

# 一种新的求解置换 Flowshop 问题的粒子群算法<sup>①</sup>

## A Novel Particle Swarm Optimization Algorithm for Permutation Flowshop Problem

任红燕 张文国 ( 沈阳化工学院计算机科学与技术学院 辽宁沈阳 110142 )

**摘要:** 提出了一种通用的基于位置排序的粒子群算法(PSMPSO)并应用于置换 Flowshop 问题。采用三维粒子表示法,通过对粒子位置排序生成调度方案,将实数编码的粒子位置映射到自然数序列,采用基于粒子位置互换的局部搜索策略来提高算法收敛精度。仿真结果显示了该算法的可行性和有效性。

**关键词:** 置换 Flowshop 位置排序 粒子群算法 粒子表示

### 1 引言

置换 Flowshop 问题(Permutation Flowshop Scheduling Problem, PFSP)广泛应用于实际生产,尤其适用于单件大批量生产背景的制造企业,对流水车间生产的优化调度,可有效提高企业设备利用率和生产效益。然而,PFSP 是一个经典的 NP-hard 问题,难以求得最优解,对其进行理论研究和工程应用仍是当前的一个热点。

粒子群算法(Particle Swarm Optimization, PSO)是一个基于群智能的优化算法,该算法主要用于解决连续空间的数值优化问题,效果显著<sup>[1]</sup>。然而,将粒子群算法用于解决 PFSP 的研究并不多,文献<sup>[2]</sup>引入交换子和交换序的概念,运用实数构造了一种特殊的微粒编码,文献<sup>[3]</sup>结合遗传算法和模拟退火算法的思想,采用混合粒子群算法进行求解,文献<sup>[4]</sup>结合遗传算法的思想定义了多种交叉和变异算子,重新设计了粒子、速度和计算公式。以上三种策略都重新定义了粒子和运算符,改变了原有的定义,不再是简单的数值计算,同时单就某一个问题的编码适用范围太小,没有通用性。

考虑能否不改变粒子群算法原有定义,在连续空间和离散空间建立一种映射,直接将其在连续空间的优良性能应用到离散的数值空间,提出了一种新的基于位置排序的粒子群算法,基本思想是采用三维粒子表示法,通过对粒子位置排序生成调度方案,将实数编

码的粒子位置映射到自然数序列,针对粒子群算法易陷入局部最优、早熟收敛的缺点,加入了合适的局部搜索策略。

首先介绍了置换 Flowshop 问题和标准粒子群算法,然后阐述了基于位置排序的粒子群算法以及在置换 Flowshop 问题上的应用,最后进行了仿真实验,实验表明该算法在求解置换 Flowshop 问题上是可行的和有效的。

### 2 置换 Flowshop 问题和标准粒子群算法

#### 2.1 置换 Flowshop 问题

工件在机器上的加工称为操作,工件基于加工工艺所决定的限制在某些机器上的加工顺序为约束条件。设定每个操作所需要的加工时间是确定的,每一时刻每台机器只能加工一个工件,且每个工件只能被一台机器所加工,加工过程不间断,机器间缓冲区容量为无限<sup>[5]</sup>。PFSP 中每个工件均具有相同的工艺路径,且每台机器上加工的工件顺序也相同。

因此,PFSP 主要是解决各工件的加工次序问题,所有工件的任意排序都可以形成调度问题的一个可行解,我们的目标是找出使最大完工时间(Makespan)最小的一个加工次序。其数学模式是:

设  $t_{ij}$  为工件  $i$  在机器  $j$  上的加工时间,  $C_{ij}$  为工件  $i$  在机器  $j$  上的加工完成时间,令  $J = \{J_1, J_2, \dots, J_n\}$  为一

① 基金项目 辽宁省教育厅高等学校科学研究项目(05L345)

个加工排序,则有

$$\begin{aligned} C_{J_1,1} &= t_{J_1,1} \\ C_{J_j,1} &= C_{J_{j-1},1} + t_{J_j,1} \\ C_{J_1,i} &= C_{J_1,i-1} + t_{J_1,i} \end{aligned} \quad (1)$$

$$C_{J_j,i} = \max\{C_{J_j,i-1}, C_{J_{j-1},i}\} + t_{J_j,i}$$

则目标为最小化最大完工时间:

