探索木を用いた優先順位付き割当問題のための近似解法

錦織昭峰　渡辺展男　*一森哲男　青木兼一　金指正和　伊藤雅
広島県立大学　*大阪工業大学

本論文では、利益あるいは費用が金額として数値的に明確に与えることができない割当問題に対して、優先順位に従った割当をした解を求めるための近似解法を提案する。提案法では、(i) 単調減少しにくい実行不可能和の最小化を行うためには、どのような複数の整数変数を一度に変化させるべきかを探索木を用いて調べており、(ii)得られた探索によって逐次割当を変更する際には、変更したところ以外の探索木のアークは残しておいて、次の探索の際の情報として用いており、(iii) 割当状態を変更した結果不要となるアークの削除個数をできるだけ少なくしている。大規模な割当問題を用いて数値シミュレーション実験を行った結果、提案法は効率的な方法であることが示された。

AN APPROXIMATION METHOD USING SEARCH TREES FOR SOLVING ASSIGNMENT PROBLEMS WITH PRIORITY ORDER

Akimine Nishikori　Norio Watanabe　*Tetuo Ichimori　Kenichi Aoki　Masakazu Kanezashi　Masaru Itoh

Hiroshima Prefectural University　*Osaka Institute of Technology
Shobara, Hiroshima 727, Japan
*Osaka 535, Japan

This paper presents an approximation method for obtaining a feasible solution of an assignment problem with priority order. The proposed method is applicable to assignment problems whose cost coefficients cannot clearly be determined. In the method, the following three procedures are iteratively carried out to decrease the sum of infeasibilities ; (i) a set of integer variables to be changed is chosen by means of search trees, (ii) the trees are modified after the integer variables are changed, and (iii) the assignment is altered so that the number of deleted arcs is as small as possible. The method has proved to be efficient as a result of applying it to large-scale assignment problems.

1．まえがき

割当問題は、整数変数が0あるいは1の値しかとらない0－1整数計画問題[1,2] の一種である。この割当問題では、ある期間帯 i（あるいは場所 i）に、ある資源 j（例えば、人数、設備、資金など）を割り当てたときの、「利益」（あるいは「コスト」）C_{ij} を目的関数の利益係数として用いている。この係数 C_{ij} が、「利益」あるいは「コスト」という明確な金額で表すことができず、単に、ある期間帯 i に、ある資源 j を割り当てる「重要度」あるいは「優先度」という「重み付け」を表している場合がある。従来このような場合には、資源 j に対して、期間帯 i の重要な順

$$i_1,\ i_2,\ \cdots,\ i_m$$

に対して、目的関数に用いられる利益係数 C_{ij}を

$$C_{i_1 \cdot j} > C_{i_2 \cdot j} > \cdots > C_{i_m \cdot j}$$

となるように適当な比率を設けて重要な順に重み付けを行って、できるだけ重要度を満足させてきた[3,4]。

しかしながら、このような割当問題で用いられるのは0－1整数変数であるので、資源の割当を表す変数の値は0あるいは1で求まり、必ずしも重み付けに比例した値が求められない。すなわち、優先度は本来数値化すること自体が困難であり、優先度をどのように重み付けしたらよいという明確な基準がない。

このような場合、割当をする資源が人であり、その人の満足度を最大化する場合には、各人の満足度の総計を各々１００％として、各人の好みに応じてその１００の重みを各期間帯の利益 C_{ij} に案分して定式化すれば良い[5]。しかしながら、一般には、すべての人の満足度の総計が同じであるとは限らず、資源自身も人を表している場合だけではない。

ところで、実際問題では、ある期間帯に、どのような資源の割当をするかしないかという候補が事前に決まっており、更にその候補の中にも、好ましい割当候補と、そうでない割当候補があることが多い。このような場合には、予め決められた複数の割当候補の中で、割当を要望する順序、すなわち「優先順位」に従った割当を行ったほうが良い。このような点を考慮した割当を行う問題を、以降、「優先順位付き割当問題」と呼ぶことにする。この問題は、優先順位を考慮した割当を行って、多数の実行可能解のうちで、ある一つの実行可能解を求める問題となる。筆者らは、電力システムにおいて、このような優先順位付きの最適化問題を考察してきた[6]。従来、利益あるいは費用が金額として明確に与えられた場合に割当を行う方法[1,2,3]は多数提案されているが、優先順位付きの割当を行う方法は未だに提案されていない。

また、実行可能解を求めるための方法[2,11] に関して述べると、従来では、不等式制約式の実行不可能和を評価関数にして、整数変数を変化させてその和を減少させて零にしていた。しかしながら、評価関数は多数の実行不可能量の和として表されるため、整数変数を変化させて、ある制約式の実行不可能量を減少させても、別の制約式の実行不可能量が増加することが起こって、非増加ではあるが単調減少しにくくなることがある。このようなときには、更に続いて他のどのような整数変数を同時に変化させることが必要かを組織的に探索してみなければならない。

本論文では、利益あるいは費用が金額として明確に与えられていない割当問題を解く際に、割当を行う順番を表す優先順位を設定し、その優先順位に従った割当を行って実行可能解を求めるアルゴリズムを提案する[7]。提案法[8,9] では、

