

## ジョブショップ・スケジューリング問題のPSO的解法 —連続型PSOによる方法との比較—

酪農学園大 ○高取則彦 旭川高専 古川正志, 渡辺美知子 函館高専 浜克己  
北海道工業大 川上敬, 木下正博 会津大 成瀬継太郎

### 要 旨

PSOは、動物や人間の集団的行動に着想を得た最適化手法である。前回、ジョブショップ・スケジューリング問題に関して、2値変数型PSOをベースにした解法を提案したが、今回はその改良を試みる。最近、SPVルールという、連続型PSOを順序付け問題に適用する方法が提案されている。これら2つの方法を比較した結果、提案手法の方がSPVルールによる方法より同程度の質の解を速く得られることがわかった。

### 1. はじめに

PSO (Particle Swarm Optimization) は、動物や人間の集団的行動に着想を得た、最適化問題に対するメタヒューリスティクスの1つである[1]。集団に属する個体が取れる行動を観察して得られた知見を、問題の解法に取り入れている。

前回、2値変数型PSOをベースにしたジョブショップ・スケジューリング問題の解法について報告した[2]。この問題は、規模が大きくなると最適解を求めることは事実上不可能になるため、準最適解を効率的に求める試みがなされているものである。

最近、SPV (smallest position value) ルールという、連続型PSOアルゴリズムを順序付け問題に適用する方法が提案され、良好な結果が報告されている。

ここでは、前回報告した解法の改良を試み、これらの方法を比較した計算機実験の結果を報告する。

### 2. PSO

この方法を提案した Kennedy らは、社会学的考察から、集団における個体は

- 評価する (evaluate)
- 比較する (compare)
- 模倣する (imitate)

という3つの原理にもとづいて行動するとし、これらを最適化問題の解法に取り入れた。すなわち、個体を解候補と考え、各個体は自分の状態を評価し他の個体と比較する。そして、自分よりよい個体と、記憶している自分の最良状態とを模倣することにより、状態の改善を試みる。

個体間にはあらかじめ「つながり」が定められ、直接情報交換ができるのはつながりのあるものに限定される。そのため、模倣する相手はつながりのある個体から選ばれる。つながり方には  $g_{best}$  と  $l_{best}$  の2種類がある。 $g_{best}$  は各個体が集団中最良の個体とつながりを持つものである。一方  $l_{best}$  は個体  $i$  が個体  $i-c, i-c+1, \dots, i+c-1, i+c$  とつながりを持つもので、 $c$  の値によりつながりの範囲が決められる。以下では  $l_{best}$  を用いることにする。

PSOでは、個体  $i$  は組  $(x_i, v_i)$  によって指定される。 $x_i$  は位置、 $v_i$  は速度と呼ばれる  $n$  次元ベクトルであり、 $x_i$  の成分が連続変数の場合[1]と2値変数の場合[3]とがある。

$x_i$  の成分が連続変数の場合、 $v_i$  の各成分  $v_{ik}$  ( $k=1, 2, \dots,$

$n$ ) は、次式により更新される。

$$v_{ik} \leftarrow v_{ik} + r_1 \cdot (x_{ik}^{best} - x_{ik}) + r_2 \cdot (x_{ik}^{local} - x_{ik}) \quad \dots(1)$$

ここで、 $x_i^{best}$  は個体  $i$  のこれまでで最良の位置、 $x_i^{local}$  は個体  $i$  とつながりのある中で最良の個体の位置である。 $r_1, r_2$  は一様乱数で、 $r_1, r_2 \in [0, R]$  ( $R > 0$ ) である。 $v_{ik}$  があらかじめ定められた範囲  $[-V, V]$  ( $V > 0$ ) を超えた場合は、この範囲内に収める。そして  $x_i$  の各成分  $x_{ik}$  ( $k=1, 2, \dots, n$ ) は

$$x_{ik} \leftarrow x_{ik} + v_{ik} \quad \dots(2)$$

により更新される。これらの処理を各個体の各成分について、終了条件を満たすまで繰り返すことにより、最適解の探索が行われる。

$x_i$  の成分が2値変数の場合、 $x_{ik}$  は式(2)ではなく

$$\text{if } \rho < g(v_{ik}) \text{ then } x_{ik} = 1 \text{ else } x_{ik} = 0 \quad \dots(3)$$

