リエントラントフローショップ スケジューリング問題に関する研究 (第2報)*

発見的なスケジューリング手法の提案

樋野 励** 上川 大輔*** 森川 正憲† 鈴木 教和** 社本 英二**

Study on Re-entrant Flowshop Scheduling Problem(2nd report)*
Proposal of Heuristic Scheduling Method

Rei HINO, Daisuke UEKAWA, Masanori MORIKAWA, Norikazu SUZUKI and Eiji SHAMOTO

Two new heuristic procedures for re-entrant flow shop scheduling problem are proposed to plan a feasible production schedule in industrial manufacturing factories. Each job is assigned to one of idle periods within a limited time on a cyclic Gantt chart. One heuristic procedure is for an assignment of a target job on the machine while an adequate the idle time is remained for undecided jobs. Another heuristic procedure is based on solutions for two classical optimization problems, or a job assignment problem and a knapsack problem. The starting time of the target job is optimally determined at each step in the scheduling. The results indicate that the proposed procedures give a good schedule in comparison with the optimal schedule derived theoretically. The Calculation times are also evaluated and verified a validity of the use in actual manufacturing factories.

Key words : re-entrant flowshop, scheduling, heuristic procedure, feasible solution

1. 緒 言

リエントラントフローショップスケジューリング¹⁾²⁾は、1つの製品の生産過程において、同一の生産装置を繰り返し使用する処理工程が必要になる生産計画に対して重要な役割を果たす。この生産スケジューリングが望まれる代表的な製品として半導体³⁾があげられる。半導体の製造過程では、回路形成に対して同一の処理が繰り返し行われ、また装置が高額であることが、リエントラントな処理工程を避けることができない理由のひとつに考えられる。しかし、最近では旋削加工とフライス加工のどちらも可能な複合加工機のように、1台でさまざまな加工を行うことができる工作機械が開発され、多くの生産現場に導入されつつある。このような多機能の加工機が利用できる場合、処理工程に類似性が無くとも、工程計画の段階でリエントラント工程を含めることが可能である。その場合、比較的少量の生産量の商品に対しても設備投資に要する費用を抑えることができるため、大量生産と同程度の費用効果をもたらす可能性がある。

本報では、現実的な規模の生産計画に対するスケジュール案を得るために、発見的なスケジューリング手法を提案する。リエントラントフローショップスケジューリング問題に対しては、S.C.Gravesらがサイクリックガントチャートを用いた発見的な手法を提案している¹⁾。本報では、ボトルネック工程を基準としたスケジューリング手順を提案することにより、フロータイムが短いスケジュールを得ることを目的とする。まず、S.C.Gravesらと類似の方法として、仕事を割り付けた後の機械の遊休時間が断片化されないような割り付け規則を提案する。その際、処理工程の順方向の割り付けに加え、一部の

仕事については逆方向に割り付けることにより、より良いスケジュールの立案を試みる。また、割当問題とナップサック問題に対する最適化手法を利用した新しいスケジューリング方法の提案を行う。一般にリエントラントフローショップスケジューリングを含むスケジューリング問題は、順序付最適化問題と認識されているが、提案手法は非順序付け問題に対する最適化手法を利用する点で独自性を有する。

2. リエントラントフローショップスケジューリング問題

本研究で対象とするリエントラントフローショップスケジューリング問題を以下のように定義する。なお、単一の製品を大量生産することを前提にする。

- i. 複数の処理工程から成る単一製品を対象
- ii. 個々の処理工程に使用される機械および処理時間は既知
- iii. 異なる処理工程に対し同一の機械による処理を許可
- iv. 各機械が一度に処理することができる処理工程はひとつ

リエントラントフローショップスケジューリング問題に対する基本的な特徴は前報³⁾に述べている通りである。本論文では現実的な規模のスケジューリングを目的とし、数百工程からなる製品のスケジューリングを行う。本論文で使用する記号は基本的に前報³⁾に準ずるが、必要に応じて本文中にて適宜説明する。また、前報³⁾と同様にスケジューリングの2つの評価値であるサイクルタイムとフロータイムについては、サイクルタイムを単位時間当たりの生産量が最大になるように予め設定しておき、工程間仕掛在庫が最小になるようにフロータイムが短いスケジュールを得ることを目的とする。

