

## 多目的評価による JSP の最適化

## Optimization on Job-shop Scheduling Problem with Multiple Purpose Evaluation

○学 島倉 渉 (北見工業大学) 正 渡辺 美知子 (北見工業大学)  
正 鈴木 育男 (北見工業大学) 正 岩館 健司 (北見工業大学)

Wataru SHIMAKURA, Kitami Institute of Technology, simakurawataru@gmail.com  
Michiko WATANABE, Kitami Institute of Technology  
Ikuo SUZUKI, Kitami Institute of Technology  
Kenji IWADATE, Kitami Institute of Technology

The Job-shop Scheduling Problem is a classical, but yet modern problem. Because it belongs to the NP-hard problem and it is almost impossible to find out the exact solution, many heuristic methods have been proposed. Recently, Genetic Algorithm (GA), Local Clustering Organization (LCO) and so on have been proposed as a powerful tool to solve the problem. The optimal schedule becomes very important from a viewpoint of cost reduction in the manufacturing industry. This study optimizes the job-shop scheduling with multiple purpose functions by using evolutionary computation. Numerical experiments verify that the evolutionary computation obtains (quasi-) optimal schedule with multiple purpose functions.

**Key Words:** Genetic Algorithm, Job-shop Scheduling Problem, Random Expert Method, Multiple Genes

## 1. 緒言

最適化問題とは、制約条件を満たす範囲内で目的関数の最大値(最小値)をとるような設計変数を決定する問題のことである。しかし、最適化の対象となる問題の目的関数が複雑である場合、その厳密な最適解を求めることは非常に困難な問題となる。そのような困難な問題の1つにジョブショップスケジューリング問題 (Job-shop Scheduling Problem, JSP) がある。

ジョブショップスケジューリング問題は、複数の仕事と工作機械が与えられたときの順序付け組み合わせ問題である。この問題は、作業時間、製造コスト削減等に直結するため、会社存続や利益拡大をともなった大きな課題となっている。しかし、スケジュール作成では、組み合わせ数が増加すると計算時間も膨大となるため、熟練者でも多大な時間を要しているのが現状である。

本研究では、組合せ最適化問題の中でも最適解を得ることが困難とされている多目的のジョブショップスケジューリング問題に生物の進化を模倣した遺伝的アルゴリズム(Genetic Algorithm, GA)を適用して最良あるいは最適解を求めることを目的とする。また、多目的関数がどのように最適化されたのか自己組織化マップ(Self Organizing Map, SOM)を用いて可視的に分かるようにする。

この遺伝的アルゴリズムで用いる遺伝子の解表現は、渡辺らが開発した多重遺伝子を採用する。数値実験では、実際の製造業の生産環境の変化に対応させるために、複数の目的関数、制約条件下でスケジューリング作成を行い、提案手法の有効性を検証する。

## 2. ジョブショップスケジューリング問題 (Job-shop Scheduling Problem, JSP)

ジョブショップスケジューリング問題は複数の仕事と機械が与えられたとき、各機械上での最良の作業順序を求めることであり、すべての仕事に要する時間を最小にするようなスケジュール問題を基本としている。この時、各仕事を加工する機械の順序、各仕事の機械上での加工時間は既知である。

また、各仕事の機械上での加工時間は工具の取替時間などの段取り時間を含んだ時間として取り扱う。この条件を考慮した上で、以下の様な性質をもつ多目的スケジューリング問題である。

## 2.1 目的関数の設定

スケジューリング作成において考慮すべき点は、すべての仕事に要する時間を最小にすることであり、最大滞留時間の最小化を図ることが求められる。しかし、スケジュール作成の目的は、製造業によって様々であり、多くの目的関数で評価を行う必要がある。本論文では、納期遵守かつ需要と供給のバランスを重点に置き、不確実性への対応を行うためのスケジューリングをするために複数の目的関数を以下に設定する。

- 1) 最大滞留時間:  $F_1$
- 2) 平均滞留時間:  $F_2$
- 3) 機械遊休時間:  $F_3$
- 4) 納期遅れ時間:  $F_4$
- 5) 納期余裕時間:  $F_5$

## 2.2 制約条件

多目的スケジューリング問題の制約条件は、以下のとおりである。

- 仕事の処理順序は指定された作業からなる(技術的順序)
- 仕事の機械上での加工時間は既知である
- 機械は同時に複数の作業を処理することはできない
- 各機械は必ず全ての仕事を中断せずに処理する
- 設定した目的関数に反した値が算出された場合、ペナルティが発生する

