

基于混合差分进化算法的无等待流水车间 总流水时间优化*

董 斌¹, 高开周², 潘全科², 孙强强¹

(1. 滨州学院 计算机科学技术系, 山东 滨州 256603; 2. 聊城大学 计算机学院, 山东 聊城 252059)

摘要: 以调度的总流水时间为优化目标, 提出一种混合差分进化算法。首先, 建立无等待流水车间调度的问题模型, 并用快速方法评估总流水时间指标。其次, 采用 LPV 规则, 实现离散问题的连续编码; 用差分进化算法对总流水时间指标执行优化; 引入插入邻域和基于 pairwise 的局部搜索算法, 分别对差分进化算法产生的新个体和差分进化算法的最优解执行邻域搜索, 达到优化目标全局和局部的最优。最后, 通过计算标准算例, 并与其他算法比较, 验证该混合差分进化算法的有效性。

关键词: 无等待流水车间; 差分进化; 总流水时间; 邻域搜索

中图分类号: TP278

文献标志码: A

文章编号: 1001-3695(2010)08-2875-03

doi:10.3969/j.issn.1001-3695.2010.08.016

Hybrid differential evolution optimization algorithm for no-wait flow shop problem with total flow time criterion

DONG Bin¹, GAO Kai-zhou², PAN Quan-ke², SUN Qiang-qiang¹

(1. Dept. of Computer Science & Technology, Binzhou University, Binzhou Shandong 256603, China; 2. College of Computer Science, Liaocheng University, Liaocheng Shandong 252059, China)

Abstract: This paper researched on a hybrid optimization algorithm based on differential evolution. First, built the model of no-wait flow shop and calculated the total flow time. Second, used large position value rule to realize DE solving no-wait flow shop scheduling. Improved optimum using insert-neighbor search and pairwise based neighbor search. Finally, simulations and comparisons based on well-known benchmarks demonstrate the efficiency, effectiveness and robustness of the proposed algorithm.

Key words: no-wait flow shop (NWFS); differential evolution (DE); total flow time; neighbor search

生产调度问题是一类复杂的系统性问题, 其问题描述简单, 但已被证明是 NP-Hard 问题^[1-4]。无等待流水车间 (NWFS) 调度是一类典型的调度问题, 广泛存在于炼钢、食品加工、化工和制药等领域。

智能优化算法因能在合理的时间内求得较高质量的次优解而备受关注^[5]。常见的智能优化算法有遗传算法 (GA)、模拟退火算法 (SA)、粒子群算法 (PSO)、和声搜索 (HS) 等^[6-9], 已成为解决车间调度优化问题的有效方法。差分进化 (differential evolution, DE) 算法是一种基于群体的并行搜索算法, 它已经在许多领域得到了应用, 如人工神经网络、化工、电力、机械设计、机器人、信号处理、生物信息、经济学、现代农业、食品安全、环境保护和运筹学等^[10]。近年来, DE 在多目标、约束、离散和噪声等复杂环境下的优化也得到了一些发展, 但是基本差分进化算法相比混合算法收敛速度较慢, 并且标准 DE 擅长全局搜索, 局部搜索精度较低。已有的用 DE 优化 NWFS 问题总流水时间指标的论文少见, 且结果比其他算法稍差, 因此, 本文提出混合差分进化 (HDE) 的算法, 并以 NWFS 的总流

水时间作为优化目标优化求解。

1 无等待流水车间调度模型

1.1 问题描述

NWFS 调度问题可描述为, 给定 n 个工件和 m 台机床, 所有工件在各机床上的加工顺序都相同, 同时满足以下约束:

- 所有工件在零时刻可以被加工;
- 一个工件在某时刻只能在一台机床上加工;
- 一台机床在某时刻只能加工一个工件;
- 同一工件的所有工序必须连续加工, 即要求任一工件在某台机床上的加工完成时间等于其在下一机床上的加工开始时间;
- 工件的运输时间和机床的启动时间包含在加工时间内。

无等待流水车间调度模型如图 1 所示。各工件在各机床上加工时间已知, 问题是如何安排生产, 使调度方案的总流水

收稿日期: 2009-12-22; **修回日期:** 2010-03-04 **基金项目:** 国家自然科学基金资助项目 (60874075, 70871065); 山东省教育厅科技发展计划资助项目 (J09LG29); 聊城大学基金资助项目 (08x013)

