

【論文】

ファジィ作業時間をもつ非一様型並列機械 スケジューリング問題の最適化解法

董彦文

【概要】本研究では、ファジィ作業時間をもつ非一様型並列機械スケジューリング問題を対象として、分岐限定法を用いて、新たな分岐操作と下界値の計算手順を考案のうえ、ファジィスケジューリング最適化解法を提案する。また、数値計算例を通じて提案解法の適用性を示し、その有効性を明らかにする。

1. はじめに

スケジューリングとは、与えられた仕事(ジョブ)をいつ、どの機械(生産機械、作業員など)に割当てるかを計画することである。スケジューリング問題はいろいろなタイプに分類されるが、並列機械スケジューリング問題とは、 $m(m>1)$ 台の機械と n 個のジョブが与えられ、各ジョブが任意の1台機械で加工できる場合のスケジューリング問題である。さらに、ジョブの作業時間が機械とジョブとの両方に依存すれば、非一様型並列機械スケジューリング問題と呼ばれる。

総所要時間の最小化を目的関数とする非一様型並列機械スケジューリング問題は、並列処理システムのコンピュータ運用計画、複合作業機械を配置したFMS(フレキシブル生産システム)のスケジューリング、組立ロボットの動作計画などに幅広く応用できることがよく知られている。この問題についてもこれまでにいくつかの研究がなされて、Ibarraら〔1〕とDavisら〔2〕が発見的近似解法を、Lenstraら〔3〕が線形計画法に基づく近似解法を、野村ら〔4〕が分岐限定法に基づく最適化解法を提案している。

しかしながら、これまでの研究では、作業時間を既知の定数とするものがほとんどであるのに対して、実際の生産現場では、作業時間が正確に決められない場合が多い。特に受注生産の場合、作業時間をあらかじめ決められず人間の経験と主観判断により大まかな作業時間しか与えられないことは良く見られる。作業時間が不確かな場合には、3点見積もり法という確率的取扱い方が主に用いられてきた。最近、不確かな作業時間をファジィ数として取り扱う研究が注目されている〔5〕～〔9〕。このファジィ的取扱いは、作業時間を確率的なものではなく、あいまいなものとして取り扱う点で従来の確率的取扱い方と基本的に発想が異なり、人間の主観的判断に合致しており、管理者に理解しやすいなどの利点を持っている。その反面、ファジィ作業時間の表現と演算方法が通常の数値を扱う場合と異なるので、これまでに提案された数多くの通常スケジューリング解法は、そのままファジィ作業時間を持つスケジューリング問題の解決には適用できなくなり、従来の解法を修正し

たり、新しい解法を提案する必要が出てきたりして、いろいろな計算上の問題が生じる。

そこで、本論文では、非一様型並列機械スケジューリング問題を対象として、作業時間が三角型または台形型ファジィ数の場合、新たな分岐操作と下界値の計算手順を考案のうえ、分岐限定法を用いて、ファジィスケジューリング最適化解法を提案する。また、数値計算例を通じて提案解法の適用性を示し、その有効性を明らかにする。

2. 問題の定義

本研究は、次の非一様型並列機械スケジューリング問題を対象とする。

- (1) n 個のジョブ $\{J_i; i=1, 2, \dots, n, n>1\}$ を、性能の異なる m 台の機械 $\{M_j; j=1, 2, \dots, m, m>1\}$ で加工する。
- (2) 各ジョブは任意の 1 台の機械だけで加工できるが、加工時間は使用機械とジョブとの両方に依存し、機械 M_j でジョブ J_i を加工する場合の加工時間を t_{ij} とする。そして t_{ij} は普通の定数ではなく、正規性を持つファジィ数である。
- (3) ジョブの加工時間は、正味加工時間と段取り時間を含む既知のファジィ数であり、ジョブの加工順序によって変わらない。
- (4) 各ジョブ間には先行関係がなく、任意の順序で加工できる。ただし、1つのジョブを分割して加工したり、1台の機械で2個以上のジョブを同時に加工したりすることはできない。
- (5) 計画期間の初めに、 n 個のジョブは全部着手可能である。また、計画期間にわたって、全部の機械が常に使用可能である。
- (6) 目的関数は総所要時間の最小化とする。

また、本論文で使う記号を次のように定義する。

- t_{ij} : M_j で J_i を加工する場合の作業時間;
 $\mu_{t_{ij}}(x)$: t_{ij} のメンバーシップ関数;
 $m(t_{ij})$: t_{ij} の一般化期待値 GMV;
 $\text{GMV}(t_{ij})$: t_{ij} の GMV の値を求める演算子;
 x_{ij} : M_j で J_i を加工することを表す 0-1 変数;
 (+) : ファジィ加法演算子;
 Fuzzy Min : ファジィ最小化演算子。
 Fuzzy Max : ファジィ最大化演算子。

