数理モデル化と問題解決　21－11
（1998．10．7）

バックトラックの不要なタスク割付可能性判定手法

毛受 哲　中莖洋一郎　岩本雅彦 (NEC)

各製品が複数の設備で生産可能であるような生産計画を立案する場合、製品を設備に割り付けながら立案するとバックトラックが発生して効率が悪い。本稿では、製品を設備に割り付けることなく生産可能性をチェックできる手法を提案する。これにより設備割付によるバックトラックが不要になり、しかも処理が簡単のため生産可能性チェック回数が多くても高速に処理することが可能である。本手法の正当性を示し生産計画への適用について検討する。

A Task Assignment Check Method without Backtrack Search

Satoshi MENJU, Yoichiro NAKAKUKI, Masahiko IWAMOTO

NEC Corporation

This paper shows a resource capacity check method for production scheduling problems in which each task (product) is assignable to resources (equipment). If each task is assigned to a resource in making a production schedule, many backtrack searches occur. Because our method does not assign each task to a resource but a resource group, it is efficient to remove backtrack searches. This paper shows completeness of this method and application to production scheduling problems.

1　はじめに

生産現場では多品種小量生産に移り、納期、資材の有無、使用する設備の負荷など個々の製品の生産条件を考慮して最適な計画を立案することが難しくなってきている。特に数千品種を扱う大規模生産計画などでは規模的にまだOR的な立案が難しく、また時間を掛けて最適解を求めるよりは短時間に準最適解を求める方が良いため、初期計画を立案して評価しながら調整する手法が取られている。また、一部にOR的な手法を取り入れて精度を上げる試みも行なわれている [1]。

生産状況が厳しい場合、すべての条件を満たす計画を立案できないことも多い。このような場合、条件を制約として表現し制約を違反した場合の違反点数を定義して、全制約の違反点数の合計が小さくなるように計画を調整していく制約主導の立案手法が有効である。このとき、対象分野に合わせて様々な調整手法が考えられている [2, 3]。

立案した計画が生産可能か判定するには、設備の能力と負荷をチェックする必要がある。特に計画を評価しながら調整する手法では、調整の度に設備の能力チェックが必要になる。各製品が複数の設備で生産可能な場合には、設備への割付を考慮すると問題が非常に難しくなる。既に割り付けた設備を他の設備に割り付け直したりバックトラックが必要になるためである。製品数が少なければ設備割付を含めて考慮して生産の組合せを決めることも可能だが、製品数が多くなれば設備への割付を行ないながら生産計画を立案することは実用上不可能に近い。

本稿では、各製品が複数の設備で生産可能である生産計画を立案するとき、製品を設備に割り付けることなく生産可能性を効率良くチェックできる方式について提案し、その正当性を証明する。本稿では生産計画問題の上で議論を行なうが、タスクをリソースに割り付ける問題に応用できる。以下、2章では生産可能性問題の定義と検証手法、その正当性について検討し、3章では検証に必要な集合を効率良く求めるアルゴリズムと正当性について検討し、4章では生産計画への適用について検討する。

2　生産可能性問題と検証手法

2.1　生産可能性問題

まず生産可能性問題を定義する。

生産可能性問題

各製品 $p(i)$ を生産可能な設備の集合を $u(i)$ とする。ある計画が与えられたとき生産可能な設備割付があるか判定する問題を生産可能性問題と呼ぶ。すなわち、設備 $e(j)$ の能力を $c(j)$、設備 $e(j)$ で生産する製品 $p(i)$ の量を $a(i,j)$ と表すと、

$$\sum_{\{i \mid e(j) \in u(i)\}} a(i,j) \leq c(j)$$

$$a(i,j) \begin{cases} \geq 0 & (e(j) \in u(i)) \\ = 0 & (e(j) \notin u(i)) \end{cases}$$

となる $a(i,j)$ が存在するか判定する。

生産可能性問題により生産可能性が確認できれば、線形計画法により個々の設備に負荷を割り付けることが可能である。

例 1　設備 A、B、C の能力が 10、50、10 で製品 1、2 が {A, B}、{B, C} で生産可能とする。このとき製品 1、2 を 40 ずつ生産する計画は生産不可能である。また、製品 1、2 を 30 ずつ生産する計画は生産可能で、製品 1 を 30 と製品 2 を 20 設備 B に割り付け、製品 2 を 10 設備 C に割り付けることができる。

