
NEHアルゴリズムに基づくジョブショップ・スケジューリングのためのヒューリスティック

松 浦 春 樹

アブストラクト

ジョブショップにおける構成的スケジューリング手法は、手軽に使えるなどの利点を有している。しかしながら、2機械の場合のジャクソンルール、広く使える優先番号法を除き、有効なルールは見当たらないのが現状である。本研究では、著者の定義した、加工時間変動に対して頑健であるN-FCFSスケジュール集合内から、総所要時間の意味で良好なスケジュールを構成的に生成するヒューリスティックを提案する。このヒューリスティックでは、第一オペレーションのスケジュールをフローショップ向けのNEHアルゴリズムによって、第二オペレーション以降はFCFSに従って生成する。広く知られたベンチマーク問題およびショップのフロー化度をパラメータとする一般ジョブショップを対象に、その性能を評価した。ベンチマーク問題では、探索的手法によるスケジューリング結果には及ばないことは当然ながら、機械台数に対してジョブ数が多い問題で有効性が増す結果を得た。ショップのフロー化度をパラメータとする一般ジョブショップを対象とする実験では、ショップがフローショップに近づくにしたがって優先番号法よりも圧倒的に優れた結果を得た。

キーワード：ジョブショップ、NEHアルゴリズム、優先番号法、フローショップ、構成的スケジューリング、ヒューリスティック

1. はじめに

本研究の目的は、加工時間の変動に強く、総所要時間の意味で良好な構成的ジョブショップ・スケジューリング手法を提供することにある。ジョブショップとは、ジョブ群が必要とする機械加工順序が全て同一であるフローショップに対して、必要な機械加工順序が個々のジョブにより異なるランダムショップのことを意味する。本研究の特徴は、第一に、生成するスケジュールが、加工時間の環境の変動に強いことを前提とすることにある（図1）。このようなジョブショップ・スケジューリング分野の研究は、実務上極めて重要であると考えられるが、環境が変わらないことを前提として、たとえば総所要

時間を最小化するような最適化スケジューリング手法の開発に多くの関心が集中しており、加工時間の変動のような環境の変化を考慮してスケジュールを作成する方法の研究は、現時点で著者の知る限り、多くない。これらの環境の変化を加味したスケジューリングの研究は、実際上の必要性から脚光を浴びつつあり、[1]によってサーベイされている。

本研究の第二の特徴は、構成的なスケジューリング方法の開発を意図することにある。ジョブショップ・スケジューリングの方法は、スケジュール生成の手順から構成的スケジューリング法と探索的スケジューリング法に大別できる。[7]によれば、構成的スケジューリング法は、簡単なルールによって加工順序を一意に決める

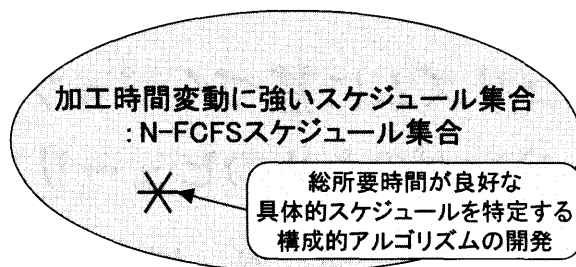


図1 本研究の目的

方法であり、探索的スケジューリング法は、可能なスケジュールをリスト又は列挙し、最適でないものをリストから削除しつつ、最適スケジュールを探索してゆく方法である。両方法の長短であるが、構成的スケジューリング法は、計算が比較的簡単でありスケジューリングのプロセスが直接的であるため手軽さに大きな実務上の有用さがある反面、例外を除いて最適性を探索的スケジューリング法に譲る。構成的スケジューリング法は、探索的スケジューリング法に逐次改良の出発点としての初期スケジュールを提供する役割も担っている[11]。反対に、探索的スケジューリング法は、当初スケジュールからの逐次的改良計算に時間がかかり、スケジューリングのプロセスも逐次的との意味で直接的ではない。しかし、時間をかける探索の結果、構成的スケジューリング法よりも最適性の追求において当然ながら有利である。しかしながら、時々刻々変わるような環境では、環境を固定した最適性の追求の意味が、現実問題として薄れることは確かである。

本研究では、従来のジョブショップ・スケジューリングの方法を概観した後、著者が先に提案した総所要時間が短かつ加工時間変動に頑健であるスケジュールの集合、N-FCFS[18]から良好なスケジュール一つを生成する新たなアルゴリズム (NEH-FCFS) を提案する。繰り返しになるが、このアルゴリズムは、ノンディレイ・スケジュールの部分集合であるスケジュールの新たなタイプN-FCFS集合に属している。このことから、NEH-FCFSアルゴリズムによるスケ

ジュールは、加工時間変動に頑健であることが総体的に保証される。提案したNEH-FCFSアルゴリズムの最適性側面の検証であるが、総所要時間最小の意味での良好度を2つの方法で検証している。第一に、公開されているジョブショップのベンチマーク問題にNEH-FCFSアルゴリズムを適用し、従来発見されている最良解 (必ずしも最適解ではない) とNEH-FCFSアルゴリズムで得られた解を比較している。第二に、NEH-FCFSアルゴリズムの発想の原点から、NEH-FCFSアルゴリズムは、フローショップに近いジョブショップで高性能が期待できる。このことから、ジョブショップのフローショップへの近さを表すパラメータであるフロー化率を導入したジョブショップモデルにおいて、NEH-FCFSアルゴリズムによる総所要時間を、優先番号法によるものと比較している。以上の結果、とくにフローショップに近いジョブショップにおいて、NEH-FCFSアルゴリズムが有効であることを示している。

