法政大学学術機関リポジトリ
HOSEI UNIVERSITY REPOSITORY

PDF issue: 2025-07-15

# 資源配分を伴うNetwork Scheduling問題 : (0-1)整数計画法によるApproach

FUKUMA, Toshiko / YAMAMOTO, Masaaki / 山本, 正明 / 福馬, 敏子

(出版者 / Publisher)
法政大学工学部
(雑誌名 / Journal or Publication Title)
Bulletin of the Technical College of Hosei University / 法政大学工学部研究集報
(巻 / Volume)
19
(開始ページ / Start Page)
161
(終了ページ / End Page)
170
(発行年 / Year)
1983-03
(URL)
https://doi.org/10.15002/00004083

# 資源配分を伴う Network Scheduling 問題

## ——(0-1) 整数計画法による Approach——

福馬敏子*・山本正明*

# Network Scheduling Problem with Limited Resources

## ——(0-1) Integer Programming Approach——

Toshiko FUKUMA and Masaaki YAMAMOTO


### Abstract

A multiproject scheduling problem with limited resources is solved by using a zero one (0-1) linear programming.

We developed an algorithm which examine the feasibility of (0-1) programming problems rather than solve it exactly. It can attain an optimal solution in less iteration steps than PATTERSON and HUBER did previously.

The formulation requires the huge size of matrix and it must be coded as a special form of LP-code which depends on a computer used. In the algorithm it is necessary to change the matrix in each iteration. To generate an adequate matrix automatically we developed the MATRIX-GENERATOR and connect it with LP-code.


## §1. 緒　　論

資源制約を伴うマルチプロジェクトのスケジューリング問題に対して，(0-1) 整数計画法を用いて定式化を行う方法は，PRITSKER (1969) により提案された[1]。この論文では，マルチプロジェクトの最適スケジュールを得るため，各制約式を作成するときに任意の絶対納期を用いて定式化を行っている。その場合絶対納期の値によって，計算ステップ数が大幅に変わり，実行可能解が得られなかったり，解を求めるために，膨大な計算時間が必要となることがある。

そこで，PATTERSON and HUBER (1974) は，PRITSKER の定式化を用いて，(0-1)整数計画法により，確実に，また効率的に最適解を得るための手法として，Bounding-Algorithm を提案した[2]。

(0-1) 整数計画法による解法は，その性質から，変数の数，制約式の数が膨大になる。例をと

* 経営工学科

ると，ジョブの数が8個程度の問題で，変数は37，制約式は42本にも達する程である。

これ等の問題を電子計算機で解く場合，問題に応じた定式化と，通常特定の LP コードをもつ特有な入力データフォームへの変換が必要となる。

この研究は，資源制約をもつマルチプロジェクト・スケジューリングに対する (0-1) 整数計画法の計算効率を改善し，解法としての実用可能性を検討することを目的としている。そのため，

1) 既存の数理計画法の計算コードを，直接利用するための Matrix-Generator を作成した。

2) Bounding-Algorithm のステップ数を短縮することにより，計算時間の短縮をはかった。

## §2. 資源配分を伴うプロジェクト・スケジューリング問題

1つの目的を達成するための作業の集まりをプロジェクトとよび，プロジェクトを構成する個々の作業をジョブとよぶことにする。資源配分を伴うプロジェクト・スケジューリング問題とは，限られた資源の中で，いかに効率良くプロジェクト内のジョブを終えることができるかを考えることである。