・単調減少しにくい実行不可能和の最小化を行うためには、どのような複数の整数変数を一度に変化させるべきかを複数の探索木を用いて調べており、

・得られた探索によって逐次割当を変更する際には、変更したところ以外の探索木のアークは残しておいて、次の探索の際の情報として用いており、

・割当状態を変更した結果不要となる探索情報をもつアークの削除個数をできるだけ少なくしている。

このアルゴリズムでは、実行不可能和を最小化する整数計画問題を繰り返し解く必要があるが、整数変数の個数が非常に多い大規模な優先順位付き割当問題では、真の最適解を効率よく求めることが困難であることが知られている[1,2]。市販の大型計算機により解ける問題の大きさの限界を述べると、現在のところ、連続変数と整数変数の両方を含む混合整数計画問題では、0－1整数変数の個数が４５０個程度までならば真の最適解を求め、また、２１００個程度までならば実行可能解が求められている[10]。従って、本研究では、現実に起こる非常に大規模でスパースな実際問題にも対応できるようにするために、真の最適解を求めることはあきらめて、満足できる実行可能解を少ない計算量で求めることができる近似解法を開発して、提案法で用いている。

この割当問題を解くための簡易的な方法として、ある期間帯に割り当ててあった資源の一つを、他の期間帯へと割当を逐次変更する方法[7] がある。これは、二つの整数変数を一度に変化させる２－最適化法[2] による近似解法である。提案法では、二つ以上の資源の割当を一度に変更することも考慮しているので、このような「一割当変更法」よりも良い解を求めることができる。

本論文での提案法を用いて、乱数により作成した数万個程度の整数変数をもつ大規模な割当問題を解いた結果、実用的な計算時間内に、優先順位を考慮した実行可能解を求めることができた。

2．制約条件

対象とする割当問題の制約条件を、以下に記す。

$$\sum_{j \in \Gamma^{il}} x_{ij} \leqq b_{il} \qquad l \in \Lambda^i \ ; \ i=1,2,\ldots,m \qquad (1)$$

$$\sum_{i=1}^{m} x_{ij} = a_j \qquad j=1,2,\ldots,n \qquad (2)$$

$$x_{ij} = 0 \text{ or } 1 \qquad i=1,2,\ldots,m \ ; \ j=1,2,\ldots,n \qquad (3)$$

ここで Λ^i と Γ^{il} は添字の集合、b_{il}, a_j は正の整数であり、

$$x_{ij} = \begin{cases} 1 & \text{期間帯 } i \text{（すなわち節点 } i \text{）に資源 } j \text{ を割り当てるとき} \\ 0 & \text{そうでないとき} \end{cases}$$

である。以降、提案法を説明する際には、「期間帯 i

に割り当てる」という代わりに、「節点 i に割り当てる」ということもある。

この定式化は、次に述べるような二つの点から、普通の割当問題の制約条件を一般化している。

(i) 普通の割当問題では、 (2)式の a_j はつねに 1 であるが、上記の定式化では一般には 1 でない。これは、同一の資源が複数個ある場合も考慮するためであり、各期間帯 i と各資源 j は一般には 1 対 1 に対応していない。従って、提案法では、同一複数個の資源 j を、異なった複数の期間帯 i に一つずつ割り当てることを考えている。

(ii)各期間帯 i へ、異なった複数の資源を割り当てることも考慮するが、割り当てることができる資源の総数に上限を設ける。この際に、各資源には種類（あるいはタイプ）があるならば、その種類毎に、割り当てられる資源の総数に上限を設ける。それ故、 (1)式のように各期間帯 i 毎に制約式番号を表す添字 l を設けて、複数の制約式を考慮できるようにしている。ここで、各 i での制約式の数は $|\Lambda^i|$ である。

割当問題[1,2] の評価関数で用いられる利益係数 C_{ij} が割当の「優先度」あるいは「重要度」を表している例として、

(i) 講義室の数とその大きさに種類がある場合に、各時間帯毎に、各先生の授業科目を開講する講義室を決定する時間割の問題[4,11] 、

(ii)設備などの資源制約を満たしつつ、複数の作業期間帯に、何種類かの仕事を割り当てる生産計画の問題[10]

などがある。

3．提案法の概要

すべての資源 j = 1, 2,･･･, n の第 1 候補が異なっており、その結果すべての期間帯 i = 1, 2,･･･, m に偏りなく割り当てられれば良い。しかしながら、多数の資源の第 1 候補がいくつかのある特定の期間帯に集中したときには、第 2 候補以下に繰り下げなければならない。この繰り下げが、多数の期間帯で起こる場合には、各々の資源を割り当てた期間帯の優先順位ができるだけばらつかないように考慮しつつ、各々の資源毎の優先順位にできるだけ添った割当を決定するために、各資源間での割当の調整を行うことが必要となる。

提案法では、(2), (3) 式を満足させながら、節点 i への資源 j の割当を変更していく。いま、節点 i に割当られている資源 j の集合を X^i と表す。すなわち、

$$X^i = \{ j \mid x_{ij} = 1 \ ; \ j=1,2,...,n \} \tag{4}$$

制約条件の実行不可能量の総和を

$$f(X) = \sum_{i=1}^{m} f^i(X^i) \tag{5}$$

と表す。ここで、

$$f^i(X^i) = \sum_{l\in\Lambda^i} \max \{ (\sum_{j\in\Gamma^{il}} x_{ij} - b_{il}), 0 \} \tag{6}$$

