整数計画問題に対する分枝カット法とカットの理論

藤江　哲也

1. はじめに

分枝カット法は，分枝限定法に切除平面法を組み込んだ解法である（これら解法の概略は次節で取り扱う）．近年，この分枝カット法の進歩によって，かなり規模の大きい整数計画問題が解けるようになっている．中には，小規模でありながら解くことが非常に難しい問題も存在する（例えば文献[8]）が，一般には数千～数万変数の問題が扱えるようになった．まず，このように進展してきた理由を考えることにする．

分枝限定法・分枝カット法の計算時間を，かなり大雑把に

【生成される子問題数】×【子問題を解く時間】

とする（厳密な議論については文献[12]を参照されたい）．近年，規模の大きい整数計画問題が解けるようになった背景に，ハードウェアの進歩を含む計算機環境の劇的な変化があったことはもちろんである．この変化は【子問題を解く時間】の減少に貢献し，よって規模の大きい問題が解けるようになったものと理解できる．また，特にここ数年来の線形計画ソルバの進歩も無視することはできない．整数計画問題に対する分枝限定法や分枝カット法の計算は，その大部分を線形計画問題を解くことに費やされ，線形計画問題が高速に解けるようになれば，その分だけ【子問題を解く時間】の減少に貢献する．線形計画ソルバの進歩については，例えば文献[6]に述べられている．文献[6]によると，かつては数時間～数日を要したが現在では数分あれば解けてしまう問題も少なくないようである．このような進歩が，大規模な整数計画問題を解くことにつながっているのも明らかであろう（実用的な線形計画問題の解法について，日本語で書かれた本として文献[30]がある）．

一方，例えば文献[16，第4章]は，分枝限定法のアルゴリズムとしての様々な工夫（技法）がまとめられている（その一部は次節で取り上げる）．これは【生成される子問題数】の減少に関する貢献とみることができる（親問題の情報を利用して子問題の線形計画緩和問題を解く，といった【子問題を解く時間】に関する技法もある）．1982年に刊行された文献[16]では，その当時までに提案された技法についてまとめられている．そしてその多くが，現在においても【生成される子問題数】の減少に重要な役割を演じており，特に近年の計算機環境や線形計画ソルバの進歩によって実用レベルでの重要性が認識されるようになった．線形計画ソルバの進歩は，文献[16，第6章]や文献[15]で取り上げられている切除平面法，特にカット生成が実用的になった要因にもなり，現在のような分枝カット法の進展につながったと言える．

別の見方をすると，整数計画ソルバのベースとなる技術は新しいものばかりではなく，文献[15]や文献[16]に既に現れているものも多いわけである．筆者の私見では，このことはあまり認識されていないようである．整数計画問題は一般にNP困難であり，【生成される子問題数】減少の技法について理論的な評価を行うことは難しく，実験的な評価が重要となる．しかし，技法が提案された当時と現在とでは計算機環境が大きく異なるため，これら技法の（特に，当時は有効でないとされた技法の）再評価があってもよいはずである．実際，現在成功をおさめているソルバは，これらの技法を積極的に取り入れている[1，7]．その一方で，文献[15，16]の刊行後，これら技法の改良・発展が続いている．そして，今後さらに【生成される子問題数】減少の技法が重要になると思われる．本稿では，分枝カット法の入門として，その概略を説明し，その中の重要な技法について紹介する．続いて，カットの理論を取り上げ，現在ソルバで使用されているカット（の一部）を紹介する．最後に，整数計画を扱った文献および整数計画ソルバを紹介する．最近の整数計画ソルバは利便性においても向上している（文献[1]や

ふじえ　てつや
神戸商科大学 商経学部
〒651-2197 神戸市西区学園西町8-2-1

4節に挙げるURLを参照のこと）ため，今後，整数計画問題の適用例・応用例がさらに増えることを期待したい．

2．整数計画問題と分枝限定法，分枝カット法

紙面の都合上，概略のみを説明する．詳細については，4節で紹介する文献を参考にされたい．

本節では，次の混合整数計画問題（mixed integer programming problem）を考える：

$$
\begin{array}{lll}
\text{(MIP)} & \text{最小化} & c^{\top}x \\
 & \text{条　件} & Ax \leq b, \\
 & & x_i \in \mathbb{Z} \quad (i=1,2,\cdots,p).
\end{array}
$$