上記の記号を用いれば、対象とするスケジューリング問題は次のように定式化することができる。

問題 P

$$\text{Minimize } C = \text{Fuzzy Max } \{C_j; j=1, 2, \dots, m\} \quad (1)$$

Subject to

$$C_j = \left(\sum_{i=1}^n x_{ij} t_{ij} \right) \quad (2)$$

董：ファジィ作業時間をもつ非一様型並列機械スケジューリング問題の最適化解法

$$\sum_{i=1}^n x_{ij} = 1 \quad (3)$$

$$x_{ij} = \begin{cases} 1; & M_j \text{ で } J_i \text{ を加工するとき} \\ 0; & \text{その他} \end{cases} \quad (4)$$

$$i=1, 2, \dots, n; \quad j=1, 2, \dots, m.$$

3. 本研究の位置づけ

ファジィ作業時間を持つスケジューリング問題に関しては、McCahonら [5] が Compell, Dudek & Simth の提案したフローショップスケジューリング問題の近似解法を、台形型ファジィ作業時間の場合に拡張した。Tsumimuraら [6] は、Ignall & Schrage の提案した3工程フローショップ総所要時間最小化問題の分岐限定解法を、三角型ファジィ作業時間の場合に拡張した。また、Tsumimuraら [7] は、ファジィ作業時間を持つジョブショップスケジューリング問題を取り上げ、Genetic Algorithm を用いた近似解法を与えた。また、董ら [8] [9] はファジィ作業時間をその一般化期待値 GMV で置き換えて、得られた通常スケジューリング問題の解を求めることにより、元のファジィスケジューリング問題の解を求める簡便解法を提案した。

しかし、非一様型並列機械スケジューリング問題に関しては、他のタイプのスケジューリング問題に比べて研究が少なく、不十分なところが多いため、ファジィ作業時間を持つ非一様型並列機械スケジューリング問題に関する研究はさらに少なく、董らの研究以外には他の研究報告がまだ見られていない。特に作業時間がファジィである場合の最適化解法は報告されていない。

これまでの研究に比べて、本研究は以下のことを特色として、重点に置き考究したい。

- (1) これまでにあまり研究されていないファジィ作業時間を持つ非一様型並列機械スケジューリング問題を取り上げる。
- (2) ファジィ作業時間を考慮した場合、分岐限定法を適用するのに必要な下界値の計算方法を新たに提案する。
- (3) 分岐限定法に基づく最適化解法を提案する。提案解法は大規模な問題の解を求めるには向かないかも知れないが、他の有効解法を提案しその効果を検証するために、問題の最適解を求めることが不可欠であるので、この意味では最適化解法を提案することは必要かつ重要である。
- (4) 通常のスケジューリング問題に比べて、ファジィスケジューリング問題は、作業時間がファジィ数であるため、その解を求める過程がより複雑で難しい。計算資源に対する要求などの面では、ファジィ作業時間を考慮することにより、どんな変化が生じるかを、本研究を通して明確にしたい。

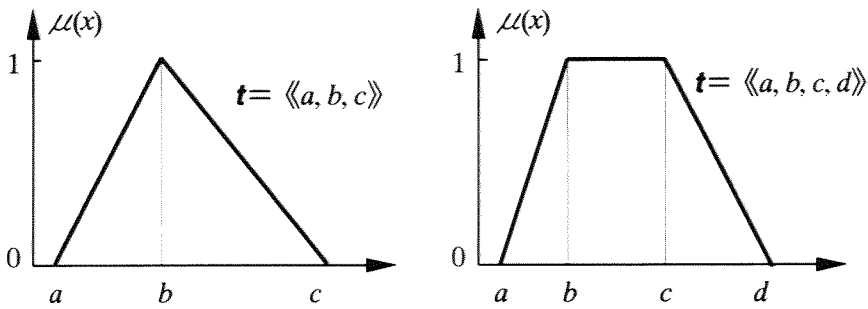


図 1. 三角型ファジィ数と台形型ファジィ数

4. ファジィ数とファジィ数演算

4.1 三角型ファジィ数と台形型ファジィ数

本論文では、図 1 に示す三角型ファジィ数と台形型ファジィ数で不確かな作業時間 t を表わす。 t のメンバーシップ関数を $\mu(x)$ とし、正規性（高さが 1）を満足すると仮定する。

このとき、三角型ファジィ数 t を a, b, c の 3 つのパラメータを用いて

$$t = \langle\langle a, b, c \rangle\rangle \tag{5}$$

のように表現する。

同様に、台形型ファジィ数 t を a, b, c, d の 4 つのパラメータを用いて

$$t = \langle\langle a, b, c, d \rangle\rangle \tag{6}$$

のように表現する。

ファジィ数で不確かな作業時間を表わすとき、 t のメンバーシップ関数 $\mu(x)$ は、作業時間が x である可能性と解釈することができる。三角型ファジィ数の場合、従来の 3 点見積り法と類似し、 $t = \langle\langle a, b, c \rangle\rangle$ の a, b, c がそれぞれ不確かな作業時間の楽観値、最頻値と悲観値と見なせる。これは最頻値が 1 つの特定の時間値で表現できる場合に適合する。さらに、台形型ファジィ数 $t = \langle\langle a, b, c, d \rangle\rangle$ は作業時間の最頻値が 1 つの特定の時間値でなく、1 つの時間区間である場合に適合する。つまり、正常な場合、作業時間が区間 (b, c) の範囲にあり、かつ可能性の度合いがほぼ等しい。また、なにか予測不可能または制御不可能な要因で、作業時間が a まで速くなったり d まで遅くなる可能性もあるがその度合いが単調に下がる。台形型ファジィ作業時間の例としては、ある製品の製造における自然乾燥や自然冷却時間、計測機器の最終テストとチューニング作業時間、ソフトウェア開発におけるプログラミングとデバッグ作業時間などが考えられる。このように、ファジィ数による不確かな作業時間の表現は確率分布などの仮定を必要とせず、かつ三角型か台形型ファジィ数を適切に選ぶことによってより現実に近い表現が得られる。