$$\min f = \min\{C_{J_m}\} \quad (2)$$

## 2.2 标准粒子群算法一般描述

假设粒子群中的第  $i$  个粒子在  $n$  维空间中的位置表示为  $n$  维向量  $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$ , 速度  $v_i = (v_{i1}, v_{i2}, \dots, v_{in})$  决定了粒子在搜索空间单次迭代的位移。当前粒子群中的个体最优粒子表示为  $p_i = (p_{i1}, p_{i2}, \dots, p_{in})$ , 全局最优粒子表示为  $g = (g_1, g_2, \dots, g_n)$  粒子群中的所有粒子根据公式 (3) 和 (4) 来更新其速度和位置<sup>[6]</sup>。

$$v_{i+1} = \omega v_i + c_1 \text{rand}() (p_i - x_i) + c_2 \text{rand}() (g - x_i) \quad (3)$$

$$x_{i+1} = x_i + v_{i+1} \quad (4)$$

其中  $\omega$  为惯性权重,用于均衡全局和局部搜索能力,一般是随迭代次数的增加从 0.9 线性递减到 0.4; 速度  $v_i$  一般设定在  $v_{\max}$  和  $v_{\min}$  之间,当超过这个区间时就取上下值,  $\text{rand}()$  为 (0, 1) 之间的随机数;  $c_1$  和  $c_2$  为学习因子,通常  $c_1 = c_2 = 2$ 。

## 3 基于位置排序的粒子群算法

标准粒子群算法是一种属于连续的实数空间域的优化算法,而调度问题的解空间属于离散的整数域空间,如何在两种不同的空间域之间建立一种联系,即提出适当的粒子表示方法,在粒子位置和调度问题的解空间之间建立一种映射关系,是使用粒子群算法求解调度问题时需要解决的关键问题。

粒子群算法在设计粒子表示方法时主要考虑了三个问题。首先,粒子表示方法所对应的调度方案能否映射调度问题的解空间;其次,粒子表示方法是否适用于粒子群算法的位置-速度模型;再者,粒子表示方法与调度方案映射的复杂度。

通过实验和分析发现,在粒子群算法中,每个粒子的位置分量虽然是实数,但不同位置分量之间是有大小之分的,而且各分量的大小随迭代的进行是不断变化的。那么将粒子各位置分量按大小排序后,即可构

成一个自然数向量,这样就映射到调度问题的解空间,也适用于位置-速度模型,解码也很简单。我们把这种方法称为位置排序法( Position Sequence Method, PSM),具体描述如下:

对于搜索空间  $R^n$  中的任意一个粒子  $i$  位置向量  $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$ , 则向量  $x_i$  对应解空间  $S^n$  中唯一的一个排列  $P(x_i) = \{P_{i1}, P_{i2}, \dots, P_{in}\}$ , 其中  $P_{ik} = (k = 1, \dots, n)$  是向量  $x_i$  中按数值大小排列第  $k$  小的分量  $x_{ij}$  所对应的下标号  $j$ , 即对于  $x_{ij}$  有:

$$k = 1 + \left( \sum_{m=1, m \neq j}^n i(x_{im} < x_{ij}) \right) \text{ else } 0, \text{ 即 } P_{ik} = j.$$

位置向量中最小的分量所对应的下标作为  $p_{i1}$ , 次小的下标为  $p_{i2}$ , 依次类推。这样就将连续空间的实数编码映射到了离散空间的自然数编码。

将 PSM 和标准粒子群算法相结合,就形成基于位置排序的粒子群算法( PSMP SO),可以求解置换 Flowshop 调度等组合优化问题。

## 4 PSMP SO 在置换 Flowshop 上的应用原理

### 4.1 粒子的表示

针对置换 Flowshop 调度问题,粒子位置采用三维向量表示,假设工件的总数为  $N$ ,第一维表示工件,以自然数 1, 2, ...,  $N$  标识不同的工件,在迭代过程中始终不变;第二维表示粒子的位置,对粒子位置按照从小到大进行排序(反之亦可);第三维用于记录第二维位置的排序,初值为 1。假设有 5 个工件,那么粒子表示如表 1 所示,其位置在一定范围内随机生成,初始时计数值为 1。

表 1 初始粒子表示

工件	1	2	3	4	5
位置	1.27	0.85	0.66	1.59	1.34
计数	1	1	1	1	1

### 4.2 调度方案的生成

对初始粒子的位置进行排序即生成一个调度方案  $J = \{3, 2, 1, 5, 4\}$ 。

表 2 排序后的粒子表示

工件	1	2	3	4	5
位置	1.27	0.85	0.66	1.59	1.34
计数	3	2	1	5	4

然后对初始粒子,当其速度和位置根据公式(3)和(4)更新并重新排序后,则生成新的调度方案,假设为如表3所示,其调度方案为 $J = \{3, 1, 5, 2, 4\}$ 。再根据公式(1)和(2)计算其最大完工时间,选择每个粒子所经历过的最小的完工时间为个体极值,所有粒子中最小的为全局极值。

