

基于改进智能水滴算法求解流水车间干扰管理

刘 欣, 叶春明

(上海理工大学 管理学院, 上海 200093)

摘 要: 流水车间问题(Flow-Shop Problem)是一类典型的生产调度简化模型, 加入干扰因素的 FSP 问题具有较高的实际应用价值. 智能水滴算法(IWD)是一种较新颖的群智能算法, 针对算法存在的早熟问题, 文中结合变邻域搜索算法, 从而有利于在寻优时跳出局部最优, 很大程度上提高了搜索效率和性能. 构建了干扰为新工件突然到达的调度模型, 选取 CAR8 作为数据进行测试分析和仿真, 证明了该算法的可行性.

关键词: 改进的智能水滴算法; 变邻域搜索; 干扰管理; 流水车间

中图分类号: TP18

文献标识码: A

文章编号: 1000-7180(2014)09-0042-06

Based on Improved Intelligent Water Drops Optimization for Disruption Management on Flow-shop Scheduling

LIU Xin, YE Chun-ming

(Business School, University of Shanghai for Science & Technology, Shanghai 200093, China)

Abstract: Flow shop problem (FSP) is a kind of typical scheduling model, which has higher practical value when added interfering factors. Intelligent water drops optimization (IWD) is a kind of new swarm intelligence algorithm. In order to resolve the inadequacies when solve FSP, the algorithm combined with Pairwise local search. This approach not only help algorithm escape from local optimization, but also improves the efficiency and performance of the algorithm. By constructing disruption management model which is a new job sudden arrival scheduling, and doing a simulation of the CAR8 problem, it turns out that the proposed algorithm is feasible and effective.

Key words: improved intelligent water drops optimization; variable neighborhood search; disruption management; flow-shop

1 引言

流水线调度问题(FSP)是一种最典型的生产调度模型问题, 同时也是研究人员研究领域最多的 NP 完全问题. 目前有很多方法来解决 FSP 比如较为常见的 CDS 算法、神经网络法、遗传算法^[1]、粒子群算法^[2]等.

大多数智能算法都是模仿自然界某个领域的行为. 智能水滴算法(Intelligent water drops algorithm, IWD)是受到自然界的水群流动过程的启发所形成的一种最新研究成果. 该算法是由 Shah

Hosseini 于 2007 年提出^[3]. 在 2008 年 Haibin Duan 等人利用该算法成功解决机器人路径规划问题^[4], 在 2010 年 Iman Kamkar 等人运用智能水滴算法解决了 VRP 问题^[5]. 目前该算法已成功解决 TSP 问题, 多维背包问题(MKP), 在移动中的通信应用^[6]以及多阈值的灰度级图像. 本文提出一种改进的智能水滴算法, 通过对基准数据 CAR 类测试并与 PSO 算法相比较, 同时测算了在不同加权系数情况下 CAR8 加干扰的问题, 仿真结果进一步证实了改进后算法的有效性.

目前大多数所研究的流水线调度都是在理想的

收稿日期: 2013-11-13; 修回日期: 2014-01-08

基金项目: 国家自然科学基金资助项目(71271138); 教育部人文社会科学规划基金项目(10YJA630187); 上海市教育委员会科研创新项目(12ZS133); 上海市一流学科项目(S1201YLXK)

状态下,并没有考虑外来因素.然而在实际生产中,往往有许多不确定的事件发生,很容易打乱原定计划的生产调度,针对这种情况,干扰管理(disruption management)最早由丹麦学者 Clausen 提出,干扰管理^[7]的定义是在干扰发生后,要使得原计划尽可能地回归到最初的目标上,并且让新计划相对于原目标所产生的扰动及影响最小,对于企业来说应该把所造成的各种损失成本和费用考虑进新方案里.

近年来对于干扰管理的研究越来越多,比如: Akturk 等运用了经典的匹配算法求解 flow-shop 调度下的机器故障问题. Qi 等^[8]提出了建立原方案和新方案的扰动偏差最小化的数学模型,应用最短加工时间(shortest processing time, SPT)规则的干扰调度问题,针对出现的工件及机器干扰情况,运用动态规划来求得最优解. Abumaizar 等研究学者研究了在多种干扰源发生时如何有效地解决 job-shop 的调度问题. Lee 等^[9]对于未来得及按时完工的工件的处置情况,提出了2种解决方式:可将没完工的工件放在其他闲置的机器上进行加工,这样就会产生多余的生产费用及运输时间也可考虑先修复好受到干扰的机器,然后再对需要的工件继续进行加工处理.对于以上的解决方法需建立包括原始的费用函数,在调度中所产生的运输费用以及相对于原方案所造成的偏差的干扰费用最小化的数学模型.

