数理解析研究所講究録
947 巻 1996 年 162-171



# ナップザック問題の確率アルゴリズムの解析

福田 和真　萩原 斉　中森 眞理雄
東京農工大学 工学部 電子情報工学科 情報工学大講座

ナップザック問題の1つである部分和問題を解く分枝限定アルゴリズムをランダム化して，その計算の手間を解析する．

一般に，0–1 変数をもつ問題に対する分枝限定アルゴリズムは，分枝操作において変数を次々と選んでは，それらの値が 0 の場合と 1 の場合に分けて問題をより小さな部分問題に分割する．本論文では，変数を選ぶ部分をランダム化したアルゴリズムを考察する．

アルゴリズムの手間の評価は生成される部分問題の数について注目する．また，部分問題の限定操作が最小数しか行われない確率についても考察する．

## 1 はじめに

変数が整数値など離散的な値をとるときに，与えられた制約条件の下で与えられた関数の値を最大 (あるいは最小) とする変数の値を求める問題は**離散計画問題** (discrete optimization problem) と呼ばれ，数理計画問題の中でも最も難しい問題とされている．例えば，変数の値が整数値である 2 次計画問題に対する一般的なアルゴリズムは存在しないことが知られている[1]．特に，変数の値が 0 または 1 に限られ 1 つの線形不等式の下で線形関数を最大あるいは最小にする問題は**ナップザック問題** (knapsack problem) と呼ばれている．ナップザック問題は **NP 完全** (NP complete) あるいは **NP 困難** (NP hard) と呼ばれるクラスに属し (どちらのクラスに属するかは問題の定義のしかたによって異なる)，手間が変数の個数に対して指数関数的なアルゴリズムしか知られていない．ナップザック問題の特殊な例として，**部分和問題** (subset–sum problem) がある．これは，いくつかの棒が与えられているとき，長さの和が指定された値以下であってしかも最大である棒の組合せを求める問題である．部分和問題も NP 完全あるいは NP 困難である (ただし，$\varepsilon(>0)$ 程度の誤差を許容することにして，$\varepsilon$ を定数として扱うならば，多項式オーダのアルゴリズムが存在することが古くから知られている)．本論文では，部分和問題を考察の対象とする．

一般に離散計画問題に対しては**分枝限定法** (branch and bound method) が広く用いられている．これは，変数を次々と選んでは値を固定することにより原問題をより小さな部分問題に分割し，すべての場合を尽くすという方法である．ただし，いくつかの変数の値が固定されているときに残りの変数に関して当該部分問題の最適解を見積り，既に得られている実行可能解より良い解が得られる見込みがないとき，残りの変数に関して部分問題を生成することはしない．

このように，分枝限定法は，部分問題を系統的に生成する**分枝操作** (branching) と，見込みのない部分問題を避ける**限定操作** (bounding) から成り，一部の可能解だけを生成することによりすべての可能解を調べたのと同等の効果を与える方法である．

分枝限定法は，離散計画問題に対するほとんど唯一の汎用的な実用アルゴリズムであるが，依然として，実行時間は大きい．そのため，個別の問題ごとに，最適解が早く得られるように，分枝操作や限定操作に**確率的要素**を導入する手法が注目されている．これにより，最悪の場合の手間は大きくても，平均の手間や大半の場合の手間を小さくすることが期待される．本論文では，部分問題に対する分枝限定アルゴリズムの分枝操作をランダム化して，そのアルゴリズムの振舞を調べる．

## 2 条件

本論文の前提となる仮定や分枝限定アルゴリズムについて説明する．

## 2.1 定式化

0–1 整数に対するナップザック問題は，一般に次のように定式化される．(0–1 変数の総数を $n$ 個とする．)

目的関数：

$$z = \sum_{j=1}^{n} c_j x_j \rightarrow max$$

制約条件：

$$\sum_{j=1}^{n} a_j x_j \leq b$$

$$x_j \in \{0,1\} \quad , \quad j = 1, \ldots, n$$

本論文で考える 0–1 部分和問題は，$c_j$ と $a_j$ が共に同じ値である問題であり，最適解における目的関数の値は，$b$ 以下である．すなわち，

目的関数：

$$z = \sum_{j=1}^{n} a_j x_j \rightarrow max$$

制約条件：

$$\sum_{j=1}^{n} a_j x_j \leq b$$

$$x_j \in \{0,1\} \quad , \quad j = 1, \ldots, n$$

なお，本論文では，各 $a_j$ $(j = 1, \cdots, n)$ と $b$ は正とする．

## 2.2 分枝限定アルゴリズム

本論文では，部分和問題に対するきわめて単純な分枝限定アルゴリズムを考察する．このアルゴリズムのあらましは次のとおりである．

すべての変数は**自由**あるいは 0，1 のいずれかの値に**固定**の状態を取るものとし，初期値として，すべての変数を自由としておく．自由変数の添字の集合を $F$，0 に固定された変数の添字の集合を $S_0$，1 に固定された変数の添字の集合を $S_1$ とする．

