資源利用権の保留負担割当問題

社会情報学科　行　方　常　幸

1．はじめに

大学や企業などの組織は複数の部門（学部，学科，部等）からなる。各部門の独立性が高くそれらの統廃合が困難であり，各部門の仕事内容の重要度も優劣がつかない状況を想定する。人件費削減のために，組織全体の人員削減計画が作成されたとする。この組織では定年退職等による組織外への人の移動がある程度あり，あらかじめ把握されている。この移動により生じた人員枠を補充せずに削減人員に利用することを計画している。注意点は，これら組織外へ移動した人員枠を削減人員に利用せず，通常のように補充すると，後の削減人員に利用できないことである。各部門になるべく公平に負担してもらうには，これら移動により生じた人員枠を削減人員にどのように割当てればよいだろうか？　以上の状況において，移動により生じた人員枠を利用権の保留が可能な資源，削減人員に利用することを資源利用権の保留とみなし，次節で述べるような「資源利用権の保留負担割当問題」と捉え，それを解くためのソフトウェアを Java で作成したので本稿で紹介する。

2.「資源利用権の保留負担割当問題」

この節では「資源利用権の保留負担割当問題」とその解を説明する。

n 個の部門からなる組織がある（$N:=\{1,...,n\}$ とおく）。部門 i ($\in N$) において，考慮すべき計画期間 T の間に生じる利用権の保留が可能な資源の集合はあらかじめ与えられていて，$AR(i)$ とする。（利用権の保留が可能な全資源を $AR:=AR(1)\cup...\cup AR(n)$ とおく。）正確には，利用権の保留が可能な資源 j ($\in AR$) は第 S_j 期以降に利用権の保留が可能であり，保留された場合の負担に対する重みを w_j とする。部門 i がこの計画期間が始まる前に既に利用権を保留された資源の重み付き累積期間を $PB(i)$ とする。部門 i を $Dep(i):=((S_j, w_j)_{j\in AR(i)}; PB(i))$ で表し，組織全体を $Deps:=(Dep(i))_{i\in N}$ で表す。この組織 $Deps$ が利用権を保留しなければならない資源の量は，あらかじめ決まっており，次で与えられる：第 t 期（$t=0,...,T-1$）に利用権を保留しなければならない資源の個数は d_t (非負の整数) である（$d:=(d_0,...,d_{T-1})$ とおく）。

以上の状況の下で，利用権が保留される期間の重み付き総和を各部門になるべく公平に割当てる問題を「資源利用権の保留負担割当問題」と呼び，組 $(Deps,d)$ で表す。

利用権の保留が可能な資源を保留するか否かを表す変数 $f:=(f(j,t))_{j\in AR,t=0,...,T-1}$ を定義する：

$$f(j,t)=\begin{cases}1 & \text{資源 } j \text{ の利用権を } t \text{ 期に保留する時}\\ 0 & \text{資源 } j \text{ の利用権を } t \text{ 期に保留しない時}\end{cases} \quad (1)$$

資源 j は第 S_j 期に利用権の保留が可能になり，一旦，利用権を保留されなかった資源は後に利用権を保留できないという資源の性質により，次が成立する。

$$f(j,t)=0\ (t=0,...,S_j-1),\ f(j,t)\geq f(j,t+1)\ (t=S_j,...,T-2) \quad (2)$$

各期において利用権を保留する資源の個数が保留しなければならない個数 d_t 以上でなければならないので，次が成立する。

$$\sum_{j\in AR} f(j,t) \geq d_t \ (t=0,\ldots,T-1) \tag{3}$$

条件(1)，(2)，(3)を満たす f の集合を F とおく。資源の利用権の保留負担割当方法 $f(\in F)$ のもとで，計画期間終了までに部門 i が負担する資源利用権の重み付き累積保留期間 $B(f,i)$ は次のように与えられる。ただし，δ $(0<\delta\leq 1)$ は割引率である。

$$B(f,i) := PB(i) + \sum_{t=0}^{T-1} \delta^t \sum_{j\in AR(i)} w_j f(j,t)$$

部門 i は $B(f,i)$ が小さいことを望む。従って本稿では，これら $B(f,i)$ $(i\in N)$ を大きいものから並べたものを $B^*(f)$ (負担ベクトルと呼ぶ) とし，辞書式順序 $\leq_L$[1]における最小値を与える $\{f\in F \mid B^*(f) \leq_L B^*(g)\ (\forall g\in F)\}$ を解とする。

