An Evolutionary Lagrangian Method for the 0/1 Multiple Knapsack Problem

Yourim Yoon
School of Computer Science & Engineering
Seoul National University
Sillim-dong, Gwanak-gu,
Seoul, 151-744 Korea
yryoon@soar.snu.ac.kr

Yong-Hyuk Kim
School of Mechanical & Aerospace Engineering
Seoul National University
Sillim-dong, Gwanak-gu,
Seoul, 151-744 Korea
yhdfly@soar.snu.ac.kr

Byung-Ro Moon
School of Computer Science & Engineering
Seoul National University
Sillim-dong, Gwanak-gu,
Seoul, 151-744 Korea
moon@soar.snu.ac.kr

ABSTRACT

We propose a new evolutionary approach to solve the 0/1 multiple knapsack problem. We approach the problem from a new viewpoint different from traditional methods. The most remarkable feature is the Lagrangian method. Lagrange multipliers transform the problem, keeping the optimality as well as decreasing the complexity. However, it is not easy to find Lagrange multipliers nearest to the constraints of the problem. We propose an evolution strategy to find the optimal Lagrange multipliers. Also, we improve the evolution strategy by adjusting its objective function properly. We show the efficiency of the proposed methods by the experiments. We make comparisons with existing general approach on well-known benchmark data.

Categories and Subject Descriptors

G.1.6 [**Mathematics of Computing**]: NUMERICAL ANALYSIS —*Optimization*

General Terms

Algorithms, Experimentation, Theory

Keywords

Zero/one multiple knapsack problem, Lagrange multiplier, Evolution strategy

1. INTRODUCTION

The 0/1 multiple knapsack problem (0/1MKP) is a well-known NP-complete problem [10] which is formally defined as follows:

$$\text{maximize } \boldsymbol{v}^T\boldsymbol{x} \quad \text{subject to} \quad \boldsymbol{x} \in \{0,1\}^n,\ W\boldsymbol{x} \leq \boldsymbol{b},$$

where the weight $W = (w_{ij})$ is a $d \times n$ matrix with no negative elements, the value $\boldsymbol{v}$ is an n-dimensional vector, and the capacity $\boldsymbol{b}$ is a d-dimensional vector.

The knapsack problem has a number of applications in various fields, e.g., cryptography, economy, etc. For the knapsack problem with only one constraint, a simple heuristic algorithm works well and there have been a number of researches about the efficient approximation algorithm to find the near-optimal solution [2, 16, 18, 27]. In this paper, we are interested in the problem with more than one constraint, i.e., the multiple knapsack problem. The method to solve the multiple knapsack problem has been extensively studied in the past [4, 8, 9, 12, 19, 21, 31]. Also, a number of genetic algorithms [14] to solve the problem have been proposed [5, 6, 7, 15, 17, 25, 30].

However, most researches directly deal with the discrete search space. In this paper, we try out a method that transforms the search space of the problem to another space and searches the solution in the transformed space instead of managing the original space directly. Zero/one MKP is the optimization problem with constraints. We transform it using the Lagrange multipliers [20]. However, we have a lot of limitations since the domain is not continuous but discrete. There have been a number of papers that studied Lagrange multiplier method for the discrete problems [3, 11, 13, 28, 29, 32, 33, 34]. There were also a few methods that used Lagrange multipliers for 0/1MKP. Typically, most of them found just the upper bound and could not find the exact solution [22]. One of the Lagrangian methods to find the lower bound is the constructive heuristic proposed by Magazine and Oguz [21] (MO-CONS). There was also the method that improved the performance by combining MO-CONS with genetic algorithms [24]. It used the real-valued weight-codings to make a variant of the original problem and then applied MO-CONS. It provided a new viewpoint to solve 0/1MKP, but it just used MO-CONS for evaluation and did not research in the aspect of Lagrange multipliers. In this paper, we study the properties of Lagrange multipliers and also propose a variant of MO-CONS. However, it is not easy to obtain good solutions by just using the heuristic. So we use the Lagrangian method combined with the evolution strategy [1] and improve it by using the properties of Lagrange multipliers.

The remainder of this paper is organized as follows. We propose Lagrangian method for 0/1MKP in Section 2. In

GECCO'05, June 25–29, 2005, Washington, DC, USA.

Section 3, we describe Lagrange duality and the limitations arising from the fact that the domain of the problem is not continuous. The tendency for searching optimal Lagrange multipliers effectively is discussed in Section 4. We describe a constructive heuristic to optimize Lagrange multipliers in Section 5 and propose the method based on evolution strategy in Section 6. Finally, we present the experimental results in Section 7 and make conclusions in Section 8.

2. LAGRANGE MULTIPLIERS FOR 0/1MKP

Consider the following maximization problem with constraints:

$$\text{maximize } f(\boldsymbol{x}) \quad \text{subject to} \quad \boldsymbol{x} \in \Omega,\ \boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{0}.$$

Assume that there exists the maximum μ_0. If the domain Ω is convex, the real vector $\boldsymbol{\lambda} \geq \boldsymbol{0}$ such that

$$\mu_0 = \max_{\boldsymbol{x}\in\Omega} \{f(\boldsymbol{x}) - \langle \boldsymbol{\lambda}, \boldsymbol{g}(\boldsymbol{x})\rangle\}$$

always exists, where $\langle \cdot, \cdot \rangle$ is an inner product [20]. In this case, if the objective function $f(\boldsymbol{x})$ achieves the maximum when $\boldsymbol{x}$ is $\boldsymbol{x_0}$, then $\langle \boldsymbol{\lambda}, \boldsymbol{g}(\boldsymbol{x_0})\rangle = 0$.

