北海道大学工学部研究報告
第146号（平成元年）

Bulletin of the Faculty of Engineering,
Hokkaido University. No. 146 (1989)

変数上下限制約付き全整数計画問題に対する陰的列挙法の効率改善について

姉崎　淳　大柳　俊夫　加地　郁夫

（昭和63年12月20日受理）

Improvement in Efficiency of Implicit Enumeration Method for the Bounded Variable Pure Integer Programming Problem

Jun ANEZAKI, Toshio OHYANAGI and Ikuo KAJI

(Received December 20, 1988)

Abstract

In this paper an implicit enumeration method for the bounded variable pure integer programming problem developed by Trotter and Shetty is introduced and some new augmentation methods for the algorithm are proposed. Then several numerical experiments with those proposed methods are made using two different types of random problem. From the results, the best augmentation method is presented.

1．はじめに

全整数計画問題において，変数に対する実際的な意味合いからすべての変数に上下限制約を付けられる場合は多く，このような問題は変数上下限制約付き全整数計画問題と呼ばれる。この問題の解法はいろいろと提案されており，その1つにTrotterとShettyによる陰的列挙法がある[1]。この方法は分枝限定法に基づく探索的手法で，実行可能解と実行不可能解を陽的に列挙する分枝操作と陰的に列挙する限定操作により探索は進められる。

本論文はTrotterとShettyの陰的列挙法で行われる分枝操作として，TrotterとShettyとは異なる方法を提案し，その有効性を調べるために行った数値実験について述べるものである。まずTrotterとShettyの陰的列挙法を説明する。次に，今回提案する分枝操作の方法を説明する。そして，その方法を用いた場合の探索効率の変化を調べる数値実験について述べる。最後に実験結果について考察を加え，最良と思われる分枝操作の方法を示す。

情報工学科　システム工学講座

2．Trotter と Shetty の陰的列挙法

2.1　陰的列挙法のアルゴリズム

一般に変数上下限制約付き全整数計画問題は

問題(P)　目的関数　$z=\sum_{j=1}^{n} c_j x_j \rightarrow$ 最小化

$$\text{制約条件}\quad \sum_{j=1}^{n} a_{ij}x_j \geqq b_j \quad (i=1,\ 2,\ \cdots,\ m) \tag{2.1}$$

$$0 \leqq x_j \leqq d_j \quad \text{かつ} \quad x_j \text{は整数} \quad (j=1,\ 2,\ \cdots,\ n)$$

と定式化される。ただし，$c_j \geqq 0\ (j=1,\ \cdots,\ n)$である。この問題は，変数変換を行い0－1全整数計画問題に定式化し直し，Balasの陰的列挙法で解くことができる[2),3),4)]。しかし，変数変換を行うと一般に扱う変数の数が多くなり，最適解の探索が難しくなる。これに対し，変数変換を行わずに問題(P)を解く方法の1つにTrotterとShettyによる陰的列挙法がある。この方法は，Balasの陰的列挙法と同様に分枝限定法を基本とする探索的手法で，分枝操作と限定操作による最適解の探索過程で，すべての実行可能解と実行不可能解を陽的あるいは陰的に列挙する。この方法における分枝操作とは，分枝変数すなわち値を固定する変数を1つ決めてその値を上下限制約を満足する任意の整数値に固定し，現在の部分解に追加して新しい部分解を生成する操作である。ここで部分解とは，分枝によって現在値が固定されている変数とその変数の値の集合のことである。また限定操作とは，ある部分解に対してもはやこれ以上分枝操作を続ける必要がないことがわかったときに後戻りをする操作である。後戻りすることを部分解の終決と呼ぶ。この探索の様子は，図1に示すような列挙木で表される。列挙木において各ノードは陽的に列挙された1つの解を表し，ノードとして表れない解は陰的に列挙された解となる。