如何把干扰管理运用在流水车间调度中,也能保障生产系统同样有效的运转能力.这将成为以后研究领域的一个关键问题.

2 Flow-shop 干扰管理调度模型

在实际生产流水线里通常碰到一种情况: n 个不同工件需在 m 台机器上进行加工处理,一个工件要完全加工完毕,需要进行多步操作,并且各工件的加工路线是完全一致的.假设在原计划实施之前所有工件到达生产车间的时间相同.那怎样对这些工件进行加工排序,最终得到企业所需要的加工总完工时间最小,这就是常见的流水线排序问题^[10].干扰管理调度是在已完成的原始工件集合调度的基础上,新的一批工件在加工前突然到达,使得整个生产调度受到干扰,这时的目标就该包括生产加工的所有工件的总流程时间和原始工件受到的干扰时间差的总和.针对这种问题,假设:

- (1) 每个工件在所有机器上的加工顺序一样;
- (2) 一个工件同一时间只能在一台机器上

加工;

- (3) 每台机器在同一时间只能处理一个工件;

- (4) 各工序的加工时间和准备时间是给定的;

设 $P_{i,j}$ 表示工件 i 在机器 j 上的所需的加工时间; $c(j_i, k)$ 表示在机器 k 上工件 j_i 所需的加工时间, $k = 1, 2, \dots, m$; $c_{\max}(\sigma)$ 表示在调度 σ 中的最大加工完成时间; $J = \{j_1, j_2, \dots, j_n\}$ 表示所有工件的集合调度; π^* 表示所有调度中的最优调度排序;所以当干扰因素出现时,此时的流水线调度问题的数学模型为

$$c(j_i, 1) = P_{j_i, 1} \quad (1)$$

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

$$c(j_i, k) = c(j_i, k-1) + P_{j_i, k} \quad (3)$$

$$k = 2, \dots, m$$

$$c(j_i, k) = \max\{c(j_{i-1}, k) + c(j_i, k-1)\} + P_{j_i, k} \quad (4)$$

$$c(j_n, m) + \alpha \sum |c_s(\sigma) - c_s(\pi^*)| \quad (5)$$

$$s = 1, 2, \dots, m$$

$$\min [c_{\max}(\sigma) + \alpha \sum c_s(\sigma)] \quad (6)$$

该模型中的式(6)为目标函数(最大加工完成时间以及干扰时间差总和).并且最优调度 π^* 是已知的,其中 α 可体现出最初原工件按时完工对整个生产系统的重要性,若 α 越大,说明干扰事件对生产目标的影响就越大.如何合理确定 α 的取值,也是一个值得探讨的问题.

3 智能水滴算法原理及改进

3.1 算法原理

智能水滴(intelligent water drops, IWD)具有两个重要的特性^[11]:

当前水滴所携带的泥土量,表示为 Soil (IWD).

水滴流动中所具有的速度,表示为 Velocity (IWD).

对每个智能水滴来讲,这两个特性都是随着自身的流动而不断变化的,该算法的目标是按照某些约束条件找到从起点到终点一条最佳路径.

在实际环境中水流是连续的.然而在抽象的 IWD 环境中,水滴是离散流动的,水滴从当前位置 i 移到下个位置 j ,其速度的增量为 $\Delta velocity(IWD)$.具体说来,IWD 速度的增量 $\Delta velocity(IWD)$ 与位置 i 到位置 j 之间路径中的泥土量 $soil(i, j)$ 成非线性反比关系.

$$\Delta velocity(IWD) \propto^{NL} \frac{1}{soil(i, j)} \quad (7)$$

这里 $\propto^{NL}$ 来表示非线性正比关系, 并且存在以下关系.

$$\Delta vel^{IWD}(t) = \frac{a_v}{b_v + c_v \cdot soil^{2\vartheta}(i, j)} \quad (8)$$

这里的 a_v, b_v, c_v 和 ϑ 都是用户自己选定的参数.