In particular, if f and $\boldsymbol{g}$ are differentiable, $\nabla f(\boldsymbol{x_0}) = \lambda_i \nabla g_i(\boldsymbol{x_0})$ holds for each i. If we are lucky, we can easily find the value of $\boldsymbol{\lambda}$ for the problem and solve the problem by transforming the original problem with constraints into the easier one without constraints. The method that solves the problem by finding $\boldsymbol{\lambda}$ and transforming the given constrained problem into the unconstrained problem is called *Lagrange multiplier method*.

Zero/one MKP is also the maximization problem with constraints in the following formulation:

$$\text{maximize } \boldsymbol{v}^T\boldsymbol{x} \quad \text{subject to} \quad \boldsymbol{x} \in \{0,1\}^n,\ W\boldsymbol{x} - \boldsymbol{b} \leq \boldsymbol{0}.$$

We can not always apply Lagrange multiplier method to 0/1MKP because the domain $\{0,1\}^n$ is not convex. However, for some constraints, their corresponding $\boldsymbol{\lambda}$'s exist. Assume that the real vector $\boldsymbol{\lambda}$ corresponding to the constraints is given. Then, it is possible to transform the original optimization problem into the following problem using Lagrange multipliers:

$$\text{maximize } \{\boldsymbol{v}^T\boldsymbol{x} - \langle \boldsymbol{\lambda}, W\boldsymbol{x} - \boldsymbol{b}\rangle\} \quad \text{subject to} \quad \boldsymbol{x} \in \{0,1\}^n.$$

It is easy to find the maximum of the transformed problem using the following formula.

$$\begin{aligned}
& \boldsymbol{v}^T\boldsymbol{x} - \langle \boldsymbol{\lambda}, W\boldsymbol{x} - \boldsymbol{b}\rangle \\
= & \sum_{i=1}^n v_i x_i - \sum_{j=1}^d \lambda_j \left(\sum_{i=1}^n w_{ji} x_i\right) + \langle \boldsymbol{\lambda}, \boldsymbol{b}\rangle \\
= & \sum_{i=1}^n v_i x_i - \sum_{j=1}^d \sum_{i=1}^n \lambda_j w_{ji} x_i + \langle \boldsymbol{\lambda}, \boldsymbol{b}\rangle \\
= & \sum_{i=1}^n v_i x_i - \sum_{i=1}^n \sum_{j=1}^d \lambda_j w_{ji} x_i + \langle \boldsymbol{\lambda}, \boldsymbol{b}\rangle \\
= & \sum_{i=1}^n \left(v_i x_i - \sum_{j=1}^d \lambda_j w_{ji} x_i\right) + \langle \boldsymbol{\lambda}, \boldsymbol{b}\rangle \\
= & \sum_{i=1}^n x_i \left(v_i - \sum_{j=1}^d \lambda_j w_{ji}\right) + \langle \boldsymbol{\lambda}, \boldsymbol{b}\rangle.
\end{aligned}$$

```
LMMKP(λ)
{
    for i = 1 to n
        if v_i > Σ_{j=1}^d λ_j w_ji then x*_i = 1;
        else x*_i = 0;
    b* = W x*;
    μ* = v^T x*;
    return μ*, x*, and b*;
}
```

$\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_d) \in \mathbb{R}^d$

Figure 1: Lagrangian method for the 0/1 multiple knapsack problem

To maximize the above formula for the fixed $\boldsymbol{\lambda}$, we have to set x_i to be 1 only if $v_i > \sum_{j=1}^d \lambda_j w_{ji}$ for each i. Since each v_i does not have an effect on the others, getting the maximum is fairly easy. Since this algorithm computes just $\sum_{j=1}^d \lambda_j w_{ji}$ for each i, its time complexity becomes $O(nd)$.

If we only find out $\boldsymbol{\lambda}$ for the problem, we get the optimal solution of 0/1MKP in polynomial time. We may have the problem that such $\boldsymbol{\lambda}$ never exists or it is difficult to find it although it exists. However, this method is not entirely useless. For arbitrary $\boldsymbol{\lambda}$, let the maximum of the above formula be μ^* and the vector $\boldsymbol{x}$ which achieves the maximum be $\boldsymbol{x^*}$. Since $\boldsymbol{\lambda}$ is chosen arbitrarily, we do not guarantee that $\boldsymbol{x^*}$ satisfies the constraints of the original problem. Nevertheless, letting the capacity be $\boldsymbol{b^*} = W\boldsymbol{x^*}$ instead of $\boldsymbol{b}$ makes μ^* be the optimal solution by Theorem 1. We call this algorithm *Lagrangian method for the 0/1 multiple knapsack problem(LMMKP)*. Figure 1 shows this algorithm.

THEOREM 1. *The vector $\boldsymbol{x}^*$ obtained by applying LMMKP with given $\boldsymbol{\lambda}$ is the maximizer of the following problem:*

$$\textit{maximize } \boldsymbol{v}^T\boldsymbol{x} \quad \textit{subject to} \quad \boldsymbol{x} \in \{0,1\}^n,\ W\boldsymbol{x} \leq \boldsymbol{b}^*.$$

PROOF: Let $\boldsymbol{x}$ be an arbitrary element in $\{0,1\}^n$ satisfying $W\boldsymbol{x} \leq \boldsymbol{b}^*$. Then, for each i,

$$x_i^* \left(v_i - \sum_{j=1}^d \lambda_j w_{ji}\right) \geq x_i \left(v_i - \sum_{j=1}^d \lambda_j w_{ji}\right)$$

by LMMKP.

