≪解 説≫

数理計画システムの開発

反 町 洋 一* 岡 本 吉 晴* 玉 井 哲 雄*

1. はじめに

情報処理振興事業協会の委託により，今度国産大型電子計算機を対象とした数理計画システムIMPS(IPA Mathematical Programming System) の開発を行なった．この種のシステムとしては IBM 社開発の MP-SX (Mathematical Programming System Extended) B&M 社の FMPS (Functional Mathematical Programming System) 等々いくつかの汎用システムが開発されているが，本システムは，機能，性能面でこれらの先進的な既存数理計画システムと競合できることを目標とし，さらに，これら類似システムと比較して，本システムの特徴として，今後の数理計画法の新しい適用分野である混合整数計画問題，及び大規模線型計画問題に対して，実効あるアルゴリズムの研究開発[1)9)]を行ない，国産では初めての本格的な数理計画システムを開発した．

IMPS は，最終的には FACOM 230/45 S, 55 にインプリメントしたが，システム設計は特定の電子計算機に依存せず行ない，次の 4 種のサブシステムから構成されている．このうち IMPS/LP が IMPS での基本サブシステムであり，他のサブシステムは，利用者の多様な要求に応じて，任意の構成が可能であるように設計されている．

- IMPS/LP　線型計画法，セパラブルプログラミングサブシステム
- IMPS/DMS　データ管理サブシステム
- IMPS/MIP　混合整数計画法サブシステム
- IMPS/EGUB　拡張 GUB 法サブシステム

IMPS での計算手順は，類似システムと同様に，FORTRAN に似たコントロール言語によって制御するが，これらについての詳細は省略して，本報告では IMPS の開発の主要な成果である IMPS/MIP, IMPS/EGUB サブシステムについて主に上記 IBM 社 MPSX との比較を中心に報告する．

2. 混合整数計画システム

* (株)三菱総合研究所ソフト開発部

IMPS には混合整数線形計画問題，すなわち『線形計画モデルを構成する変数の中で，そのとり得る値が整数値でなければならないという条件が付加されている変数（整数変数）を含む問題』を処理することが出来る IMPS/MIP サブシステムを含んでいる．

2.1 概要

我々は，IMPS/MIP を実用的なものとするために，この設計に際して，

（i） 汎用性―どの様なタイプの整数計画問題でも解くことができ，また，経験のない利用者でも簡単に使用できること．

（ii） 特殊性―個々の問題の構造によって，その問題に対する最良の解法戦略を，ある程度自由に利用者が選択でき，それによって高い計算効率をあげることができること．

（iii） 安定性―数値的な面および計算時間に関して，安定しており，利用者が安心して利用できること．

以上のような設計方針に基づいて，IMPS/MIP は，次に述べる理由から分枝限定法 (Branch-and-Bound Method) と R-最適法 (R-Optimal Method) の 2 つ解法を採用することにした．

分枝限定法は，非線形性の高い最適化問題，特に組み合せ論的な問題の解法として一般的なものである．混合整数計画問題に対しては，与えられた問題の整数変数のバウンド（上限値，下限値によってはさまれる区間）を逐次分割して，木構造的に副問題を構成し，それぞれを最適化してゆくことによって，最終的にすべての実行可能な整数解を調べることになり，最適整数解を求めることができる．この方法は，他の類似システム（例えば，OPHELIE, MPSX, UMPIRE 等）でも多く採用され，その実用性が実証されており，非常に強力な方法である．特に，最終的に最適解が得られること，また，与えられた問題の整数変数間の構造を分枝のルールに反映させることによって計算効率をあげることができること等の長所を有する．