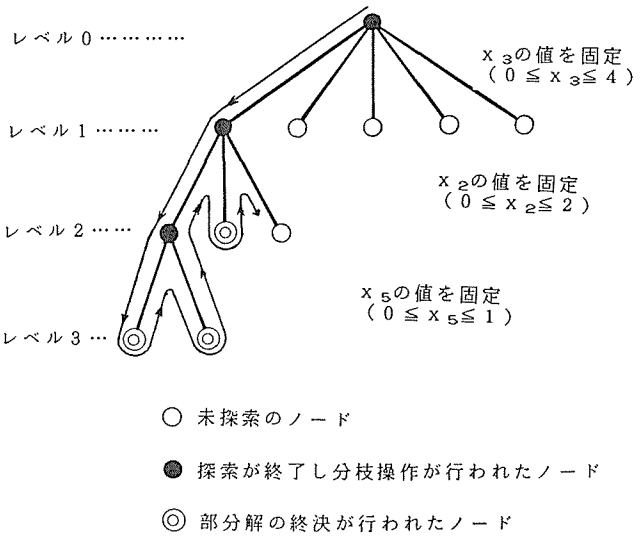


図1　探索経路の図式（列挙次）

さて，第k番目に生成される部分解の終決の判断は，以下で定義される部分問題を考えて行われる。

問題(P_k)　目的関数　$z_k=\sum_{j\in F^k} c_j x_j+\sum_{j\in S^k} c_j x_j \rightarrow$ 最小化

$$\text{制約条件}\quad \sum_{j\in F^k} a_{ij}x_j \geqq b_i^k \quad (i=1,\ 2,\ \cdots,\ m) \tag{2.2}$$

$$0 \leqq x_j \leqq d_j \quad \text{かつ} \quad x_j \text{は整数} \quad (j\in F^k)$$

ただし，S^kは部分解に含まれる変数の添字集合，F^kは部分解に含まれない変数(自由変数)の添字集合，そして，

$$b_i^k=b_i-\sum_{j\in S^k} a_{ij}x_j$$

である。この部分問題に対して自由変数の値を上下限制約を満たす整数値とした解の集合を部分解の完備解と呼ぶ。また，探索の過程で発見された実行可能解の中でzを最小にする解を暫定解$(\bar{x})$，暫定解に対するzの値を暫定値$(\bar{z})$と呼ぶことにすると，部分解の終決はつぎの2つの場合に行われる。

(1) 実行可能性による終決

ある部分解の完備解の中で，zを最小にする実行可能解を発見した場合には探索は終決する。この際，その解のみが陽的に列挙され他の解は陰的に列挙される。そして，発見された解に対するzの値が$\bar{z}$よりも小さくなる場合は，$\bar{x}$と$\bar{z}$はそれぞれ発見された解およびその解に対するzの値で更新される。

(2) 実行不可能性による終決

ある部分解に対する完備解がすべて実行不可能である場合，あるいは完備解の中のすべての実行可能解に対するzの値が$\bar{z}$よりも小さくないことが判明した場合には探索は終決する。

TrotterとShettyの陰的列挙法では，最適解の探索効率を高めるために，この(1), (2)を判断するためのテストが繰返し行われる。これに関しては次節でのべる。

また，部分問題(P_k)に対して，ある自由変数x_jの値がd_j^k $(<d_j)$より大きい完備解の中には暫定解を更新する解が存在しないことが判明した場合，また，ある自由変数x_jの値が$l_j^k(>l_j)$より小さい完備解の中には暫定解を更新する解が存在しないことが判明した場合には，変数の上下限制約をそれぞれ，$x_j \leqq d_j^k$, $l_j^k \leqq x_j$と変更することができる。そして，このように変更すると各変数の取り得る値の範囲が狭くなり，列挙木の拡大を抑えることができるので，探索効率を上げることが可能となる。そこで，部分問題として，

問題(P_k')　目的関数　$z_k' = \sum_{j\in F^k} c_j x_j + \sum_{j\in S^k} c_j x_j \rightarrow$ 最小化

制約条件　$\sum_{j\in F^k} a_{ij} x_j \geqq b_i^k \quad (j=1, 2, \cdots, m)$　(2.3)

$l_j^k \leqq x_j \leqq d_j^k$　かつ　x_jは整数　$(j\in F^k)$

を考えることにする。

2.2 LPを用いたテストと代理制約式について

TrotterとShettyの陰的列挙法の探索効率を高める方法として，部分解の終決を判断するために線形計画法（Linear Programming, 以下LP）を用いたテストを行うこと，およびLP問題を解いた結果から計算される代理制約式を部分問題に追加して探索を進めることが知られている。これらの方法では，問題(P_k')から変数の整数制約を取り除いた連続緩和問題の双対問題