ただし，決定変数 $x=(x_1,x_2,\cdots,x_n)^{\top}$ は n 次元ベクトルである．また，c は n 次元ベクトル，b は m 次元ベクトル，A は $m\times n$ 行列であり，p は $p\leq n$ を満たす整数である．整数変数 x_i の取り得る値が0または1の場合，すなわち $0\leq x_i\leq 1 (i=1,2,\cdots,p)$ が制約式に含まれているとき，(MIP) は0-1混合整数計画問題と呼ばれる．

ほとんどの整数計画ソルバで採用されている分枝限定法・分枝カット法では，(MIP) の線形計画緩和問題（以後LP緩和問題と呼ぶ）

$$
\begin{array}{lll}
\text{(LP)} & \text{最小化} & c^{\top}x \\
 & \text{条　件} & Ax \leq b
\end{array}
$$

を解く．(LP) の最適解を $\bar{x}$ とする．もし，$\bar{x}_1,\cdots,\bar{x}_p$ がすべて整数値をとれば，$\bar{x}$ は (MIP) の最適解となる．いま，このような状況ではないとする．このとき分枝限定法（branch-and-bound algorithm）では，整数でない $\bar{x}_j$ を選び，(MIP) に $x_j\leq\lfloor\bar{x}_j\rfloor$ という制約を加えた子問題（MIP_1）と，(MIP) に $x_j\geq\lceil\bar{x}_j\rceil$ という制約を加えた子問題（MIP_2）を生成する．一方，切除平面法（cutting plane algorithm）では，新しい不等式の追加を試みる．(MIP) のすべての実行可能解 x に対して $h^{\top}x\leq r$ が成り立つときこれを妥当不等式（valid inequality）と呼ぶ．また，妥当不等式 $h^{\top}x\leq r$ を加えることによって (LP) の領域が真に小さくなるとき，この不等式を切除平面またはカット（cutting plane, cut）と呼ぶ．切除平面法では，$\bar{x}$ を削除するようなカットを追加し，新しいLP緩和問題を作る．そしてそれを再び解く，という操作を繰り返す．分枝カット法（branch-and-cut algorithm）は，分枝限定法に切除平面法を組み込んだ解法であり，LP緩和問題（LP）を解いて終わるのではなく，カットを追加することにより（LP）の強化を試みる．カットの追加は，各子問題についても行われる．このとき，追加されるカットが，その子問題だけでなく元問題（MIP）に対して妥当なときglobalと呼ぶ．(MIP) に対して妥当な不等式はすべての子問題に対しても妥当であるため，globalなカットはカットプールに保持され，これ以降の子問題の計算で使われる．分枝カット法のプロトタイプは図1のようになるが，実装の際には，図1(a)〜(e)に対応した次の技法が重要となる．

(a) **前処理（Preprocessing）**：分枝カット法や分枝限定法を実行する前に，前処理を実行することがある．すなわち，冗長な変数/制約式の削除，変数の上下限制約の強化，LP緩和問題を強化するための係数行列 A の改善等である[28]．前処理は非常に効果的であることが知られている[1, 22]．文献[17]では，元問題だけでなく各子問題に対して前処理を行った効果について検証している．

(b) **子問題の選択（Node selection）**：深さ優先探索，幅優先探索，下界値優先探索，ヒューリスティック探索[22]等が知られている．また，ソルバによっては，これらを組み合わせた探索方法を実装している．

(c) **カット（Cut）**：LP緩和問題を強化する意味で，カットは重要である．これについては次節で取り

```
     分枝カット法;
     begin
(a)  (MIP) に対する前処理を実行;
       z* := ∞;
       (MIP) を子問題プール L に入れる;
       カットプールを空にする;
       while L ≠ ∅ do begin
(b)      子問題プール L から子問題 S を選択する;
         L から S を除去する;
         カットプールに含まれる不等式を S に追加する;
(c)      切除平面法を実行する;
(d)      if z* より良い実行可能解が見つかった then
           x* と z* を更新する;
(e)      if 新しい子問題が生成された then
           それらを子問題プール L に入れる
       end;
       return x*
     end.
```