例として次のように与えられた6部門からなる組織の10期間問題を考える：

$AR(1)=\{1,2,3,4,5\}$,

$AR(2)=\{6,7,8,9,10,11,12\}$,

$AR(3)=\{13,14,15,16\}$,

$AR(4)=\{17\}$,

$AR(5)=\{18,19,20,21,22,23\}$,

$AR(6)=\{24,25,26,27,28,29\}$

$AR=AR(1)\cup AR(2)\cup AR(3)\cup AR(4)\cup AR(5)\cup AR(6)$,

$Dep(1)=((0,1),(6,1),(7,1),(8,1),(9,1);0)$,

---

1) 2つのベクトル $B^*(f)$ と $B^*(g)$ の要素を左から順番に比較し，異なる最初の要素の大小を $B^*(f)$ と $B^*(g)$ の大小とする。

$Dep(2)=((0,3/4),(2,1),(6,1),(7,3/4),(8,1),(9,1),(9,1);3/4)$,

$Dep(3)=((2,1),(7,1),(9,1),(9,1);1)$,

$Dep(4)=((0,3/4);7/4)$,

$Dep(5)=((0,1),(4,1),(6,1),(6,1),(7,1),(7,1);1)$,

$Dep(6)=((6,1),(7,1),(7,1),(7,1),(8,1),(9,1);2)$,

$Deps=(Dep(1),Dep(2),Dep(3),Dep(4),Dep(5),Dep(6))$

$d=(2,2,3,4,5,5,5,5,5,5)$

第0期は平成16年度で，年次進行で発生する削減人員は，既に発生しているもの，平成16年度に発生するもの，…，平成25年度に発生するもの，の順に，1,1,0,1,1,1,0,0,0,0,0とする（これから上記のdが得られる）。この問題を表1のように表すことにする。表において，「S」はその列で表される人員がその行で表される期間において，保留可能になることを意味する。

これを（次節で紹介するプログラムで）解くと，最適な負担ベクトルは$\left(8,8,8,\frac{31}{4},7,7\right)$であり，540通りの解が得られる。これは，すべての解の個数であり，第0期が異なるものだけを求めると，1通りになる。これは表2で与えられている。表において「1」は保留することを意味する。すなわち，第0期において，部門1と部門4の人員の利用権が保留され，部門2と部門5の人員は補充される。

3．「資源利用権の保留負担割当問題」プログラムの計算方法

この節では「資源利用権の保留負担割当問題」を解くプログラムを紹介する。この問題を解く計算方法を簡単に説明し，問題点を述べ，それへの対処法を述べる。

計算方法は深さ優先探索を用いた分枝限定法である。分枝操作は保留する資源を各部門に割当てる，割当て方の，第0期から始まる単純な列挙であり，限

表1　資源利用権の保留負担割当問題の例

(t)	部門1					部門2							部門3				部門4	部門5						部門6							要
以前	0					3/4							1				7/4	1						2							1
重み	1	1	1	1	1	3/4	1	1	3/4	1	1	1	1	1	1	1	3/4	1	1	1	1	1	1	1	1	1	1	1	1		
16(0)	S					S											S	S												>=	2(1)
17(1)																														>=	2(0)
18(2)							S						S																	>=	3(1)
19(3)																														>=	4(1)
20(4)																			S											>=	5(1)
21(5)																														>=	5(0)
22(6)		S						S												S	S			S						>=	5(0)
23(7)			S						S					S								S	S		S	S	S			>=	5(0)
24(8)				S						S																		S		>=	5(0)
25(9)					S						S	S			S	S													S	>=	5(0)

表2　数値例の1つの解

(t)	部門1					部門2							部門3				部門4	部門5						部門6							要
以前	0					3/4							1				7/4	1						2							1
重み	1	1	1	1	1	3/4	1	1	3/4	1	1	1	1	1	1	1	3/4	1	1	1	1	1	1	1	1	1	1	1	1		
16(0)	S1					S											S1	S												>=	2(1)
17(1)	1																1													>=	2(0)
18(2)	1						S1						S1				1													>=	3(1)
19(3)	1						1						1				1													>=	4(1)
20(4)	1						1						1				1		S1											>=	5(1)
21(5)	1						1						1				1		1											>=	5(0)
22(6)	1	S1						S									1		1	S1	S			S						>=	5(0)
23(7)			S						S1					S			1		1	1		S	S		S1	S	S			>=	5(0)
24(8)				S					1	S1									1						1			S1		>=	5(0)
25(9)					S				1		S	S			S1	S1									1			1	S	>=	5(0)
計	8					8							7				31/4	8						7							