本研究での提案法の概要を、図 1 に示すような例を用いて説明する。この例では、節点 i = 1, 2 に、それぞれ、資源 j = 1, 2, 3 と j = 3, 4, 5 がすでに割り当てられている。

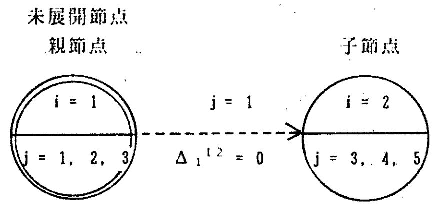

図 1　割当変更のための展開

更に、節点 i = 1, 2 の制約条件として、それぞれ、

$$x_{11} + x_{12} + x_{13} \leq 2 \tag{7}$$

$$x_{11} + x_{13} + x_{14} \leq 1 \tag{8}$$

及び

$$x_{21} + x_{22} + x_{23} + x_{24} \leq 2 \tag{9}$$

$$x_{21} + x_{22} + x_{24} + x_{25} \leq 2 \tag{10}$$

があるとする。このとき節点 1 の実行不可能量 $f^1(X^1) = 2$ であり、節点 2 は $f^2(X^2) = 0$ である。ここで $X^1 = \{1, 2, 3\}$, $X^2 = \{3, 4, 5\}$ である。実行不可能である節点 1 は◎で表すことにする。いま、節点 1 に割り当てられている資源 1, 2, 3のうちのどれかを、他の節点へ移すことを考える。これを、「未展開節点 1 を展開する」ということにする。例えば、資源 1 を節点 1 から 2 へ移動するならば、節点 1 、 2 の実行不可能量の変化量をそれぞれ ${}_P\Delta_1^1$, ${}_C\Delta_1^2$ と表すと、それぞれ-2 と 2 となり、結局実行不可能和の変化量 Δ_1^{12} は

$$\Delta_1^{12} = {}_P\Delta_1^1 + {}_C\Delta_1^2 = 0 \tag{11}$$

となり変化しない。ここで、節点 1 を親節点、節点 2 を子節点と呼ぶことにする。次に、資源 1 を節点 1 から 2 へ移した後、更に、節点 2 にすでに割り当てられている資源を、別の節点 3 に移すことを考える。これを、「深さ 2 で未展開節点 2 を展開する」ということにする。いま、図 2 のように、節点 3 には、資源 1, 2, 3, 5 が割り当てられており、制約条件として、

$$x_{32} + x_{33} + x_{34} + x_{35} \leq 3 \tag{12}$$

があるとする。

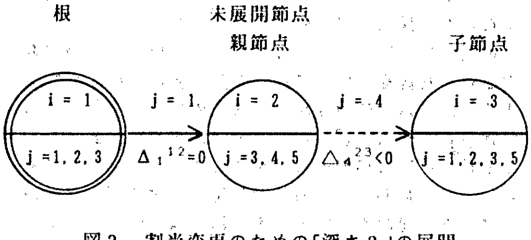

図 2　割当変更のための「深さ 2 」の展開

例えば、資源 4 を節点 2 から 3 へ移すならば、節点 2 と 3 の実行不可能量の変化量は、それぞれ、-2, 1 であるので、結局実行不可能和の変化量 Δ_4^{23} は -1 となり減少する。従って、実行不可能和 f(X) が減少することがわかったので、実際に、節点 1 から 2 へ資源 1 の割当を変更し、節点 2 から 3 へ資源 4 の割当を変更する。これを、「根節点 1 から節点 3 までのパスによる割当の同時変更」ということにする。

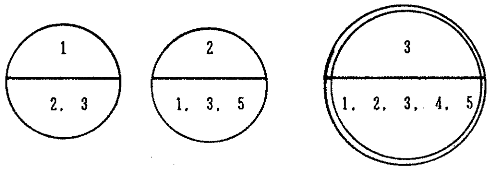

図３　根節点１から節点３までのパスによる割当変更

このような割当の変更を行った結果、節点３では逆に実行不可能になったので、節点３を根とする新しい探索木を設けて、節点３を未展開節点とする。しかしながら、実行不可能和を単調に減少させている。

以下同様にして、制約を違反した◎の節点ｉを探索木の根とし、ある親節点に割り当てられていた資源を、別の子節点への割当に変更を考えてみるならばアーク → を付けるという手続きを繰り返して探索していく。そして、探索した情報を基にして、実行不可能量が零でない節点に割り当てられている資源を、他の節点へと移動することにより、すべての節点での実行不可能量を零にする。文献[9] を参照。

４．アークの削除

提案法では、複数の探索木を構成して、どの資源の割当を変更すべきかを探索して調べ、探索したときの情報をアークとして保存している。ある探索木によって割当を変更した結果、資源の割当状態が変化したために、他の探索木で調べた探索情報が役に立たなくなることがある。探索が進むにつれて探索木が大きくなると、役にたたなくなって削除される既存のアークの個数が多くなる。それ故、提案法では、既存アークの削除個数が少ない割当変更から調べている。