2. ジョブショップ・スケジューリングの方法

(1) 構成的スケジューリング

構成的アルゴリズムと多項式オーダの時間で解を得ることとは異なった概念であるが、構成的アルゴリズムであれば多項式オーダの時間で解を得ることができるのが自然である。

総所要時間最小の構成的アルゴリズムが発見されているのは、機械台数2かつジョブのオペレーション数が2以下の場合のジャクソンルー

ル[10]、ジョブ数が2の場合[4]などの極めて限られた場合だけである。むしろ、以上を除いたほとんどの場合がNP困難であることが証明されている。これは、ジョブショップ・スケジューリング問題では、例外的な場合を除いて、多項式オーダーの時間で最適解を得ることができないことを示している。

ジャクソンルールなどは、構成的アルゴリズムである。前述のジャクソンルールなどの例外を除いて多項式オーダーの時間で最適解を得ることができないのであれば、総所要時間を最小化する構成的アルゴリズムもできないと考えられる。したがって、ジョブショップにおける構成的アルゴリズムでは、最適性を望まない良好な性能を追い求めることになる。優先番号法が代表的な構成的アルゴリズムである。探索的アルゴリズムであるブランチアンドバウンド法 (BAB) のバックトラッキングを省略した方法も、構成的アルゴリズムである。

このようにジョブショップにおける構成的アルゴリズムは、数多く提案されたわけではなく、新たなヒューリスティック・アルゴリズムの開発が待たれている。

(2) 探索的スケジューリング

最適解を含む可能性のあるスケジュール集合を列挙する方法が、この方法の基盤である。特殊な場合を除き一般に最適解を含む最も小さな集合はアクティブ・スケジュール集合であって、この集合の列挙法は[8]によって示された。その列挙アルゴリズムは[2]、[7]にも示されている。アクティブ・スケジュールの総数は m 機械 n ジョブ問題で $(n!)^m$ のオーダーとなるために完全列挙によって最適解を探索することは事実上不可能である。したがって、どのように探索範囲を見込みのあるスケジュール集合に絞り込むかが鍵となる。

ブランチアンドバウンド法 (BAB) は、部分スケジュールを生成しながら、部分スケジュールに基づく完全スケジュールによる総所要時間の可能性のある最小値、すなわち下界を計算し、

下界が最も小さい部分スケジュールを生成するジョブを追加し、新たな部分スケジュールとする。これを繰り返し、一旦完全スケジュールに至る。その後、以下のバックトラッキングと呼ばれる処理を行う。得られた完全スケジュールによる総所要時間より下界が小さな部分スケジュールに戻り、同様に完全スケジュールに至る。途中でこれまで得られた総所要時間の最小値を当該スケジュールによる総所要時間が越すようであれば、その部分スケジュールから出発する探索を打ち切る。このように、総所要時間が最小となる可能性のある部分スケジュール全てを探索することによって、最終的に総所要時間最小のスケジュールが得られる。

BABについては下界の計算方法が解法の効率を左右するため、下界の計算方法について多くの論文がある。しかしながら、基本的には列挙法であるので、最適解が得られる保証がある反面、大きな問題に対して実務で使用するには計算時間がかかりすぎるとの難点がある。

ローカルサーチ法は、BAB法が系統立ててスケジュール集合全体を探索するのに対して、適当な完全スケジュールから出発して、小変更を繰り返し、逐次的にスケジュールを改良してゆく方法である。既存スケジュールにどのようなルールで小変更を加えるによって数々の方法が提案され、研究されている。これらは、たとえば[16]にまとめられている。

3. 新たなアルゴリズム：NEH-FCFS

(1) N-FCFSスケジュール集合

著者は、[18]において環境変動に対して頑強であって、ノンディレイ・スケジュールの部分集合であるN-FCFSスケジュールとの新たなスケジュールのタイプを提案した。背景には、実務上良好な総所要時間を提供するスケジュールのタイプであるノンディレイ・スケジュールの集合総数が、アクティブ・スケジュールの集合総数よりも小さいが、実際のスケジュールを探索するためにはいまだ大きいことがあった。同

論文で、著者は、N-FCFSタイプのスケジュール集合が、ノンディレイ・スケジュール集合内の総所要時間の小さいスケジュールをコンパクトにまとめた集合であることを総列挙法によって確認するとともに、加工時間の変動に対して安定度の高い集合であることを明らかにしている。

N-FCFSスケジュールは、各ジョブの第一オペレーションの加工順序が決まると、第二オペレーション以降はFCFS (First Come First Service) によって生成される。所与のジョブショップに対してN-FCFSスケジュール集合の要素数は第一オペレーションの加工順序の総列挙数となる。N-FCFSスケジュールは、フローショップにおける順列スケジュールのジョブショップへの自然な拡張となっている。フローショップにおいては、順列スケジュールを最適スケジュールの探索の対象とすることが合理的であるとされている。順列スケジュールとは、フローショップにおいて第一機械でのスケジュールを第二機械以降でも踏襲するスケジュールである。

(2) フローショップのNEHアルゴリズム

N-FCFSスケジュール集合は、総所要時間が短かつ加工時間変動に頑健なスケジュールを生成する方法論の基礎となり得る。構成的アルゴリズムによって、N-FCFSタイプのスケジュールから、一つの有望なスケジュールを作成するには、第一オペレーションの加工順序を決定する必要がある。N-FCFSスケジュールは、フローショップにおける順列スケジュールのジョブショップへの自然な拡張であることから、フローショップにおいて有効なヒューリスティックを援用して第一オペレーションの加工順序を決めることを考える。