3. 発見的なスケジューリング

3.1 サイクルタイム内での仕事の割り付

本報では、300~400工程からなる製品に対するスケジュールの立案を目的とする。そのため、リエントラントフローショップ

* 原稿受付 平成 19 年 11 月 28 日

** 正 会 員 名古屋大学大学院 (名古屋市千種区不老町)

*** 名古屋大学 (現ヤマハ発動機株式会社)

† 東芝セミコンダクター社 (三重県四日市市山之一色町)

Fig.1 Concept to plan schedule with short flow time

Fig.2 Status of assignment of jobs on each machine

プスケジューリング問題を、混合整数計画問題として定式化し数値計画法による解を求めるのではなく、工程の処理順序に従って、一つずつ機械に割り付ける方法を提案する。

本研究で取り扱うリエントラントフローショップスケジューリング問題の前提条件として、サイクルタイムは、単位時間当たりの生産量が最大になるように設定する。すなわち、各機械が処理すべき仕事の処理時間の総和の最大値をサイクルタイムとして定める。このサイクルタイムを与える機械は、生産効率の上限を決定するため、生産工程のボトルネックと認識される。また、この機械により処理される仕事は、すべてボトルネック工程と考えることができる。明らかに、このボトルネック工程は、フロータイムの遅延を許さずには、処理開始時刻を遅くすることはできない。

ところで、フロータイムが短いスケジュールを立てるためには、仕事の工程間での待ち時間を出来るだけ無くす必要がある。そのため、本論文では、ボトルネック工程の一つに着目し、このボトルネック工程を起点として各機械に仕事の割り付けを行うことを考える。図1は、ボトルネックを起点とした仕事の関係を示している。図1に示すように、ボトルネックより上工程は後詰めに、また下工程は前詰めに処理時刻を決定する。この方針が発見的スケジューリングの中核となり、スケジュールの良否を決める要因の一つになる。

リエントラントフローショップスケジューリング問題は、サイクルタイムを1周期とするサイクリックガントチャート上での仕事の処理順序を考えれば十分である。このとき、各機械が処理すべき仕事のうち、最も遅く処理される仕事の処理終了時刻と最も早く処理される仕事の処理開始時刻の差は、このサイクルタイム以下でなければならない。サイクルタイムを、ボトルネックとなる機械の総処理時間に設定することで、全ての機械がこの条件の下で仕事の処理開始時刻の計画が可能であることが保証される。

本論文では、工程順序に従って仕事を一つずつ機械に割り付けるスケジュール方法を提案する。このとき、一度定めた仕事の処理時刻は、他の仕事を割り付ける際に一切変更できないことを仮定する。図2は、直前の処理工程を割り付けた機械 R_2 に対し、機械 R_1 に新たに仕事を割り付けるときの状態を示している。機械 R_1 には、すでに複数の仕事が割り付けられてお

 Fig.3 Classification based on relationship between previous process and idle time from s_i to f_i

り、新たな仕事に加え残りの全ての仕事は、機械の v だけある遊休時間 ($I_1 \sim I_v$) に割り付けなければならない。すなわち、新たな仕事を割り付けることにより、機械の遊休時間がさらに細分化されることがあれば、残りの仕事を割り付けるための十分な時間が確保できなくなることがあるため注意しなければならない。

3.2 発見的手法 1 の提案 (同一遊休時間内の割当)

仕事の処理時刻を定める際、処理開始時刻および処理終了時刻がその機械の遊休時間内になければならない。また、割り付けた後の遊休時間を利用して、その機械により処理される仕事のうち、まだ処理時刻が決められていないすべての仕事を割り付けることを保証しなければならない。この条件を満たすスケジュールの立案方法として、以下の手順を提案する。ただし、この手法は、割り付ける仕事のひとつ前の処理工程がすでに他の機械に割り付けられており、下工程に対して1つずつ仕事を割り付ける場合の手順を示している。

サイクルタイムの決定 各機械が処理しなければならない仕事の処理時間の総和を求める。総処理時間の最大値を、サイクルタイム T として定める。

ボトルネック工程の決定 サイクルタイムを与える機械の処理工程の1つをボトルネック工程として、サイクリックガントチャートに割り付ける。このボトルネック工程の処理開始時刻は $s = 0$ とする。この仕事の次工程を着目する仕事として、手順1の処理を行う。