図４の例を用いて、あるパスによる割当変更に伴って、不要となるアーク個数の算定方法を以下に記す。この例では、節点１～18、資源 j_1～j_4, j_a などがある。

$f^i(X^i) > 0$ であるような節点ｉを根にもつ探索木を T^i と表し、すべての探索木の集合をＴとすると、

$$T = \{ T^i \mid f^i(X^i) > 0 \} \qquad (13)$$

また、ある探索木 T^k の有向アークを e^{ks}（ｓはアークの通し番号）と表し、このアークに添って割当が変更される資源を j^{ks} 、アークの矢印側の節点を i^{ks} と表す。例えば、節点３から５までのアーク e^{34} に関しては、$j^{34} = j_1$、$i^{34} = 5$ である。ただし、$s = 0$ のときには i^{ks} は根節点を表すことにし、$i^{k0} = k$ である。木 T^k において、節点 i^{ks} を根とする部分木にあるアークの総数を u^{ks} と表す。例えば、$u^{34} = 3$ である。根節点ｋから未展開節点 i^{ks} までのパス上の節点を

$$i^{ks0}\ (= k),\ i^{ks1},\ \ldots,\ i^{kst}\ (= i^{ks})$$

とする。図の例では、未展開節点 $i^{37} = 8$ までのパス上の節点は、 3, 5, 6, 8 である。更に根節点 i^{ks0} から出るアークのうちで、i^{ks0} から i^{ks1} に割当を変更する資源 j^{ks1} と同じ資源を変更するアークの番号の集合を O_1^k とし、異なる資源を変更するアークの番号の集合を O_2^k と定義する。例えば、$O_1^3 = \{3\}$, $O_2^3 = \{1, 2\}$ である。ただし、$t = 0$ すなわち根節点ｋ自身が未展開節点である場合には、この未展開節点から子節点に移す資源が未定であるので、$O_1^k = O_2^k = \phi$（空集合）とする。次に、パス上の節点 $i^{ks0}, i^{ks1}, \ldots, i^{kst}$ のうちから、 T^k 以外の探索木 T^h にある節点と同じ節点を選び出す。例では木 T^2 の節点６と８である。同様に、探索木 T^k 自身についても、i^{ks} $(s \varepsilon O_2^k)$ を根とする部分木に対しても同様に選び出す。例では節点６である。選び出された各節点の親節点を根とする部分木をつくり、重なり合う部分木を取り除いた後、部分木の根節点に入るアークの通し番号の集合を D_h^{ks} と表す。ただし、根節点の場合にはその親節点がないので、その木全体を部分木として取り扱う。図の例では、$D_2^{37} = \{2, 3\}$, $D_3^{37} = \{2\}$ である。

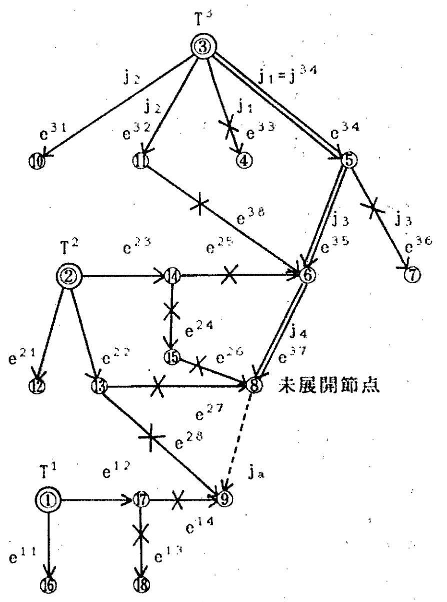

図４　節点３から９までのパスによるアークの削除

以上の定義より、未展開節点 i^{ks} までのパスによって割当を変更した際に、すべての探索木で削除されるアークの総数 v^{ks} は次のように表される。

$$v^{ks} = \Big\{ \sum_{d \varepsilon O_1^k} (u^{kd} + 1) - t + \sum_{d \varepsilon D_k^{ks}} u^{kd} \Big\} \quad (\text{第１項})$$

$$+ \sum_{\substack{h \varepsilon T \\ h \neq k}} \sum_{d \varepsilon D_h^{ks}} u^{hd} \quad (\text{第２項}) \qquad (14)$$

ただし、このパス上のある節点 i^{ksg} が

$$i^{ksg} = k\ (g \neq 0) \quad または \quad i^{ksg} \varepsilon O_2^k$$

である場合、または、この割当変更の結果節点ｋでの実行不可能量 $f^k(X^k)$ が零となる場合には、第１項は $(u^{k0} - t)$ とする。 図の例で削除されるアークのうち、第１項については e^{33}, e^{36}, e^{38} であり、第２項については e^{24}～e^{28} である。

実際にはこのパスのみならず、資源を移した子節点

によってもアークが削除される。パスあるいは子節点 i により削除されるアークの総数を z_i^{k*} ($\geq v^{k*}$) とすると、この割当変更によるアークの削除率は

$$z_i^{k*} \;/\; \Big(\sum_{h \in T} u^{h0} \Big) \tag{15}$$

と表される。この分母はすべての探索木のアークの総数を表している。例では、×印を付けたアークが削除されるので、削除率は 10 / 20 = 50 % である。

5．余裕量の算定

(1)式の中で活性 (active) な制約式をもつ節点に更に資源を割り当てると、上限値 b_{il}をこえるので、その節点には余裕がなく、割り当てたい資源があってもこれ以上割り当てられない。そこで、活性な制約をもつ節点にすでに割り当てられている資源を、割当可能な他の節点へと割当変更した結果、活性な制約をもつ節点に新たに割り当てられるようになる余裕量をΘと表すことにする。