定操作は「負担は正であるので，第0期から第t期までの負担ベクトルが暫定ベクトルよりも悪ければ最適解にはならない」という性質を利用している（表3参照）。

前節の数値例は数分で解けるが，問題によっては，列挙する場合の数が多すぎ，実用的な時間で解くことが不可能な場合がある。従って，厳密な最適解を求めることはあきらめ，何らかの近似解で満足する必要がある。

本プログラムにおいて，暫定負担ベクトル（最適とは限らないが，その時点で，最も良い負担ベクトル）は計算時間の経過と共に減少する。従って，状況が許す限り計算を行い，その時点での負担ベクトルを与える実行可能解を近似解とすることが考えられる。

本プログラムの場合，$f(\bullet,\bullet)$の列挙の方法が部門の配列の保持の方法に依

表3 計 算 方 法

$UV:=(UV_1,\ldots,UV_n)$は負担の上界ベクトル（暫定ベクトル；初期値は$UV_i=1,000,000$とした）
solve (t) ｛
(1)と(2)と(3)を満たす$f(j,t)$を列挙する。その各々に対して ｛

$$B^t(f,i):=PB(i)+\sum_{k=0}^{t}\delta^k\sum_{j\in AR(i)}w_jf(j,k)$$

を求める。これらを大きい順に並べたものを$B^{t*}(f)$とする。
tが最後の期ならば ｛
$B^{t*}(f)$がUVと一致する場合：今の$f(\bullet,\bullet)$を最適解の候補として追加保存する。
$B^{t*}(f)$がUVを改良している場合：UVを改良し，今の$f(\bullet,\bullet)$を最適解の候補として新規保存する。
｝最後の期でなければ ｛
$f(\bullet,\bullet)$がUVを改良している場合には，*solve* $(t+1)$を呼ぶ。
｝
｝
｝
と *solve* (t) を定義する。
solve (0) を実行すれば，最適解が求まる。

存し，例えば，（部門1,･･･,部門6）のように保持した場合，部門の番号が大きい方，すなわち，右にある部門が多くの利用権を保留する場合から，列挙するため，計算が途中で打ち切られた場合，右にある部門が不利益をこうむる可能性がある。この現象に対処するために，各部門が左端に来る6通りの配列を無作為に発生させ，例えば，

（部門1，部門5，部門4，部門3，部門2，部門6）
（部門2，部門4，部門3，部門6，部門1，部門5）
（部門3，部門6，部門5，部門1，部門4，部門2）
（部門4，部門1，部門2，部門5，部門6，部門3）
（部門5，部門2，部門6，部門4，部門3，部門1）
（部門6，部門3，部門1，部門2，部門5，部門4）

の6つの問題[2)]を同時に（すなわち，同じプライオリティを持つスレッドで）解く。指定された時間で最適解が見つからなければ，この6つの問題の中で負担ベクトルが最小のものの実行可能解を近似解とする。具体的には6つのスレッドで負担ベクトルを共有すればよい。

さて，6つのスレッドを同時に解くので，各スレッドに与えられる計算時間は単純に見積もれば，総計算時間の1/6になる。この方法は各部門が公平に扱われることを事後的に保証するが，計算時間が1/6となるため，最適解から離れる可能性がある。そこで，事前的に各部門を公平に扱うだけにする，もうひとつの方法も考える。すなわち，上記の6つのスレッドをすべて実行するのではなく,1/6の確率でどれか1つを選び,そのスレッドだけを実行するのである。

以上の考察により，次の3種類の計算方法を実装した：

A）部門の配列を無作為に1通り選び，そのスレッドだけで計算する。

B）各部門が最初に来る6通りの配列を無作為に選び，6つのスレッドで計算する。ただし，この6通りの配列を，同じ部門が配列の同じ位置に来ないように選ぶ。

2）この行列の各行各列には各部門が丁度1回登場していることに注意！

C）部門の配列を1通り指定し，そのスレッドだけで計算する。

A）は事前の公平性だけしか保証されないが，長く計算でき，最適解により近い解が求まる可能性がある。B）は事後的な公平性が保証される。公平性の立場からはB）が望ましいと思われる。