図1　分枝カット法のプロトタイプ

扱う．

(d) **ヒューリスティックス** (Heuristics)：(MIP) に対する分枝カット法や分枝限定法では，実行可能解の更新は，LP 緩和問題の最適解が整数解となった場合にのみ行うのが一般的である．しかし，最近のソルバの中には，LP 緩和問題に基づくヒューリスティックスを組み込んでいるものもある．

(e) **分枝変数の選択** (Variable selection)：整数値をとらない $\bar{x}_j$ の選択によって，【生成される子問題数】は大きく変化する．最大実行不可能性ルール (maximum infeasibility rule) はテキストでよく目にする方法である．より現実的な方法として，文献[16，第 4 章]では，疑似コストルール (pseudo-cost rule) が紹介されている．また近年，x_j を選択したことによって生成される子問題を，そのLP 緩和問題に対して数回ピボット操作を実行することで評価する強分枝ルール (strong branching rule) が提案されている[2，14]．実験結果によると，一つの方法が他よりも優位になることはないが，疑似コストルールが一般に有効のようである[19，22]．

3．カットの理論

本節では，整数計画ソルバで使用されているカットの紹介を行う．そのすべてを紹介することはできないため，興味を持たれた方は，4 節で紹介する文献が参考になると思われる．カットは，問題の構造に依存しないものと依存するものに大別される．前者は「(MIP) の制約式 $Ax \leq b$」+「変数の整数性」からカットを生成するものであり，所与の不等式系 $Ax \leq b$ から作られる．一方，後者は (MIP) の実行可能解集合を明らかにするものであり，所与の不等式系 $Ax \leq b$ の形によらない不等式を生成する．なお，本節の内容の一部は文献[9]でも取り上げている．

3.1 問題の構造に依存しないカット

3.1.1 整数計画問題に対する Gomory カット

$P_{\mathrm{IP}}=\{x \in \mathbb{Z}^n \mid Ax \leq b, x \geq 0\}$ とする．ただし，A, b の成分はすべて整数であると仮定し，スラック変数 s を導入して $P'_{\mathrm{IP}}=\{(x, s) \in \mathbb{Z}^n \times \mathbb{Z}^m \mid Ax+s=b, x, s \geq 0\}$ とする．Gomory カットは，LP 緩和問題の最適基底解から生成されるが[15]，ここではより一般的な導出方法を紹介する．$u \in \mathbb{R}^m$ を任意のベクトルとする．このとき

$$u^\top Ax+u^\top s=u^\top b \tag{1}$$

が任意の $(x, s) \in P'_{\mathrm{IP}}$ に対して成り立つ．一般に，実数 α に対して $0 \leq \alpha-\lfloor\alpha\rfloor<1$ であり，また $x, s \geq 0$ であるから

$$(u^\top A-\lfloor u^\top A\rfloor)x+(u-\lfloor u\rfloor)^\top s \\ -(u^\top b-\lfloor u^\top b\rfloor)>-1 \tag{2}$$

が任意の $(x, s) \in P'_{\mathrm{IP}}$ に対して成り立つ．また，(1)式より

$$(u^\top A-\lfloor u^\top A\rfloor)x+(u-\lfloor u\rfloor)^\top s-(u^\top b-\lfloor u^\top b\rfloor) \\ =-\lfloor u^\top A\rfloor x-\lfloor u\rfloor^\top s+\lfloor u^\top b\rfloor$$

であり，$(x, s) \in P'_{\mathrm{IP}}$ に対して右辺は整数になるので左辺も整数でなければならない．よって(2)式より

$$(u^\top A-\lfloor u^\top A\rfloor)x+(u-\lfloor u\rfloor)^\top s \\ -(u^\top b-\lfloor u^\top b\rfloor) \geq 0 \tag{3}$$

は P'_{IP} に対する妥当不等式となることがわかる．s を消去すると

$$(\lfloor u^\top A\rfloor-\lfloor u\rfloor^\top A)x \leq \lfloor u^\top b\rfloor-\lfloor u\rfloor^\top b \tag{4}$$

