

文章编号: 1001-0920(2008)02-0191-04

# 解决零空闲流水线调度问题的离散粒子群算法

潘全科<sup>1,2</sup>, 王 凌<sup>2,3</sup>, 赵保华<sup>1</sup>

(1. 中国科学技术大学 信息科学技术学院, 合肥 230026; 2. 聊城大学 计算机学院  
山东 聊城, 252059; 3. 清华大学 自动化系, 北京 100084)

摘 要: 研究了以最大完工时间为目标的零空闲流水线调度问题. 提出一种复杂度为  $O(nm)$  的最大完工时间算法和一种快速插入邻域搜索算法; 提出了解决该问题的离散粒子群调度算法, 并结合简化邻域搜索算法给出了提高调度算法性能的措施. 仿真实验表明了所得算法的有效性.

关键词: 零空闲流水线调度; 粒子群算法; 最大完工时间

中图分类号: TP182

文献标识码: A

## Discrete particle swarm optimization for no-idle flow shop problem

PAN Quan-ke<sup>1,2</sup>, WANG Ling<sup>2,3</sup>, ZHAO Bao-hua<sup>1</sup>

(1. School of Information Science and Technology, University of Science and Technology of China, Hefei 230026, China; 2. College of Computer Science, Liaocheng University, Liaocheng 252059, China; 3. Department of Automation, Tsinghua University, Beijing 100084, China. Correspondent: PAN Quan-ke, E-mail: qkpan@lctu.edu.cn)

Abstract: No-idle flow shop (NIFS) problem with makespan criterion is studied. A method with the computational complexity  $O(mn)$  is developed to calculate the makespan of a permutation, and then a short cut for insert neighborhood search algorithm is proposed. A discrete particle swarm optimization (DPSO) algorithm is presented for the NIFS, which is improved by employing the short cut. The computational results show that the effectiveness of the presented DPSO.

Key words: No-idle flow shop; Particle swarm optimization; Makespan

### 1 引 言

零空闲流水线调度问题(NIFS)是一类十分重要的调度问题<sup>[1-4]</sup>,它可描述为:有  $n$  个零件按照相同的工艺路线在  $M$  台机床上加工;在所有机床上的零件加工次序都相同;在同一机床上加工的相邻工件之间没有等待时间;机床之间存在无限缓冲区;一个零件在某一时刻只能在一台机床上加工;一台机床在某一时刻只能加工一个零件;零件的加工时间已知;问题是如何安排加工次序使所有零件完工的时间最短. NIFS 是一类强 NP 难题. 最近发展起来的粒子群算法(PSO)为该类问题的求解提供了一种新思路<sup>[5,6]</sup>. 但 PSO 具有连续本质,求解这类复杂离散问题面临诸多困难. 为此,文献[7,8]基于 PSO 优化机理的分析,提出了一种离散域粒子群算法(DPSO).

鉴于 DPSO 的成功应用,本文用其解决以最大完工时间为目标的 NIFS 问题,并通过典型算例验证了所得算法的可行性和优越性.

### 2 NIFS 的最大完工时间及邻域搜索算法

#### 2.1 最大完工时间的计算

记  $\pi = \{ (1), (2), \dots, (n) \}$  为一个零件排列,  $p_{(j),i}$  为零件  $(j)$  在机床  $i$  上的加工时间,  $f_{i-1,i}(\pi)$  为机床  $i-1$  和  $i$  的完工时间之差,如图 1 所示. 则排列  $\pi$  对应的最大完工时间为

$$M(\pi) = \sum_{i=2}^m f_{i-1,i}(\pi) + \sum_{j=1}^n p_{(j),1}. \quad (1)$$

若在排列  $\pi$  之后增加零件  $k$ , 得到新排列  $\pi' = \{ (1), (2), \dots, (n), k \}$ , 则有

$$f_{i-1,i}(\pi') = p_{k,i} + \max\{ f_{i-1,i}(\pi) - p_{k,i-1}, 0 \}. \quad (2)$$

结合式(1)和(2),求解  $M(\pi')$  的算法如下(其

收稿日期: 2006-11-06; 修回日期: 2007-02-07.

基金项目: 国家自然科学基金项目(60774082); 中国博士后科学基金项目(20070410791).