4．プログラムを利用した数値例の解法

この節では第2節で導入した例を解くことにより，プログラムの利用法を説明する。

プログラムを起動したのが図1である。表1のデータを入力したのが図2である。既定では，第0期目の解が異なる解だけが表示される。すべての解を表示させる場合は，右上の「全ての解を表示」チェックボックスをオンにする[3)]。左上の「実行」ボタンを押すと，A）の方法で計算が始まり図3のダイアログボックスが表示される。このダイアログボックスで，計算の取消し，計算の終

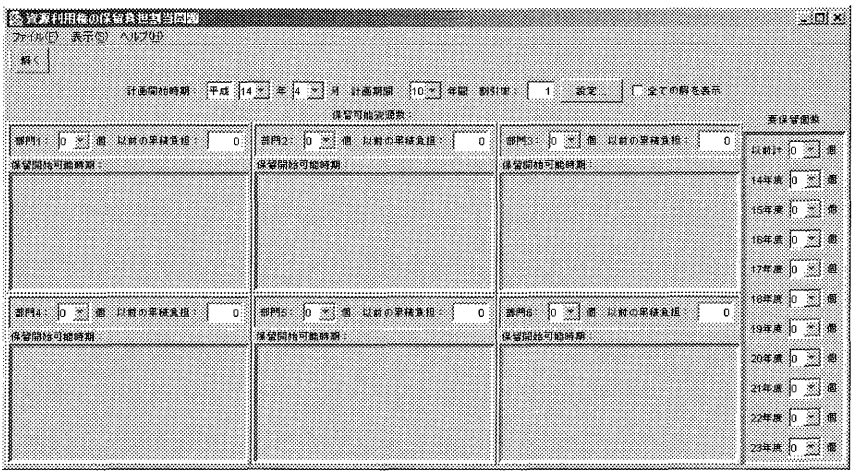

図1　起　動　画　面

3）オンにして計算を実行すると，540通りの解が表示される。

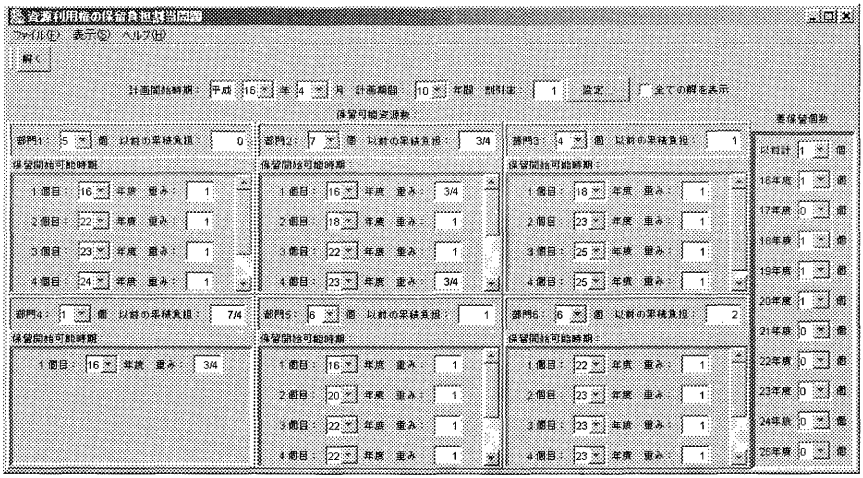

図2　データを入力したところ

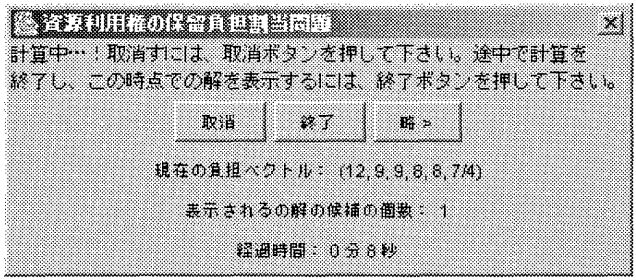

図3　このダイアログボックスで取消し可能

了（打ち切り）が可能である。また，その時点での負担ベクトルの値と表示される解の候補の個数が表示される。しばらくすると，計算が終了し，図4のダイアログボックスが表示される。計算時間が4分21秒であったことが分かる。図2の「表示」メニューから「部門の配列」を選ぶと，無作為に選ばれた部門の配列が図5のように表示される。得られた解は図6である。これを表にしたのが表2である。

