

粒子群最適化と近傍探索を融合した ジョブショップスケジューリング問題の一解法

片岡 隆之, 金指 正和

A Solution Method for Job Shop Scheduling Problem using Particle Swarm Optimization and Local Search

Takayuki KATAOKA and Masakazu KANEZASHI

Synopsis

The job-shop scheduling problem (JSSP) is a branch of production scheduling, and it is well known that this problem is NP-hard. Many different approaches have already been applied to JSSP. However, some JSSP cannot be solved to guarantee optimality. This paper shows a similar algorithm using PSO and Local search to solve JSSP. At the same time, some heuristic valid algorithm operations are proposed to minimize running time or *makespan* in this paper.

Key words: Job Shop Scheduling Problem, Particle Swarm Optimization, Local Search

1. はじめに

生産スケジューリングの分野であるジョブショップスケジューリング問題(Job Shop Scheduling Problem: 以下 JSSP)は, 製造工程における複数の仕事と複数の機械の組合せ順序を考え, 製品完成の総所要時間などの最適化を目指す組合せ最適化問題である。ただし, 仕事と機械の数が増加するにつれ, 組合せの数も増加するため, 最適化が非常に困難となる[1]。そこで, 本研究でも扱う粒子群最適化(Particle Swarm Optimization: 以下 PSO)などの最適化ツールに注目した近似解最適化手法の研究が数多くされている。近年では, PSOに遺伝的アルゴリズムの機能を導入した混成粒子群最適化を用いて, 一般的な遺伝的アルゴリ

ズムよりも短時間でより良質な解の探索ができるアルゴリズムの提案などが行われている[2]。しかし, 現実的な時間に最適解を保障するアルゴリズムはまだ確立されていないのが現状である。本研究では, 計算処理工程やパラメータ設定が少なく単純な近傍探索(Local Search: 以下 LS)とPSOに注目した最適化アルゴリズムを提案し, 短時間で良質な解の探索を目的とする。

2. JSSP

2.1 JSSP の定義と前提条件

JSSP には, 複数の仕事と複数の機械が存在する。仕事には, その仕事の完了に必要なとされる作業が処理順序制約

付きで設定されている。また、その他の制約条件も存在し、これらを満たした上ですべての仕事が完了する時間(総所要時間)の最小化などを目的としたものがJSSPである。

本研究では、以下の前提条件に基づく一般JSSPを用いて、総所要時間とスケジューリング時間の最小化を評価基準として取り扱う。

[前提条件]

1. 各仕事には、機械の台数分だけ作業が存在し、各機械で必ず一度は処理される。
2. 1つの仕事で同時に複数の作業を処理できない。
3. 1つの機械で同時に複数の作業を処理できない。
4. 各作業には、作業完了に必要とする作業時間が決定されており、0以上の整数値(非負整数)とする。
5. 機械の故障時間や作業間の運搬時間などは考えず、連続的に作業が行えるものとする。
6. 開始時刻を0とし、すべての仕事が完了した時点を経済所要時間とする。

この前提条件を踏まえ、JSSPの例を表1に示す。

表1. 3ジョブ3マシン問題の例

| | 作業1 | 作業2 | 作業3 |
|-----|--------|--------|---------|
| 仕事1 | [3, 1] | [1, 3] | [2, 6] |
| 仕事2 | [2, 8] | [3, 5] | [1, 10] |
| 仕事3 | [3, 5] | [1, 9] | [2, 1] |

表1は3ジョブ3マシン問題の例であり、各仕事に対する作業内容(作業機械番号, 作業時間)と作業の処理順序を示している。例えば、仕事1の作業1では、機械3で作業時間1の作業を行う。また、この問題に対する実行可能スケジュールの一例を図1に示す。

