

(研 究 資 料)

林業試験場電算機プログラミング報告 (4)

0-1 整 数 計 画 法

黒 川 泰 亨⁽¹⁾

Yasuaki KUROKAWA : Report on Computer Programming (4)

——0-1 Integer Programming——

(Research materials)

要 旨：線型計画問題において、すべての変数が0あるいは1の整数しかとらないという条件が附加されたとき、0-1整数計画問題をもつことになる。整数線型、あるいは整数非線型計画問題も0-1整数計画問題として組み直すことができることなどから、きわめて多くの経営上の問題が0-1計画問題として定式化できる。近年この0-1計画法が、オペレーションズリサーチの分野において注目され、経営経済に関する意志決定のための基本的道具として広く利用されるようになった。

本報告の目的は、(1)この0-1整数計画法のコンピュータプログラムを作成し、(2)その利用法を書式化する。そして(3)造林事業計画における造林方法採択の問題にたいする具体的な適用例と、その計算結果を示すことにある。

1. は じ め に

数理計画法は比較的新しい応用数学の1分科にもかかわらず、電子計算機の発達とともに急速に発展してきた。その応用される範囲も自然科学はもとより、社会科学にも広く活用され、実際的な効果をあげてきた。

電子計算機で数理計画問題の最適解を求める数学的方法是、探索法つまり1つの任意の解を出発点としてそれを反復手順によって逐次改良しながら最適解に到達するまで探索してゆくのが一般的である。ある1つの手順を用いることによって有限回のステップで最適解に到達できるとき、この手順をアルゴリズム algorithm と呼ぶが、1つの効果的なアルゴリズムを実行できるプログラムが開発されれば、電子計算機はもはやこのアルゴリズムの適用可能な計画問題にたいする最適化機械として使用することができる。

本報告は、農林省林業試験場に設置されている電子計算機 OKITAC-4500 システムを対象として開発した0-1整数計画法のプログラムとその使用方法を利用者に便利な形で書式化し、広く利用できるよう研究資料として発表するものであるが、同時にこれは先に発表した線型計画法のプログラム*1とともに今後すすめるようとしている林業経営計画問題を解く場合の有力な手段となるべきものである。また本報告は筆者の東京工業大学への国内留学における研究成果の1部として作成したものである。多くの指導を得た同大学工学部社会工学科鈴木光男教授ならびに中村健二郎助手に謝意を表する次第である。

*1 黒川泰亨：林業試験場電算機プログラミング報告 (2) —線型計画法— 林業試験場研究報告 No. 271, 1975

1975年10月9日受理

経 営—2 For. Manage.—2

(1) 経 営 部

2. 0-1 整数計画法の考え方

0-1 整数計画法 zero-one integer programming は通常の線型計画法においてすべての変数が 0 または 1 の整数しかとらないという条件が付加された場合である。線型計画法は可分性の仮定を前提とするが、この仮定は変数のとる値はいくらでも細分可能であるとするものである。しかし、問題によっては可分性の仮定が許されない場合がある。たとえば、最適解において機械を 1.4 台使用するという解は実行不能である。また、ある計画案を 0.3 だけ採用するというのも無意味である。かかる場合、通常は、四捨五入などの手続きによって実行可能にするが、たとえそれが可能解であるとしても変数が整数値であるという条件をつけた場合の最適解になっている保障はなく、一般にはたんなる近似解にとどまる。そこで、はじめから変数のとるべき値を指定して最適解を求める必要が生ずることになる。本報告で扱う 0-1 整数計画法は線型計画法において変数のとる値が 0 あるいは 1 の整数に限定された特殊な場合を問題にする。

0-1 整数計画法は巡回セールスマン問題、投資決定問題、配送問題、スケジューリング問題、プロジェクト選択問題など広い分野において多くの優れた適用事例が報告されている。また、林業経営の問題に関しても最適林道網計画や最適施業順序決定の問題などにたいして応用が試みられている^{*2}。この計画法のコンピューターに密着した直接的な方法は、MARKOWITZ, MANN, LAND, DOIG, KAREL, SWEENEY らによって試みられたが、これらは計算誤差にたいする考慮が不要なこととアルゴリズムが簡明なことで注目を浴びた。これらを最初に系統的にまとめたのが BALAS による Additive Algorithm である。この方法はその後、種々の簡明化、簡略化が行われるとともに、線型計画法の結果を積極的に利用して代替制約式を導入し、解の収束を早める方法などへ発展しつつある。BALAS の Additive Algorithm はインプリシット列挙法 implicit enumeration method に基礎を置くものであり、その原理的方法は次のごとくである。

いま、0-1 整数計画問題の一般形は次のように定式化することができる。

$$\left. \begin{array}{l} \text{minimize } \mathbf{c}\mathbf{x} \\ \text{subject to } \mathbf{b} + \mathbf{A}\mathbf{x} \geq 0 \\ x_j = 0 \text{ または } 1 \end{array} \right\}$$

ここで $\mathbf{c}$ は各要素が非負の n 次元行ベクトル、 $\mathbf{b}$ は m 次元列ベクトル、 $\mathbf{A}$ は $m \times n$ 次元行列、 $\mathbf{x}$ は 2 進変数列ベクトルとする。 $\mathbf{c} \geq 0$ であるという制約は、負の要素をもった変数 x_j について変数変換 $x_j' = 1 - x_j$ を行くと、 x_j が 0 あるいは 1 に制限されていることから式の一般性は失われない。 n 個の変数をもつ式は 2^n とおりの解の組合せが考えられ、そのすべてについて目的関数値と実行可能性を調べて最適実行可能解を求めることは有限回のステップで可能である。しかし、 n が少し大きくなると考慮すべき解の組合せは膨大になり、事実上計算不可能となる。たとえば、組合せの数は $n=20$ のとき $2^{20} \div 1.05 \times 10^6$ 、 $n=30$ のとき $2^{30} \div 1.07 \times 10^9$ 、 $n=40$ のとき $2^{40} \div 1.10 \times 10^{12}$ となる。

インプリシット列挙法は、基本的には上の方法を用いるが、そのいくつかの解をまとめた集合に対して目的関数値と実行可能性を調べることによってステップ数を大幅に削減するところに特徴がある。

いま、 n 個の変数の部分集合に 0 または 1 の値を固定したものを部分解と呼び S で表わす。 S に含まれ

^{*2} たとえば、南雲秀次郎：線型モデルによる収穫予定法の研究（I）——0-1 計画法の適用——，日本林学会誌 56 巻 4 号，1974 年 4 月，pp. 128~135

ない変数を自由変数といい、部分解 S にたいして自由変数が 0 か 1 の値をとるときに考えられる解の組合せを S の完結形 completion と呼ぶ。部分解 S の要素数を s とすると、 S の完結形は 2^{s-s} の要素からなる集合となる。この完結形が前述の 1 つの単位となり、この集合にたいして次の 3 条件が成立するかチェックする。

- (1) 部分解 S の完結形はいずれも実行可能解をもたない。
- (2) 部分解 S の完結形は今までに求められた実行可能解の最良目的関数値よりも改良された解をもたない。
- (3) 部分解 S の完結形の中に今までに求められた実行可能解の最良目的関数値よりも改良された解をもつ実行可能解がある。

以上 3 条件のいずれかが成立すれば部分解 S の完結形の各要素は暗黙的 implicit に調べられたことになる。これを部分解 S の完結形はファザム fathom されたという。もし、この 3 条件のうちいずれも成立しない場合は部分解 S の要素を 1 つ増し、上記 3 条件のチェックを行う。この部分解 S の要素を 1 つ増加させることをアーギュメント augment という。 $S = \phi$ (空集合) から出発してファザムされる部分解が出現するまでアーギュメントを行い、この反復によって 2^n とおりの解すべてについて調べ上げ最適解を求める。

ところで、この手順を行う際に生成した部分解の完結形がすでに調べ上げた解を含まないようにする必要がある。この重複を避ける方法としてアンダーライン under line の約束とバックトラック back truck の手法を導入する。部分解 S の i 番目の要素がアンダーラインされているというのは $i + 1$ 番目以降の要素を除き、 i 番目の要素の符号を逆転した部分解の完結形がファザムされたことを意味する。部分解 S がファザムされたとき、 S の最も右側のアンダーラインされていない要素の符号を逆転し、それにアンダーラインをつけ、それより右側の要素を S から除外したものを新規に部分解として採用する。これがバックトラックの手法である。この手法は片方のブランチ先に属する解が調べ上げられていない節までレベルを戻し、そのブランチ先を新たな部分解とすることを意味する。そして $S = \phi$ から出発して ① 3 条件のチェック、② アーギュメント、③ バックトラックの手順をくり返し $S = \phi$ に戻るまで行う。この方法によって少ないステップで 2^n とおりの解をもれなく重複なく調べ上げ、その中で最良目的関数値をとる可能解を最適解とする。

3. プログラムの内容

(1) プログラムの識別

プログラム名: IPK/45

作成者名: 黒川泰亨

作成年月日: 1975年7月15日

形式: 独立したプログラム

使用言語: FORTRAN (JIS 水準 7000)

(2) 目的

このプログラムは、一般的な 0-1 整数計画問題の最適解を BALAS の Additive Algorithm (加法算法) で求め、同時に解の分析に必要な一連の数値を計算する。

(3) プログラムの機能

このプログラムは、次に述べる計算処理機能をもつ、制約式の数 ≤ 250 、変数の数 ≤ 250 の 0-1 整数計画問題について、目的関数の係数、制約式の係数および制約量を読み込み最適解を求める。制約式の不等号の向きは右向き、左向きいずれでもよく、等号が含まれていてもよい。また最大化問題、最小化問題のいずれも扱うことができる。問題を解いた後で目的関数の係数、制約式の係数あるいは制約量を単独にまたは 2 つ以上同時に変化させた場合の解を求めることができる。計算の結果印刷されるものの 1 部は必要のない場合は省略できる。また、計算の進捗状態を知るために中間結果をコンソールタイプに打ち出すことができる。

4. 印刷結果の見方

このプログラムを使用して 0-1 整数計画問題を解いた場合の印刷結果と、その もつ意味を順を追って解説する (図 2 参照)。

(1) PROBLEM NAME

これは、これから最適解を求めようとする 0-1 整数計画問題の内容を表示するために利用者が任意に与えた標題である。これに続く 3 行にこの 0-1 整数計画問題が最大化問題であるか最小化問題であるかの区別と変数の数ならびに制約式の数印刷される。

(2) INPUT DATA MATRIX

ここに入力データつまり目的関数の係数、制約式の係数ならびに制約量が印刷される。すなわち、第 1 行の OBJECTIV FUNCTION に続いて目的関数の係数を印刷し、第 2 行めの CONSTRAINTS 以下に制約式を 1 本ずつ右向きに不等式、等式、左向き不等式の区別をしながらその係数と制約量を印刷する。ただし制約量以外のゼロ要素は印刷されない。

(3) MATRIX STATISTICS

ここでは入力データならびに計算の経過に関する情報が 5 項目印刷される。ELEMENTS は、データマトリックスの元の総数をあらわす、DATA DENSITY はデータマトリックスに占めるノンゼロ要素の密度を % で示す。ITERATIONS AT FEASIBLE は可能解を得るまでに要したイタレーション数である。ITERATIONS AT OPTIMAL は最適解に到達するまでに要したイタレーション数である。さらに ITERATIONS AT FINISH は計算が完了するまでに要したイタレーション数である。つまり ITERATIONS AT OPTIMAL と ITERATIONS AT FINISH の差が最後に求めた可能解の他に改良された可能解が存在しないことを計算的に確かめるのに要したイタレーション数となる。DATA DISTRIBUTION はデータマトリックスの元の数値 (ノンゼロ要素の絶対値) の分布状態を示す。すなわち、第 1 列は個数、第 2 列はノンゼロ要素に占める割合、第 3 列は全体に占める割合を % であらわす。

(4) OPTIMAL SOLUTION

ここに 0-1 整数計画問題の最適解に関する一連の数値が印刷される。OBJECTIV FUNCTION VALUE は目的関数の値である。SECTION (1)—COLUMNS は最適解のうち変数に関係するもので VALUE は各変数のとる値を示し、この値は 0 または 1 をとる。SECTION (2)—ROWS は最適解のうち行に関係するもので 3 項目の情報を表示する。VALUE は各制約式の移動水準をあらわす。SLACK VALUE は各制約式の制約量の残量を示す。ACT RATIO は各制約式の制約量にたいする移動量の比率