問題(D_k')　目的関数　$\sum_{i\in M} u_i b_i^k - \sum_{j\in F^k} v_j d_j^k + \sum_{j\in F^k} w_j l_j^k + \sum_{j\in S^k} c_j x_j \rightarrow$ 最大化

ただし，$M=\{1, 2, \cdots, m\}$とする

制約条件　$\sum_{i\in M} u_i a_{ij} - v_j + w_j \leqq c_j \qquad (j\in F^k)$　(2.4)

$u_i \geqq 0$, $w_j \geqq 0 \qquad (i\in M, j\in F^k)$

を考え，その結果を用いることになる。問題(D_k')を解いた結果，部分解の終決が生じるのは以下の3つの場合である。

(1) 非有界な解が得られた場合は，部分問題の緩和問題は解なしとなり現在の部分解は終決する。

(2) 最適解が得られ，かつ最適解の目的関数値が暫定値より大きい場合は，これ以上分枝操作を続けても暫定解を更新できないので現在の部分解は終決する。

(3) 最適解が得られ，かつ最適解の目的関数値が暫定値より小さいかあるいは等しく，かつ最適解が得られたときのシンプレックス乗数ベクトルの成分がすべて整数である場合は，部分問題

の連続緩和問題の最適解の成分がすべて整数であるから，部分問題の完備解のなかで目的関数値を最小にする実行可能解を発見したことになり，現在の部分解は終決する。

またこれら以外の場合は，代理制約式を作成し制約式として部分問題に追加する。

3．分枝操作の改良

3.1 探索効率を良くするための要点

Trotter と Shetty の陰的列挙法の探索効率を良くするには，列挙木の拡大を抑えることが重要で，そのためには，探索の早い段階で目的関数値の小さい暫定解を発見し，限定操作が有効に行われるようにしなければならない。暫定解の発見は分枝変数の選択方法とその値の決定方法に大きく依存するので，結局，このことが探索の効率を良くするための要点となる。

そこで本論文では，Trotter と Shetty が用いている方法の他に，新たに3つの分枝変数の選択方法と2つの値の決定方法を提案する。

3.2 分枝変数の選択方法について

一般的に分枝変数の選択は，解こうとする全整数計画問題の特殊構造や変数の現実的な意味合いを考慮して，過去に同種の問題を解いた経験から最良と思われる方法を用いることが有効であるとされている。しかし，本実験で取り扱うランダムな問題のように，特殊な構造を持たない問題や変数の意味付けのできない問題も数多くあり，それらの問題に対しても，探索効率を良くする分枝変数の選択方法が必要となる。Trotter と Shetty は数値実験の中で，次の2つの選択方法を用いている。

(1) 実行可能性に基づく選択方法

なるべく早い段階に問題の実行可能解を得るように分枝変数を選択する。具体的には，問題 (P_k') に対して，

$$\alpha_t=\sum_{i=1}^{m}\min\{0,\ a_{it}d_t^k-b_i^k\} \qquad (t\in F^k) \tag{3.1}$$