ここで，ネットワークを用いて，各々4個，2個のジョブを含む2プロジェクト・スケジューリ

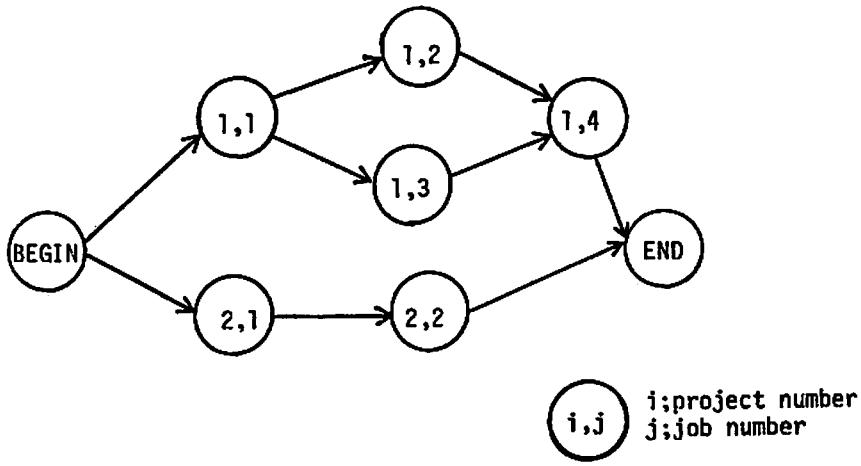



Fig. 1 Network of the test problem.

Table 1 Project number, Job number, Job Durations, Resource Requirements, and Arrival Times.

| $i$ | $j$ | $d_{ij}$ | $r_{ijk}$ | $a_{ij}$ |
|---|---|---|---|---|
| 1 | 1 | 4 | 3 | 1 |
| 1 | 2 | 2 | 2 | 1 |
| 1 | 3 | 3 | 1 | 1 |
| 1 | 4 | 2 | 3 | 1 |
| 2 | 1 | 3 | 1 | 3 |
| 2 | 2 | 3 | 3 | 3 |

($G_1=G_2=13$)
($k=1$, $R_1=4$)

$i$ : project number
$j$ : job number
$d_{ij}$: job duration of job $j$ in project $i$.
$r_{ijk}$: amount of type $k$ resource required on job $j$ in project $i$.
$a_{ij}$: arrival period of job $j$ in project $i$.

$G_i$: absolute due date
$R_k$: resource availability
$k$ : resource number

ング問題の例を Fig. 1 に示す。各ジョブには先行関係があり，また，Table 1 に示すような所要時間，資源要求量，ジョブの到着時刻が与えられている。プロジェクトに対しては，絶対納期が決められており，資源に対しては，その種類と各時刻における総使用可能量が示されている。