作者简介: 董斌 (1982-), 男, 山东滨州人, 助教, 主要研究方向为计算机应用; 高开周 (1983-), 男, 山东临沂人, 助教, 硕士, 主要研究方向为智能优化、知识工程 (gaokaizh@yahoo.com.cn); 潘全科 (1971-), 男, 山东聊城人, 教授, 博士后, 主要研究方向为生产调度、智能优化; 孙强强 (1981-), 男, 山东潍坊人, 助教, 硕士, 主要研究方向为计算机应用。

时间最小。

1.2 优化目标的计算

车间调度的优化目标很多,如最小化成本、最小化最大完工时间、最小化工件的平均流水时间、最小化总流水时间、最小化最大或平均延迟时间等。本文以 NWFS 调度问题的总流水时间作为优化目标。总流水时间即从第一个工件的第一道工序开始加工计时,每个工件加工完成时间的和。

记 $\pi = \{1, 2, \dots, i, i+1, \dots, n\}$ 为一次调度结果,如图 2 所示,NWFS 的总流水时间可表示为

$$F(\pi) = \sum_{i=1}^n L_i \quad (1)$$

由于受同一工件的工序必须连续生产加工的限制, NWFS 调度总流水时间的计算不同于一般流水车间调度问题。

令 $t_{i,j}$ 为工件 i 在机床 j 上的加工时间,记 $d_{i,i+1}$ 为满足无等待连续加工要求引起的相邻工件 i 与 $i+1$ 之间的开工时间差,如图 3 所示,则有

$$d_{i,i+1} = \max \left\{ \max_{2 \leq j \leq m} \left\{ \sum_{k=1}^j t_{i,k} - \sum_{k=1}^{j-1} t_{i+1,k} \right\}, t_{i,1} \right\} \quad (2)$$

在引入 $d_{i,i+1}$ 后,NWFS 总流水时间的评价可由 $d_{i,i+1}$ 和各工件在各机床上的加工时间表示。如图 2、3 所示,调度 π 的总流水时间为

$$F(\pi) = \sum_{i=1}^n L_i = \sum_{i=1}^{n-1} (n-i) d_{i,i+1} + \sum_{i=1}^n \sum_{j=1}^m t_{i,j} \quad (3)$$

2 混合差分进化算法

2.1 差分进化算法

DE^[11]是由 Storn 等人于 1995 年提出的一种新兴的进化计算技术。DE 是基于群体进化的算法,具有记忆个体最优解和种群内信息共享的特点^[11]。DE 算法的基本思想如下:

- a) 初始化种群 $X = [x_1, x_2, \dots, x_{Np}]$, Np 为种群规模,个体 $x_i = [x_{i,1}, x_{i,2}, \dots, x_{i,D}]$ 表示一个问题解, D 是问题的维数。
- b) 对当前种群进行变异操作,得到变异个体 v_i^{t+1} 。
- c) 对变异个体做交叉操作,得到实验个体 u_i^{t+1} 。
- d) 对实验个体和原种群的个体进行评价,择优选择,产生最终的新一代种群。
- e) 判断结束条件,满足结束条件则停止;否则,转 b)。

2.2 LPV 规则

LPV (largest position value)^[12] 规则是一种利用连续算法解决离散问题的数据映射方法。本文利用 LPV 规则实现 DE 解决离散的调度优化问题。

如表 1 所示,个体 $x_i = [x_{i,1}, x_{i,2}, \dots, x_{i,D}]$ 的每一维对应一个连续数,个体的维数也是调度工件的个数。假设初始状态工件的序号就是个体 x_i 中每一维的序号,对个体 x_i 中的所有维进行非递增的排序。排序完成后,个体 x_i 中的维数形成一个序列,也即一个工件的调度,如表 2 所示。如此,LPV 规则实现了从个体 x_i 中的连续值转换为离散工件序列的操作。

表 1 个体 x_i 的值

值	维/工件 j							
	8	1	2	3	4	5	6	7
$x_{i,j}$	1.2	6.0	5.1	2.3	0.9	4.4	2.8	3.7

表 2 执行 LPV 规则后得到的工件序列

序列	$x_{i,j}$ 非递增排序							
	6.0	5.1	4.4	3.7	2.8	2.3	1.2	0.9
工件序列	2	3	6	8	7	4	1	5

2.3 邻域搜索

2.3.1 插入邻域搜索

HDE 算法在 DE 执行过程中,对每次调度的工件序列 π 执行插入邻域搜索^[13],步骤如下:

- a) π 中第一个工件分别插入到其他 $m-1$ 个位置,并得到序列 π^* ,评价 π^* 的目标值;
- b) 如果 $F(\pi^*) < F(\pi)$,则 $\pi = \pi^*$;
- c) 对 π 中其他工件分别做 a) 和 b) 中的操作。

2.3.2 基于 pairwise 的局部搜索

基于 pairwise 的局部搜索^[14]被证实对排列类型的解具有较好的局部改善作用。在 DE 执行完成后,对最优解执行 LPV 规则,转换为工件序列 π 。对工件序列 π 执行基于 pairwise 的局部搜索,主要步骤如下:

- a) 将第一位的工序与后续的工序依次进行交换。如果交换后的调度总流水时间得到改善,则交换这两个工序。
- b) 对剩余的工件,逐一执行 a) 中的操作。

2.4 HDE 算法

HDE 算法是一种将 DE 算法、插入邻域搜索算法和基于 pairwise 的局部搜索算法整合的混合算法,如图 4 所示。

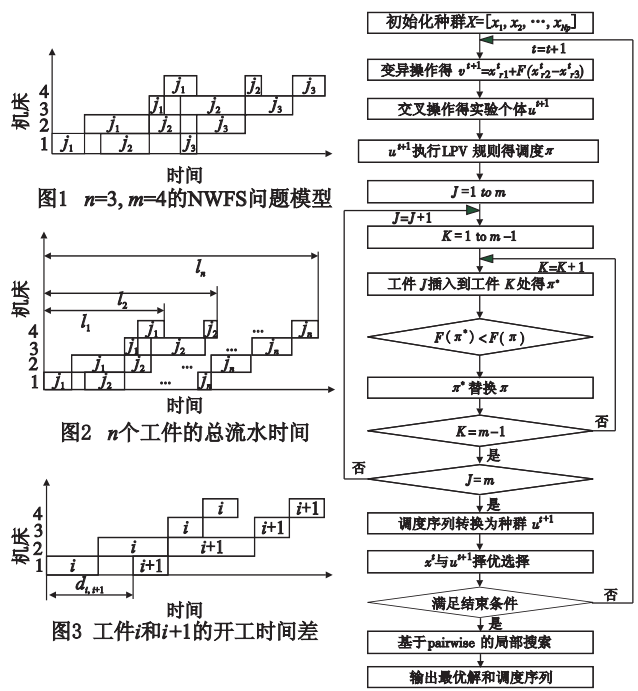


图 4 HDE 算法流程

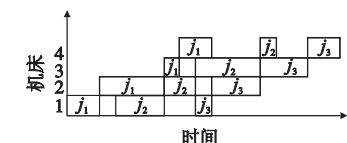


图 1 n=3, m=4 的 NWFS 问题模型

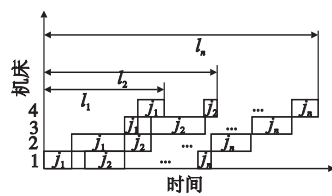


图 2 n 个工件的总流水时间

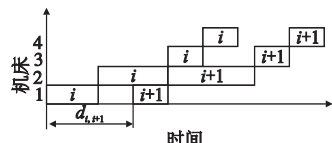


图 3 工件 i 和 i+1 的开工时间差

HDE 算法的具体步骤如下:

a) 初始化 DE 算法的种群 $X = [x_1, x_2, \dots, x_{Np}]$ 和参数;对种群中个体执行 LPV 规则,转换为工件序列 π ,并评价相应的总流水时间。

b) 变异操作。在 t 时刻的任一个体 x_i^t 通过变异操作,得到相应的变异个体为

$$v_i^{t+1} = x_{i1}^t + F(x_{i2}^t - x_{i3}^t) \quad (4)$$

其中: $r_1, r_2, r_3 \in \{1, 2, 3, \dots, Np\}$ 互不相同且与 i 不同; x_{i1}^t 为父代基向量; $x_{i2}^t - x_{i3}^t$ 为父代差分向量; F 为缩放因子。

c) 交叉操作。对变异个体 v_i^{t+1} 实施交叉操作,得到实验个体为

$$u_{i,j}^{t+1} = \begin{cases} v_{i,j}^{t+1} & \text{if } (\text{rand}(j) \leq CR) \\ x_{i,j}^t & \text{otherwise} \end{cases} \quad (5)$$

