

一种求解作业车间调度的文化粒子群算法*

朱 霞

(南京航空航天大学 航空宇航学院, 南京 210016)

摘 要: 提出了一种文化粒子群算法用于求解置换流水车间调度问题中的最小化最大完成时间。算法设置了群体空间和信念空间两类独立空间,群体空间采用自适应粒子群算法完成进化,信念空间通过更新函数来进行演化。算法中群体空间的粒子群不但通过跟踪个体极值和全局极值来更新自己,实现群体演化,而且通过不断与信念空间中的优秀个体交互,加快群体的收敛速度。该算法在不同规模的问题实例上与其他几个具有代表性的算法的比较结果表明,该算法具有较快的收敛速度,无论是在求解质量还是稳定性方面都优于比较的算法。

关键词: 粒子群算法; 文化算法; 作业车间调度

中图分类号: TP301

文献标志码: A

文章编号: 1001-3695(2012)04-1234-03

doi:10.3969/j.issn.1001-3695.2012.04.009

Cultural particle swarm optimization algorithm for Job-Shop scheduling problem

ZHU Xia

(College of Aerospace Engineering, Nanjing University of Aeronautics & Astronautics, Nanjing 210016, China)

Abstract: This paper proposed an algorithm for the minimization of the makespan in permutation flow shop scheduling problem (FSSP), which combined cultural particle swarm optimization (CPSO). The algorithm set two kinds of spaces, population space and belief space. The population space was evolved with adaptive PSO strategy, and the belief space was evolved with update function. Particles of population space not only tracked individual extreme and global extreme to update themselves, but also exchanged with good individuals of belief space to speed up the convergence speed. The proposed algorithm was tested on different scale benchmarks and compared with the other representative algorithms. The result shows that CPSO has faster convergence speed and is better than those algorithms in not only the solution quality but also the stability.

Key words: particle swarm optimization; cultural algorithm; Job-Shop scheduling

0 引言

作业车间调度问题(JSP)研究 N 个工件在 M 台机器上的加工过程,每个工件在各台机器上的加工时间已知,并且每个工件必须按照事先规定的顺序完成其在所有机器上的加工,即技术约束条件,要求确定在每台机器上加工的所有工件的顺序或者开工与完工时间,使得在符合各工件技术约束条件的前提下最优化某个或者某些调度指标^[1]。进一步,若约定每台机器上所有工件的加工顺序相同,则这类 FSP (flow-shop scheduling problem) 称为 PFSP (permutation FSP)。尽管 PFSP 的工艺约束比较简单,但是已经证明三台以上机器的 PFSP 即为 NP-hard 问题,至今没有一个具有多项式计算复杂性的全局优化算法能求解。本文主要研究 PFSP 中的最大完成时间问题。最大完成时间是生产调度中最常见的性能指标之一,最大完成时间越短,说明产品总的生产周期越短,生产能力越大,即最小化最大完成时间意味着最大化生产能力。考虑最小化最大完成时间的经典置换流水线调度问题,记 p_{ij} 为工件 i 在机器 j 上的加工时间, $\pi = (j_1, j_2, \dots, j_n)$ 为所有工件的一个排序, Π 为所有排序的集合, $C(j_i, k)$ 为工件 j_i 在机器 k 的加工完毕时间。假设各工件按机器 1 至机器 m 的顺序进行加工,则各工件在每台机器上完成时间的数学公式可描述如下:

$$C(j_1, 1) = p_{j_1, 1} \quad (1)$$

$$C(j_i, 1) = C(j_{i-1}, 1) + p_{j_i, 1}, \quad i = 2, \dots, n \quad (2)$$

$$C(j_1, k) = C(j_1, k-1) + p_{j_1, k}, \quad k = 2, \dots, m \quad (3)$$

$$C(j_i, k) = \max\{C(j_{i-1}, k), C(j_i, k-1)\} + p_{j_i, k} \\ i = 2, \dots, n; k = 2, \dots, m \quad (4)$$

$$C_{\max}(\pi) = C(j_n, m) \quad (5)$$

$$\pi^* = \arg\{C_{\max}(\pi) = C(j_n, m)\} \rightarrow \min, \quad \forall \pi \in \Pi \quad (6)$$