## 2.3 定式化

仕事数を  $n$ 、機械数を  $m$  とした時に、仕事集合は  $J = (J_i : i = 1, 2, \dots, n)$  となり、機械集合の場合は  $M = (M_j : j = 1, 2, \dots, m)$  となる。工程による機械  $M_j$  で加工される時の工程時間  $T$  は、 $T = (T_{ij} : i = 1, 2, \dots, n, j = 1, 2, \dots, m)$ 、機械  $M_j$  の前工程から次工程への待ち時間  $W$  は、 $W = (W_{ij} : i = 1, 2, \dots, n, j = 1, 2, \dots, m)$  となる。

この時の最大滞留時間は、式(1)の総作業時間 $F_1$ を取り扱い、最小値を求める問題となる。

$$F_1 = \max \left( \sum_{j=1}^{q_i} T_{ij} + \sum_{j=1}^{q_i} W_{ij} \right) \quad (1)$$

平均滞留時間 $F_2$ は、

$$F_2 = \sum_{j=1}^{q_i} \left\{ \left( \sum_{j=1}^{q_i} T_{ij} + \sum_{j=1}^{q_i} W_{ij} \right) / m \right\} \quad (2)$$

となる。機械遊休時間 $F_3$ は、

$$F_3 = \sum_{j=1}^{q_i} \sum_{j=1}^{q_i} W_{ij} \quad (3)$$

となる。納期設定では、納期を $Z_j$ とした時、納期遅れ $F_4$ 、納期余裕 $F_5$ は以下の(4)、(5)で示す。

$$F_4 = \max \left( \sum_{j=1}^{q_i} T_{ij} + \sum_{j=1}^{q_i} W_{ij} \right) - Z_j \quad (4)$$

$$F_5 = Z_j - \min \left( \sum_{j=1}^{q_i} T_{ij} + \sum_{j=1}^{q_i} W_{ij} \right) \quad (5)$$

この時、評価関数 $F$ は、

$$F = F_1W_1 + F_2W_2 + F_3W_3 + F_4P_1 + F_5P_2 \quad (6)$$

となる。ここで、 $W_1$ 、 $W_2$ 、 $W_3$ は各評価のウェイト、 $P_1$ 、 $P_2$ は各評価でのペナルティを示している。

Table 1 Technical order of JSP

| Order<br>Job | Processing<br>in the first | Processing<br>in the second | Processing<br>in the third |
|--------------|----------------------------|-----------------------------|----------------------------|
| Job[1]       | ( $M_1,4$ )                | ( $M_2,5$ )                 | ( $M_3,3$ )                |
| Job[2]       | ( $M_2,3$ )                | ( $M_1,6$ )                 | ( $M_3,4$ )                |
| Job[3]       | ( $M_2,2$ )                | ( $M_3,4$ )                 | ( $M_1,2$ )                |

表1は、各種類の仕事を経験的順序により機械で処理する例を示している。例えば仕事[2]の場合、まず $M_2$ で3時間仕事を処理した後、 $M_1$ で6時間仕事を処理し、 $M_3$ で4時間仕事を処理する。

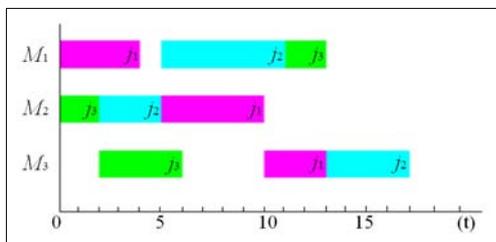


Fig1. Gantt chart of three jobs and three machines

図1は、表1を基に縦軸に複数の機械、横軸に時間とし、各機械で各仕事が行われる処理開始時間から処理終了時間までのガントチャートである。この例では、 $M_3$ が仕事を完了した時点で作業が完了し、最大滞留時間は17となる。

### 3. 遺伝的アルゴリズム(Genetic Algorithm, GA)

遺伝的アルゴリズムは、自然界の生物の進化を模倣した最

適化アルゴリズムである。具体的には、図2に示すように集団の個体間で淘汰(selection)、交叉(crossover)や突然変異(mutation)、逆位(inverse)などの遺伝オペレータによって次世代の集団個体が形成される。このように世代の更新が繰り返されることによって、最良解や最適解が求められる。