$$\begin{aligned}
\boldsymbol{v}^T\boldsymbol{x}^* &= \sum_{i=1}^n x_i^* \left(v_i - \sum_{j=1}^d \lambda_j w_{ji}\right) + \langle \boldsymbol{\lambda}, \boldsymbol{b}^*\rangle \\
&\geq \sum_{i=1}^n x_i \left(v_i - \sum_{j=1}^d \lambda_j w_{ji}\right) + \langle \boldsymbol{\lambda}, \boldsymbol{b}^*\rangle \\
&= \boldsymbol{v}^T\boldsymbol{x} - \langle \boldsymbol{\lambda}, W\boldsymbol{x} - \boldsymbol{b}^*\rangle \\
&\geq \boldsymbol{v}^T\boldsymbol{x}. \quad (\because \boldsymbol{\lambda} \geq \boldsymbol{0} \ \&\ W\boldsymbol{x} - \boldsymbol{b}^* \leq \boldsymbol{0}.) \quad \boxtimes
\end{aligned}$$

In particular, in the case that λ_k is 0, the k^{th} constraint is ignored. That is, $\boldsymbol{x}^*$ is the maximizer of the problems which have the capacities $\boldsymbol{c}$'s such that $c_k \geq b_k^*$ and $c_i = b_i^*$ for all $i \neq k$. In general, the following theorem holds.

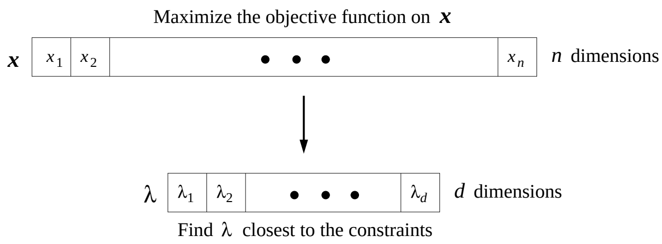

Figure 2: The alternative way to solve 0/1MKP
(n : the number of objects, d: the number of constraints)

THEOREM 2. *In particular, if LMMKP is applied with $\boldsymbol{\lambda}$ such that $\lambda_{k_1} = 0, \lambda_{k_2} = 0, \ldots,$ and $\lambda_{k_m} = 0$, replacing the capacity $\boldsymbol{b^*}$ by $\boldsymbol{c}$ such that*

$$\begin{cases} c_i = b_i^*, & \text{if } i \neq k_j \text{ for all } j, \\ c_i \geq b_i^*, & \text{otherwise} \end{cases}$$

in Theorem 1 makes the theorem still hold.

PROOF: Omitted since it is similar to Theorem 1.

Instead of finding the optimal solution of the original 0/1MKP directly, we consider the problem of finding $\boldsymbol{\lambda}$ corresponding to given constraints. That is, we transform the problem of dealing with n-dimensional binary vector $\boldsymbol{x}$ into the one of dealing with d-dimensional real vector $\boldsymbol{\lambda}$ (see Figure 2). If there are Lagrange multipliers corresponding to given constraints and we find them, we easily get the optimal solution of 0/1MKP. If there are no such Lagrange multipliers, we try to get the solution close to the optimum by devoting to find Lagrange multipliers which satisfy given constraints and are nearest to them.

3. GAPS AND DUALITY

Lagrange multipliers may not exist for all 0/1MKP instances. Now, we investigate the constraints whose corresponding Lagrange multipliers do not exist.

3.1 Notations

We denote several notations for convenience of describing the next definition and theorem.

$$\Omega = \{0,1\}^n$$
$$\omega(\boldsymbol{z}) = \max_{\boldsymbol{x}\in\Omega, W\boldsymbol{x}\leq\boldsymbol{z}} \boldsymbol{v}^T\boldsymbol{x}$$
$$\varphi^{\boldsymbol{b}}(\boldsymbol{\lambda}) = \max_{\boldsymbol{x}\in\Omega}\{\boldsymbol{v}^T\boldsymbol{x} - \langle\boldsymbol{\lambda}, W\boldsymbol{x}-\boldsymbol{b}\rangle\}$$
$$x(\boldsymbol{\lambda}) = \arg\max_{\boldsymbol{x}\in\Omega}\{\boldsymbol{v}^T\boldsymbol{x} - \langle\boldsymbol{\lambda}, W\boldsymbol{x}-\boldsymbol{b}\rangle\}$$
$$b(\boldsymbol{\lambda}) = W\cdot x(\boldsymbol{\lambda})$$

$\boldsymbol{\lambda}$ corresponds to $\{x(\boldsymbol{\lambda}), b(\boldsymbol{\lambda})\}$ by the LMMKP.

3.2 Gaps

The condition for the capacity $\boldsymbol{c}$ of Theorem 2 means that $\langle\boldsymbol{\lambda}, W\boldsymbol{x}^* - \boldsymbol{c}\rangle = 0$ and $W\boldsymbol{x}^* \leq \boldsymbol{c}$. Using the above notation, $W\boldsymbol{x}^*$ is equal to $b(\boldsymbol{\lambda})$. We define *gap* using $b(\boldsymbol{\lambda})$.

DEFINITION 3 (GAP). *A capacity $\boldsymbol{b}$ is called* gap *if there is no $\boldsymbol{\lambda} \geq \boldsymbol{0}$ such that $\langle\boldsymbol{\lambda}, b(\boldsymbol{\lambda}) - \boldsymbol{b}\rangle = 0$ and $b(\boldsymbol{\lambda}) \leq \boldsymbol{b}$.*

Intuitively, gap is the capacity of the constraints whose corresponding Lagrange multipliers do not exist. So, if the capacity of given 0/1MKP is a gap, LMMKP does not guarantee the optimal solution. Instead, we can try to find the nearest solution to the optimum by searching the most similar capacity of the constraints whose corresponding Lagrange multipliers exist. The gap is highly related with the property of duality in the next subsection.