を表示する。つまりこの値が0に近い制約式ほど緩い制約であり逆に1に近づくほど強い制約になっていることを意味する。

5. データシートの書き方

(1) カードの種類

このプログラムを使用して0-1整数計画問題を解く場合には、次の8種類のデータカードをセットする必要がある。なお*印のものは不要の場合は省略できる(表4参照)。

- ① PROBLEM カード
- ② SIZE カード
- ③ DATA カード
- ④ 実際のデータカード
- ⑤ ENDATA カード
- ⑥ PARAMT カード*
- ⑦ パラメトリック問題のためのデータカード*
- ⑧ STOP カード

(2) 個々のカードの書き方

(I) は整数型=小数点を含まない数字, (R) は実数型=小数点を含む数字であり, 整数型はかならず指定されたカラムの右づめに書く。実数型は指定されたカラムの範囲内にあればどこに書いてもよい。以下この記号を使用する。なお, [] はフォーマットの形式を示す。

① PROBLEM カード [A7, 3X, 70A1]

データセットの先頭にくるカードは、つねに PROBLEM カードである。このカードは使用者のデータセット名を指定するカードで、これによって各データセットを区別する。そして問題の標題を兼用する。

1~7 カラム: PROBLEM と書く。

8~10 カラム: 空白。

11~80 カラム: 標題を書く。標題は任意の英数字, 特殊文字で構成する。標題の最初にかならず IPK 01 と書く。

② SIZE カード [A4, 1X, 3I5]

このカードで変数の数, 制約式の数および最大化問題か, 最小化問題かを指示する。

1~4 カラム: SIZE と書く。

5~10 カラム: 変数の数を書く (I)。

11~15 カラム: 制約式の数を書く (I)。

20 カラム: 最大化問題のとき 1, 最小化問題のとき 0 と書く。

③ DATA カード [A4, 76X]

このカードに続いて実際のデータを含むデータカードがくることを指示する。

1~4 カラム: DATA と書く。

5~80 カラム: 空白。

④ 実際のデータカード [FREE FORMAT]

データカードはまず目的関数の係数を記入し、続いて制約式 1 本ずつ技術係数と制約量を記入する。自由書式 free format 入力を採用しているため、とくにカラムの指定はない。目的関数の係数は変数番号 (I) とその係数 (R) を 1 つずつ対にして記入し、数字と数字との間に 1 カラム以上の空白を置く。カード 1 枚が一杯 (80 カラム) になれば改行する。すべての係数の記入が終了すれば必ず 0 と書きその後に 0. と書く。

制約式の技術係数の記入は目的関数に準じて行うが、すべての係数の記入が終了すれば 0 と書き、その後に制約量 (R) を書く。そして、制約量の後にこの制約式のもつ等号、不等号の別ならびに不等号の向きを指定する数字を書く。すなわち等号のときは 0、右向き不等号 ($\leq$) のときは -1、左向き不等号 ($\geq$) のときは 1 と書く。目的関数の係数ならびに制約式の係数はノンゼロ要素についてのみ書けばよく、また、その順序は自由である。第 1 カラムに * 印を書き 2~80 カラムに任意の英数字、特殊文字を書いたコメントカードあるいは 1~80 カラム空白のブランクカードをデータカードの間に適宜何枚挿入してもよい。

⑤ ENDATA カード [A6, 74X]

データカードがこれで終了することを指示する。

1~6 カラム: ENDATA と書く。

7~80 カラム: 空白。

⑥ PARAMT カード* [A6, 4X, 70A1]

パラメトリック問題を解く場合にだけ使用する。パラメトリック問題は、目的関数の係数、制約式の係数、制約量いづれについても扱うことができる。また、これらを同時に 2 つ以上実行することも可能である。このカードによってパラメトリック問題を実行することを指示し、そしてパラメトリック問題の標題を兼用する。

1~6 カラム: PARAMT と書く。

7~10 カラム: 空白。

11~80 カラム: パラメトリック問題のための標題を書く。標題は任意の英数字、特殊文字で構成する。

⑦ パラメトリック問題のデータカード* (FREE FORMAT)

最初に変更する数字の個数 (I) を書く。続いて変更する目的関数および制約式の係数番号 (I) とその係数のとる新しい値 (R) を書く。目的関数の係数を変更する場合は、最初に 0 と書き、その後に変数番号 (I) とその値 (R) を記入する。制約量を変更する場合は、最初に制約式番号 (I) を書き、そして変数番号に相当するところを 0 と書き、その後に新しい制約量 (R) を書く。80 カラム一杯で書き終ら

ない場合は改行し、第 1 カラムから順次同様に記入する。このカードの場合も自由書式 free format を採用しているためとくにカラムの指定はなく、数字と数字の間に 1 つ以上のブランクを置けばよい。

⑧ STOP カード [A4, 76X]

このカードによって計算の終了を指示する。

1~4 カラム: STOP と書く。

5~80 カラム: 空白。

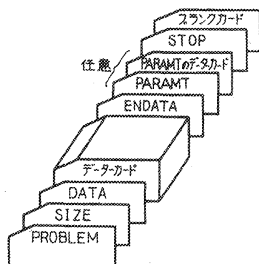


図 1. DATA DECK の構成

以上で入力カードの準備はすべて完了する。別のデータで計算を続行

する場合は STOP カードを入れずに PROBLEM カードから上述の順序に従ってセットすればよい。なお ① PROBLEM カードと ② SIZE カードの間にブランクカード(1~80 カラム空白のカード)を1杯挿入すれば INPUT DATA MATRIX の印刷は省略される。また ② SIZE カードと ③ DATA カードの間にブランクカードを1杯挿入すれば計算の進捗状況を表示する中間結果をコンソールタイプに出力させることができる。この中間結果は改良された可能解が見つかるごとにそのときのイタレーション数と解の値を印刷する。データデックの構成は図1のとおりである。

6. 適 用 例

前節で解説したデータシートの書き方に従い0-1整数計画問題を解きその結果を示す、次の例は造林事業計画に0-1整数計画法を適用したものであり、一般にプロジェクト選択問題といわれるものである。造林事業計画の立案に際し、計画者は限られた労働力や資金の範囲内において造林投資の優先順位や採用すべき造林技術を決定し、最善の計画を選択しなければならない。いままで、かかる場合の意志決定には明確な基準がないまま計画者の勘と経験によって試行錯誤的に行われることが多かった。したがって、計画者が異なれば採択される計画が異なり、きわめて主観的なものであった。この意志決定の過程を合理的に行うためには、各林分ごとに可能な造林技術を定め、それぞれの技術を採用した場合に期待される利益を客観的に評価したうえで与えられた諸資源の許容範囲内において、この利益額を最大にする造林事業計画を採択することが要求される。

造林技術は地ごしらえ方法、植栽方法および下刈方法の3つの指標を用いて各種のタイプに分類することができる。当然これらの各指標を細分すればするほど造林技術は多くのタイプに区分されるわけであるが、細かく区分された造林技術のタイプが、いわゆる造林の技術体系として意味があるかは疑問であるが、数理計画の考え方からすれば、技術的な投入産出関係の基礎資料が整っているものはこれを1つの技術として採用することに問題はない。ただ、現在のところ、育林施業に関する試験データの多くは短期的な効果を示しているにすぎず、伐期収穫と関係づけることが困難である。また、現在の収穫予定表などもきわめて平均的な性格の強いものである。したがって、情報の不足している部分は大胆な仮説のもとに推定せざるを得なくなり、技術的な投入産出関係の数字を多くの造林技術について明確にすることはきわめて困難である。

いま、ここで、話を具体化するために集約作業と粗放作業の2つの造林方法を考えてみる。これらの方法は東京営林局高萩営林署における過去の実績をもとに、一応確立しているとみられる常識的な造林方法のうち、投入量の比較的小さい方法を粗放作業、投入量の多い方法を集約作業として区別している。なお、ここにおける粗放作業は当署において一応成林可能と思われる必要最小投入量にほぼ近い作業を意味し、これと集約作業とは主として施肥の有無と下刈回数の多少によって区別されている。なお、これらの技術評価の詳細は p. 120 脚注*3 の pp. 79~87 を参照していただきたい。

さて、造林計画において造林の対象となる林分があり、この林分にたいして集約作業ならびに粗放作業の2つの造林方法のいずれの方法によっても造林が可能であるとする。この場合、集約作業で造林すれば、粗放作業に比較して林木の成長もよく、高い収益が期待できる。しかし、集約作業で造林すれば、粗放作業に比較して当然多くの労働力や費用を必要とする。利用可能労働量や総投入費用にたいする制約が全然なければ、すべての林分を集約作業によって実行する計画がもっとも良く、計画は単純に決定され

表 1. 造林計画に関する基礎資料

樹 種		ス ギ						ヒ ノ キ				マ ツ	
地 位		1		2		3		1		2		2	
造 林 方 式		粗放	集約	粗放	集約	粗放	集約	粗放	集約	粗放	集約	粗放	集約
必要労働量 人日/ha	—1年	17.8	36.0	17.8	36.0	17.5	35.0	17.8	36.0	17.5	35.0	18.2	37.0
	0	25.8	67.6	26.1	68.2	26.5	69.0	23.3	52.6	23.6	53.2	20.7	50.3
	1	4.9	25.6	5.2	22.4	5.6	23.2	4.9	21.8	5.1	22.2	4.9	21.8
	2	5.3	22.6	5.6	23.2	5.9	23.8	5.3	22.6	5.6	23.2	5.6	23.2
	3	5.4	10.8	5.7	11.4	5.8	11.6	5.7	11.4	5.9	11.8	5.9	11.8
	4	3.3	5.0	3.6	5.4	5.3	7.9	4.7	7.0	5.1	7.7		
必要造林費 千円/ha	—1年	18.5	37.0	18.5	37.0	18.2	36.4	18.5	37.0	18.2	36.4	18.8	37.6
	0	59.1	112.1	59.6	112.9	60.1	114.0	58.0	93.5	59.0	94.3	52.8	93.5
	1	6.8	35.6	7.2	36.0	7.7	37.4	6.7	35.4	7.5	36.0	6.8	35.6
	2	7.3	36.6	7.7	37.4	8.2	38.4	7.3	36.6	7.7	37.4	7.7	37.4
	3	7.5	15.0	7.8	15.6	8.0	16.0	7.8	15.6	8.2	16.4	8.2	16.4
	4	4.5	6.8	5.0	7.5	7.3	11.0	6.5	9.8	7.0	10.5		
割引純収益 千円/ha		917.0	1,359.0	685.0	1,006.0	482.0	691.0	708.0	1,086.0	467.0	694.0	128.0	131.0

る。しかし、現実には労働力や資金に制限があり、また、経営的観点や施業実行上の要請から採択すべき造林方法の組合せがあらかじめ決められている場合がある。そこで、必要労働量は利用可能労働量の範囲内におさめ、総投入費用は資金の許容範囲内におさめ、同時にあらかじめ決められた造林方法の組合せを満足させたうえで、これから期待できる利益を最大にする造林方法はいかなるものであるか、すなわち、どの林分をいかなる造林方法で造林するのが最適であるかについて、計画者は意志決定しなければならない。

表 1 は造林対象林分について樹種別、地位別、造林方式別に造林前年から 4 年後までの 5 年間に於いて ha 当たり必要とする各年次別の造林費ならびに必要労働量である。そして一番下の行はこれから期待できる伐期割引純収益である。

造林費の計算では、管理費や固定資産償却費など全体にかかる費用ないし固定費用的なものは除外し、変動費用的なものがとりあげられている。また、伐期割引純収益は（素材販売収入－伐出費）の現在価値－造林費の現在価値として計算されている。ここで用いられている素材販売収入は、収穫予想表などから得られた径級別品等別の素材材積にそれぞれの市場価格を乗じて合計したものであり、伐出費は、立木の径級分布を考慮して算出された。これを齢級ごとに算定し、収益曲線を求め、最適伐期とその時点での割引純収益を片対数グラフから読み取っている。なお割引利率として 3.5% が使用されている。賃金および素材価格は昭和 42 年度の数値である。また、伐出および造林の工期も同年度の実績が使われている。なお、これらのデータは東京営林局高萩営林署における数値である。割引純収益や造林費などが低く評価されているが、計画決定に関しては相対的な関係しか問題にならないので、絶対的な関係が計画決定に影響を及ぼすことはない*3。

ところで、この造林事業計画は次の 5 つの条件を満足していなければならない。

*3 これらの数字は、真柴孝司『国有林における造林事業の生産性について』、1969 年 12 月、林野庁、pp. 105～113 から得た。推計の詳細は同書を参照されたい。

