数理解析研究所講究録 1477 巻 2006 年 92-101



# 多目的ナップサック問題のマックスミン最適化

防衛大学校情報工学科　谷口 史晃（Fumiaki Taniguchi）
防衛大学校情報工学科　片岡 靖詞（Seiji Kataoka）
防衛大学校情報工学科　山田 武夫（Takeo Yamada）
Department of Computer Science,
The National Defense Academy

**要旨**：*The max-min knapsack problem* is an extension of the classical knapsack problem. Each item has a weight and multiple profits depending on several scenarios. The problem is about maximizing the minimum total-profit among the scenarios under the knapsack constraint. We apply Lagrangian relaxation method for obtaining upper bounds and lower bounds. We also apply the pegging test for the reduction of problem size. For more reduction, we propose *the virtual pegging test*, in which we estimate a tighter lower bound, use it in the pegging test, and verify the correctness. Computational experiments show that our algorithm solves the problem more effectively than the past researches.
*Keywords* :**knapsack problem, robust optimization, pegging test**

## 1　はじめに

### 1.1　定式化

商品 $j(\in N=\{1,\ldots,n\})$ は，シナリオ $s(\in S)$ に基づく利得 $p_j^s$ と重量 $w_j$ を持っている．この商品を重量制限 $c$ のナップサックに詰め込み，各シナリオに対する総利得のうち最小のものを最大にする問題を**マックスミンナップサック問題**（Max-miN Knapsack problem : MNK）[13] と呼ぶ．MNK は，以下のように定式化できる．

**MNK:**

$$\begin{aligned} \text{maximize} \quad & z := \min_{s\in S} \sum_{j=1}^{n} p_j^s x_j \\ \text{subject to} \quad & \sum_{j=1}^{n} w_j x_j \le c \\ & x_j \in \{0,1\}, \quad j=1,\ldots,n \end{aligned}$$

### 1.2　研究の概要

本研究では，MNK に対し**ラグランジュ緩和**による上界値を用いた**分枝限定法**のアルゴリズムを提案する．また，問題の規模を縮小するために，仮想的に見積もった下界値を与えて行う**仮想釘付けテスト**を提案する．これらを用いて MNK に対する厳密解法を構築し，同種の問題を扱った Iida[4] の実験結果と比較し評価する．

本論文を通して以下を仮定するが，これらにより一般性を失うことはない．

$\mathbf{A}_1$. 重量制限 $c$，利得 $p_j^s$，重量 $w_j$ はすべて正整数．

$\mathbf{A}_2$. $\sum_{j=1}^{n} w_j > c$ かつ $w_j \leq c\ (j = 1, \ldots, n)$.

0-1 決定変数ベクトル $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ を**解**，解の中ですべての制約式を満たすものを**実行可能解**と呼ぶ．実行可能解 $\boldsymbol{x}$ に対し各シナリオ毎の評価値を $z^s(:= \sum_{j=1}^{n} p_j^s x_j)$ とし，$z^s$ のうち最小のものを目的関数値 $z(:= \min_{s \in S} z^s)$ と呼ぶ．また，目的関数値を最大にする解を**最適解** $\boldsymbol{x}^\star$ と呼び，最適解における目的関数値を**最適値** $z^\star$ と呼ぶ．

# 2 上下界値と分枝限定法

## 2.1 ラグランジュ緩和

MNK は変数 $v$ を用いて以下のように変換できる．

$$
\begin{aligned}
\text{maximize} \quad & v \\
\text{subject to} \quad & \sum_{j=1}^{n} p_j^s x_j \geq v, \quad \forall s \in S \\
& \sum_{j=1}^{n} w_j x_j \leq c \\
& x_j \in \{0, 1\}, \quad j = 1, \ldots, n
\end{aligned}
$$

1 本目の制約式に，非負の乗数 $\lambda^s \geq 0\ (s \in S)$ を掛けて目的関数に取り込みラグランジュ緩和をする．さらに，$\sum_{s \in S} \lambda^s = 1$ という制限をつけ，0-1 条件を連続緩和した問題は次のようになる．ただし，$\bar{p}_j(\boldsymbol{\lambda}) := \sum_{s \in S} \lambda^s p_j^s$ とする．

**LMNK($\boldsymbol{\lambda}$):**

$$
\begin{aligned}
\text{maximize} \quad & z(\boldsymbol{\lambda}) := \sum_{j=1}^{n} \bar{p}_j(\boldsymbol{\lambda}) x_j \\
\text{subject to} \quad & \sum_{j=1}^{n} w_j x_j \leq c \\
& 0 \leq x_j \leq 1, \quad j = 1, \ldots, n
\end{aligned}
$$

本論文では，LMNK($\boldsymbol{\lambda}$) を MNK のラグランジュ緩和問題と呼ぶことにする．LMNK($\boldsymbol{\lambda}$) の最適目的関数値 $\bar{z}(\boldsymbol{\lambda})$ とし，そのときの解を**ラグランジュ解** $\bar{\boldsymbol{x}}(\boldsymbol{\lambda}) = (\bar{x}_j)$ とする．

LMNK($\boldsymbol{\lambda}$) には，次のような特質がある [2].