2.2　検証手法

生産可能性問題は条件式が線形であるため線形計画法で解くことができる。しかし、大規模生産計画の場合に変数の制限があったり、オーダを積み上げながら立案する場合に積み上げる度に生産可能性をチェックすることが計算時間的に無理があったりする。そこで簡単にチェックする手法を検討する。

全製品が全設備で生産可能な場合には、全設備の能力合計に対して全製品の負荷の合計が越えていなければ生産可能であり、生産可能性のチェックには設備への割付の必要が無い。しかも現在の能力合計と負荷合計を覚えておけば、次のチェックは追加削除の変更分を計算するだけで良い。

その延長で、例えば例 1 の場合に {A, B} の能力合計 60 と {B, C} の能力合計 60 に対して負荷をチェックするだけでは、製品 1、2 を 40 ずつ生産する計画も負荷が越えていないため生産可能のように見えてしまう。これは重複する設備 B の能力を双方に加えているためである。そこで、{A, B} と {B, C} の和集合である {A, B, C} に対するチェックを追加する。すると製品 1、2 を 40 ずつ生産する計画は {A, B, C} の負荷が能力を越え、30 ずつ生産する計画は負荷が能力を越えないため正しく判定できることが分かる。

以上のアイデアをまとめると次のようになる。まずパワーユニオンについて定義する。

定義 (パワーユニオン (PU))

各製品 $p(i)$ を生産可能な設備の集合を**基本パワーユニオン**と呼び、1 つ以上の基本パワーユニオンの和集合を**パワーユニオン (PU)** と呼ぶ。パワーユニオンに含まれる設備の能力の合計を**パワーユニオンの能力**と呼ぶ。パワーユニオン U1 の設備がパワーユニオン U2 に含まれるとき、U1 を U2 の**下位のパワーユニオン**、U2 を U1 の**上位のパワーユニオン**と呼ぶ。同じ基本パワーユニオンにある 2 つの設備は、**競合状態**にあると呼ぶ。また、設備 A と B、設備 B と C が競合状態にある場合、設備 A と C も**競合状態**にあると呼ぶ。

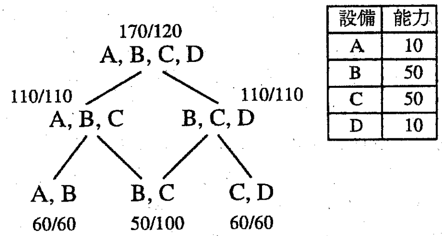
例えば基本 PU が U1={A, B}、U2={B, C}、U3={D} であるとき、U4={A, B, C} は U1 と U2 の上位の PU である。U1 で生産可能な製品の設備 A への割付を少なくしようとしたら、U1 の設備 B への割付を多くする必要がある。従って A と B は競合状態にある。さらに設備 B の能力に上限があるため U2 の設備 B への割付を少なくする必要が生じる可能性がある。このように PU を仲立ちにして設備 A と C も競合状態にあると言える。ただし、設備 D は完全に独立であるため他の設備と競合状態にはない。

まったく競合の無い設備は別にチェックすることができる。従って、以後はすべての設備に競合があると仮定する。

生産可能性検証手法

計画に対し、生産する製品の基本 PU とその上位の PU に負荷を加算する。そしてすべての PU に対して負荷が能力を越えなければ生産可能である。

例 2　設備 A, B, C, D の能力が 10, 50, 50, 10 で製品 1, 2, 3 が {A, B}, {B, C}, {C, D} で生産可能とする。このとき PU は {A, B}, {B, C}, {C, D}, {A, B, C}, {B, C, D}, {A, B, C, D} となる (図 1)。このとき製品 1, 2, 3 を 60, 50, 60 生産する計画は、製品 1 の 60 を {A, B}, {A, B, C}, {A, B, C, D} に、製品 2 の 50 を {B, C}, {A, B, C}, {B, C, D}, {A, B, C, D} に、製品 3 の 60 を {C, D}, {B, C, D}, {A, B, C, D} に加算する。すると {A, B, C, D} の能力が 120 のところ負荷は 170 のため生産不可能である。



パワーユニオンの側の数字は "負荷/能力"

