繰返し近似によるジョブショップスケジューリング*

木 瀬

洋*¹

Jobshop Scheduling by an Iterative Approximation

Hiroshi KISE

This paper proposes an iterative approximation method for minimizing the maximum lateness for a jobshop in which n jobs with ready times and due dates are processed on m machines in different orders. The proposed method consists of two phases I and II. Phase I sequences m machines one by one successively to obtain a feasible schedule of the m -machine shop, repeats this procedure m times by changing the first machine to be sequenced, and selects the best one. Phase II locally reoptimizes each machine based on sequences of the remaining $(m-1)$ machines that have been the best ones obtained thus far. This procedure is repeated until no improvement is possible. Computer simulations show that the proposed method can give schedules comparable to the SBI method by Adams et al., and better ones than representative dispatching rules.

Key Words: Production System, Production Management, Production Planning, Jobshop Scheduling, One-Machine Scheduling, Maximum Lateness

1. 緒 言

代表的なジョブショップスケジューリング問題(以後, JSP と記す)の一つは次のように記述される⁽¹⁾.

(a) それぞれ1機械からなる m 工程生産システムで n ジップが処理される。ただし, 各工程は十分な容量のバッファを有する。

(b) 各工程による各ジョブの処理(作業という)の処理時間はあらかじめ与えられている。ただし, 段取時間や搬送時間は考慮しない。

(c) 各工程は一時に高々1作業を処理し, 作業の中断は許されない。

(d) 全ジョブの終了時間(総完了時間)が最小となる様に各工程における作業の最適順序を決定する。

JSP は組合せ最適化の中でも最も難しい問題とされ, その研究は Johnson 以来, 30 余年を経た今日でさえなお活発である。この間の多数の研究報告⁽¹⁾は厳密解法と近似解法に大別される。前者は分枝限定法⁽²⁾⁻⁽⁴⁾が大半である。しかしながら, これらの方法は計算時間の点で大規模問題に対応できない。ごく最近, ペン

チマークとして有名な $(m \times n) = 10 \times 10$ 問題⁽⁵⁾の最適解が初めて得られたとの報告⁽⁶⁾がある程度である。他方, 近似解法は単純な発見的規則(ディスパッチングルール)が大半である。これまで多くのルールが提案され⁽⁷⁾, 総完了時間に関しては MWKR⁽⁸⁾, 納期遅れについては最小スラック規則⁽⁹⁾が比較的良いと報告されているが, 常に他を優越するルールは存在しない。これら両接近法の間には計算時間と近似精度の両面において大きなギャップがある。この意味において Adams ら⁽¹⁰⁾によって最近開発された移動隘路法(Shifting Bottleneck Procedure)が注目されている。この方法は m 工程問題を m 種の1機械問題(以後, OSP と記す)に分解し, これらを順次解くことによって元問題の近似解を得る SBI 法が中心となっている。OSP に分解する考えは既に下限の計算法に用いられているが⁽³⁾, SBI 法の特徴は m 工程の中から特定した隘路工程に対する OSP を厳密に解くという手順を繰返すことによって JSP の近似解を得る点にある。数値実験では $m \leq 10, n \leq 30$ の範囲でディスパッチングルールより常に良い解を得ている。しかし, この方法の核となる OSP 自身が実は強 NP 完全であるので⁽¹¹⁾, 解法として分枝限定法⁽¹²⁾を用いている。この分枝限定法は大規模問題も短時間で解くことができるこ

* 原稿受付 平成2年10月18日。

*¹ 正員, 京都工芸繊維大学工芸学部(〒606 京都市左京区松ヶ崎御所海道町)。

とが数値実験結果として示されているが、NP 完全理論によれば⁽¹⁾、この事実がどんな問題例に対しても成り立つことを保証することにはならない。したがって SBI 法は近似解法にもかかわらず n や m の多項式時間で解を見出す保証がない。

本研究では以下に示すようにより一般化された JSP を対象とする。すなわち、前述の (a)～(c) に加えて、

(e) 各ジョブは準備時間を有し、これより早く開始できない。準備時間はジョブの到着時間や開始前の段取りを表し、準備時間が異なる場合の JSP は動的であると言われることがある⁽¹⁾。

(f) 各ジョブは納期を有し、最大納期ずれ (Max. Lateness) 最小化が求められている。最大納期ずれは正の値をとるとき、納期遅れ (Tardiness) となり、納期零のときは総完了時間となる。

