

ジョブの投入順序の変更量を考慮した再スケジューリング手法の提案

1X21C034-4 鬼頭拓海
指導教員 谷水義隆

1. はじめに

製造業の生産現場では作業の遅延や機械の故障などの予期せぬ外乱が発生する[1]。これらの外乱に対応するスケジューリングのアプローチは大きく2つに分類される。1つは初期スケジュールを外乱に対して耐性の高い、すなわちロバスト性の高いスケジュールにするという静的アプローチであり、もう1つは外乱の発生に応じて生産の途中で再スケジューリングを行う動的アプローチである。実際の製造現場では影響の大きな外乱が発生した場合、初期スケジュールに従って生産を進めるのは困難である。そのため、本研究では後者の動的アプローチに焦点を当てる。

再スケジューリングを行ってスケジュールを変更すると現場の混乱やリソースの再配置コスト、原材料の再注文コストなどが生じる[1]。これらのスケジュール変更コストは製造現場にとっても企業にとっても望ましくないものである。スケジュール変更コストを小さくするためには、元のスケジュールから大幅に変更しすぎないスケジュールを作成することが求められる。

そこで本研究では製造業の生産現場において外乱が発生した際に、元のスケジュールからの変更の度合いを抑制しながら再スケジューリングを行う手法を提案する。

2. 再スケジューリングにおける安定性

2.1. 既存研究

スケジューリングの評価関数として、スケジュールの効率性を評価する総所要時間や納期遅れ和などを用いた先行研究は多く存在する。しかし再スケジューリングにおいて、元のスケジュールからの変更度合いを表す評価関数である安定性についての研究は少ない。安定性という用語はロバスト性を意味する場合にも使用されるが、本研究では前述の変更度合いの意味で用いる。

Wu ら [2] は安定性の評価指標として元のスケジュールと修正されたスケジュールの各ジョブの作業開始時刻の絶対値の差の総和を提案した。これは開始時刻偏差と呼ばれる。この評価指標はこれまでのほとんどの研究において安定性の指標として用いられている。

Ruedee ら [3] は再スケジューリング時刻からの時間的距離が近いジョブほどスケジュール変更コストが大きいという事実を指摘し、開始時刻が早まったジョブに対して、開始時刻からの時間的距離に応じたペナルティを開始時刻偏差に乗算した関数を安定性の指標として提案した。

2.2. 本研究における安定性の評価

既存の研究は一般的に開始時刻偏差を基準にした関数を安定性の指標として用いている。しかし実際の作業現場ではジョブの開始時刻の変化だけではなく、ジョブの作業順序の違いも同様にスケジュール変更コストの重要な要因と考えられる。開始時刻偏差は時間的な影響の大きい外乱が発生した場合、例えば機械の故障などが発生した際に、

作業順序を正しく評価できない場合があることを実験によって確認した。

そこで本研究では安定性の評価指標として開始時刻偏差ではなく、ジョブの投入順序に着目し、順位偏差を基準にした新たな評価関数を提案する。順位偏差は各リソースのジョブの投入順序を順位として評価し、再スケジューリング前後の各ジョブの順位変更量の総和として定義する。本研究では投入順序が早いジョブを順位が高いジョブと表現する。また Ruedee らが指摘した時間的距離によるスケジュール変更コストの違いを考慮して、順位的距離に応じた順位ペナルティを順位偏差に乗算した関数を安定性関数の評価指標とする。

3. 提案手法

3.1. ジョブショップスケジューリング問題の解法

ジョブショップスケジューリング問題に対して、遺伝的アルゴリズム (GA) によって求解を行う。コーディング手法は Giffler-Thompson 法 (GT 法)、遺伝的操作に関しては、選択はトーナメント選択、交叉は Partially Mapped Crossover、突然変異は逆位を用いる。

3.2. 再スケジューリング手法

本研究では外乱の中でもジョブの終了時刻の遅延に焦点を当てる。遅延したジョブの終了時刻を x 分として、 $x + \Delta t$ 分を再スケジューリング開始時刻とする。その時点で作業がまだ始まっていないジョブを再スケジューリング対象のジョブとしてコーディングを行い、遺伝子を作成する。

3.3. 目的関数

安定性の評価関数と効率性の評価関数の二目的最適化を行う。効率性を評価する関数には総所要時間、安定性を評価する関数には式(1)を用いる。

$$D(S_p, S_q) = \sum_{i=1}^M \sum_{j=1}^J \omega_j |r_{i,j}^p - r_{i,j}^q| \quad (1)$$

$$\omega_j = \frac{1}{(r_{i,j}^q)^\beta} \quad (2)$$

- S_p : 初期スケジュール
- S_q : 再スケジューリング後のスケジュール
- M : リソースの数
- J : ジョブの数
- $r_{i,j}^p$: S_p における機械 i のジョブ j の順位
- $r_{i,j}^q$: S_q における機械 i のジョブ j の順位
- ω_j : ジョブの順位に対するペナルティ関数
- β : ペナルティの傾斜を調整する係数