図 1: 例 2 のパワーユニオン

この生産可能性検証手法は、生産量の追加や変更のときの生産可能性チェックが容易にできる。追加 (削除) する場合は、製品に該当する基本 PU と上位の PU に負荷を加算 (減算) し、負荷が能力を越える PU が無ければ追加可能 (削除後生産可能) と分かる。つまり大規模生産計画を立案するとき、生産可能性をチェックしながらオーダを積み上げたり計画を一部修正することを頻繁に行なうが、本手法ではその生産可能性チェックが低コストで済むため立案時間を短縮することが可能となる。

2.3 正当性の証明

前節の生産可能性検証手法が正しいことを以下に証明する。

定理 1 生産可能性検証手法は正しく生産可能性をチェックする。

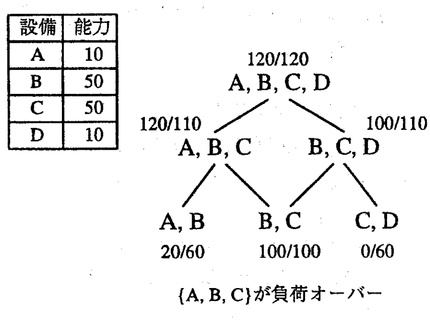
証明

まず、負荷が能力を越える PU があれば生産不可能であることを示し、次に生産不可能であれば負荷が能力を越える PU があることを示す。これにより、負荷が能力を越える PU があるときのみ生産不可能であることが証明できる。

PU の負荷は下位の基本 PU の負荷の合計であり、従って PU に含まれる設備以外使用できない製品の負荷の合計である。よって負荷が能力を越える PU があれば生産不可能である。

ある計画が生産不可能だが、すべての PU で負荷が能力以内とする。このとき、設備に全生産量を割り付けられない製品 $p1$ が存在し、他の製品を減らさなければ $p1$ の割付を増やせないとすると、$p1$ を生産できる設備の負荷はすべて能力以上である。この設備の集合を E とし $P = \{p1\}$ とする。さらに E に含まれる設備 $e1$ で生産量のある製品 $p2(\notin P)$ があれば、$p2$ を P に追加し $p2$ を生産できる設備を E に追加する処理を繰り返す。ここで新規追加した設備に能力の余裕がある設備 $e2$ があれば、$p2$ に対する $e2$ の生産量を増やして $e1$ の生産量を減らすことができ、$e1$ による $p1$ の生産量を増やせることになって矛盾する。従って、新規追加する設備もすべて負荷が能力以上である。このような処理を繰り返すと、製品と設備は有限のため処理は終了し、E に含まれる設備は P に含まれる製品以外には生産しないがすべて負荷が能力以上であるため、P に含まれる製品のみで負荷が能力以上になっている。P に含まれる製品を生産できる設備の集合が E であるから、P に含まれる製品に該当する基本 PU のすべての和であるパワーユニオン U が E と等しくなり、P の製品の負荷はすべて U にも加算される。U の設備はすべて負荷が能力以上のため、$p1$ の割り付けられなかった生産量を計画通り加えれば、U の負荷は能力を越えることになる。□

例 3 設備や製品が例 2 と同じとき製品 1, 2, 3 を 20, 100, 0 生産する計画は、例えば製品 2 を設備 B と C に 50 ずつ割り付けると製品 1 は設備 A に 10 しか割り付けられない。$P = \{製品1\}$、$E = \{A, B\}$ となる。製品 2 は設備 B で生産するので製品 2 が追加され $P = \{製品1, 製品2\}$、$E = \{A, B, C\}$ となり、製品 3 は E の設備で生産しないので終了する。このとき E と等しい PU の {A, B, C} は負荷オーバーである (図 2)。



{A, B, C}が負荷オーバー

図 2: 負荷オーバーのパワーユニオン

3　パワーユニオンを求めるアルゴリズム

基本PUは製品情報から与えられる。すべてのPUを求めるのに、PUのすべての組合せを取るのは効率的ではない。そこでPUを2つずつ組み合わせて新しいPUを作り、既にあるPUしかできなくなれば終了することにより全PUを求める手法を示す。

アルゴリズムの前に、次のように定義されるPUの集合 S_i を考える。

パワーユニオンの集合 $S(i)$

$S_0 = \emptyset$
$S_1 = \{U \mid U\text{は基本パワーユニオン}\}$
$S_{i+1} = S_i \cup \{U \cup V \mid U, V \in S_i\}$

