コンピュータと教育 38－4
数理モデル化と問題解決 4－4
(1995. 11. 17)

ナップザック問題の分枝限定アルゴリズムのランダム化とその解析

福田 和真　岩沢 京子　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．

An Analysis of Randomized Branch and Bound Algorithm of the Knapsack Problem

Kazuma Fukuda　Kyoko Iwasawa　Mario Nakamori
Department of Computer Science
Tokyo University of Agriculture and Tecnology

A randomized branch and bound algorithm for the subset-sum problem which is a special case of the knapsack problem is analized from the point of view of computational complexity.

A branch and bound algorithm chooses variables successively, fixing each to value 0 or value 1 and thus deviding the original problem into smaller subproblems. This is called the branching process. In the present paper variables are chosen at random in the branching process.

The computational complexity of our randomized algorithm is measured by the number of subproblems generated. The expected value of the number of subproblems is estimated and that of the probability that that worst case takes place is obtained.

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，ε 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 は非負とする．

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 $\boldsymbol{x}_{\rm opt}$ の目的関数の値 $z_{\rm temp}$ が，それまでに得られている最良の解 $\boldsymbol{x}_{\rm opt}$ の目的関数の値 $z_{\rm opt}$ よりも良ければ，$\boldsymbol{x}_{\rm opt}$，$z_{\rm opt}$ を $\boldsymbol{x}_{\rm temp}$，$z_{\rm temp}$ 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 x_j $(j=1,\cdots,n)$ は，配列 $x[1\ldots n]$ に(暫定解 x_{temp}，最適解 x_{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 − a[p], 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, zopt, 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，最後に選択された解の値として 1 となった変数までに選択された解の中で 0 となった変数のところであると考えることができる．m 個目の 0 の変数が選択されたときの順番を r とする

と，生成される部分問題数は，2^{n-r} と考えることができる．したがって，それぞれの m について r の期待値を求め，すべての m についての部分問題数の総和を求めれば良い．

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．しかし，本論文では，分枝操作における確率的方法による効果について考察するため，限定操作による効果には触れない．

3.2 変数を取り出す順番の期待値

最適解の中で 0 として m 番目に選択される分枝変数が全体の 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．

全体として，n 個 $(= k + (n-k))$ のうち $(n-k)$ 個取り出す組合せの数は $\begin{pmatrix} n \\ n-k \end{pmatrix}$ であるので，求める場合の組合せは，初めの $(r-1)$ 個のどこかに $(m-1)$ 個，r 番目に 1 個，残りの $(n-r)$ 個のどこかに $(n-k-m)$ 個あればよい．

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

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．

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

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，

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

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．

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

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．

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

言うまでもなく，上式は限定操作の効果を考慮していないので，過大評価である．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，

$$O\left(\sum_{k=1}^{n} 2^n \cdot \mathrm{P}(k)\right)$$

しかし，$\mathrm{P}(k)$ については，本論文では未知であり，この部分を含めた研究は今後の課題である．限定操作についての考察も含めたときに，意味があると考えている．

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 1 参照)

$$\begin{aligned} P(Y=k) &= \frac{n-k}{n}\cdot\frac{n-k-1}{n-1}\cdots\frac{1}{n-k+1} \\ &= 1\Big/\binom{n}{n-k} \\ &= 1\Big/\binom{n}{k} \end{aligned} \tag{3}$$

4.2 限定される部分問題数

図 1 で示してあるように，限定操作される部分問題の数は，まず，図 1.(1) の部分について計算し，その深さが 2 から $(k-1)$ としたものの総和 (図 1.(2)) を求めたものである．

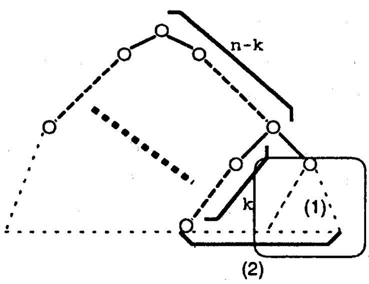

図 1: 探索木

まず，図 1.(1) の部分問題の数について考えると，これは公比 2 の等比数列の和であるから，

$$\sum_{i=2}^{k-1} 2^i = O(2^k) \tag{4}$$

また，図 1.(2) の部分問題の数は，式 (4) で k を i とおきかえ，それの 2 から $(k-1)$ までの総和だから，次のように求まる．

$$O\left(\sum_{i=2}^{k-1} 2^i\right) = O(2^k) \tag{5}$$

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，

$$E[X_w] = O\left(\sum_{i=k}^{n-1}\left\{2^i\Big/\binom{n}{n-i}\right\}\right)$$

この計算には，次の関係を利用する．

$$\binom{n}{k} \le \frac{n^k}{2^{k-1}}$$

この不等式の証明は，数学的帰納法により証明できる．この式を変形すると，次の不等式も得られる．

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

k を $n-i$ で置き換えると，

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

と変形できる．これらより，

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

最悪の場合の部分問題の総数を X とおくと，期待値は部分問題の総数 2^n だから，

$$\begin{aligned} E[X] &= 2^n - E[X_w] \\ &\le 2^n - O\left(\frac{2^n}{n}\right) \end{aligned}$$

ランダム化によって，部分問題は最悪でも $O(2^{n-\log n})$ だけ少なく列挙される．

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 (5) より $\mathrm{P}(X = 2^n - 2^{k+d} + 2(k+d))$ である．また，確率は式 (4) より求められる．

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

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 $O\left(\frac{k}{n}\right)^k$ より小さい．

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．

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)