ある優先順位の下で実行不可能量の和 f(X) を最小化する際には、最初に、1資源ずつの割当変更のみを行う。これをSTAGE 1 とする。STAGE 1 が終わった段階で、余裕量Θの値を求める。このΘの値が f(X) の値よりも大きくなった場合には、更に3章で示したような2資源以上の割当の同時変更を行えば、 f(X) の値が更に減少して零になることがある。それ故、$\Theta \geq f(X)$の場合には、2資源以上の割当を同時に変更するSTAGE 2 を行う。このSTAGE 2 の実行中に、多数のアークを新たに作成しても f(X) が減少しなくなり零にならないならば、このSTAGE 2 を終了する。

余裕量Θは、活性な制約式のみならず、実行不可能な制約式もまた考慮して、以下のようにして求める。各優先順位の下でのSTAGE 1 が終わった段階で、節点 i に割り当てられている資源の集合 X^i に対して、(1) 式の中で、節点 i での活性な制約式の添字 l の集合を$\Lambda_A{}^i$と表す。ある節点 i_P から、割り当てられていた資源 j_a を取り去り、ある節点 i_C へと割当を変更したとすると、節点 i_P の活性である制約式へ、新たに割当可能となる量 Δ_Aは

$$\Delta_A = \sum_{l \in \Lambda_A{}^{i_P}} \Big\{ b_{i_P, l} - \sum_{\substack{j \in \Gamma^{i_P, l} \\ j \neq j_a}} x_{i_P, j} \Big\} \geq 0 \tag{16}$$

である。すなわち、Δ_A は集合$\Gamma^{i_P, l}$ ($l \in \Lambda_A{}^{i_P}$) に属する資源 j_a の個数に等しい。この際の、節点i_P と i_C の実行不可能の変化量 ${}_P\Delta_{ja}{}^{i_P}$ と ${}_C\Delta_{ja}{}^{i_C}$ の和を、(11)式で説明したように、 $\Delta_{ja}{}^{i_P i_C}$ と表すことにする。ここで、符号を考慮して、

$$\theta = \Delta_A + (-\Delta_{ja}{}^{i_P i_C}) \tag{17}$$

と定義する。もし、

$$\theta > 0 \quad かつ \quad \Delta_{ja}{}^{i_P i_C} \leq 0 \tag{18}$$

ならば、このような割当変更をする場合には、実行不可能和 f(X) を増加することなく、この節点 i_P にθだけ余裕ができる。ただし、制約式の実行不可能量 $f^{i_C}(X^{i_C})$ が増加しない節点 i_C を選ぶこととする。ある優先順位の下で割当可能な節点をすべて調べて、このような節点 i_C がなければ、この資源 j_a の余裕量$\theta = 0$ とする。($\sum_j a_j$) 個のすべての種類のすべての資源の各々に対して、資源が割り当てられている節点 i_P から割当を変更する場合に伴う余裕量θを求めて、その余裕量の総和をΘとおく。なお、${}_C\Delta_{ja}{}^{i_C}$ の値を求める際には、このような割当変更を考慮する毎に、節点 i_C には資源 j_a が新たに割り当てられているとしている。

6．提案法

提案法の概要で述べたように、探索木で削除されなかった節点i^{k*} から、割当が実際に修正された節点(すなわちパス上の節点)以外の節点へと、ある資源 j の割当を変更することを一度調べ終わっている場合には、優先順位を一つ増やしても再度調べ直す必要がない。この事を考慮するために、第 r 位の優先順位を新たに加えて割当の変更を調べる際に、以下に記すような集合を用いる。ある探索木 T^k に対して、

- $\Omega_N{}^k$は、まだ一度も展開したことがない節点i^{k*} の集合
- $\Omega_A{}^k$は、一つ前の (r-1)位の優先順位で割当の変更を調べた際に展開したことがあるけれども、再度展開してみる節点i^{k*} の集合

を表す。このとき、すべての未展開節点を要素とする集合Ωは、

$$\Omega = \cup_h \; [\; \Omega_N{}^h \cup \Omega_A{}^h \;] \tag{19}$$

となる。また、

- Q^{k*} は、(r-1) 位までの優先順位で、節点i^{k*} を展開した際にできた、すべての子節点の集合

を表すことにする。更に、各資源 j に対して、m個の期間帯を r^{max} 個に分割し、

- I^{jr} は、第 r 位の優先順位で割り当ててよい期間帯 i の集合

を表すことにする。ここで各I^{jl} ($l=1, 2, \cdots, r^{max}$) は、同じ要素 i を含まず、

$$\bigcup_{l=1}^{r^{max}} I^{jl} = \{ 1, 2, \cdots, m \} \quad j=1, 2, \cdots, n \tag{20}$$