しかしながら，解く問題の構造と分枝する戦略によって，打ち切られる分枝の状況が異なるので，同一の

規模の問題でも，計算に要する時間が著しく異なる場合が多いという欠点をもつ．

R-最適法は，ヒューリスティックな解法の一種で，現在の整数解に対して，同時にR個までの整数変数の値を変えて，でき得る限り目的関数値を改善し，もはや改善できなくなった時，その解を近似的な最適整数解と見なすという方法である．この方法は，他の類似システムでは採用されておらず，混合整数計画問題に対して数学的最適解より，むしろ実用的な近似最適解を得ようとする立場であり，IMPS の他類似システムと比較して大きな特徴の一つとなっている．この方法は，得られた解が，あくまで最適解に対する近似解でしかないという欠点を持つが，整数解探索の回数が，l^R のオーダーであり，分枝限定法のそれが高々 $(G+1)^l$ という l に関して指数関数的であるのに対して，非常に小さいこと（l は整数変数の数，Gは整数関数の上下限値の幅）と，同一の規模の問題に対する計算時間が安定しているという長所を持っている．また，いくつかの数値実験（後述）から，R-最適解は，$R=2$ 程度で，最適解の非常に良い近似解となっていることが，実験的に言えている．

IMPS/MIP は，以上のような2つの互いにそれぞれの欠点を補完する方法を採用することによって，先にあげた我々の基本設計方針を満足する混合整数計画システムになっている．

2.2 アルゴリズムの概説

IMPS/MIP に採用した2つの整数解探索（分枝限定法，R-最適法）のアルゴリズムについてその概要を説明する．

2.2.1 分枝限定法

分枝限定法の計算は，次のように行なわれる．

（1） 連続線形計画問題の最適実行可能解の計算

整数制約をとり除いた問題の最適解（連続最適実行可能解）をノード1とする．その解が整数解ならば，終了する．そうでなければ，ノード1を分枝ノードKとする．

（2） 分割変数の決定

分枝ノード K の分割変数 y_S を選び，2つの副問題 $K+1$ と $K+2$ を構成する．（K は，今までに作られたノードの数）．この2つの副問題は，ノード K に対応する副問題 K の整数変数 y_S のバウンドを，$L_K^S \leqq y_S \leqq [V_S]$ と $[V_S]+1 \leqq y_S \leqq U_K^S$ にかえた問題である．但し，V_S はノード K の整数変数 y_S の値，L_K^S と U_K^S は，副問題 K の整数変数 y_S の上限値と下限値，[ ] はガウス記号である．

（3） 副問題の最適化

副問題 $K+i\,(i=1,2)$ を最適化する．解が実行不可能，または，目的関数値が，その限定値よりも悪い場合は，ノード $K+i$ は作られないが，それ以外の場合は，整数解ノード，または，未処理ノードとなる．整数解ノードが得られれば，目的関数値の限定値が更新される．

（4） 分枝ノードの選定

未処理ノードのうちから，次の分枝ノードKを選定して，（2）へ行く．未処理ノードが無ければ，最適整数解が得られて（整数解ノードが得られている場合），または，実行不可能（整数解ノードが得られていない場合）であるとして終了する．

2.2.2 R-最適法

整数変数を $y=(y_1, y_2, \cdots y_l)$ とし，現在の整数解の整数変数の値 $\bar{y}_1, \cdots \bar{y}_l$ を要素とする l-ベクトル $\bar{y}$ を

$\bar{y}=(\bar{y}_1, \bar{y}_2, \cdots \bar{y}_l)$

とする時，$\bar{y}$ の R-近傍を，

$N_R(\bar{y})=\{R$ 個の要素だけが，$\bar{y}$ の要素と，$+1$ または -1 だけ異なっている l-ベクトルの集合$\}$

とする．R-最適法は，初期整数解 $\bar{y}$ が与えられて，$\bar{y}$ の R-近傍 $N_R(\bar{y})$ に属する整数解を調べて，目的関数値の限定値（この値は，通常，現在の整数解の目的関数値とされている）よりも悪くない実行可能整数解を捜してゆく方法である．IMPS/MIP では，$R=1$ および2として，整数解探索を行なう機能を有している．

R-最適法の計算は，次のような手順で行なわれる．

（1） 初期整数解 $\bar{y}$ を与える．

（2） $\Omega=N_R(\bar{y})$ とする．

（3） Ω に属する1つの整数解 w を取り出す．

（4） $y=w$ として，連続変数に関して最適化し，この解が，現在の目的関数の限定値よりも悪くなければ，$\bar{y}=w$ として，（2）へ行く．そうでなければ，$\Omega=\Omega-\{w\}$ とする．