作者简介: 潘全科(1971—),男,山东阳谷人,教授,博士后,从事计算智能及其应用等研究; 赵保华(1947—),男,安徽霍邱人,教授,博士生导师,从事软件工程、协议理论与协议工程等研究.

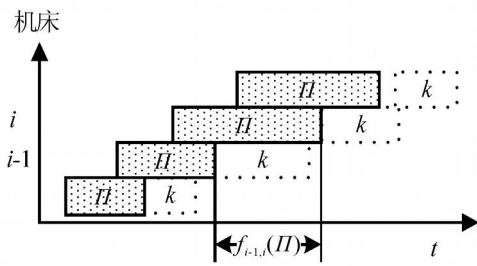


图 1 在排列 后增加零件 k 对机床完工时间的影响  
复杂度为  $O(mn)$  ) :

- 1) 令  $j = 1, \pi = \{ (j) \}$ , 则  $f_{i-1,i}(\pi) = p_{(j),i}, i = 2, 3, \dots, m$ ;
- 2)  $j = j + 1, \pi = \{ (j) \}$ , 按式(2) 计算  $f_{i-1,i}(\pi), i = 2, 3, \dots, m$ ;
- 3) 若  $j < n$ , 转步骤 2) ;
- 4) 按式(1) 计算最大完工时间, 算法结束.

2.2 简化邻域搜索算法

定义 1 在零件排列  $\pi$  中, 随机选择不同的两个位置  $k$  和  $j$ , 把位置  $k$  对应的零件插入位置  $j$ , 称为插入移动. 由该移动得到的排列称为原排列的一个邻居, 所有这样的邻居构成插入邻域. 显然该邻域的规模为  $(n - 1)^2$ , 评价该邻域的时间复杂度为  $O(mn^3)$ .

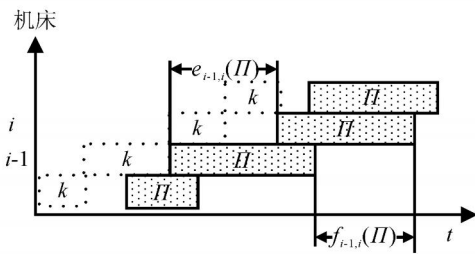


图 2 在排列 前增加零件 k 对机床完工时间的影响

在图 2 中, 记  $e_{i-1,i}(\pi)$  为相邻两机床  $i - 1$  和  $i$  的开工时间之差. 若在 前增加零件  $k$ , 得到新排列  $\pi = \{ k, (1), (2), \dots, (n) \}$ , 则有

$$e_{i-1,i}(\pi) = p_{k,i-1} + \max\{ e_{i-1,i}(\pi) - p_{k,i}, 0 \}, \quad (3)$$

$$f_{i-1,i}(\pi) = f_{i-1,i}(\pi) + \max\{ p_{k,i} - e_{i-1,i}(\pi), 0 \}. \quad (4)$$

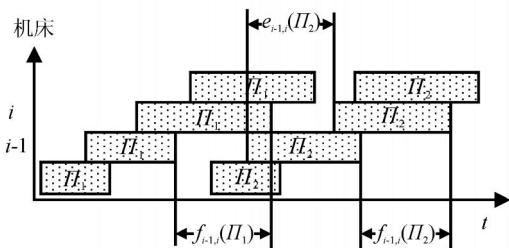


图 3 两个部分调度的合并

如图 3 所示,  $\pi_1$  和  $\pi_2$  为两个部分调度. 若将  $\pi_2$

安排在  $\pi_1$  之后, 得到  $\pi = \pi_1 \pi_2$ , 则有

$$f_{i-1,i}(\pi) =$$

$$f_{i-1,i}(\pi_2) + \max\{ f_{i-1,i}(\pi_1) - e_{i-1,i}(\pi_2), 0 \}. \quad (5)$$

根据上述分析, 可得到复杂度为  $O(mn^2)$  的邻域搜索算法如下:

- 1) 令  $k = 1$ .
- 2) 从  $\pi$  中取出  $(k)$ , 记所得新排列为

$$\pi = \{ (1), (2), \dots, (n - 1) \},$$

并令  $\pi_j$  和  $\pi_{-j}$  分别为  $\pi$  的部分排列, 且

$$\pi_j = \{ (1), (2), \dots, (j) \},$$

$$\pi_{-j} = \{ (j), \dots, (n) \}.$$

令  $j = 1, \dots, n - 1$ , 计算  $f_{i-1,i}(\pi_j)$ .