(1) 各年次に要する総費用ならびに労働

力は、表2に示す額および量以上に使用できない。

(2) マツ2等地は、総費用ならびに労働

力に余裕がなければ造林する必要はないが、スギ3等地が粗放作業で造林される場合には必ず造林する。

(3) 施業上の必要からスギ2等地とヒノ

キ2等地は、同一の造林方法で造林する。

(4) 立地的条件から、もしスギ1等地ならびに2等地がともに集約作業で造林される場合にはスギ3等地も集約作業で造林する。

(5) 施業上の考慮から、スギ1等地とヒノキ1等地がともに集約作業で造林されることを避ける。

以上の条件を満足させ、かつ、表1に示された単位面積当りの係数を林分面積倍し、制約量と不等式ならびに等式の区別を記入して造林計画表を作成すれば、表3のごとくなる。

この計画問題(計画1)を解いた後で、パラメトリック問題として利用可能資金量ならびに利用可能労働量が表2の計画2、計画3のように変化した場合の最適解を計算する。表3にもとづき、これをデータシートに記入すれば表4のようになる。

計算結果は図2～4のとおりである。計算時間はデータカードの読み込み、ならびに計算結果の印刷をすべて含めて12分20秒であった。

さて、計画1の OPTIMAL SOLUTION で SECTION (1)—COLUMNS の VALUE 欄における各変数の値が1のものが、この造林計画において採択された造林方法である。つまり $x_1=1$, $x_4=1$, $x_5=1$, $x_8=1$, $x_{10}=1$, $x_{11}=1$ は、次のことを意味する。スギについては、地位1等地は粗放作業で、2等地は集約作業で、3等地は粗放作業でそれぞれ造林し、ヒノキについては、1等地、2等地とも集約作業で造林する。そしてマツについては、粗放作業で造林する計画が最適計画となる。このとき最大の割引純収益4,516.0万円が得られ、この値は図2の OBJECTIV FUNCTION VALUE に示されている。

SECTION (2)—ROWS の欄に注目すると、VALUE は各制約式の稼働水準を示す。SLACK VALUE は制約量からこの制約式の稼働水準を差し引いた値である。つまり、各資源の残量を示す。ACT RATIO は各制約式における制約量にたいする稼働量の比率、つまり、各資源の稼働率を意味する。したがって、この値が1に近づくほど制約量の余りが小さいことになる。いま計画1～計画3の最適解を一覧表にまとめれば、表5のようになる。この表において1のついた造林方法が各最適計画において採択された方法である。

表6は各最適計画における各資源の稼働量と稼働率を一表にまとめたものである。稼働率はその資源がどの程度使用されているかを示している。計画ごとに各資源の稼働率を比較することによって、その資源が利用される状況を動態的につかむことができる。

計画者は、制約量、制約式の係数あるいは目的関数の係数を単独にまたは2つ以上同時に変化させることによって計画モデルがあらわしているシステム内におこる変化が、最適解にいかなる影響を及ぼすかを調べる場合が多い。最適解の限界行動を分析する際にシステムが受ける変化は、その計画モデルがあらわ

表2. 造林計画における制約量

計画 番号	計画 1		計画 2		計画 3	
	労働量	資金	労働量	資金	労働量	資金
年次	人日	万円	人日	万円	人日	万円
—1年	1,650	175	1,650	175	1,650	175
0	2,550	500	2,400	450	2,250	400
1	950	160	750	160	600	160
2	900	160	900	160	900	160
3	550	80	550	80	550	80
4	300	65	300	65	300	65

表 3. 造林面積

樹種		スギ		ヒノキ				マツ				制約量					
地	種位	1		2		3		1		2		2		粗放	集約	粗放	集約
		粗放	集約	粗放	集約	粗放	集約	粗放	集約	粗放	集約	粗放	集約				
面積		10.2		14.9		6.8		10.5		8.4		2.4					
造林方式		1	2	3	4	5	6	7	8	9	10	11	12				
必要労働量 (人日)	- 1年	1	181.6	367.2	265.2	536.4	119.0	238.0	186.9	378.0	147.0	294.0	43.7	88.8	≦	1,650.0	
	0	2	263.2	689.5	388.9	1,016.2	180.2	469.2	244.7	552.3	198.2	446.9	49.7	120.7	≦	2,550.0	
	1	3	50.0	261.1	77.5	333.8	38.1	157.8	51.5	228.9	42.8	186.5	11.8	52.3	≦	950.0	
	2	4	54.1	230.5	83.4	345.7	40.1	161.8	55.7	237.3	47.0	194.9	13.4	55.7	≦	900.0	
	3	5	55.1	110.2	84.9	169.9	39.4	78.9	59.9	119.7	49.6	99.1	14.2	28.3	≦	550.0	
	4	6	33.7	51.0	53.6	80.5	36.0	53.7	49.4	73.5	42.8	64.7			≦	300.0	
必要造林費 (万円)	- 1年	7	18.9	37.7	27.6	55.1	12.4	24.8	19.4	38.9	15.3	30.6	4.5	9.0	≦	175.0	
	0	8	60.3	114.3	88.8	168.2	40.9	77.5	60.9	98.2	49.6	79.2	12.7	22.4	≦	500.0	
	1	9	6.9	36.3	10.7	53.6	5.2	25.4	7.4	37.2	6.3	30.2	1.6	8.5	≦	160.0	
	2	10	7.5	37.3	11.5	55.7	5.6	26.1	7.7	38.4	6.5	31.4	1.9	9.0	≦	160.0	
	3	11	7.7	15.3	11.6	23.2	5.4	10.9	8.2	16.4	6.9	13.8	2.0	3.9	≦	80.0	
	4	12	4.6	6.9	7.5	11.2	5.0	7.5	6.8	10.3	5.9	8.8			≦	65.0	
	13	1.0	1.0		1.0	1.0	1.0								=	1.0	
	14			1.0											=	1.0	
	15				1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	=	1.0	
	16														=	1.0	
	17														=	1.0	
	18														=	1.0	
	19			1.0											≦	1.0	
	20														≦	1.0	
	21														=	1.0	
	22		1.0			1.0									≦	1.0	
	23		1.0			1.0									≦	1.0	
割引純収益(万円)			935.3	1,386.2	1,020.7	1,498.9	327.8	469.9	743.4	1,140.3	392.3	583.0	30.7	31.4			

PROBLEM:

WRITTEN BY

PAGE

030

No.																								
1	PROBLEM	IPK01	SILVICULTURAL INVESTMENT PLANNING MO-1																					
2	SIZE	12	23	1																				
3	DATA																							
4	1	935.3	2	1386.2	3	1020.7	4	1498.9	5	327.8	6	469.9	7	743.4	8	1140.3	9	392.3						
5	10	583.1	11	30.7	12	31.4	0	0.																
6	1	181.6	2	367.2	3	265.2	4	536.4	5	179.	6	238.	7	186.9	8	378.	9	147.	10	294.				
7	11	43.7	12	88.8	0	1660.	-1																	
8	1	263.2	2	689.5	3	388.9	4	1016.2	5	180.2	6	469.2	7	244.7	8	552.3	9	196.2						
9	10	446.9	11	49.7	12	120.7	0	2850.	-1															
10	1	50.	2	261.1	3	77.5	4	333.8	5	38.1	6	157.8	7	57.5	8	228.9	9	42.8	10	186.5				
11	11	11.8	12	52.3	0	950.	-1																	
12	1	54.7	2	230.5	3	83.4	4	345.7	5	40.7	6	161.8	7	55.7	8	237.3	9	47.	10	194.9				
13	11	15.4	12	55.7	0	900.	-1																	
14											B8													
15	-1.	11.1	12.1	0.0.	1																			
16	31.9	-1.	0.0.	0																				
17	41.10	-1.	0.0.	0																				
18	21.4	1.6	-1.0	-1																				
19	21.8	1.0	1.1.	-1																				
20	ENDATA																							
21	PARAMT	PLANNING NO-2																						
22	3	2	0	2400.	3	0	750.	8	0	950.														
23	PARAMT	PLANNING NO-3																						
24	3	2	0	2350.	3	0	600.	8	0	900.														
25	STOP																							

図 2. 計画 1 の計算結果

..... 0-1 INTEGER PROGRAMMING

.. PROBLEM NAME .. IPK01 SILVICULTURAL INVESTMENT PLANNING NO-1

** IPK/45 **

FUNCTIONAL OBJECT : MAXIMIZE

NUMBER OF VARIABLES = 12
NUMBER OF CONSTRAINTS = 23

.. INPUT DATA MATRIX ..

OBJECTIV FUNCTION

1	935.299800	2	1386.200000	3	1020.700000	4	1498.900000	5	327.799800	6	469.899900
7	743.399900	8	1140.300000	9	392.299800	10	583.000000	11	30.700000	12	31.399900

CONSTRAINTS

ROW	--	1	2	3	4	5	6	7	8	9	10	11	12
1	181.600000	2	367.200000	3	265.200000	4	536.399900	5	119.000000	6	238.000000		
7	186.900000	8	378.000000	9	147.000000	10	294.000000	11	43.700000	12	88.799900		
<	1650.000000												

ROW -- 2

1	263.200000	2	689.500000	3	388.899900	4	1016.200000	5	180.200000	6	469.200000
7	244.700000	8	552.299800	9	198.200000	10	446.899900	11	49.700000	12	120.700000
<	2550.000000										

ROW -- 3

1	50.000000	2	261.099900	3	77.500000	4	333.799800	5	38.099900	6	157.800000
7	51.500000	8	228.900000	9	42.799900	10	186.500000	11	11.800000	12	52.299900
<	950.000000										

ROW -- 4

1	54.099900	2	230.500000	3	83.399900	4	345.700000	5	40.099900	6	161.800000
7	55.700000	8	237.300000	9	47.000000	10	194.900000	11	13.400000	12	55.700000
<	900.000000										

ROW -- 5

1	55.099900	2	110.200000	3	84.899900	4	169.900000	5	39.399900	6	78.899900
7	59.899900	8	119.700000	9	49.599900	10	99.099900	11	14.200000	12	28.299900
<	550.000000										

ROW -- 6

1	33.700000	2	51.000000	3	53.599900	4	80.500000	5	36.000000	6	53.700000
7	49.399900	8	73.500000	9	42.799900	10	64.700000				
<	300.000000										

ROW -- 7

1	18.899900	2	37.700000	3	27.599900	4	55.099900	5	12.400000	6	24.799900
7	19.399900	8	38.899900	9	15.300000	10	30.599900	11	4.500000	12	9.000000
<	175.000000										

ROW -- 8

1	60.299900	2	114.300000	3	88.799900	4	168.200000	5	40.899900	6	77.500000
7	60.899900	8	98.200000	9	49.599900	10	79.200000	11	12.700000	12	22.399900
<	500.000000										

図 2. (つづき)

.. MATRIX STATISTICS ..

ELEMENTS	=	276.	DATA	0.000	-----	1.000	=	24	14.63	8.70
DATA DENSITY	=	59.42	DISTRIBUTION	1.000	-----	10.000	=	28	17.07	10.14
ITERATIONS AT FEASIBLE	=	7		10.000	-----	100.000	=	76	46.34	27.54
ITERATIONS AT OPTIMAL	=	81		100.000	-----	1000.000	=	35	21.34	12.68
ITERATIONS AT FINISH	=	93		1000.000	-----		=	1	0.61	0.36

.. OPTIMAL SOLUTION ..