（5） $\Omega=\phi$ であれば，現在の整数解 $\bar{y}$ は，R-最適解であるとして，終了し，そうでなければ，（3）へ行く．

2.3 機能

IMPS/MIP は，IMPS の制御言語を用いて使用することができ，次のようなプロシージャが設けられている．

(1) STARTMIX

プロシージャ BBMIX および HEUMIX による整数解探索の計算を行なうために必要な初期化を実行する．また，保存された BBMIX の整数解探索の情報を回復し，そこから BBMIX の計算を再開することができるようにする．

(2) BBMIX

分枝限定法による整数解探索を行なう．

(3) HEUMIX

R-最適法による整数解探索を行なう．

(4) SAVEMIX

プロシージャ BBMIX による整数解探索の途中の情報を問題ファイルに保存する．

以上の4つのプロシージャを用いて，IMPS の制御言語で記述された3つのシステム・マクロ，

(5) OPTMIX 1, OPTMIX 2, OPTMIX 3

が用意されており，経験のない利用者でも，簡単に IMPS/MIP を利用することができるようになっている．

以上のプロシージャやシステム・マクロには，整数解探索の計算戦略に関する種々のオプションが設けられており，利用者は限られた計算費用で，与えられた混合整数計画問題に適した計算を行なうことができるようになっている．

IMPS/MIP では，計算戦略選択について次のようなオプションが設けられている．

＜分枝限定法のオプション＞

(i) 分割変数の選択に関して

利用者の与えた優先順位，整数変数の罰金コスト，利用者の与えた階層木構造，等に従って分割変数を選択してゆく方法がある．特に，階層木構造を用いることによって，整数変数間に構造的な関係のあるモデルに対して，非常に効果をあげることができる．

(ii) 分枝ノードの選択法に関して

分枝ノードは，未処理ノードのうちから，

a．最もノード番号の大きいもの

b．最も目的関数値が良いもの

c．最もノード評価値（そのノードから分枝して得られる整数解の目的関数値の予測値）の良いもの

d．最もノード木構造上で深い位置にあるもの

等の選択法に従って，選ばれる．

(iii) 第1分枝の選択に関して

分枝ノードより作られる2つの副問題のどちらを先に最適化すべきかの評価方法に関するオプションが与えられている．

＜*R*-最適法のオプション＞

HEUMIX は，STARTMIX で与えられた初期整数解に対しても，BBMIX で得られた整数解に対しても *R*-最適法を適用することができる．

(i) 整数変数の優先順位は，

a．限界コスト d_j の大きさを基準とするもの

b．利用者の与えた優先順位を基準とするもの

(ii) 計算戦略に関しては，

a．1-最適法だけを行なうもの

b．2-最適法だけを行なうもの

c．1-最適法を行なった後，2-最適法を行なうもの

d．1-最適解に対して 2-最適法で 1つでも改善された解が見つかったら，また 1-最適法を行ない，1-最適かつ 2-最適解を求めるもの

e．1-最適法と 2-最適法を交互に適用し，1-最適かつ 2-最適解を求めるもの

等がある．

＜整数セットの使用＞

整数変数をいくつかのセットに分割して，ある整数セットの整数変数に対してだけ分枝限定法や *R*-最適法を適用したり，いくつかの整数セットの整数変数の値を固定して，他の整数セットに対して整数解探索を行なったりすることによって，計算の効率を高めることができるようになっている．

2.4 数値実験結果

IMPS/MIP の開発に際して，分枝限定法の性能に関しては，多くの類似システムに採用され，十分に実証されているという理由から，特に *R*-最適法の有効性の検討に主眼を置くこととし，MPSX の分枝限定法のプログラムと，*R*-最適法のプログラムとの性能を比較するという形で，*R*-最適法の検討を行なうことにした．

まず，FORTRAN で，イン・コア処理の *R*-最適法のプログラムを作成し，それと MPSX とを比較することを行なった（表1）．この結果から，*R*-最適法による解法の有効性の次の検証がなされ，更にMPSX とプログラムを同じ条件にするために，I/O 処理が MPSX と同じものを用いた *R*-最適法のプログラムを作成した．この結果が表2である（使用機種は IBM 370/165，使用コア・サイズは 400 kB）．この数値実験より，*R*-最適法の有効性が実用規模の混合整数計