3.3 Lagrange Duality

If Ω is convex, it is known that the maximum of the objective function subject to the constraints is the same as the minimum of $\varphi^{\boldsymbol{b}}(\boldsymbol{\lambda})$ for $\boldsymbol{\lambda}$ [20]. This property is called *duality.* Since the domain of 0/1MKP is not convex, this property does not hold in general. Nevertheless, we can compare the values of the optimum and $\varphi^{\boldsymbol{b}}(\boldsymbol{\lambda})$ in the following theorem. This is an example of weak duality in general integer programming [23, 26].

THEOREM 4 (WEAK DUALITY).

$$\max_{\boldsymbol{z}\leq\boldsymbol{b}} \omega(\boldsymbol{z}) \leq \min_{\boldsymbol{\lambda}\geq\boldsymbol{0}} \varphi^{\boldsymbol{b}}(\boldsymbol{\lambda}).$$

PROOF: For any $\boldsymbol{\lambda} \geq \boldsymbol{0}$,

$$\begin{aligned} \varphi^{\boldsymbol{b}}(\boldsymbol{\lambda}) &= \max_{\boldsymbol{x}\in\Omega}\{\boldsymbol{v}^T\boldsymbol{x} - \langle\boldsymbol{\lambda}, W\boldsymbol{x}-\boldsymbol{b}\rangle\} \\ &\geq \max_{\boldsymbol{x}\in\Omega, W\boldsymbol{x}\leq\boldsymbol{b}}\{\boldsymbol{v}^T\boldsymbol{x} - \langle\boldsymbol{\lambda}, W\boldsymbol{x}-\boldsymbol{b}\rangle\} \\ &\geq \max_{\boldsymbol{x}\in\Omega, W\boldsymbol{x}\leq\boldsymbol{b}} \boldsymbol{v}^T\boldsymbol{x} \\ &= \omega(\boldsymbol{b}) \\ &= \max_{\boldsymbol{z}\leq\boldsymbol{b}} \omega(\boldsymbol{z}). \quad (\because \omega \text{ is increasing.}) \end{aligned}$$

$$\therefore \quad \max_{\boldsymbol{z}\leq\boldsymbol{b}} \omega(\boldsymbol{z}) \leq \min_{\boldsymbol{\lambda}\geq\boldsymbol{0}} \varphi^{\boldsymbol{b}}(\boldsymbol{\lambda}). \quad \boxtimes$$

From this theorem, the minimum of $\varphi^{\boldsymbol{b}}(\boldsymbol{\lambda})$ is greater than or equal to the maximum of the original problem. We use this fact to find out a good upper bound for the optimal solution of 0/1MKP. The (near) minimum among $\varphi^{\boldsymbol{b}}(\boldsymbol{\lambda})$'s is chosen for an upper bound.

If $\boldsymbol{b}$ is not a gap, the property of general duality holds. This is given in the next corollary.

COROLLARY 5 (STRONG DUALITY). *If the capacity $\boldsymbol{b}$ is not a gap,*

$$\max_{\boldsymbol{z}\leq\boldsymbol{b}} \omega(\boldsymbol{z}) = \min_{\boldsymbol{\lambda}\geq\boldsymbol{0}} \varphi^{\boldsymbol{b}}(\boldsymbol{\lambda}).$$

PROOF: Since $\boldsymbol{b}$ is not a gap, there exists a $\boldsymbol{\lambda}_0 \geq \boldsymbol{0}$ such that $\langle\boldsymbol{\lambda}_0, b(\boldsymbol{\lambda}_0) - \boldsymbol{b}\rangle = 0$ and $b(\boldsymbol{\lambda}_0) \leq \boldsymbol{b}$. Then,

$$\begin{aligned} \min_{\boldsymbol{\lambda}\geq\boldsymbol{0}} \varphi^{\boldsymbol{b}}(\boldsymbol{\lambda}) &\leq \varphi^{\boldsymbol{b}}(\boldsymbol{\lambda}_0) = \boldsymbol{v}^T\boldsymbol{x}(\boldsymbol{\lambda}_0) = \omega(b(\boldsymbol{\lambda}_0)) \\ &\leq \omega(\boldsymbol{b}) = \max_{\boldsymbol{z}\leq\boldsymbol{b}} \omega(\boldsymbol{z}). \end{aligned}$$

Hence, by Theorem 4,

$$\max_{\boldsymbol{z}\leq\boldsymbol{b}} \omega(\boldsymbol{z}) = \min_{\boldsymbol{\lambda}\geq\boldsymbol{0}} \varphi^{\boldsymbol{b}}(\boldsymbol{\lambda}). \quad \boxtimes$$

4. THE TENDENCY OF THE LAGRANGE MULTIPLIERS FOR 0/1MKP

If we are to find the solution of 0/1MKP using our Lagrangian method, we have to choose the maximum of $\boldsymbol{v}^T x(\boldsymbol{\lambda})$ for nonnegative real vector $\boldsymbol{\lambda}$ such that $b(\boldsymbol{\lambda}) \leq \boldsymbol{b}$. However, if we use randomly generated Lagrange multipliers, they may not satisfy the constraints or it is probably hard to find the capacity close to the original one. The following theorem makes it easy to adjust Lagrange multipliers.