表3 迭代更新后的粒子表示

工 件	1	2	3	4	5
位 置	-0.81	1.34	-1.9	5.7	0.62
计 数	2	4	1	5	3

在整个迭代过程中,第一维工件顺序始终不变,第二维位置值每次迭代中不断进行更新,第三维随第二维位置向量值的大小而变化,生成不同的调度方案。这样就将连续的实数空间映射到调度的离散的整数空间,保证了调度的可行性和多样性,也保证了粒子群算法速度-位置的可操作性。

#### 4.3 粒子群模型及参数的选择

在实际设计时我们采用了流行的惯性权重递减的粒子群模型,见公式(3)和(4)所示, $\omega$ 从0.9递减到0.4, $c_1 = c_2 = 2$ ,初始化时取 $[0, 2]$ 之间的随机数,迭代过程中对粒子的位置不加限制,因为在迭代过程中是对位置排序生成调度方案,若超过区间取设定的值,则经常会出现相同的 $x_{ij}$ ,不仅增加了解码的复杂,还会出现不可行调度方案。速度 $v \in [-40, 40]$ ,超过此区间取上下值,速度的取值范围广,会使粒子在较大的范围探索,有利于保持粒子的多样性。

#### 4.4 局部搜索策略

带惯性权重的粒子群算法虽然能够在全局搜索和局部搜索之间取得平衡,但对于复杂的组合优化问题,仍然容易陷入局部最优,从而过早收敛,因此需要采用一定的局部搜索方法。

文献<sup>[7]</sup>中介绍了多种局部搜索,如互换、插入、逆序等。我们采用了基于工件位置的互换操作,效果较好,具体方法是:对每个三维粒子,随机选取两个不同的位置 $j_1$ 和 $j_2$ ,交换其位置向量 $x_{i,j_1}$ 和 $x_{i,j_2}$ ,经排序产生新的调度方案。如对表3所示的调度方案 $J = \{3, 1, 5, 2, 4\}$ ,假设随机选取出的交换位置为 $j_1 = 2, j_2 = 4$ ,交换并重新排序后就生成了新的调度方案为 $J = \{3, 1, 5, 4, 2\}$ ,如表4所示。对粒子局部搜索完毕后,立即确

定全局最优粒子,并对粒子种群进行更新。

表4 交换后的粒子表示

工 件	1	2	3	4	5
位 置	-0.81	5.7	-1.9	1.34	0.62
计 数	2	5	1	4	3

#### 4.5 算具体步骤

Step1:对种群中每个粒子的位置和速度初始化,设定种群数量和最大迭代次数。

Step2:对每个粒子位置向量从小到大排序,生成调度方案,按公式(1)计算最大完工时间,确定个体最优粒子。

Step3:对粒子进行局部搜索,更新个体最优粒子,局部搜索完毕,根据公式(2)确定全局最优粒子。

Step4:根据公式(3)和(4)更新每个粒子的速度和位置。

Step5:迭代是否结束,若是则输出全局最优粒子和最小完工时间,否则转向步骤(2)。

### 5 数值仿真与分析

为验证混合算法的合理性,我们选取被广泛应用的8个Car类置换Flowshop Benchmark问题进行算法的性能研究,并与传统的遗传算法GA和启发式构造方法NEH作了比较<sup>[5]</sup>。实验中设定种群数量为50,最大迭代次数为300,局部搜索次数为3。另外为了验证基于位置排序的粒子群算法在局部搜索上是否有效,我们采用了不使用局部搜索的PSO和使用局部搜索的PSMPSO。各种算法均随机运行20次(运行环境为Pentium4 CPU 2.56GHz,内存256MB),运行结果见表5所示。表中最小偏差是指各种算法20次优化计算所找到的最小完工时间,与该问题目前已知最优解的相对误差,平均偏差是指20次优化计算的最优完工时间的平均值与已知最优解的相对误差。

为了更形象地表示,图1和图2分别给出它们的变化曲线示意图,由仿真结果可见:

(1)PSMPSO算法和PSO算法都具有好的优化质量,对Car类问题均能得到最优解,甚至是100%的最优解,解的质量大大优于NEH和GA。

(2)PSMPSO和PSO算法的平均误差指标很小,这反映了粒子群算法较好的初值鲁棒性。

(3) PSO 算法加入局部搜索后, 优化性能有了进一步的提高, PSMP SO 平均偏差比 PSO 要好, 局部搜索在一定程度上能够避免了粒子群算法的早熟收敛, 使搜索跳出局部最优, 这说明局部搜索的策略是可行和有

效的。从图 2 所示的 PSMP SO 和 PSO 对 Car2 优化的最优完工图中可以看出, 使用了局部搜索的 PSMP SO 收敛速度更快。

表 5 算法性能比较表

问题规模 $n * m$	已知最优解	NEH	GA		PSMP SO			PSO		
		最小	最小	平均	最小	最大	平均	最小	最大	平均
Car1(11 * 5)	7038	0	0	0.27	0	0	0	0	0	0
Car2(13 * 4)	7166	2.93	0	4.07	0	0	0	0	2.93	0.15
Car3(12 * 5)	7312	1.79	1.19	2.95	0	0	0	0	0.74	0.11
Car4(14 * 4)	8003	0.39	0	2.36	0	0	0	0	0	0
Car5(10 * 6)	7720	4.24	0	1.46	0	0.23	0.02	0	1.31	0.18
Car6(8 * 9)	8505	3.62	0	1.86	0	0.76	0.08	0	2.15	0.67
Car7(7 * 7)	6590	6.34	0	1.57	0	0	0	0	0	0
Car8(8 * 8)	8366	1.09	0	2.59	0	0	0	0	0	0

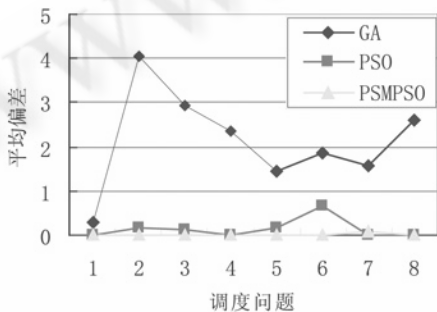


图 1 各种算法优化 Car 类问题的平均偏差

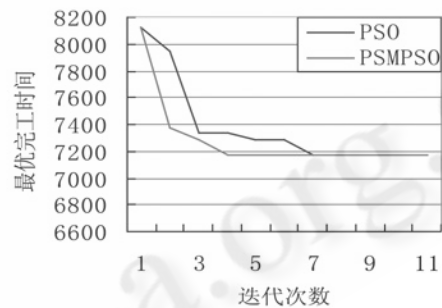


图 2 PSMP SO 和 PSO 优化 Car2 的最优完工图

## 6 结论

粒子群算法在连续空间已表现出良好性能, 通过三维粒子法将连续的实数域空间映射到离散的自然数空间, 改造后的粒子群算法求解置换 Flowshop 问题是成功和有效的。用同样思路, 基于位置排序的粒子群算法可求解旅行商问题、二次分配等组合优化问题。这种方法也对其它基于实数编码的演化算法, 如差分演化算法和思维进化算法应用于组合优化问题提供了一种通用的思路。文章只是针对 Car 类问题进行探讨, 对大规模的调度如何保证质量和效率需要作进一步研究。

## 参考文献

- 1 Kennedy J, Eberhart R. C. Particle Swarm Optimization. In: Proc. of IEEE International Conference on Neural Networks, Perth, Australia, 1995: 1942 - 1948.
- 2 梅慧, 叶春明. 微粒群算法的置换 Flow - Shop 调度问题[J]. 工业工程与管理, 2006, (4): 94 - 96.
- 3 高尚, 杨静宇. 求解流水作业调度问题的混合粒子群优化算法[C]. 中国控制与决策学术年会论文集, 2006: 1006 - 1008.

(下转第 23 页)

(上接第 27 页)

- 4 Zhigang Lian, Xingsheng Gu, Bin Jiao. A novel particle swarm optimization algorithm for permutation flow-shop scheduling to minimize makespan[ DB/OL ]. <http://www.sciencedirect.com>.
- 5 王凌. 智能优化算法及其应用[ M ]. 第 1 版. 北京: 清华大学出版社, 2001.
- 6 Shi Y H, Eberhart R. C. A Modified Particle Swarm Optimizer. In: IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, May 4 - 9, 1998: 69 - 73.
- 7 高尚, 杨静宇. 群智能算法及其应用[ M ]. 第 1 版. 北京: 中国水利水电出版社, 2006.