4.2 ファジィ数の演算

ファジィ数の加法演算については、下記のことが知られている。

(1) 2つの三角型ファジィ数 t_1 と t_2 の和は三角型ファジィ数 t_3 を与える。そして、 $t_1 = \langle a_1, b_1, c_1 \rangle$, $t_2 = \langle a_2, b_2, c_2 \rangle$ ならば、

$$t_3 = t_1 + t_2 = \langle a_1 + a_2, b_1 + b_2, c_1 + c_2 \rangle \quad (7)$$

となる。

(2) 2つの台形型ファジィ数 t_1 と t_2 の和は台形型ファジィ数 t_3 を与える。そして、 $t_1 = \langle a_1, b_1, c_1, d_1 \rangle$, $t_2 = \langle a_2, b_2, c_2, d_2 \rangle$ ならば、

$$t_3 = t_1 + t_2 = \langle a_1 + a_2, b_1 + b_2, c_1 + c_2, d_1 + d_2 \rangle \quad (8)$$

となる。

4.3 ファジィ数の順序づけ

ファジィ作業時間を持つスケジューリング問題を解決するには、最大のファジィ数またはファジィ数の順序づけ（ランキング）を決める必要がある。ファジィ数の順序づけについては、これまでにいろいろな判定基準が提案されており [10]、その中でよく用いられるのは Lee & Li [11] の重心法である。これは、次の2つの基準 R1, R2 によりファジィ数を順序づけるというものである。

[R1: 一般化期待値 GMV による順序づけ]

式 (9) によりファジィ数 t の一般化期待値 GMV (Generalized Mean Value) を計算し、GMV の値 $m(t)$ でファジィ数の順序づけを決める。

$$m(t) = \frac{\int_s x \mu_t(x) dx}{\int_s \mu_t(x) dx} \quad (9)$$

[R2: 広がりによる順序づけ]

判定基準 R1 が一意な全順序関係を与えない（同じ GMV 値を持つファジィ数が複数ある）場合、R1 によって得られたファジィ数のそれぞれのクラス内において、さらに式 (10) で広がり $s(t)$ を計算し、 $s(t)$ の値から順序づけを決める。

$$s(t) = \sqrt{\frac{\int_s x^2 \mu_t(x) dx}{\int_s \mu_t(x) dx} - [m(t)]^2} \quad (10)$$

5. ファジィスケジューリング問題の最適化解法

5.1 分岐手順

上述した問題は分岐限定法を用いて、最適解法を与えることができる。分岐限定法における分岐

手順としては、深さ優先探索（縦型探索）法を用いる。つまり、分岐した最新ノードの中で目的関数の下界値が一番小さいノードを選択し、試行解（終端ノード）に達するまで連続的に分岐手続きを繰り返す。試行解に達した後、これを暫定解とする。それから暫定解の目的値より小さい下界値を持つノードの中で前に求めた試行解に最も近いノードに戻って、分岐する。もし得られた新しい試行解の目的値がこれまでの暫定解より小さければ、これを新しい暫定解とする。終端ノードではなく、かつ下界値が暫定解の目的値より小さいノードがなくなるまで上述の手順を繰り返していけば、得られた暫定解が最適解となる。

具体的に分岐操作手順を次のとおりに提案する。

【分岐操作手順】

[ステップ 1] 分岐レベル $r=0$ として、ルートノード $s=0$ を生成する。

[ステップ 2] 任意のノード s では、 s の分岐レベルを r とする。もし $r=n$ ならば、ステップ 4 へ。そうでなければ、ステップ 3 へ進む。

[ステップ 3] ノード s では、ジョブ J_{r+1} の加工機械をそれぞれ M_1, M_2, \dots, M_m として、 m 個のノードを新たに生成し、それぞれノード s_1, s_2, \dots, s_m とする（図 2）。そして、これらのノードの分岐レベルを $r+1$ 、親ノードをノード s とする。つまり、ノード s_j では、

$$x_{r+1,k} = \begin{cases} 1; & k=j \\ 0; & k \neq j \end{cases}; \quad k=1, 2, \dots, m; \quad j=1, 2, \dots, m \quad (11)$$

とする。

[ステップ 4] 分岐操作を終了する。

5.2 下界値の計算方法

分岐レベル r の任意ノード s では、すでに割り当てられた（分岐済の）ジョブの集合を $X_s = \{J_1, J_2, \dots, J_r\}$ 、未分岐ジョブの集合を $Y_s = \{J_{r+1}, J_{r+2}, \dots, J_n\}$ とする。また、ノード s に対応する部分解を $\{x_{ij}^{(X_s)}; i=1, 2, \dots, r, j=1, 2, \dots, m\}$ と記すれば、ノード s での下界値 LB_s は次式 (12) により計算して得ることができる。

$$LB_s = \text{Fuzzy Max} \{C_j^{(X_s)} + C_j^{(Y_s)}; j=1, 2, \dots, m\} \quad (12)$$

ただし、 $C_j^{(X_s)} = \sum_{i \in X_s} x_{ij}^{(X_s)}$ $t_{ij} = \sum_{i=1}^r x_{ij}^{(X_s)} t_{ij}$ (13)