- LMNK($\boldsymbol{\lambda}$) は連続型ナップサック問題となり，その解は高速かつ容易に求められる．
- $\bar{z}(\boldsymbol{\lambda})$ は MNK の**上界値**を与える．すなわち $z^\star \leq \bar{z}(\boldsymbol{\lambda})$ である．
- $\bar{z}(\boldsymbol{\lambda})$ は $\boldsymbol{\lambda}$ について**区分的に線形な凸関数**である．
- $\boldsymbol{\lambda}$ において，$\bar{z}(\boldsymbol{\lambda})$ が微分可能であれば，$\frac{\partial \bar{z}(\boldsymbol{\lambda})}{\partial \lambda^s} = \sum_{j=1}^{n} p_j^s \bar{x}_j$ である．

精度の良い上界値を得るためには，$\bar{z}(\boldsymbol{\lambda})$ を最小にする最適なラグランジュ乗数 $\boldsymbol{\lambda}^\dagger$ を見つけなければならない．このとき，$\bar{z}(\boldsymbol{\lambda})$ は $\boldsymbol{\lambda}$ について区分的に線形な凸関数であることが分かっているので，通常は**劣勾配法**によって $\boldsymbol{\lambda}^\dagger$ を求めることができる．しかしながら，本研究で扱うラグランジュ緩和問題にはラグランジュ乗数に $\sum_{s \in S} \lambda^s = 1$ という制限がついているので，多少の工夫が必要である（2.2 節）．

このときの目的関数値 $\bar{z}(\boldsymbol{\lambda}^\dagger)$ を MNK の**上界値** $\bar{z}$ とする．また，**下界値** $\underline{z}$ は，**LMNK($\boldsymbol{\lambda}^\dagger$)** の最適解から，分数値を持つ変数を強制的に 0 にすると，MNK の実行可能解になることで得られる．

## 2.2 単体上での劣勾配法

劣勾配法は，微分不可能な関数の最適化手法のひとつとして知られている [1, 3]．$\bar{z}(\boldsymbol{\lambda})$ も $\boldsymbol{\lambda}$ について区分的に線形な凸関数であるため，通常ならば劣勾配法を用いて，$\bar{z}(\boldsymbol{\lambda})$ が最小となる $\boldsymbol{\lambda}^{\dagger}$ を求めることができる．しかしながら，本研究ではラグランジュ乗数に $\sum_{s\in S}\lambda^s = 1, \lambda^s \geq 0\ (s \in S)$（単体）という制限があることに注意しなければならない．このため，以下のようなアルゴリズム SSUB を提案する．

**アルゴリズム：SSUB**

**Step.1** $\boldsymbol{\lambda}$ の初期値を重心 $\boldsymbol{\lambda} := (1/|S|, \ldots, 1/|S|)$ または，任意の要素のみ 1 にした $\boldsymbol{\lambda} := (0, \cdots, 0, 1, 0, \cdots, 0)$ に選定し，任意の精度の終了条件 $\epsilon(>0)$ を設定する．

**Step.2** LMNK($\boldsymbol{\lambda}$) を解く．

**Step.3** 劣勾配 $\boldsymbol{g}$ を計算し，探索方向 $\boldsymbol{d}$ を求める．

**Step.4** $\boldsymbol{d} \cong \boldsymbol{0}$ ならば終了．

**Step.5** $\bar{z}(\boldsymbol{\lambda} + \alpha\boldsymbol{d})$ が最小になるステップ幅 $\alpha(\geq 0)$ を 2 分探索で求める．

**Step.6** $\alpha < \epsilon$ ならば終了し，そうでなければ $\boldsymbol{\lambda} := \boldsymbol{\lambda} + \alpha\boldsymbol{d}$ として **Step.2** へ戻る．

以上のアルゴリズム SSUB を数回反復し，最も良い（小さい）$\bar{z}(\boldsymbol{\lambda})$ を上界値 $\bar{z}$ とし，そのときの $\boldsymbol{\lambda}$ を $\boldsymbol{\lambda}^{\dagger}$ とする．また，LMNK($\boldsymbol{\lambda}$) を解く際の副産物として下界値（実行可能解）も同時に得ることができる．