フローショップにおいては、良好な性能を示す構成的ヒューリスティックが提案されてきている。代表的なものとしてCDSアルゴリズム[5]とNEHアルゴリズム[12]がある。フローショップの構成的ヒューリスティックの性能が近年の研究で比較されている (たとえば[6]、[13])。

これらの研究のおおむね一致した実験結果は、フローショップにおいて最も総所要時間を短くできるヒューリスティックは現時点でNEHアルゴリズムであることである。[6]では、NEHアルゴリズムをランダムショップ問題へ応用する可能性を示唆している。

以下にNEHアルゴリズムを紹介する。

1. 各ジョブの総加工時間を求める。
2. 全ジョブを1.で求めた総加工時間の大きい順に並べたリストを作成する。
3. 2.で作成したリストから上位2ジョブを取り出し (リストからこの2ジョブは削除する)、この2ジョブによる加工順序2通りのうち、総所要時間が短いほうを初期部分スケジュールとする。以後、この2つのジョブの先行関係は変わらない (間に別のジョブが挿入されることはあるが)。
4. リストより最上位のジョブを選び、既存の部分スケジュールに挿入する。リストからこの最上位のジョブを削除する。全ての挿入箇所 (2.で作成したリストで*i*番目のジョブであれば挿入可能箇所は*i*箇所である) を試し、総所要時間最小となる挿入箇所のスケジュールを新たな部分スケジュールとする。
5. 全てのジョブを挿入し終わるまで4.を繰り返す。

(3) NEH-FCFSアルゴリズムと例題

NEHアルゴリズムを援用し、第一オペレーションの加工順序付けを決め、第二オペレーション以降をFCFSでスケジュールするヒューリスティック・ルールを提案する。これを以下、NEH-FCFSアルゴリズムと呼ぶ。NEH-FCFSアルゴリズムで生成されるスケジュールは、N-FCFSスケジュールの一種である。

NEH-FCFSアルゴリズム (図2)

1. 各ジョブの総加工時間を求める。
2. 全ジョブを1.で求めた総加工時間の大きい

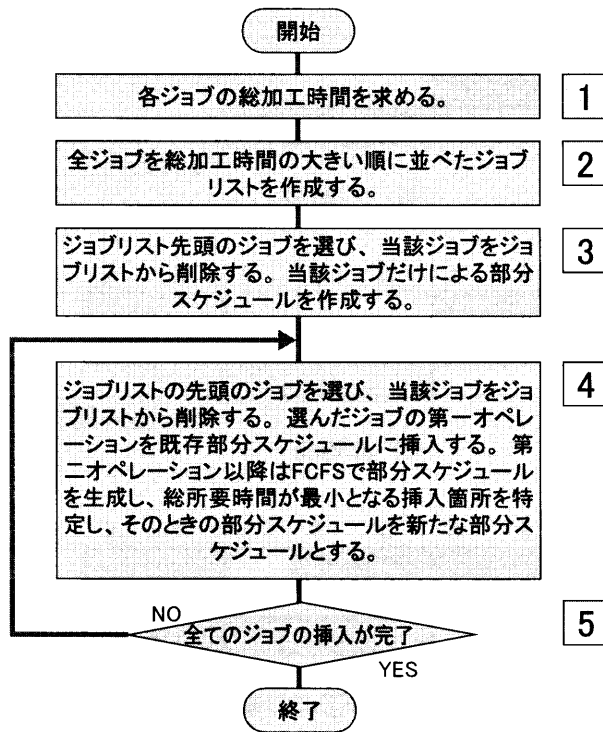


図2 NEH-FCFSアルゴリズムのフローチャート

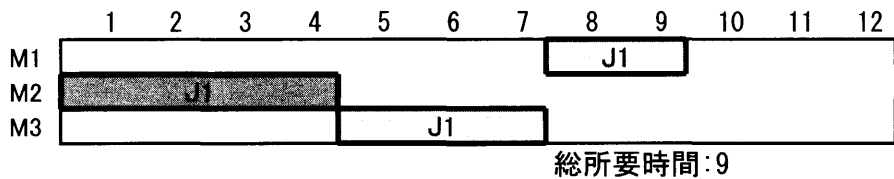
加工時間				加工経路			
	1	2	3		1	2	3
J1	4	3	2	J1	2	3	1
J2	2	1	1	J2	2	1	3
J3	4	1	1	J3	1	3	2
J4	2	1	3	J4	1	3	2

図3 例題の加工時間と加工経路

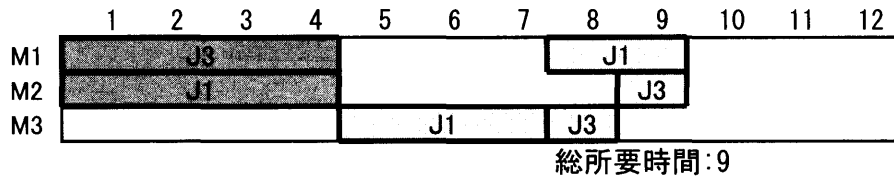
- 順に並べた（総加工時間が同じジョブはジョブ番号の小さいもの優先）ジョブリストを作成する。
- ジョブリストの先頭のジョブ（総加工時間が最大なジョブ）を選び、当該ジョブだけによる部分スケジュールを作成する。当該ジョブをジョブリストから削除する。
 - ジョブリストの先頭のジョブを選び、当該ジョブをジョブリストから削除する。選んだジョブをスケジュールに挿入する。挿入箇所は既存部分スケジュールに対して可能な挿入箇所全てについて試す。第二オペレーション以降はFCFSで当該ジョブを加えた