である。また、

- I_R は、第 (r-1)位の優先順位を加えて割当の変更を調べた際に、実際に割当を修正したパス上の節点 i の集合

を表すことにし、

$$I_N{}^j := \cup_{l=1}{}^{l=r} \; I^{jl} \tag{21}$$

$$I_A{}^j := I^{jr} \cup [\; (\cup_{l=1}{}^{l=r-1} \; I^{jl}) \cap I_R \;] \tag{22}$$

とすると、

- $I_N{}^j$ は、1位から r 位までの優先順位で割当可能な節点 i の集合
- $I_A{}^j$ は、r 位の優先順位で割当可能な節点、または、1位から (r-1)位までの優先順位で割当可能な節点のうち, (r-1)位の優先順位で実際に割当を修正したことがある節点 i の集合

である。

以上の集合を定義することにより、アルゴリズムは以下のようになる。

<u>ステップ0：初期状態</u>

各資源を優先順位第1位の期間帯に割り当てる。

$f^i(X^i) > 0$、すなわち、制約式が実行不可能である節点 i を根とする探索木 T^h を設けて、

$$\Omega_N{}^h := \{ i \} \; ; \; \Omega_A{}^h := \phi \qquad (23)$$

とし、すべての h に対して $Q^{h0} := \phi$ とする。また、

$$I_N{}^j := I^{j1} \cup I^{j2} \; ; \; I_A{}^j := I^{j2} \; ; \; I_R := \phi \qquad (24)$$

とし、優先順位 r = 2 とする。ARC := 0, ARC^U := n とし、STAGE = 1 とする。

ステップ１：展開節点の選択

未展開節点の集合Ωの中から、削除されるアーク数が最小値 v^{k*} である未展開節点 i^{k*}（この探索木番号は k）を求めて、$i_P := i^{k*}$ とする。

ステップ２：親節点の展開

未展開節点 i_P に対して、${}_P\Delta_{ja}{}^{iP} < 0$ となる資源 j_a の集合を $J_{ja}{}^{iP}$ と表す。ここで、

$${}_P\Delta_{ja}{}^{iP} = f^i({}_PX_{ja}{}^{iP}) - f^i(X^{iP}) \qquad (25)$$

$${}_PX_{ja}{}^{iP} = X^{iP} - \{ j_a \} \qquad (26)$$

ステップ３：子節点の決定

すべての資源 $j_a \varepsilon J_{ja}{}^{iP}$ に対して、$\Delta_{ja}{}^{iPiC}$ を計算する。ここで、

$$\Delta_{ja}{}^{iPiC} = {}_P\Delta_{ja}{}^{iP} + {}_C\Delta_{ja}{}^{iC} \qquad (27)$$

$${}_C\Delta_{ja}{}^{iC} = f^i({}_CX_{ja}{}^{iC}) - f^i(X^{iC}) \qquad (28)$$

$${}_CX_{ja}{}^{iC} = X^{iC} + \{ j_a \} \qquad (29)$$

ここで子節点 i_C は、

$i_C \; \varepsilon \; I_N{}^j \qquad i_P \; \varepsilon \; \Omega_N{}^k$ のとき

$i_C \; \varepsilon \; I_A{}^j \qquad i_P \; \varepsilon \; \Omega_A{}^k$ のとき

である。

ステップ４：割当変更の判定

ステップ３で $\Delta_{ja}{}^{iPiC} < 0$ となる $(i,j)=(i_C, j_a)$ があるならば、次のステップへ。さもなければ、STAGE = 1 ならばステップ８へ、STAGE = 2 ならばステップ７へ。

ステップ５：割当の変更

根節点から節点 i_C までのパスによって資源の割当を変更して、X^i を修正する。探索木 T^h と未展開節点の集合Ωから、不要になった節点を削除する。その結果、新たに制約式が実行不可能になった節点 i' に対しては、新しい探索木 T^h を設けて $\Omega_N{}^h = \{ i' \}$ として i' を未展開節点とする。$\Omega_A{}^h = \phi$, $Q^{h0} = \phi$ とし、パス上の節点 i^{kg*} (g=0, 1, ⋯, t) と子節点 i_C を集合 I_R に加え、v^{k*} を修正する。ARC := 0 とする。

ステップ６：解の判定

探索木が一つもない、すなわち、f(X) = 0 ならば、実行可能解が求まったので終了する。さもなければ、ステップ８へ。

ステップ７：新たなアークと未展開節点

$$\Omega_A{}^k := \Omega_A{}^k \cup \{ i^x \mid i^x \; \varepsilon \; Q^{k*} \} \qquad (30)$$

として、i^{k*} のすべての既存の子節点 i^x を未展開節点とする。また、ステップ３で $\Delta_{ja}{}^{iPiC} = 0$ となる $(i,j)=(i_C, j_a)$ があるならば、i_P から i_C までのアークを新たに設けて、

$$\Omega_N{}^k := \Omega_N{}^k \cup \{ i_C \} \qquad (31)$$

として、すべての新しい子節点 i_C を未展開節点とする。新たにできたアークの個数を ARCに加える。Q^{k*} と v^{k*}を修正する。

ステップ８：展開節点の変更

未展開節点の集合Ωが空であるか、または STAGE = 2 において ARC ≧ ARC^U ならば、次のステップへ。さもなければ、ステップ１へ。

ステップ９：余裕量との比較

STAGE = 2 ならば次のステップへ。STAGE = 1 のとき

$$余裕量\Theta \geqq f(X) \qquad (32)$$

ならば、STAGE = 2 としてステップ１２へ。さもなければ次のステップへ。

ステップ１０：優先順位の更新

STAGE = 1 とする。

$$I_A{}^j := [I_N{}^j \cap I_R] \cup I^{j(r+1)} \qquad (33)$$

$$I_N{}^j := I_N{}^j \cup I^{j(r+1)} \qquad (34)$$

と更新して、資源 j が割当可能な節点 i の数を増やす。ただし、$r \geqq r^{max}$ ならば、$I^{j(r+1)} = \phi$ として同様に更新する。r := r + 1 とする。