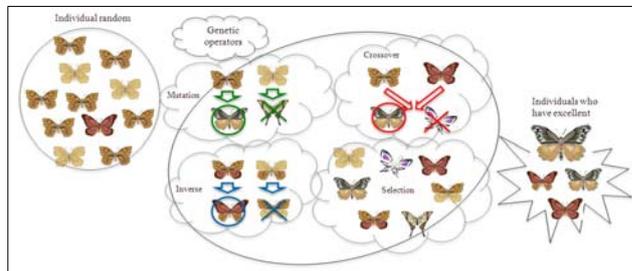


Fig2. Mechanism of genetic operators

### 3.1 多重遺伝子の設計

最初に、各仕事の遺伝子を機械数と同じだけ作成し、加工順序の情報を持つ遺伝子を初期遺伝子とする。次に初期遺伝子長間で乱数を発生させ、順序表現による遺伝子を作成する。この順序表現による遺伝子と初期遺伝子を基にジョブ遺伝子(Job Genes)を作成する。これらの手順を図3に示す。

具体的な多重遺伝子の手順は、初期遺伝子のビット数である $0 \sim n-1$ までの乱数 $r$ を発生し、その $r$ 番目の遺伝子を取り除き、初期遺伝子を左へ詰める。更に、順序表現遺伝子の左端に乱数 $r$ を入れる。この手順を繰り返す。ジョブ遺伝子は、初期遺伝子と順序表現遺伝子から作成される。

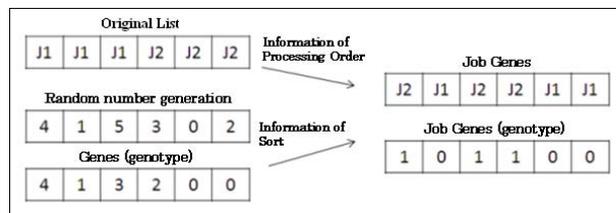


Fig3. Multiple Genes

## 4. 自己組織化マップ(Self Organizing Map)

T.Kohonenにより提唱された<sup>4)</sup>自己組織化マップ(Self Organizing Map, SOM)とは、ニューラルネットワークの一手法で与えられた入力情報の類似度をマップ上での距離で表現するモデルである。

SOMのアルゴリズムは、図4に示すようにまず参照ベクトルから1要素をランダムに選択し、選択された参照ベクトルからユークリッド距離が最短のニューロンを見つける。見つけた最短のニューロンの結合係数を以下の式により更新する。このとき、 $W$ はニューロンの結合係数、 $X$ は選択された参照ベクトル、 $\alpha$ は学習係数を示す。

$$W(t+1) = W(t) + \alpha(x(t) - W(t)) \quad (7)$$

近傍のニューロンにも同様の更新を行い、結合の強さに対して学習を行う。その後、学習係数、近傍係数を減らしながら繰り返していくことで入力パターン群をその類似度に応じて自律的に獲得する。

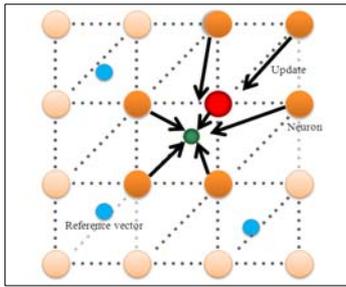


Fig4. Principle of SOM

JSPにSOMを適用することによって、最適化されたスケジューリングはどの多目的の評価に依存しているのか視覚的にとらえることが可能となる。

## 5. 熟練手法(Random Expert Method, REM)

熟練手法(Random Expert Method, REM)は、実際に人間がスケジューリングを作成する際に考慮するものを制約条件として設定したランダム手法である。最初に発生させた個体の目的関数の値を4分1以上削減させた個体群とし、納期設定を行っている。REMはGAと比較するために数値計算実験で用いる。

## 6. 数値計算実験

### 4.1 数値実験条件

数値実験の機械数, 仕事数, 加工順序, GAに関する条件は, 以下のように設定した。

#### • abz5

仕事数: 10, 機械数: 10  
加工順序: abz5のデータを基に設定  
厳密解: 1234  
個体数: 20, 世代数: 5000, エリート保存  
ウェイト( $W_1, W_2, W_3$ ): (1, 1, 0.5)  
ペナルティ( $P_1, P_2$ ): (0.5, 0)

#### [遺伝オペレータ]

交叉率: 0.6, 突然変異率: 0.2, 逆位率: 0.4

#### • abz9

仕事数: 20, 機械数: 15  
加工順序: abz9のデータを基に設定  
厳密解: 679  
個体数: 100, 世代数: 10000, エリート保存  
ウェイト( $W_1, W_2, W_3$ ): (1, 1, 0.5)  
ペナルティ( $P_1, P_2$ ): (0.5, 0)