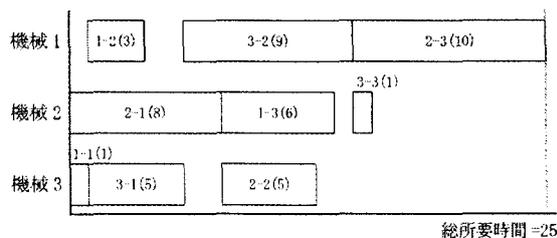


図1. 3ジョブ3マシン問題の実行可能スケジュール例

図1は、実行可能スケジュールをガントチャート形式で表したもので、横軸を時間、縦軸を機械とした作業の振分けを示したグラフである。例えば、1-2(3)では、仕事1の作業2で作業時間3を表している。また、制約条件を満たしたスケジュールを実行可能スケジュールとする。

2.2 スケジュールの表現

JSSPのスケジュールである解(作業順序)を表現する必要がある。本研究で用いるスケジュール表現は次を例とする表現形式を用いる。

[2, 1, 1, 3, 3, 1, 2, 2, 3]

これは、図1のスケジュールを表現したものである。左側を先頭とし、表記されている数字は仕事番号を表している。また、先頭から順に同じ仕事番号に当たるたびに作業が進むものになっている。例えば、上記の表現であれば、先頭から順に「仕事2の作業1」、「仕事1の作業1」、「仕事1の作業2」、「仕事3の作業1」、「仕事3の作業2」、「仕事1の作業3」、「仕事2の作業2」、「仕事2の作業3」、「仕事3の作業3」という並びになっている。つまり、 m ジョブ n マシン問題の場合、仕事 i は n 個表記されることになり、 $m \times n$ 個の数字で表現する。

3. LS と PSO

3.1 LS

LSは、総所要時間の最小化などの探索条件を設定した上で、問題に対しての解に変形を加えた近傍解を生成する。近傍解が、探索条件を満たしていれば近傍解を新しい解に、満たさなければ別の近傍解で探索するか、探索を終了する最適化ツールである。ただし、近傍解の基となる解によっては、すぐに探索限界に達してしまう場合がある。

3.2 PSO

PSOは、鳥や魚などの生物が群を成して餌場などの目的地を目指す群知能をモチーフにした最適化ツールである。問題を多次元空間上にモデル化し、その中を粒子群が探索を行い、解の座標情報を粒子同士で共有することにより、目標の探索を行いやすくしている。ただし、制御を誤れば、粒子群が発散または収束してしまう場合がある。

4. アルゴリズムの提案

4.1 LS と PSO の融合

本研究では、LSとPSOを組み合わせたアルゴリズム(以下LS+PSOアルゴリズム)を提案する。粒子群によるLSを行い、LSの欠点である探索限界にある粒子をPSOで方向転換させることで補う。これにより、LSのみのアルゴリズム(以下LSアルゴリズム)と比較し、粒子群全体でより良質な解の探索を行う。