其中:rand(j)为[0,1]均匀分布的随机数;CR为[0,1]的交叉概率。

d)对新种群中的个体执行 LPV 规则,转换为调度序列 π 。

e)对调度 π 执行插入邻域搜索,完成后再转换为种群个体。

f)对于 NWFS 问题,依据目标函数对原种群中的个体 x_i^t 和实验种群中的实验个体 u_i^{t+1} 分别进行评价,选择最优的一个作为新种群个体 x_i^{t+1} ,即

$$x_i^{t+1} = \begin{cases} u_i^{t+1} & \text{if } (F(\pi^*) < F(\pi)) \\ x_i^t & \text{otherwise} \end{cases} \quad (6)$$

g)判断。如果满足结束条件转步骤 h),否则转步骤 b)。

h)对上述步骤得到的最优调度 π^* 执行基于 pairwise 的局部搜索:

```
for (i = 0; i < n; i++)
{
  for (j = 0; j < n - 1; j++)
  {
    当 i ≠ j 时,交换第 i 个和第 j 个工件位得到调度  $\pi^{**}$ ;
    if (F( $\pi^{**}$ ) < F( $\pi^*$ )), 则  $\pi^* = \pi^{**}$ ;
    对更新后的  $\pi^*$  执行基于 pairwise 的局部搜索;
  }
}
```

i)输出最终的调度序列和最优解。

3 仿真实验与分析

为评价算法的性能,本文采用 31 个典型调度算例进行仿真,并将结果与 PSO、基本 DE 算法作比较,验证了 HDE 算法的有效性。程序运行环境为 VC++ .NET,机器的处理器为双核 Genuine Intel 1.6 GHz CPU,内存为 512 MB,三种算法的种群规模都是 20。在相同的运行环境下,三种算法解决 31 个算例的最优解,如表 3 所示。三种算法对算例 rec39 的优化仿真过程如图 5 所示。由表 3 可知,在相同条件下,HDE 算法比 PSO、基本 DE 算法具有更高的求解质量,而且随着调度规模的增大,HDE 算法的优势越明显。由图 5 可知,HDE 算法比其他两种算法的收敛更快,在相同迭代次数下的解更优。

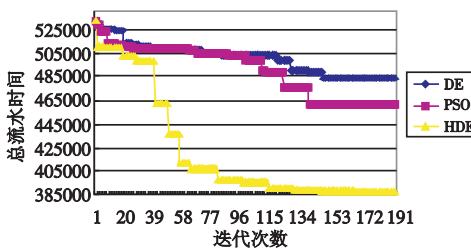


图5 三种算法对算例rec39的仿真趋势图

4 结束语

本文研究了利用 DE 算法解决无等待流水车间调度中总流水时间指标的优化问题;提出一种基于差分进化算法的 HDE 算法。以标准算例作仿真,并通过实验数据和仿真图验证了 HDE 算法优化结果的可行性和有效性。算法相比标准 DE 存在耗时增加的问题。在后续工作中,将重点研究如何提高此类混合算法的鲁棒性,如何降低算法的复杂度,以及算法的进一步改进等问题,并在解决无等待流水车间调度问题及相关的车间调度问题中加以应用。