は P_{IP} に対する妥当不等式になる．

LP 緩和問題の最適基底解の情報を使うと，必ずその解を削除するように u を決定することができる．Gomory の小数カット[10]は，このようにして構成されるカットである．

3.1.2 混合整数計画問題に対する Gomory カット

ここで紹介するカットの導出方法は，文献[26]に基づいている．(MIP) の連続変数を y と書くこととし，(MIP) の実行可能解集合が $P_{\mathrm{MIP}}=\{(x, y) \in \mathbb{Z}^p \times \mathbb{R}^n \mid Ax+Dy=b, x, y \geq 0\}$ として与えられているとする．ふたたび，$u \in \mathbb{R}^m$ を任意のベクトルとする．このとき

$$u^\top Ax+u^\top Dy=u^\top b$$

が任意の $(x, y) \in P_{\mathrm{MIP}}$ に対して成り立つ．$S \subseteq \{1, \cdots, p\}$ を x の添字部分集合とする．このとき

$$(u^\top A-\lfloor u^\top A\rfloor)x-\sum_{j \in S} x_j+u^\top Dy \\ -(u^\top b-\lfloor u^\top b\rfloor)=-\lfloor u^\top A\rfloor x-\sum_{j \in S} x_j+\lfloor u^\top b\rfloor \tag{5}$$

は $(x, y) \in P_{\mathrm{MIP}}$ に対して整数値をとる．よって，(5)式の左辺は 0 以上または -1 以下となる．簡単のため $f=u^\top A-\lfloor u^\top A\rfloor, f_0=u^\top b-\lfloor u^\top b\rfloor$ とおくと，このことは

$$\begin{cases} \sum_{j \in N-S} f_j x_j-\sum_{j \in S}(1-f_j)x_j+u^\top Dy \geq f_0 \\ \text{or} \\ \sum_{j \in N-S} f_j x_j-\sum_{j \in S}(1-f_j)x_j+u^\top Dy \leq -(1-f_0) \end{cases} \tag{6}$$

に等しい．ここで変数の非負条件などを考慮すると，

(6)式から

$$\begin{cases} \sum_{j\in N-S} f_j x_j + (u^\top D)^+ y \ge f_0 \\ \text{or} \\ -\sum_{j\in S}(1-f_j)x_j + (u^\top D)^- y \le -(1-f_0) \end{cases} \tag{7}$$

が導かれる．ただし，$(u^\top D)^+$ は $u^\top D$ の負成分を0に置き換えたベクトル，$(u^\top D)^-$ は $u^\top D$ の正成分を0に置き換えたベクトルである．つまり(7)式は(6)式を緩めたものである．(7)式は

$$\begin{cases} \sum_{j\in N-S} f_j x_j + (u^\top D)^+ y \ge f_0 \\ \text{or} \\ \frac{f_0}{1-f_0}\left(\sum_{j\in S}(1-f_j)x_j - (u^\top D)^- y\right) \ge f_0 \end{cases} \tag{8}$$

と変形できる．ここで二つの不等式には変数の重なりがないことに注意する．そこで，

$\alpha \ge 0$ かつ $\beta \ge 0$ であってしかも $\alpha \ge f$ or $\beta \ge f$ ならば，$\alpha + \beta \ge f$ が成り立つ

という事実を(8)式に適用することができ，

$$\sum_{j\in N-S} f_j x_j + \frac{f_0}{1-f_0}\sum_{j\in S}(1-f_j)x_j + (u^\top D)^+ y - \frac{f_0}{1-f_0}(u^\top D)^- y \ge f_0 \tag{9}$$

は P_{MIP} に対する妥当不等式となる．(9)式において，S を $S=\{j|f_j>f_0\}$ とおくのが最良である．Gomoryの混合整数カット[11]は，S をこのように選び，u をLP緩和問題の最適基底解に従って決定した不等式(9)である．