表 1

問題		$m*n$	l	計算時間(CPU 分)			目的関数値
				t_0	t_2	T	
I	MPSX	111*507	7	0.18	0.42	0.60	850430.
	R-Opt.			0.26	0.06	0.32	850430.
II	MPSX	165*426	12	0.11	0.97	1.08	1001287.
	R-Opt.			0.20	0.60	0.80	1001287.
III	MPSX	126*179	25	0.04	0.12	0.16	885.15
	R-Opt.			0.02	0.03	0.05	885.15
IV	MPSX	131*184	20	0.04	2.10	2.14	736.37
	R-Opt.			0.22	0.10	0.32	763.55
V	MPSX	18*78	36	0.02	0.21	0.23	1702.7
	R-Opt.			0.01	0.14	0.15	1715.6
VI	MPSX	67*114	28	0.02	0.05	0.07	224.13
	R-Opt.			0.01	0.10	0.11	224.13

表 2

問題		$m*n$ (密度)	l	計算時間 (CPU 秒)			最適性*	目的関数 値
				t_0	t_1	t_0		
VII	MPSX	72*1092	20	2.4	291.0	595.0	no	3244
	R-Opt.	(3.97%)		2.4	53.4	95.4	yes	3329
VIII	MPSX	92*1512	20	3.0	30.6	292.8	no	3149
	R-Opt.	(3.11%)		3.0	58.2	66.3	no	3052
IX	MPSX	72*1092	20	1.8	45.6	138.0	yes	2868
	R-Opt.	(3.97%)		1.8	34.8	64.7	yes	2868
X	MPSX	92*1512	20	2.4	77.4	294.0	no	2682
	R-Opt.	(3.11%)		2.4	69.0	110.4	yes	2666
XI	MPSX	197*498	85	4.2	32.4	294.0	no	458947
	R-Opt.	(1.43%)		4.2	32.4	105.0	yes	458947

* 表 2, 3 で, R-最適法採用のプログラムでは, R-最適の意味での最適性である.

表 3*

問題		$m*n$ (密度)	l	計算時間 (CPU 分)			最適性	目的関数 値
				t_0	t_1	t_2		
VII	BBMIX	72*1082	20	0.62	5.53	30.0	no	3244
	HEUMIX	(3.97%)		0.62	9.96	20.71	yes	3285
IX	BBMIX	72*1092	20	0.47	7.27	11.02**	yes	2868
	HEUMIX	(3.87%)		0.47	7.56	15.02	yes	2868
XII****	BBMIX	561*376	112	3.80	51.33	59.70	no	2236
	HEUMIX	(0.40%)		0.0	10.15	12.14	yes	2463
XIII	BBMIX	307*421	200	5.32	17.44	—***	no	187.5
	HEUMIX	(1.04%)		0.0	11.79	—	no	178.0

* この種の問題に対する FACOM 230-45 S と IBM 370/165 のハードウェア性能比は, 他の数値実験から約 10.0〜15.0 と推計した.

** 表 2, 問題IXのケースとは, 計算戦略オプションが異なっている.

*** 最初の整数解が得られた時点で計算を打切った.

**** 問題 XII, XIII について整数セット及び階層木構造を使用しない数値実験を FACOM 230-45 S で行なわなかったが, これは, これらの問題の整数変数の数が多いことから, 莫大な計算時間が想定されるためである.

画問題に対しても実証されたため, これを IMPS にインプリメントすることを決定したのである. 表 3 が IMPS/MIP での数値実験結果である (使用機種は FACOM 230/45 S, 使用コア・サイズは 160 kB である). 表 3 で, 問題 VII, IX は MPSX との比較のため, 問題 XII は HEUMIX での整数セットの使用の効果測定のため, 問題 XIII は BBMIX での階層木構造の使用の効果測定のためのものである. なお, 問題 XII, XIII で, HEUMIX の t_0 が 0.0 となっているのは, STARTMIX で初期整数解を与えて, 計算を行なったものである.

なお, 表 1〜3 で,

- $m*n$: 行の数*列の数
- l : 整数変数の数
- t_0: 連続最適解を求めるために要した時間
- t_1: 最良の整数解を得るまでの, 整数解探索に要した時間
- t_2: 整数解探索に要した総時間
- T: すべての計算に要した総時間
- 最適性: 最良の整数解の最適性が保証された場合は yes, そうでない場合は no
- 目的関数値: 最良の整数解の目的関数値

である. また, 問題 I 〜XI は最小化問題, 問題 XII と XIII は最大化問題である.