以上のプロジェクト内のジョブ構成を，ガントチャートを用いて表現すると Fig. 2 のようになる。ここで使用されている添字及び変数は，次のように説明される。

$I$：プロジェクト数

$i$：プロジェクト番号（$i=1, 2, \cdots, I$）

$N_i$：プロジェクト $i$ を構成するジョブ数

$j$：ジョブ番号（$j=1, 2, \cdots, N_i$）

$t$：時刻（$t=1, 2, \cdots, \max G_i$）

$G_i$：プロジェクト $i$ の絶対納期。プロジェクト $i$ は期間 $G_i$ の内に遂行されなければならない

$c_i$：プロジェクト $i$ の最早完了時刻に 1 を加えたもの

$a_{ij}$：プロジェクト $i$ のジョブ $j$ の到着時刻

$d_{ij}$：プロジェクト $i$ のジョブ $j$ の所要時間

$l_{ij}$：プロジェクト $i$ のジョブ $j$ の最早完了時刻。$a_{ij}$ を基にして計算される

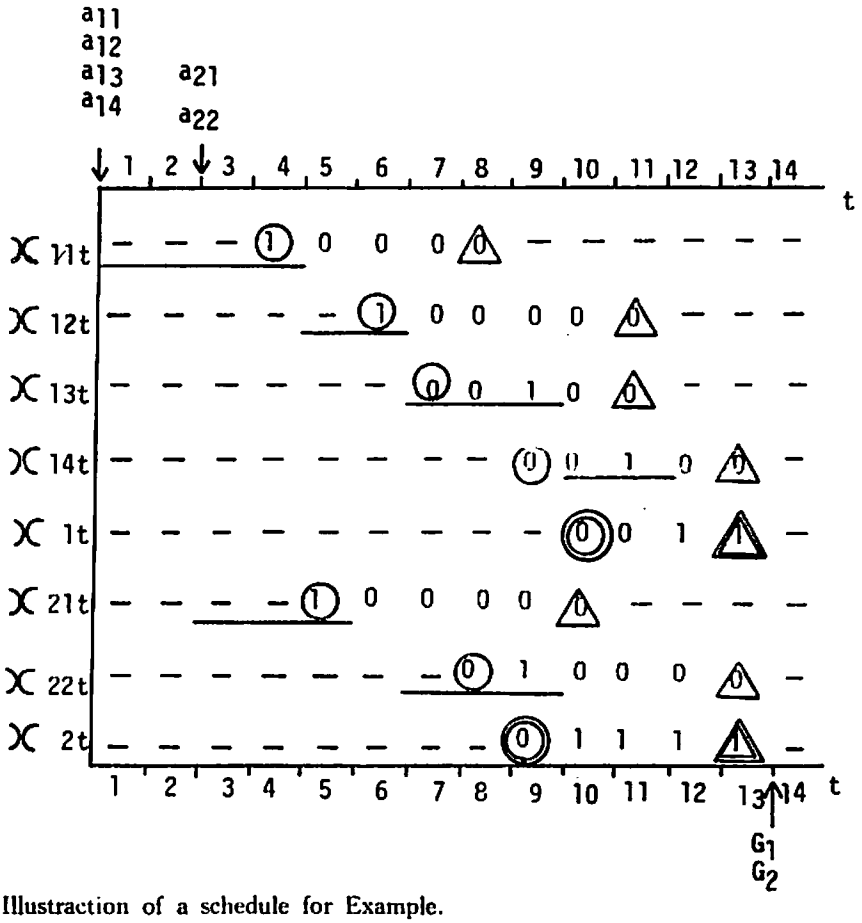



Fig. 2　Illustraction of a schedule for Example.

$u_{ij}$: プロジェクト $i$ のジョブ $j$ の早遅完了時刻。$G_i$ をもとにして計算される

$K$: 異なる資源の種類数

$k$: 資源番号（$k=1, 2, \cdots, K$）

$r_{ijk}$: プロジェクト $i$ のジョブ $j$ の種類 $k$ に対する資源要求量

$R_{kt}$: 時刻 $t$ における種類 $k$ の資源の利用可能量

$x_{ijt}$: 変数。 $x_{ijt}=1$（ジョブ $j$ が区間 $t$ で終了）

$=0$（その他）

変数 $x_{ijt}$ は $l_{ij} \leqq t \leqq u_{ij}-1$ で有効である

$x_{it}$: 変数。 $x_{it}=1$（区間 $t$ でプロジェクトが完了済）

$=0$（その他）

また，Fig. 2 中の○印は，ジョブの最早完了時刻，△印はジョブの最遅完了時刻，◎印はプロジェクトの最早完了時刻に 1 を加えたもの，▲印はプロジェクトの最遅完了時刻を示している。

## §3. 問題の完式化

目的関数は，個々のプロジェクトの完了時刻の合計を最小にすることを目的としている。そのため，プロジェクトが完了した後に，絶対納期に対して残っている期間の数を最大化することに等しくなる。同時に個々のジョブの完了時刻を最早にすることも関数に含めてある。ここで，$M$ はプロジェクト完了時刻の和の最小化を，最初にはかるための重み係数である（$M>\sum_{i=1}^{I}\sum_{j=1}^{N_i} u_{ij}$）。

$$\text{Maximize} -\sum_{j=1}^{I}\sum_{j=1}^{N_i}\sum_{t=l_{ij}}^{u_{ij}-1} t x_{ijt} + \sum_{i=1}^{I}\sum_{t=e_i}^{G_i} M x_{it} \qquad (1)$$

制約条件は，ジョブの完了，プロジェクトの完了，ジョブ間の先行関係，資源の制限の 5 式になる。

ジョブの完了は次のように定式化される。

$$\sum_{t=l_{ij}}^{u_{ij}-1} x_{ijt} \leqq 1 \quad (i=1, 2, \cdots, I,\ j=1, 2, \cdots, N) \qquad (2)$$