とし，

$$\alpha_b=\max_{t\in F_k}\alpha_t \tag{3.2}$$

に対応する自由変数 x_b を分枝変数として選択する。

(2) 変数の上下限値に基づく選択方法

列挙木に付け加えられる枝の数をなるべく少なくするように分枝変数を選択する。具体的には問題 (P_k') に対して，

$$d_b^k-l_b^k=\min_{t\in F^k}(d_t^k-l_t^k) \tag{3.4}$$

に対応する自由変数 x_b を分枝変数として選択する。

本論文では，これら2つの選択方法に対して，新しい選択方法として以下の3つを提案する。

(3) 目的関数値に基づく選択方法（その1）

暫定解を発見した場合に，暫定値ができる限り小さくなるように分枝変数を選択する。具体的には，

$$c_b=\min_{t\in F^k}c_t \tag{3.5}$$

に対応する自由変数 x_b を分枝変数として選択する。

(4)　目的関数値に基づく選択方法（その２）

暫定解がすでに発見されている場合に，“現在の部分解の完備解の中には暫定値を更新する解は存在しない”という判断が，有効に行われるようにする。具体的には，

$$c_b=\max_{t\in F^k} c_t \tag{3.6}$$

に対応する自由変数 x_b を分枝変数として選択する。

(5)　ＬＰ緩和解に基づく選択方法

部分問題の緩和問題の最適解の成分の中で，値が上限値あるいは下限値に最も近い変数は，問題(P)の最適解においてもそれぞれ上限値あるいは下限値に近い値を取ることが予測される。そこで，その変数を分枝変数として選択する。具体的には，問題(P_k')の緩和問題の最適解を $x^{k(\mathrm{LP})}=\{x_t^{k(\mathrm{LP})} \mid t\in F^k\}$ とすると，分枝変数の選択方法として以下の３つが考えられる。

（ａ）　$d_b^k-x_b^{k(\mathrm{LP})}=\min_{t\in F^k}(d_t^k-x_t^{k(\mathrm{LP})})$ (3.7)

に対応する自由変数 x_b を分枝変数として選択する。

（ｂ）　$x_b^{k(\mathrm{LP})}-l_b^k=\min_{t\in F^k}(x_t^{k(\mathrm{LP})}-l_t^k)$ (3.8)

に対応する自由変数 x_b を分枝変数として選択する。

（ｃ）　$\beta_t=\min\{d_t^k-x_t^{k(\mathrm{LP})},\ x_t^{k(\mathrm{LP})}-l_t^k\}\quad(t\in F^k)$ (3.9)

とし，

$$\beta_b=\min_{t\in F^k}\beta_t \tag{3.10}$$

に対応する自由変数 x_b を分枝変数として選択する。

ただし，（ａ）から（ｃ）の選択は，次節で述べる値の決定方法に依存し，値の決定方法が(イ)のときは（ａ），(ロ)のときは（ｂ），そして(ハ)と(ニ)のときは（ｃ）となる。

3.3　分枝変数の値の決定方法について

第ｋ番目の部分解から新しい部分解を生成するときに用いる分枝変数を x_b とし，x_b の上下限値をそれぞれ d_b^k，l_b^k とする。すると新しく生成可能な部分解は，固定する x_b の値によって $d_b^k-l_b^k+1$ 通り考えられる。分枝操作では，まず最初にこの中の１つを生成し，そして探索が進み，限定操作により第ｋ番目の部分解へ後戻りしたときに x_b の値を変更して，まだ生成されていない部分解を生成する。この部分解の生成は，第ｋ番目の部分解が終決するまで続けられる。

ここで問題となることは，最初の x_b の値の固定方法と後戻りした際の値の変更方法である。Trotter と Shetty は，以下の(イ)，(ロ) ２つの方法を用いて実験を行っている。

(イ)　x_b の値を l_b^k として部分解を生成する。そして，この部分解が終決した際には x_b の値を l_b^k+1 に変更する。さらに終決が生じ x_b の値を変更する場合には，その値を１ずつ増加する。

(ロ)　x_b の値を d_b^k として部分解を生成する。そして，この部分解が終決した際には x_b の値を d_b^k-1 に変更する。さらに終決が生じ x_b の値を変更する場合には，その値を１ずつ減少する。

本論文では，これらの方法に対し 2.2 で述べたＬＰを用いたテスト結果から得られる部分問題の緩和問題の最適解を用いて，x_b の最初の値を固定する方法と後戻りした際に x_b の値を変更する方法を２つ提案する。

(ハ)(a) $x_b^{k(\mathrm{LP})}-l_b^k \leqq d_b^k-x_b^{k(\mathrm{LP})}$ ならば，最初の x_b の値を l_b^k として部分解を生成する。そして，この部分解が終決した際には x_b の値を l_b^k+1 に変更する。さらに x_b の値を変更する場合はその値を1増加させる。

(b) $x_b^{k(\mathrm{LP})}-l_b^k > d_b^k-x_b^{k(\mathrm{LP})}$ ならば，最初の x_b の値を d_b^k として部分解を生成する。そして，この部分解か終決した際には x_b の値を d_b^k-1 に変更する。さらに x_b の値を変更する場合はその値を1減少させる。