このとき明らかに次の2つの補題が成り立つ。

補題1　n 個の基本PUから生成されるPUは S_n に必ず含まれる。

補題2　$S_{k+1} = S_k$ ならば $S_i = S_k\ (i \geq k)$ である。

補題1、2により、S_i を順に生成していき差が無くなったとき終了すれば、全PUが求まることが明らかである。さらに次の補題が成り立つ。

補題3　S_i を

$S_0 = \emptyset$
$S_1 = \{U \mid U\text{は基本パワーユニオン}\}$
$S_{i+1} = S_i \cup \{U \cup V \mid U, V \in S_i - S_{i-1}\}$

と定義する。このとき

$S_{i+1} = S_i \cup \{U \cup V \mid U, V \in S_i\}$

である。

証明

$S_0 = \emptyset$ なので、$S_1 = S_1 - S_0$ である。よって $i = 1$ のときは正しい。

$i = k$ のとき正しいとする。

$S_k = S_{k-1}$ のとき定義より

$$S_{k+1} = S_k \cup \{U \cup V \mid U, V \in S_k - S_{k-1}\} = S_k$$

さらに

$$\begin{aligned} S_k \cup \{U \cup V \mid U, V \in S_k\} &= S_{k-1} \cup \{U \cup V \mid U, V \in S_{k-1}\} \\ &= S_k \\ &= S_{k+1} \end{aligned}$$

より $S_k = S_{k-1}$ のとき $i = k+1$ でも正しい。

$S_k \neq S_{k-1}$ のとき $S_k \supset \{U \cup V \mid U, V \in S_{k-1}\}$ より

$$\begin{aligned} S_k \cup \{U \cup V \mid U, V \in S_k\} &= S_k \cup \{U \cup V \mid U \in S_{k-1}, \\ &\qquad V \in S_k - S_{k-1}\} \\ &\quad \cup \{U \cup V \mid U, V \in S_k - S_{k-1}\} \end{aligned}$$

そこで

$$\begin{aligned} S_k \cup \{U \cup V \mid U \in S_{k-1}, V \in S_k - S_{k-1}\} \\ \subset S_k \cup \{U \cup V \mid U, V \in S_k - S_{k-1}\} \end{aligned}$$

を示せば $i = k+1$ のときも正しいことが言える。それは、S_k に含まれない $\{U \cup V \mid U \in S_{k-1}, V \in S_k - S_{k-1}\}$ の要素が $\{U \cup V \mid U, V \in S_k - S_{k-1}\}$ にすべて含まれることを示せば良い。

いま $S_k \neq S_{k-1}$ なので $Us \in S_k - S_{k-1}$ が存在する。このとき仮定より $Us \in \{U \cup V \mid U, V \in S_{k-1}\}$ なので $Ux \cup Uy = Us$ である $Ux, Uy \in S_{k-1}$ が存在する。ここで $Us \cup Uz \notin S_k$ なる $Uz \in S_{k-1}$ があると仮定する。$Ux \cup Uz \in S_k$ であり、$Ux \cup Uz \in S_{k-1}$ であれば $(Ux \cup Uz) \cup Uy \in S_k$ となって矛盾するため $Ux \cup Uz \in S_k - S_{k-1}$ である。同様に $Uy \cup Uz \in S_k - S_{k-1}$ である。$Us \cup Uz = (Ux \cup Uz) \cup (Uy \cup Uz)$ なので、$Us \cup Uz \in \{U \cup V \mid U, V \in S_k - S_{k-1}\}$ が言える。よって $i = k+1$ のときも正しい。□

補題3より、S_i に追加するPUの候補は新規に追加されたPU同士の和集合だけで良いことが分かる。そこで以下のアルゴリズムとなる。

パワーユニオン生成アルゴリズム

以下のアルゴリズムで終了したときの S_i が全パワーユニオンを含む。

初期設定

$S_0 = \emptyset$
$S_1 = \{U \mid U\text{は基本パワーユニオン}\}$
$i = 1$

繰り返し

while $(S_i - S_{i-1} \neq \emptyset)$ {
　$S_{i+1} = S_i \cup \{U \cup V \mid U \neq V$
　　　　$U, V \in S_i - S_{i-1}\}$
　$i = i + 1$
}