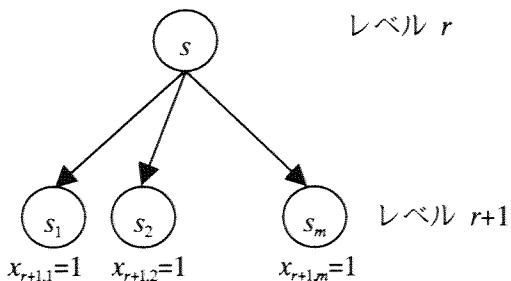


図 2. ノード s での分岐操作

董：ファジィ作業時間をもつ非一様型並列機械スケジューリング問題の最適化解法

$$C_j^{(Y_s)} = \left(+ \right)_{i \in Y_s} x_{ij}^{(Y_s)} ts_i = \left(+ \right)_{i=r+1}^n x_{ij}^{(Y_s)} ts_i \quad (14)$$

ts_i は次節の式 (16) により求められるジョブ J_i の最短作業時間である。 $x_{ij}^{(Y_s)}$ は次の手順により決められた部分解である。

[ステップ 1] 各加工機械での全作業加工終了時刻を Ce_j , Ce_j の初期値として $Ce_j = C_j^{(X_s)}$; $j=1, 2, \dots, m$ とする。

[ステップ 2] $i=r+1$ と設定する。

[ステップ 3] $Ce_q = \text{Fuzzy Min} \{Ce_j; j=1, 2, \dots, m\}$ を満たす q を求め、

$$x_{ik}^{(Y_s)} = \begin{cases} 1; & k=q \\ 0; & k \neq q \end{cases}; k=1, 2, \dots, m \quad (15)$$

と設定して、 $Ce_q = Ce_q + ts_i$ とする。

[ステップ 4] $i=n$ になれば、終了する。そうでなければ、 $i=i+1$ としてステップ 3 へ戻る。

5.3 最適化解法

上述に基づいて、総所要時間を最小にするファジィスケジューリングの最適化アルゴリズムを次のように提案する [12]。

【提案解法】

[ステップ 1] 与えられた各ジョブの最短作業時間を次式 (16) により求める。

$$ts_i = \text{Fuzzy Min} \{t_{ij}; j=1, 2, \dots, m\}, i=1, 2, \dots, n \quad (16)$$

ts_i の大きいジョブから与えられた n 個ジョブを降順で並べて、ジョブ番号 i を付け直す。ただし、ファジィ最小化 Fuzzy Min の計算は上述した基準 R1, R2 を用いて行われる。

[ステップ 2] 分岐レベル $r=0$, $X_0 = \emptyset$ (空集合), $Y_0 = \{J_1, J_2, \dots, J_n\}$, 分岐対象ノード $s^*=0$, 暫定解の下界値 $f^* = \infty$ と設定する。

[ステップ 3] 対象ノード s^* の分岐レベルが r であるため、集合 Y_0 に属するジョブ J_{r+1} に対して、5.1 節の分岐手順に従って分岐操作を行う。新規に生成された m 個のノードにノード番号をつけ、分岐レベル $r=r+1$ とする。それから、5.2 節の計算方法を用いて各ノードでの目的関数の下界値を計算する。また、ジョブ J_{r+1} を Y_0 から取り除いて、 X_0 に追加する。

[ステップ 4] 新規ノードの中で、最小の下界値をもつノード s^* を見つけ、最小下界値を $LB(s^*)$ とする。このノード s^* を新たな分岐対象ノード s^* に選んで、マークをつける。

[ステップ 5] $Y_0 = \emptyset$ (終端ノード) になれば、ステップ 6 へ。 $Y_0 \neq \emptyset$ かつ $LB(s^*) \leq f^*$ ならば、ステップ 3 へ。 $Y_0 \neq \emptyset$ かつ $LB(s^*) \geq f^*$ ならば、ステップ 7 へ。

[ステップ 6] $LB(s^*) < f^*$ ならば、 s^* を新たな暫定解とし、 $f^* = LB(s^*)$ と置き、ステップ 7 へ。 $LB(s^*) \geq f^*$ ならば、そのままステップ 7 へ。

[ステップ 7] すでに生成されたノードの中で、 f^* より小さい下界値を持ちかつマークをつけられていない、ノード s^* に最も近いノードを新たな分岐対象ノード s^* とする。分岐レベル r をこ

の新しい分岐対象ノードの分岐レベル数, 分岐済ジョブの集合 $X_0 = \{J_1, J_2, \dots, J_r\}$, 未分岐ジョブの集合 $Y_0 = \{J_{r+1}, J_{r+2}, \dots, J_n\}$ と設定する。それからステップ 3 へ戻る。もしこのような分岐対象ノードを新たに見つけなければ, ステップ 8 へ。

[ステップ 8] 現時点の暫定解が最適解であり, 終了する。

6. 計算例による考察

6.1 計算例の生成

(1) 三角型ファジィ作業時間の生成

三角型ファジィ作業時間 $t_{ij} = \langle a_{ij}, b_{ij}, c_{ij} \rangle$ は次のように生成される。

b_{ij} を $(1, 100)$ の一様乱数で与え, それから次式 (17) により a_{ij} と c_{ij} を決める。

$$a_{ij} = b_{ij}(1 - \alpha \cdot rnd) \qquad c_{ij} = b_{ij}(1 + \alpha \cdot rnd) \qquad (17)$$