アルゴリズムは以下のとおり再帰的に書かれる．手続き $B\_B$ が呼び出されたとき，暫定解 $\boldsymbol{x}_{\mathrm{opt}}$ の目的関数の値 $z_{\mathrm{temp}}$ が，それまでに得られている最良の解 $\boldsymbol{x}_{\mathrm{opt}}$ の目的関数の値 $z_{\mathrm{opt}}$ よりも良ければ，$\boldsymbol{x}_{\mathrm{opt}}$，$z_{\mathrm{opt}}$ を $\boldsymbol{x}_{\mathrm{temp}}$，$z_{\mathrm{temp}}$ で置き換える．さもなければ，自由な変数の 1 つ $x_p (p \in F)$ を選び，$x_p = 1$ に固定したものと $x_p = 0$ に固定したものについて $B\_B$ をそれぞれ呼ぶ．

本論文では，$B\_B$ 中において自由な変数 $x_p (p \in F)$ を選ぶ部分をランダム化する．

実際の分枝限定アルゴリズムでは，自由な変数の選び方や限定操作に種々の工夫がなされているが，本論文では，とりあえず分枝操作のランダム化の効果だけを調べるのが目的であるので，そのような工夫は考察しない．

以下のアルゴリズムにおいて，解 $\boldsymbol{x}_j$ $(j = 1, \cdots, n)$ は，配列 $x[1 \ldots n]$ に (暫定解 $\boldsymbol{x}_{\mathrm{temp}}$，最適解 $\boldsymbol{x}_{\mathrm{opt}}$ についても同様)，係数 $a_j$ $(j = 1, \cdots, n)$ は配列 $a[1 \ldots n]$ に入っているものとし，集合 $S_0, S_1$ は変数 $S0, S1$ で扱う．

```
procedure B_B(F, sumfree, S0, S1, sum1, x);
  begin
    if sumfree + sum1 ≤ b then
      begin
        x[j] := 1 (j ∈ F ∪ S1), x[i] := 0 (i ∈ S0)
          とした解を xtemp とし,
          ztemp := sumfree + sum1;
        if zopt < ztemp then
          xopt, zopt を
            xtemp, ztemp で置き換える;
      end
    else
      begin
        if sum1 + a[p] ≤ b then
          { このような p ∈ F をランダムに選ぶ }
          begin
            x[p] := 1;
            B_B(F\{p}, sumfree − a[p], S0,
                S1 ∪ {p}, sum1 + a[p], x);
          end;
        x[p] := 0;
        B_B(F\{p}, sumfree, S0 ∪ {p},
            S1, sum1, x);
      end;
  end
begin
  F := すべての変数の添字の集合;
  zopt := 0;
```

**for** $j = 1$ **to** $n$ **do** $x[j] := 0$;
$B\_B(F, \sum_{i=1}^{n} a_i, empty, empty, 0, x)$;
**end**.

### 2.3 仮定

話を単純にするために，次のような仮定をする．

(a) 制約条件式の定数 $b$ は，$a_j$ の組合せで得られる値とする．

(b) $\sum_{j \in S_1} c_j = b$ となったとき，(実質的に) 終了．(それ以上部分問題が作成されない．)

(c) 変数は係数の降順に並べ替えされているとする．

(d) 分枝変数を選択する確率は，一様とする．

$(a)$ は，必ず解が得られることを意味している．また，解は唯一ではなく，複数存在する場合もある．

$(b)$ は，解が複数ある場合，最初に発見した最適解までということである．(ランダム化ではこの点が重要であろう．) また，最適解が見つかった後は，部分問題が作成されないということも意味する．

$(c)$ は，ランダム化という観点からは，あまり影響はないが，ランダム化していないものについても考える場合，話をわかりやすくするための仮定である．

$(d)$ は，分枝変数を選択するのに，(ランダム的にも) とくに戦略的なことは行わないという意味である．

これらの仮定は，入力される問題についてだけでなく，アルゴリズムの振舞においても含まれているものである．

また，アルゴリズムの手間は，$B\_B$ が呼ばれる回数に等しいと考える．これは，生成される部分問題数にも等しいと考えることができる．そのため，本論文では，生成される部分問題数について考察する．