4.2 LS+PSOアルゴリズム

提案するLS+PSOアルゴリズムについて、簡単なフローを図2に示す。

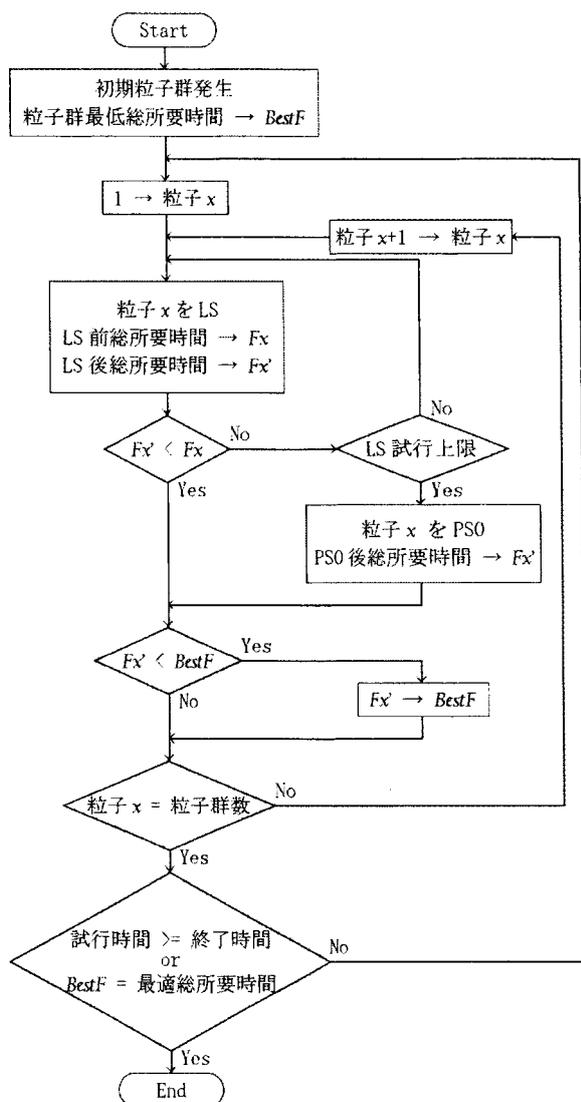


図2. LS+PSO アルゴリズム フローチャート

[アルゴリズムフロー]

- Step.1 初期粒子群をランダムで発生させる。
- Step.2 初期粒子群の中から最小の総所要時間 ($BestF$)を持つ粒子の情報を記録する。同時に、これを PSO の共有情報として扱う。
- Step.3 粒子番号1(粒子 x)から順にLSを行う。
- Step.4 LS 前の総所要時間(F_x)とLS後の総所要時間($F_{x'}$)を比較し、総所要時間が短縮($F_{x'} < F_x$)されていればLSを終了(LS成功)し、次の粒子の探索に進む。短縮されていなければStep.2に戻り、再度LSを行う(LS失敗)。ただし、LS失敗が続き、LSリミットに達した粒子はLSを終了する。

- Step.5 LSリミットに達した粒子(総所要時間の短縮が見込まれない解)は、粒子群中で最小の総所要時間を記録している座標の情報を利用しPSOを行う。
- Step.6 LSまたはPSOによって粒子群中で今までより小さい総所要時間を持った座標が探索された場合は、その座標情報を記録し、PSOに反映させる。
- Step.7 Step.2～Step.6の操作を全粒子で繰り返し、全粒子の探索が終了した場合は、また粒子1から順にさらなる探索を繰り返す。
- Step.8 スケジューリング時間に達したまたは最適解が求まった場合、そこですべての探索を終了する。

4.3 独自LS

本研究のアルゴリズムに組み込まれている独自のLS(以下 独自LS)について述べる。独自LSでは、近傍解生成に2種の方式を採用している。

- 1 種目は、作業順序配列の中からランダムに1作業を選択し、ランダムな位置に挿入する方式(以下 移動法)である。
- 2 種目は、作業順序配列の中からランダムに2作業を選択し、互いの位置を交換する方式(以下 交換法)である。

今回示した移動法、交換法で生成された近傍解は、総所要時間が短縮される。しかし、近傍解次第では総所要時間が増加する可能性もある。

独自LSでは、移動法および交換法の2種を用いて近傍解の生成を行う。LS+PSOアルゴリズム中では、粒子に対して2種の方法をランダムに試行する。また、一定回数以上試行しても、粒子の総所要時間が短縮されない場合は、次に示す独自PSOを試行する。

4.4 独自PSO

本研究では、粒子の発散を防ぐ措置として、次の4条件を実験条件と評価基準に基づき検証し、有効な条件を選択し実験を行った。この結果に基づき、本研究では、条件4に合致する粒子のみをPSOの対象とした。

[条件]

1. PSOを行わない
2. $if \ countLS_i > \ limitLS \ then \ PSO$
3. $if \ countLS_i > \ limitLS$
 $and \ F_i' \geq \ averageF \ then \ PSO$
4. $if \ countLS_i > \ limitLS \ and \ F_i \neq \ bestF$
 $and \ F_i' \geq \ averageF \ then \ PSO_i$

[実験条件]

1. 粒子群は20粒子とする。
2. 1粒子のLS試行上限回数を500回とする。
3. 1回の最大スケジューリング時間は300秒とし、300秒経過した時点で最も良い解とそれを求めるのに要したスケジューリング時間を記録する。ただし、最大スケジューリング時間内に最適解が探索できた場合は、300秒経過するのを待たず探索を終了する。
4. スケジューリング実行回数は300回とする。
5. 実験CPUはIntel社 Core 2 Duo E6300 (1.86GHz)を使用する。