THEOREM 6. *Suppose that $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}'$ corresponds to $\{\boldsymbol{x}, \boldsymbol{c}\}$ and $\{\boldsymbol{x}', \boldsymbol{c}'\}$ by the LMMKP, respectively. Let $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_d)$ and $\boldsymbol{\lambda}' = (\lambda'_1, \lambda'_2, \ldots, \lambda'_d)$, where $\lambda_i = \lambda'_i$ for $i \neq k$ and $\lambda_k \neq \lambda'_k$. Then, if $\lambda_k < \lambda'_k$, $c_k \geq c'_k$ and if $\lambda_k > \lambda'_k$, $c_k \leq c'_k$.*

PROOF: Since $\boldsymbol{\lambda}$ corresponds to $\boldsymbol{x}$,

$$\sum_{i=1}^{n} x_i \left(v_i - \sum_{j=1}^{d} \lambda_j w_{ji} \right) \geq \sum_{i=1}^{n} x'_i \left(v_i - \sum_{j=1}^{d} \lambda_j w_{ji} \right).$$

Similarly, since $\boldsymbol{\lambda}'$ corresponds to $\boldsymbol{x}'$,

$$\sum_{i=1}^{n} x'_i \left(v_i - \sum_{j=1}^{d} \lambda'_j w_{ji} \right) \geq \sum_{i=1}^{n} x_i \left(v_i - \sum_{j=1}^{d} \lambda'_j w_{ji} \right).$$

By summing above two inequalities,

$$\sum_{i=1}^{n} x_i \left(v_i - \sum_{j=1}^{d} \lambda_j w_{ji} \right) + \sum_{i=1}^{n} x'_i \left(v_i - \sum_{j=1}^{d} \lambda'_j w_{ji} \right) \geq \sum_{i=1}^{n} x'_i \left(v_i - \sum_{j=1}^{d} \lambda_j w_{ji} \right) + \sum_{i=1}^{n} x_i \left(v_i - \sum_{j=1}^{d} \lambda'_j w_{ji} \right).$$

By cancellation and multiplying both sides by -1,

$$\sum_{i=1}^{n} x_i \sum_{j=1}^{d} \lambda_j w_{ji} + \sum_{i=1}^{n} x'_i \sum_{j=1}^{d} \lambda'_j w_{ji} \leq \sum_{i=1}^{n} x'_i \sum_{j=1}^{d} \lambda_j w_{ji} + \sum_{i=1}^{n} x_i \sum_{j=1}^{d} \lambda'_j w_{ji}.$$

Since $\lambda_k \neq \lambda'_k$ and $\lambda_i = \lambda'_i$ for $i \neq k$,

$$\sum_{i=1}^{n} x_i \lambda_k w_{ki} + \sum_{i=1}^{n} x'_i \lambda'_k w_{ki} \leq \sum_{i=1}^{n} x'_i \lambda_k w_{ki} + \sum_{i=1}^{n} x_i \lambda'_k w_{ki}.$$

Substitute c_k for $\sum_{i=1}^{n} x_i w_{ki}$ and c'_k for $\sum_{i=1}^{n} x'_i w_{ki}$. Then,

$$\lambda_k c_k + \lambda'_k c'_k \leq \lambda_k c'_k + \lambda'_k c_k.$$

Now, we obtain the following inequality.

$$(\lambda_k - \lambda'_k)(c_k - c'_k) \leq 0. \qquad \boxtimes$$

Let $\boldsymbol{b} = (b_1, b_2, \ldots, b_d)$ and $W\boldsymbol{x}^* = (b^*_1, b^*_2, \ldots, b^*_d)$ for $\boldsymbol{x}^*$ that is obtained by LMMKP with $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_d)$. By the above theorem, if $b^*_k > b_k$, choosing $\boldsymbol{\lambda}' = (\lambda_1, \ldots, \lambda'_k, \ldots, \lambda_d)$ such that $\lambda'_k > \lambda_k$ and applying LMMKP with $\boldsymbol{\lambda}'$ makes the value of b^*_k smaller. It makes the k^{th} constraint satisfied or the exceeded capacity decreased. Of course, another constraint may become violated by this operation. Also, which λ_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 λ_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.

```
CH(MKP instance)
{
        λ = 0;
        I = {1, 2, ..., n};
        do
                k = random integer in [1, d];
                for i ∈ I
                        α_i = (v_i − Σ_{j=1}^{d} λ_j w_ji)/w_ki;
                λ_k = λ_k + min_{i∈I} α_i;
                I = I \ {argmin_{i∈I} α_i};
                (μ*, x*, b*) = LMMKP(λ);
        until x* satisfies all the constraints;
        return μ*;
}
```

Figure 3: Constructive heuristic for LMMKP
(A variant of MO-CONS [21])

5. CONSTRUCTIVE HEURISTIC FOR LMMKP

Magazine and Oguz [21] proposed a constructive method using Lagrange multipliers. We devise a method similar to it to find $\boldsymbol{\lambda}$ for LMMKP.

First, $\boldsymbol{\lambda}$ is set to be $\mathbf{0}$. Consequently, x_i becomes 1 for each $v_i > 0$. It means that all positive-valued objects are put in the knapsack and so almost all constraints are violated. If $\boldsymbol{\lambda}$ is increased, some objects become taken out. We increase $\boldsymbol{\lambda}$ 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 λ_k.

Reconsider

$$\boldsymbol{v}^T\boldsymbol{x} - \langle \boldsymbol{\lambda}, W\boldsymbol{x} - \boldsymbol{b} \rangle \quad = \quad \sum_{i=1}^{n} x_i (v_i - \sum_{j=1}^{d} \lambda_j w_{ji}) + \langle \boldsymbol{\lambda}, \boldsymbol{b} \rangle.$$

Making $(v_i - \sum_{j=1}^{d} \lambda_j w_{ji})$ be negative by increasing λ_k let $x_i = 0$ by LMMKP. For each i such that $x_i = 1$, let α_i be the increment of λ_k to make x_i be 0. Then, $(v_i - \sum_{j=1}^{d} \lambda_j w_{ji} - \alpha_i w_{ki})$ have to be negative. That is, if we increase λ_k by α_i such that $\alpha_i > (v_i - \sum_{j=1}^{d} \lambda_j w_{ji})/w_{ki}$, the i^{th} object is taken out. So, if we just change λ_k to $\lambda_k + \min_i \alpha_i$, leave λ_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 if every constraint is satisfied.