## 3 部分問題数の期待値

分枝限定アルゴリズムの生成する部分問題の数の期待値はどの程度になるのかを考察する．

話を単純にするため，$n$ 個の変数のうち最適解において値が "1" となる変数の数を $k$ 個とする．(実際には，事前に値 $k$ を得るのは，不可能である．)

### 3.1 考え方

原問題の解が得られたとき，部分問題が生成されたところは，最適解において 1 となる変数がすべて選択されるまでに，最適解において 0 となる変数が選択された部分である (図 1 参照) と考えることができる．最適解において $(n-k)$ 個の 0 となる変数のうち $m$ 個目が選択されたときに，$n$ 個のすべての変数における順番を $r$ とする．また，そのときに生成される部分問題数を $g(n-r)$ とおくことにする．したがって，それぞれの $m$ について $r$ の期待値を求め，そのときの部分問題数 ($g(n-r)$) の総和を求めれば良い．

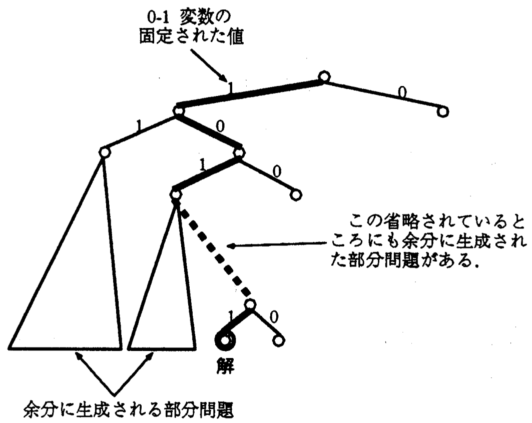


図 1: 部分問題の生成されるところ

また，その中には，$\sum_{j \in S_1} a_j > b$ で限定操作されるものが含まれている．

### 3.2 変数を取り出す順番の期待値

最適解の中で 0 となる変数が $m$ 番目に選択されるときに，すべての変数の中で選択された順番が $r$ 番目であるときの確率とその期待値について考察する．ここで，$r$ の取り得る値は $m, m+1, \ldots, m+k$ である．

まず，それぞれの $r$ における確率を考える．

すべての変数の $n$ 個 $(= k + (n-k))$ の中から

$(n-k)$ 個取り出す組合せの数は $\binom{n}{n-k}$ である．求める場合の組合せは，すべての変数の中の初めの $(r-1)$ 個のどこかに 0 となる変数が $(m-1)$ 個，$r$ 番目に 1 個，すべての変数の中の残りの $(n-r)$ 個のどこかに 0 となる変数の $(n-k-m)$ 個あればよい．(図 2 参照)

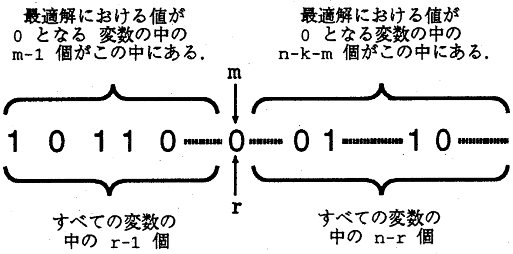


図 2: 変数の選択される組合せ

$$\begin{aligned}&\binom{r-1}{m-1}\cdot\binom{1}{1}\cdot\binom{n-r}{n-k-m}\\&=\binom{r-1}{m-1}\cdot\binom{n-r}{n-k-m}\end{aligned}$$

したがって，求める確率は $r$ を表す確率変数を $R$ とすると次式で与えられる．

$$\begin{aligned}&\mathrm{P}_{k,m}(R=r)\\&=\binom{r-1}{m-1}\cdot\binom{n-r}{n-k-m}\Big/\binom{n}{n-k}\end{aligned}$$

この確率は，**負の超幾何分布**と呼ばれるものであるので，$R$ についての期待値は，

$$\mathrm{E}_{k,m}(R)=\frac{m(n+1)}{n-k+1} \tag{1}$$

と求められ，同様に最適解の中で 1 となる変数の場合は，確率変数を $M$ とすると，次のように求められる．

$$\mathrm{E}_{k,m}(M)=\frac{m(n+1)}{k+1} \tag{2}$$

## 3.3 部分問題数の計算

前節より，生成される部分問題数について考える．

$m$ の取り得る値であるが，これは，初めて最適解の中で 0 となる変数が現れたときから最適解の中で 1 となる変数の $k$ 個目が現れたときまでと考えることができる．これらの値は，式 (1) で $m=1$ とした値から 式 (2) で $m=k$ とした値までのことである．