[評価基準]

総所要時間およびスケジューリング時間の最小化

5. 比較と結果

5.1 プログラムの仕様

アルゴリズムを実装したプログラムを用いて実験する。プログラム化を行う言語は、計算部にFortran90/95、出力部にCを使用している。そして、コンパイラに富士通社 Fujitsu Workbench 開発マネージャ(v4.0L10)を使用し、コンパイルおよび最適化する。また、モンテカルロ法を適用し、完全一様乱数を用いる。ここで用いる完全一様乱数とは、L'Ecuyerの長周期(2×10^{18} 以上)乱数生成法とBays-Durhamの切り混ぜ法を用いた乱数とする。

5.2 アルゴリズムの比較

LSとPSOを組み合わせることで、LSの欠点である探索限界の克服ができていないかを評価するために、LSアルゴリズムとLS+PSOアルゴリズムをJSSPベンチマークで比較する。共通の実行条件は次のように設定する。

[実行条件]

1. 粒子群は10粒子とする。
2. 1粒子のLS試行上限回数を500回とする。
3. 1回のスケジューリング時間は300秒とし、300秒経過した時点での最も良質な解とそれを求めるのに要したスケジューリング時間を記録する。
4. スケジューリング実行回数は300回とする。
5. CPUはIntel社 Core 2 Duo E6300(1.86GHz)を用いる。

[評価基準]

総所要時間およびスケジューリング時間の最小化

[評価方法]

評価方法は、評価基準に基づいて「最適化率」、「平均総所要時間」、「最良総所要時間」の順に評価を行い、「最適化率」の高いアルゴリズム、「平均総所要時間」、「最良総所要時間」の低いアルゴリズムを優位とする。

5.3 実行結果

Fisher and Thompsonの10ジョブ10マシン問題(ft10)と20ジョブ5マシン問題(ft20)の2種で比較する。以下の表は、平均総所要時間と平均スケジューリング時間(平均)、最も良質な総所要時間とスケジューリング時間(最低)、最適解に到達できた回数(最適化)を示している。

表2. ft10 ベンチマーク

| 最適総所要時間：930 | | | |
|-------------|----------------|-------------|---------|
| | 平均 | 最低 | 最適化 |
| LS | 945.93/172.24秒 | 930/175.53秒 | 2/300回 |
| LS+PSO | 941.17/148.74秒 | 930/32.78秒 | 66/300回 |

表3. ft20 ベンチマーク

| 最適総所要時間：1165 | | | |
|--------------|-----------------|--------------|--------|
| | 平均 | 最低 | 最適化 |
| LS | 1180.29/164.53秒 | 1173/26.75秒 | 0/300回 |
| LS+PSO | 1180.19/142.22秒 | 1165/132.05秒 | 4/300回 |

表2と表3より、提案法(LS+PSO)が有効であるといえる。ft10では、時間内の最適化率に優れており、最適解到達スケジューリング時間も短くなっている。ft20では、平均総所要時間に大差は見られないが、時間内に最適解に到達できている。また、本稿には記載していないが、Adams, Balas and Zawackの問題(abz5~abz9)やLawrenceの問題(la01~la40)の多くでも提案法の有効性が示している。

6. おわりに

本研究では、JSSPに対する最適化アルゴリズムとしてLS+PSOアルゴリズムの提案を行った。提案法は、同一条件下でのLSアルゴリズムと比較して、短いスケジューリング時間でより良質な解の探索を行うことができた。

今後の研究課題として、粒子数やLS試行上限回数などの各パラメータの最適化や新たなLS方法の導入を行い、さらなる改善をしていくことが挙げられる。

最後に、本研究に貢献してくれた岸本直也君と丸尾将嘉君に感謝したい。

参考文献

- [1]黒田 充, 村松健児編, 「生産スケジューリング」, 朝倉書店, 2002年.
- [2]Zhigang Lian, Bin Jiao, Kingsheng Gu: A similar particle swarm optimization algorithm for job-shop scheduling to minimize makespan, Applied Mathematics and Computation, Vol.183, pp.1008-1017, 2006.