令  $j = n - 1, \dots, 1$ , 计算  $e_{i-1,i}(\pi_{-j})$  和

$$f_{i-1,i}(\pi_{-j}).$$

- 3) 令  $j = 0, \dots, n - 1$  且  $j \geq k - 1$ , 作以下循环:

计算  $e_{i-1,i}(\pi_{-j} \cup \{ (k) \})$ .

令  $\pi = (\pi_{-j} \cup \{ (k) \}) \pi_j$ , 计算

$$f_{i-1,i}(\pi), \text{ 其中 } i = 2, 3, \dots, m.$$

根据式(1) 计算最大完工时间(式(1) 中,

$p_{(j),i}$  可看作常数).

- 4)  $k = k + 1$ . 如  $k = n$ , 返回步骤 2) ; 否则算法结束.

3 DPSO 调度算法

3.1 DPSO 算法

PSO 的位置矢量和速度矢量的表达以及粒子更新策略等均具有连续本质, 而 NIFS 是复杂的离散问题, 故需要专门设计位置编码及其更新方法.

1) 位置矢量编码. 对于 NIFS 问题, 最直接的编码方法就是用位置矢量的一维代表一个零件, 这样粒子本身就表示所有零件的一个排列, 即一个调度.

2) 位置更新公式. 粒子群算法的实质在于粒子根据自己和同伴的飞行经验不断调整位置和速度, 从而向最优位置飞行. 粒子的新位置是粒子的速度、个体极值和全体极值相互作用的结果. 新粒子可按下式产生:

$$X_i^{k+1} =$$

$$c_2 \otimes g(c_1 \otimes g(w \otimes h_{a,b}(X_i^k), pB_i^k), gB^k). \quad (6)$$

式中:  $X_i^k$  为粒子  $i$  在第  $k$  次迭代中的位置,  $pB_i^k$  为粒子  $i$  在第  $k$  次迭代中的个体极值,  $gB^k$  为第  $k$  次迭代中的全体极值,  $\text{Rand}()$  为区间  $[0, 1]$  上的随机数,  $w$  为惯性系数,  $c_1$  为认知系数,  $c_2$  为社会系数.

3) 位置更新公式的构成. 位置更新公式由 3 部分构成:

第 1 部分为

$$E_i^k = w \otimes h_{a,b}(X_i^k) =$$

$$\begin{cases} h_{a,b}(X_i^k), \text{rand}() < w; \\ X_i^k, \text{rand}() < w. \end{cases}$$

上式表示粒子对飞行速度的思考. 其中  $h_{a,b}(X_i^k)$  表示粒子的速度, 它的实现方法为随机交换粒子位置矢量的第  $a$  维和第  $b$  维.

第 2 部分为

$$\begin{cases} F_i^k = c_1 \otimes g(E_i^k, pB_i^k) = \\ g(E_i^k, pB_i^k), \text{rand}() < c_1; \\ E_i^k, \text{rand}() < c_1. \end{cases}$$

上式表示粒子根据  $pB_i^k$  调整位置. 其中  $Y_i^k = g(E_i^k, pB_i^k)$ , 实现方法为: 先从  $E_i^k$  中随机抽取一段, 放在  $pB_i^k$  的前面或后面, 再删除  $pB_i^k$  中与该段相同的零件.

第 3 部分为

$$\begin{cases} X_i^{k+1} = c_2 \otimes g(F_i^k, gB^k) = \\ g(F_i^k, gB^k), \text{rand}() < c_2; \\ F_i^k, \text{rand}() < c_2. \end{cases}$$

上式表示粒子根据  $gB^k$  调整位置.