3.1.3　離接カット

LP緩和問題の実行可能領域を $P=\{x\in\mathbb{R}^n|Ax\le b\}$ とする．LP緩和問題の最適解を $\bar{x}$ とし，$\bar{x}_j$ が整数でないとする．このとき，$P_j := (P\cap\{x|x_j\le\lfloor\bar{x}_j\rfloor\})\cup(P\cap\{x|x_j\ge\lceil\bar{x}_j\rceil\})$ は $\bar{x}$ を含まない．よって，$\bar{x}$ を削除する P_j の妥当不等式が存在する．このようにして生成されるカットを離接カット（disjunctive cut）という．離接カットはBalasらによって1970年代から研究が行われている[3]．また近年，Balas-Ceria-Cornéjolsは0-1混合整数計画問題に対するlift-and-projectカットを提案し，これが離接カットと等価であることを示した[4, 5]（日本語による解説として文献[18]がある）．

3.1.4　MIRカット

まず，$P_1=\{(x,y)\in\mathbb{R}\times\mathbb{Z}|y\le b+x, x\ge 0\}$ を考える．ただし，b は整数でないとし，b の小数部分を $f=b-\lfloor b\rfloor$ とする．このとき，

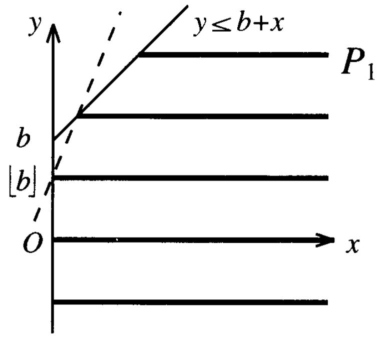

図2　カット(10)式

$$y \le \lfloor b\rfloor + \frac{x}{1-f} \tag{10}$$

は P_1 に対する妥当不等式（図2の点線）であり，点 $(0, b)$ を削除している（(10)式は点 $(0, b)$ を削除する離接カットと見ることもできる）．次に(10)式の導出方法を

$$P_{\mathrm{MIP}}=\{(x,y)\in\mathbb{R}\times\mathbb{Z}^n|\textstyle\sum_{j=1}^n a_j y_j\le b+x,\ x, y\ge 0\}$$

に対して適用する．$f=b-\lfloor b\rfloor$, $f_j=a_j-\lfloor a_j\rfloor (j=1,\cdots,n)$ とし，$S=\{j|f_j\ge f\}$ とおく．$\lfloor a_j\rfloor\le a_j$ であるから，

$$\sum_{j\in S} a_j y_j + \sum_{j\in N-S}\lfloor a_j\rfloor y_j \le b+x \tag{11}$$

が任意の $(x,y)\in P_{\mathrm{MIP}}$ に対して成り立つ．(11)式を

$$\sum_{j\in S}\lceil a_j\rceil y_j + \sum_{j\in N-S}\lfloor a_j\rfloor y_j \le b+x+\sum_{j\in S}(1-f_j)y_j$$

と変形して，(10)式の導出方法を適用すると

$$\sum_{j\in S}\lceil a_j\rceil y_j + \sum_{j\in N-S}\lfloor a_j\rfloor y_j \le \lfloor b\rfloor + \frac{x+\sum_{j\in S}(1-f_j)y_j}{1-f}$$

となる．これを整理すると

$$\sum_{j=1}^n\left(\lfloor a_j\rfloor + \frac{(f_j-f)^+}{1-f}\right)y_j \le \lfloor b\rfloor + \frac{x}{1-f} \tag{12}$$

が得られる．ただし，$(a)^+=\max\{a, 0\}$ である．(12)式をMIR（mixed integer rounding）不等式と呼ぶ．文献[20]では，多くの種類のカットがMIR不等式となることを示し，また数値実験によりMIR不等式の有効性を示している．

3.2　問題の構造に依存したカット

3.2.1　被覆不等式

まず，ナップサック問題の制約式を考える：

$$\sum_{j=1}^n a_j x_j \le b,\ x_j = 0 \text{ or } 1 (j=1,\cdots,n) \tag{13}$$

ただし，$a_j>0 (j=1,\cdots,n)$ である．(13)の実行可能解集合の凸包はナップサック多面体と呼ばれ，その不等式表現に関して多くの研究がある（詳細は文献[15, 18, 24, 31]等を参照されたい）．例えば，$C\subseteq\{1,\cdots,n\}$