準備時間や納期は現場では極めて重要視されるが、問題をより難しくするためか、これまであまり取り上げられていないようである。この問題に対してここで提案する近似解法 (Machine By Machine Scheduling, 以後、MBMS 法と記す) はフェーズ I とフェーズ II からなる。フェーズ I では SBI 法と同じく OSP をくり返し解くが、前述の計算上の不安を取り除くため分枝限定法の代わりに Kise らの近似解法 ms⁽¹³⁾ を用いる。ms 法は病的な例 (人工的に構成された最悪例) を除けば、相対誤差が 2% 以下の近似解を与えることが数値実験で示されている。次に OSP を解く順序は隘路工程に関係なく機械的である代わり、順序の出発点を変えることによって m 種の解 (代替案) を生成し、それらの最良案を初期最良解とする。フェーズ II では初期最良解をベースにして各機械に対する OSP の再最適化が行われる。MBMS 法の基本的な考え方は繰返し代替案を生成することによって SBI 法で行われているような隘路工程の特定と OSP の厳密解によるメリットをカバーしようという点にある。いうまでもなく MBMS 法は SBI 法に比して計算時間とプログラミングの容易さにおいてはるかに有利である。

2. 問題の定式化

JSP における n ジョブの集合を $J = \{J_1, J_2, \dots, J_n\}$, m 機械の集合を $M = \{M_1, M_2, \dots, M_m\}$ と記す。また、ジョブ J_i に属する作業集合も J_i, M_k で処理される作業集合も M_k で表す。各ジョブ J_i の準備時間および納期をそれぞれ r_i, d_i 各作業 u の処理時間を $p(u)$ と記す。このとき、JSP は以下で定義されるグラフ $G(N, A, D, P)$ で表わされる。 $G(N, A, D, P)$ はディスジャ

ンクティブグラフと言われることがある⁽²⁾。図 1 に 3×2 JSP に対するグラフの例を示す。

$N = \{0, *\} \cup \bigcup_{i=1}^n N_i$: ノードの集合,

$N_i = \{n_{-i}, \bar{n}_i\} \cup J_i$

$n_{-i} = (m+2)(i-1)+1$

$\bar{n}_i = (m+2)_i$

$J_i = \{n_{-i}+1, \dots, \bar{n}_i-1\}, i=1, 2, \dots, n \dots\dots (1)$

ノード零, $*$ はそれぞれ計画の開始と終了を表すダミーノードである。 $p(n_{-i})$ と $p(\bar{n}_i)$ はジョブ J_i の準備時間と納期に相当する。 J_i はジョブ J_i の作業集合である。

$A = \bigcup_{i=1}^n A_i$: 先行関係を表すアーク集合,

$A_i = \{(0, n_{-i})\} \cup \{(u, u+1) | u = n_{-i}, n_{-i}+1, \dots, \bar{n}_i-1\} \cup \{(\bar{n}_i, *)\}, i=1, 2, \dots, n \dots\dots (2)$

$(0, n_{-i})$ はノード零が n_{-i} に、 $(u, u+1)$ は作業 u が $u+1$ 、また、 $(\bar{n}_i, *)$ は $\bar{n}_i$ が $*$ にそれぞれ先行することを意味する。すなわち、 A_i はジョブ J_i の作業列に相当する。(図 1 の例では実線のアークが A に相当する。)

$D = \bigcup_{k=1}^m D_k$: ディスジャンクティブアーク対 (Disjunctive Arc Pair) の集合,

$D_k = \{(u, v), (v, u) | u, v (\neq u) \in M_k\}$

$k=1, 2, \dots, m \dots\dots (3)$

(u, v) は機械 M_k において作業 u が v に先行した順序付けを (v, u) はその逆を意味する。(図 1 では破線のアークに相当する。) よって機械 M_k における作業の順序付けは $(u, v), (v, u)$ のいずれか一方を選び、他方を棄却することによって行われる。