### 2.2.1 探索方向 $\boldsymbol{d}$ の決定

**Step.3** の探索方向 $\boldsymbol{d}$ の決定には少し注意が必要である．ラグランジュ緩和において $\boldsymbol{\lambda}$ につく制限は $\lambda^s \geq 0\ (s \in S)$ が一般的であるが，本研究においては，さらに $\sum_{s\in S}\lambda^s = 1$ という制約がついているため，$\boldsymbol{\lambda}$ のとるべき範囲は図 1 の灰色部分のような $|S|-1$ 次元の単体上となる（図 1 は $|S|=3$ の例）．そして，**Step.5** の 2 分探索において $\alpha$ の動ける範囲は，新しい $\boldsymbol{\lambda}$ が単体上に存在できる範囲でなければならない．

通常，$-\boldsymbol{g}$（$\bar{z}(\boldsymbol{\lambda})$ を減少させる方向）は，図 1 のように単体上に乗っていないベクトルなので，求めるべき探索方向 $\boldsymbol{d}$ は

$$d^s := -g^s + \frac{\sum_{s\in S} g^s}{|S|} \tag{1}$$

のように単体上に射影する必要がある．

また，現在の $\boldsymbol{\lambda}$ が単体の**内点にないとき**は，探索方向 $\boldsymbol{d}$ を

$$s_+ := \arg\ \min\{g^s |\ s \in S\}$$
$$s_- := \arg\ \max\{g^s | \lambda^s > 0,\ s \in S\}$$
$$\boldsymbol{d} := \begin{cases} 1 & s = s_+ \text{のとき} \\ -1 & s = s_- \text{のとき} \\ 0 & \text{それ以外} \end{cases} \tag{2}$$

と定め，$\boldsymbol{\lambda}$ を更新しても単体から飛び出さないようにすればよい．ただし，$g^{s_+} = g^{s_-}$ のとき減少方向はなく $\boldsymbol{d} := \boldsymbol{0}$ として **Step.4** において終了する．このときの $\boldsymbol{\lambda}$ を最適な $\boldsymbol{\lambda}^{\dagger}$ とする．

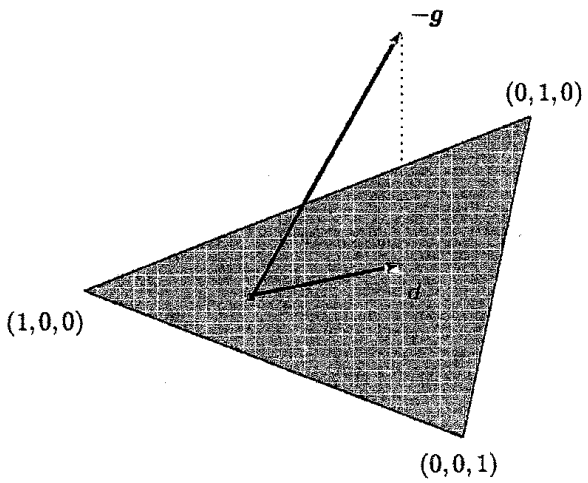


図 1: 劣勾配 $\boldsymbol{g}$ と $|S|-1$ 次元の単体上の探索方向 $\boldsymbol{d}$

### 2.3 部分問題

分枝限定法は，整数計画問題の厳密解法の 1 つとして知られている [5, 9]．分枝限定法では，0-1 問題の場合，いくつかの変数を 1 や 0 に固定した**部分問題**を考える．商品全体の添字集合を $N=\{1,\ldots,n\}$ とし，1 に固定された変数の添字集合を $F_1$，0 に固定された変数の添字集合を $F_0$，固定されていない変数の添字集合を $U(=N\setminus(F_1\cup F_0))$ とする．また，$F_1\cap F_0=\emptyset$ である．MNK において，$F_1,F_0$ に属する変数が固定された部分問題 $\mathrm{MNK}(F_1,F_0)$ は，式 (3) のように記述できる．

**MNK$(F_1,F_0)$:**

$$\begin{aligned}
\text{maximize}\quad & z(F_1,F_0):=\min_{s\in S}\sum_{j=1}^{n}p_j^s x_j \\
\text{subject to}\quad & \sum_{j=1}^{n}w_j x_j\le c \\
& x_j=1,\quad \forall j\in F_1 \\
& x_j=0,\quad \forall j\in F_0 \\
& x_j\in\{0,1\},\quad \forall j\in U
\end{aligned}\tag{3}$$

$\mathrm{MNK}(F_1,F_0)$ の最適目的関数値を $z^*(F_1,F_0)$ と記す．実行不可能 ($\sum_{j\in F_1}w_j>c$) であった場合には，$z^*(F_1,F_0)=-\infty$ と定義する．最適なラグランジュ乗数 $\boldsymbol{\lambda}^\dagger$ を用いて式 (3) をラグランジュ緩和し，0-1 条件も線形緩和すると，式 (4) を得る．