に関して$\sum_{j\in C} a_j > b$となるとき，$\sum_{j\in C} x_j \leq |C|-1$はナップサック多面体の妥当不等式である．被覆不等式（cover inequality）と呼ばれる不等式は，この式を精密にしたものであり，さらにナップサック多面体の極大面を定めるための拡張も行われている．

もし（MIP）の制約の中に(13)式の形をしたものが含まれるならば，そこから得られる被覆不等式などを（MIP）のカットとして使うことができる．さらに，負の係数が含まれる場合，つまり

$$\sum_{j\in N_1} a_j x_j - \sum_{j\in N_2} a_j x_j \leq b,\ x_j = 0 \text{ or } 1 \quad (j\in N_1\cup N_2) \tag{14}$$

$(a_j>0(j\in N_1\cup N_2))$という場合にも，$j\in N_2$に関する$x_j$を$y_j=1-x_j$に置き換えれば(13)式の形になり，同様の議論ができる．

3.2.2 フロー被覆不等式

ここでは，次の集合を考える．

$$P_{\mathrm{MIP}} = \left\{ (x,y) \in \mathbb{Z}^n \times \mathbb{R}^n \;\middle|\; \begin{array}{l} \sum_{j=1}^n y_j \leq b, \\ 0 \leq y_j \leq a_j x_j (j=1,\cdots,n), \\ x_j \in \{0,1\} (j=1,\cdots,n) \end{array} \right\}.$$

被覆不等式と同様，$\sum_{j\in C} a_j > b$を満たす$C\subseteq\{1,\cdots,n\}$をとり，$\lambda=\sum_{j\in C} a_j - b > 0$とする．また，$\max_{j\in C} a_j > \lambda$であるとする．このとき，

$$\sum_{j\in C} (y_j + (a_j-\lambda)^+ (1-x_j)) \leq b \tag{15}$$

はP_{MIP}に対する妥当不等式である[25]，(15)式はフロー被覆不等式（flow cover inequality）と呼ばれるものの一種である．フロー被覆不等式はその後拡張され，カットとして広く利用されている．

4. おわりに

文献[9]と重複するが，最後に代表的な文献および整数計画ソルバの紹介をする．

整数計画を扱った本として文献[12, 15, 16, 18, 23, 24, 27, 29, 31]等がある．分枝カット法については文献[1, 18, 22, 31]が詳しい．また，カットの理論については新しいサーベイ論文[21]がある．次に整数計画ソルバをリストアップする．まず，商用ソフトウェアをアルファベット順に列挙すると，CPLEX (ILOG, Inc.)[1]，LINDO (LINDO Systems, Inc.)[2]，NUOPT (㈱数理システム)[3]，OSL (IBM)[4]，SOPT (㈱サイテック・ジャパン)[5]，Xpress-MP (Dash Optimization, Inc.)[6]等がある．また，フリーで入手できるソフトウェアとしてGLPK[7]，lp_solve[8]等がある．さらに，最近では分枝カット法およびその並列化に関するオープンソースプロジェクトCOIN (IBM)[9]が活発である．なお，整数計画ソルバ（商用/フリー）の比較をMittelmannのホームページ[10]で見ることができる．

[1] http://www.ilog.co.jp/products/cplex/
[2] http://www.lindo.com/
[3] http://www.msi.co.jp/nuopt/index.html
[4] http://www-3.ibm.com/software/data/bi/osl/index.html
[5] http://www.saitech-inc.com/
[6] http://www.dashoptimization.com/
[7] http://www.gnu.org/software/glpk/glpk.html
[8] ftp://ftp.es.ele.tue.nl/pub/lp_solve/
[9] http://www.coin-or.org/
[10] ftp://plato.asu.edu/pub/milpc.txt, ftp://plato.asu.edu/pub/milpf.txt

参考文献

[1] Atamtürk, L. and Savelsbergh, W. P.: Commercial integer programming software systems, Research report BCOL. 03.01, University of California at Berkeley, 2003.

[2] Applegate, D., Bixby, R., Chvâtal, V. and Cook, W.: Finding cuts in the TSP, Technical report 95-05, DIMACS, 1995.

[3] Balas, E.: Disjunctive programming, *Annals of Discrete Mathematics*, 5, 3-51, 1999.