$P = \{p(u) | u \in N\}$: 処理時間の集合,

$p(0) = p(*) = 0$

図 1 グラフ $G(N, A, D, P)$ による JSP の表現

図 2 グラフ $G(N, A', \phi, P)$ によるスケジュールの表現

$$p(n_{-i})=r_i, p(\bar{n}_i)=-d_i, i=1, 2, \dots, n,$$

$$p(u), u \in u_{i=1}^n J_i: \text{作業 } u \text{ の処理時間} \dots\dots\dots (4)$$

各ノード u の零からの最短経路長を $T(u)$ と記し,

$$T=\{T(u)|u \in N\} \dots\dots\dots (5)$$

をスケジュールという。このとき $T(u)$ は作業 u の最早開始時刻を表し, JSP は次のように定式化される。

$$T(0)=0 \dots\dots\dots (6)$$

$$T(u)+p(u) \leq T(v), (u, v) \in A_i, i=1, 2, \dots, n \dots\dots\dots (7)$$

$$[T(u)+p(u) \leq T(v)] \vee [T(v)+p(v) \leq T(u)], \\ ((u, v), (v, u)) \in D_k, k=1, 2, \dots, m \dots\dots\dots (8)$$

という条件のもとで

$$L(T) \equiv T(*) = \max\{T(\bar{n}_i) - d(\bar{n}_i) | i = 1, 2, \dots, n\}, \dots\dots\dots (9)$$

を最小にする最適スケジュール T^* を決定する。式(6)は計画開始時刻が零であることを意味する。式(7)はジョブ内の作業順序制約 [1の条件(a), (c)および式(4)参照]を表す。式(8)の $\vee$ は OR 記号で、不等式のいずれか一方が成立しなければならないことを表している。これはディスジャンクティブアーク対 $(u, v), (v, u)$ のいずれか一方を選び、他方を棄却することに相当する。この結果、機械 M_k の作業順序

$$S_k=(s_k(1), s_k(2), \dots, s_k(n)) \dots\dots\dots (10)$$

が決定する。ただし、 $S_k(i)$ は処理されるべき i 番めの作業を表す。以後、順序 S_k に対応する D_k のアーク集合もアーク集合 S_k と記すことにする。アーク集合 S_k が選ばれたときのグラフを $G(N, A \cup S_k, D - D_k, P)$ で表現する。このとき、グラフ $G(N, A \cup_{k=1}^m S_k, \phi, P)$ は JSP のスケジュールを表し、 $T(u)$ は作業 u の開始時刻となる。ただし、 $T(n_{-i}), T(\bar{n}_i)$ はそれぞれ、ジョブ J_i の開始と終了を表す。よって式(9)の $L(T)$ は式(4)より最大納期ずれとなる [図2にグラフ $G(N, A \cup_{k=1}^m S_k, \phi, P)$ の例を示す]。

3. 1 機械スケジューリング

提案する JSP の近似解法 (MBMS) は機械ごとに順序付けを行う。機械 M_k の順序付けの際、既に順序付けされている機械の集合を K とする。このとき、機械 M_k に対する 1 機械スケジューリング問題 (OSP) は

$$\text{OSP: } G(N, A_k, D_k, P), A_k = A \cup_{k \in K} S_k \quad (11)$$

で定義される。もし、 $K=M-\{M_k\}$ ならば、OSP の解は JSP の近似解となり、両者の最大納期ずれは一致する。次にグラフ $G(N, A_k, \phi, P)$ において $L(u, v)$ をノード u から v までの最大経路長とする。このとき、

$$r(u)=L(0, u)$$

$$d(u)=L(0, *)-L(u, *)+p(u), u \in M_k \dots\dots\dots (12)$$

とすると、式(11)の OSP は準備時間 $r(u)$ 、処理時間 $p(u)$ 、納期 $d(u)$ を有する最大納期ずれ最小化 1 機械スケジューリング問題と見なすことができる⁽¹³⁾。この問題は強 NP 完全で、いかなる方法によっても n の多項式時間で厳密解を得る保証はないとされているので、ここでは極めて最適に近い近似解を与えることが確かめられている Kise ら⁽¹³⁾ の近似解法 ms を用いる。ms 法は以下の Schrage のアルゴリズム S と問題の可逆性に基づく。

アルゴリズム S 機械は処理中の作業を終了するとにその時刻 t までに準備完了している作業 $u[r(u) < t]$ の中から最小納期のジョブを次に処理する。

問題の可逆性 準備時間 $r(u)$ 、納期 $d(u)$ を持つ問題に対して準備時間 $-d(u)$ 、納期 $-r(u)$ を持つ問題を逆問題という。このとき、逆問題の最適順序の逆順序は元の問題の最適順序となり、両問題の最大納期ずれは一致する。ただし、両問題に同じ近似解法を適用しても同じ近似解が得られるとは限らない⁽¹³⁾。よって次の近似解法 $\bar{S}$ は有意義である。