- 部分スケジュールを生成し、総所要時間を求め、総所要時間が最小となる挿入箇所を特定し、そのときの部分スケジュールを新たな部分スケジュールとする。FCFSとは、加工待ちのジョブの中から、前の機械での加工が最も早く終わったジョブを次の加工ジョブとする。前の機械での加工が同時に終わっている場合は、ジョブ番号の小さいものを優先する。
- 全てのジョブの挿入が終了したらスケジューリング完了、さもなければ4.に戻る。

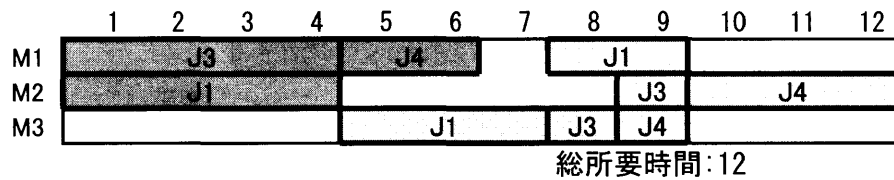
例題



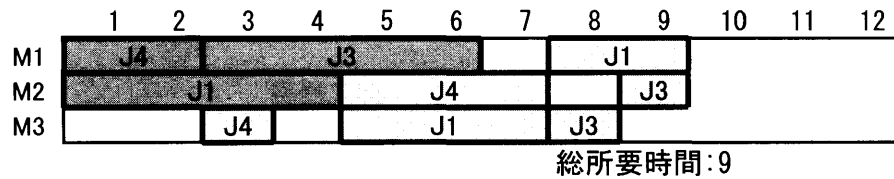
(a) J1追加



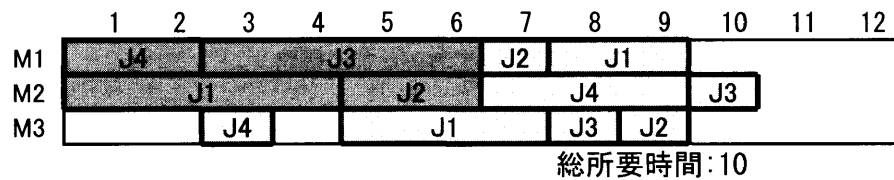
(b) J3追加



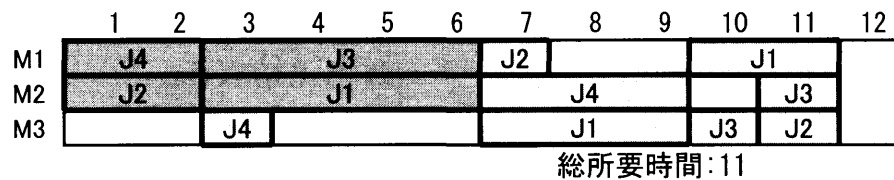
(c) J4追加①J3の後に挿入



(d) J4追加②J3の前に挿入



(e) J2追加①J1の後に挿入



(f) J2追加②J1の前に挿入


 第一オペレーション

図4 例題に関するガントチャート

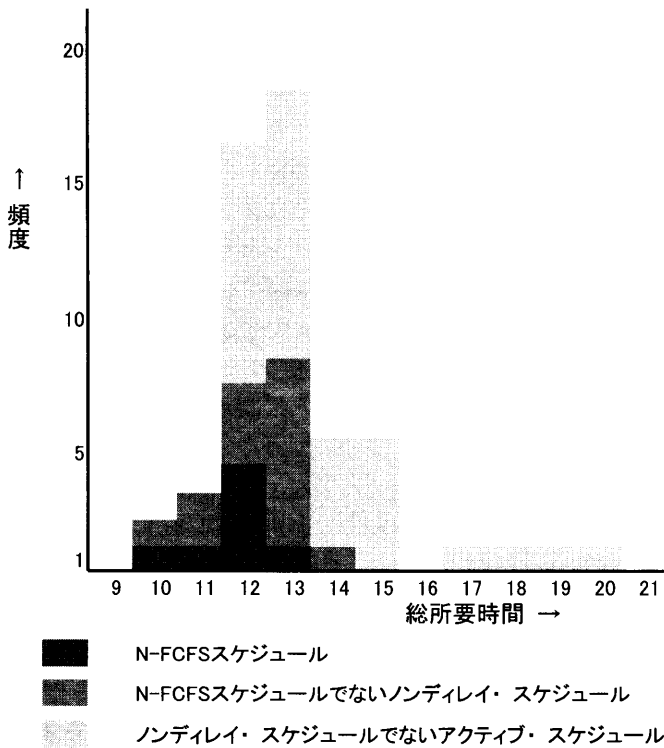


図5 例題の全アクティブ・スケジュールによる総所要時間分布

以下にNEH-FCFSアルゴリズムによるスケジュール作成例を述べる。

図3の3機械4ジョブ問題を考える。

Step 1. J1の総加工時間：4+3+2=9

J2の総加工時間：2+1+1=4

J3の総加工時間：4+1+1=6

J4の総加工時間：2+1+3=6

Step 2. J1、J3、J4、J2

Step 3. J1による部分スケジュール生成。総所要時間：9（図4(a)参照）。

Step 4. 部分スケジュールにJ3追加。第一オペレーション挿入可能箇所は1箇所。総所要時間：9（図4(b)参照）

Step 4. 部分スケジュールにJ4追加。第一オペレーション挿入可能箇所は2箇所。総所要時間：12と9（図4(c)と(d)参照）、総所要時間：9の挿入箇所を採択。こちらを新部分スケジュールとする。