プロジェクトの完了は，プロジェクト $i$ のすべてのジョブが時刻 $t$ で $\sum_{q=l_{ij}}^{t} x_{ijq}=1$ になったとき，プロジェクトが完了するという条件を示している。

$$\sum_{j=1}^{N_i}\sum_{q=l_{ij}}^{\min(t-1,\,u_{ij})} x_{ijq} + N_i x_{it} \leqq 0 \quad (i=1, 2, \cdots, I,\ t=e_i, \cdots, G_i) \qquad (3)$$

ジョブ間の先行関係は，1 つまたはそれ以上の他のジョブが完了するまで，あるジョブが実行を始めることが不可能な状態を示すものである。

$(i, m) \rightarrow (i, n)$ のとき，

$$-\sum_{t=1}^{u_{im}-l_{im}} (u_{im}-l_{im}-t+1)\, x_{im(t+l_{im}-1)} + \sum_{t=1}^{u_{in}-l_{in}} (u_{in}-l_{in}-t+1)\, x_{in(t+l_{in}-1)} \leqq u_{in}-u_{im}-d_{in} \qquad (4)$$

資源制約は，各時刻においてジョブに要求される資源の総量が，利用可能量を超えてはならないという定式化であり，ジョブに要求される資源は，そのジョブが終るまで使用され，作業の分割は許されないものとする。

$$\sum_{i=1}^{I}\sum_{j=1}^{N_i}\sum_{q=t}^{t+d_{ij}-1} r_{ijk}x_{ijq}\leqq R_{kt} \qquad (6)$$

$$\begin{pmatrix} t=\min\{a_{ij}+d_{ij}\}-1,\ \cdots,\ \max G_i \\ k=1,\ 2,\ \cdots,\ k \end{pmatrix}$$

## §4. Bounding-Algorithm

PATTERSON and HUBER は，次に示す区間$T$を用いて，Bounding-Algorithm を提案した。この方法は，Horizon-Varying 法ともよばれ，直接(0-1)整数計画問題を解くことはせずに，実行可能解が存在するか否かをある区間$T$から1ずつ区間を増しながら調べる。区間増加は，解が実行不可能の間繰返され，実行可能になった区間が確認されたときに，そこで最適解が得られている。

ここで，区間$T$の初期値は，$W$を次のように算出してから求める。

$$W=\max\left\{CP,\ \max_k\left[\frac{1}{R_k}\sum_{i=1}^{I}\sum_{j=1}^{N_i} r_{ijk}d_{ij}\right]\right\} \qquad (7)$$

$$T=[W]^+ \qquad (8)$$

$[W]^+$: $W$より大きい最小整数

$CP$: クリティカルパス長

この$T$が，スケジュール長の Lower-Bound となっている。以上の最適解を求めるアルゴリズムを要約すると以上のようになる。

(Step 1)　(7)，(8)により$T$を求める。

(Step 2)　$T$を絶対納期とした (0-1) 整数計画法を解く。

(Step 3)　解が実行可能なら最適解，さもなければ $T=T+1$ として (Step 2) へ。

ここで，(7)，(8)式で求めた$T$は，資源の延べ要求量をその資源の利用可能量で割って求めた値であるが，$W^*$, $T^*$ を次のように与えることにより，最適解により近い Lower-Bound を得ることが期待できる。

$W^E$ は，資源制約を入れないで，最早開始時刻において完了するスケジュールを求め，最早開始時刻で資源の山積みを行って得られた $t$ から(9)を用いて求める。