アルゴリズム $\bar{S}$ 逆問題にアルゴリズム S を適用し、得られた順序の逆順序を近似解とする。

これらのことに基づいて近似解法 ms は次のように記述される。

近似解法 ms 与えられた問題例にアルゴリズム S と $\bar{S}$ を適用し、得られた結果の良い方を選ぶ。

4. 近似解法 MBMS

近似解法 MBMS はフェーズ I と II からなる。フェーズ I では JSP の初期最良解を得る。すなわち、ある機械 M_f を出発点とし、機械 $M_k, k=f, f+1, \dots, m, 1, 2, \dots, f-1$ について m 個の OSP をこの順番で解き、順序 $S_k, k=1, 2, \dots, m$ 、および最大納期ずれ L^* を得る。この手順を $f=1, 2, \dots, m$ についてこの順番で繰返すことによって m 個の JSP の解が得られ、その中の最良順序を $S^*=\{S_k^*|k=1, 2, \dots, m\}$ および S^* の最大納期ずれを L^* とする。フェーズ II では機械 M_k 以外は S^* に固定した状態で M_k について更めて OSP を解く。この結果さらに良い解が得られれば、 S_k^* および L^* を更新する。この手順を $k=1, 2, \dots, m$ についてもはや解の更新の可能性がなくなるまで繰返す。以下に手順の詳細を示す。

アルゴリズム MBMS

(フェーズ I)

手順 1 (フェーズ I の初期化)。

(a) 与えられた JSP をグラフ $G(N, A, D, P)$ に変換。

(b) $f \leftarrow 0, L^* \leftarrow \infty$

手順 2 (OSP の初期化)。

$f \leftarrow f+1, i \leftarrow 0, L^0 \leftarrow -\infty$

手順 3 (OSP を解く)。

(a) $k \leftarrow (f+i-1) \bmod(m)+1$

(b) OSP: $G(N, A \cup S, D_k, P)$ を近似解法 ms によって解く。ただし、

$$S = \begin{cases} S_f \cup S_{f+1} \cup \dots \cup S_{k-1}, & k > f \\ \phi, & k = f \\ S_f \cup \dots \cup S_m \cup S_1 \cup \dots \cup S_{k-1}, & k < f \end{cases}$$

とする。得られた M_k の順序を S_k , 最大納期ずれを L とする。

(c) $L^0 \leftarrow \max\{L^0, L\}$

(d) $i < m-1$ ならば, $i \leftarrow i+1$ として手順 3 を繰返す。 $i = m$ ならば, JSP の実行可能解が得られた。手順 4 へ進む。

手順 4 (L^* の更新)。

(a) $L^0 < L^*$ ならば

$L^* \leftarrow L^0, S_k^* \leftarrow S_k, k=1, 2, \dots, m$ 。

(b) $f < m$ ならば, 手順 2 へ戻る。 $f = m$ ならば,

$i \leftarrow 0, g \leftarrow 1$ 。

としてフェーズ II へ進む。

(フェーズ II)

手順 5 (OSP を解く)。

(a) $k \leftarrow i \bmod(m)+1$ 。

(b) OSP: $G(N, A \cup (S^* - S_k^*), D_k, P)$ を ms 法で解き, 得られた順序を S_k , 最大納期ずれを L^0 とする。

(c) もし, $L^* > L^0$ ならば

$S_k^* \leftarrow S_k, S^* \leftarrow \{S_k^* | k=1, 2, \dots, m\}$,

$L^* \leftarrow L^0, i \leftarrow i+1, g \leftarrow 1$,

として手順 5 を繰返す。そうでなければ, 手順 6 へ進む。

手順 6 (終了テスト)。 $g < m$ ならば,

$g \leftarrow g+1, i \leftarrow i+1$

として手順 5 へ戻る。そうでなければ, 計算終了。 S^*, L^* が求める JSP の近似解である。□

フェーズ II において $g = m$ は解の更新なしにフェーズ II を m 回繰返したことを意味するので, この手順ではこれ以上解の改良が望めない。なお, OSP を解く際に必要な $L(0, u), L(0, *), L(u, *)$ [式(12)参照] の計算は $P(u)$ に正負の値が含まれることを考慮して逐次反復法⁽¹⁴⁾を用いている。図 3 に MBMS 法におい