Step 4. 部分スケジュールにJ2追加。第一オペレーション挿入可能箇所は2箇所。総所要時間：10と11（図4(e)と(f)参照）、10の挿入箇所を採択。

Step 5. 全ジョブをスケジューリング完了

例題についてのスケジュール総列挙実験の結果、を図5に示す。得られた総所要時間10は、総所要時間の最小値である。

NEH-FCFSアルゴリズムの所要計算時間オーダ

NEH-FCFSアルゴリズムの所要計算時間の意味で最悪ケースは、第一オペレーションが同一機械に集中する場合である。このとき、ジョブ数が n とすれば、第一オペレーションの挿入箇所合計は、 $1+2+\dots+(n-1)+n=n(n-1)/2$ となる。このことから類推できるNEH-FCFSアルゴリズムによる計算時間の増加オーダは、 n^2 である。これは、NEHアルゴリズムと同等であ

る。

4. ベンチマーク問題による検証実験

構成的アルゴリズムであるNEH-FCFSによって得られるスケジュールによる総所要時間を、探索的アルゴリズムによってこれまで明らかになってきたベンチマーク問題についての最良解と比較し、総所要時間の意味におけるNEH-FCFSアルゴリズムの良好度を絶対評価することが目的である。

(1) ベンチマーク問題とは

Taillard[15]が指摘するように、ジョブショップ・スケジューリング問題を完全に解く方法が存在せず、ヒューリスティックな方法による改善の解にとどまる現状から、ヒューリスティックな方法の性能の基準を示すためのベンチマーク問題が必要とされる。10機械以下のベンチマーク問題は[3]などにある。Taillard論文では、10機械以下の問題はおおむね良好な解が得られるようになったことから、15機械以上の具体的な問題を提示している。Taillardは、良い解に到達するのが「困難な問題」を選んだとしている。ここで「困難な問題」とは、探索的な繰り返し法で良い解を探索する場合に、総所要時間の下界と得られた良い解がかけ離れている問題や初期値を変えるたびに解が異なってくる問題である。

(2) ベンチマーク問題による実験結果

表1に、Taillard以前の10機械以下のベンチマーク問題およびTaillardによる15機械以上のベンチマーク問題に対するNEH-FCFSアルゴリズムによる結果と、これまでに既知となっている最良解([9]などによる)を示す。最良解は、多くの研究者が様々な繰り返し最適化アルゴリズムで競って探索してきている解であるのに対して、NEH-FCFSアルゴリズムは構成的(一度得た完全解を繰り返し計算で改良しない)であるので、結果は、NEH-FCFSアルゴリズムが劣

る。NEH-FCFSアルゴリズムによる解は、繰り返し最適化アルゴリズムに使う初期解と解釈すべきである。

表1は、NEH-FCFSアルゴリズムによる解は、問題が小規模な問題であるほど、また、機械台数に比してジョブ数が多いとき有効性を増すことを示している。この現象は、問題が小規模な問題であるほど第一オペレーションのスケジュールで性能が左右されやすいことと、機械台数に比してジョブ数が多いほど、NEH-FCFSアルゴリズムで試行する部分スケジュール数(NEH-FCFSアルゴリズム4.の繰り返し回数)が増えることで説明できよう。表1の結果は、[14]による機械台数とジョブ数が近い場合にとりわけ最適解を探索することが困難である(最適解を求めにくいような解の分布となる)との指摘を思い起こさせる。また、[17]は、最適解の5/4(1.25)倍未満の長さの総所要時間を厳格に保証するスケジュールを多項式時間で生成するアルゴリズムが存在しないほどジョブショップ問題は困難であると、述べていることも付記しておく。

5. フロー化率をパラメータとしたジョブショップ・シミュレーションによる検証実験

本研究で提案しているNEH-FCFSアルゴリズムは、フローショップにおける順列スケジュール集合をランダムショップへ拡張したものであるN-FCFSタイプのスケジュール集合から、フローショップで定評のあるNEHアルゴリズムを使って、性能の良いものを選び出すとの発想に基づいている。したがって、ランダムショップからフローショップまでの広い範囲のジョブショップ条件で性能の検証を行うことにする。性能の比較対象は、NEH-FCFSアルゴリズムと同じく構成的アルゴリズムである優先番号法とする。

本実験の目的は、第一に、ベンチマーク問題がとくに困難な純然たるランダムショップを想定しているのに対して、フローショップに近い

ランダムジョブまでの広い範囲のジョブジョブ条件で性能の検証を行うことである。提案したNEH-FCFSと同じ構成的アルゴリズム（優先番号法）の枠内で、NEH-FCFSアルゴリズムの良好度を総所要時間で評価すること、いわば相対評価が、第二の目的である。

(1) ジョブジョブと実験の条件

全てのジョブが全ての機械による加工を一度ずつ必要とする古典ジョブジョブモデルを対象とする。すなわち各ジョブのオペレーション数は機械台数に等しい。各オペレーションの加工時間は、平均1の指数分布乱数から発生させた数値を切り上げた整数値に従うものとした。各ジョブの加工経路は、フロー化率が0のとき、ランダムに生成させている。フロー化率が f のとき、各ジョブの加工経路が確率 f で機械番号順になるように生成させた。したがって、フロー化率が f のとき、ジョブ数を N とすれば、生成されたジョブジョブが純然たるフロージョブである確率は、 f^N である。 $f=0.8$ 、 $N=5$ で0.3268となる。ジョブ数が大きければ、この確率はさらに小さくなる。

総所要時間を性能評価尺度として、FCFS、SPT、MOPR、MWRの4つの優先番号法を比較対象とした。

5機械5ジョブ（5M×5N）、10機械10ジョブ（10M×10N）、15機械15ジョブ（15M×15N）の3つの規模のジョブジョブについて、フロー化率0、0.2、0.4、0.8の各水準について、1000のジョブジョブを発生させ、性能比較を行った。

