2), 165–178.

Wah, B.W., Shang, Y., 1997. A discrete Lagrangian-based global-search method for solving satisfiability problems. In: Satisfiability Problem: Theory and Applications. DIMACS Series in Discrete Mathematics and Theoretical Computer Science. American Mathematical Society, pp. 365–392.

Wah, B.W., Wu, Z., 1999. The theory of discrete Lagrange multipliers for nonlinear discrete optimization. In: Principles and Practice of Constraint Programming, pp. 28–42.

Weingartner, H.M., Ness, D.N., 1967. Methods for the solution of the multidimensional 0/1 knapsack problem. Operations Research 15 (1), 83–103.

Wilbaut, C., Hanafi, S., 2009. New convergent heuristics for 0-1 mixed integer programming. European Journal of Operational Research 195 (1), 62–74.

Wilbaut, C., Salhi, S., Hanafi, S., 2009. An iterative variable-based fixation heuristic for the 0-1 multidimensional knapsack problem. European Journal of Operational Research 199 (2), 339–348.

Wolsey, L.A., 1998. Integer Programming. John Wiley & Sons, Inc.

Yoon, Y., Kim, Y.-H., Moon, B.-R., 2005. An evolutionary Lagrangian method for the 0/1 multiple knapsack problem. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 629–635.

Yu, G., 1990. Algorithms for optimizing piecewise linear functions and for degree constrained minimum spanning tree problems. Ph.D. Thesis, University of Pennsylvania.