### 3.2 DPSO 算法的改进

采用上述位置编码和更新公式代替传统 PSO 的相应操作, 就得到 DPSO 算法. 这种 DPSO 算法虽然收敛快, 但容易陷入局部最优. 由算法原理知, DPSO 的信息传递是单向的.  $gB^k$  将信息传给其他粒子, 带动其他粒子在较好的区域搜索. 因此, 加强  $gB^k$  的局部探测能力可提高算法性能. 改进 DPSO 的一种方法是在每次迭代中, 对新得到的  $gB^k$  执行简化插入邻域搜索算法, 记该算法为 DPSO1. DPSO1 虽然加强了局部搜索能力, 提高了收敛速度, 但更容易陷入局部最优. 为了增强算法跳出局部

最优的能力, 可先对  $gB^k$  执行几次随机插入移动, 再对所得结果执行邻域搜索算法. 采用这种改进措施后的 DPSO1 算法记为 DPSO2. 模拟退火算法的研究表明<sup>[9]</sup>, 采用概率接受准则能有效地跳出局部最优, 因此对于 DPSO2, 可采用概率接受准则进一步提高算法性能, 记该算法为 DPSO3.

### 3.3 DPSO 算法的初始化

较好的初始群体能有效加快算法的收敛速度, 提高求解质量. 鉴于 NEH 算法在求解流水线调度时表现出的优越性能, 采用 NEH 算法产生一个粒子, 其余粒子随机产生.

## 4 仿真实验

Taillard Benchmark 包含 120 个流水线调度问题实例, Taillard 的主页 (<http://mistic.heig-vd.ch/taillard>) 上有这些实例的数据和当前最好解. 将这些实例当作 NIFS 问题, 每个实例计算 5 次, 求算法所得解的平均相对偏差 (PRD) 和平均均方差 (SD). 设置群体规模为 20,  $c_1 = c_2 = w = 0.8$ , 对  $gB^k$  执行 4 次随机插入移动, 模拟退火接受标准的温度系数为  $t = 0.05 \prod_{j=1}^n \prod_{i=1}^m p_{ji} / (mn)$ . 同时采用文献<sup>[10]</sup>提出的遗传算法 (GA) 解决这些问题. 各算法采用相同的终止条件, 即最大运行时间为 10 mn/ ms. 结果如表 1 所示.

由表 1 可知:

- 1) 与 GA 相比, DPSO 所得的平均偏差和平均均方差较小, 表明 DPSO 优于 GA 算法.
- 2) 从 DPSO 到 DPSO1, DPSO2 和 DPSO3, 各算法所得平均偏差和平均均方差逐渐减小, 表明本

表 1 各算法的计算结果

问题	GA		DPSO		DPSO1		DPSO2		DPSO3	
	PRD	SD	PRD	SD	PRD	SD	PRD	SD	PRD	SD
20 × 5	10.06	0.24	10.31	0.16	10.24	0.01	9.84	0.00	9.85	0.02
20 × 10	24.11	1.20	25.38	1.25	25.51	0.45	22.94	0.06	22.93	0.06
20 × 20	46.10	1.30	48.87	1.21	49.44	0.81	45.41	0.16	45.39	0.11
50 × 5	9.20	0.17	9.22	0.09	9.34	0.11	9.11	0.00	9.11	0.00
50 × 10	20.95	0.66	21.53	0.63	21.25	0.13	18.63	0.24	18.62	0.13
50 × 20	50.17	0.70	51.59	0.82	51.90	0.17	46.69	0.37	46.66	0.45
100 × 5	5.39	0.21	4.94	0.10	4.94	0.01	4.75	0.01	4.74	0.00
100 × 10	21.19	0.18	21.05	0.16	20.70	0.04	20.06	0.06	20.08	0.07
100 × 20	43.12	0.70	44.15	0.60	42.52	0.09	39.78	0.20	39.77	0.22
200 × 10	15.45	0.18	13.47	0.16	13.01	0.07	12.32	0.07	12.31	0.09
200 × 20	34.37	0.39	32.94	0.30	31.38	0.03	29.70	0.20	29.68	0.25
500 × 20	26.10	0.18	21.41	0.12	19.98	0.00	19.32	0.16	19.31	0.13
平均	25.52	0.51	25.40	0.47	25.02	0.16	23.21	0.13	23.20	0.13

文采取的提高算法性能的措施是有效的. 特别地, DPSO2 较 DPSO1 的性能有明显提高, 表明随机扰动对提高算法性能具有关键作用.