表2に実験結果を示す。表中、WinはNEH-FCFSアルゴリズムによる総所要時間のほうが短い場合、Evenは等しかった場合、Lostは長かった場合を示す。Caseは場合の数である。 Δ は、両方法による総所要時間の差の平均値を、勝った方法による総所要時間の平均値で除したものである。 Δ は、どれほどの差でWinおよびLostとなったかの程度を表す。

(2) 実験結果と考察

表2より以下のことが読み取れる。

1. 問題規模が小さいほうが、優先番号法よりもNEH-FCFSアルゴリズムの有効度が高い。
2. フロー化率が高いほうが、優先番号法よりもNEH-FCFSアルゴリズムの有効度が高い。この傾向は問題規模が大きいくほど強くなる。
3. 実験範囲で、NEH-FCFSアルゴリズムよりも優先番号法の有効度が高いのは、優先番号法MOPRとMWRに対して、フロー化率が低い場合である。この傾向は問題規模が大きいくほど強くなる。

問題の規模が小さい場合、第一オペレーションの順序付けをNEHアルゴリズムに従って行うことによる優先番号法に対する優位性が（初期条件の優位性が）持続するものと考えると1.が説明できる。フロー化率が高いほうが、NEH-FCFSアルゴリズムの原点であるフロージョブに近くなることによって、1.の優位性の問題規模が大きくなることによる消失を補ってゆくことが、2.の理由であろう。問題規模が大きく、フロー化率が小さい場合、1.および2.のNEH-FCFSの有効性を支える理由が弱まり、第二オペレーション以降の順序付けに、当該ジョブの未完了オペレーションに関する情報を加味するMOPR、MWRの相対的優位性が増すものと考えられる。このことは、第二オペレーション以降の順序付けにFCFSではなくMOPR、MWRを使うことの可能性を示唆している。しかし、頑健性はN-FCFSよりも悪化する可能性がある。

ベンチマーク問題のところですでに触れているが、フロー化率が高いほうがNEH-FCFSアルゴリズムの有効度が高い現象を、より掘り下げれば以下の通りである。フロージョブに近いほど、第一オペレーションの順序付けが全体スケジュールの性能をより強く支配することになる。また、フロージョブに近いほど、同一機械に第一オペレーションが集中するため、NEH-FCFSアルゴリズムで総所要時間を試す部分スケジュール数（NEH-FCFSアルゴリズム4.の繰り返し回数）が増えることになる。試す部分

表1 ベンチマーク問題による総所要時間

問題規模	10M × 10N									
問題名	MT10	ABZ5	ABZ6	LA19	LA20	ORB1	ORB2	ORB3	ORB4	ORB5
最良値	930	1234	943	842	902	1059	888	1005	1005	887
NEH-FCFS	1226	1376	1174	979	1052	1236	1016	1437	1194	1028
性能比	1.32	1.12	1.24	1.16	1.17	1.17	1.14	1.43	1.19	1.16
問題規模	5M × 10N					5M × 15N				
問題名	LA01	LA02	LA03	LA04	LA05	LA06	LA07	LA08	LA09	LA10
最良値	666	655	597	590	593	926	890	863	951	958
NEH-FCFS	735	834	711	679	610	934	985	978	951	996
性能比	1.10	1.27	1.19	1.15	1.03	1.01	1.11	1.13	1.00	1.04
問題規模	15M × 15N									
問題名	TA01	TA02	TA03	TA04	TA05	TA06	TA07	TA08	TA09	TA10
最良値	1231	1244	1222	1181	1233	1243	1228	1220	1282	1259
NEH-FCFS	1535	1502	1519	1535	1459	1528	1569	1479	1558	1548
性能比	1.25	1.21	1.24	1.30	1.18	1.23	1.28	1.21	1.22	1.23
問題規模	15M × 20N									
問題名	TA11	TA12	TA13	TA14	TA15	TA16	TA17	TA18	TA19	TA20
最良値	1376	1377	1367	1345	1366	1371	1480	1413	1352	1362
NEH-FCFS	1852	1687	1807	1771	1822	1749	1925	1830	1742	1680
性能比	1.35	1.23	1.32	1.32	1.33	1.28	1.30	1.30	1.29	1.23
問題規模	20M × 20N									
問題名	TA21	TA22	TA23	TA24	TA25	TA26	TA27	TA28	TA29	TA30
最良値	1663	1626	1574	1660	1598	1657	1704	1626	1629	1614
NEH-FCFS	2063	2059	1961	2097	2226	2155	2026	1921	2006	1921
性能比	1.24	1.27	1.25	1.26	1.39	1.30	1.19	1.18	1.23	1.19
問題規模	15M × 30N									
問題名	TA31	TA32	TA33	TA34	TA35	TA36	TA37	TA38	TA39	TA40
最良値	1770	1841	1832	1851	2007	1844	1815	1700	1811	1720
NEH-FCFS	2294	2315	2389	2468	2342	2326	2287	2256	2204	2075
性能比	1.30	1.26	1.30	1.33	1.17	1.26	1.26	1.33	1.22	1.21
問題規模	20M × 30N									
問題名	TA41	TA42	TA43	TA44	TA45	TA46	TA47	TA48	TA49	TA50
最良値	2064	1983	1896	2031	2032	2057	1947	2001	2013	1973
NEH-FCFS	2631	2653	2339	2799	2600	2514	2418	2524	2437	2677
性能比	1.27	1.34	1.23	1.38	1.28	1.22	1.24	1.26	1.21	1.36
問題規模	15M × 50N									
問題名	TA51	TA52	TA53	TA54	TA55	TA56	TA57	TA58	TA59	TA60
最良値	2760	2756	2717	2839	2679	2781	2943	2885	2655	2723
NEH-FCFS	3508	3303	3067	3251	3259	3211	3370	3396	3180	3106
性能比	1.27	1.20	1.13	1.15	1.22	1.15	1.15	1.18	1.20	1.14
問題規模	20M × 50N									
問題名	TA61	TA62	TA63	TA64	TA65	TA66	TA67	TA68	TA69	TA70
最良値	2868	2902	2755	2702	2725	2845	2841	2784	3071	2995
NEH-FCFS	3579	3547	3281	3242	3361	3443	3379	3221	3527	3675
性能比	1.25	1.22	1.19	1.20	1.23	1.21	1.19	1.16	1.15	1.23
問題規模	20M × 100N									
問題名	TA71	TA72	TA73	TA74	TA75	TA76	TA77	TA78	TA79	TA80
最良値	5464	5181	5568	5339	5392	5342	5436	5394	5358	5183
NEH-FCFS	6118	5681	6153	5913	6344	5871	5898	6048	5845	5789
性能比	1.12	1.10	1.11	1.11	1.18	1.10	1.08	1.12	1.09	1.12