其中,式(5)即为最大完成时间,式(6)表示最小化最大完成时间的调度排序方案。

粒子群优化算法(PSO)最早是 Kennedy 等人受鸟群觅食行为的启发,于 1995 年提出的一种生物进化算法。PSO 算法基于群体与适应度,它将每个个体视为无体积的粒子,在搜索空间以变速飞行,其飞行速度根据该粒子本身的历史经验以及同伴的历史经验进行动态调整。Lian 等人^[2]基于微粒子群优化的思想提出了一种 PSO 算法,采用基于操作的编码,直接将遗传算法的交叉和变异操作替代 PSO 的速度和位置更新操作,进而用于求解最小化最大完成时间的 JSP。Liu 等人将微粒子群优化算法和 VNS 相结合,给出了求解最小化最大完成时间和最小化最大流经时间的柔性 JSP 的混合算法。Sha 等人^[3]针对最小化最大完成时间的 JSP,采用基于先后表的编码方式,利用 Giffler-Thompson 启发式算法将基于先后表编码的

收稿日期: 2011-09-06; 修回日期: 2011-10-21 基金项目: 淮安市科技计划资助项目(SN1045); 淮安市科技局资助项目(HAG09052)

作者简介: 朱霞(1980-),女,江苏通州人,讲师,博士研究生,主要研究方向为嵌入式系统及计算机测控技术、计算机应用、智能算法(snail2024@163.com)。

微粒位置解码为可行调度,进而采用 SWAP 操作更新微粒子的速度,并结合基于块结构的禁忌搜索来求解问题。

Reynolds^[4]于1994年提出了文化算法(cultural algorithm),从微观和宏观的不同层面模拟生物层面的进化和文化层面的进化,该算法的双层演化结构框架有利于算法的全局收敛性。因此本文将文化算法引入到粒子群算法,提出一种文化粒子群算法(CPSO),并将该算法用于求解作业车间调度问题中的最小化最大完成时间问题。该算法采用文化算法的双层演化结构,设立两类粒子空间:群体空间和信念空间。群体空间采用自适应粒子群算法完成独立进化,信念空间通过更新函数来进行演化,两类空间的交互通过接受操作和影响操作采用同步传输方式定期地完成。该算法利用优秀个体所包含的信息提高算法的收敛性。通过在相同参数设置下分别对本文算法与文献算法进行比较,结果表明,文化粒子群算法不但在提高收敛速度、避免局部最优问题上有一定的优越性,而且避免了群体早熟的发生。

1 文化粒子群算法

本文提出的文化粒子群算法采用文化算法的双层演化结构^[5]。算法包含两个进化空间,一个是由具体个体组成的群体空间;另一个是由在进化过程中获取的经验和知识组成的信念空间。其演化框架如图1所示,其中群体空间是算法进行问题求解的主空间,通过演化操作和性能评价进行自身的迭代求解,形成个体经验,并通过 accept() 函数将个体经验传递到信念空间;信念空间将接收到的个体经验不断地进行自身的性能评价与演化,形成群体经验,并通过 influence() 函数对群体空间的个体行为规则进行修改,以使群体空间得到更高的进化效率^[6]。

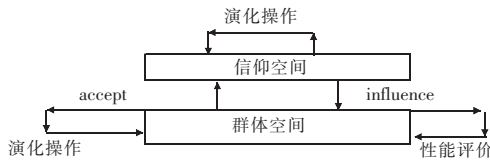


图1 文化粒子群算法的基本框架

文化算法总体上包括三大元素:种群空间(population space)、信念空间(belief space)和沟通渠道(communication channel),其中沟通渠道又包括接受函数(acceptance function)、更新函数(update function)和影响函数(influence function)。初始化种群空间、信念空间及沟通渠道后,通过性能函数(performance function)评价种群空间中的个体适应值。将种群空间个体在进化过程中所形成的个体经验,通过接受函数传递给信念空间;信念空间将收到的个体经验按一定的规则进行比较优化,形成群体经验,并根据新获取的个体经验通过更新函数更新现有的信念空间。信念空间再用更新后的群体经验通过影响函数来对种群空间中个体的行为规则进行修改,进而高效地指引种群空间的进化。选择函数从现有种群中选择一部分个体作为下代个体的父辈,进行下一轮的迭代直至满足终止条件。