このアルゴリズムは、補題1～3によって正しくPUを求められることが分かる。

定理2

パワーユニオン生成アルゴリズムが終了したとき、S_i は全パワーユニオンを含む。

これまでは設備が競合関係にあるとして全PUを対象にしてきた。実際には競合関係に無い設備も混ざっており、また、定理1の証明より共通設備が無い2つのPUの和を取ってPUを作

る必要が無いことがわかる。そこで、PU 生成アルゴリズムは次のように改良することができる。

パワーユニオン生成アルゴリズム (改良版)

初期設定

$S_0 = \emptyset$

$S_1 = \{U \mid U は基本パワーユニオン\}$

$i = 1$

繰り返し

while $(S_i - S_{i-1} \neq \emptyset)$ {

$S_{i+1} = S_i \cup \{U \cup V \mid U \neq V$
$U \cap V \neq \emptyset \;\; U, V \in S_i - S_{i-1}\}$

$i = i + 1$

}

例 4　製品 1, 2, 3, 4 が {A, B, C}, {B, C}, {C, D}, {D, E} で生産可能とする。このとき

$S_1 = \{\{A,B,C\},\{B,C\},\{C,D\},\{D,E\}\}$

となる。次に、S_1 から共通設備のある PU を 2 つずつ取って和を取り S_1 に加えると

$$S_2 = \{\{A,B,C\},\{B,C\},\{C,D\},\{D,E\}, \{A,B,C,D\},\{B,C,D\},\{C,D,E\}\}$$

となる。さらに処理を続けると

$$S_3 = \{\{A,B,C\},\{B,C\},\{C,D\},\{D,E\}, \{A,B,C,D\},\{B,C,D\},\{C,D,E\}, \{A,B,C,D,E\},\{B,C,D,E\}\}$$

$S_4 = S_3$

となって終了する (図 3)。

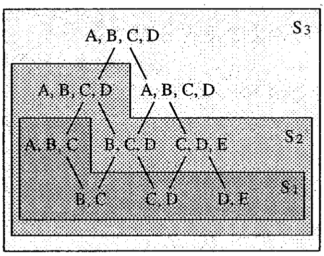


図 3: パワーユニオンの生成 (例 4)

4　生産計画への適用

生産現場で実際に PU による生産可能性チェックを行なう場合、いくつかの問題点がある。まず、バックトラックが不要な点は効率的であるが、PU の数が爆発的に増えた場合に逆にコストが掛かる可能性がある。実際の生産現場で試算した結果を示し、通常は問題にならないことを示す。また、生産現場では使用設備に優先順位を付け、個々の設備の割付状態をチェックすることが多い。そこで、使用設備に優先度がある場合の設備割付についても考察する。

4.1　パワーユニオンの数

PU は基本 PU の和集合を取るため数が爆発することも考えられる。しかし、生産現場では使用設備のパターンが限られていることが多く、まず爆発することはない。以下に実際の生産現場で試算した結果を示す。基本 PU が 1 つの設備種は省略した。

生産現場 1 (設備種 =10)

	1	2	3	4	5
設備数	31	92	43	17	136
基本 PU	4	21	2	12	30
PU	5	26	2	26	40

生産現場 2 (設備種 =8)

	1	2	3	4	5	6	7
設備数	6	7	10	5	5	5	4
基本 PU	3	3	9	4	5	7	6
PU	3	4	33	6	5	11	6

このように、実際の生産現場では稀に PU が基本 PU の数倍になることがあるが、ほとんどの場合わずかに増える程度である。また、数倍に増える場合でも負荷を積むのは製品の基本 PU と上位の PU だけであるので、PU の数が多くても製品毎に PU に負荷を積む回数は少ないことが多い。

4.2　設備への割付

計画に対して PU による能力チェックを行ない生産可能と判明したとき、個々の設備への割付を見たいことがある。条件はすべて線形式であり解があることが分かっているので、線形計画法で解くことができる。

解が複数ある場合には、優先順位を付けることができる。製品から見た設備の優先順と、設備から見た製品の優先順があるため、次のように設備と製品の組み合わせの優先順が高ければ小さい重みを付けて目的関数を最小化する。

設備への割付

各製品 $p(i)$ に該当する基本パワーユニオン $U(i)$ の負荷を $L(i)$ とし、設備 $e(j)$ の能力を $c(j)$、設備 $e(j)$ で生産する製品 $p(i)$ の量を $a(i,j)$、その優先度 (重み) を $w(i,j)$ と表す。$a(i,j)$ が変数で求める値である。