ただし, $0 < \alpha < 1$ は与えられた定数で, rnd は $(0, 1)$ の一様乱数である。

また, 本研究で使用する一様乱数は ANSI 規格準拠の Turbo C++ ライブラリ関数を用いて生成された擬似乱数である。

(2) 台形型ファジィ作業時間の生成

台形型ファジィ作業時間 $t_{ij} = \langle a_{ij}, b_{ij}, c_{ij}, d_{ij} \rangle$ は次のように生成される。

b_{ij} を $(1, 100)$ の一様乱数で与え, それから次式 (18) (19) により a_{ij}, c_{ij}, d_{ij} を決める。

$$a_{ij} = b_{ij}(1 - \beta \cdot rnd) \qquad c_{ij} = b_{ij}(1 + \gamma \cdot rnd) \qquad (18)$$

$$d_{ij} = c_{ij}(1 + \delta \cdot rnd) \qquad (19)$$

ただし, $0 < \beta, \gamma, \delta < 1$ は与えられた定数で, rnd は $(0, 1)$ の一様乱数である。

6.2 4 ジョブ 2 機械の例

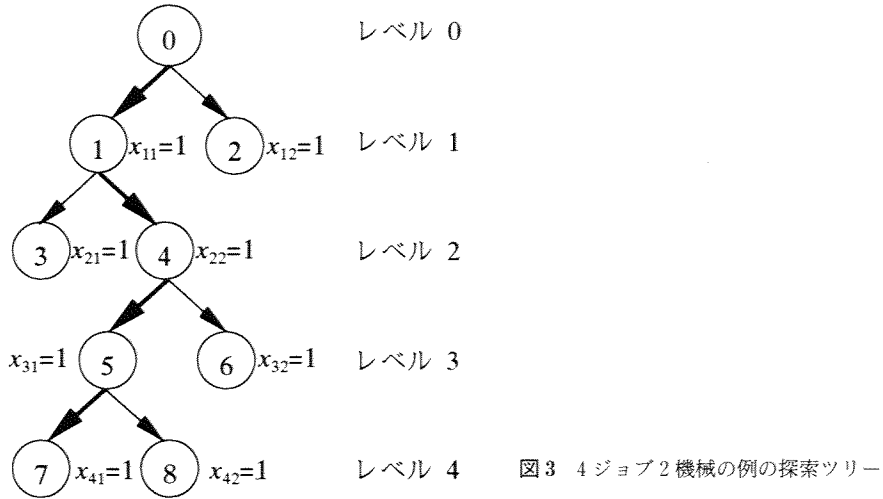
上述のアルゴリズムを, 表 1 に示す三角型ファジィ作業時間をもつ 4 ジョブ 2 機械の計算例に適用した。

(1) 分岐操作

レベル $r=0$ のルートノード 0 では, ジョブ J_1 の加工機械をそれぞれ M_1, M_2 として, 2 個のノード 1, 2 を生成した。また, ノード 1, 2 における下界値を計算し, $LB_1 = \langle 33, 40, 55 \rangle$, $LB_2 = \langle 88, 95,$

表 1. 4 ジョブ 2 機械の計算例

ジョブ J_i	機械 1 での作業時間 t_{i1}	機械 2 での作業時間 t_{i2}
1	$\langle 33, 40, 55 \rangle$	$\langle 88, 95, 100 \rangle$
2	$\langle 21, 27, 31 \rangle$	$\langle 32, 37, 49 \rangle$
3	$\langle 3, 6, 8 \rangle$	$\langle 36, 56, 73 \rangle$
4	$\langle 2, 3, 3 \rangle$	$\langle 12, 17, 20 \rangle$



100)となる。さらに最小の下界値をもつノード1を選んで、分岐操作を行い、ジョブ J_2 の加工機械をそれぞれ M_1, M_2 とすることによって、レベル $r=2$ の2個のノード3,4を新規に生成した。この操作を繰返して、図3に示す探索ツリーが得られた。

(2) 下界値の計算：

図3に示したノード4における下界値の計算を例として説明する。ノード4においては、分岐済のジョブ集合 $X_4=\{J_1, J_2\}$ 、未分岐のジョブ集合 $Y_4=\{J_3, J_4\}$ となる。 J_1 の加工機械が M_1 、 J_2 の加工機械が M_2 なので、

$$C_1^{(X_4)} = \left(\sum_{i \in X_4} x_{i1}^{(X_4)} t_{i1} \right) = \binom{2}{+} x_{11}^{(X_4)} t_{11} = 1 \times t_{11} + 0 \times t_{21} = \langle 33, 40, 55 \rangle$$

$$C_2^{(X_4)} = \left(\sum_{i \in X_4} x_{i2}^{(X_4)} t_{i2} \right) = \binom{2}{+} x_{12}^{(X_4)} t_{12} = 0 \times t_{12} + 1 \times t_{22} = \langle 32, 37, 49 \rangle$$

また、次の手順に基づき、 $x_{ij}^{(Y_4)}$ を決める。

[ステップ1] 各加工機械での全作業加工終了時刻を Ce_j 、 Ce_j の初期値として $Ce_j = C_j^{(X_4)}$ ； $j=1,2$ とすれば、 $Ce_1 = \langle 33, 40, 55 \rangle$ 、 $Ce_2 = \langle 32, 37, 49 \rangle$ となる。また、 $r=2$ である。

[ステップ2] $i=r+1=3$ とする。

[ステップ3] $Ce_q = \text{Fuzzy Min}\{Ce_1, Ce_2\} = \{\langle 33, 40, 55 \rangle, \langle 32, 37, 49 \rangle\}$ を満たす q を求め、 $Ce_q = Ce_2 = \langle 32, 37, 49 \rangle$ 、 $q=2$ となるから、 $x_{31}^{(Y_4)}=0$ 、 $x_{32}^{(Y_4)}=1$ と設定して、 $Ce_2 = Ce_2 + ts_3 = \langle 32, 37, 49 \rangle + \langle 3, 6, 8 \rangle = \langle 35, 43, 57 \rangle$ とする。