[4] Balas, E., Ceria S. and Cornuéjols G.: A lift-and-project cutting plane algorithm for mixed 0-1 programs, *Mathematical Programming*, 58, 295-324, 1993.

[5] Balas, E., Ceria, S. and Cornuéjols, G.: Mixed 0-1 programming by lift-and-project in a branch-and-cut framework, *Management Science*, 42, 1229-1246, 1996.

[6] Bixby, R. E.: Solving real-world linear programs: A decade and more of progress, *Operations Research*, 50, 3-15, 2002.

[7] Bixby, R. E., Fenelon M., Guand Z., Rothberg E. and Wunderling R.: MIP: Theory and practice—Closing the gap, Technical Report, ILOG Inc., Paris, France, 1999.

[8] Cornuéjols, G. and Dawande, M.: A class of hard small integer programs, *INFORMS Journal of Computing*, 11, 205-210, 1999.

[9] 藤江哲也：“混合整数計画問題に対する分枝カット法”，計測と制御，42, 770-775, 2003.

[10] Gomory, R. E.: Outline of an algorithm for integer solutions to linear programs, *Bulletin of the American Mathematical Society*, 64, 275-278, 1958.

[11] Gomory, R. E.: An algorithm for the mixed integer problem. RM-2597, The Rand Corporation, 1960.
[12] 茨木俊秀: "組合せ最適化―分枝限定法を中心として―", 講座・数理計画法 8, 産業図書, 1981.
[13] 茨木俊秀, 福島雅夫: "最適化の手法", 情報数学講座 14, 共立出版, 1993.
[14] ILOG CPLEX 8.0 Reference Manual. ILOG Inc, 2002.
[15] 今野浩: "整数計画法", 講座・数理計画法 6, 産業図書, 1981.
[16] 今野浩, 鈴木久敏(編): "整数計画法と組合せ最適化", 日科技連出版社, 1982.
[17] 鴻池祐輔: "新技法の効果検証に有効なLPベース・MIPソルバーの試作", 東京農工大学大学院工学研究科電子情報工学専攻 修士論文, 2003.
[18] 久保幹雄, 田村明久, 松井知己(編): "応用数理計画ハンドブック", 朝倉書店, 2002.
[19] Linderoth, J. and Savelsbergh, M. W. P.: A computational study of search strategies for mixed integer programming, *INFORMS Journal on Computing*, **11**, 173-187, 1999.
[20] Marchand, H. and Wolsey, L.: Aggregation and mixed integer rounding to solve MIPs. *Operations Research*, **49**, 363-371, 2001.
[21] Marchand, H., Martin, A., Weismantel, R. and Wolsey, L.: Cutting planes and mixed integer programming, *Discrete Applied Mathematics*, **123**, 397-446, 2002.
[22] Martin, A.: General mixed integer programming: Computational issues for branch-and-cut algorithms. Computational Combinatorial Optimization (Jünger, M. and Naddef, D., eds.), *Lecture Notes in Computer Science*, **2241**, 1-25, 2001.
[23] Martin, R. K.: "Large Scale Linear and Integer Optimization―A Unified Approach", Kluwer Academic Publishers, 1999.
[24] Nemhauser, G. L. and Wolsey, L.: "Integer and Combinatorial Optimization", John Wiley & Sons, 1988.
[25] Padberg, M., Van Roy, T. J. and Wolsey, L. A.: Valid linear inequalities for fixed charge problems, *Operations Research*, **33**, 842-861, 1985.
[26] Padberg, M.: Classical cuts for mixed-integer programming and branch-and-cut, *Mathematical Methods of Operations Research*, **53**, 173-203, 2001.
[27] 坂和正敏: "離散システムの最適化 〈一目的から多目的へ〉", 森北出版, 2000.
[28] Savelsbergh, W. P.: Preprocessing and probing techniques for mixed integer programming problems, *ORSA Journal on Computing*, **6**, 445-454, 1994.
[29] Schrijver, A.: "Theory of linear and integer programming", John Wiley & Sons, 1986.
[30] 反町洋一(編): "線形計画法の実際", 講座・数理計画法 3, 産業図書, 1992.
[31] Wolsey, L.: "Integer Programming", John Wiley & Sons, 1998.