$$W^E=\max_k\left[t-1+\frac{A_k}{R_k}\right] \qquad (9)$$

$t$：資源要求量が利用可能量を超える時刻

$A_k$: 時刻 $t$ 以後の，資源 $k$ に対する延べ要求量

$W^L$ は，最遅完了時刻において完了するスケジュールを求め，最遅完了時刻で資源の山積みを行って計算した $t'$ を用いて10）から求める。

$$W^L=\max_k\left[(CP-t')+\frac{B_k}{R_k}\right] \tag{10}$$

$t'$: 最後に資源要求量の利用可能量を超える時刻

$B_k$: 時刻 $t'$ 以前の資源 $k$ に対する延べ要求量

$$W^*=\max[CP, W^E, W^L] \tag{11}$$

$$T^*=[W^*]^+ \tag{12}$$

$W^E$ については，資源要求量の最大となるところが，プロジェクトの後半の部分に存在している場合に有効になり，$W^L$ ついては，その逆の場合に有効である。

## §5. Matrix-Generator の作成

ACOS-6 のアプリケーション・プログラムである整数計画法システム MPS-6 を利用して問題を解くためには，MPS-6 の指定する特有の入力フォームをもった Matrix を作成する必要がある。この目的のために，Matrix-Generator を作成した。入力する項目は，ジョブ番号，所要時間，後続ジョブ番号，種類別資源利用可能量，ジョブの種類別資源要求量（ジョブ数だけ存在する），プロジェクト $i$ のジョブ数，プロジェクト $i$ ジョブ $j$ の到着時刻，プロジェクト数である。

Table 2 Elements of the input Matrix.

| | | $X1$ $X2$ $X3$……$XN$ | RHS |
|---|---|---|---|
| | | Coefficients of objective function | |
| ↓ $M$ | $V1$<br>$V2$<br>⋮<br>$VM$ | Constraint coefficients | Right Hand Side |

$N$: number of (0-1) variables
$M$: number of constraints
$X1, X2, \cdots, XN$: name of variables
$V1, V2, \cdots, VM$: name of constraints

Matrix には，Table 2 に示す要素が必要である。この Matrix から，整数計画法解法システムの入力フォームを作成している。次に，2×2の整数計画問題に対する入力フォーム作成例を Table 3 で示す。

Table 3 MPS-6 input form for 2×2 problem.

Objective function　$OBJ$: $5x_1+3x_2 \xrightarrow{\text{RHS}} \text{Max}$

Constraints　$\begin{cases} V_1 : x_1+ x_2 \leqq 7 \\ V_2 : x_1- x_2 \leqq 4 \end{cases}$

$x_1$ and $x_2$ are integer variables of $0 \leqq x_1, x_2 \leqq 5$

<Input form>

| 1 | 8 |
|---|---|
| FILE | REIDAI |
| LGL | $OBJ$ $(F)$ |
| LGL | $V1$ $(P)$ |
| LGL | $V2$ $(P)$ |
| STR | $X1$ (INTEGER=0, 5) |
| STR | $X2$ (INTEGER=0, 5) |
| MATRIX | $OBJ, X1=5$ |
| MATRIX | $OBJ, X2=3$ |
| MATRIX | $V1, X1=1$ |
| MATRIX | $V1, X2=1$ |
| MATRIX | $V2, X1=1$ |
| MATRIX | $V2, X2=-1$ |
| RHS | $V1$, RHS=7 |
| RHS | $V1$, RHS=4 |
| END*** | |

| | |
|---|---|
| FILE | Define file name. |
| LGL | Define logical variables. |
| STR | Define structure variables. |
| MATRIX | Define elements of matrix. |
| RHS | Define right hand side. |
| END*** | End card of file deck. |
| (PLUS) or (P) | ≦ |
| (FREE) or (F) | Indicate logical variables which can take any value. |
| (INTEGER) | Indicate integar variables and give the interval where integer variable can take. |

## §6. 実行可能解の判定とその自動処理流れ図

資源制約をもつプロジェクトスケジューリング問題の最適解は，Fig. 3 の流れ図により求めることができる。

この処理により，最適解が得られるまで $T^*$ を 1 ずつ増すとともに，Matrix を更新し，(0-1) 整数計画問題の解が実行可能であるか否かを自動的に判定することが可能になる。

解が非許容と判定されたときに，$T^*$ が増されることで Matrix を構成する要素は更新され，その要素数は急速に増加していく。また，解の判定を行う計算時間は，初期には，整数制約をはずした一般 LP の段階ですでに非許容となることが多く比較的短い。しかし，実行可能解に近く