1.1 群体空间的设计

由于 PSO 是一种基于个体改进、种群协作与竞争的进化计算方法,具有理论简单、易于编码实现的特点,因此群体空间采用基本粒子群算法进行演化。粒子群^[1]可用下面五元素形式描述:

$$PSO = (n, K_{iter}, V, X, F_{fit}) \quad (7)$$

其中: n 为群体规模, K_{iter} 为迭代次数, V 和 X 分别表示所有粒子的速度空间和位置空间, F_{fit} 为适应度,从位置空间映射到实数空间。在每一次迭代中,粒子通过跟踪个体极值 $pbest$ 与全局极值 $gbest$ 来更新自己的位置和速度,公式如下:

$$v_i(t+1) = \omega \times v_i(t) + c_1 \times rand() \times (p_i \text{best} - x_i(t)) + c_2 \times rand() \times (gbest - x_i(t)) \quad (8)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (9)$$

其中: $v(t)$ 是粒子的当前速度; $x(t)$ 是粒子的当前位置; $rand()$ 是 $[0,1]$ 间的随机函数; c_1 和 c_2 是学习因子,通常都取值为 2; ω 是加权系数,用来控制历史速度对当前速度的影响程度,一般在 $[0.1, 0.9]$ 间取值。

1.2 信念空间的设计

信念空间^[5]定义为 $\langle S, N[n] \rangle$ 。其中, S 表示环境知识中存储的一组由历代种群所产生的最优个体集合;标准知识 N 保存目标函数 n 个主变量参数的变化区间。每个区间又描述为 $\langle I_r, L_r, U_r \rangle, r = 1, 2, \dots, n$ 。其中, $I = [l, u] = \{l \leq x \leq u, x \in R\}$ 表示变量 x 定义域边界的值。上下界 u 和 l 由给定的值域初始化, L_j 表示参数 j 区间下限 l_j 对应的目标函数的适应值, U_j 是参数 j 区间上限 u_j 对应的目标函数的适应值。

1.3 接受函数

种群空间通过接受函数将个体经验传递到信念空间,实际上是向信念空间提供一组最优子集。在最优优化问题中一般是按一定的百分比取排行头几名,通常取 20% ~ 25%,但也可以根据问题环境的变化按一定的规则选取参数。

接受操作^[6]是指在群体空间的粒子群演化过程中,每运行 acc 代时,用群体空间当前的全局最优值来替换信念空间中适应值较差的个体。

$$acc = B_{num} + \frac{iter}{iter_{max}} \times D_{num} \quad (10)$$

其中: B_{num} 和 D_{num} 为文化算法中的两个参数,用于调节接受操作和影响操作的完成次数,一般取常值。

1.4 更新信念空间

不同的知识种类均有其各自相应的更新函数。这里简单介绍环境知识和标准知识相应的更新函数。环境知识中存储了当代以及前一代最优个体,即 $S = \{S^t, S^{t-1}\}$ 表示第 t 代及 $t-1$ 代产生的最优个体。其更新规则如下:

$$S^{t+1} = x_{best}^t \quad \text{if } f(x_{best}^t) < f(S^t)$$

$$S^{t+1} = S^t \quad \text{otherwise}$$

标准知识 N 的更新规则如下:

$$l_j^{t+1} = x_{i,j}^t \quad \text{if } x_{i,j}^t \leq l_j^t \text{ or } f(x_i^t) < L_j^t$$

$$l_j^{t+1} = l_j^t \quad \text{otherwise}$$

$$L_j^{t+1} = f(x_i^t) \quad \text{if } x_{i,j}^t \leq l_j^t \text{ or } f(x_i^t) < L_j^t$$

$$L_j^{t+1} = L_j^t \quad \text{otherwise}$$

这里,第 i 个个体影响参数 j 的区间下边界, l_j^t 表示参数 j 第 t 次迭代时的下边界, L_j^t 表示 l_j^t 对应目标函数的适应值。同理,第 k 个个体影响参数 j 的区间上边界, u_j^t 表示参数 j 第 t 次迭代时的上边界, U_j^t 表示 u_j^t 对应目标函数的适应值。

$$u_j^{t+1} = x_{k,j}^t \quad \text{if } x_{k,j}^t \geq u_j^t \text{ or } f(x_k^t) < U_j^t$$

$$u_j^{t+1} = u_j^t \quad \text{otherwise}$$

$$U_j^{t+1} = f(x_k^t) \quad \text{if } x_{k,j}^t \geq u_j^t \text{ or } f(x_k^t) < U_j^t$$

$$U_j^{t+1} = U_j^t \quad \text{otherwise}$$

1.5 影响函数

信念空间在形成群体经验后,通过影响函数对群体空间中个体的行为规则进行修改,以使个体空间得到更高的进化效率。例如,在标准知识 N 中所存储的更新后的变量区间 I 内产生部分新的种群与原有种群一起进入下一步迭代,使之在信念空间的指导下能进一步缩小搜索空间,更快找到可行域及更优解。影响函数决定哪种知识将被用于指导需要解决的问题。根据具体问题的不同,影响函数的设计也不同,不同的知识源对种群空间也有相应不同的影响函数。同时也可设计一个主影响函数(main influence function),根据每个知识种类的性能来决定其所起作用的轻重。