Figure 3 shows this constructive algorithm. The number of operations to take out the object is at most n, and computing α_i for each i takes $O(d)$ time. Hence, the total time complexity becomes $O(n^2 d)$.

6. EVOLUTIONARY APPROACH

We apply evolution strategy (ES) [1] for obtaining an upper bound and a lower bound of 0/1MKP. ES has been showing good performance for the problem which deals with real-valued encoding. We use $(1 + \lambda)$-ES, in which population size is one and the parent produces λ offspring. Offspring are produced by a simple Gaussian mutation. That is, for each $i = 1, 2, \ldots, n$,

$$\sigma_i' = \sigma_i \cdot \exp(\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)),$$
$$x_i' = x_i + \sigma_i' \cdot N_i(0,1),$$

where $\tau = 1/\sqrt{2\sqrt{n}}$ and $\tau' = 1/\sqrt{2n}$. The best individual is chosen among the current population and their offspring to the next population.

6.1 Upper Bound

A simple method to find an upper bound is to compute the values $\varphi^{\boldsymbol{b}}(\boldsymbol{\lambda})$'s for randomly generated $\boldsymbol{\lambda}$'s and choose the minimum among them. We get a better upper bound by using $\boldsymbol{\lambda}$ found by ES instead of determining it randomly. We search Lagrange multipliers by letting the objective function be $\varphi^{\boldsymbol{b}}(\boldsymbol{\lambda})$ and executing ES.

6.2 Lower Bound

We search Lagrange multipliers closer to the constraints by applying Theorem 6 to ES. Standard mutation operator of ES decreases or increases the value of λ_i. We modify it to increase each λ_i only when $b_i^* > b_i$, i.e., the i^{th} constraint is not satisfied, and decrease it otherwise. Then, we can find the solution which satisfies the constraints more easily. The following formula shows the modified mutation operator.

$$\lambda_i' = \begin{cases} \lambda_i + \sigma_i' \cdot |N_i(0,1)|, & \text{if } b_i^* > b_i, \\ \lambda_i - \sigma_i' \cdot |N_i(0,1)|, & \text{otherwise.} \end{cases}$$

We use

$$\boldsymbol{v}^T \boldsymbol{x}^* - \gamma \sum_{\boldsymbol{b}_i^* > \boldsymbol{b}_i} (\boldsymbol{b}_i^* - \boldsymbol{b}_i)$$

as the objective function to maximize, which is the function obtained by subtracting the penalty from the objective function of 0/1MKP. γ is a constant which indicates the degree of penalty and it can be tuned appropriately.

7. RESULTS

7.1 Test Set and Test Environment

We test the proposed algorithms on benchmark data. Most of them are the same data used in Chu and Beasley [6]. They are composed of 120 instances with 10 constraints. Thirty instances with $1,000$ objects are newly generated by the procedure of [8].[1] All tested data have different number of objects and different tightness ratios. The tightness ratio means α such that $b_j = \alpha \sum_{i=1}^{n} w_{ji}$ for each j. The class of instances are briefly described below.

- $d.n\text{-}\alpha$: d constraints, n objects, and tightness ratio α. Each class has 10 instances.

The proposed algorithms were implemented with *gcc* compiler on a Pentium PC (1.75GHz) using Linux operating system. As the measure of performance, we used the percentage difference-ratio $100 \times |LP\_optimum - output| / output$ which was used in [6].[2] It has a value in the range $[0, 100]$. The smaller the value, the smaller the difference from the optimum. To compare our algorithms, we used the constructive heuristic and the hybrid genetic algorithm proposed by Cotta and Troya [7].

[1]The remaining 90 instances were generated in [6] with the same procedure.

[2]$LP\_optimum$ is the optimal solution of the linear programming relaxation over $\mathbb{R}$.

Table 1: Upper Bounds

Instances	LM-Random	LM-ES
10.100-0.25	98.31	0.02
10.100-0.50	98.27	0.01
10.100-0.75	98.58	0.01
10.250-0.25	99.27	0.01
10.250-0.50	99.31	0.00
10.250-0.75	99.26	0.00
10.500-0.25	99.59	0.00
10.500-0.50	99.56	0.00
10.500-0.75	99.59	0.00

Average percentage difference-ratio over 10 instances.

Table 2: Comparison of Constructive Heuristics

Instances	CT-CONS [7]	LM-CONS
10.100-0.25	22.95	10.78
10.100-0.50	11.97	7.02
10.100-0.75	5.70	3.18
10.250-0.25	18.80	10.35
10.250-0.50	7.81	5.94
10.250-0.75	4.46	3.33
10.500-0.25	15.03	9.83
10.500-0.50	6.87	5.30
10.500-0.75	3.51	3.11
10.1000-0.25	13.90	8.88
10.1000-0.50	8.36	4.84
10.1000-0.75	5.47	2.74

Average percentage difference-ratio over 10 instances.

7.2 Upper Bound

Table 1 shows the performance of our ES. Our ES is (1+30)-ES and stops after 10^4 generations. LM-Random means the best result among randomly generated 3×10^5 Lagrange multiplier vectors. LM-Random performed poorly. However, when Lagrange multipliers are optimized by ES (LM-ES), we obtained high-quality upper bounds.