**LMNK$(\boldsymbol{\lambda}^\dagger:F_1,F_0)$:**

$$\begin{aligned}
\text{maximize}\quad & z(\boldsymbol{\lambda}^\dagger:F_1,F_0):=\sum_{j=1}^{n}\bar{p}_j(\boldsymbol{\lambda}^\dagger)x_j \\
\text{subject to}\quad & \sum_{j=1}^{n}w_j x_j\le c \\
& x_j=1,\quad \forall j\in F_1 \\
& x_j=0,\quad \forall j\in F_0 \\
& 0\le x_j\le 1,\quad \forall j\in U
\end{aligned}\tag{4}$$

緩和問題LMNK($\lambda^\dagger : F_1, F_0$)の最適目的関数値を$\bar{z}(\lambda^\dagger : F_1, F_0)$と記す．実行不可能の場合，$\bar{z}(\lambda^\dagger : F_1, F_0) = -\infty$とする．

MNK($F_1, F_0$)とLMNK($\lambda^\dagger : F_1, F_0$)の関係は，MNKとLMNK($\lambda^\dagger$)とほとんど同じである．それぞれの最適値においては，

$$z^\star(F_1, F_0) \leq \bar{z}(\lambda^\dagger : F_1, F_0) \tag{5}$$

が成り立ち，$\bar{z}(\lambda^\dagger : F_1, F_0)$は部分問題MNK($F_1, F_0$)の上界値を与える．LMNK($\lambda^\dagger : F_1, F_0$)は連続型ナップサック問題で簡単に解くことができる．LMNK($\lambda^\dagger : F_1, F_0$)の最適解で分数値を持つ変数を0にした解は，MNK($F_1, F_0$)の実行可能解を与え，その値を$\underline{z}(\lambda^\dagger : F_1, F_0)$と表記する．また，MNK($F_1, F_0$)の実行可能解は元問題MNKの実行可能解でもある．

これ以降，部分問題の緩和問題としてLMNK($\lambda^\dagger : F_1, F_0$)を解くことになるので$\mathbf{A}_3$を仮定する．

$\mathbf{A}_3$. 商品は，相対利得$\bar{p}_j(\lambda^\dagger)/w_j$の降順に並んでいる．

部分問題毎に$\lambda^\dagger$を求め直すことも可能であるが，本研究では，$\lambda^\dagger$をすべての部分問題に共通して上界値$\bar{z}(\lambda^\dagger : F_1, F_0)$の計算に用いる．上界値の精度は若干落ちるが，各部分問題においての上界値は高速に計算できる．

### 2.4 分枝限定法のアルゴリズム

**アルゴリズム：BBMNK**

**Step.0** アルゴリズムSSUBを用いて$\lambda^\dagger$と下界値$\underline{z}$を求め，スタックへMNK($\emptyset, \emptyset$)を積む．

**Step.1** スタックが空であれば，現在の$\underline{z}$を最適値$z^\star$，最良解$\boldsymbol{x}^\star$を最適解と出力して終了．

**Step.2** スタックが空でなければ先頭にある部分問題を取り出し，その緩和問題LMNK($\lambda^\dagger : F_1, F_0$)を解き，部分問題の上界値$\bar{z}(\lambda^\dagger : F_1, F_0)$と下界値$\underline{z}(\lambda^\dagger : F_1, F_0)$および対応する実行可能解$\underline{\boldsymbol{x}}$を求める．

**Step.3** $\underline{z} < \underline{z}(\lambda^\dagger : F_1, F_0)$ならば，下界値と最良解をそれぞれ$\underline{z} := \underline{z}(\lambda^\dagger : F_1, F_0)$および$\boldsymbol{x}^\star := \underline{\boldsymbol{x}}$に更新する．

**Step.4** LMNK($\lambda^\dagger : F_1, F_0$)が整数最適解ならば，MNK($F_1, F_0$)を見切り，**Step.1**へ戻る．

**Step.5** $\left\lfloor \bar{z}(\lambda^\dagger : F_1, F_0) \right\rfloor \leq \underline{z}$ならば，MNK($F_1, F_0$)を見切り，**Step.1**へ戻る．

**Step.6** $U$の中から適当な変数添字$\hat{j}$を取り出し，MNK($F_1, F_0 \cup \{\hat{j}\}$)とMNK($F_1 \cup \{\hat{j}\}, F_0$)の2つの部分問題を生成しスタックへ積み，**Step.1**へ戻る．