手順1(第1作業の割付) 着目する仕事が、その機械に割り付ける最初の作業の場合、前工程の処理終了時刻 t をその仕事の処理開始時刻とし、処理を手順7へ移す。一方、すでに他の仕事はその機械に割り付けられている場合、手順2の処理を行う。

手順2(遊休時間の同定) すでに割り付けられている仕事のうち、最も早い処理開始時刻からサイクルタイム T だけ遅い処理時刻を求める。このサイクルタイム内で機械が

仕事の処理を行わない遊休時間を求める。遊休時間が複数ある場合は最も早い時刻に始まる遊休時間の区間を求める。手続きを手順3の処理へ移す。

手順3(割付箇所の分類) 遊休時間と前工程の関係を調べる。遊休時間が始まる時刻より早い時刻に前工程の処理終了時刻が計画されている場合(図3(a)), 手順4へ、遊休時間が終わる時刻より遅い時刻に前工程の処理終了時刻が計画されている場合(図3(b))には、手順5の処理を行う。遊休時間の間に前工程の処理終了時刻が計画されている場合(図3(c)), 手順6の処理を行う。

手順4(最小待ち時間での割付) 前工程が遊休時間が始まる前に処理を終える計画が立てられている場合、着目する仕事の処理開始時刻を、遊休時間の開始時刻に一致させる。手順7の処理を行う。

手順5(遊休時間の変更) 前工程の処理終了時刻が遊休時間が終わる時刻より遅い場合、着目する遊休時間をひとつ後に移し、手続き3の処理を行う。

手順6(残り仕事の処理時間を考慮に入れた割付) 前工程の処理終了時刻に着目する仕事の処理開始時刻を仮に定めた場合に生じる2つの新たな遊休時間を求める。このどちらか一方の遊休時間が、まだ処理時刻が決まっていない残りの仕事の仕事の処理時間総和より大きい場合、着目する仕事の処理開始時刻を前工程の処理終了時刻に定める(図3(c-1)あるいは(c-2))。2つの遊休時間のいずれにも残りの仕事の総処理時間より短い場合、着目する仕事を遊休時間内に後ろ詰めに割り付け、処理開始時刻を定める(図3(c-3))。処理を手順7へ移す。

手順7(終了判定) 着目する仕事が最終工程の場合、スケジューリングを終える。次工程がある場合、次工程に着目する仕事を換え、再び手順1の処理を行う。

ここに記した手順は機械に割り付けられて仕事に対し、その下工程の処理開始時刻を割り付けるための手順である。上工程に対して仕事の処理開始時刻を定める場合には、図4に示すように鏡像の位置関係の仕事を想定して同様の手続きを行えば良い。

本提案は、着目する仕事を割り付ける遊休時間と同一の時間内に、できるだけ残り仕事の全てを割り付けるように処理時刻を定めることを特徴とする。S.C.Gravesらによる手法については詳細な手順について明らかにされていないため推測の域は出ないが、類似の手法¹⁾を用いていると考えられる。ただし、S.C.Gravesらは、製品の第1工程から順に機械に仕事の割り付けており、リエントラントフローショップスケジューリング問題1題につき、1つのスケジュール案しかできない。本手法は任意の仕事を起点として割り付けを行うことができ、複数のスケジュール案により評価できる点に特徴を持つ。

3.3 発見的手法2の提案(非順序付け最適化問題の適用)

着目する仕事に対し、前工程あるいは次工程との待ち時間が最小になるように処理開始時刻を定め、同時に残りの仕事の割り付けを保证するスケジューリング手順について、さらに異なる手順の提案を行う。前報³⁾にて、リエントラントフローショップスケジューリング問題を混合整数計画問題として定式化した場合、同一時刻での複数の処理を禁止するために、処理の先行関係を表す0-1整数変数を用いている。本論文で提案する方法は、このような順序付の最適化問題を解かず、

Fig.4 Classification based on relationship between next process and idle time from s_i to f_i

2つの非順序付け最適化問題を利用する。ひとつは、割付問題(Job assignment problem)であり、もうひとつはナップサック問題(Knapsack problem)である。いずれの最適化問題も、決定変数に0-1整数変数を含むため、最適解を求めることは難しいことに変わりはない。しかし本論文で提案する手法は、各機械に割り当てられた仕事についての最適化問題を解くために、個々の問題規模を抑えることができる。そのため、本論文で扱うような300~400工程からなる製品に対するスケジュール問題に対しても利用が可能である。