The average percentage difference-ratio of the upper bounds by LM-ES decreases as the number of objects increases. In particular, the values of the instances with 500 objects were very close to zero. Thus, it provides a rational reason for using the upper bound obtained by LM-ES instead of $LP\_optimum$ for the instances with more than 500 objects. Since we do not know $LP\_optimum$'s of the newly generated instances with $1,000$ objects, we will use the upper bound by LM-ES in computing the percentage difference-ratio of the lower bound for the instances with $1,000$ objects.

7.3 Lower Bound

We compare our algorithms with traditional approaches that search the domain $\{0,1\}^n$ directly. First, we compare constructive heuristics. To compare our constructive heuristic (LM-CONS), we adopted the constructive heuristic of [7] (CT-CONS). CT-CONS was designed from a typical constructive heuristic for the simple knapsack problem with one constraint. It chooses the value $\delta_i = min_j\{b_j v_i / w_{ji}\}$ as the profit density of each object i. Table 2 shows their performance. For all classes of instances, LM-CONS outperformed CT-CONS.

Next, we compare evolutionary algorithms. To compare our evolution strategies (LM-ES and LM-ES-AP), we adopted

Table 3: Comparison of Evolutionary Algorithms

Instances	CT-GA [7]		LM-ES		LM-ES-AP†	
	Ave	CPU	Ave	CPU	Ave	CPU
10.100-0.25	**3.25**	15.2	5.05	60.8	4.11	60.8
10.100-0.50	**1.60**	16.0	2.05	60.9	1.75	60.3
10.100-0.75	**0.93**	16.5	1.28	60.4	1.13	60.4
10.250-0.25	**1.31**	71.1	2.95	137	2.03	138
10.250-0.50	**0.83**	72.4	1.66	138	0.84	138
10.250-0.75	**0.44**	72.9	0.87	137	0.62	137
10.500-0.25	3.54	250	2.48	266	**1.29**	265
10.500-0.50	1.97	248	1.01	264	**0.62**	266
10.500-0.75	1.14	254	0.71	264	**0.34**	264
10.1000-0.25	8.30	937	1.40	522	**0.79**	524
10.1000-0.50	5.24	943	0.99	525	**0.47**	525
10.1000-0.75	2.66	952	0.59	521	**0.18**	522

Average percentage difference-ratio over 10 instances.
Each run stops after 10^5 generations.
CPU seconds on Pentium 1.75GHz.
(† It did not perform better than the state-of-the-art methods proposed in [6] and [25].)

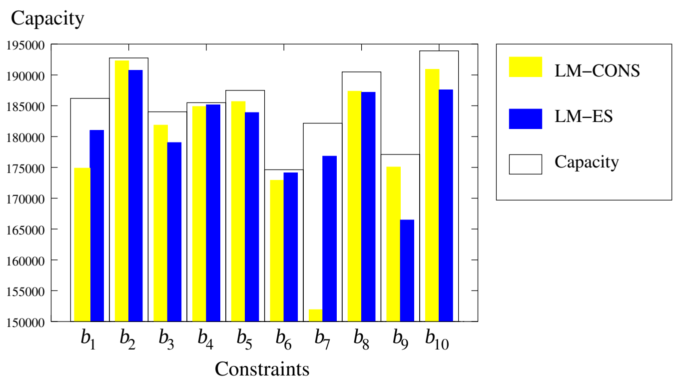

Figure 4: Constraints corresponding to Lagrange multipliers optimized by each method: an example case 10.500-0.75

the hybrid genetic algorithm of [7] (CT-GA). Our ESs are (1+10)-ES and stop after 10^5 generations. LM-ES has a fixed value 0.7 as the penalty degree γ, but LM-ES-AP adjusts γ from 1.0 down to 0.1; as the initial value, γ is 1.0, and then γ decreases by 0.001 per 100 generations. Table 3 shows their performance. The difference-ratios of our ESs were inversely proportional to n and α. LM-ES-AP dominated LM-ES for all instances. For the instance classes with small numbers of objects ($n = 100,\ n = 250$), CT-GA performed best. However, for the classes with large numbers of objects ($n = 500,\ n = 1000$), our ESs outperformed CT-GA. The results show the superiority of our ESs for instances with $d \ll n$.

LM-ES outperformed LM-CONS (see Table 2 and Table 3). We also investigated constraints corresponding to Lagrange multipliers optimized by each method. That is, for $\boldsymbol{x}^*$ which is the solution obtained by each method, we computed the corresponding capacity $b_j^* = \sum_{i=1}^{n} w_{ji} x_i^*$ for each constraint j. This value means the amount put in each knapsack. LM-CONS has a defect that some constraints are poorly satisfied. Figure 4 shows a typical example. In this example, we observed that the seventh value of LM-CONS is much lower than that of LM-ES. However, with the Lagrange multipliers optimized by ES, constraints were fit more evenly than those by LM-CONS.

8. CONCLUSIONS

In this paper, we proposed Lagrangian method for the 0/1 multiple knapsack problem. We also provided some theoretical arguments supporting our Lagrangian method. Our Lagrangian method is fast and guarantees optimality with the constraints which may be different from the original problem. However, it is not easy to find Lagrange multipliers that accord with all the constraints. We found high-quality Lagrange multipliers by combining Lagrangian method with evolution strategy. We computed upper bounds using ES from duality theorem. Also, we computed lower bounds using ES with the modified mutation operator from tendency theorem and improved the performance of ES by adjusting the penalty in objective function. We obtained a significant performance improvement over Cotta and Troya's method [7] on the instances with a large number of objects. Although our methods did not dominate the state-of-the-art methods such as [6] and [25], we believe that there is room for further improvement because we just used a simple ES. Also, we guess that the performance highly depends on the distribution of gaps of given instance. More studies about gaps are left for future study.