て OSP が繰返し解かれる過程を示す。

5. 数 値 実 験

MBMS 法の性能を検討するために 2 種の数値実験を行った。第 1 はよく知られたベンチマーク問題を通して前述の SBI 法と比較する。第 2 はより大規模な問題に対する代表的なディスパッチングルールとの比較である。以下にこれらの結果を示す。

5.1 SBI 法との比較 これまでベンチマークとしてしばしば用いられてきた 3 種の問題例⁽⁵⁾ に対する SBI 法の結果が公表されている⁽¹⁰⁾ ので, これと比較した結果を表 1 に示す。これより MBMS 法は SBI

表 1 ベンチマーク問題に対する SBI 法との比較

EX.	m	n	SBI		MBMS	
			Value	CPU(s)	Value	CPU(s)
1	6	6	55	1.5	55	0.13
2	10	10	1015	10.1	1020	1.45
3	5	20	1290	3.5	1235	0.38
Computer Used			VAX 780/11		ACOS 850/10

図 3 近似解法 MBMS の繰返し計算過程

法と同程度の結果を得ていることがわかる。なお、計算時間は使用した計算機が異なるので一概に比較できないが参考のため示した。

5・2 代表的なディスパッチングルールとの比較

次の3種のディスパッチングルールを取り上げた。

MINSL 法⁽²⁾⁽⁹⁾ アクティブな各作業 u について最遅開始時刻 $d(u)-p(u)$ と最早開始時刻 $r(u)$ の差で与えられる余裕, $SL(u)=d(u)-p(u)-r(u)$ 最小作業を次にスケジュールする [式(12)参照]。

BM 法⁽⁴⁾ Barker と McMahon によるもので、実行可能な作業群の中から最早開始時刻を持つ作業を次にスケジュールする。ただし、同値の場合は納期の小さい順、それも同値の場合は処理時間の小さい順とする。

LLR 法⁽⁹⁾ Lagewege, Lenstra, Rinnooy Kan によるもので実行可能かつ最早終了が見込まれる作業を処理する機械 M_k を次にスケジュールする。すなわち、 M_k に属する作業 u のうち、 M_k の最早終了時間と作業 u が属するジョブの最早終了時間の大きい方を最小にする作業を次に M_k にスケジュールする。LLR 法は MWKR 法⁽⁸⁾ の拡張である。

これらのディスパッチングルールの中には準備時間 r_j や納期 d_j を考慮しないものもあるので以下の実験では $r_j=d_j=0$ とした。この場合、評価規準は総完了時間 $C_{\max}$ となる。ただし、ここでは得られた $C_{\max}$ の絶対的な良さをも見るため、簡単な下限値

$C_{lb} = \max \{ \sum_{u \in M_k} p(u) | k=1, 2, \dots, m \}$ (13)
との差

$$\Delta C_{\max} = C_{\max} - C_{lb} \text{ (14)}$$

図4 ジョブ数 n が近似解法の精度に及ぼす影響

で評価した。なお、処理時間 $p(u)$ は1から100の間の離散型一様乱数で与え、作業の機械割当も一様乱数で決定した。また、以下の実験結果(図4～図7)はすべて20例題の平均値を示している。

図4と図5はそれぞれジョブ数 n と機械数 m が $\Delta C_{\max}$ に及ぼす影響を示す。図6はこれらの結果を補足するために C_{lb} についての結果を示す。表2は得られた結果の有意差をTテストによって検計したものである。また、図7はMBMSの計算時間を示している。これらの結果から次の結論が得られる。

(a) MBMS法は比較したどのディスパッチングルールより高い近似性能を有する。

(b) ジョブ数が多いとき、あるいは機械数が少ないとき、MBMS法は最適値に近い解を与える。

(c) 機械数が多くなるに従って $\Delta C_{\max}$ は大となるが、これは必ずしも最適値からのずれが大きいことを意味しない。なぜなら図6に示すように、 C_{lb} は m に対してほとんど変化せず、大きい m に対して極めて緩い下限となっている。

(d) Tテストの結果は図4と図5に見られるほどの有意差を示していない。これはここでは示していないがディスパッチングルールの結果のバラツキが大きいと思われる。これに対しMBMS法のバラツキは比較的小さい。

(e) MBMS法の計算時間は $O(n^2 \sim n^3)$ の割合で増加する。これは最短経路の計算に最悪の場合、 $O(n^3)$ の計算時間を要するためと考えられる。ここでは示し