[ステップ4] $i=i+1=4$ としてステップ3へ戻る。そこで、上述と同じに、 $Ce_q = \text{Fuzzy Min}\{Ce_1, Ce_2\} = \{\langle 33, 40, 55 \rangle, \langle 35, 43, 57 \rangle\}$ を満たす q を求め、 $Ce_q = Ce_1 = \langle 33, 40, 55 \rangle$ 、 $q=1$ となるから、 $x_{41}^{(Y_4)}=1$ 、 $x_{42}^{(Y_4)}=0$ と設定して、 $Ce_1 = Ce_1 + ts_4 = \langle 33, 40, 55 \rangle + \langle 2, 3, 3 \rangle = \langle 35, 43, 58 \rangle$ とする。

ここまで決められた $x_{ij}^{(Y_4)}$ に基づき、

表 2. 各ノードでの下界値 ($n=4, m=2$)

ノード s	親ノード	$x_{ij} \neq 0$	下界値 LB_s
1	0	$x_{11}=1$	$\langle 33, 40, 55 \rangle$
2	0	$x_{12}=1$	$\langle 88, 95, 100 \rangle$
3	1	$x_{21}=1$	$\langle 54, 67, 86 \rangle$
4	1	$x_{22}=1$	$\langle 35, 43, 58 \rangle$
5	4	$x_{31}=1$	$\langle 36, 46, 63 \rangle$
6	4	$x_{32}=1$	$\langle 68, 93, 122 \rangle$
7	5	$x_{41}=1$	$\langle 38, 49, 66 \rangle$
8	5	$x_{42}=1$	$\langle 44, 54, 69 \rangle$

$$C_1^{(Y_4)} = \left(\sum_{i \in Y_4} \right) x_{i1}^{(Y_4)} t_{i1} = \left(\sum_{i=3}^4 \right) x_{i1}^{(Y_4)} t_{i1} = 0 \times t_{31} + 1 \times t_{41} = \langle 2, 3, 3 \rangle$$

$$C_2^{(Y_4)} = \left(\sum_{i \in Y_4} \right) x_{i2}^{(Y_4)} t_{i2} = \left(\sum_{i=3}^4 \right) x_{i2}^{(Y_4)} t_{i2} = 1 \times t_{32} + 0 \times t_{42} = \langle 3, 6, 8 \rangle$$

となる。さらに、式 (12) により、

$$\begin{aligned} LB_4 &= \text{Fuzzy Max}\{C_1^{(X_4)} + C_1^{(Y_4)}, C_2^{(X_4)} + C_2^{(Y_4)}\} \\ &= \text{Fuzzy Max}\{\langle 33, 40, 55 \rangle + \langle 2, 3, 3 \rangle, \langle 32, 37, 49 \rangle + \langle 3, 6, 8 \rangle\} \\ &= \text{Fuzzy Max}\{\langle 35, 43, 58 \rangle, \langle 35, 43, 57 \rangle\} \\ &= \langle 35, 43, 58 \rangle \end{aligned}$$

となる。

最後に、図 3 のような探索ツリーを通じて、 $f^* = \langle 38, 49, 66 \rangle$ の最適解 (図 3 において太い線で最適解を示す) が得られた。また、各ノードでの下界値を表 2 に示す。

6.3 多数計算例の解と考察

6.3.1 三角型ファジィ作業時間

$\alpha=0.4$ として、式 (17) により三角型ファジィ作業時間を生成し、($m=2, 3; n=20 \sim 100$) の計算例を 50 問作成する。提案したアルゴリズムを適用し、各問題の最適解を求めた。その結果を表 3 に示す。ただし、表 3 に示した解数は、生成するノード数の上限を 4,100,000 とした場合、最適解が得られた問題の数であり、問題数は、10 問の解を求めたとき、提案したアルゴリズムを適用した問題の総数である。計算時間は、Turbo C++ 5.0J 言語で提案解法のプログラムを組み、FUJITSU IA ワークステーション FMV-PRO 6300T7A (Pentium II 300 MHz×2 CPU, Memory 128 MB) 上で実行した場合、最適解が得られた各問題の計算時間であり、1 秒未満の場合 0 と表わす。また、ノード数は最適解が得られた各問題を解くために生成したノード数であり、カット率は全列挙法を適用する場合のノード数に対する比率であり、次式 (20) (21) により計算される。

$$\text{カット率} = \frac{\text{提案解法を適用する場合生成したノード数}}{\text{全列挙法のノード数}} \quad (20)$$

表 3(a). 三角型ファジィ作業時間の例 ($m=2$)

ジョブ数 n	問題数	解数	計算時間 (秒)		ノード数 (カット率)	
			最大	平均	最大	平均
20	10	10	0	0	2,552 (1.21E-03)	760 (3.60E-04)
40	10	10	1	0	439,308 (1.79E-08)	9,638 (4.38E-09)
60	10	10	160	17	4,003,190 (1.74E-12)	426,730 (1.85E-13)
80	10	10	86	26	2,854,942 (1.18E-18)	872,153 (3.61E-19)
100	11	10	47	6	1,342,232 (5.30E-25)	175,884 (6.94E-26)

表 3(b). 三角型ファジィ作業時間の例 ($m=3$)