これらより，$k$ をある値に固定した場合について考える．

$$\sum_{i=\mathrm{E}_{k,1}(R)}^{\mathrm{E}_{k,k}(M)} g(n-i) = \sum_{i=\frac{n+1}{n-k+1}}^{\frac{k}{k+1}n} g(n-i)$$

次に，$k$ が確率分布をする場合の，上式の期待値を考える．

$k$ の分布は原問題の $a_1,\cdots,a_n,b$ の分布に依存するものであり，本論文ではそれらの分布について何の仮定も設けていないので，これ以上のことは言えない．

仮に，ある問題で最適解において "1" となる変数の数が $k$ 個である確率を $\mathrm{P}(k)$ とすると，部分問題の数は，

$$O\left(\sum_{k=1}^{n}\left\{\sum_{i=\frac{n+1}{n-k+1}}^{\frac{k}{k+1}n} g(n-i)\cdot\mathrm{P}(k)\right\}\right) \tag{3}$$

しかし，$\mathrm{P}(k)$ については，本論文では未知であり，3.7 節で原問題の分布が一様であると仮定したものについて計算も行うが，この部分を含めた研究は今後の課題である．

## 3.4 限定操作における部分問題数

分枝限定アルゴリズムでの限定操作による部分問題数の振舞について考え方を中心に述べる．

分枝限定アルゴリズムの振舞において，部分問題の作成について探索木を描いたと考える．そのとき，ある深さ $t$ における部分問題数 (前出したアルゴリズムで $B\_B$ が呼ばれる回数) を $T(t,b-sum1)$ で表すとする．それは，式 (3) までに $g(n-i)$ とされていた部分である．

このとき，自由変数の中から分枝変数として $a_p$ が選択されたとすると，ランダム化していない場合，次のようにモデル化できると考えられる．

$$\begin{aligned}T(t,b-sum1) &= T(t-1,b-sum1-a_p)\\&\quad+T(t-1,b-sum1)\end{aligned} \tag{4}$$

また，ここで初期条件として次のものを与えることができる．

任意の $x, y(>0)$ と $z\ (= b - sum1 - a_p < 0)$ について，

$$T(x,0) = T(0,y) = T(0,0) = 1 \quad (5)$$
$$T(x,z) = 0 \quad (6)$$

ランダム化した場合には，$a_p$ の選択する確率も含めて考える必要がある．$a_p$ を選択する確率を $\mathrm{P}(p)$ と表すとすると，次のようなモデル化が考えられる．

$$\begin{aligned} &T(t, b - sum1) \\ &= \sum_{p \in F} \{T(t-1, b - sum1 - a_p) \\ &\quad + T(t-1, b - sum1)\} \cdot P(p) \end{aligned} \quad (7)$$

初期条件は，同じである．

### 3.4.1 初期条件について

式 (5)，(6)）で示したものは，解が複数存在するものを許す場合である．解が複数存在する場合を考えると複雑になるので，今後は解が唯一であると仮定した場合について考える．

そのときの初期条件は，次のようにすることができる．

任意の $x(>0)$ と $y(<0)$ について，

$$T(x,0) = T(0,0) = 1 \quad (8)$$
$$T(0,y) = 0 \quad (9)$$

これは，解が唯一であるため (最適解が存在しない部分での選択による部分問題の生成であるため)，第 2 引数は (最適解が見つかったという意味で) **0** になることはないからである．そのため，第 2 引数が負になるときを，“0” と表記しても差し支えないと考える．

## 3.5 漸化式の計算について

モデル化した式 (4)，(7) は漸化式であると考えられるが，視点を変えると**偏差分方程式**であるとも考えられる．

しかし，引数を 2 つ以上もつ偏差分方程式を解くことは，一般に難しいとされている．そのため，どのように考えればよいのかについて述べる．

### 3.5.1 考え方

漸化式での 2 つの引数に着目して探索木を描いた場合 (分枝限定アルゴリズムにおいて限定操作に関して考えた場合)，どのような形になるのか考える．

まず，明らかなのは，第 1 引数による深さよりも深くはならないことである．

では，それぞれの分枝変数が 1 に固定されている葉にあたるところの深さの範囲について考える．( 0 となる葉にあたるところは，第 1 引数による深さにほぼ等しいと考えることができる．)