のように更新される。ここで  $g(v_{ik})$  はシグモイド関数

$$g(v_{ik}) = \frac{1}{1 + \exp(-v_{ik})} \quad \dots(4)$$

で、 $\rho \in [0, 1]$  の一様乱数である。このように、 $v_{ik}$  は  $x_{ik} = 1$  とする確率を与えるものとして計算に用いられる。

### 3. スケジューリング問題の解法

#### 3.1 解の表現

ここでは、解すなわちスケジュールを、ジョブ番号の重複順列により表現する。この表現では、各ジョブの番号はそれが持つタスク数だけ繰り返し現れる。 $m$  回目に現れたジョブ番号は、そのジョブの  $m$  番目のタスクを所定の機械に割り付けることと解釈して、スケジュールに変換される。

#### 3.2 提案する解法

前回同様、2値変数のPSOを基本に考えるが、模倣の処理を次のように変更する。

**模倣のしかた** 部分列すなわち1つ以上の連続した要素を模倣の単位とする。各個体は、模倣相手の部分列をそれと同じ位置に複写する。残りの部分は、現在の要素を先頭から順に複写する。そのとき、各ジョブ番号がそのタスク数を超えて現れないようにして、実行不能解を生じないようにする。

**模倣相手の決定** 個体  $i$  は、 $x_i^{best}$  と  $x_i^{local}$  のいずれか

を確率的に選んで模倣相手とする。標準的なPSOでは速度はベクトルとしているが、ここではスカラー $u_i$ を用いることにし、これを次式により更新する。

$$u_i \leftarrow u_i + r \cdot h(x_i, x_i^{\text{best}}, x_i^{\text{local}}) \quad \dots(5)$$

ここで、 $r$ は $r \in [0, R]$  ( $R > 0$ ) の一様乱数である。関数 $h(x_i, x_i^{\text{best}}, x_i^{\text{local}})$ は、個体の評価関数 $f(x_i)$ を用いて次のように定義する。

$$h(x_i, x_i^{\text{best}}, x_i^{\text{local}}) = \begin{cases} 1; & \text{if } (f(x_i) \geq f(x_i^{\text{best}}) \text{ or } f(x_i) \geq f(x_i^{\text{local}})) \\ & \text{and } f(x_i^{\text{best}}) < f(x_i^{\text{local}}) \\ -1; & \text{if } (f(x_i) \geq f(x_i^{\text{best}}) \text{ or } f(x_i) \geq f(x_i^{\text{local}})) \\ & \text{and } f(x_i^{\text{best}}) > f(x_i^{\text{local}}) \\ 0; & \text{otherwise} \end{cases} \quad \dots(6)$$

$u_i$ があらかじめ定められた範囲 $[-V, V]$  ( $V > 0$ ) を超えた場合は、連続型PSOと同様にこの範囲内に収めて、式(4)を用いて $g(u_i)$ を計算する。 $g(u_i)$ を $x_i^{\text{best}}$ を模倣する確率と見なし、一様乱数 $\rho \in [0, 1]$ に対して $\rho < g(u_i)$ ならば $x_i^{\text{best}}$ を、そうでなければ $x_i^{\text{local}}$ を選ぶことにする。

### 3.3 SPVルール

Tasgetirenらは、SPVというヒューリスティックルールを提案し、連続型PSOをさまざまな順序付け問題に適用することを試みている[4]。これまでは主にフローショップ・スケジューリング問題を扱っており、良好な結果が得られたとしている。

SPVルールは、個体 $i$ の位置 $x_i$ の各成分 $x_{ik}$ を昇順にソートし、ソート後のインデックスの列を解として用いるというものである。このルールは個体の評価のときにのみ用いられるので、 $v_i, x_i$ は式(1), (2)により更新される。したがって、全体の処理手続きは2で述べた標準的な連続型PSOと同じである。

このルールを用いて連続型PSOによりジョブショップ・スケジューリング問題を解く場合、ソート後のインデックス $k$ の列ではなく $k \text{ Mod } J$  ( $J$ はジョブ数)の列を用いることにより、3.1で述べた表現形式の解が得られる。