提案する手順は以下の通りである。ただし、サイクルタイムおよびボトルネック工程の決定については、前節にて提案した手順と同様である。また、第1作業の割り付けについても同様であるため、ここでは、既に機械に仕事が割り付けられており、遊休時間が同定できる状態について手続きを示す。

手順1(遊休時間 I_i での処理可能性の判定) 着目する仕事を割り付ける機械の遊休時間とその仕事の直前の工程の関係を調べる。ここでは、図5(a)に示すように、仕事を割り付ける機械 R_1 には、 $I_1 \sim I_v$ のように v 区間の遊休時間があり、この内 i 番目の遊休時間 I_i 内に前工程の処理が終わる状態を考える。次に処理を手続き2へ移す。

手順2(遊休時間の順位付け) 着目する仕事はできるだけ前工程の処理終了時刻に近い時刻に処理を始めることが望まれる。そのため、最適な処理開始時刻は、前工程の処理が終わる時刻 t ($s_i \leq t \leq f_i$) であり、次に良い時刻は遊休時間 I_i 内の時刻 $t \leq f_i$ 内に処理を行うことである。この順位は、 $I_i, I_{i+1}, \dots, I_v, I_1, \dots, I_{i-1}$ の順に低くなる。すなわち、遊休時間 I_i 内の時刻 $s_i \leq t$ に処理を行う場合に、最もサイクルタイムが長くなる。ここで、着目する仕事の処理時間と時間 $f_i - t$ の関係により、遊休時間の評価値を定め、その順位付けを行う。ここでの処理は、遊休時間と着目する仕事の処理時間との

関係を判定し、 $f_i - t < p$ ならば手順 3 の $p \leq f_i - t$ ならば手順 4 の処理を行う。ここで、 p は着目する仕事の処理時間である。

手順 3 (割当問題の評価値の決定, type I) 前工程の処理終了時刻と遊休時間 I_i が終わる時間帯に着目する仕事を割り付けることはできないので、遊休時間の順位は、 I_{i+1} が最も高くなる。そこで、遊休時間 I_i の評価値 c_i として以下の値を与える (図 5(b))。

$$c_{i+1} = 1, c_{i+2} = 2, \dots, c_v = v - i,$$

$$c_1 = v - i + 1, c_2 = v - i + 2, \dots, c_i = v$$

この評価値に対して、後述する最適化問題である割当問題を解く。割当問題の結果、遊休時間 I_j が選ばれた場合、その遊休時間 I_j が始まる時刻 b_j を着目する仕事の処理開始時刻とする。次に、手順 6 の処理を行う。

手順 4 (割当問題の評価値の決定, type II) 前工程の処理終了時刻と遊休時間 I_i が終わる時間帯に着目する仕事を割り付ける可能性があるため、遊休時間の順位は、 I_i が最も高くなる。そこで、遊休時間 I_i の評価値 c_i として以下の値を与える (図 5(c))。

$$c_i = 1, c_{i+1} = 2, \dots, c_v = v - i + 1,$$

$$c_1 = v - i + 2, c_2 = v - i + 3, \dots, c_i = v$$

この評価値に対して、後述する最適化問題である割当問題を解く。割当問題の結果求められる遊休時間 I_j が、 $I_j = I_i$ の場合、手順 5 の処理を行う。 $I_j \neq I_i$ の場合、その遊休時間 I_j が始まる時刻 b_j を着目する仕事の処理開始時刻とする。次に、手順 6 の処理を行う。

手順 5 (ナップサック問題の定義) 遊休時間 I_i を $s_i \sim t$ の時間帯 I_i^- と $t \sim f_i$ の時間帯 I_i^+ に分ける (図 5(d))。手順 4 における割当問題に対する最適解から、着目する仕事以外に、遊休時間 I_i に割り当てられる仕事に対して、後述するナップサック問題を解く。ここでは時間帯 I_i^+ に割り当てることができる最大の仕事の組合せを、ナップサック問題を解くことにより求め、選ばれなかった仕事の処理時間総和を求める。遊休時間 I_i の処理開始時刻からこの処理時間総和だけ遅い時刻と、前工程の処理終了時刻の内、遅い方の時刻を着目する仕事の処理開始時刻とする。手順 6 の処理により、終了判定を行う。