ジョブ数 n	問題数	解数	計算時間 (秒)		ノード数 (カット率)	
			最大	平均	最大	平均
20	10	10	0	0	8,259 (3.94E-03)	4,699 (8.98E-07)
40	10	10	29	7	886,902 (4.87E-14)	290,854 (1.60E-14)
60	19	10	98	32	2,824,722 (4.44E-23)	854,505 (1.34E-23)
80	29	10	122	61	2,922,024 (1.32E-32)	1,309,099 (5.90E-33)
100	37	10	281	112	3,484,020 (4.51E-42)	1,652,972 (2.14E-42)

$$\text{全列挙法のノード数} = \frac{m}{m-1}(m^n - 1) \quad (21)$$

表 3 から、次のことが分かる。

- (1) 提案解法を適用して、3 機械 100 ジョブの大規模問題を含めて、多数の計算例の最適解が得られた。このなかでは、解の評価にあたりファジィ数の順序づけ基準 R1 と R2 をともに考慮して、最適な総所要時間も通常の定数ではなく、ファジィ数で与えることができた。このため、提案解法を用いて、他の解法の効果を検証するのに必要な最適解を提供することができる。
- (2) 最適解が得られた場合、提案解法の計算時間は非常に短い。100 ジョブの問題でも最大で 281 秒であり、実用的に容認できる範囲にあると考えられる。
- (3) 分岐限定法のキーポイントは、いかに無駄な探索操作を無くし、効率よく最適解を見つけることである。つまり、分岐ノード数を最大限で抑えることである。ここでは、提案解法を適用する場合生成したノード数を全列挙法のノード数と比較し、式 (20) によりカット率を計算した。このカット率が低いほど、探索効率が高くなる。表 3 に示したように、ジョブ数が増えるにつれて、カット率は 10^{-3} から 10^{-42} まで低下し、提案解法の探索効率は非常に高いことがわかる。
- (4) 探索効率を上げるために、より最適値に近い下界値を求めて、これをもって無駄な探索操作を無くすことが必要である。しかし、下界値の計算も計算時間やメモリ容量を必要とするので、下界値の計算方法は複雑過ぎれば、マイナスな効果をもたらすことも有り得る。野村ら

(4) から提案されたラグランジュ緩和問題の最適解に基づいた下界値の計算方法を適用しようとする、ラグランジュ緩和問題を解くために提案解法より 2~3 倍ほど多くのメモリ容量を必要とするので、メモリの容量オーバーが起こればプログラムは実行できなかつたり、ノード数の上限を低く抑えざるを得なくてほとんどの問題の最適解を得られなかつたりしたことが確認された。このため、ここで提案した下界値の計算方法はシンプルなものでありながら、必要なメモリ容量が少なく済むことから、CPU の処理能力が飛躍的に向上している現在の時代において、有効な方法であると考えられる。

- (5) 通常のスケジューリング問題に比べて、作業時間がファジィ数であるファジィスケジューリング問題では、三角型ファジィ数の場合 a, b, c の 3 つのパラメータを用いて作業時間 $t = \langle a, b, c \rangle$ 、台形型ファジィ数の場合 a, b, c, d の 4 つのパラメータを用いて作業時間 $t = \langle \langle a, b, c, d \rangle \rangle$ を表現する。また、下界値もファジィ数であるため、必要なメモリ容量は非常に大きい。表 3 に示したように、生成するノード数の上限を制限することにより、ジョブ数の多い一部分の問題については最適解を得ることができなかった。メモリ容量を使用可能な量以内に抑えることは、ファジィスケジューリング解法を適用するにあたり特に大切なことであり、注意を払う必要がある。

6.3.2 台形型ファジィ作業時間

$\beta=0.4, \gamma=0.4, \delta=0.4$ として、式 (18) (19) により台形型ファジィ作業時間を生成し、($m=2, 3; n=20 \sim 100$) の計算例を 50 問作成する。提案したアルゴリズムを適用し、各問題の最適解を求めた。その結果を表 4 に示す。ただし、表 4 に示した解数は、生成するノード数の上限を 3,800,000 とした場合、最適解が得られた問題の数であり、問題数は、10 問の解を求めたとき、提案したアルゴリズム

表 4(a). 台形型ファジィ作業時間の例 ($m=2$)

ジョブ数 n	問題数	解数	計算時間 (秒)		ノード数 (カット率)	
			最大	平均	最大	平均
20	10	10	0	0	966 (4.61E-04)	557 (2.66E-04)
40	10	10	2	0	46,706 (2.12E-08)	7,482 (3.40E-09)
60	10	10	8	2	160,312 (6.95E-14)	36,986 (1.60E-14)
80	10	10	51	11	712,354 (2.95E-19)	151,344 (6.26E-20)
100	14	10	246	42	2,738,500 (1.08E-24)	464,329 (1.83E-25)

表 4(b). 台形型ファジィ作業時間の例 ($m=3$)

ジョブ数 n	問題数	解数	計算時間 (秒)		ノード数 (カット率)	
			最大	平均	最大	平均
20	10	10	1	0	6,744 (1.29E-06)	32,529 (4.84E-07)
40	10	10	27	7	458,394 (2.52E-14)	133,779 (7.34E-15)
60	14	10	327	67	3,252,477 (5.11E-23)	739,889 (1.16E-23)
80	18	10	324	96	2,230,799 (1.46E-32)	894,211 (4.03E-33)
100	16	10	596	185	3,634,926 (4.71E-42)	1,472,921 (1.91E-42)