3. 拡張 GUB システム

IMPS には, 特殊な構造をもつ線型計画問題を効率よく処理する IMPS/EGUB システムと呼ばれるモジュールがある. この章ではこの EGUB システムについて述べる.

3.1 概要

IMPS/EGUB システムは, 特殊な構造をもつ大型の線型計画問題を効率よく解くためのシステムである.

最近の線型計画問題は, ますます大型のものを解くことが要求されてきている. それに伴って, 種々の数理計画システムにおいても, 大型の問題を取り扱えるよう工夫がなされており, IMPS/LP システムでも, 主記憶装置の有効利用等のいろいろな工夫によって制約式の約 2000〜3000 の問題が取り扱えるようになっている.

一方では, 問題の特殊な構造を利用した効率化によって, 大型問題を取り扱うという方向での研究もなされてきた. この分野では, 1960 年に出された G. B.

Dantzig と P. Wolfe による，いわゆる“分解原理”の手法が有名である[12]．

この手法が対象とする線型計画問題は，次のような構造をしている．

$$\min\ (\max)\ \sum_{i=1}^{p} c_i x_i,$$

$$\begin{pmatrix} A_0 & A_1 \cdots A_i \cdots A_p \\ & B_1 \\ & \ddots \\ & & B_i \\ & & & \ddots \\ & & & & B_p \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_i \\ \vdots \\ x_p \end{pmatrix} \leqq \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_i \\ \vdots \\ b_p \end{pmatrix}, \qquad (3\text{-}1)$$

$$0 \leqq x_i \leqq u_i.$$

ここで，A_0 は $m_0 \times n_0$ 行列，A_i は $m_0 \times n_i$ 行列，B_i は $m_i \times n_i$ 行列 $(i=1, \cdots p)$ であり，x_i は n_i 個の変数よりなる列ベクトル，c_i は n_i 個の定数よりなる行ベクトル，b_i は m_i 個の定数よりなる列ベクトル，u_i は n_i 個の（非負の）定数よりなる列ベクトル $(i=0, 1, \cdots p)$ である．

このような構造をもつ線型計画問題に対するアルゴリズムとしては，他に J.B. Rosen により 1964 年に発表されたもの[17]などがある．

Dantzig-Wolf による解法は，エレガントなものであるが，収束性などの面から，その後余り満足できる結果が得られていない．Rosen による方法にも，同じようなことがいえる．

これに対し，反復の進み方は通常の Simplex 法のそれと全く同じであるが，逆行列のもち方をブロック（ここでいうブロックとは，(3-1) 式で B_i とかいた制約式の行列の部分行列をいう）単位に分割するという工夫を加えることにより，逆行列を簡潔な形にすることによって，計算の手間を減らすという手法が開発された[15][16]．これは，拡張 GUB 法 (Extended Generalized Upper Bounding Technique) と呼ばれているが，その名の由来は，この手法が Dantzig などによる GUB 法 (Generalized Upper Bounding Technique)[11]の，ある意味での拡張とみなすこともできるからである．

我々の IMPS/EGUB システムが対象とする問題は (3-1)式の形をしたものであり，アルゴリズムとしては，この拡張 GUB 法によっている．もちろん，これを数理計画システムに加えるにあたって，種々の工夫を行なった．

このような構造を有する問題としては，多事業所を含む問題，多期間に渡る問題，不確定要素を含む問題などが知られている．この他，一般に大型問題は，係数行列における非零要素の割合が低く潜在的にこのような構造をもっているか，あるいは適当な行と列の並べかえによってこのような形に行列を配列できる場合が多い．この点を考慮して IMPS には，EGUB システムのサブシステムとして ARRANGE と呼ぶ，与えられた行と列とを適当に並べかえて，EGUB システムを適用できる形の問題につくりかえるプログラムを有している．