## 4. 実験

上述の提案手法についてプログラムを作成し、計算機実験を行った。模倣する部分列の長さは $1 \sim L$  (個体の長さ) からランダムに選び、その位置もランダムに決定した。また、SPVルールによる連続型PSOについてもプログラムを作成し、結果を比較した。動作条件は両者ともに、個体数 $N = 200$ 、繰返し回数10000とし、個体の初期状態はランダムに生成した。

総処理時間 $C_{\text{max}}$ の最小化問題を対象とし、よく知られているベンチマーク問題‘ft10’ (10ジョブ10機械、最適解930)と‘ft20’ (20ジョブ5機械、最適解1165)を問題例として用いた。

式(1)と式(5)の $r, r_1, r_2$ の範囲 $[0, R]$ と $u_i, v_{ik}$ の限界値 $V$ を変え、さらにつながりの範囲 $c$ を変えて、それぞれの条件について10回ずつ実行した。表1にその結果を示す。ft10では条件によってはSPVの方がよいが、その差は小さい。ft20の場合はすべての条件で提案手

表1 実行結果

(1) ft10

		c = 1		c = 2		c = 3	
		SPV	提案手法	SPV	提案手法	SPV	提案手法
R=1.0 V=2.0	平均	1009.6	1006.7	996.6	999.3	1003.8	997.5
	最小値	991	973	971	963	967	965
	最大値	1028	1036	1026	1037	1028	1033
R=2.0 V=2.0	平均	1004.5	1010.0	1006.7	1013.9	991.6	1004.8
	最小値	984	982	990	973	958	958
	最大値	1023	1045	1028	1045	1010	1059
R=1.0 V=4.0	平均	1006.9	1017.4	1002.5	1008.7	1004.2	1012.3
	最小値	989	985	979	977	973	988
	最大値	1034	1049	1015	1037	1025	1038
R=2.0 V=4.0	平均	1003.6	1011.9	996.9	999.5	996.1	1024.4
	最小値	988	991	958	959	979	989
	最大値	1027	1034	1012	1046	1022	1077

(2) ft20

		c = 1		c = 2		c = 3	
		SPV	提案手法	SPV	提案手法	SPV	提案手法
R=1.0 V=2.0	平均	1284.7	1238.0	1263.4	1229.7	1252.8	1233.5
	最小値	1249	1203	1228	1192	1218	1203
	最大値	1313	1265	1285	1273	1294	1271
R=2.0 V=2.0	平均	1281.8	1237.0	1268.9	1223.8	1252.4	1244.3
	最小値	1259	1210	1253	1190	1216	1201
	最大値	1311	1266	1294	1269	1280	1291
R=1.0 V=4.0	平均	1274.5	1245.0	1272.0	1239.5	1255.1	1244.6
	最小値	1248	1208	1251	1207	1232	1207
	最大値	1313	1280	1311	1306	1276	1303
R=2.0 V=4.0	平均	1271.6	1250.6	1266.7	1246.7	1257.5	1228.0
	最小値	1245	1207	1244	1204	1228	1195
	最大値	1301	1287	1300	1283	1285	1273

法の方が上回っている。

また、SPVによる方法と提案手法との処理時間の比は1:0.2であった。SPVでは、個体の評価の際にソートが必要になるため、処理時間が増加したと考えられる。

## 5. おわりに

2値変数型PSOを基本にしたジョブショップ・スケジューリング問題の解法を提案した。計算機実験により、この解法とSPVルールを用いた連続型PSOとを比較した。その結果、提案手法の方が同程度の解をより高速に得られることがわかった。

## 参考文献

- [1] J. Kennedy and R. Eberhart, Particle Swarm Optimization, Proc. of the IEEE International Conf. on Neural Networks IV (ICNN 95), 1942-1948 (1995).
- [2] 高取 他, ジョブショップ・スケジューリング問題のPSO的解法, 2004年度精密工学会北海道支部学術講演会, 75-76 (2004).
- [3] J. Kennedy and R. Eberhart, A Discrete Binary Version of the Particle Swarm Algorithm, Proc. of the 1997 Conf. on Systems, Man, and Cybernetics, 4104-4108 (1997).
- [4] M. F. Tasgetiren, et al., Particle Swarm Optimization Algorithm for Single Machine Total Weighted Tardiness Problem, Proc. of the 2004 Congress on Evolutionary Computation (CEC 2004), 1412-1419 (2004).