表 3 三种算法所得最优解

算例名称	n × m	PSO	DE	HDE
Car1	11, 5	54 245	55 955	53 951
Car2	13, 4	61 638	68 768	58 968
Car3	12, 5	65 508	65 199	62 432
Car4	14, 4	79 348	79 604	75 716
Car5	10, 6	60 304	60 497	60 160
Car6	8, 9	53 470	52 946	52 946
Car7	7, 7	36 534	37 061	36 534
Car8	8, 8	53 175	52 912	52 703
hel1	100, 10	54 168	54 833	39 422
hel2	20, 10	2 373	2 476	2 201
rec01	20, 5	19 556	19 938	17 594
rec03	20, 5	17 417	17 869	16 235
rec05	20, 5	19 210	19 055	17 910
rec07	20, 10	28 407	28 841	24 978
rec09	20, 10	26 796	29 254	26 234
rec11	20, 10	25 362	25 657	23 324
rec13	20, 15	36 669	35 091	33 279
rec15	20, 15	35 905	35 035	32 451
rec17	20, 15	35 215	35 563	33 178
rec19	30, 10	59 231	62 458	53 609
rec21	30, 10	57 782	60 206	51 234
rec23	30, 10	56 316	57 992	47 901
rec25	30, 15	76 201	78 315	66 566
rec27	30, 15	73 432	74 699	66 679
rec29	30, 15	72 693	79 649	62 985
rec31	50, 10	153 358	160 666	122 585
rec33	50, 10	159 562	166 772	132 920
rec35	50, 10	160 196	177 408	130 129
rec37	75, 20	455 366	471 108	365 061
rec39	75, 20	462 432	483 443	387 592
rec41	75, 20	469 509	493 196	382 008

参考文献:

[1] GRABOWSKJ J, PEMPORA J. Some local search algorithms for no wait flowshop problem with make span criterion [J]. Computers & Operations Research, 2005, 32(8):2197-2122.

[2] COOK S A. The complexity of theorem-proving procedures[C]//Proc of the 3rd Annual ACM Symposium on Theory of Computing. New York:ACM Press, 1971: 151-158.

[3] GAREY M R, GRAHAM R L, JOHNSON D S. Performance guarantees for scheduling algorithms [J]. Operations Research, 1978, 26(1):3-21.

[4] GONZALEZ T, SAHNI S. Flowshop and jobshop schedules: complexity and approximation [J]. Operations Research, 1978, 26(1):36-52.

[5] JARDINE A K S, LIN Da-ming, BANJEVIC D. A review on machinery diagnostics and prognostics implementing condition based maintenance [J]. Mechanical Systems and Signal Processing, 2005, 20(7):1483-1510.

[6] PAN Quan-ke, 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 International Workshop on UK Planning and Scheduling Special Interest Group. London: City University, 2005:312-411.

[7] ALDOWAISAN T, ALLAHVERDI A. New heuristic for no-wait flow shops to minimize makespan [J]. Computer & Operation Research, 2003, 30(8):12192-12311.

[8] SCHUSTER C J, FRAMINAN J M. Approximative procedure for no-wait job shop scheduling [J]. Operations Research Letters, 2003, 31(4):308-318.

别的最优特征子集,形成分类器的训练模板即用户模板,使用改进后的贝叶斯公式进行文本分类。

1) $P(w_i|C_j)$ 改进后的效果验证

为验证改进后的特征词的类条件概率计算方法是否有效,对单独使用词频、文频、改进后的方法来计算特征词的类条件概率所得到的分类性能进行了比较,实验结果分别如表 1 和 2 所示。

表 1 三种计算方法得到的封闭性能比较 %

类别	使用词频 计算 $P(w_i C_j)$	使用文频 计算 $P(w_i C_j)$	改进后的 方法
农业	77.09	86.58	87.44
环境	84.03	84.62	84.21
政治	82.09	87.36	82.73
军事	84.31	85.75	86.83
招聘	94.31	92.95	94.73
教育	83.97	83.62	84.12
计算机	79.37	77.64	79.9
财经	80.82	81.6	82.52

表 2 三种计算方法得到的开放性能比较 %

类别	使用词频 计算 $P(w_i C_j)$	使用文频 计算 $P(w_i C_j)$	改进后的 方法
农业	79.02	87.3	88.84
环境	87.73	87.56	87.53
政治	83.33	87.98	84.34
军事	84.82	84.77	85.58
招聘	95.91	94.23	96.35
教育	84.54	83.78	84.49
计算机	81.62	79.39	81.12
财经	82.12	82.61	82.67

由表 1 和 2 可知,不论是从封闭性能还是从开放性能看,使用文档频率和改进后的方法来计算特征词的类条件概率得到的分类性能均优于单独使用词频来计算词的条件概率所得到的性能。虽然在某些类别上,使用文频计算得到的性能和使用改进后的方法得到的结果相差不大,但总体来看,使用改进后的方法优于其他两种方法,这主要是因为改进后的方法将词频和文频这两种因素都进行了考虑。