手順 6 (終了判定) 着目する仕事が最終工程の場合、スケジューリングを終える。次工程がある場合、次工程に着目する仕事を変え、再び手順 1 の処理を行う。

ここでは前工程に対して下工程の仕事を割り付けるときの手順について記述している。後工程に対して上工程の仕事を割り付ける時には、同一遊休時間内に仕事を割り付ける手法と同様に鏡像の関係にある状態を想定し、本節に示す手順 1 から手順 6 までの手続きを行う。

割当問題 (Job assignment problem) は、複数の仕事の負荷が既知の場合に、与えられた各作業者の能力、処理可能な作業量、および賃金の情報に基づいて、仕事を作業者に割り当てる問題である。ここでは、複数の仕事の集合 $J = \{j_1, j_2, \dots, j_m\}$ を複数の遊休時間の集合 $I = \{I_1, I_2, \dots, I_v\}$ に割り当てる問題を考える。ただし、 j_i ($i = 1, 2, \dots, m$) は対象とする機械により処理される仕事の内、まだその機械に割り付けられていな

Fig.5 Definition of idle time on target resource

い m 個の仕事を意味する。このとき、着目する仕事が、遊休時間の評価値に対して最小になるように最適化を行う。この割当問題は次のように定義することができる。

$$\text{minimize} \quad \sum_{i \in I} c_i x_{i1} \quad (1)$$

subject to

$$\sum_{i \in I} x_{ij} = 1, \quad \forall j \in J \quad (2)$$

$$\sum_{j \in J} p_j x_{ij} \leq I_i, \quad \forall i \in I \quad (3)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i \in I, \forall j \in J$$

まず、式 (1) は割当問題の目的関数を表している。 x_{ij} は、0-1 整数決定変数であり、 i 番目の遊休期間内で j 番目の仕事を処理するとき $x_{ij} = 1$ 、処理しないとき $x_{ij} = 0$ の値をとる。ここでは、着目する仕事を $j = 1$ と定義し、この仕事が割り付けられた遊休時間の評価値 c_i を求めている。一方、式 (2) は、仕事毎に一つの制約式を表す。たとえば着目している仕事 $j = 1$ について書き下せば次のように記述される。

$$x_{11} + x_{21} + \dots + x_{v1} = 1 \quad (4)$$

この式は、仕事 $j = 1$ が v 個ある遊休期間のうち、ただ一つの遊休期間内に処理されることを表している。次に、式 (3) は、遊休期間内に処理される仕事の処理時間に対して制約を与える。同様に書き下せば以下のようになる。

$$x_{i1} p_1 + x_{i2} p_2 + \dots + x_{im} p_m \leq I_i \quad (5)$$

同様にこの式では、 i 番目の遊休期間に処理される仕事の処

理時間の総和は、その遊休期間の時間以下でなければならないことを表している。

ナップサック問題は、与えられた容量を超えない範囲で可能な限り多くの荷物を詰め込む問題である。ここでは、遊休時間 I_i に割り当てる仕事の集合 $J_i = \hat{j}_1, \hat{j}_2, \dots$ に対して t と f_i の時刻により定義される遊休時間 I_i^+ に割り当てることができる仕事を選択する。ここで、仕事の集合 J_i には、着目する仕事 j_1 は含めない。本論文におけるナップサック問題は次のように定義される。

$$\text{maximize} \quad \sum_{j \in J_i} p_j y_j \quad (6)$$

$$\text{subject to} \quad \sum_{j \in J_i} p_j y_j \leq I_i^+ - p_1, \quad \forall i \in J_i \quad (7)$$

$$y_j \in \{0, 1\}, \quad \forall j \in J_i$$

式 (6) は、目的関数を表す。ここで y_j は 0-1 整数変数であり、対象とする仕事を、遊休時間 I_i^+ に含める場合 1 を含めない場合に 0 の値をとる。 p_j は仕事の処理時間であり、この目的関数では割り当てる仕事の処理時間総和を最大にするように最適化を行う。制約式 (7) は、選択した仕事の処理時間総和が I_i^+ から着目する仕事の処理時間を引いた値より小さくなければならないことを表している。着目する仕事 $j = 1$ は、遊休時間 I_i^+ に必ず割り当てることが決まっているためである。