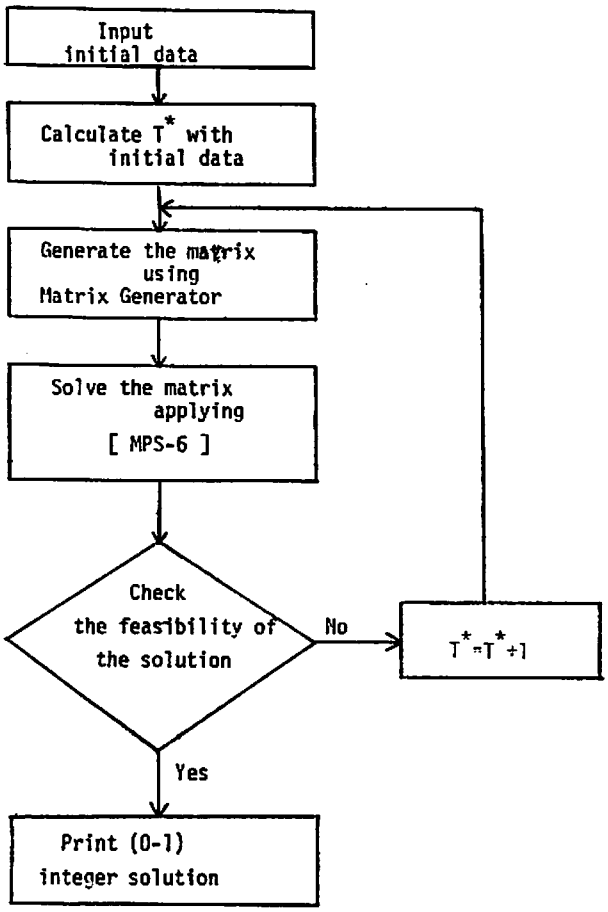



Fig. 3 Flow chart of the proposed method.

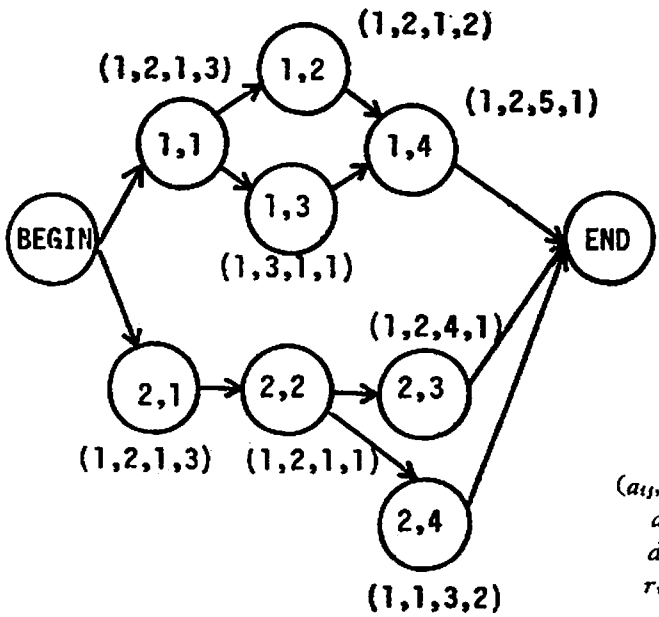
なると，この LP では実行可能となり次に，Branch-and-Bound による (0-1) 整数解を探索する計算が始まり，多大な時間が要求されてくる。これは MPS-6 のアルゴリズム自体に依存して生じる結果である。

## §7. 計算結果と評価

問題解法例として，各々4個のジョブを含む2つのプロジェクトを有するスケジューリング問題を Fig. 4 に示す。

また，Table 4 に示す計算結果が問題に対して得られている。改良法においては，PATTERSON and HUBER が行った初期区間 $T$ よりも 2 step 短縮された区間 $T^*$ が得られた。これにより最適解までは，2 step で到達できる。

しかし，$T=10$ の区間において，Branch-and-Bound 計算による (0-1) 整数解の探索に，多大な時間を費していることが解かる。




$(a_{ij}, d_{ij}, r_{ij1}, r_{ij2})$
$a_{ij}$; arrival period of job $j$, project $i$.
$d_{ij}$; job duration of job $j$, project $i$.
$r_{ijk}$; amount of type $k$ resource

Fig. 4 Network of test problem of eight jobs in two projects requiring two resources.