OBJECTIVE FUNCTION VALUE = 4515.9920000

SECTION (1) - COLUMNS

VALUE										
X(1) =	1.0000000	X(2) =	0.0000000	X(3) =	0.0000000	X(4) =	1.0000000			
X(5) =	1.0000000	X(6) =	0.0000000	X(7) =	0.0000000	X(8) =	1.0000000			
X(9) =	0.0000000	X(10) =	1.0000000	X(11) =	1.0000000	X(12) =	0.0000000			

SECTION (2) - ROWS

VALUE										
X(1001) =	1552.7000000	X(1002) =	2508.5000000	X(1003) =	849.0994000	X(1004) =	885.4993000			
X(1005) =	497.3997000	X(1006) =	288.3999000	X(1007) =	160.4000000	X(1008) =	459.4998000			
X(1009) =	134.7000000	X(1010) =	140.5000000	X(1011) =	68.4999700	X(1012) =	39.8999600			
X(1013) =	1.0000000	X(1014) =	1.0000000	X(1015) =	1.0000000	X(1016) =	1.0000000			
X(1017) =	1.0000000	X(1018) =	1.0000000	X(1019) =	0.0000000	X(1020) =	0.0000000			
X(1021) =	0.0000000	X(1022) =	1.0000000	X(1023) =	1.0000000					

SLACK VALUE										
X(1001) =	97.3002900	X(1002) =	41.5004900	X(1003) =	100.9006000	X(1004) =	14.5007300			
X(1005) =	52.6003400	X(1006) =	11.6001000	X(1007) =	14.6000400	X(1008) =	40.5002400			
X(1009) =	25.3000300	X(1010) =	19.5000300	X(1011) =	11.5000300	X(1012) =	25.1000400			
X(1013) =	0.0000000	X(1014) =	0.0000000	X(1015) =	0.0000000	X(1016) =	0.0000000			
X(1017) =	0.0000000	X(1018) =	0.0000000	X(1019) =	0.0000000	X(1020) =	0.0000000			
X(1021) =	0.0000000	X(1022) =	0.0000000	X(1023) =	0.0000000					

ACT RATIO										
X(1001) =	0.9410301	X(1002) =	0.9837252	X(1003) =	0.8937888	X(1004) =	0.9838880			
X(1005) =	0.9043630	X(1006) =	0.9613330	X(1007) =	0.9165712	X(1008) =	0.9189995			
X(1009) =	0.8418748	X(1010) =	0.8781248	X(1011) =	0.8562496	X(1012) =	0.8138456			
X(1013) =	1.0000000	X(1014) =	1.0000000	X(1015) =	1.0000000	X(1016) =	1.0000000			
X(1017) =	1.0000000	X(1018) =	1.0000000	X(1019) =	0.0000000	X(1020) =	0.0000000			
X(1021) =	0.0000000	X(1022) =	1.0000000	X(1023) =	1.0000000					

図 3. 計画 2 の計算結果

.. PARAMETRIC NAME .. PLANNING NO-2

** IPK/45 **

ROW -- 2 0 2400.000000
ROW -- 3 0 750.000000
ROW -- 8 0 450.000000

.. OPTIMAL SOLUTION ..

OBJECTIV FUNCTION VALUE = 4119.7930000

SECTION (1) - COLUMNS

VALUE						
X(1) =	1.0000000	X(2) =	0.0000000	X(3) =	0.0000000	X(4) =
X(5) =	1.0000000	X(6) =	0.0000000	X(7) =	1.0000000	X(8) =
X(9) =	0.0000000	X(10) =	1.0000000	X(11) =	0.0000000	X(12) =
						1.0000000

SECTION (2) - ROWS

VALUE						
X(1001) =	1406.6990000	X(1002) =	2271.9000000	X(1003) =	712.1995000	X(1004) =
X(1005) =	451.6995000	X(1006) =	254.2998000	X(1007) =	145.4000000	X(1008) =
X(1009) =	111.8000000	X(1010) =	118.9000000	X(1011) =	62.1999700	X(1012) =
X(1013) =	1.0000000	X(1014) =	1.0000000	X(1015) =	1.0000000	X(1016) =
X(1017) =	1.0000000	X(1018) =	1.0000000	X(1019) =	0.0000000	X(1020) =
						0.0000000
X(1021) =	0.0000000	X(1022) =	1.0000000	X(1023) =	0.0000000	

SLACK VALUE						
X(1001) =	243.3005000	X(1002) =	128.1003000	X(1003) =	37.8005400	X(1004) =
X(1005) =	98.3005400	X(1006) =	35.7002000	X(1007) =	29.6000400	X(1008) =
X(1009) =	48.2000300	X(1010) =	43.1000200	X(1011) =	17.8000300	X(1012) =
X(1013) =	0.0000000	X(1014) =	0.0000000	X(1015) =	0.0000000	X(1016) =
X(1017) =	0.0000000	X(1018) =	0.0000000	X(1019) =	0.0000000	X(1020) =
						0.0000000
X(1021) =	0.0000000	X(1022) =	0.0000000	X(1023) =	1.0000000	

ACT RATIO						
X(1001) =	0.8525451	X(1002) =	0.9466248	X(1003) =	0.9495993	X(1004) =
X(1005) =	0.8212717	X(1006) =	0.8809993	X(1007) =	0.8308569	X(1008) =
X(1009) =	0.6987498	X(1010) =	0.7306249	X(1011) =	0.7774996	X(1012) =
X(1013) =	1.0000000	X(1014) =	1.0000000	X(1015) =	1.0000000	X(1016) =
X(1017) =	1.0000000	X(1018) =	1.0000000	X(1019) =	0.0000000	X(1020) =
						0.0000000
X(1021) =	0.0000000	X(1022) =	1.0000000	X(1023) =	0.0000000	

図 4. 計画 3 の計算結果

```

** PARAMETRIC NAME **   PLANNING NO-3
ROW -- 2      0      2250.000000
ROW -- 3      0      500.000000
ROW -- 8      0      400.000000

** OPTIMAL SOLUTION **
OBJECTIVE FUNCTION VALUE      =      4012.4940000
SECTION (1) - COLUMNS
VALUE
X( 1) =      0.0000000    X( 2) =      1.0000000    X( 3) =      1.0000000    X( 4) =      0.0000000
X( 5) =      0.0000000    X( 6) =      1.0000000    X( 7) =      1.0000000    X( 8) =      0.0000000
X( 9) =      1.0000000    X(10) =      0.0000000    X(11) =      0.0000000    X(12) =      0.0000000

SECTION (2) - ROWS
VALUE
X(1001) =      1204.3000000    X(1002) =      1990.5000000    X(1003) =      590.6995000    X(1004) =      578.3997000
X(1005) =      383.4995000    X(1006) =      250.5000000    X(1007) =      124.8000000    X(1008) =      391.0996000
X(1009) =      86.0999600    X(1010) =      89.0999800    X(1011) =      52.8999800    X(1012) =      34.5999800
X(1013) =      1.0000000    X(1014) =      1.0000000    X(1015) =      1.0000000    X(1016) =      1.0000000
X(1017) =      1.0000000    X(1018) =      0.0000000    X(1019) =      0.0000000    X(1020) =      0.0000000
X(1021) =      0.0000000    X(1022) =      0.0000000    X(1023) =      1.0000000

SLACK VALUE
X(1001) =      445.7002000    X(1002) =      259.5002000    X(1003) =      9.3005370    X(1004) =      321.6003000
X(1005) =      166.5005000    X(1006) =      49.5002400    X(1007) =      50.2000400    X(1008) =      8.9003910
X(1009) =      73.9000400    X(1010) =      70.9000200    X(1011) =      27.1000200    X(1012) =      30.4000200
X(1013) =      0.0000000    X(1014) =      0.0000000    X(1015) =      0.0000000    X(1016) =      0.0000000
X(1017) =      0.0000000    X(1018) =      1.0000000    X(1019) =      0.0000000    X(1020) =      0.0000000
X(1021) =      0.0000000    X(1022) =      1.0000000    X(1023) =      0.0000000

ACT RATIO
X(1001) =      0.7298787    X(1002) =      0.8846665    X(1003) =      0.9844991    X(1004) =      0.6426663
X(1005) =      0.6972718    X(1006) =      0.8349999    X(1007) =      0.7131426    X(1008) =      0.8774000
X(1009) =      0.5381247    X(1010) =      0.5588748    X(1011) =      0.8612497    X(1012) =      0.8323073
X(1013) =      1.0000000    X(1014) =      1.0000000    X(1015) =      1.0000000    X(1016) =      1.0000000
X(1017) =      1.0000000    X(1018) =      0.0000000    X(1019) =      0.0000000    X(1020) =      0.0000000
X(1021) =      0.0000000    X(1022) =      0.0000000    X(1023) =      1.0000000
  
```

<<<< IPK/45 JOB COMPLETED >>>>

** IPK/45 **

表5. 最適造林計画

樹 種	地 位	面 積 ha	造林方式	計 画 1	計 画 2	計 画 3
ス ギ	1	10.2	粗 放	1	1	0
			集 約	0	0	1
	2	14.9	粗 放	0	0	1
			集 約	1	1	0
	3	6.8	粗 放	1	1	0
			集 約	0	0	1
ヒ ノ キ	1	10.5	粗 放	0	1	1
			集 約	1	0	0
	2	8.4	粗 放	0	0	1
			集 約	1	1	0
マ ツ	2	2.4	粗 放	1	0	0
			集 約	0	1	0

表6. 最適計画における資源稼働量と稼働率

計 画 番 号		計 画 1		計 画 2		計 画 3	
資 源	年 次	稼 働 量	稼 働 率	稼 働 量	稼 働 率	稼 働 量	稼 働 率
必 要 労 働 量	— 1年	人日 1,552.7	0.941	人日 1,406.7	0.853	人日 1,204.3	0.730
	0	2,508.5	0.984	2,271.9	0.947	1,990.5	0.885
	1	849.1	0.894	712.2	0.950	590.7	0.984
	2	885.5	0.984	746.2	0.829	578.4	0.643
	3	497.4	0.904	451.7	0.821	383.5	0.697
	4	288.4	0.961	264.3	0.881	250.5	0.835
必 要 造 林 費	— 1年	万円 160.4	0.917	万円 145.4	0.831	万円 124.8	0.713
	0	459.5	0.919	431.9	0.959	391.1	0.978
	1	134.7	0.842	111.8	0.699	86.1	0.538
	2	140.5	0.878	116.9	0.731	89.1	0.557
	3	68.5	0.856	62.2	0.777	52.9	0.661
	4	39.9	0.614	36.4	0.560	34.6	0.532
割引純収益 万円		4,516.0		4,119.8		4,012.5	

している現実のシステムが稼働している最中に起こる変化である場合もあり、また、基礎データのもつ不確実性の影響を調べるための仮想的な変化である場合もある。かかる方法を用いることによって、計画者は最終的な意志決定をする前に当面する計画問題についてさまざまな試行実験やシミュレーションをこころみることができる。本報告において制約量を変化させて計画1～計画3の最適解を評価したのもこのシミュレーションの1種である。

7. プログラムの構造

このプログラムは、1個の主プログラムと15個のサブルーチン副プログラムから構成されている。多くの副プログラムを使用する最大の理由は、主記憶装置の記憶容量が小さく、一度にすべてのプログラム

を主記憶装置の中に格納できないためである。そこで、プログラムを多くの副プログラムに分割し、オーバーレイ over-lay 構造の下に実行する。なお、オーバーレイの場合は、補助記憶装置からのアクセスが必要になるので計算速度が遅くなる。必要総メモリ語数は、オーバーレイ構造をとるとき 13,029 語 (1 語=16 bit)、オーバーレイ構造をとらないとき 24,358 語となる。主プログラムはサブルーチン副プログラムの呼び出しのみを実行し、実際の計算はすべてサブルーチン副プログラムで行う。

各サブルーチン副プログラムのもつ主要な機能を解説すれば次のようになる。なお () 内は必要メモリ語数を示す。

- (1) SUBS01 (935) : PROBLEM, SIZE, PARAMT カードなどを読み込み計算の準備をする。
- (2) SUBS02 (984) : 目的関数の係数を読み込み基礎ファイルに書き込む。
- (3) SUBS03 (1249) : 制約式の係数と制約量を読み込み基礎ファイルに書き込む。
- (4) SUBS04 (877) : パラメトリック問題のための変更された係数や制約量を読み込む。
- (5) SUBS05 (767) : 基礎ファイルから問題ファイルを作成する。
- (6) SUBS06 (998) : 部分解 S の完結形で乱された制約式があるかチェックする。
- (7) SUBS07 (1084) : 自由変数の中から部分解 S に入れてよい変数を探す。
- (8) SUBS08 (608) : 制約式の 2 進実行不可能性をチェックする。
- (9) SUBS09 (1613) : 制約式の不実行性を最小にする変数を選択し、アーギュメントを行う。
- (10) SUBS10 (478) : 改良された実行可能解があるかチェックする。
- (11) SUBS11 (348) : バックトラックの処理をする。
- (12) SUBS12 (273) : 部分解 S が空集合であるかチェックする。
- (13) SUBS13 (1177) : 入力データの統計量を計算し印刷する。
- (14) SUBS14 (3105) : 最適解を印刷の順序に整理し印刷する。
- (15) SUBS00 (1136) : フリーフォーマットで入力データを読み込む。

これらのソースプログラムは図 5 のとおりである。

8. 周辺機器構成

このプログラムの実行には次の周辺機器を必要とする。

- (1) カード読み込み装置
- (2) ラインプリンター (1 行 132 字)
- (3) 補助記憶装置 (DISC 1, DISC 2, DISC 3 の 3 つの論理名が使用できるもの)。