## 3 釘付けテスト

### 3.1 通常の釘付けテスト

通常の**釘付けテスト**[5, 9]は，上界値と下界値の差を利用したものであり，各変数$x_k$に対して

$$\bar{z}(\lambda^\dagger : \emptyset, \{k\}) < \underline{z} \quad \rightarrow \quad x_k^\star = 1 \quad \text{に固定}$$
$$\bar{z}(\lambda^\dagger : \{k\}, \emptyset) < \underline{z} \quad \rightarrow \quad x_k^\star = 0 \quad \text{に固定}$$

とすることで，問題を縮小することができる．

実際に釘付けテストを行う際には，各変数に対して，2通りのラグランジュ緩和問題 LMNK($\lambda^\dagger : \emptyset, \{k\}$) と LMNK($\lambda^\dagger : \{k\}, \emptyset$) を毎回解かなければならない．この手間を省くために，分枝限定法に入る前の LMNK($\lambda^\dagger : \emptyset, \emptyset$) を1度解くだけで，効率よく釘付けを判定する手法を提案する．

仮定 $\mathbf{A}_3$ より横軸を累積重量 $W$，縦軸を利得の和 $P$ とすると図2のように利得の和は**区分的線形**で**単調増加な凹関数**となっている．この折れ線と $W = c$ の交点は上界値となる．また，そのときの交点に相当する商品 $r$ を**臨界商品**と呼ぶ．

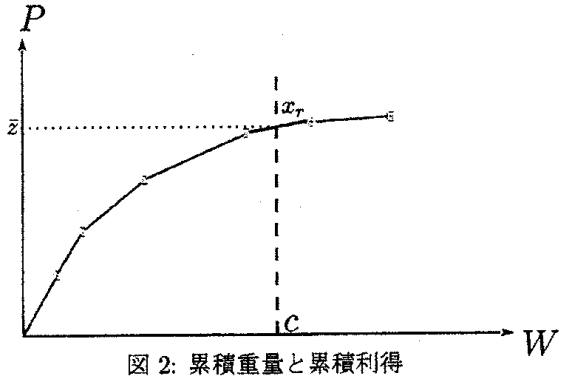


図 2: 累積重量と累積利得

釘付けテストは，$k < r$ と $r \leq k$ の2つの場合で考える．$k < r$ の場合，$x_k = 0$ に固定した部分問題の上界値は，商品 $k$ の部分が無くなり，それ以降の折れ線が商品 $k$ の分だけ左下に $(-w_k, -\bar{p}_k(\lambda^\dagger))$ 平行移動した折れ線と，$W = c$ の交点から得られる．

実際には，さらに計算を簡単にするため，$\bar{z}(\lambda^\dagger : \emptyset, \{k\})$ の上限（**近似的な上界値**）を用いる．この値は臨界商品 $r$ の直線（傾き $\bar{p}_r(\lambda^\dagger)/w_r$）が平行移動後に $W = c$ と交わる点を

$$\bar{z} - \left( \bar{p}_k(\lambda^\dagger) - w_k \frac{\bar{p}_r(\lambda^\dagger)}{w_r} \right)$$

として求めればよい．さらに，しきい値 $\theta_k$ を

$$\theta_k := \bar{p}_k(\lambda^\dagger) - w_k \frac{\bar{p}_r(\lambda^\dagger)}{w_r} \tag{6}$$

とおけば，釘付けテストの判別は，

$$\bar{z} - \underline{z} < \theta_k$$

のとき，$x_k^\star = 1$ に固定できる．

次に，$r \leq k$ の場合 $x_k = 1$ に固定した部分問題の上界値は，商品 $k$ を先頭に挿入し，1から $k-1$ までの折れ線が右上に $(w_k, \bar{p}_k(\lambda^\dagger))$ 平行移動したものと，$W = c$ の交点から得られる．この場合も同様に，$\bar{z}(\lambda^\dagger : \{k\}, \emptyset)$ の上限を

$$\bar{z} + \left( \bar{p}_k(\lambda^\dagger) - w_k \frac{\bar{p}_r(\lambda^\dagger)}{w_r} \right)$$

として求める．式 (6) のしきい値 $\theta_k$ を用いれば，

$$\bar{z} - \underline{z} < -\theta_k \tag{7}$$

のときに，$x_k^\star = 0$ に固定できる．以上から通常の釘付けテストはアルゴリズム NPEG のようになる．

**アルゴリズム：NPEG**

**Step.1** 臨界商品 $r$ を求める．

**Step.2** すべての商品 $k$ に対し，$\theta_k = \bar{p}_k(\boldsymbol{\lambda}^\dagger) - \bar{p}_r(\boldsymbol{\lambda}^\dagger)/w_r$ を求め，$\bar{z} - \underline{z} < \theta_k$ ならば，$F_1 \cup \{k\}$，$\bar{z} - \underline{z} < -\theta_k$ ならば，$F_0 \cup \{k\}$ とする．