表2 NEH-FCFSアルゴリズムと優先番号法の総所要時間比較

優先番号法種別	0						0.2						0.4						0.8						
	FCFS	SPT	MOPR	MWR	FCFS	SPT	MOPR	MWR	FCFS	SPT	MOPR	MWR	FCFS	SPT	MOPR	MWR	FCFS	SPT	MOPR	MWR	FCFS	SPT	MOPR	MWR	
5M X 5N	Case	481	483	405	376	486	453	413	398	486	453	413	398	524	473	468	487	524	473	468	487	703	533	682	679
	△	0.127	0.129	0.119	0.117	0.124	0.122	0.116	0.119	0.126	0.118	0.118	0.122	0.126	0.118	0.118	0.122	0.142	0.111	0.139	0.141	0.142	0.111	0.139	0.14
	Even	422	380	465	466	416	404	461	429	400	397	423	351	264	392	266	265	265	266	266	266	264	392	266	265
10M X 10N	Case	97	137	130	158	98	143	126	173	98	143	126	173	76	130	109	162	76	130	109	162	33	75	52	56
	△	0.086	0.096	0.086	0.088	0.081	0.092	0.087	0.09	0.088	0.098	0.096	0.088	0.086	0.098	0.096	0.088	0.086	0.094	0.087	0.094	0.086	0.094	0.087	0.093
	Win	614	586	429	383	668	564	479	444	749	561	591	562	953	791	915	893	953	791	915	893	953	791	915	893
15M X 15N	Case	0.087	0.095	0.068	0.066	0.089	0.092	0.074	0.071	0.103	0.088	0.089	0.087	0.103	0.088	0.089	0.087	0.15	0.094	0.14	0.141	0.15	0.094	0.14	0.141
	△	0.057	0.068	0.064	0.07	0.068	0.07	0.078	0.078	0.054	0.066	0.062	0.065	0.045	0.052	0.052	0.052	0.045	0.052	0.052	0.045	0.052	0.052	0.052	0.05
	Even	643	562	332	297	666	524	386	342	809	532	602	546	984	814	936	954	809	532	602	546	984	814	936	954
5M X 5N	Case	0.07	0.084	0.05	0.05	0.081	0.073	0.061	0.054	0.095	0.071	0.076	0.069	0.153	0.089	0.142	0.137	0.095	0.071	0.076	0.069	0.153	0.089	0.142	0.137
	△	0.047	0.06	0.06	0.064	0.051	0.063	0.058	0.063	0.039	0.058	0.053	0.06	0.02	0.041	0.037	0.047	0.039	0.058	0.053	0.06	0.02	0.041	0.037	0.047
	Even	120	115	157	168	119	116	152	121	62	111	113	117	9	72	30	18	62	111	113	117	9	72	30	18
10M X 10N	Case	237	323	511	535	215	360	462	537	215	360	462	537	129	357	285	337	129	357	285	337	7	114	34	28
	△	0.047	0.06	0.06	0.064	0.051	0.063	0.058	0.063	0.039	0.058	0.053	0.06	0.02	0.041	0.037	0.047	0.039	0.058	0.053	0.06	0.02	0.041	0.037	0.047
	Lost	0.047	0.06	0.06	0.064	0.051	0.063	0.058	0.063	0.039	0.058	0.053	0.06	0.02	0.041	0.037	0.047	0.039	0.058	0.053	0.06	0.02	0.041	0.037	0.047

Win: NEH-FCFSによる総所要時間が各優先番号法による総所要時間より短い
 Even: NEH-FCFSによる総所要時間が各優先番号法による総所要時間と同じ
 Lost: NEH-FCFSによる総所要時間が各優先番号法による総所要時間より長い
 Case: 場合の数 (合計は実験回数1000)
 △: NEH-FCFSによる総所要時間と各優先番号法による総所要時間の差の平均を短かった方法の総所要時間平均で除したたもの

スケジュール数が少なくなる条件下（機械台数に比較してジョブ数が少ない）では、NEH-FCFSアルゴリズムは実質的にFCFSそのもの（NEH-FCFSアルゴリズム 4. の繰り返し回数が1であればアルゴリズムはFCFSと一致する）となり、MOPR、MWRに敗れる場合が増えてくる。このような条件下で試す部分スケジュール数が増えるようにすることによって、NEH-FCFSアルゴリズムを改良する余地があるかもしれない。