提案する手法は、非順序付け問題である割当問題とナップサック問題を 1 台の機械に割り当てられる仕事について解くことにより、前工程あるいは次工程との間の待ち時間が最小になるように仕事の処理時間を定める。割当問題とナップサック問題は共に、良く知られた最適化問題であり、スケジューリング問題と同様、問題規模の増加によって最適解を得ることが難しくなる。しかし、本研究で提案する手法は、300~400 もの工程数が必要な場合でも、各機械に着目すれば数十工程しかないことが多く、実用的な処理時間でスケジュールの立案ができる可能性がある。

4. 数 値 計 算 例

4.1 ボトルネック工程の選定と割付順序

本論文で提案する手法は、処理しなければならない仕事の処理時間総和が最も長い機械に着目し、その機械により処理される仕事の 1 つをボトルネック工程として選定する。ここでは、ボトルネック工程は次の 2 つの基準により選定する。

ボトルネック工程の選択

- 1) 候補となる仕事のうち、処理時間が最も長い工程をボトルネック工程とする。
- 2) 候補となる仕事を、工程順に並べたとき、中央に位置する工程をボトルネック工程とする。

また、機械に割り付ける順序は以下の 3 通りについて検討を行う。

割付順序

- 1) ボトルネック工程を基準に上工程に向かって 1 つずつ仕事を割り当てる。すべての上工程が割付けられた後は、ボトルネック工程を基準に下工程を割付ける。
- 2) 下工程を先に割付け、その後上工程を割り付ける。

Table 1 An example of re-entrant scheduling problems

Process	1	2	3	4	5	6	7	8
Machine	1	5	4	2	4	5	3	2
Time	3	10	6	12	9	6	9	7
Process	9	10	11	12	13	14	15	16
Machine	3	4	6	1	2	6	3	1
Time	11	13	12	9	9	13	2	4
Process	17	18	19	20	21	22	23	24
Machine	6	5	1	3	6	3	6	5
Time	8	10	3	12	13	3	7	7
Process	25	26	27	28	29	30		
Machine	2	3	2	4	3	1		
Time	11	8	4	12	1	4

Fig.6 Cyclic Gantt chart obtained heuristic scheduling procedure 1 (flow time : 441)

Fig.7 Cyclic Gantt chart obtained heuristic scheduling procedure 2 (flow time : 421)

- 3) 上工程と下工程を交互に 1 つずつ割付ける。

本論文では、2 つの異なるスケジューリング手法を提案している。ボトルネック工程の選択として 2 通り、また割付順序として 3 通りの選択肢があるため、1 題のスケジューリング問題に対して、 $2 \times 2 \times 3 = 12$ 通りのスケジュールを評価することになる。

4.2 スケジューリング結果の評価

提案するスケジューリング手法の妥当性を確かめるために、数値シミュレーションによる評価を行う。検討に用いたりエントラントフローショップスケジューリング問題の 1 例を表 1 に示す。ここでは、6 台の機械 machine1~machine6 により処理される 30 工程の処理を必要とする製品について、使用する機械の番号と処理時間を示している。

図 6 は、同一遊休時間内に仕事を割り付ける発見的手法 1 により得られた 6 通りのスケジュールの内、最良のスケジューリング結果を示している。このスケジュールはボトルネック工程として、機械 machine6 により処理される中央の工程を選

Fig.8 Comparison between best schedule planned by proposed procedures and optimum schedule