3.2 アルゴリズムの概説

拡張 GUB 法のアルゴリズムの詳細については，文献 14)，16) 等を参照されたい．

ここでは，逆行列をどういう形でもてばよいかということのみを，簡単に述べる．

ある基底解に対応する基底行列 B は，適当な列の並べかえにより，次のような形にできる．

$$B = \overbrace{\left(\begin{array}{c|c} \overbrace{F}^{m_0} & H \\ \hline J & G \end{array}\right)}^{m} \begin{array}{l} \}\, m_0 \\ \\ \end{array} \Big\}\, m$$

$$= \left(\begin{array}{cccc|ccccc} F_0 & F_1 & F_2 \cdots & F_k & H_1 \cdots\cdots & H_k & H_{k+1} \cdots & H_p \\ \hline J_1 & & & & G_1 & & & \\ & J_2 & & & & \ddots & & \\ & & \ddots & & & & \ddots & \\ & & & J_k & & & G_k & \\ & & & & & & G_{k+1} & \\ & & & & & & & \ddots \\ & & & & & & & G_p \end{array}\right). \qquad (3\text{-}2)$$

m は行の数．m_0, m_i は (3-1) 式の，A_0, A_i の行の数．これから，B の逆行列 B^{-1} は，

$$B^{-1} = \left(\begin{array}{c|c} I_0 & \\ \hline V & I_1 \end{array}\right) \left(\begin{array}{c|c} W^{-1} & -W^{-1}HG^{-1} \\ \hline & G^{-1} \end{array}\right).$$

ただし，

$W = F - HG^{-1}J$,

I_0, I_1 はそれぞれ $m_0 \times m_0$, $(m-m_0) \times (m-m_0)$ の単位行列，

$V = -G^{-1}J$

とかける．すなわち B^{-1} は W^{-1}, G^{-1}, V により決定されることになる．ここで G^{-1} は，各 G_i^{-1} の直和で表すことができる．

これにより，B^{-1} 全体の情報をもつより，W^{-1}, G^{-1}, V の情報により，はるかに簡約された形で，B^{-1} を保持できる．なお，W^{-1}, G^{-1} に対しては，積

型式の適用が可能である.

3.3 構成と機能

IMPS/EGUB システムは, IMPS/LP あるいは IMPS/MIP システムと異なり, FORTRAN 言語で書かれているために, これらのシステムとは独立したシステムとなっている.

しかし, 入力データの形式が IMPS/LP システムと同じであること, EGUB システムの出力した基底解を, LP システムの初期解設定のための入力データとして使えること等により両者の結合が考慮されている.

EGUB システムは, プロシージャと呼ぶ7つの主たるプログラム (サブルーチン) からなる. 各々の機能の概略は, 次の通りである.

(1) INPUT

入力データを読み込み, 係数行列に関する情報をファイルにつくる.

(2) START

与えられた問題の規模に応じて, 記憶域の配分を行ない, 初期基底の設定, その他の初期化を行なう.

(3) PRIMAL

与えられた問題の最適解を求める.

(4) INVERT

基底行列の逆行列をつくり直す.

(5) OUTPUT

解の状態を出力する.

(6) INSERT

利用者がデータで指定した基底を, 初期基底とする.

(7) BASIS

現在の基底解を, INSERT への入力データの形で出力する.

3.4 数値実験結果

IMPS/EGUB システムについては, 大型問題のデータが手近に不足していること等もあって, まだ充分な実験がなされているとはいえない.

次のような, 5つのモデルについて, 結果を得ているので表4に示す. 使用機種は, FACOM 230/55 である. 解法戦略は, 標準的なものを使用した.

CPU 時間は, PRIMAL 及び INVERT の総計時間で, INPUT, OUTPUT, START は含まない. 密度とは, 係数行列中の非零要素の占める割合である.

IMPS/EGUB の性能は, IMPS/LP システムとの比較で検討を行なっている. この結果を表5に示す.

表 4

	行の数	列の数	密度(%)	最大/最小	ブロック数	CPU 時間(分)
モデルA	99	512	3.34	MIN	5	9.93
モデルB	182	555	2.76	MIN	5	8.65
モデルC	388	3186	1.73	MIN	95	17.61
モデルD	501	440	0.68	MAX	5	6.21
モデルE	886	1603	0.27	MAX	5	35.63

表 5*

	IMPS/LP	IMPS/EGUB
モデルA	23.15	30.58
モデルB	61.94	29.57

* モデルA, モデルBと, (3-1) の様な構造を有する LP モデルのサイズが大きくなると共に, IMPS/EGUB が IMPS/LP に比べて効率が上がることが期待される.