このときの最小の場合であるが，それは，自由変数が大きな順に選択されたときであり，逆に，最大の場合は，小さな順に選択されたときであると考えることができる．

(ランダム化されていないとき)

これまで述べたような漸化式で計算することだけについて考えると，ランダム化したものでは，選択する順序が固定されていないため考えるのが難しくなってしまうことが予想される．しかし，ランダム化していないものでは，**順序が固定**されているので (計算できるかどうかは別として) 考えやすくなる．しかも，0–1 変数は係数の降順に並べ替えられていると仮定しているので，ランダム化していない場合はそれぞれ最小の深さになることがわかる．

しかし，最適解がこの漸化式で計算されるものの中に含まれるのなら (例えば，解が複数ある場合など)，完全には当てはまらない．なぜなら，解が見つかればそこで実質的に終了すると仮定してあるからである[1]．

(ランダム化されているとき)

ランダム化されているときは，次のように考えることができる．

| |
|---|
| すべての並べ方 (**順列**) について考え，それぞれの出現確率を考慮する． |

順列それぞれの出現確率については，分枝変数の選択の確率が一様であると仮定しているため，同じく一様であると仮定することができる．

また，順列の総数は $n$ 個の変数では $n!$ 個となる．しかし，その中には順列どうしの一部が違って

[1] これは，ナップザック問題一般で考えるときに，また，有効になるであろう．

も，漸化式で得られる値は同じものがある．(限定操作により，実際には順列としては途中までしか現れないためである．)

それらが計算できれば，ランダム化したときの値がわかると考えられる．

## 3.6 計算について

一般的に考えるには難しいので，計算の仕方の概要を述べる．

また，係数がすべて同じ値 (例えば，すべての係数の平均に置き換えたもの) に変換したものと仮定し，ランダム化していない場合 (もし，平均で置き換えたとすれば，アルゴリズムがランダム化されていても同じである．) について述べる．

値を一定とした係数の値を $h$ とする．そうすると，漸化式の第 2 引数の部分を

$$b - sum1\,(\,-\,a_p\,) \Longrightarrow \frac{b - sum1}{h}\,(\,-\,1\,) \quad (10)$$

とすることにより，($\frac{b-sum1}{h}$ を $w$ として)

$$\begin{aligned} T(t,w) &= T(t-1,w-1) \\ &\quad + T(t-1,w) \end{aligned} \quad (11)$$

と書き換えることができる．これによりいくらかは計算しやすくなる．

### 3.6.1 母関数の利用による計算

ここで，第 1 引数である $t$ の値をある値に固定し，母関数を利用した計算について考える．

$\alpha$ を任意の実数変数として，式 (11) を母関数 $\phi_t(\alpha)$ で表現すると，

$$\begin{aligned} \phi_t(\alpha) &= T(t,0) + T(t,1)\cdot\alpha \\ &\quad + T(t,2)\cdot\alpha^2 + \cdots \\ &= \sum_{k=0}^{\infty} T(t,k)\cdot\alpha^k \end{aligned} \quad (12)$$

となる．

また，式 (11) より，

$$T(t,w) - T(t-1,w-1) - T(t-1,w) = 0$$

と置き換えることができる．これを (右辺の第 2 項の第 2 引数だけが違うのに注意して) 母関数で書き直して，

$$\begin{aligned} &\phi_t(\alpha) - \phi_{t-1}(\alpha)\cdot\alpha - \phi_{t-1}(\alpha) \\ &\quad = \{T(t,0) - T(t-1,0)\} \\ &\qquad + \alpha\cdot\{T(t,1) - T(t-1,0) - T(t-1,1)\} \\ &\qquad + \alpha^2\cdot\{T(t,2) - T(t-1,1) - T(t-1,2)\} \\ &\qquad + \cdots\cdots\cdots \end{aligned}$$

初期条件である式 (9) より，

$$\begin{aligned} \phi_0(\alpha) &= 1 \\ \phi_1(\alpha) &= \alpha\cdot\phi_0(\alpha) + \phi_0(\alpha) = \alpha + 1 \\ \phi_2(\alpha) &= \alpha\cdot\phi_1(\alpha) + \phi_1(\alpha) = (\alpha + 1)^2 \\ &\vdots \\ \phi_t(\alpha) &= (\alpha + 1)^t \end{aligned} \quad (13)$$

と，次々に求めていくことができる．

これで母関数の形がわかった．ここで母関数を展開し $\alpha$ の昇順に並べ替えたものの第 $w$ 項の係数は，求めている $T(t,w)$ の値である．また，式 (13) を展開したときの係数は 2 項係数である．