9. 注 意 事 項

計算を実行中、必要に応じて次のメッセージを印刷してその計算を中断し、新しい問題の処理を開始する。

- (1) NO FEASIBLE SOLUTION EXISTS : 解が不能である。
- (2) INPUT CARD IS INCORRECT : 入力データが誤って記入されたりパンチされている。

なお、このプログラムでは補助記憶装置のファイルにたいして DISC 1~DISC 3 の 3 個の論理名を使用しているが、この論理名が機種によって整数型に限定されている場合があるので注意する必要がある。

また, BACKSPACE 命令によって 1 レコード全部読み戻される場合はサブルーチン SUBS 04 の 12 行目を MD=1 に変更しなければならない。

10. 制 限 事 項

このプログラムで処理できる問題のサイズは制約式 ≤ 250 , 変数の数 ≤ 250 である。なお, 計算機の主記憶装置の記憶容量が大きい場合は, プログラム中の DIMENSION を大きくとることによって取り扱う問題のサイズをいくらでも大きくすることができる。

文 献

- 1) BALAS, E. : An Additive Algorithm for Solving Linear Programs with Zero-One Variables. Operation Research, vol. 13, (1965)
- 2) FRANTISEK, F. : A 449-Solution of Linear Programming Problems in 0-1 Variables. Communication of the ACM, vol. 16, (1973)
- 3) GLOVER, F. : A Multiphase-Dual Algorithm for the Zero-One Integer Programming Problems. Operation Research, vol. 13, (1965)
- 4) LAWLER, E. L. and M. D. BELL : A Method Solving Discrete Optimization Problems. Operation Research, vol. 14, (1966)
- 5) PETERSEN, C. C. : Computational Experience with Variants of the Balas Algorithm Applied to the Selection of R & D Project. Management Science, vol. 13, (1967)
- 6) MACMILLAN, Jr., C. : Mathematical Programming, An Introduction to the Design and Application of Optimal Decision Machines. John Wiley & Sons, Inc., (1970) (前田功雄訳: マクミラン数理計画入門 1, 2; 東京図書, 1970)
- 7) 茂木俊秀: 整数計画法 (3), オペレーションズ・リサーチ vol. 15, No. 11, (1970)

Report on Computer Programming (4)

—0-1 Integer Programming—

Yasuaki KUROKAWA⁽¹⁾

Summary

In general, programming problems are concerned with the allocation or efficient use of limited resources to meet desired objectives. These problems are characterized by the large number of solutions that satisfy the basic conditions of each problem. The selection of a particular solution as the best one to a problem depends upon some aim or over-all objective that is implied in the statement of the problem. A solution that satisfies both the conditions of the problem and the given objective becomes the optimal solution. Although there are many methods available for these programming problems, linear programming is the most popular method because of its wide applicability and high practicality. But the important classes of economic problems find their mathematical models in linear programs with integer variables. Prominent among these problems are those that correspond to linear programs with variables taking only one of the values 0 or 1.

The general form of a 0-1 integer program may be stated as follows :

Find x minimizing (maximizing)

$$cx,$$

subject to $Ax \leq b,$

$$x = 0 \text{ or } 1,$$

where x is a n -component row-vector, c is a given n -component row-vector, A is a given $m \times n$ matrix and b is a given m -component column-vector.

An algorithm for solving 0-1 integer program proposed by Egon BALAS starts by setting all the n variables equal to 0, and consists of a systematic procedure of successively assigning to certain variables the value 1, in such a way that after trying a (small) part of all the 2^n possible combinations, one obtains either an optimal solution, or evidence of the fact that no feasible solution exists. This algorithm is a search or enumeration method which examines branches of the combinatorial "tree" with the use of certain tests, to determine whether the branches are useless for further search (in which case they are abandoned and backtracking is performed) or are possible sources of improved, lower cost, feasible solutions (in which case the search continues along such branches). The only operations required under the algorithm are additions and subtractions; thus round-off errors are excluded. The FORTRAN computer program presented in this paper is proposed for solving 0-1 integer program by using the BALAS algorithm in its literal form and in reformed form at several parts. This program is divided into 15 sub-programs and reduces the storage demand and permits use of the much smaller computer (core capacity is 16 k wards, 1 ward is 16 bits). This program can solve the problem which has fewer than 250 variables and 250 constraints. In this paper a silvicultural investment problem which has 12 variables and 23 constraints is solved for example, and it takes only 4 minutes for one case.

Received October 9, 1975

(1) Forest Management Division

図 5. ソースプログラムリスト

OKITAC 4500 FORTRAN SOURCE PROGRAM LIST

PAGE 0001

LINE-NO	STATEMENT	
0001	C 0-1 INTEGER PROGRAMMING (PROGRAMED BY Y.KUROKAWA)	IPK01010
0002	C -----	IPK01020
0003	C MAIN PROGRAM	IPK01030
0004	C -----	IPK01040
0005	COMMON ROW,COL,OBJ,S,SS,Y,T,X,F,PM,IP,L,M,N,M1,N1,	IPK01050
0006	11S,MN,ND,IA,IB,IC,II,Z	IPK01060
0007	INTEGER S(251),SS(251),Y(251),T(251),X(251),F(251),	IPK01070
0008	IPM(251),IP(10)	IPK01080
0009	REAL ROW(251),COL(251),OBJ(251)	IPK01090
0010	10 II=1	IPK01100
0011	15 CALL SUBS01 (\$20)	IPK01110
0012	CALL SUBS02 (\$10)	IPK01120
0013	CALL SUBS03 (\$10,\$25)	IPK01130
0014	20 CALL SUBS04 (\$10)	IPK01140
0015	25 CALL SUBS05	IPK01150
0016	30 CALL SUBS06 (\$35)	IPK01160
0017	CALL SUBS07	IPK01170
0018	CALL SUBS08 (\$45)	IPK01180
0019	CALL SUBS09 (\$30)	IPK01190
0020	35 CALL SUBS10 (\$45)	IPK01200
0021	40 CALL SUBS11 (\$30)	IPK01210
0022	45 CALL SUBS12 (\$40)	IPK01220
0023	CALL SUBS13	IPK01230
0024	50 CALL SUBS14 (\$15)	IPK01240
0025	STOP	IPK01250
0026	END	IPK01260

図 5. (つづき)

OKITAC 4500 FORTRAN SOURCE PROGRAM LIST

PAGE 0001

LINE-NO	STATEMENT	
0001	C	IPK02010
0002	SUBROUTINE SUBS01 (*)	IPK02020
0003	C	IPK02030
0004	COMMON ROW,COL,OBJ,S,SS,Y,T,X,F,PM,IP,L,M,N,M1,N1,	IPK02040
0005	1IS,MN,MD,IA,IB,IC,II,Z	IPK02050
0006	INTEGER S(251),SS(251),Y(251),T(251),X(251),F(251),	IPK02060
0007	IPM(251),IP(10)	IPK02070
0008	REAL ROW(251),COL(251),OBJ(251),IDF(5)	IPK02080
0009	DATA IDF/4HPROB,4HSTOP,4HPARA,4HSIZE,4HDATA/	IPK02090
0010	15 READ(5,5010)HED,(S(J),J=1,70)	IPK02100
0011	5010 FORMAT(A4,6X,70A1)	IPK02110
0012	DO 100 K=1,II	IPK02120
0013	IF(HED.EQ.IDF(K))GO TO(10,30,23),K	IPK02130
0014	100 CONTINUE	IPK02140
0015	GO TO 15	IPK02150
0016	10 WRITE(6,6010)(S(J),J=1,70)	IPK02160
0017	6010 FORMAT(1H1,48X,32H..... 0-1 INTEGER PROGRAMMIN,	IPK02170
0018	11HG,3X,5H.....//1H0,18H.. PROBLEM NAME ...3X,70A1,	IPK02180
0019	220X,12H** IPK/45 **//1H0,5X,19HFUNCTIONAL OBJECT :)	IPK02190
0020	DO 110 J=1,10	IPK02200
0021	110 IP(J)= 0	IPK02210
0022	11 READ(5,5020)HED,N,M,MN	IPK02220
0023	5020 FORMAT(A4,1X,315)	IPK02230
0024	IF(HED.EQ.IDF(4))GO TO 12	IPK02240
0025	IP(1) =1	IPK02250
0026	GO TO 11	IPK02260
0027	12 MN1=MN+1	IPK02270
0028	GO TO(14,13),MN1	IPK02280
0029	13 WRITE(6,6020)	IPK02290
0030	6020 FORMAT(1H*,25X,8HMAXIMIZE)	IPK02300
0031	GO TO 20	IPK02310
0032	14 WRITE(6,6030)	IPK02320
0033	6030 FORMAT(1H*,25X,8HMINIMIZE)	IPK02330
0034	20 WRITE(6,6040)N,M	IPK02340
0035	6040 FORMAT(1H0,5X,19HNUMBER OF VARIABLES,4X,1H=,I4/1H ,	IPK02350
0036	15X,24HNUMBER OF CONSTRAINTS =,I4)	IPK02360
0037	21 READ(5,5030)HED,IP(8)	IPK02370
0038	5030 FORMAT(A4,73X,I3)	IPK02380
0039	IF(IP(8).GT.0)IP(2)=1	IPK02390
0040	IF(HED.EQ.IDF(5))GO TO 22	IPK02400
0041	IP(2) =1	IPK02410
0042	GO TO 21	IPK02420
0043	22 RETURN	IPK02430
0044	23 RETURN 1	IPK02440
0045	30 WRITE(6,6060)	IPK02450
0046	6060 FORMAT(///1H0,7X,	IPK02460
0047	137H <<<< IPK/45 JOB COMPLETED >>>>)	IPK02470
0048	STOP	IPK02480
0049	END	IPK02490

LINE-NO	STATEMENT	
0001	C	IPK03010
0002	SUBROUTINE SUB502 (*)	IPK03020
0003	C	IPK03030
0004	COMMON ROW,COL,OBJ,S,SS,Y,T,X,F,PM,IP,L,M,N,M1,N1,	IPK03040
0005	I15,MN,MD,IA,IB,IC,I1,Z	IPK03050
0006	INTEGER S(251),SS(251),Y(251),T(251),X(251),F(251),	IPK03060
0007	IPM(251),IP(10)	IPK03070
0008	REAL ROW(251),COL(251),OBJ(251)	IPK03080
0009	K=1	IPK03090
0010	M1=M+1	IPK03100
0011	N1=N+1	IPK03110
0012	REWIND DISC1	IPK03120
0013	IF(IP(1).EQ.1)GO TO 1	IPK03130
0014	WRITE(6,6010)	IPK03140
0015	6010 FORMAT(/1H0,23H.. INPUT DATA MATRIX ..	IPK03150
0016	1/1H0,3X,17H0BJECTIV FUNCTION)	IPK03160
0017	1 CALL SUB500(3,ID,D)	IPK03170
0018	100 CALL SUB500(1,SS(K),D)	IPK03180
0019	CALL SUB500(2,ID,COL(K))	IPK03190
0020	IF(SS(K).LT.0)GO TO 30	IPK03200
0021	IF(SS(K).FQ.0)GO TO 10	IPK03210
0022	K=K+1	IPK03220
0023	GO TO 100	IPK03230
0024	10 IF(COL(K).NE.0.)GO TO 30	IPK03240
0025	IF(IP(1).EQ.1)GO TO 20	IPK03250
0026	JS=1	IPK03260
0027	JL=MINO(JS+5,K-1)	IPK03270
0028	WRITE(6,6020)(SS(J),COL(J),J=JS,JL)	IPK03280
0029	6020 FORMAT(1H ,3X,6(I6,F14.6))	IPK03290
0030	IF(MOD(JL,30).EQ.0)WRITE(6,6030)	IPK03300
0031	6030 FORMAT(1H)	IPK03310
0032	IF(JL.EQ.K-1) GO TO 20	IPK03320
0033	JS=JL+1	IPK03330
0034	GO TO 11	IPK03340
0035	20 DO 110 J=1,N1	IPK03350
0036	110 OBJ(J)=0	IPK03360
0037	K=K-1	IPK03370
0038	DO 120 J=1,K	IPK03380
0039	L=SS(J)	IPK03390
0040	IF(L.GT.N.OR.OBJ(L).NE.0.)GO TO 30	IPK03400
0041	120 OBJ(L)=COL(J)	IPK03410
0042	WRITE(DISC1)(OBJ(J),J=1,N1)	IPK03420
0043	RETURN	IPK03430
0044	30 WRITE(6,6040)	IPK03440
0045	6040 FORMAT(1H0,23HINPUT CARD IS INCORRECT)	IPK03450
0046	RETURN 1	IPK03460
0047	END	IPK03470