(ニ) 最初の x_b の値は $x_b^{k(\mathrm{LP})}$ を丸めた値（これを $[x_b^{k(\mathrm{LP})}]$ と書く）として部分解を生成する。以後，x_b の値を変更する場合は上下限制約の範囲内で，$[x_b^{k(\mathrm{LP})}]+1$，$[x_b^{k(\mathrm{LP})}]-1$，$[x_b^{k(\mathrm{LP})}]+2$，$[x_b^{k(\mathrm{LP})}]-2$，…という順序で変更する。

4．数値実験

4.1 実験方法

分枝変数の選択方法と分枝変数の値の決定方法を変えたときの探索効率の変化を調べる数値実験を行う。なお実験では，プログラムの全実行時間からデータの入出力に要した時間を除いた時間を測定する。そして，この時間を〃実行時間〃と呼ぶ。また，部分解を列挙した回数を“ステップ数”と呼び，これを列挙木の拡大の尺度とする。

実験では，3.2で述べた5通りの分枝変数の選択方法と3.3で述べた4通りの値の決定方法を組み合せた20通りの分枝操作の方法を考える。このうち(1)－(ロ)と(2)－(イ)の2つの組み合わせはTrotterとShettyが用いたもので，これら2つの組み合わせと他の18の組み合わせとの比較を，探索終了までのステップ数と実行時間に着目し，以下の2種類のランダム問題とシステム設計を定式化した問題[4]を使って行う。

第1のランダム問題はLinとRardinの方法[6]により発生させる問題である。この方法は，まず上下限制約付きランダムＬＰ問題を発生させ，次にこの問題の最適解を丸めた解が実行可能になるようにＬＰ問題の係数ベクトルを修正して，上下限制約付き全整数計画問題を発生するものである。この問題は実行可能解を持ち，かつ，問題の制約領域は有限であるので最適解の存在が保証される。本実験では，10変数5制約式，15変数5制約式そして15変数10制約式の問題をそれぞれ3題づつ発生させて用いる。

第2のランダム問題は，Trotterと Shettyが数値実験で用いている問題である。その発生方法は，問題(P)の c_j，a_{ij}，b_i，d_j を，

$$c_{\mathrm{L}} \leqq c_j \leqq c_{\mathrm{U}},\quad a_{\mathrm{L}} \leqq a_{ij} \leqq a_{\mathrm{U}},\quad b_{\mathrm{L}} \leqq b_i \leqq b_{\mathrm{U}},\quad d_{\mathrm{L}} \leqq d_j \leqq d_{\mathrm{U}} \tag{4.1}$$

を満足する範囲内で発生させた整数一様乱数値とするものである。本実験では，c_e，a_{ij}，b_i，d_j の値の上限値と下限値の組合せ（c_{L}，c_{U}，a_{L}，a_{U}，b_{L}，b_{U}，d_{L}，d_{U}）として（0，25，0，25，50，150，5，15）と（0，100，0，100，1000，2000，5，15）の2つを考え，それぞれの組み合わせに対して，30変数15制約式と30変数30制約式の問題をそれぞれ3題づつ発生させて用いる。

なお，実験はEPSON PC－286V上で，LOGITECH MODULA－2 ver3.10を用いて行う。

4.2 実験結果と考察

実験結果の中から，15変数5制約式のLinとRardinのランダム問題に対する結果の1つと，15変数30制約式のTrotterとShettyのランダム問題に対する結果の1つをそれぞれ表1，表2に示す。この結果，最適解を発見するまでのステップ数が小さく実行時間が最も短くなるような分枝変数の選択方法と値の決定方法の組み合わせは，解く問題により異なると考えられる。表1で

表1　15変数5制約式のLinとRardinのランダム問題に対する実験結果

分枝変数の決定方法と値の決定方法の組み合わせ	最適解を発見するまでのステップ数	探索を終了するまでのステップ数	実行時間 (sec)
(1)-(イ)	1151	1164	440
(1)-(ロ)	228	236	46
(1)-(ハ)	166	188	43
(1)-(ニ)	12	46	17
(2)-(イ)	220	225	44
(2)-(ロ)	140	141	31
(2)-(ハ)	70	77	22
(2)-(ニ)	42	48	20
(3)-(イ)	870	892	159
(3)-(ロ)	4917	4946	1034
(3)-(ハ)	320	349	76
(3)-(ニ)	320	349	76
(4)-(イ)	202	209	72
(4)-(ロ)	66	80	37
(4)-(ハ)	30	40	22
(4)-(ニ)	30	40	22
(5)-(イ)	400	409	152
(5)-(ロ)	11	34	17
(5)-(ハ)	507	521	93
(5)-(ニ)	507	521	93