**Step.3** $U := N \setminus (F_1 \cup F_0)$ として終了．

この後，釘付けテストによって得られた MNK$(F_1, F_0)$ にアルゴリズム BBMNK を適用すればよい．

### 3.2 仮想釘付けテスト

**仮想釘付けテスト** [14] は，下界値の代わりに**仮想下界値** $\hat{z}$ $(\underline{z} \leq \hat{z} \leq \bar{z})$ を用いて，釘付けテストを行うものである．このとき 1(0) に固定される変数の添字集合を $\hat{F}_1(\hat{F}_0)$ とすると，$\hat{z}$ の適否は定理 1 により検証できる．

**定理 1** $\hat{z} \leq z^*(\hat{F}_1, \hat{F}_0)$ ならば，$z^*(\hat{F}_1, \hat{F}_0)$ は MNK の最適値 $z^*$ である． ■

もし，$\hat{z} > z^*(\hat{F}_1, \hat{F}_0)$ であれば，再度 $\hat{z}$ を小さめに調整して $\hat{z} \leq z^*(\hat{F}_1, \hat{F}_0)$ を満たすまで釘付けテストおよび縮小問題を解くことを繰り返せばよい．

仮想下界値の決定法として，Lueker[8] が，通常のナップサック問題 $(p_j^1 = p_j^s,\ \forall s \in S)$ において，利得と重量が $[0, 1]$ の連続一様乱数にしたがうときに示した定理 2 を利用する．

**定理 2** 上界値と最適値の差 $(\bar{z}(\boldsymbol{\lambda}) - z^*)$ の期待値は，$O(\log^2 n/n)$ に収まる． ■

定理 2 より $\bar{z} - z^*$ の概型を予想して期待できる仮想下界値を決めることができる．MNK は通常のナップサック問題とは前提条件が若干異なるが，実験的に $|S|$ が小さく，$n$ が十分大きいときに定理 2 はよく当てはまることが分かった．ゆえに，本研究では仮想下界値 $\hat{z}$ を式 (8) にしたがって定めることにする．

$$\hat{z} := \bar{z} - \alpha \frac{\log^2 n}{n} \tag{8}$$

また，定数 $\alpha$ は，利得の乱数発生区間の最大値とする．

## 4 数値実験結果と考察

### 4.1 問題の設定

過去の Yu[13] や Iida[4] の研究と同じ設定とした．

- 商品数 $n$ : 60, 200～1000, 2000～10000
- 重量 $w_j$ : [1, 100] の一様乱数
- 重量制限 $c$ : $\sum_{j=1}^{n} w_j/m, \quad m = 2, 3, 4$
- 利得 $p_j^s$ : $[\hat{p}_j(1-\delta), \hat{p}_j(1+\delta)]$ の一様乱数
  - ベース値 $\hat{p}_j$ : $[1, 100]$ の一様乱数
  - 偏差パラメータ $\delta = 0.3,\ 0.6,\ 0.9$（強相関，中相関，弱相関）

一様乱数の発生については，Mersenne Twister（mt19937.c）[10] を使用した．利得 $p_j^s$ の発生は，偏差パラメータ $\delta$ が小さくなるにつれて，各商品ごとのシナリオ間の利得のばらつきが小さくなり相関が強くなることを意味している．

実験環境は，一部を IBM RS/6000 SP44 Model 270（CPU : POWER3-II，375MHz）上で行い，残りを DELL Dimension 8400（CPU : Pentium 4, 3.4GHz）で行った．すべての実験において，1800 秒で計算打ち切りとした．

## 4.2 過去の研究との比較

表 1 は，$n = 60$ のときの通常の釘付けテストを使用した MNK の厳密解法の CPU 時間（秒）を Iida の実験結果 [4] と比較している (CPU 性能差約 20 倍 [11])．各数値は 100 問の平均である．

表 1: Iida との CPU 時間の比較

| 相関 ($\delta$) | | 強相関 (0.3) | | 中相関 (0.6) | | 弱相関 (0.9) | |
|---|---|---|---|---|---|---|---|
| $\|S\|$ | $m$ | Iida | 本研究 | Iida | 本研究 | Iida | 本研究 |
| 10 | 2 | 3.0 | 0.007 | 4.9 | 0.017 | 9.1 | 0.040 |
| | 3 | 3.6 | 0.010 | 6.8 | 0.025 | 10.7 | 0.092 |
| | 4 | 3.0 | 0.009 | 5.7 | 0.027 | 12.4 | 0.066 |
| 20 | 2 | 6.0 | 0.014 | 9.3 | 0.036 | 20.0 | 0.115 |
| | 3 | 7.3 | 0.019 | 13.2 | 0.064 | 26.0 | 0.254 |
| | 4 | 6.3 | 0.022 | 8.6 | 0.061 | 21.4 | 0.230 |
| 30 | 2 | 8.1 | 0.025 | 13.8 | 0.062 | 36.9 | 0.342 |
| | 3 | 9.6 | 0.035 | 20.6 | 0.097 | 48.0 | 0.834 |
| | 4 | 7.9 | 0.033 | 15.8 | 0.094 | 38.9 | 0.849 |