ムを適用した問題の総数である。同様に、計算時間は、Turbo C++ 5.0J 言語で提案解法のプログラムを組み、FUJITSU IA ワークステーション FMV-PRO 6300T7A (Pentium II 300 MHz×2 CPU, Memory 128 MB) 上で実行した場合、最適解が得られた各問題の計算時間であり、1秒未満の場合 0 と表わす。

表 4 から、次のことが分かる。

- (1) 三角型ファジィ作業時間の場合と同様に、作業時間が台形型ファジィ数の場合も、提案解法を用いて、3 機械 100 ジョブの大規模問題を含めて、多数の計算例の最適解が得られた。
- (2) 最適解が得られた場合、提案解法の計算時間は非常に短い。100 ジョブの問題でも最大で 596 秒であり、実用的に容認できる範囲にあると考えられる。
- (3) ジョブ数が増えるにつれて、カット率は 10^{-4} から 10^{-42} まで低下し、提案解法の探索効率は非常に高い。
- (4) 三角型ファジィ作業時間に比べて、台形型ファジィ作業時間を表わすのに必要なパラメータが 1 つ多く、より多くのメモリ容量が必要となる。このため、生成するノードの上限を三角型ファジィ作業時間の場合よりもっと低く抑えることが必要である。

7. おわりに

本論文では、ファジィ作業時間をもつ非一様型並列機械スケジューリング問題を検討し、分岐限定法に基づく最適解法を提案したうえ、数値計算例によりその効果を検証した。得られた主な結果は次のとおりである。

- (1) ファジィ作業時間を考慮した場合の分岐限定法を適用するのに必要な下界値の計算方法を新たに提案した。
- (2) ファジィ数の順序づけ基準 R1 と R2 をともに考慮して、解の評価を行い、最適な総所要時間もファジィ数で与えるファジィスケジューリング最適化解法を提案した。
- (3) 提案解法を用いて、3 機械 100 ジョブの大規模問題を含めて、多数の計算例の最適解が得られた。これにより他の解法の効果を検証するのに必要な最適解を提供することができる。
- (4) 最適解が得られた場合、提案解法の計算時間は非常に短い。100 ジョブの問題でも実的に容認できる計算時間で最適解を与えることができる。
- (5) 全列挙法に比べて、提案解法を適用する場合生成したノードの数は、 $\frac{1}{10^3} \sim \frac{1}{10^{42}}$ まで少なく抑えることができ、提案解法の探索効率は非常に高い。
- (6) 通常のスケジューリング問題に比べて、作業時間がファジィ数である場合、必要なメモリ容量は非常に大きい。メモリ容量を使用可能な量以内に抑えることは、ファジィスケジューリング解法を適用するにあたり特に大切であり、注意を払う必要がある。

参考文献

- [1] Ibarra, O.H. & Kim, C.E.: "Heuristic Algorithms for Scheduling Independent Tasks on Nonidentical Processors", *Journal of The Association of Computing Machinery*, Vol. 24, No. 2, pp. 721~736 (1977).
- [2] Davis, E. & Jaffe, J.M.: "Algorithms for Scheduling Tasks on Unrelated Processors", *Journal of The Association of Computing Machinery*, Vol. 28, No. 4, pp. 721~736 (1981).
- [3] Lenstra, J.K., Shmoys, D.B. & Tardos, E.: "Approximation Algorithms for Scheduling Unrelated Parallel Machines", *Mathematical Programming*, Vol. 46, pp. 259-271 (1990).
- [4] 野村弘光, 羽田隆男: "分岐限定法による非一様型並列機械スケジューリング問題の解法", *日本経営工学会誌*, Vol. 45, No. 2, pp. 155-161 (1994).
- [5] McCahon, C.S. & Lee, E.S.: "Fuzzy Job Sequencing for A Flow Shop", *European Journal of Operations Research*, Vol. 62, pp. 294-301 (1992).
- [6] Tsujimura, Y., Park, S.H., Chang, I.S. & Gen, M.: "An Effective Method for Solving Flow Shop Scheduling Problems with Fuzzy Processing Times", *Computers and Industrial Engineering*, Vol. 25, Nos 1-4, pp. 239-242 (1993).
- [7] Tsujimura, Y., Gen, M. & Kubota, E.: "Solving Job-shop Scheduling Problem with Fuzzy Processing Time Using Genetic Algorithm", *日本ファジィ学会誌*, Vol. 7, No. 5, pp. 1073-1083 (1995).
- [8] 董彦文, 中村 壘, 北岡正敏, 太田 宏: "ファジィ作業時間を持つスケジューリング問題の簡便解法", *日本経営工学会論文誌*, Vol. 48, No. 1, pp 17-24 (1997).
- [9] Dong, Y., Ohta, H., Kitaoka, M.: "An Effective Method to Solve A Practical Fuzzy Scheduling Problem", *Proceedings of The 14th International Conference on Production Research (ICPR)*, pp. 446-449, Osaka, Japan, Aug. 4-8 (1997).
- [10] Bortolan, V. & Degani, R.: "A Review of Some Methods for Ranking Fuzzy Numbers", *Fuzzy Sets and Systems*, Vol. 15, pp. 1-15 (1985).
- [11] Lee, E.S. & Li, R.J.: "Comparison of Fuzzy Numbers Based on The Probability Measure of Fuzzy Events", *Computers and Mathematics with Applications*, Vol. 15, pp. 887-896 (1988).
- [12] 董彦文, 北岡正敏: "複数台機械を持つ 2 工程フローショップの総所要時間最小化スケジューリング問題に関する研究", *日本経営工学会誌*, Vol. 43, No. 5, pp. 363-370 (1992).