表2　15変数30制約式のTrotterとShettyのランダム問題に対する実験結果

分枝変数の決定方法と値の決定方法の組み合わせ	最適解を発見するまでのステップ数	探索を終了するまでのステップ数	実行時間 (sec)
(1)-(イ)	9388	9413	4388
(1)-(ロ)	2385	2402	664
(1)-(ハ)	554	571	389
(1)-(ニ)	214	329	255
(2)-(イ)	9607	9664	1797
(2)-(ロ)	****[1)]	****	****
(2)-(ハ)	468	533	267
(2)-(ニ)	282	356	241
(3)-(イ)	****	****	****
(3)-(ロ)	113	186	195
(3)-(ハ)	489	566	277
(3)-(ニ)	7	126	164
(4)-(イ)	95	354	195
(4)-(ロ)	****	****	****
(4)-(ハ)	92	351	192
(4)-(ニ)	57	320	183
(5)-(イ)	45	306	199
(5)-(ロ)	375	436	233
(5)-(ハ)	48	346	197
(5)-(ニ)	31	328	195

注1）　"****"は65000ステップ以上探索しても探索を終了できない場合を表す

は(1)－(ニ)または(5)－(ロ)の組み合わせのとき実行時間が最も短くなっており，表2では(3)－(ニ)の組み合わせのときである。また，実行時間が最も長くなる組み合わせも問題によって異なると考えられる。これら2つの推測は，すべての実験結果を調べた結果さらに明かなものとなり，例えば(5)－(ロ)の組み合わせは，ある問題に対しては実行時間が最も短かったが，別のある問題に対しては最も長くなっていた。従って，実行時間が最も短くなる組み合わせを決定することは困難であると考えられる。しかし，今回の実験結果から，多くの問題に対して実行時間が短くなる組み合わせを挙げることが可能で，その組み合わせは(2)－(ニ)である。そこで，この組み合わせとTrotterとShettyが用いている2つの組み合わせの比較を行う。図2の(a)，(b)に比較のためのグラフを示す。このグラフより，(2)－(ニ)の組み合わせを用いた場合，問題22を除いてTrotterとShettyが用いている選択方法に比べて実行時間がほぼ等しいか短くなっていることがわかる。特に問題19，20では10倍以上高速に問題を解いている。一方，実行時間が長くなる問題22については，(1)－(ニ)の組み合わせを用いた場合に比べ約2倍の実行時間を要している。しかしこの場合も，実行時間は35秒であり，実用上問題のない時間と考えることができる。従って，(2)－(ニ)は，TrotterとShettyが用いている2つの組み合わせよりも良い組み合わせであるといえる。また，問題を変えた場合の実行時間のばらつきを調べた結果，(2)－(ニ)の組み合わせを用いた場合にばらつきが一番小さくなっていた。以上のことから，(2)－(ニ)は実用的には最良の方法と考えられる。