LINE-NO	STATEMENT	
0001	C	IPK04010
0002	SUBROUTINE SUBS03 (*, *)	IPK04020
0003	C	IPK04030
0004	COMMON ROW,COL,OBJ,S,SS,Y,T,X,F,PH,IP,L,M,N,H1,N1,	IPK04040
0005	1 IS,MN,MD,IA,IB,IC,II,Z	IPK04050
0006	INTEGER S(251),SS(251),Y(251),T(251),X(251),F(251),	IPK04060
0007	IPM(251),IP(10),H(3)	IPK04070
0008	REAL ROW(251),COL(251),OBJ(251)	IPK04080
0009	DATA H/2H <,2H =,2H >/,EN/4HENDA/	IPK04090
0010	IROW=0	IPK04100
0011	IF(IP(1).EQ.1)GO TO 1	IPK04110
0012	WRITE(6,6000)	IPK04120
0013	6000 FORMAT(1H,3X,11HCONSTRAINTS)	IPK04130
0014	1 K=1	IPK04140
0015	IROW=IROW+1	IPK04150
0016	CALL SUBS00(3,ID,D)	IPK04160
0017	100 CALL SUBS00(1,SS(K),D)	IPK04170
0018	CALL SUBS00(2,ID,COL(K))	IPK04180
0019	IF(SS(K).LT.0)GO TO 20	IPK04190
0020	IF(SS(K).EQ.0)GO TO 10	IPK04200
0021	K=K+1	IPK04210
0022	GO TO 100	IPK04220
0023	10 I=IROW	IPK04230
0024	CALL SUBS00(1,F(1),D)	IPK04240
0025	IF(1ABS(F(1)).GT.1)GO TO 20	IPK04250
0026	DO 110 J=1,3	IPK04260
0027	IF(F(1)+2.EQ.J)IS=H(J)	IPK04270
0028	110 CONTINUE	IPK04280
0029	IF(IP(1).EQ.1)GO TO 13	IPK04290
0030	WRITE(6,6010)IROW	IPK04300
0031	6010 FORMAT(1H,3X,6HROW ---,I3)	IPK04310
0032	JS=1	IPK04320
0033	11 JL=M1NO(JS+5,K-1)	IPK04330
0034	WRITE(6,6020)(SS(J),COL(J),J=JS,JL)	IPK04340
0035	6020 FORMAT(1H,3X,6(16,F14.6))	IPK04350
0036	IF(MOD(JL,30).EQ.0)WRITE(6,6030)	IPK04360
0037	6030 FORMAT(1H)	IPK04370
0038	IF(JL.EQ.K-1)GO TO 12	IPK04380
0039	JS=JL +1	IPK04390
0040	GO TO 11	IPK04400
0041	12 WRITE(6,6040)IS,COL(K)	IPK04410
0042	6040 FORMAT(1H,7X,A2,F14.6/)	IPK04420
0043	13 DO 120 J=1,N1	IPK04430
0044	120 ROW(J)=0	IPK04440
0045	DO 130 J=1,K	IPK04450
0046	IF(SS(J).EQ.0)SS(J)=N1	IPK04460
0047	L=SS(J)	IPK04470
0048	IF(L.GT.N1.OR.ROW(L).NE.0.)GO TO 20	IPK04480
0049	130 ROW(L)=COL(J)	IPK04490
0050	WRITE(DISC1)(ROW(J),J=1,N1)	IPK04500

LINE-NO	STATEMENT	
0051	IF(IROW.LT.N)GO TO 1	IPK04510
0052	15 READ(5,5010)HED	IPK04520
0053	5010 FORMAT(A4)	IPK04530
0054	IF(HED.NE.EN)GO TO 15	IPK04540
0055	ENDFILE DISC1	IPK04550
0056	RETURN 2	IPK04560
0057	20 WRITE(6,6050)	IPK04570
0058	6050 FORMAT(1H0,23HINPUT CARD IS INCORRECT)	IPK04580
0059	RETURN 1	IPK04590
0060	END	IPK04600

図 5. (つづき)

OKITAC 4500 FORTRAN SOURCE PROGRAM LIST

PAGE 0001

LINE-NO	STATEMENT	
0001	C	IPK05010
0002	SUBROUTINE SUBS04 (*)	IPK05020
0003	C	IPK05030
0004	COMMON ROW,COL,OBJ,S,SS,Y,T,X,F,PM,IP,L,M,N,M1,N1,	IPK05040
0005	IIS,MN,MD,IA,IB,IC,II,Z	IPK05050
0006	INTEGER S(251),SS(251),Y(251),T(251),X(251),F(251),	IPK05060
0007	IPM(251),IP(10)	IPK05070
0008	RFAL ROW(251),COL(251),OBJ(251)	IPK05080
0009	ICONT=0	IPK05090
0010	WRITE(6,6010)(S(J),J=1,70)	IPK05100
0011	6010 FORMAT(////1H0,21H.. PARAMETRIC NAME ..,3X,70A1,9X,	IPK05110
0012	18X,12H** IPK/45 **//)	IPK05120
0013	MD=(N1*2-1)/256+1	IPK05130
0014	CALL SUBS00(3,ID,D)	IPK05140
0015	CALL SUBS00(1,IP(6),D)	IPK05150
0016	1 CALL SUBS00(1,IP(4),D)	IPK05160
0017	CALL SUBS00(1,IP(5),D)	IPK05170
0018	CALL SUBS00(2,ID,Z)	IPK05180
0019	IF(IP(4).GT.M)GO TO 30	IPK05190
0020	IF(IP(5).GT.N)GO TO 30	IPK05200
0021	IF(IP(4).GE.1)GO TO 10	IPK05210
0022	IF(IP(5).EQ.0)GO TO 20	IPK05220
0023	WRITE(6,6020)IP(5),Z	IPK05230
0024	6020 FORMAT(1H ,3X,12H0BJ.FUNCTION,I5,F19.6)	IPK05240
0025	GO TO 11	IPK05250
0026	10 WRITE(6,6030)IP(4),IP(5),Z	IPK05260
0027	6030 FORMAT(1H ,3X,6HROW --,I3,I8,F19.6)	IPK05270
0028	11 REWIND DISC1	IPK05280
0029	K=IP(4)+1	IPK05290
0030	DO 110 L=1,K	IPK05300
0031	110 READ(DISC1)(ROW(J),J=1,N1)	IPK05310
0032	IF(IP(5).EQ.0)IP(5)=N1	IPK05320
0033	J=IP(5)	IPK05330
0034	ROW(J)=Z	IPK05340
0035	DO 120 K=1,MD	IPK05350
0036	120 BACKSPACE DISC1	IPK05360
0037	WRITE(DISC1)(ROW(J),J=1,N1)	IPK05370
0038	20 ICONT=ICONT+1	IPK05380
0039	IF(ICONT.LT.IP(6))GO TO 1	IPK05390
0040	IP(3)=1	IPK05400
0041	RETURN	IPK05410
0042	30 WRITE(6,6040)	IPK05420
0043	6040 FORMAT(1H0,23HINPUT CARD IS INCORRECT)	IPK05430
0044	RETURN 1	IPK05440
0045	END	IPK05450

LINE-NO	STATEMENT	
0001	C -----	IPK06010
0002	SUBROUTINE SUBS00 (N,IN,RN)	IPK06020
0003	C -----	IPK06030
0004	LOGICAL INTEGR,MINUS,FRACTN	IPK06040
0005	DIMENSION JCHAR(13),INBUFF(80),SUM(2)	IPK06050
0006	DATA JCHAR/1H.,1H.,1H.,1H0,1H1,1H2,1H3,1H4,1H5,1H6,	IPK06060
0007	11H7,1H8,1H9/,1ST/1H*/	IPK06070
0008	GO TO(1,2,41,42),N	IPK06080
0009	1 INTEGR=.TRUE.	IPK06090
0010	GO TO 3	IPK06100
0011	2 INTEGR=.FALSE.	IPK06110
0012	3 MINUS=.FALSE.	IPK06120
0013	FRACTN=.FALSE.	IPK06130
0014	SUM(1)=0	IPK06140
0015	SUM(2)=0	IPK06150
0016	ISWITCH=1	IPK06160
0017	4 IF(ICOL.EQ.80.AND.ISWITCH.NE.1)GO TO 120	IPK06170
0018	5 IF(ICOL=80)7,6,6	IPK06180
0019	6 HEAD(5,5010)(INBUFF(J),J=1,80)	IPK06190
0020	5010 FORMAT(80A1)	IPK06200
0021	IF(INBUFF(1).EQ.1ST)GO TO 6	IPK06210
0022	ICOL=0	IPK06220
0023	7 ICOL=ICOL+1	IPK06230
0024	DO 100 JCOL=1,13	IPK06240
0025	IF(INBUFF(ICOL).EQ.JCHAR(JCOL))GO TO 10	IPK06250
0026	100 CONTINUE	IPK06260
0027	GO TO 42	IPK06270
0028	10 IF(JCOL=4)11,14,14	IPK06280
0029	11 IF(ISWITCH=1)12,12,13	IPK06290
0030	12 GO TO(5,110,20),JCOL	IPK06300
0031	13 GO TO(120,110,42),JCOL	IPK06310
0032	14 ADD=JCOL-4	IPK06320
0033	IF(FRACTN)GO TO 15	IPK06330
0034	SUM(1)=SUM(1)+10.*ADD	IPK06340
0035	GO TO 21	IPK06350
0036	15 SUM(2)=SUM(2)+10.*ADD	IPK06360
0037	ISUM2=ISUM2+1	IPK06370
0038	GO TO 4	IPK06380
0039	110 IF(FRACTN)GO TO 42	IPK06390
0040	FRACTN=.TRUE.	IPK06400
0041	ISUM2 =0	IPK06410
0042	GO TO 21	IPK06420
0043	20 MINUS=.TRUE.	IPK06430
0044	21 ISWITCH=2	IPK06440
0045	GO TO 4	IPK06450
0046	22 IF(MINUS)SUM(1)=-SUM(1)	IPK06460
0047	IN=SUM(1)	IPK06470
0048	IF(FRACTN)GO TO 42	IPK06480
0049	GO TO 50	IPK06490
0050	120 IF(INTEGR)GO TO 22	IPK06500

☒ 5. (つづき)

OKITAC 4500 FORTRAN SOURCE PROGRAM LIST

PAGE 0002

LINE-NO	STATEMENT	
0051	IF(ISUM2)40,40,30	IPK06510
0052	30 DO 130 I=1,ISUM2	IPK06520
0053	130 SUM(2)=SUM(2)/10.	IPK06530
0054	40 RN=SUM(1)+SUM(2)	IPK06540
0055	IF(MINUS)RN=-RN	IPK06550
0056	IF(.NOT.FRACTN)GO TO 42	IPK06560
0057	GO TO 50	IPK06570
0058	41 ICOL= 80	IPK06580
0059	GO TO 50	IPK06590
0060	42 WRITE(6,6010)(INBUFF(J),J=1,80)	IPK06600
0061	6010 FORMAT(1H0,3X,10HINPUT CARD,	IPK06610
0062	113H IS INCORRECT/1H0,3X,80A1)	IPK06620
0063	STOP	IPK06630
0064	50 RETURN	IPK06640
0065	END	IPK06650