B）の方法で計算を行うには，図1または図2のメインウィンドウで「設定」

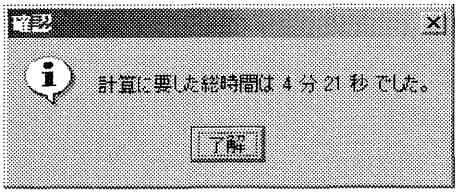

図4　確認ダイアログボックス

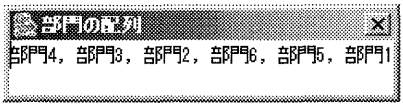

図5　部 門 の 配 列

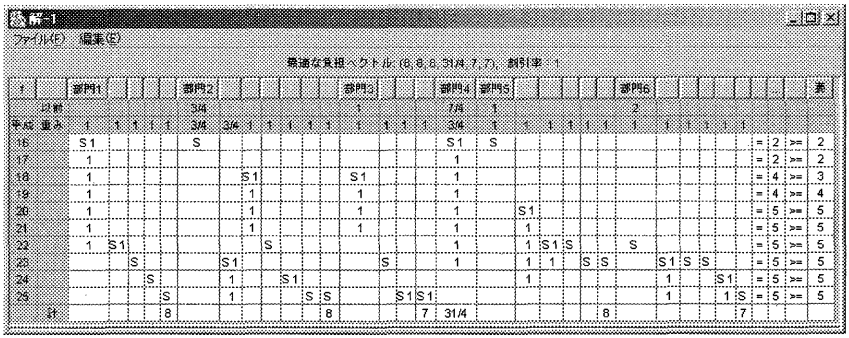

図6　数 値 例 の 解

ボタンを押す。図7のダイアログボックスが現れるので，中央の「計算の取消可」のチェックを外し，「最大計算時間の指定」をチェックし，その下で最大計算時間を指定し，「設定」ボタンを押す。例として，最大計算時間を1分に指定し，計算を行うと，図8のように表示され，1分計算した段階で停止する。その時点での解は図9で与えられている。負担ベクトルは $\left(9,8,\frac{31}{4},\frac{31}{4},7,7\right)$ なので，この時点では，まだ最適解は得られていない。タイトルバーからこれはスレッド2によるものであることが分かる。「表示」メニューから「部