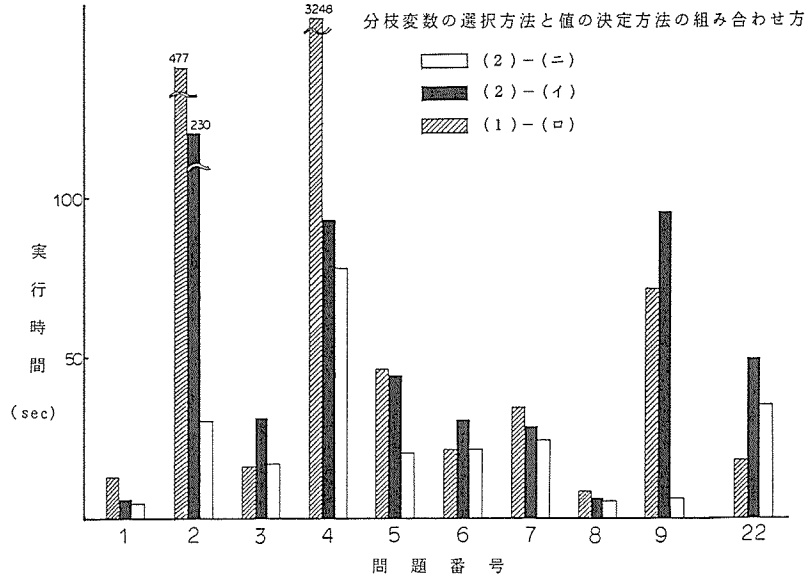
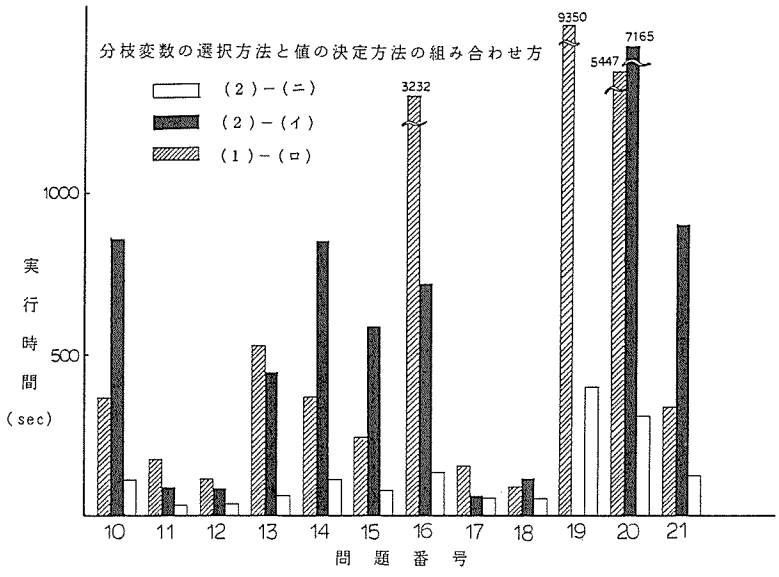


図2(a)　3種類の分枝操作の方法に対する実行時間の変化（その1）



注1）問題19に対して(2)−(イ)の組み合わせを用いた場合には、
65000ステップ以上探索しても探索を終了できなかった。

図2(b)　3種類の分枝操作の方法に対する実行時間の変化（その2）

5．おわりに

本論文では，まずTrotterとShettyの陰的列挙法で行われる分枝操作として，TrotterとShettyが用いた方法とは異なる3つの分枝変数の選択方法と2つの値の決定方法を提案した。そして，TrotterとShettyが用いた方法に加え，分枝操作の方法として5つの分枝変数の選択方法と4つの値の決定方法を組み合わせた20通りの方法を考え，最良と思われる方法を実験的に調べた。その結果，分枝変数の選択方法としては変数の上下限値に基づく方法とし，分枝変数の値の決定をＬＰ緩和解の成分を丸めた値とする方法が実用的に最も有効であることがわかった。

文　献

1）L. E. Trotter Jr. & C. M. Shetty : "An Algorithm for the Bounded Variable Integer Progremming Problem" : Journal of the Association for Computing Machinery, Vol. 21, No. 3, July 1974, pp. 505-513.
2）今野浩，鈴木久敏：“整数計画法と組み合わせ最適化”：日科技連 pp. 24-80
3）姉崎淳，大柳俊夫，加地郁夫："代理制約式を用いた陰的列挙法の効率解に関する数値実験"：北海道大学工学部研究報告 第140号（昭和63年）
4）D. R. Plane, C. McMillan Jr. : "整数計画法入門" 培風社 pp. 137-148
5）Fred Grover : "Surrogate Constraints" : Operations Reserch, Vol. 16 1968 pp. 714-749.
6）Benjamin W. Y. Lin & Ronald L. Rardin : " Development of a Paramrtric Generating Procedure for Integer Programming Test Problems" : Journal of the Association for Computing Machinery, Vol. 24, No. 3, July 1977, pp. 465-472.