LINE-NO	STATEMENT	
0001	C	IPK07010
0002	SUBROUTINE SUBS05	IPK07020
0003	C	IPK07030
0004	COMMON ROW,COL,OBJ,S,SS,Y,T,X,F,PM,IP,L,M,N,M1,N1,	IPK07040
0005	1IS,MN,MD,IA,IB,IC,II,Z	IPK07050
0006	INTEGER S(251),SS(251),Y(251),T(251),X(251),F(251),	IPK07060
0007	IPM(251),IP(10)	IPK07070
0008	REAL ROW(251),COL(251),OBJ(251)	IPK07080
0009	4010 FORMAT(/1H,3X,3HNO,4X,9HOBJ,VALUE)	IPK07090
0010	REWIND DISC1	IPK07100
0011	REWIND DISC2	IPK07110
0012	REWIND DISC3	IPK07120
0013	IA=0	IPK07130
0014	IB=0	IPK07140
0015	IC=0	IPK07150
0016	IS=0	IPK07160
0017	Z=1.0E30	IPK07170
0018	DO 120 I=1,M1	IPK07180
0019	READ(DISC1)(ROW(J),J=1,N1)	IPK07190
0020	IF(1.GE.2)GO TO 10	IPK07200
0021	DO 100 J=1,N	IPK07210
0022	IF(MN.EQ.1)ROW(J)=-ROW(J)	IPK07220
0023	SS(J)=0	IPK07230
0024	PM(J)=0	IPK07240
0025	IF(ROW(J).GE.0.)GO TO 100	IPK07250
0026	PM(J)=1	IPK07260
0027	ROW(N1)=ROW(N1)+ROW(J)	IPK07270
0028	ROW(J)=-ROW(J)	IPK07280
0029	100 CONTINUE	IPK07290
0030	GO TO 20	IPK07300
0031	10 COL(I-1)=ROW(N1)	IPK07310
0032	ROW(N1)=-ROW(N1)	IPK07320
0033	DO 110 J=1,N	IPK07330
0034	IF(PM(J).EQ.0)GO TO 110	IPK07340
0035	ROW(N1)=ROW(N1)+ROW(J)	IPK07350
0036	ROW(J)=-ROW(J)	IPK07360
0037	110 CONTINUE	IPK07370
0038	20 WRITE(DISC2)(ROW(J),J=1,N1)	IPK07380
0039	120 CONTINUE	IPK07390
0040	WRITE(DISC3)(COL(I),I=1,M)	IPK07400
0041	IF(IP(2).EQ.1)WRITE(4,4010)	IPK07410
0042	ENDFILE DISC1	IPK07420
0043	ENDFILE DISC2	IPK07430
0044	ENDFILE DISC3	IPK07440
0045	RETURN	IPK07450
0046	END	IPK07460

図 5. (つづき)

OKITAC 4500 FORTRAN SOURCE PROGRAM LIST

PAGE 0001

LINE-NO	STATEMENT	
0001	C	IPK08010
0002	SUBROUTINE SUBS06 (*)	IPK08020
0003	C	IPK08030
0004	COMMON ROW,COL,OBJ,S,SS,Y,T,X,F,PM,IP,L,M,N,M1,N1,	IPK08040
0005	IIS,MN,MD,IA,IB,IC,II,Z	IPK08050
0006	INTEGER S(251),SS(251),Y(251),T(251),X(251),F(251),	IPK08060
0007	IPM(251),IP(10)	IPK08070
0008	REAL ROW(251),COL(251),OBJ(251)	IPK08080
0009	DO 100 J=1,N	IPK08090
0010	S(J)=0	IPK08100
0011	IF(IIS.EQ.0)GO TO 10	IPK08110
0012	DO 110 K=1,IIS	IPK08120
0013	J=SS(K)	IPK08130
0014	IF(J.LE.0)GO TO 110	IPK08140
0015	S(J)=1	IPK08150
0016	110 CONTINUE	IPK08160
0017	10 ICONT=0	IPK08170
0018	REWIND DISC2	IPK08180
0019	READ(DISC2)(OBJ(J),J=1,N1)	IPK08190
0020	DO 130 I=1,M	IPK08200
0021	READ(DISC2)(ROW(J),J=1,N1)	IPK08210
0022	Y(I)=0	IPK08220
0023	COL(I)=ROW(N1)	IPK08230
0024	DO 120 J=1,N	IPK08240
0025	IF(S(J).EQ.1)COL(I)=COL(I)+ROW(J)	IPK08250
0026	120 CONTINUE	IPK08260
0027	IF(ABS(COL(I)).LT.0.5E-5)COL(I)=0	IPK08270
0028	IF(COL(I))11,13,12	IPK08280
0029	11 IF(F(I).GE.0)Y(I) =1	IPK08290
0030	GO TO 13	IPK08300
0031	12 IF(F(I).LE.0)Y(I) =2	IPK08310
0032	13 IF(Y(I).GE.1)ICONT=1	IPK08320
0033	130 CONTINUE	IPK08330
0034	IA=IA+1	IPK08340
0035	IF(IP(2).EQ.0)GO TO 140	IPK08350
0036	Z0=Z	IPK08360
0037	IF(MN.EQ.1)Z0=-Z0	IPK08370
0038	IM=IP(8)	IPK08380
0039	IF(IP(8).EQ.0)IM=25	IPK08390
0040	IF(MOD(IA,IM).EQ.0)WRITE(4,4010)IA,Z0	IPK08400
0041	4010 FORMAT(1H ,I5,F14.6)	IPK08410
0042	140 CONTINUE	IPK08420
0043	IF(ICONT.EQ.1)GO TO 20	IPK08430
0044	RETURN	IPK08440
0045	20 RETURN	IPK08450
0046	END	IPK08460

LINE-NO	STATEMENT	
0001	C	IPK09010
0002	SUBROUTINE SUBS07	IPK09020
0003	C	IPK09030
0004	COMMON ROW,COL,OBJ,S,SS,Y,T,X,F,PM,IP,L,M,N,M1,N1	IPK09040
0005	1 IS,MN,MD,IA,IB,IC,II,Z	IPK09050
0006	INTEGER S(251),SS(251),Y(251),T(251),X(251),F(251),	IPK09060
0007	IPM(251),IP(10),R(251)	IPK09070
0008	REAL ROW(251),COL(251),OBJ(251)	IPK09080
0009	Z0=OBJ(N1)	IPK09090
0010	DO 100 J=1,N	IPK09100
0011	T(J)=0	IPK09110
0012	R(J)=0	IPK09120
0013	IF(S(J).EQ.1)Z0=Z0+OBJ(J)	IPK09130
0014	100 CONTINUE	IPK09140
0015	IF(IS.EQ.0)GO TO 20	IPK09150
0016	DO 110 K=1,IS	IPK09160
0017	J=IABS(SS(K))	IPK09170
0018	IF(J.LE.0)GO TO 110	IPK09180
0019	R(J)=1	IPK09190
0020	110 CONTINUE	IPK09200
0021	20 REWIND DISC2	IPK09210
0022	READ(DISC2)(OBJ(J),J=1,N1)	IPK09220
0023	IP(7)=1	IPK09230
0024	DO 130 I=1,M	IPK09240
0025	READ(DISC2)(ROW(J),J=1,N1)	IPK09250
0026	K=Y(I)	IPK09260
0027	IF(K.EQ.0)GO TO 130	IPK09270
0028	DO 120 J=1,N	IPK09280
0029	IF(S(J).EQ.1.OR.R(J).EQ.1)GO TO 120	IPK09290
0030	IF(T(J).EQ.1)GO TO 120	IPK09300
0031	IF(Z0+OBJ(J).GE.Z)GO TO 120	IPK09310
0032	GO TO(30,40),K	IPK09320
0033	30 IF(ROW(J).GT.0.)T(J)=1	IPK09330
0034	GO TO 50	IPK09340
0035	40 IF(ROW(J).LT.0.)T(J)=1	IPK09350
0036	50 IF(T(J).EQ.0)GO TO 120	IPK09360
0037	IP(7)=0	IPK09370
0038	120 CONTINUE	IPK09380
0039	130 CONTINUE	IPK09390
0040	RETURN	IPK09400
0041	END	IPK09410

図 5. (つづき)

OKITAC 4500 FORTRAN SOURCE PROGRAM LIST

PAGE 0001

LINE-NO	STATEMENT	
0001	C	IPK10010
0002	SUBROUTINE SUBS08 (*)	IPK10020
0003	C	IPK10030
0004	COMMON ROW,COL,OBJ,S,SS,Y,T,X,F,PM,IP,L,M,N,M1,N1,	IPK10040
0005	1IS,MN,MD,IA,IB,IC,II,Z	IPK10050
0006	INTEGER S(251),SS(251),Y(251),T(251),X(251),F(251),	IPK10060
0007	IPM(251),IP(10)	IPK10070
0008	REAL ROW(251),COL(251),OBJ(251)	IPK10080
0009	IF(IP(7).EQ.1)GO TO 30	IPK10090
0010	REWIND DISC2	IPK10100
0011	READ(DISC2)(OBJ(J),J=1,N1)	IPK10110
0012	DO 110 I=1,M	IPK10120
0013	READ(DISC2)(ROW(J),J=1,N1)	IPK10130
0014	K=Y(I)	IPK10140
0015	IF(K.EQ.0)GO TO 110	IPK10150
0016	SUM=COL(I)	IPK10160
0017	DO 100 J=1,N	IPK10170
0018	IF(T(J).EQ.0)GO TO 100	IPK10180
0019	GO TO(10,11),K	IPK10190
0020	10 IF(ROW(J),GT.0.)SUM=SUM+ROW(J)	IPK10200
0021	GO TO 100	IPK10210
0022	11 IF(ROW(J),LT.0.)SUM=SUM+ROW(J)	IPK10220
0023	100 CONTINUE	IPK10230
0024	GO TO(20,21),K	IPK10240
0025	20 IF(SUM,LT.0.)GO TO 30	IPK10250
0026	GO TO 110	IPK10260
0027	21 IF(SUM,GT.0.)GO TO 30	IPK10270
0028	110 CONTINUE	IPK10280
0029	RETURN	IPK10290
0030	30 RETURN 1	IPK10300
0031	END	IPK10310

LINE-NO	STATEMENT	
0001	C	IPK11010
0002	-----	IPK11020
0003	C SUBROUTINE SUBS09 (*)	IPK11030
0004	-----	IPK11040
0005	COMMON ROW,COL,OBJ,S,SS,Y,T,X,F,PN,IP,L,M,N,M1,N1,	IPK11050
0006	11S,MN,MN,IA,IB,IC,II,Z	IPK11060
0007	INTEGR S(251),SS(251),Y(251),T(251),X(251),F(251),	IPK11070
0008	IPM(251),IP(10)	IPK11080
0009	REAL ROW(251),COL(251),OBJ(251),BUF(251)	IPK11090
0010	D=-1.0E30	IPK11100
0011	E= 1.0E30	IPK11110
0012	DO 100 J=1,N	IPK11120
0013	S(J)=0	IPK11130
0014	100 BUF(J)=0	IPK11140
0015	REWIND DISCZ	IPK11150
0016	READ(DISCZ)(OBJ(J),J=1,N1)	IPK11160
0017	DO 110 I=1,M	IPK11170
0018	READ(DISCZ)(ROW(J),J=1,N1)	IPK11180
0019	DO 110 J=1,N	IPK11190
0020	IF(T(J).EQ.0)GO TO 110	IPK11200
0021	B1=ROW(J)*COL(I)	IPK11210
0022	IF(ABS(B1).LT.0.5E-5)B1=0	IPK11220
0023	IF(F(I).EQ. 0)B1=-ABS(B1)	IPK11230
0024	10 IF(B1.GE.0.5)S(J)=S(J)+1	IPK11240
0025	IF(B1.LT.0.5)B1=-{B1*B1}	IPK11250
0026	BUF(J)=BUF(J)+B1	IPK11260
0027	110 CONTINUE	IPK11270
0028	DO 120 J=1,N	IPK11280
0029	IF(S(J).LT.M)GO TO 11	IPK11290
0030	DO 115 J=1,N	IPK11300
0031	IF(S(J).LT.M.OR.OBJ(J).GT.E)GO TO 115	IPK11310
0032	E=OBJ(J)	IPK11320
0033	K=J	IPK11330
0034	115 CONTINUE	IPK11340
0035	GO TO 30	IPK11350
0036	11 IF(T(J).EQ.0.OR.BUF(J).LT.D)GO TO 120	IPK11360
0037	D=BUF(J)	IPK11370
0038	K=J	IPK11380
0039	120 CONTINUE	IPK11390
0040	DO 130 J=1,N	IPK11400
0041	IF(J.EQ.K)GO TO 130	IPK11410
0042	IF(T(J).EQ.1.AND.BUF(J).GE.D)GO TO 20	IPK11420
0043	130 CONTINUE	IPK11430
0044	GO TO 30	IPK11440
0045	20 DO 140 J=1,N	IPK11450
0046	IF(T(J).EQ.0)GO TO 140	IPK11460
0047	IF(BUF(J).LT.D.OR.OBJ(J).GT.E)GO TO 140	IPK11470
0048	E=OBJ(J)	IPK11480
0049	K=J	IPK11490
0050	140 CONTINUE	IPK11500

図 5. (つづき)