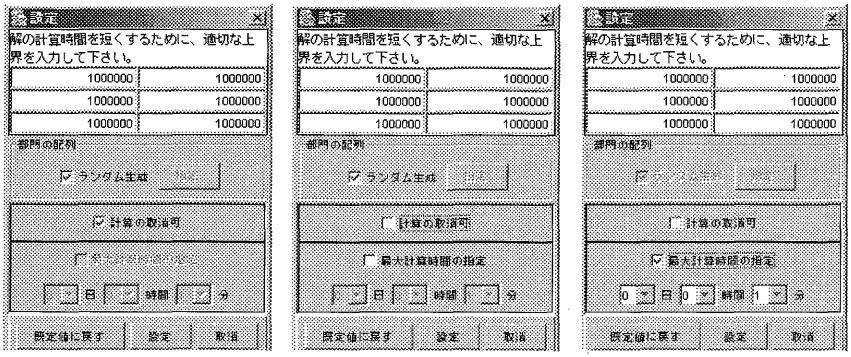

図7　計算方法B）の設定の仕方

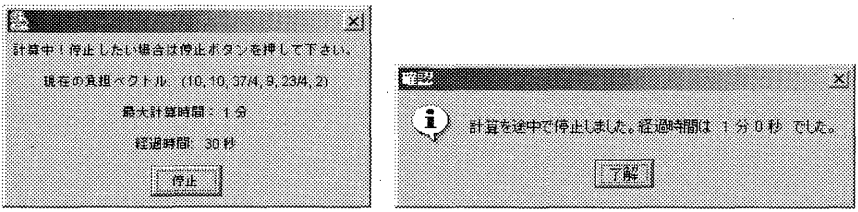

図8　方法B）の途中経過と結果

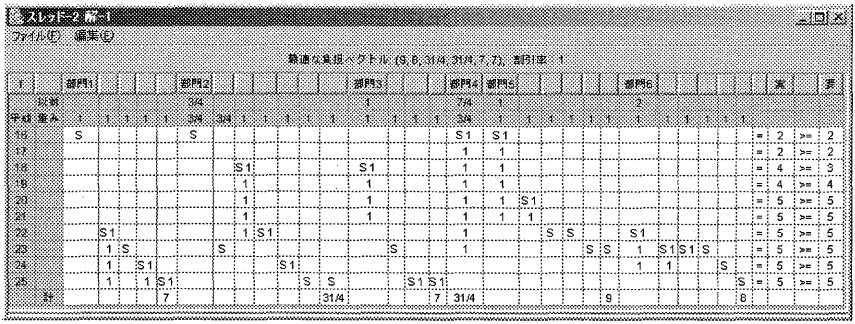

図9　方法B）の例の解

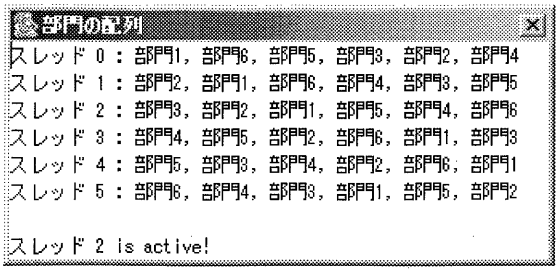

図10 部 門 の 配 列

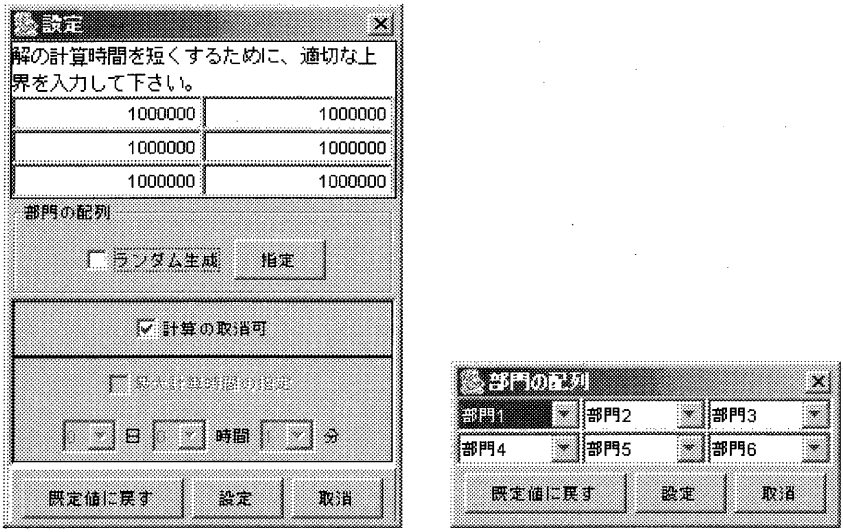

図11 C）の方法の設定の仕方

門の配列」を選ぶと（図10を参照），実際に選ばれた部門の配列が分かる。

C）の方法で計算を行うには，図11のように，ダイアログボックスで「ランダム生成」のチェックを外し，「指定」ボタンを押し，表示されるダイアログボックスで適切に指定し「設定」ボタンを押す。例えば，

（部門４，部門１，部門６，部門３，部門２，部門５）

を指定し，計算を行うと，最適解が求まるまで５分15秒かかった。別の例とし

て,

（部門２，部門５，部門４，部門１，部門６，部門３）

を指定すると，２分41秒で最適解が求まった。

実際の場面ではないことであろうが，最適解が求まるまでの計算時間が短くなる部門の配列があらかじめ予想できるならば，このC）の方法が適している。

5．おわりに

本稿では「資源利用権の保留負担割当問題」に対し，負担ベクトルを辞書式順序の意味で最小にする保留負担割当を解として提案し，それを計算するプログラムを紹介した。厳密な最適解が求まらない場合の近似解として，事後的な意味で各部門を公平に扱うものと，事前的な意味でしか扱わないが，より最適解に近いものを目指すもの，の２つの方法を提案した。

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］ 茨木俊秀：「離散最適化法とアルゴリズム」岩波講座応用数学，岩波書店，(1993)。