CPU 性能差：約 20 倍

表 1 から，シナリオ数が増えるほど，またシナリオ間の相関が弱まるほど解き難くなることが分かる．さらに，強相関の場合（$\delta = 0.3$）で，Iida と本研究の計算時間の違いは約 200〜500 倍ほどあり，20 倍の CPU 性能差を考慮しても本研究の提案手法が効率的に MNK を解いていることが分かる．同様に，中相関（$\delta = 0.6$）においては約 150〜200 倍，弱相関（$\delta = 0.9$）においても，約 50〜150 倍となっている．シナリオ間の相関が弱まるにつれて，計算時間の差が縮まるのは，Iida が部分問題ごとに代理制約乗数（ラグランジュ乗数と読み替えてもよい）を決定し直しているのに対し，本研究の手法が初めに決定した $\boldsymbol{\lambda}^\dagger$ のみを使用して部分問題の上界値を求めているため，相関が弱まるにつれて部分問題ごとの上界値が弱くなり，見切り（限定）され難くなっているためと予想される．これは分枝限定ノードがほかの手法に比べ非常に多いことに表れている．しかしながら，部分問題の上界値を求めること自体は極めて高速にできるため，総合的に優れた実効効率を出していると考える．

## 4.3 仮想釘付けテストの効果

図 3 は，$\delta = 0.6, m = 2, |S| = 2$ において，規模の大きな問題を通常の釘付けテストを使用したときと，仮想釘付けテストを使用したときの CPU 時間（秒）を比較したものある．数値は，DELL Dimension 8400 上で行った 100 問の平均である．図 4 は，図 3 と同じ問題の釘付けテストによる縮小効果を示したものである．ただし，縮小率 (%) = 元問題の変数の数/縮小問題の変数の数 とする．

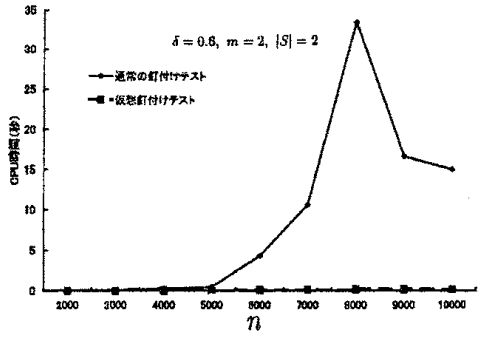


図 3: シナリオ数が 2 における CPU 時間の比較

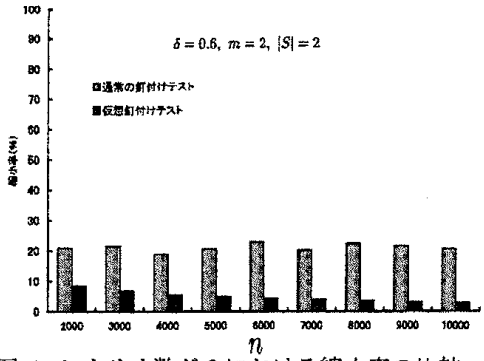


図 4: シナリオ数が 2 における縮小率の比較

図 3 より商品数が多くなると，仮想釘付けテストにより CPU 時間が劇的に改善できることが分かる．図 4 より縮小率は，通常の釘付けテストでは商品数に関係なくほぼ一定で，仮想釘付けテストでは商品数の増加につれてその効果が強くなっていることが分かる．これは，仮想下界値が商品数の増加につれて漸近的に小さくなるためと考える．

図 5 は，$\delta = 0.6, m = 2$ において，通常の釘付けテストを使用したときと，仮想釘付けテストを使用したときの CPU 時間（秒）をシナリオ数が 10，20，30 ごとに比較したものである．また，数値は，RS/6000 上で行った 10 問の中，両方とも解けた問題の平均である．図 6 は，図 5 と同じ問題の $\bar{z} - z^*$ と上界値と仮想下界値の差 $\bar{z} - \hat{z}$ の関係を示したものである．数値は 10 問中解けた問題の平均である．