図5 機械数 m が近似解法の精度に及ぼす影響

表 2 総完了時間 $C_{\max}$ の T -test(対 MBMS 法)

○…MBMS 法のほうが有意に有効である。
 —…有意の差はない。(有意水準 5%, 片側検定, 20 例)
 ※…20 例中, MBMS 法より良い $C_{\max}$ は存在しなかった。

			ジョブ数				
			20	40	60	80	100
MINSL 法	工程数	2	-※	-※	-※	-※	-※
		4	○※	-※	-※	-※	-※
		6	○	○	-	-	-
		8	○	○	-	-	-
		10	○	-	-	-	-
LRL 法	工程数	2	-	-※	-※	-※	-※
		4	○	○※	○※	○※	-※
		6	○	○※	○	○	○※
		8	○	○	○	○	○
		10	-	-	-	○	-
BM 法	工程数	2	-	-※	-※	-※	-※
		4	○	○※	○※	○※	-※
		6	○	○※	○※	○※	○※
		8	○	○	○	○※	○※
		10	○	○	-	○	○

図 6 下限 C_{lb} の値

図 7 MBMS 法の計算時間

ていないが, m についても同様の傾向が見られた。これに対しディスパッチングルールの計算時間は MBMS 法の 1/10~1/20 程度である。

6. 結 論

準備時間と納期を有するジョブショップスケジュー

リング問題 (JSP) に対し 1 機械問題 (OSP) を繰返し解くことによって近似解を得る手法 MBMS を提案した。MBMS 法は Adams らの SBI 法とほぼ同程度、また、分枝限定法の初期解として使われている代表的なディスパッチングルールより良い近似を与えることが数値実験によって示された。近年の計算機の高高速化を考えると、MBMS 法のような反復的手法は単純なディスパッチングルールよりむしろ実用的であると言える。一方、JSP がさらに実用的であるためには、代替機械、有限バッファ、搬送時間、段取時間^{(15)~(17)} やリスケジュールリング⁽¹⁸⁾ も無視できない。これらに対する MBMS 法の拡張を今後の検討課題としたい。

最後に、数値実験の一部は橋本孝司氏[現、住金システム開発(株)]、大規高幹氏[現、富士通テン(株)]の卒研によることを記したい。また、京都大学の茨木俊秀教授および大阪産業大学の串山正教授には有益な助言をいただいたことに対し深謝の意を表す。

文 献

- (1) 例え、French, S., Sequencing and Scheduling, *An Introduction to the Mathematics of the Job Shop*, (1982), Wiley.
- (2) Balas, E., *Oper. Res.*, 17 (1969), 941.
- (3) Lageweg, B. J., Lenstra, J. K. and Rinnooy Kan, A. H., *Management Sci.*, 24 (1977), 441.
- (4) Barker, J. R. and McMahon, G. B., *Management Sci.*, (1985), 594.
- (5) Muth, J. F. and Thompson, G. L. (関根訳), インダストリアルスケジューリング, (1966), 280, 竹内書店.
- (6) Carlier, J. and Pinson, E., *To Appear in Management Sci.*, (1988).
- (7) Panwalker, S. S. and Iskander, W., *Oper. Res.*, 25 (1977), 45.
- (8) 宮崎・橋本, 日本経営工学会誌, 27 (1976), 154.
- (9) 中沢・近藤, 精密工学会誌, 52 (1986), 471.
- (10) Adams, J., Balas, E. and Zawack, D., *Management Sci.*, 34 (1988), 391.
- (11) Garey, M. R. and Johnson, D. S., *Computer and Intractability, A Guide to the Theory of NP-Completeness*, (1979), Freeman.
- (12) Carlier, J., *European J. Oper. Res.* 11 (1982), 42.
- (13) Kise, H., Ibaraki, T. and Mine, H., *J. Oper. Res. Soc. Japan*, 22 (1979), 205.
- (14) 茨木, アルゴリズムとデータ構造, (1989), 昭晃堂.
- (15) 由良・人見, 機論, 48-431, C (1982), 1094.
- (16) 岩田・ほか 4 名, 機論, 49-437, C (1983), 133.
- (17) 李・ほか 2 名, 機論, 54-502, C (1988), 1370.
- (18) 藤井・ほか 2 名, 1990 年度 OR 学会秋期研究発表会予稿集, 162.