Table 4 Calculation time, number of constraints and number of variables on each $T$ and $T^*$ period

| Calculation time \ Method | | $T$ (Patterson method) | $T^*$ (Proposed method) |
|---|---|---|---|
| Calculation time to get initial value | | 1.296 | 1.305 |
| $T=7$<br>$M=30$<br>$N=7$ | MAT. GEN<br>MPS-6<br>others | 1.679<br>2.563<br>2.094 | |
| $T=8$<br>$M=34$<br>$N=17$ | MAT. GEN<br>MPS-6<br>others | 1.818<br>3.284<br>2.339 | |
| $T=9$<br>$M=38$<br>$N=27$ | MAT. GEN<br>MPS-6<br>others | 2.199<br>4.283<br>2.487 | 2.186<br>4.250<br>2.512 |
| $T=10$<br>$M=42$<br>$N=37$ | MAT. GEN<br>MPS-6<br>others | 2.489<br>*49.181<br>5.146 | 2.474<br>*48.772<br>5.053 |
| Total | | 1 : 20.858 | 1 : 06.552 |

$T$: time period
$M$: number of constraints
$N$: number of variables
MAT. GEN: Matrix-Generator
MPS-6: Application paccage of LP

一方，節約された $T=7, 8$ の区間での，MPS-6 による非許容の判定に必要とする計算時間はこれに比してわずかである。

次に，9種類の問題に対して計算を行った結果を Table 5 に示す。

いずれの問題に対しても，$T^*$ による改良で PATTERSON and HUBER による $T$ の区間を短縮させることができた。しかし，始めの頃の Step に比べて，最適解付近での Branch-and-Bound の計算時間が大きいので，総所要時間では，期待される程の差が得られていない。

Table 5 Results of nine test problems

| Test problem number | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 9 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| Size of the problem | Number of jobs | 9 | 13 | 8 | 8 | 7 | 7 | 10 | 11 | 8 |
| | Number of resources | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| Size of the final matrix | | 57×48 | 50×49 | 42×37 | 33×32 | 45×37 | 37×24 | 42×18 | 53×54 | 42×33 |
| Patterson & Huber's method | Number of steps | 3 | 2 | 4 | 4 | 5 | 3 | 2 | 2 | 4 |
| | Processing time* | 47.07 | 3:38.51 | 1:46.33 | 51.39 | 3:18.17 | 30.19 | 20.63 | 2:34.54 | 1:22.69 |
| Proposed method | Number of steps | 2 | 2 | 2 | 3 | 3 | 2 | 1 | 1 | 3 |
| | Processing time* | 38.01 | 3:38.00 | 1:31.22 | 45.55 | 3:04.83 | 23.63 | 14.16 | 2:24.23 | 1:15.46 |

* Processing time is given by (minuit: second)

## §8. ま と め

Matrix-Generator を作成したことにより，簡単な入力方法で，既存の(0-1)整数計画法を利用して資源配分問題を解くことが可能になった。

更に，試算したところの4種類の問題に対しても，PATTERSON and HUBER の提案による$T$に対して，改良した$T^*$が，より最適解に近い区間の値をとり，$T$の Step を減少させることにおいて有効であることが確かめられた。

しかし，最適解の近くに$T$が近づくと，Branch-and-Bound に入るので，(0-1) 整数解を探索する計算が，1回の Step に非常に多くの時間を費やす。そのため，Step 数の短縮が，総所要時間においての期待された向上にはつながらなかった。

この問題点を解決するためには，今後，整数計画法での非許容解の判定がさらに速く行えるアルゴリズムが必要であり，またプロジェクト・スケジューリング問題の定式化に適応した解法を導入することが課題である。

### 参考文献

1) A. Alan, B. Pritsker, Lawrence J. Watters and Philip M. Wolfe, Multi Project Scheduling with Limited Resources, Management Science, 1969. 9, Vol. **16**, No. 1.
2) James H. Patterson and Walter D. Huber, A Horizon-Varying, Zero-One Approach to Project Scheduling, Management Science, 1974. 2, Vol. **20**, No. 6.
3) 山本正明，福馬敏子，(0-1) 整数計画法によるネットワーク・プロジェクト・スケジューリング問題の研究，1982. 5，経営工学会春季大会