#### [遺伝オペレータ]

交叉率: 0.9, 突然変異率: 0.1, 逆位率: 0.4

実験結果は, 10回実行した時の平均値を示している。また, 実験時間上限を300秒に設定している。

ウェイトの数値設定については, 評価値の値がウェイトによって均一になるように設定している。今回の数値実験では, 最小化を目的としているため, 目的に沿わない $P_2$ は値を0と定めている。

図5はabz5での遺伝オペレータ数値比較結果である。適正に合うように, 遺伝オペレータの数値を変更しながら10回実行した時の平均値を求め, そこから最大滞留時間が良い遺伝オペレータを選択するように調整した結果である。

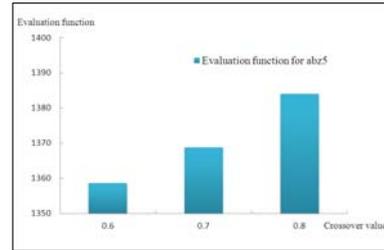


Fig.5 Convergence status of the solution

### 4.2 数値実験の結果

図6は縦軸に $F_1$ の評価値と横軸に世代数を示し, 4.1の実験条件で行ったabz5の評価値の収束状況である。世代数が少ない時は, GAの特徴である階段状に収束している。また, 1000世代目ごろからは緩やかに収束し, 5000世代目ではほぼ安定している。また, abz5からも同様の結果が得られた。

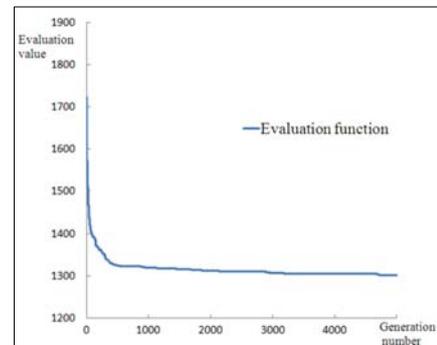


Fig6. Convergence status of the solution

図7は縦軸に $F$ の評価値と横軸に時間を示し, 4.1の実験条件で行ったabz9のGAとREMでの比較結果である。結果から300秒までの間では, GAは評価値が約4000で収束するのにに対し, REMの評価値は約7000となった。この結果からは, 膨大な数の個体を発生させても進化計算を行った方がJSPでは良好な解を短時間で導き出すことが可能であることが分かった。また, さらに時間を長くしたときの比較結果も同様の結果となった。

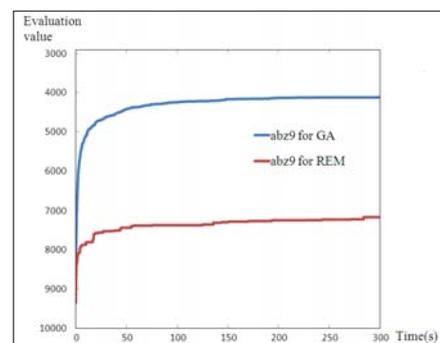


Fig7. Time comparisons using abz9

表2は, GAの評価値 $F_1$ の最良解, 最悪解, 平均値, 平均計算時間を表に示したものである。この時, 単目的はウェイト $W_1$ の値を1, その他のウェイト, ペナルティは0と設定し, 多目的はウェイト設定を実験条件の数値で算出している。こ

の結果も 10 回実行した時の値の平均値を算出している。

表 2 の結果からは、単目的と多目的とで比較した場合、abz5 では単目的と多目的では変化があまり見られなかった。しかし、abz9 では計算時間が約 1.35 倍になってしまうデメリットはあるが、多目的で実験を行った方が全ての  $F_1$  の評価値で良好な結果が算出されることが分かった。また、abz5 のように機械数、仕事数ともに 10 で小さい場合は単目的でも最適解に導くことが可能である。しかし、より複雑で大規模問題の場合は、多目的関数を導入してスケジューリングの最適化を行う必要がある。

Table 2 Experimental result GA and REM

| Problem        | Best | Worst | Average | computation time |
|----------------|------|-------|---------|------------------|
| abz5(Single)   | 1273 | 1322  | 1301.8  | 4.55(s)          |
| abz5(multiple) | 1266 | 1387  | 1320.2  | 6.08(s)          |
| abz9(Single)   | 831  | 871   | 850.4   | 177.61(s)        |
| abz9(multiple) | 786  | 853   | 821.6   | 241.20(s)        |

表 3 は、GA と REM の評価値  $F$  の最良解、最悪解、平均値、分散値、平均計算時間を表に示したものである。この結果も同様に 10 回実行した時の値の平均値を算出している。