9. ACKNOWLEDGMENTS

We would like to thank Carlos Cotta for providing his program denoted by CT-GA in our paper. This work was supported by the Brain Korea 21 Project in 2005. This was also partly supported by grant No. (R01-2003-000-10879-0) from the Basic Research Program of the Korea Science and Engineering Foundation. The ICT at Seoul National University provided research facilities for this study.

10. REFERENCES

[1] T. Bäck, F. Hoffmeister, and H. P. Schwefel. A survey of evolution strategies. In *Proceedings of the International Conference on Genetic Algorithms*, pages 2–9, 1991.

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

[3] Y.-J. Chang and B. W. Wah. Lagrangian techniques for solving a class of zero-one integer linear programs. In *Proceedings of the Computer Software and Applications Conference*, pages 156–161, Dallas, Texas, 1995.

[4] C. Chekuri and S. Khanna. A PTAS for the multiple knapsack problem. In *Proceedings of the Symposium on Discrete Algorithms*, pages 213–222, 2000.

[5] P. C. Chu. *A genetic algorithm approach for combinatorial optimization problems*. PhD thesis, The management school, imperial college of science, London, 1997.

[6] P. C. Chu and J. E. Beasley. A genetic algorithm for the multidimensional knapsack problem. *Journal of Heuristics*, 4:63–86, 1998.

[7] C. Cotta and J. M. Troya. A hybrid genetic algorithm for the 0-1 multiple knapsack problem. In G. D.

Smith, N. C. Steele, and R. F. Albrecht, editors, *Artificial Neural Networks and Genetic Algorithms 3*, pages 251–255. Springer-Verlag, 1997.

[8] A. Freville and G. Plateau. An efficient preprocessing procedure for the multidimensional 0-1 knapsack problem. *Discrete Applied Mathematics*, 49:189–212, 1994.

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

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

[11] B. Gavish. On obtaining the 'best' multipliers for a Lagrangean relaxation for integer programming. *Computers & Operations Research*, 5:55–71, 1978.

[12] B. Gavish and H. Pirkul. Efficient algorithms for solving multiconstraint zero-one knapsack problems to optimality. *Mathematical Programming*, 31:78–105, 1985.

[13] A. M. Geoffrion. Lagrangian relaxation for integer programming. *Mathematical Programming Study*, 2:82–114, 1974.

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

[15] J. Gottlieb. *Evolutionary Algorithms for Constrained Optimization Problems*. PhD thesis, Department of Computer Science, Technical University of Clausthal, 1999.

[16] H. Kellerer and U. Pferschy. A new fully polynomial approximation scheme for the knapsack problem. In *Proceedings of the 1st International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 123–134, 1998.

[17] S. Khuri, T. Bäck, and J. Heitkötter. The zero/one multiple knapsack problem and genetic algorithms. In *Proceedings of the ACM Symposium on Applied Computing*, pages 188–193. ACM Press, 1994.

[18] E. L. Lawler. Fast approximation algorithms for knapsack problems. In *Proceedings of the 17th Annual Symposium on Foundations of Computer Science*, pages 206–213, 1977.

[19] A. Løkketangen and F. Glover. Probabilistic move selection in tabu search for zero-one mixed integer programming problems. In I. H. Osman and J. P. Kelly, editors, *Meta-Heuristics: Theory and Applications*, pages 467–487. Kluwer Academic, 1996.

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

[21] M. J. Magazine and O. Oguz. A heuristic algorithm for the multidimensional zero-one knapsack problem. *European Journal of Operational Research*, 16:319–326, 1984.

[22] S. Martello and P. Toth. *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, Ltd., 1990.

[23] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons Inc., 1988.

[24] G. Raidl. Weight-codings in a genetic algorithm for the multiconstraint knapsack problem. In *Proceedings of the Congress on Evolutionary Computation*, volume 1, pages 596–603. IEEE Press, 1999.

[25] G. R. Raidl. An improved genetic algorithm for the multiconstrained 0-1 knapsack problem. In *Proceedings of the IEEE Conference on Evolutionary Computation*, pages 207–211, 1998.

[26] C. R. Reeves, editor. *Modern Heuristic Techniques for Combinatorial Problems*. Halsted Press, 1993.

[27] S. Sahni. Approximate algorithms for the 0/1 knapsack problem. *Journal of the ACM*, 22:115–124, 1975.

[28] D. Schuurmans, F. Southey, and R. C. Holte. The exponentiated subgradient algorithm for heuristic boolean programming. In *Proceedings of the International Joint Conferences on Artificial Intelligence*, pages 334–341, 2001.

[29] Y. Shang and B. W. Wah. Discrete Lagrangian-based search for solving MAX-SAT problems. In *Proceedings of the International Joint Conferences on Artificial Intelligence*, pages 378–383, 1997.

[30] J. Thiel and S. Voss. Some experiences on solving multiconstraint zero-one knapsack problems with genetic algorithms. *INFOR*, 32:226–242, 1994.

[31] M. Vasquez and J.-K. Hao. A hybrid approach for the 0-1 multidimensional knapsack problem. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pages 328–333. Morgan Kaufmann, 2001.

[32] B. W. Wah and Y. Shang. A discrete Lagrangian-based global-search method for solving satisfiability problems. In D.-Z. Du, J. Gu, and P. Pardalos, editors, *Satisfiability Problem: Theory and Applications*, pages 365–392. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, 1997.

[33] B. W. Wah and Z. Wu. The theory of discrete Lagrange multipliers for nonlinear discrete optimization. In *Principles and Practice of Constraint Programming*, pages 28–42, 1999.

[34] Z. Wu. The discrete Lagrangian theory and its application to solve nonlinear discrete constrained optimization problems. MSc thesis, Department of Computer Science, University of Illinois, 1998.