## 5 结 语

针对零空闲流水线调度问题, 得到了一种简单的最大完工时间算法和快速插入邻域搜索方法, 并提出了基于 DPSO 的全局优化算法. 仿真实验表明了所得算法的有效性和优越性.

## 参考文献 (References)

- [1] Saadani N E I, Guinet A, Moala M. A traveling salesman approach to solve the  $f/\text{no-idle}/C_{\max}$  problem [J]. *European J of Operation Research*, 2005, 161(1): 11-20.
- [2] Kalczynski P J, Kamburowski J. A heuristic for minimizing the makespan in no-idle permutation flow shops[J]. *Computer & Industrial Engineering*, 2005, 49(1): 146-154.
- [3] Kalczynski P J, Kamburowski J. On no-wait and no-idle flow shops with makespan criterion[J]. *European J of Operational Research*, 2007, 178(3): 677-685.
- [4] Cheng M B, Sun S J, Yu Y. A note on flow shop scheduling problems with a learning effect on no-idle dominant machines [J]. *Applied Mathematics and Computation*, 2006, 184(2): 945-949.
- [5] Liu Bo, Wang Ling, Jin Yihui. An effective PSO-based memetic algorithm for flow shop scheduling[J]. *IEEE Trans on Systems, Man and Cybernetics*, 2007, 37(1): 18-27.
- [6] He Qie, Wang Ling. An effective co-evolutionary particle swarm optimization for constrained engineering design problems [J]. *Engineering Applications of Artificial Intelligence*, 2007, 20(1): 89-99.
- [7] Pan Q K, Tasgetiren M F, Liang Y C. A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem with makespan criterion[C]. *Proc of the Int Workshop on UK Planning and Scheduling Special Interest Group*. London: City University, 2005: 31-41.
- [8] Pan Q K, Tasgetiren M F, Liang Y C. Minimizing total earliness and tardiness penalties with a common due date on a single-machine using a discrete particle swarm optimization algorithm[J]. *Lecture Notes in Computer Science*, 2006, 4150: 460-467.
- [9] 王凌. 智能优化算法及其应用[M]. 北京: 清华大学出版社, 2001.  
(WANG Ling. *Intelligent optimization algorithm with applications*[M]. Beijing: Tsinghua University Press, 2001.)
- [10] 潘全科, 朱剑英. 多工艺路线的批量生产调度优化. *机械工程学报*, 2004, 40(4): 36-39.  
(Pan Quar-ke, Zhu Jian-ying. Optimization method for a job shop scheduling problem with alternative machines in the batch process [J]. *Chinese J of Mechanical Engineering*, 2004, 40(4): 36-39.)
- [6] Thrun S, Fox D, Burgard W, et al. Robust monte carlo localization for mobile robots[J]. *Artificial Intelligence*, 2001, 128(1): 99-141.
- [7] Fox D, Burgard W, Thrun S. Active Markov localization for mobile robots [J]. *Robotics and Autonomous Systems*, 1998, 25(3/4): 195-207.
- [8] Kristensen S. Sensor planning with Bayesian decision analysis[D]. Aalborg: Aalborg University, 1996.
- [9] Zhou H, Sakane S. Sensor planning for mobile robot localization based on probabilistic inference using Bayesian network [C]. *Proc of the 4th IEEE Int Symposium on Assembly and Task Planning*. Fukuoka: IEEE Press, 2001: 7-12.
- [10] Zhou H, Sakane S. Sensor planning for mobile robot localization — A hierarchical approach using Bayesian network and particle filter[C]. *Proc of the IEEE Int Conf on Robotics and Biomimetics*. Shenyang: IEEE Press, 2004: 540-545.
- [11] Beetz M, Burgard W, Fox D, et al. Integrating active localization into high-level robot control systems [J]. *Robotics and Autonomous Systems*, 1998, 23(4): 205-220.
- [12] Porta J, Terwijn B, Krošič B. Efficient entropy-based action selection for appearance-based robot localization [C]. *Proc of the IEEE Int Conf on Robotics and Automation*. Taipei: IEEE Press, 2003: 2842-2847.
- [13] Porta J, Verbeek J, Krošič B. Active appearance-based robot localization using stereo vision[J]. *Autonomous Robots*, 2005, 18(1): 59-80.

(上接第 190 页)