なお 2 項係数については，次の関係を利用する．

$$\binom{n}{k} \le \frac{n^k}{2^{k-1}} \quad (14)$$

この不等式の証明は，数学的帰納法により証明することができるため，省略する．なお，2 項係数であるため，$k$ の値が $\frac{1}{2}n$ での対称性に注意が必要である．

また，この式を変形すると，次の不等式も得られる．

$$1\Big/\binom{n}{k} \ge \frac{2^{k-1}}{n^k} \quad (15)$$

さらに，$k$ を $n-i$ で置き換えると，

$$2^i\Big/\binom{n}{n-i} \ge \frac{2^{n-1}}{n^{n-i}}$$

とも変形できる．

これらの関係より，$w$ が $\frac{1}{2}t$ のとき，2 項係数の値が最大となり，そのときスターリングの公式も利用して計算すると，

$$T(t,w) = \binom{t}{w} < \frac{2^t}{\sqrt{t}} \quad (16)$$

という値が求まる．ただし，この値は期待値というよりは，最大値である．

また，本章ではそれぞれの分枝変数の選択される確率は一様であると仮定したので，ここでは，ランダム化されているものも同様に考えられる．

厳密な計算量的な値は別として，考え方や計算方法についてはこのような形になると考えてよい．

式 (10) にある $h$ や変換そのものについて適切なものを考えれば，厳密な計算量，または，近似が計算できると考えられる．

## 3.7 原問題の分布について

式 (3) における $\mathrm{P}(k)$ について考える．

$k$ は最適解が得られたときの 0-1 変数の値が 1 であるものの数であると定義した．そこで，その $k$ の分布の確率を $\mathrm{P}(k)$ とおく．

また，$k$ の分布は原問題における $a_1, \cdots, a_n, b$ の分布に関係があることがわかる．もう少し広く考えると，$k$ の分布は原問題の分布とほぼ等しいということができる．

しかし，実際に原問題の分布を得るのは，難しい．実際には，大量の原問題の標本を取り，その統計によりある確率分布に推定するのが普通であると考えられる．そのため，今回は原問題の分布は一様であると仮定して考える．

### 3.7.1 ある $k$ における確率

まず，すべての考えられる組合せは，$\binom{n}{k}$ で $k$ が 0 から $n$ までにしたものの総和である．したがって，

$$\sum_{k=0}^{n} \binom{n}{k} = \sum_{k=0}^{n} \binom{n}{k} \cdot 1^k \cdot 1^{n-k} = (1+1)^n = 2^n$$

そして，ある $k$ における確率 $\mathrm{P}(k)$ は，分布は一様であると仮定したので，そのとき，次のようになると考えられる．

$$\mathrm{P}(k) = \binom{n}{k} \Big/ 2^n \qquad (17)$$

また，すべての $k$ における総和は $\mathbf{1}$ となる．

### 3.7.2 $k$ の期待値

$k$ の分布についての確率がわかったので，その期待値を求めてみる．

$$\begin{aligned} \sum_{k=0}^{n} k \cdot \frac{\binom{n}{k}}{2^n} &= \frac{1}{2^n} \sum_{k=0}^{n} k \cdot \binom{n}{k} && (18) \\ &= \frac{1}{2^n} \cdot n \cdot 2^{n-1} && (19) \\ &= \frac{1}{2} n && (20) \end{aligned}$$

これは，直観的に考えられるものであるが，実際に計算により求めることができた．

### 3.7.3 一様分布の場合の部分問題数の期待値

そこで，原問題の分布が一様とした場合の $k$ についての期待値と確率が得られたので，これまでに求められた式も含め，式 (3) の中に代入してみる．

式 (16)，(17) より，スターリングの公式も使って期待値を求めると，

$$\begin{aligned} &\sum_{i=1}^{n} \left\{ \left( \sum_{i=\mathrm{E}_{k,1}(R)}^{\mathrm{E}_{k,k}(M)} g(n-i) \right) \cdot \frac{\binom{n}{k}}{2^n} \right\} \\ &< \sum_{i=1}^{n} \left\{ O\left( \frac{2^{n-\frac{n+1}{n-k+1}}}{\sqrt{n-\frac{n+1}{n-k+1}}} \right) \cdot \frac{\binom{n}{k}}{2^n} \right\} \\ &= O\left( \sum_{i=1}^{n} \frac{1}{\sqrt{n-\frac{n+1}{n-k+1}} \cdot 2^{\frac{n+1}{n-k+1}}} \cdot \binom{n}{k} \right) \\ &< O\left( \frac{2^n}{n} \right) \end{aligned}$$