影响操作^[6]是指在群体空间的粒子群演化过程中,每运行 inf 代时,用信念空间当前的全局最优值来替换群体空间中适应值较差的个体。

$$\text{inf} = B_{\text{num}} + \frac{\text{iter}_{\text{max}} - \text{iter}}{\text{iter}_{\text{max}}} \times D_{\text{num}} \quad (11)$$

其中: iter_{max} 为粒子群预先设定的粒子群最大演化代数, iter 为粒子群演化的当前代数。

2 实验

2.1 实验分析

以一个 3/3/G/C_{max} 作业车间调度问题为例,表 1 给出该问题的加工时间和工艺约束。鉴于 JSP 的组合特性和工艺约束,用于 CPSO 的编码必须特殊设计。因为 CPSO 算法中的微粒子的位置为连续值,首先利用 ROV 规则实现连续位置到离散值的转换,其次利用析取图编码将有序表作为优先决策来决定同台机器上发生冲突时各操作的顺序,然后利用关键路径法生成调度。

表 1 3 工件 3 机器 JSP 的加工数据和工艺约束

参数	工件	操作序号		
		1	2	3
操作时间	J_1	3	2	5
	J_2	3	5	1
	J_3	2	5	3
机器顺序	J_1	M_1	M_2	M_3
	J_2	M_1	M_3	M_2
	J_3	M_2	M_1	M_3

1) ROV 规则

每个粒子的位置矢量采用随机键编码,对于 n 个工件在 m 台机器的 JSP,微粒子 k 的位置矢量用一个 $m \times n$ 矩阵 X_k 表示。

$$X_k = \begin{bmatrix} x_{11}^k & x_{12}^k & \cdots & x_{1n}^k \\ x_{21}^k & x_{22}^k & \cdots & x_{2n}^k \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1}^k & x_{m2}^k & \cdots & x_{mn}^k \end{bmatrix}$$

X_k 的每个元素均为一个实数值, X_k 的第 i 行对应机器 i 。利用 ROV 规则,分别对 X_k 的每一行的位置值进行编码转换,得到各台机器上的所有操作的优先序。

假设第 k 个粒子的位置如下:

$$X_k = \begin{bmatrix} 1.34 & 1.16 & 3.50 \\ 3.28 & 0.01 & 0.67 \\ 3.30 & 3.41 & 2.42 \end{bmatrix}$$

则对于机器 1,利用 ROV 规则得到序列 2-1-3;对于机器 2,利

用 ROV 规则得到序列 3-1-2;对于机器 3,利用 ROV 规则得到序列 2-3-1。

2) 析取图

上面利用 ROV 规则得到一个有序操作表,先根据有序操作表设置析取图中的弧方向,得到一个非循环图,如图 2 所示。

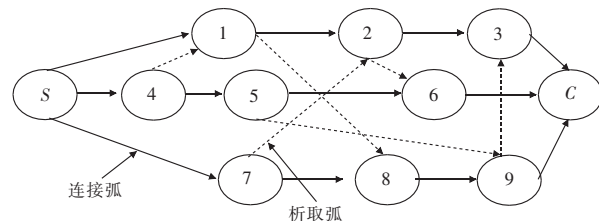


图 2 结合有序表所得的析取图

对应图 2,可以确定关键路径(4-1-8-9-3)。对应的甘特图如图 3 所示。

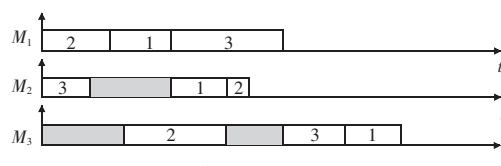


图 3 3 工件 3 机器 JSP 的调度解

基于前面几部分的介绍,本文提出的求解 JSP 的 CPSO 算法描述如下:

- 初始化设置。设置群体空间和信念空间的种群大小,群体规模 $n = 30$,惯性权因子 $\omega = 0.9$,加速因子 $c_1 = c_2 = 2$,微粒子最小位置值 $x_{\min} = 0$,最大位置值 $x_{\max} = 4.0$,微粒子最小速度值 $v_{\min} = -4.0$,最大速度值 $v_{\max} = 4.0$ 。
- 基于随机键编码初始化粒子的位置和速度,根据析取图编码评价各微粒的目标值。
- 根据群体空间粒子位置,计算粒子的实际输入和适应值函数。
- 更新信念空间粒子位置,并计算粒子的实际输入和适应值函数。
- 判断若 $\text{iter} \% \text{acc} == 0$ 时,进行接受操作。
- 更新群体空间粒子的位置和速度,根据析取图编码评价各微粒的目标值,并检验粒子的速度和位置是否越界,若是,调整速度和位置。
- 计算群体空间中各粒子的实际输入和适应值函数。
- 更新信念空间粒子的位置和速度,根据析取图编码评价各微粒的目标值,并检验粒子的速度和位置是否越界,若是,调整速度和位置。
- 计算信念空间中粒子的实际输入和适应值函数。
- 判断若 $\text{iter} \% \text{inf} == 0$ 时,进行影响操作。
- 当误差达到最初设定值或最大迭代次数时,输出 g_{best} 对应的调度方案和目标值;否则,令 $\text{iter} = \text{iter} + 1$,返回 c)。

2.2 实验仿真

为了验证 CPSO 算法求解 JSP 的性能,考虑采用一个典型 JSP 算例(LA 类),这些测试问题在文献中被广泛用于测试。LA 类问题由 Lawrence 给出,在此选取其中八个不同规模的问题:LA01,LA06,LA11,LA16,LA21,LA26,LA31,LA36。各算法均独立运行 25 次。实验结果如表 2、3 所示。

硬件仿真环境: Intel Core™ 2 DUO CPU T5250 @ 1.50 GHz/2 GB;软件平台: Vista 操作系统, MATLAB 7.0 编译软件。

(下转第 1240 页)

络社区的划分。与其他算法相比较,该算法无须预先知道复杂网络的社区数量或社区内的节点数量等先验信息,并且在种群初始化敏感度、准确率以及运算耗时等方面都具有较好的性能,因此本文提出的算法具有很强的现实应用价值。

与此同时,多种群遗传算法还将并行的思想引入到算法之中,将各子种群分别分配给多个处理器并行求解。当采用并行计算时,可以明显提高遗传算法的运行速度。在现有的普遍的串行计算机上无法体现出并行性所带来的高速计算能力,但在未来的并行计算机上,本文算法具有较强的优越性。

参考文献:

- [1] 汪小帆,李翔. 复杂网络理论及其应用[M]. 北京:清华大学出版社,2006.
- [2] KERNIGHAN B W, LIN S. An efficient heuristic procedure for partitioning graphs[J]. *Bell System Technical Journal*, 1970, 49(2): 291-307.
- [3] FIEDLER M. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory[J]. *Czechoslovak Mathematical Journal*, 1973, 23(298): 619-633.
- [4] POTHEN A, SIMON H D, LIOU Kan-pu. Partitioning sparse matrices with eigenvectors of graphs[J]. *SIAM Journal on Matrix Analysis and Applications*, 1990, 11(3): 430-452.
- [5] WU Fang, HUBERMAN B A. Finding communities in linear time: a physics approach[J]. *The European Physical Journal B, Condensed Matter Physics*, 2003, 38(2): 331-338.
- [6] NEWMAN M E J, GIRVAN M. Finding and evaluating community structure in networks[J]. *Physical Review E*, 2004, 69(2): 26-113.
- [7] NEWMAN M E J. Fast algorithm for detecting community structure in

networks[J]. *Physical Review E*, 2004, 69(6): 66-133.