2) 分类器封闭性能、开放性能评估

对传统的贝叶斯及其改进后的方法分别进行封闭性能测试和开放性能测试。在进行开放性能测试时,为了更加客观地评价其性能,将选取的 1 760 篇测试文本混合放在一个文件夹下来进行分类,然后根据分类结果将各个待测文本放入相应的类别文件夹下。分类器的性能测试结果分别如图 1 和 2 所示。

图 1 和 2 得到的结果基本一致,即改进后的 Bayesian 分类

器相比其他三种方法有较好的分类性能。使用 Laplace 平滑技术得到的分类性能与使用 Lidstone 平滑技术得到的性能相差不多,两者走向几乎趋于一致,在图 2 中甚至达到几乎重合的地步。这主要是它们所使用的平滑公式基本相同所造成的。在使用传统的朴素贝叶斯分类器进行分类时,由于单类别词汇的统治作用使各个类别的 F_1 值很低,几乎趋于零。总之,由图 1 和 2 可知,改进后的朴素贝叶斯分类器相比其他几种方法具有良好的性能。

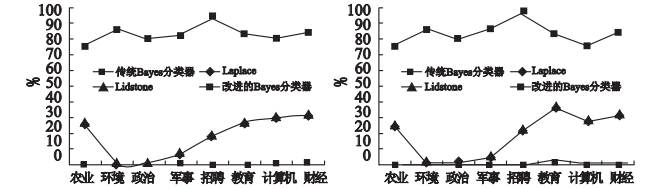


图1 四种分类方法的封闭性能评估结果比较

图2 四种分类方法的开放性能评估结果比较

5 结束语

分类器性能的好坏受很多因素影响,使用的训练语料就是其中的一个重要因素。虽然本文使用的分类器并不能得到目前最好的分类效果,但它在某些类别上的分类效果还是很显著的,并且该分类器也为单类别词汇问题的解决提供了另一种行之有效的方法。下一步的工作是在大型数据库中对改进的朴素贝叶斯公式进行测试,并对存在的不足之处继续进行探讨研究,进而提高算法的准确率和召回率。

参考文献:

[1] 阮彤,冯东雷,李京.基于贝叶斯网络的信息过滤模型研究[J].计算机研究与发展,2002,39(12):1564-1571.

[2] 张宁,贾自艳,史忠植.使用 KNN 算法的文本分类[J].计算机工程,2005,31(8):171-185.

[3] 陈剑敏.基于 Bayes 方法的文本分类器的研究与实现[D].重庆:重庆大学计算机学院,2007.

[4] 姜沛佩,刘培玉.一种新的应用于文本特征子集优化的 GATS 算法[J].计算机工程与应用,2009,45(30):121-122.

[5] NEAL R M, HINTON G E. A new view of the EM algorithm that justifies incremental, sparse, and other variants [M]//Learning in Graphical Models. Cambridge, MA:MIT Press, 1998:355-368.

[6] 王小平.遗传算法—理论、应用与软件实现[M].西安:交通大学出版社,2002.

[7] 贾兆红,陈华平.基于改进遗传算法的权重发现技术[J].计算机工程,2007,33(5):156-157.

[8] 庞秀丽,冯玉强,姜维.贝叶斯文本分类中特征词缺失的补偿策略[J].哈尔滨工业大学学报,2008,40(6):956-970.

[9] 袁方,苑俊英.基于类别核心词的朴素贝叶斯中文文本分类[J].山东大学学报,理学版,2006,41(3):46-49.

[12] PAN Quan-ke, SUGANTHAN P N, TASGETIREN M F. A harmony search algorithm with ensemble of parameter sets [C]//Proc of the 11th IEEE Conference on Congress on Evolutionary Computation. Torndheim, Norway: IEEE Press, 2009:1815-1820.

[13] MLADENOVIC N, HANSEN P. Variable neighborhood search [J]. Computers in Operations Research, 1997, 24(11):1097-1100.

[14] ALDOWAISAN T, ALLAHVERDI A. New heuristics for no-wait flowshops to minimize makespan [J]. Computer & Operation Research, 2003, 30(8):1219-1231.

(上接第 2877 页)

[9] 潘全科,赵宝华,屈玉贵.无等待流水车间调度问题的优化[J].计算机学报,2008,31(7):1147-1154.

[10] 刘波,王凌,金以慧.差分进化算法研究进展[J].控制与决策,2007,22(7):721-729.

[11] STORN R, PRICE K. Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces [R]. Berkeley: University of California, 2006.