しかし，この値は期待値として過大評価である．

# 4 最悪の場合について

本論文の分枝限定アルゴリズムは，最良の場合でも $k$ 回 $B\_B$ を呼ぶことは自明である．今，$k$ については何も仮定していないので，$k$ は $n$ 程度とすれば，この分枝限定アルゴリズムの手間の下界として $\Omega(n)$ が得られる．

しかし，最悪の場合についても，原問題の $a_1, \cdots, a_n, b$ について何らかの仮定があれば，分枝限定アルゴリズムの手間について詳細がわかるはずである．

## 4.1 最悪と考えられる場合の確率

最悪と考えられる場合は，最適解の中で 1 となる $k$ 個の変数のうちのどれか 1 つが初めて選択される前に最適解の中で 0 となる $(n-k)$ 個の変数すべてが選択された場合であり，しかも，解が唯一の場合である．

そのとき，限定される部分問題の総数は，最低でも，2 個から $(k-1)$ 個の変数により作成される部分問題より，多いと考えることができる．

ここで，$Y$ を解の変数の値が 最適解の中で "1" となる数が $k$ 個の場合に最悪となる確率を表す確率変数とする．したがって，最適解の中で 0 となる $(n-k)$ 個の変数が選択される確率は次のようになる．(図 3 参照)

$$
\begin{aligned}
\mathrm{P}(Y=k) &= \frac{n-k}{n}\cdot\frac{n-k-1}{n-1}\cdots\frac{1}{n-k+1} \\
&= 1\Big/\binom{n}{n-k} \\
&= 1\Big/\binom{n}{k} \qquad (21)
\end{aligned}
$$

## 4.2 限定される部分問題数

最悪と考えられる場合は，最適解の中で 1 となる $k$ 個の変数のうちのどれか 1 つが初めて選択される前に最適解の中で 0 となる $(n-k)$ 個の変数すべてが選択された場合 (図 3) であり，しかも，解が唯一の場合である．

図 3 で示してあるように，限定操作される部分問題の数は，$k$ がある値であるときに，

$$\sum_{i=1}^{n-k} g(n-i) \qquad (22)$$

で計算できる．ここで $g(n-i)$ は，式 (4)，(7) で示されているものが入る．

式 (3) のものと，部分問題の生成される範囲について注意が必要である．

## 4.3 最悪の場合の部分問題数の期待値

$k$ を $i$ におきかえ，それが $k$ から $(n-1)$ までについて，それぞれの確率と部分問題数の積の総和を計算すると，限定操作される部分問題の数の最悪

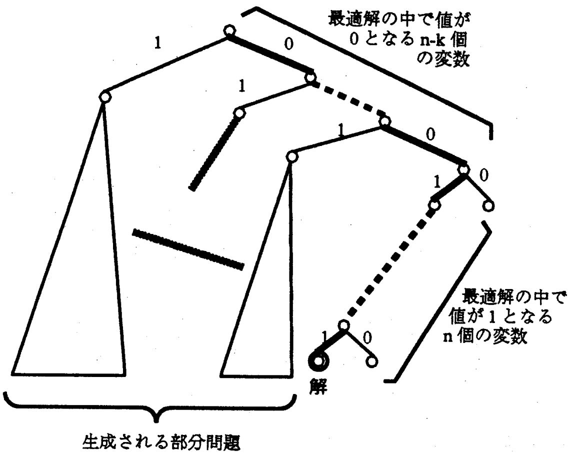


図 3: 最悪の場合の探索木

の場合の期待値が求まる．最悪の場合に限定操作される部分問題の数を $X_w$ とおくと，その期待値は，

$$\mathrm{E}[X_w] = \sum_{k=1}^{n-1}\left\{\left(\sum_{i=1}^{n-k} g(n-i)\right)\Big/\binom{n}{k}\right\}$$

と表すことができ，この式を注意深く観察し，これまでに求められている式や (最大としての) 値も利用して，

$$\mathrm{E}[X_w] \leq O\left(\frac{2^n}{\sqrt{n}}\right)$$

と求めることができる．

したがって，ランダム化により，生成される部分問題は最悪でも $O\left(\frac{2^n}{\sqrt{n}}\right)$ より少ない．

## 4.4 悪い場合の確率

悪い (つまり，良くない) と考えられるのはどの程度であるのか．具体的にすぐには出ないであろう．

そこで，本論文では次のように考える．

> 最悪の場合から，少し良いと考えられるところまでに含まれる確率はどの程度になるのか？

ここで，範囲を示す変数として，$d$ とする．ただし，$1 \leq d \ll n-k$ とする．

そのとき，求める確率は，式 (22 ) より $\mathrm{P}(X = \sum_{i=1}^{n-(k+d)} g(n-i))$ である．また，確率は式 (21)