OKITAC 4500 FORTRAN SOURCE PROGRAM LIST

PAGE 0002

LINE-NO	STATEMENT	
0051	30 IS=IS+1	IPK11510
0052	SS(IS)=K	IPK11520
0053	RETURN 1	IPK11530
0054	END	IPK11540

OKITAC 4500 FORTRAN SOURCE PROGRAM LIST

PAGE 0001

LINE-NO	STATEMENT	
0001	C	IPK12010
0002	SUBROUTINE SUBS10 (*)	IPK12020
0003	C	IPK12030
0004	COMMON ROW, COL, OBJ, S, SS, Y, T, X, F, PM, IP, L, M, N, M1, N1,	IPK12040
0005	IIS, MN, MD, IA, IB, IC, II, Z	IPK12050
0006	INTEGER S(251), SS(251), Y(251), T(251), X(251), F(251),	IPK12060
0007	IPM(251), IP(10)	IPK12070
0008	REAL ROW(251), COL(251), OBJ(251)	IPK12080
0009	SUM=OBJ(N1)	IPK12090
0010	DO 100 J=1, N	IPK12100
0011	IF(S(J).EQ.1) SUM=SUM+OBJ(J)	IPK12110
0012	100 CONTINUE	IPK12120
0013	IF(SUM.GE.Z) GO TO 10	IPK12130
0014	DO 110 J=1, N	IPK12140
0015	110 X(J)=S(J)	IPK12150
0016	Z=SUM	IPK12160
0017	IB=IA	IPK12170
0018	IF(IC.EQ.0) IC=IA	IPK12180
0019	Z0=Z	IPK12190
0020	IF(MN.EQ.1) Z0=-Z0	IPK12200
0021	IF(IP(2).EQ.1) WRITE(4, 4010) IB, Z0	IPK12210
0022	4010 FORMAT(1H, '15', F14.6)	IPK12220
0023	RETURN	IPK12230
0024	10 RETURN 1	IPK12240
0025	END	IPK12250

LINE-NO	STATEMENT	
0001	C	IPK13010
0002	-----	IPK13020
0003	SUBROUTINE SUBS11 (*)	IPK13030
0004	-----	IPK13040
0005	COMMON ROW,COL,OBJ,S,SS,Y,T,X,F,PM,IP,L,M,N,M1,N1,	IPK13050
0006	11S,MN,MD,[A,IB,IC,I1,2	IPK13060
0007	INTEGER S(251),SS(251),Y(251),T(251),X(251),F(251),	IPK13070
0008	IPM(251),IP(10)	IPK13080
0009	REAL ROW(251),COL(251),OBJ(251)	IPK13090
0010	DO 100 K=1,N	IPK13100
0011	J=1,N-K	IPK13110
0012	IF(SS(J).EQ.0)GO TO 100	IPK13120
0013	IF(SS(J).GT.0)GO TO 10	IPK13130
0014	SS(J)=0	IPK13140
0015	IS=IS-1	IPK13150
0016	GO TO 100	IPK13160
0017	SS(J)=-SS(J)	IPK13170
0018	GO TO 20	IPK13180
0019	CONTINUE	IPK13190
0020	20 RETURN 1	IPK13200
	END	

LINE-NO	STATEMENT	
0001	C	IPK14010
0002	-----	IPK14020
0003	SUBROUTINE SUBS12 (*)	IPK14030
0004	-----	IPK14040
0005	COMMON ROW,COL,OBJ,S,SS,Y,T,X,F,PM,IP,L,M,N,M1,N1,	IPK14050
0006	11S,MN,MD,[A,IB,IC,I1,2	IPK14060
0007	INTEGER S(251),SS(251),Y(251),T(251),X(251),F(251),	IPK14070
0008	IPM(251),IP(10)	IPK14080
0009	REAL ROW(251),COL(251),OBJ(251)	IPK14090
0010	DO 100 J=1,N	IPK14100
0011	IF(SS(J).GT.0)GO TO 10	IPK14110
0012	CONTINUE	IPK14120
0013	RETURN	IPK14130
0014	10 RETURN 1	IPK14140
	END	

図 5. (つづき)

OKITAC 4500 FORTRAN SOURCE PROGRAM LIST

PAGE 0001

LINE-NO	STATEMENT	
0001	C	IPK15010
0002	-----	IPK15020
0003	SUBROUTINE SUBS13	IPK15030
0004	C	IPK15040
0005	COMMON ROW,COL,OBJ,S,SS,Y,T,X,F,PM,IP,L,M,N,M1,N1,	IPK15050
0006	1IS,MN,MD,IA,IB,IC,II,Z	IPK15060
0007	INTEGER S(251),SS(251),Y(251),T(251),X(251),F(251),	IPK15070
0008	IPM(251),IP(10)	IPK15080
0009	REAL ROW(251),COL(251),OBJ(251),PT(5),QT(5)	IPK15090
0010	IF(IP(1).EQ.1)GO TO 30	IPK15100
0011	IF(IP(3).EQ.1)GO TO 30	IPK15110
0012	DO 100 J=1,5	IPK15120
0013	100 S(J)=0	IPK15130
0014	SUM=0	IPK15140
0015	REWIND DISC2	IPK15150
0016	READ(DISC2)(OBJ(J),J=1,N1)	IPK15160
0017	DO 120 I=1,M	IPK15170
0018	READ(DISC2)(ROW(J),J=1,N1)	IPK15180
0019	DO 120 J=1,N	IPK15190
0020	IF(ROW(J).EQ.0.)GO TO 120	IPK15200
0021	SUM=SUM+1.	IPK15210
0022	UNI=1	IPK15220
0023	DO 110 ICN=1,4	IPK15230
0024	IF(ABS(ROW(J)).LE.UNI)GO TO 20	IPK15240
0025	110 UNI=UNI*10.	IPK15250
0026	ICN=5	IPK15260
0027	20 S(ICN)=S(ICN)+1	IPK15270
0028	120 CONTINUE	IPK15280
0029	ELE=FLOAT(M)*FLOAT(N)	IPK15290
0030	PCT=SUM/ELE*100.	IPK15300
0031	DO 130 J=1,5	IPK15310
0032	PT(J)=FLOAT(S(J))/SUM*100.	IPK15320
0033	130 QT(J)=PT(J)*PCT/100.	IPK15330
0034	WRITE(6,6010)ELE,S(1),PT(1),QT(1),PCT,S(2),	IPK15340
0035	1PT(2),QT(2)	IPK15350
0036	6010 FORMAT(1H0,23H.. MATRIX STATISTICS .. //,6X,3HELE,	IPK15360
0037	15HMENTS,18X,1H=,F12.0,7X,4HDATA,18X,11H0.000 ----,	IPK15370
0038	29H-- 1.000,3X,1H=,I6,2F10.2/1H,5X,11HDATA DENSIT,	IPK15380
0039	31HY,14X,1H=,F11.2,8X,12HDISTRIBUTION,10X,7H1.000 ,	IPK15390
0040	413H---- 10.000,3X,1H=,I6,2F10.2)	IPK15400
0041	WRITE(6,6020)IC,S(3),PT(3),QT(3),1B,S(4),	IPK15410
0042	1PT(4),QT(4),1A,S(5),PT(5),QT(5)	IPK15420
0043	6020 FORMAT(1H,5X,22HITERATIONS AT FEASIBLE,4X,1H=,I11,	IPK15430
0044	129X,21H10.000 ---- 100.000,3X,1H=,I6,2F10.2 /1H,	IPK15440
0045	25X,21HITERATIONS AT OPTIMAL,5X,1H=,I11,28X,5H100.0,	IPK15450
0046	317H00 --- 1000.000,3X,1H=,I6,2F10.2/1H,5X,3HITE,	IPK15460
0047	417HATIONS AT FINISH,6X,1H=,I11,27X,11H1000.000 --,	IPK15470
0048	55H-----,10X,1H=,I6,2F10.2)	IPK15480
0049	30 RETURN	IPK15490
	END	

LINE-NO	STATEMENT	
0001	C	IPK16010
0002	SUBROUTINE SUBS14 (*)	IPK16020
0003	C	IPK16030
0004	COMMON ROW,COL,OBJ,S,SS,Y,T,X,F,PM,IP,L,M,N,M1,N1,	IPK16040
0005	I1S,MN,MD,IA,IB,IC,II,Z	IPK16050
0006	INTEGER S(251),SS(251),Y(251),T(251),X(251),F(251),	IPK16060
0007	IPM(251),IP(10)	IPK16070
0008	REAL ROW(251),COL(251),OBJ(251),RAT(251),H1(251),	IPK16080
0009	H2(251)	IPK16090
0010	DATA H1,H2/251*4H X(.251*4H) = /	IPK16100
0011	IF(2.EQ.1.0E30)GO TO 40	IPK16110
0012	DO 100 J=1,N	IPK16120
0013	OBJ(J)=X(J)	IPK16130
0014	IF(PM(J).EQ.0)GO TO 100	IPK16140
0015	IF(X(J).EQ.1)OBJ(J)=0	IPK16150
0016	IF(X(J).EQ.0)OBJ(J)=1	IPK16160
0017	100 CONTINUE	IPK16170
0018	REWIND DISC2	IPK16180
0019	READ(DISC2)(COL(J),J=1,N1)	IPK16190
0020	DO 120 I=1,M	IPK16200
0021	READ(DISC2)(ROW(J),J=1,N1)	IPK16210
0022	COL(I)=0	IPK16220
0023	T(I)=1+1000	IPK16230
0024	DO 110 J=1,N	IPK16240
0025	IF(OBJ(J).EQ.1)COL(I)=COL(I)+ROW(J)	IPK16250
0026	110 CONTINUE	IPK16260
0027	IF(MN.EQ.1)COL(I)=-COL(I)	IPK16270
0028	120 CONTINUE	IPK16280
0029	IF(MN.EQ.1)Z=-Z	IPK16290
0030	WRITE(6,6010)Z	IPK16300
0031	6010 FORMAT(/1H0,22H.. OPTIMAL SOLUTION ..,/1H0,3X,3H0BJ,	IPK16310
0032	120HECTIV FUNCTION VALUE,20X,1H=,F16.7/1H0,3X,3HSEC,	IPK16320
0033	219HTION (1) - COLUMNS/1H0,5X,5HVALUE)	IPK16330
0034	JS=1	IPK16340
0035	10 JL=MINO(JS+3,N)	IPK16350
0036	WRITE(6,6020)(H1(J),J,H2(J),OBJ(J),J=JS,JL)	IPK16360
0037	6020 FORMAT(1H ,3X,4(3X,A4,I4,A4,F15.7))	IPK16370
0038	IF(MOD(JL,20).EQ.0)WRITE(6,6030)	IPK16380
0039	6030 FORMAT(1H)	IPK16390
0040	IF(JL.EQ.N)GO TO 20	IPK16400
0041	JS=JL +1	IPK16410
0042	GO TO 10	IPK16420
0043	20 WRITE(6,6040)	IPK16430
0044	6040 FORMAT(/1H0,3X,19HSECTION (2) - ROWS/1H0,5X,3HVAL,	IPK16440
0045	12HUE)	IPK16450
0046	6050 FORMAT(///1H0,5X,27HNO FEASIBLE SOLUTION EXISTS)	IPK16460
0047	ICONT=0	IPK16470
0048	21 JS=1	IPK16480
0049	22 JL=MINO(JS+3,M)	IPK16490
0050	WRITE(6,6020)(H1(J),T(J),H2(J),COL(J),J=JS,JL)	IPK16500

図 5. (つづき)

OKITAC 4500 FORTRAN SOURCE PROGRAM LIST

PAGE 0002

LINE-NO	STATEMENT	
0051	IF(MOD(JL,20).EQ.0)WRITE(6,6030)	IPK16510
0052	IF(JL.EQ.M)GO TO 23	IPK16520
0053	JS=JL+1	IPK16530
0054	GO TO 22	IPK16540
0055	23 ICONT=ICONT+1	IPK16550
0056	GO TO(24,31,41),ICONT	IPK16560
0057	24 WRITE(6,6060)	IPK16570
0058	6060 FORMAT(/1H0,5X,11HSLACK VALUE)	IPK16580
0059	RFWIND DISC3	IPK16590
0060	READ(DISC3)(OBJ(I),I=1,M)	IPK16600
0061	DO 130 I=1,M	IPK16610
0062	RAT(I)=100	IPK16620
0063	IF(OBJ(I).EQ.0.)GO TO 30	IPK16630
0064	RAT(I)=COL(I)/OBJ(I)	IPK16640
0065	30 IF(COL(I).EQ.0.)RAT(I)=0	IPK16650
0066	130 COL(I)=OBJ(I)-COL(I)	IPK16660
0067	GO TO 21	IPK16670
0068	31 WRITE(6,6070)	IPK16680
0069	6070 FORMAT(/1H0,5X,9HACT RATIO)	IPK16690
0070	DO 140 I=1,M	IPK16700
0071	140 COL(I)=RAT(I)	IPK16710
0072	GO TO 21	IPK16720
0073	40 WRITE(6,6050)	IPK16730
0074	41 II=3	IPK16740
0075	RETURN 1	IPK16750
0076	END	IPK16760