使用機種は FACOM 230/45 S で, 時間は CPU 時間である.

4. おわりに

IMPS は, 約2年の期間で開発を行なったが, 本報告では紙面の都合から取りあげなかった IMPS/LP IMPS/DMS を含めて, ここで取りあげた実験結果や数多くの内部テスト, フィールド・テストを積み重ね, 最初に述べたように, 数理計画法の新しい適用分野に対して, 実用性の高いソフトウェアとして, 更に先進的な数理計画システムと競合できる, 国産では初めての本格的な数理計画システムを開発し得たと考えている. 今後, 本システムが多くの分野の問題に適用され, そこでの問題提起を中心として, 更に本システムの修正・改良を行なってゆきたいと念願している.

最後に長期間にわたって御指導御協力戴いた情報処理振興事業協会, 富士通株式会社, 並びに株式会社ユニバック総合研究所の各位に感謝の意を呈する次第である.

参考文献

1) 反町洋一, 武川博臣, 岡本吉晴: 混合整数計画システムについて―R-最適法の適用, 1973 年度春季研究発表会アブストラクト集, 日本 OR 学会, pp. 35-36 (1973).
2) M. Benichou et al.: Experiments in Mixed-Integer Linear Programming, Mathematical Programming, Vol. 1, pp. 26-94 (1971).
3) R. J. Dakin: A Tree-Search Algorithm for Mixed-Integer Programming Problems, Computer Journal, 8, 250-255 (1965).
4) F. Hillier: Efficient Heuristic Procedures for

Integer Linear Programming with an Interior, Operations Research, 17, 600-637(1969).

5) A.H. Land and A.G. Doig: An Automatic Method of Solving Discrete Programming Problems, Econometrica, 28, 497-520(1960).

6) B. Roy, R. Beuayonn and J. Terguy: From S.E.P. Procedure to the Mixed OPHELIE Program, Integer and Nonlinear Programming (J. Abadie, eds.), North-Holland Publishing Company (1969).

7) G. Sá: Branch-and-Bound and Approximate Solutions to the Capacitated Plant-Location Problems, Operation Research, 17, 1005-1016 (1969).

8) J.A. Tomlin: Branch and Bound Method for Integer and Non-Convex Programming, Integer and Nolinear Programming (J. Abadie, eds.), North-Holland Publishing Company, pp. 437-450 (1969).

9) 反町洋一, 武川博臣, 岡本吉晴：上限値付変数を含む拡張 GUB 法, 1972 年度秋季研究発表会アブストラクト集, 日本 OR 学会, pp. 65～66 (1972).

10) J.M. Benett: An Approach to Some Structured Linear Programming Problems, Operations Research, 14, 636-645 (1966).

11) G.B. Dantzig and R.M. Van Slyke: Generalized Upper Bounded Technique for Linear Programming, Journal of Computer and System Science, 1, 213-226 (1967).

12) G.B. Dantzig and P. Wolfe: Decomposition Principle for Linear Programs, Operations Research, 8, 101-111 (1960).

13) A.M. Geoffrion: Elements of Large-Scale Mathematical Programming Part I; Concepts and Part II; Synthesis of Algorithms and Bibliography, Management Science, 16, 652-675 (1970).

14) J.K. Hartman: A Primal Method for Linear Programs with Coupling Rows and Columns, Case Western Reserve University, O.R. Department, Technical Memorandom, No. 196 (1970).

15) R.N. Kaul: An Extension of Generalized Upper Bounded Techniques for Linear Programs, ORC-65-27, Operations Research Center, University of California, Berkeley (1965).

16) L.S. Lasdon: Optimization Theory for Large Systems, MacMillan Company, New York (1970).

17) J.B. Rosen: Primal Partition Programming for Block Diagonal Matrices, Numeriche Mathematik, 6, 250-264 (1964).

18) M. Sakarovitch and R. Saigal: An Extension of Generalized Upper Bounding Technique for Structured Linear Programs, SIAM Journal on Applied Mathematics, 15, 906-914 (1967).

(昭和 49 年 6 月 21 日受付)