- [8] NEWMAN M E J. Detecting community structure in networks[J]. *The European Physical Journal B, Condensed Matter and Complex Systems*, 2004, 38(2): 321-330.
- [9] HOLLAND J H. Adaptation in natural and artificial systems[M]. Ann Arbor: University of Michigan Press, 1975.
- [10] ZACHARY W W. An information flow model for conflict and fission in small groups[J]. *Journal of Anthropological Research*, 1977, 33(4): 452-473.
- [11] GIRVAN M, NEWMAN M E J. Community structure in social and biological networks[J]. *Proceedings of the National Academy of Sciences of the United States of America*, 2002, 99(12): 7821-7826.
- [12] GFELLER D, CHAPPELIER J C, De Los RIOS P. Finding instabilities in the community structure of complex networks[J]. *Physical Review E*, 2005, 72(5): 056135.
- [13] 戴飞飞,唐普英. 基于 DNA 遗传算法的复杂网络社区结构发现[J]. *计算机工程与应用*, 2008, 44(3): 53-56.
- [14] LUSSEAU D, SCHNEIDER K, BOISSEAU O J, et al. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations: can geographic isolation explain this unique trait? [J]. *Behavioral Ecology and Sociobiology*, 2003, 54(4): 396-405.
- [15] LUSSEAU D, NEWMAN M E J. Identifying the role that animals play in their social networks[J]. *Processings of the Royal Society B: Biological Sciences*, 2004, 271(6): 477-481.
- [16] NEWMAN M E J. Finding community structure in networks using the eigenvectors of matrices[J]. *Physical Review E*, 2006, 74(3): 036104.

(上接第 1236 页)

表 2 CPSO 算法性能指标

问题	<i>n</i>	<i>m</i>	最好解	平均值	最差解	平均时间/s
LA01	10	5	666	666	666	17.8
LA06	15	5	930	930	930	61.1
LA11	20	5	1 222	1 222	1 222	145.6
LA16	10	10	945	945.5	946	148.3
LA21	15	10	1 024	1 024.4	1 078	733.7
LA26	20	10	1 218	1 218	1 218	2 449.5
LA31	30	10	1 789	1 789	1 789	3 883.6
LA36	15	15	1 270	1 270	1 780	3 550.2

表 3 CPSO 算法与其他 PSO 算法的最优性能比较

问题	<i>n</i>	<i>m</i>	CPSO	文献[2]	文献[3]	文献[7]
LA01	10	5	666	666	666	666
LA06	15	5	930	930	930	930
LA11	20	5	1 222	1 222	1 222	1 222
LA16	10	10	945	945	946	946
LA21	15	10	1 024	1 056	1 078	1 078
LA26	20	10	1 218	1 218	1 218	1 218
LA31	30	10	1 789	1 789	1 789	1 789
LA36	15	15	1 270	1 280	1 287	1 271

由表 2 和 3 可见, CPSO 算法对各算例均能够获得最优解, 这表明 CPSO 算法具有很好的搜索质量。同时算法多次独立运行所得平均值和最差解非常接近, 甚至相同, 这表明 CPSO 算法对初始化解具有较好的鲁棒性。

3 结束语

本文对流水车间调度问题中的最小化最大完成时间问题

进行了研究, 并采用文化粒子群算法来求解 JSP, 取得了较理想的结果。通过对典型 JSP 算例(LA 类)的实例测试, 仿真实验表明, 与基本 PSO 算法相比, CPSO 不容易陷入早熟收敛, 具有更高的求解效率; 与混合粒子群算法相比, CPSO 具有更快的收敛速度和更高的求解质量。基于 CPSO 调度算法的时间性能还有待改善, 笔者还将对此作进一步的研究。

参考文献:

- [1] 王凌. 微粒群优化与调度算法[M]. 北京: 清华大学出版社, 2008.
- [2] LIAN Zhi-gang, GU Xing-sheng, JIAO Bin. A novel particle swarm optimization algorithm for permutation Flow-Shop scheduling to minimize makespan[J]. *Chaos, Solitons and Fractals*, 2008, 35(5): 851-861.
- [3] SHA D Y, HSU C Y. A hybrid particle swarm optimization for Job-Shop scheduling problem[J]. *Computers & Industrial Engineering*, 2006, 51(4): 791-808.
- [4] REYNOLDS R G. An introduction to cultural algorithms[C]//Proc of the 3rd Annual Conference on Evolutionary Programming. River Edge, NJ: World Scientific, 1994: 131-139.
- [5] 仲纪, 刘漫丹. 文化算法研究[J]. *计算机技术与发展*, 2008, 18(5): 126-130.
- [6] 吴亚丽, 袁瑛. 一种基于文化粒子群算法的 BP 网络优化方法[J]. *系统仿真学报*, 2011, 23(5): 930-934.
- [7] TASGETIREN M F, LIANG Y C, SEVKLI M, et al. A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem[J]. *European Journal of Operational Research*, 2007, 177(3): 1930-1947.