Fig.9 Calculation time for heuristic procedures 2

び、このボトルネック工程より下工程を先に割り付けることにより得られている。一方、図7は、2つの最適化問題を適用する発見的手法2を用い、割り付け時点での待ち時間が最小になるように仕事を割り付けることにより得られたスケジュールの内、最良のスケジューリング結果を示している。このスケジュールは、機械 machine6 による処理工程の内、最長の処理時間を必要とする工程をボトルネック工程とし、下工程を先に割り付けることにより得られている。いずれの結果も、第1工程の処理開始時刻が $s_1 = 0$ になるように、処理時刻を修正してある。本スケジューリング問題に対する最適解は、フロータイムが336のスケジュールであることが前報³⁾による検討から明らかになっている。提案手法によるスケジュール結果には、顕著な共通点は認めれないが、わずか12通りのスケジュールを検討するだけで、最適解に近いスケジュールが得られていると言える。なお、105題の問題について評価を行った結果、最良のスケジュールは、同一遊休時間内に割り付ける手法によるものが108案、2つの非順序付け最適化問題を用いる手法によるものが133案であった。ただし、ボトルネック工程および割り付け順序が異なっても同じフロータイムのスケジュールが得られた場合は、重複して積算している。これらの結果から、提案する手法の優劣を議論することはできないと考えている。むしろ、単一のスケジューリング結果によりスケジュール案を立てるのではなく、複数のスケジューリング結果から最良の案を得ることが有効であると考え。

図8は、最適解が求められる規模のスケジューリング問題に対し、提案する2つの発見的手法によるスケジューリング手法により得られる最良のスケジュールを評価した結果を示し

ている。ここでは、機械数毎に工程数を変えて評価している。例えば、6台機械による30工程の製品に対するスケジュールの場合、各機械は等しく5つの仕事を割り当てられていることを意味する。評価を行うことができた範囲内では、最適解に対して多くの場合、1.3倍以下のフロータイムのスケジュールを得ることができている。

図9は、大規模な問題に対する求解時間の変化を示している。同一遊休時間内に割り付ける手法を利用するスケジューリング手法は、3.2節に示した手順から判断できるように、工程数に比例して求解時間が増加する。ここでは、組み合わせ数の観点から、求解時間の指数関数的な増加が予測される2つの非順序付け問題を利用するスケジューリング方法についてのみ評価を行っている。この問題では、使用する機械数と各機械に割り付ける仕事数が等しい場合について評価している。例えば、400工程のスケジューリング問題については、20台の機械を想定し、各機械には等しく20工程の仕事を割り当てている。利用した2つの最適化問題は、求解困難な組合せ問題として知られているため、工程数の増加に伴い、急激に処理時間が増加する傾向が認められる。しかし、400工程を超える問題でも、求解に要する時間は14秒程度 (CPU: Intel(R) Xeon(TM) CPU 3.60 GHz, Memory 1024 Mbytes) であり、目的とした300~400工程程度の製品のスケジューリング手法としては、十分に利用が可能と判断することができる。

5. 結 言

リエントラントフローショップスケジューリング問題に対し、2つの発見的手法に基づくスケジューリング方法の提案を行った。まず、仕事を機械に割り付ける際に、同一の遊休時間内にその機械が処理しなければならない残りの仕事を割り付ける方法を提案した。さらに、非順序付け最適化問題である割付問題とナップザック問題を利用した新しいスケジューリング方法の提案を行った。後者の手法では、仕事を割り付ける時点で、直前の工程あるいは次工程との待ち時間が最短になる時刻に仕事を割り付けることが可能である。

提案する手法は、いずれもボトルネックとなる工程を基準に、上工程あるいは下工程に向かって、1つずつ仕事を機械に割り付けることができる。そのため、1題のスケジュール問題に対して、複数のスケジュール案を立案することができ、得られたスケジュール案の中から最良の結果を採択することが可能である。提案する手法は、数値計算によりその妥当性を確認した。具体的には、提案するスケジューリング手法は、最適解に対し1.3倍程度のフロータイムのスケジュールの立案が可能であることを確認した。また、400工程のスケジューリング問題に対し、14秒以下のスケジュールの立案ができることを確認しており、半導体の製造を始めとする実用的な生産現場での利用が期待できる。

参 考 文 献

- 1) Stephen C. Graves, et.al.: Scheduling of re-entrant flow shops, *Journal of Operations Management*, **3**, 4, (1983), 197.
- 2) K.A. Aldakhilallah, R. Ramesh: Cyclic scheduling heuristics for a re-entrant job shop manufacturing environment, *Int. J. Prod. Res.*, **39**, 12, (2001), 2635.
- 3) R. Hino et al.: Study on Re-entrant Flowshop Scheduling Problem (1st report) Optimization by Mathematical Programming, *J. Jpn. Soc. Precis. Eng.*, **74**, 10, (2008), 1118 (in Japanese).