より求められる．

$$\begin{aligned} \mathrm{P}\left(X = \sum_{i=1}^{n-(k+d)} g(n-i)\right) &= \mathrm{P}(Y = k+d) \\ &= 1 \Big/ \binom{n}{k+d} \\ &< O\left(\left(\frac{k}{n}\right)^k\right) \end{aligned}$$

このように，悪いと考えられる場合は，$k$ と 大きな $n$ について確率 $O\left(\left(\frac{k}{n}\right)^k\right)$ より小さい．

ただし，この近似した値は，$k$ が $\frac{1}{2}n$ について対称であることに注意する．

#### 4.4.1 悪い場合の確率の期待値

ここで期待値的に求めるために $k$ の値として式 (20) を用いると，

$$\begin{aligned} \mathrm{P}(Y = k+d) &= \mathrm{P}(Y = \frac{1}{2}n + d) \\ &= 1 \Big/ \binom{n}{\frac{1}{2}n + d} \\ &< O\left(\left(\frac{\frac{1}{2}n}{n}\right)^{\frac{1}{2}n}\right) \\ &= O\left(\left(\frac{1}{\sqrt{2}}\right)^n\right) \end{aligned}$$

このように，悪いと考えられる場合は，大きな $n$ について確率 $O\left(\left(\frac{1}{\sqrt{2}}\right)^n\right)$ より小さい．

また，どんなに悪くとも (対称性も考えて)，$k=1$ または $k=n-1$ の場合の確率である $\frac{1}{n}$ よりも小さいことも付け加えておく．(なお，$k=0$ や $k=n$ の場合は，最良，平均，最悪のすべての場合で同じである．)

## 5 考察とまとめ

本論文では，0–1 部分和問題を解く分枝限定アルゴリズムで分枝変数の選択をランダム化したときに生成される部分問題数の期待値と最悪の場合の部分問題数とその事象が起こる確率について考察した．

最悪に近い事象が起こる確率は，$n$ が十分に大きいとかなり小さくなることがわかった．

しかし，これは 2.3 節で示したとおり，さまざまな条件を付加したためである．そのうちの 1 つを除いただけでも，効率が悪くなり，また，解析するのも難しくなるだろう．これは，部分和問題だけではなく，ナップザック問題一般に拡張するということも同様である．

部分問題数は，本論文では取り上げていない戦略を考察することにより減少することができる．そのため，限定操作される部分問題の数は，本論文で述べたアルゴリズムによる厳密な値ではないこともつけ加えておく．ただし，それが求められたとき限定操作される部分問題の数はもっと増加すると考えるのは自然である．

## 6 今後の課題

最適解の中で 1 となる変数の数 $k$ の分布について考察し，期待値を求めることが残った．$k$ の分布についてどのような確率分布としてモデル化を行うのか十分に検討しなければならない．そして，限定操作における振舞についても第 3.4 章で述べたような展望から発展させなければならない．

戦略的なことやランダム化については，他にもさまざまなことについて考えることができ，それらについての分析も必要である．また，それらを組み合わせたものについても同様である．

他にも，ランダム化したものについての解析を応用し，ランダム化していない従来のものについても考えていく必要がある．これは，ランダム化との比較という意味のためだけでなく，本質的な特徴の解析という意味でも重要である．

## 参考文献

[1] R.G.Jeroslow, “There cannot be any algorithm for integer programming with quadratic constraints,” *Operations Research*, Vol.21, pp.221–224, 1973

[2] 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

[3] 徳山 豪，“ランダムアルゴリズムの話題から”，電子情報通信学会誌，Vol.77,No.9，pp.957 – 967，1994

[4] Prabhakar Raghavan, "Lecture Notes on Randomized Algorithms," IBM Research Report RC 15240, 1989

[5] G.Blom, L.Holst, D.Sandel 著, 森真 訳, "確率問題ゼミ –コイン投げからランダム・ウォークまで–", シュプリンガー・フェアラーク東京株式会社, 1995 ("Problems and Snapshots from the World of Probability," Springer-Verlag New York, Inc., 1994)