式(1)はジョブごとに順位変更量とペナルティ関数 ω_j を乗算した値の総和を取っている。この関数の目的は順位変

更量の総和の抑制と順位の高いジョブの順位変更量の抑制の2つである。前者は順位偏差、後者は ω_j によって表現されている。 ω_j は順位の高いジョブほど大きなペナルティを与える関数である。

安定性関数と総所要時間を式(3)の重みパラメータ法によって合成した関数をGAの目的関数として用いて最小化を行う。それぞれの関数はスケールをあわせるためにMin-Max正規化を行う。

$$\lambda \times D(S_1, S_2) + (1 - \lambda) \times C_{max} \quad (3)$$

4. 計算機実験

4.1. 実験条件

10ジョブ、10リソースのジョブショップスケジューリング問題を扱う。初期スケジュールとして総所要時間が1019分のスケジュールを用意し、その中でマシン1のジョブ8の作業の終了時刻を60分遅延させた状況で再スケジューリングを行う。 $\Delta t=1$, $\beta=1.25$ とする。

4.2. 実験1

実験1では安定性関数と総所要時間の重み λ を変化させた際の推移を検証する。結果を図1に示す。

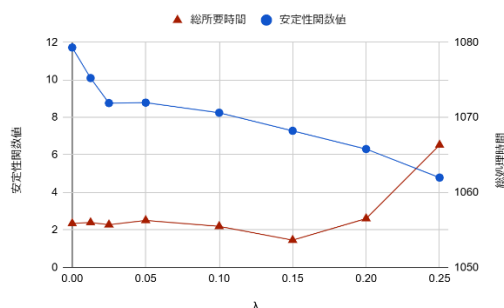


図1 安定性関数値と総所要時間の平均値

各 λ の値はGAを100回行ったうち、解が初期解と同じスケジュールになる場合、すなわち再スケジューリングの結果としてスケジュールを変更しないという解が得られた場合を除いた平均値である。解が初期解になる確率は $\lambda=0$ では0.11であるが、 λ を大きくすると確率は大きくなり、0.3以上にすると1になる。

図1から、 λ を大きくするほど安定性関数値が小さくなり安定性が高まるということがわかる。また総所要時間に関しては、 λ が0.25の場合を除くと λ を大きくしてもほとんど大きくならないということがわかる。

4.3. 実験2

実験2では実際に安定性が正しく確保できているのかを検証するためにジョブの順位変更量に関して比較実験を行う。結果を表1と図2に示す。

表1 目的関数ごとの各指標の比較

目的関数	総所要 時間/分	安定性 関数値	順位変更 量の合計
総所要時間 のみ	1055.69	10.09	38.98
総所要時間 +安定性関数	1056.00	6.76	20.36

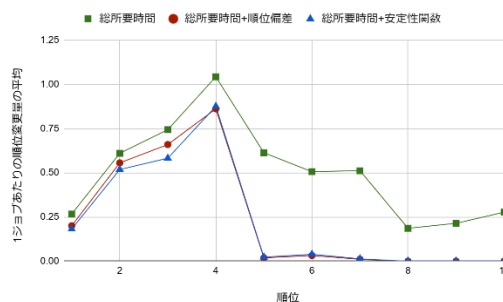


図2 目的関数ごとの各順位における1ジョブあたりの順位変更量の平均の比較

値は実験1と同様にGAを100回行ったうち、解が初期解と同じスケジュールになる場合を除いた結果の平均値である。 $\lambda=0.2$ としている。

表1の順位変更量の合計を見ると、安定性関数によって順位変更量を約半分に抑えることができていることがわかる。よって安定性関数の1つ目の目的である順位変更量の総和の抑制が達成されていることが確認できる。

図2において、目的関数が総処理時間+順位偏差である結果と目的関数が総処理時間+安定性関数である結果を比較すると、安定性関数を加える方が順位の高いジョブの順位変更量を若干量ではあるものの、抑制できていることがわかる。よって安定性関数の2つ目の目的である順位の高いジョブの順位変更量の抑制が達成されていることが確認できる。

5. おわりに

本研究ではジョブショップスケジューリング問題を対象として、作業の終了時刻の遅延が発生した際に今回新たに提案した安定性関数と総処理時間の二目的最適化をGAによって求解する手法を提案した。実験1によって効率性をほとんど損ねることなく安定性関数値を改善できることを確認し、実験2によって安定性関数の目的が達成できていることを確認した。本手法によって安定これにより、性を確保した、すなわち元のスケジュールからの変更度合いを抑制した再スケジューリングを行うことができることを示した。今後は本手法に最適化された遺伝的操作の開発やシミュレーションによる検証についても研究を行う必要がある。

参考文献

- [1] Tighazoui, A., Sauvey, C., & Sauer, N. (2021). Predictive-reactive strategy for identical parallel machine rescheduling. *Computers & Operations Research*, 134, 105372.
- [2] Wu, S. D., Storer, R. H., & Chang, P. C. (1993). One-machine rescheduling heuristics with efficiency and stability as criteria. *Computers & Operations Research*, 20(1), 1-14.
- [3] Rangsaritratamee, R., Ferrell Jr., W. G., & Kurz, M. B. (2004). Dynamic rescheduling that simultaneously considers efficiency and stability. *Computers & Industrial Engineering*, 46(1), 1-15.