ステップ１１：解なしの判定

すべての資源 j に対して $I_N{}^j$ と $I_A{}^j$ が変わらないならば、実行可能解はなしと判定して終了する。さもなければ、次のステップへ。

ステップ１２：未展開節点の再展開

制約式が実行不可能である根節点を再び未展開節点とするために、根節点が i であるすべての探索木 T^h に対して、(23)式のように $\Omega_N{}^h$ と $\Omega_A{}^h$ を再定義する。$I_R = \phi$ として、ステップ１へ。

７．数値シミュレーション実験

提案法を用いて数値計算を行った結果を表１～３に示す。表中の各項目の詳細は、文献[9] を参照。使用したコンピュータはＡＣＯＳシステム９１０　モデル１０であり、言語は FORTRANを用いた。

例１～１０の問題は、乗算合同法によって発生させた乱数を用いて作成した。文献[9] を参照。

表１にあるように、各例の０－１整数変数の個数は、値を固定したままで動かさない変数も含めると、(300x350)～(800x1000)個である。表１には、すべての資源を優先順位第１位の期間帯に割り当てた初期解において、一つ以上の制約式が実行不能である期間帯 i の個数、実行不可能な制約式の総数、実行不可能量の総和（すなわち評価関数値）を示している。

表２と３では、例１～１０の探索状況を示している。表２で示した例１を説明すると、第６章で記述したアルゴリズムに従って、割当可能な期間帯の優先順位 r を一つずつ大きくしており、例１では優先順位第３位までで実行可能解が求まっている。すなわち、第３回

めのイテレーションにおいて、優先順位第１位から第３位までの範囲の期間帯に割り当てた結果、実行可能解が求まっている。表２の上部には、各イテレーションにおいて割当を変更した結果、各優先順位の期間帯がいくつ割り当てられたかを表している。

表２からわかるように、例１はSTAGE 1 の１割当変更法だけでは優先順位第３位までで実行可能解を求めることができなかったが、提案法を用いて求めることができた。他の例においても同様に、１割当変更法のみで求めるよりも小さい優先順位で実行可能解を求めることが、提案法によって可能となった。

表２の実行不可能和の値からわかるように、例１において、イテレーション回数ｒが大きくなるにつれて、評価関数値が単調に減少している。また、１割当変更では整数変数を一度に２個しか変化させることができないが、イテレーション回数ｒ＝３では、12個の整数変数を一度に変化させることにより単調減少させている。表２の「一度に変化させた変数の最大個数」を参照。この事より、提案法では、実行不可能和の単調減少性を保証することによって収束の保証を得ていることがわかる。

本学に導入されているＡＣＯＳ４の数理計画システムＭＰＳでは、整数変数は1000個以下という制限があるが、これでは期間帯数がわずか５０でも資源数が２０あればこの制限をこえてしまう。また、このＭＰＳには、行列は5000以下、列数は 20000以下という限界があるため、この問題の整数変数を連続変数とみなしたＬＰ緩和問題を解くことでさえ困難である。従って、このＭＰＳを用いて例１～１０を解くことは不可能である。

しかしながら、提案法は、期間帯数及び資源数が例えば 100以上の大規模な問題を解くのに効率的である。例えば、初期値においてほとんどすべての節点で制約式を違反している例4では、約１０万個（300x350個）の整数変数のうち、約３万個の整数変数は固定したままであるけれども、残りの約７万個（表３での、「割当可能となった節点の総数」）の整数変数を変化させて解を求めている。

不要となったアークがたとえ少数個であっても、アーク総数の大小により削除率への影響が大きく異なるため、表３に示した各イテレーションの「アークの平均削除率」は、すべての割当変更に対して、(15) 式のすべての分子の総和を、すべての分母の総和で割った値を示している。

各イテレーションで、多数の節点に新たに割当可能になるとすると、STAGE 1 のみで実行不可能和が零になることが多い。表３の例９を参照。しかしながら、ここで示した他の数値例のように、最後のイテレーションのSTAGE 1 が終わった段階で実行不可能和が多少残った場合には、STAGE 2 を実行することが有効となる。

表３に示したように、各数値例では最後の数回STAGE 2 を実行したけれども、最後の１回以外は、実行不可能和はほとんど減少しなかった。それ故、「アークの平均削除率」などは、最後のイテレーションでのSTAGE 2 の実行結果である。

表３の下には、各例で実行可能解が求まるまでの計算時間を示している。例５と８では、初期解での実行不可能和が過度に大きいので、計算時間がそれぞれ、420分間、 300分間でアルゴリズムが終了しなかった。しかしながら、その他の数値例においては、提案法は実用的な計算時間内に優先順位付きの割当による解を求めている。