表 3 の結果からは、abz5、abz9 どちらからも REM より GA の方が良好な結果が得られていることが分かった。また、abz5 より abz9 の方が GA と REM を比較した場合に差が大きくなる結果となった。これは abz9 の方が機械数 20、仕事数 15 と大規模問題となっていることが理由であると考えられる。したがって GA ではその他の大規模スケジューリング問題にも適用できることが確認できた。

一方、REM では GA と比べて分散値が低く解が安定しているものの、実験時間上限である 300 秒まででは設定した納期設定を達成出来なかった。

今回の数値実験では、GA、REMともにabz5問題、abz9問題の両方で厳密解を算出することは出来なかった。理由としては、個体間で淘汰を行ったが局所解に陥ってしまったことで遺伝オペレータを介す意味が無くなってしまったためだと考えられる。そのため、遺伝オペレータの改善、ウェイトの最適化などの改良が必要であると考えられる。

Table 3 Experimental result GA and REM

| Problem   | Best | Worst | Average | Variance | computation time |
|-----------|------|-------|---------|----------|------------------|
| abz5(GA)  | 4316 | 5005  | 4627.4  | 132.25   | 4.55(s)          |
| abz5(REM) | 4929 | 5671  | 5252.1  | 107.39   | 14.31(s)         |
| abz9(GA)  | 3673 | 4473  | 3920.2  | 91.45    | 241.20(s)        |
| abz9(REM) | 6750 | 7468  | 7168.1  | 72.50    | 300.00(s)        |

図8は縦軸に機械、横軸に時間を示し、ランダムにジョブを発生したabz9のガントチャートである。また、図9は、同問題に多重遺伝子を用いたGAで求めたガントチャートである。両図からは、GAで求めたスケジュールが約500時間短縮されて良好な結果が得られた。

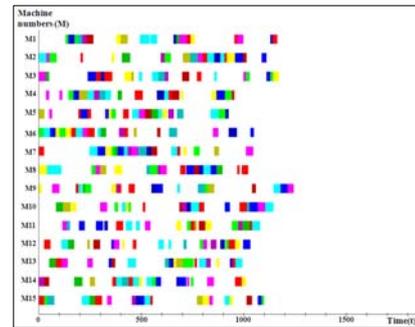


Fig.8 Gantt chart has generated the solution to random

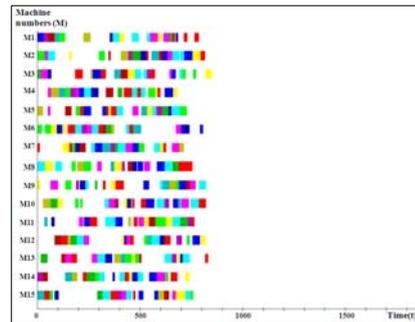


Fig.9 Gantt chart has generated the solution to GA

## 7. 結言

本研究は、組合せ最適化問題の中でも最適解が得ることが困難とされているジョブショップスケジューリング問題の多目的関数の提案を行った。その結果、解法として多重遺伝子による遺伝的アルゴリズムを採用し、以下の事が得られた。

- 1) 多重遺伝子を用いたGAは、REMとの比較から提案手法の有効性を検証した。
- 2) 今回はabz5、abz9のジョブショップスケジューリング問題を取り扱い、その他の大規模スケジューリング問題にも適用できることを確認した。
- 3) 最大滞留時間の最小化に加えて納期制限の制約条件を加えてスケジューリングを行った結果、目的に合ったスケジュール作成が可能であることを確認した。

今後の課題としては、SOMによるスケジュール結果の可視化や、実際の企業のスケジュールの実データを用いて有効性を検証することである。また、主要な探索手段である交叉や突然変異の遺伝オペレータ改善を取り扱う予定である。

## 文 献

- [1] 渡辺美知子, 古川正志, 嘉数侑昇, GAによるジョブショップスケジューリングの多目的最適化に関する研究, 2001年度精密工学会秋期大会学術講演会講演論文集, 66(8) 1185-1190, 2001
- [2] 古川正志, 川上敬, 渡辺美知子, 木下正博, 山本雅人, 鈴木育男, メタヒューリスティクスとナチュラルコンピューティング, コロナ社, 2012
- [3] 三宮信夫, 喜多一, 玉置久, 岩本貴司, 遺伝的アルゴリズムと最適化, 朝倉書店, 1998
- [4] T.コホネン, 自己組織化マップ, シュプリンガー・フェアラーク東京株式会社, 1996