6. おわりに

N-FCFSスケジュール集合が、加工時間変動に強く、総所要時間が良好なスケジュール集合であることを受けて、N-FCFSスケジュール集合に属するスケジュールを生成する構成的アルゴリズム、NEH-FCFSアルゴリズムを提案した。NEH-FCFSアルゴリズムにより生成されるスケジュールの性能を、ベンチマーク問題とフローショップからランダムショップまでのジョブショップを想定して検証した。NEH-FCFSアルゴリズムで得られるスケジュールはノンディレイ・スケジュールの部分集合であるN-FCFS集合に属するため、加工時間変動に対して頑健である可能性が高い。言い換えれば、加工時間変動に対して頑健である性質を保ったまま、良好な性能をもつスケジュールを拾い出す方法としてNEH-FCFSアルゴリズムを提案した。

以上の結果、以下の事柄が明らかになった。

1. ベンチマーク問題に対して、繰り返し法によって得られている最良解と比較するときNEH-FCFSアルゴリズムによる解は、問題が小規模な問題であるほど、また、ジョブ数・機械台数が大きいほど最良解に近いことを示した。
2. フローショップからランダムショップまでのジョブショップを想定した問題に対しては、規模が小さいほど、またジョブショップがフローショップに近いほど、優先番号

法よりもNEH-FCFSアルゴリズムの有効度が高いことを示した。

3. ジョブショップがフローショップに近いほど、優先番号法よりもNEH-FCFSアルゴリズムの有効度が高い傾向は、問題規模が大きいほど強くなる。

現時点での生産マネジメントは、ランダムショップをフローショップに向けて改善してゆくとの方向にある。加工時間の変動に頑健な性格をもつことに加え、フローショップに近いショップで有効度の高い構成的アルゴリズムである、提案したNEH-FCFSアルゴリズムは実用度が高いものと考えられる。

今後の課題として、第二オペレーション以降の順序付けにSPT、MOPR、MWRなどの優先番号法を使う場合の、加工時間変動への頑健性と性能の検討が考えられる。

参考文献

- [1] Aytug, H., Lawley, M. A., McKay, K., Mohan, S. and Uzsoy, R. : "Executing Production Schedules in the Face of Uncertainties: A Review and Some Future Directions". *European Journal of Operational Research*, Vol.161, No.1, pp.86-110 (2005)
- [2] Baker, K. R. : *Introduction to Sequencing and Scheduling*, John Wiley & Sons (1974)
- [3] Beasley, J.E. : "OR-Library: Distributing Test Problems by Electronic Mail", *Journal of Operational Research Society*, Vol.41, pp.1069-1072 (1990)
- [4] Brucker, P. : "An Efficient Algorithm for the Job-shop Problem with Two Jobs", *Computing*, Vol.40, pp.353-359 (1988)
- [5] Campbell, H.G., Dudek, R.A. and Smith, M. L. : "A Heuristic Algorithm for the n Job m Machine Sequencing Problem", *Management Science*, Vol.16, pp.630-637 (1970)

- [6] Framinan, J. M., Leisten, R. and Rajendran, C. : “Different Initial Sequences for the Heuristic of Nawaz, Ensore and Ham to Minimize Makespan, Idletime or Flowtime in the Static Permutation Flowshop Sequencing Problem”, *International Journal of Production Research*, Vol. 41, No.1, pp.121-148 (2003)
- [7] French S. : *Sequencing and Scheduling - An introduction to the mathematics of the Job-Shop*, Ellis Horwood (1982)
- [8] Giffler B. and Thompson G. L. : “Algorithm for Solving Production Scheduling Problems”, *Operations Research*, Vol.8, pp.487-503 (1960)
- [9] Goncalves, J.F., Mendes, J.J. de M. and Resende, M.G.C. : “A Hybrid Generic Algorithm for the Job Shop Scheduling Problem”, *AT&T Labs Research Technical Report TD-5EAL6J* (2002)
- [10] Jackson J. R. : “An Extension of Johnson’s Results on Job Lot Scheduling”, *Naval Research Logistics Quarterly*, Vol.3, pp.201-203 (1956)
- [11] Morton T. E. and Pentico D. W. : *Heuristic Scheduling Systems with Applications to Production Systems and Project Management*, John Wiley & Sons (1993)
- [12] Nawaz, M; Encore E.E. Jr; Ham I. : “A Heuristic Algorithm for the m-machine, n-Job Flow-shop Sequencing Problem”, *OMEGA International Journal of Management Science*, Vol.11, No.1, pp.91-95 (1983)
- [13] Ruiz, R. and Maroto, C. : “A Comprehensive Review and Evaluation of Permutation Flowshop Heuristics”, *European Journal of Operational Research*, Vol. 165, No.2, pp.479-494 (2005)
- [14] Streeter M. J. and Smith, S. F. : “How the Landscape of Random Job Shop Scheduling Instances Depends on the Ratio of Jobs to Machines”, *Journal of Artificial Intelligence Research*, Volume 26, pp.247-287 (2006)
- [15] Taillard E. : “Benchmarks for Basic Scheduling Problems”, *European Journal of Operational Research*, Vol.64, pp.278-285 (1993)
- [16] Vaessens, R.J.M., Aarts, E.H.L. and Lenstra, J.K. : “Job Shop Scheduling by Local Search”, *INFORMS Journal on Computing*, Vol. 8 , No.3, pp.302-317 (1996)
- [17] Williamson, D. P., Hall, L. A., Hoogeveen, J. A., Hurkens, C. A. J., Lenstra, J. K., Sevast’Janov, S. V. and Shmoys, D. B. : “Short Shop Schedules”, *Operations Research*, Vol. 45, No.2, pp.288-294 (1997)
- [18] 松浦春樹 : “加工時間変動に対して頑健なジョブショップ・スケジュールの新しいタイプ”、神奈川大学国際経営研究所「国際経営フォーラム」, No.19, pp.209-224 (2008)