表２　例１の探索状況

イテレーション回数ｒ ＼ 優先順位	1	2	3
1	5620	5086	4846
2	／	534	533
3	／	／	241
新たに割当可能となった節点数		866	868
深さ２以上の探索回数		0	45
探索の深さの最大値		1	19
割当変更回数		671	256
深さ２以上の割当変更回数		0	19
割当変更の深さの最大値		1	9
一度に変化させた変数の最大個数		2	12
アークの平均削除率（単位は％）		／	14
余裕量		79	／
制約を違反した節点の個数	231	164	0
制約を違反した式の個数	685	220	0
評価関数値(実行不可能和)	1006	256	0

８．あとがき

本論文では、優先順位に従った割当を行って、割当問題の実行可能解を求めるアルゴリズムを提案した。提案法は、利益あるいは費用が金額として明確に与えることができない割当問題を解くのに有効である。

提案法は、各資源の個数が一定であることを利用した近似解法であり、普通の１割当変更では実行不可能和が変わらない場合に、提案法では更に続いてどの割当の変更をすべきかを調べている。大規模な割当問題の数値シミュレーション実験を行った結果、提案法は効率的であることが示された。

参考文献

[1] 茨木俊秀、：組合せ最適化　分枝限定法を中心として、産業図書、(1983).

[2] 今野浩、鈴木久敏編、：整数計画法と組合せ最適化、日科技連、(1982).

[3] G. T. Ross and A. A. Zoltners, : Weighted Assignment Models and their Applications, Management Science, Vol.25, No.7, pp.683-696, (1979).

[4] G. Schmidt, : Timetabling Construction - an Annotated Bibliography, The Computer Journal, Vol.23, No.4, pp.307-316, (1980).

[5] 今野浩　他、：最適クラス編成問題　－東京工業大学におけるケース・スタディー　－、オペレー

ションズ・リサーチ、Vol.36, No.2, pp.85-89, (1991).
[6] K. Aoki and A. Nishikori, : An Algorithm for Constrained Load Flow, IEEE Transactions on Power Apparatus and Systems, Vol. PAS-103, No.5, pp.963-973, (1984).
[7] 錦織昭峰、：優先順位を考慮した割当問題に関する考察、広島県立大学紀要、Vol.3, No.1・2, (1991).
[8] 錦織昭峰、一森哲男、渡辺展男、金指正和、青木兼一、：優先順位を考慮した割当問題の近似解法、日本オペレーションズ・リサーチ学会秋季研究発表会、No.2-E-3, pp.222-223, (1990).
[9] 錦織昭峰、一森哲男、渡辺展男、金指正和、青木兼一、：優先順位を考慮した大規模一般化割当問題のための近似解法、情報処理学会第36回数値解析研究会研究報告、91-NA-36-3, pp.1-10, (1991).
[10]草刈君子、宮崎知明、金指哲也、：富士通ＡＭＰからみた最近の混合整数計画法　－その性能と適用例－、第二回ＲＡＭＰシンポジウム論文集、pp.11-20, 社団法人日本オペレーションズ・リサーチ学会特殊研究部会数理計画法研究会、(1990).
[11]藤野喜一、：乱数を使用した学校時間割作成プログラム、情報処理、Vol.5, No.2, pp.68-76, (1964).

表１　数値例の初期値データ

数値例	例１	例２	例３	例４	例５	例６	例７	例８	例９	例１０
変数の次元 (=mxn)	300x350	同左	同左	同左	同左	500x700	同左	同左	800x1000	同左
非零 x_{ij}(=1)の数	5620	7478	7695	5868	15677	37902	26255	34380	45369	49418
制約式の数	4741	5722	6706	9181	4462	6530	9064	11488	9204	13102
制約式の非零係数の数	90924	282013	331795	811123	400232	262503	544542	790604	560200	864631
制約を違反した節点数	231	278	292	292	291	472	484	477	766	773
制約を違反した式の数	685	2079	3942	5701	3196	4140	6454	7066	5946	9023
実行不可能和	1006	6417	13744	25588	38700	18221	29303	42541	20788	37481

表３　各例の探索状況

例	1	2	3	4	5	6	7	8	9	１０
解が求まった優先順位	3	5	9	１２	(５)	5	5	(３)	5	5
割当可能となった節点の総数	7354	18690	38577	71066	52624	70034	90413	66218	91495	125242
STAGE 2 を実行した回数	1	2	2	3	0	1	2	0	1	2
深さ２以上の探索回数	45	17	23	73	0	50	27	0	0	22
探索の深さの最大値	9	3	4	7	1	9	3	1	1	4
割当変更回数	927	5267	10480	14667	／	16969	24834	／	19336	33791
深さ２以上の割当変更回数	19	13	14	32	0	28	18	0	0	15
割当変更の深さの最大値	9	3	4	7	1	9	3	1	1	4
一度に変化させた変数の最大個数	12	6	8	14	2	14	6	2	2	8
アークの平均削除率（単位は％）	14	16	36	49	／	27	25	／	／	13
計算時間（単位は分）	0.4	11	66	319	420 以上	77	168	300 以上	53	231