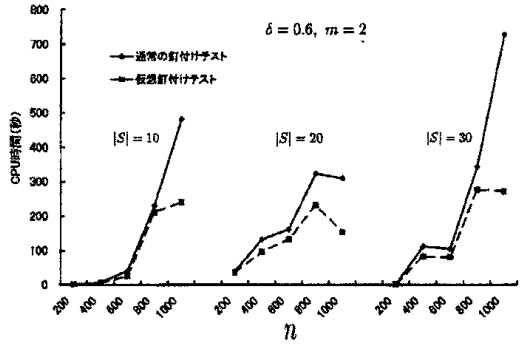


図 5: 中相関での CPU 時間の比較

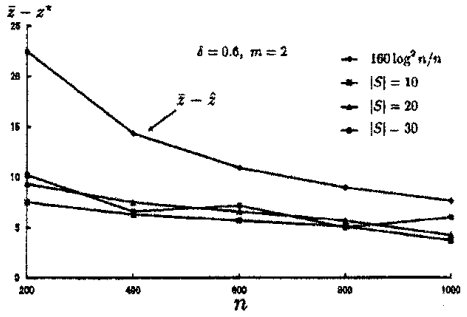


図 6: 中相関での $\bar{z} - \hat{z}$ と $\bar{z} - z^*$ の関係

シナリオ数を 2 と限定し，商品数を多くした場合では，仮想釘付けテストは効果的に作用している．しかしながら，シナリオ数が増えたときは，若干の高速化（うまく行っているもので約 2 倍程度）が図れる程度であった．理由として，商品数が 1000 程度では $\bar{z} - \alpha \log^2 n/n$ で求めた仮想下界値 $\hat{z}$ が，元々の下界値とそれほど変わらず期待される縮小効果が得られなかったためと考える．

仮想下界値の決定法の検証については，図 6 から漸近的に定理 2 にしたがうように感じ取れる．しかしながら，実験を行っている際に単純に $\alpha$ を利得の乱数発生区間の最大値とした本研究の仮想下界値の決定法のままでは，シナリオ数の増加やシナリオの相関が弱まると仮想釘付けテストをやり直すことが，全体の 1〜2 割の頻度で起こるようになる．したがって，再び解きなおす時間のロスを考えるとシナリオ数が多くなったとき，仮想下界値の決定を式 (8) のままでは行うのでは不十分であると言える．

## 5 おわりに

ラグランジュ緩和と釘付けテスト，分枝限定法を併用してMNKを効率的に解くことが可能になった．特に，シナリオが小さく，商品数が十分大きいときには，仮想釘付けテストによりさらに高速化が期待できる．

しかしながら，高速化の反面，シナリオ数の増加や相関が弱くなるにつれて仮想釘付けテストの失敗する割合が高くなり，再計算のロスを考えると本研究の仮想下界値の決定法ではまだ不十分であると言える．

今後の課題として，シナリオ数や相関も考慮した仮想下界値の決定法を考案したい．

## 参考文献


[1] S. Boyd, L. Xiao and A. Mutapcic: *Subgradient Methods* (Stanford University, 2003). http://www.stanford.edu/class/ee392o/subgrad_method.pdf

[2] M. Fisher: The Lagrangian relaxation method for solving integer programming problems. *Management Science*, **50**-12(2004)1861-1871.

[3] 茨木 俊秀, 福島 雅夫: 最適化の手法 (共立出版株式会社, 1993).

[4] H. Iida: On solving the max-min 0-1 knapsack problem. 北陸先端科学技術大学院大学, (1997)IS-RR-97-0025F.

[5] H. Kellerer, U. Pferschy and D. Pisinger: *Knapsack Problems* (Springer, 2004).

[6] 今野 浩, 鈴木 久敏 (編): 整数計画法と組合せ最適化 (日科技連, 1982).

[7] 今野 浩: 線形計画法 (日科技連, 1987).

[8] G. Lueker: On the average difference between the solutions to linear and integer knapsack problems. *Applied Probability—Computer Science, The Interface* I, (1982)489-504.

[9] S. Martello and P. Toth: *Knapsack Problrems* (John Wiley & Sons, 1990).

[10] M. Matsumoto and T. Nishimura: Mersenne Twister—A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transaction on Modeling and Computer Simulation*, **8**-1(1998) 3-30.

[11] Standard Performance Evaluation Corporation. http://www.spec.org/

[12] L.A. Wolsey: *Integer Programming* (John Wiley & Sons, 1998).

[13] G. Yu: On the max-min 0-1 knapsack problem with robust optimization applications. *Operations Research*, **44**(1996)407-415.

[14] 柳 秉俊, 山田 武夫: 序制約付きナップサック問題への仮想釘付けアプローチ. 日本オペレーションズ・リサーチ学会秋季研究発表会, (2004)2-B-6.