条件式

$$\sum_{\{i \mid e(j)\in U(i)\}} a(i,j) \le c(j)$$

$$\sum_{\{j \mid e(j)\in U(i)\}} a(i,j) = L(i)$$

$$a(i,j)\begin{cases}\ge 0 & (e(j)\in u(i))\\ = 0 & (e(j)\notin u(i))\end{cases}$$

目的関数

$$\sum w(i,j)\times a(i,j)$$

例 5　設備 A, B, C, D の能力が 10, 50, 50, 10 で製品 1, 2, 3 が {A, B}, {B, C}, {C, D} で生産可能で 20, 40, 50 生産するとする。この計画は図 4のように生産可能であり、設備への割付もいろいろと考えられる。

条件式は次のようになる。ここでは設備名をそのまま $a(i,j)$ の第 2 項に入れる。

$a(1,A) \le 10$
$a(1,B) + a(2,B) \le 50$
$a(2,C) + a(3,C) \le 50$
$a(3,D) \le 10$
$a(1,A) + a(1,B) = 20$
$a(2,B) + a(2,C) = 40$
$a(3,C) + a(3,D) = 50$

例えば製品 1 は設備 B をできるだけ使わず、製品 2 は設備 B を可能なら使いたくない場合には、例えば $a(1,B)$ と $a(2,B)$ の重みを 2 と 1、それ以外の重みを 0 として目的関数

$2 \times a(1,B) + a(2,B)$

を最小化すれば、製品 1 は設備 A と B に 10 ずつ、製品 2 は B と C に 30 と 10、製品 3 は C と D に 40 と 10 の割付となる。

設備	能力
A	10
B	50
C	50
D	10

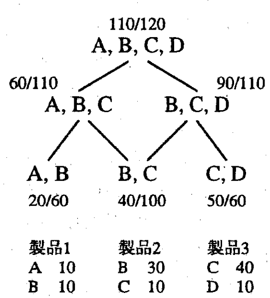


図 4: 設備割付の例 (例 5)

4.3　その他の問題点

実際の生産現場においては使用設備に優先度があることが多く、設備割付時に考慮する方法は前節で検討した。その他に、例えば通常は使用しないが納期に間に合わない場合は使用する、など他の制約との関連で生産可能性が変わる場合が考えられる。その場合には、優先度の低い設備を使用する場合に納期違反のペナルティより低いペナルティを課してトータルでペナルティを抑える、またはまず優先度の低い設備を使用せずに立案し納期に間に合わなければ設備を追加する、などが必要になる。

また、同じ製品でも使用設備により負荷が変わる場合には、PU として統一的に扱うと誤差が生じる可能性がある。どの製品でも負荷の割合が同じであれば、能力を換算して適用するなどが可能である。例えば、設備 A は設備 B の 2 倍の負荷が掛かる場合、設備 A の能力を半分として扱えば、同じ負荷を掛けてチェックすることが可能である。製品と設備により負荷がまったく変わる場合は、誤差が出るが負荷の平均を取ることもできる。

5　まとめ

大規模生産計画を実用的な時間で立案するには、バックトラックをする余裕は無い。そこでバックトラックせず処理コストが小さい生産能力検証手法が望まれていた。本稿で提案する手法はそのような条件を満たし、しかも厳密に生産可能性をチェックできる。また、生産可能性をチェックする単位となるパワーユニオンの数も、原理的には爆発する可能性があるが、実際の生産現場では問題無いことも確認した。問題無いのは、通常は使用可能設備にパターンがあるためである。使用可能設備に複雑な選択が入る生産現場では確認が必要となる。また、パワーユニオンを効率良く作成するアルゴリズムとその正当性を示した。十分実用になると思われる。

参考文献

[1] 野村、中薹：厳密モデルと近似モデルの併用による大規模生産計画問題の最適解計算の試み, 生産スケジューリング・シンポジウム'98 講演論文集, 1998.

[2] 毛受：生産目標計画における平準調整手法, 生産スケジューリング・シンポジウム '95 講演論文集, 1995,pp.187–192.

[3] 吉川：制約最適化技術のスケジューリング問題への応用, 人工知能学会誌 Vol.13 No.3, 1998, pp.379–386.