此外, 在路径当中减少的泥土量相等于 IWD 所增加的泥土量.

$$\Delta Soil(IWD) = \Delta Soil(i, j) \quad (9)$$

$$\Delta Soil(IWD) = \frac{a_s}{b_s + c_s \cdot time^{2\vartheta}(i, j; vel^{IWD})} \quad (10)$$

式中, $a_s, b_s + c_s$ 和都是大于 0 的参数. $time(i, j; vel^{IWD})$ 是 IWD 以速度 vel^{IWD} 从位置 i 移动到位置 j 所花费的时间.

可用式(11)计算出从位置 i 以 vel^{IWD} 移动到位置 j 需花费的时间.

$$time(i, j; vel^{IWD}) = \frac{HUD(i, j)}{vel^{IWD}} \quad (11)$$

式中, HUD(Heuristic Undesirability) 为反向启发式函数. 表示水滴拒绝移动的程度, 与两位置 i, j 之间的距离成正比.

从位置 i 到位置 j 路径当中的泥土量可由以下公式来更新:

$$soil(i, j) = p_o \cdot soil(i, j) - p_n \cdot \Delta Soil(i, j) \quad (12)$$

该表达式中 p_o 和 p_n 通常是小于 1 的正数. 一般满足以下条件: $p_o = 1 - p_n$ 而水滴中所携带的泥土量 $soil^{IWD}$ 按照以下关系更新:

$$soil^{IWD} = soil^{IWD} + \Delta soil(IWD) \quad (13)$$

对于 IWD 而言在可能出现的多条路径时, 每条路径以一定的概率被选择, 一般来讲水滴会更容易选择泥土量相对较少的. $p(i, j; IWD)$ 代表 IWD 在位置 i 时选择 j 为下个位置的概率.

$$P(i, j; IWD) \propto^{L} soil(i, j) \quad (14)$$

假设下个位置概率的解析表达式如下:

$$P(i, j; IWD) = \frac{f(soil(i, j))}{\sum_k f(soil(i, k))} \quad (15)$$

$$f(soil(i, j)) = \frac{1}{\epsilon_s + g(soil(i, j))} \quad (16)$$

式中, 常量 ϵ_s 一个较小的正数, 其功能是确保函数 f 的分母不为 0. 而函数 g 是保证位置 i, j 之间的 $soil(i, j)$ 为非负数, 其表达式为

$$\begin{aligned} & \text{if } \min(soil(i, j)) > 0 \\ & g(soil(i, j)) = soil(i, j) \\ & \text{else} \\ & g(soil(i, j)) = soil(i, j) = \min(soil(i, j)) \end{aligned} \quad (17)$$

其函数 $\min$ 是当前所在位置与所有能选择的下一位置之间的 $soil(i, j)$ 的最小值.

3.2 基于变邻域搜索算法和遗传算法的改进

若用原始的智能水滴算法原理来求解 flow-shop 干扰调度问题时能得到比较满意的解, 但其搜索速度较慢, 经常会陷入局部最优, 为了避免这种早熟现象, 改进其搜索机制, 本文把 IWD 算法和变邻域搜索算法^[12]结合起来, 提出了一种改进后的智能水滴算法 (improved intelligent water drops algorithm, IIWD) 用于求解流水线调度干扰管理问题, 通过混合使用后, 可有助于智能水滴算法在寻优时能跳出局部最优, 从而搜索效率得到了提高. 另一方面, 将成熟的遗传算法中的交叉及变异操作嵌入到智能水滴算法中, 以此来优化水滴的适应值.

3.3 算法流程

设置静态参数及初始值, 设定水滴个数 pop , 最大迭代次数 $maxGen$, 初始速度及水滴初始值为 0, 水滴的可选路径 $path$ 及层数 $level$, 设定静态参数 $a_v, b_v, c_v, a_s, b_s, c_s$ 的取值.

步骤 1 对水滴的位置, 路径代表值和路径中的泥土量进行随机初始化.

步骤 2 根据初始化的位置, 得到第一次迭代水滴适应度值最小的保存在 $fbest$ 中, 相应的工件排序赋给 $indi_best$, 每代最优解用 $Record$ 记录, 对应的最优工件排序记为 $indi_record$. 对所有路径进行评价, 找到最初路径代表值最优的路径保存在 $pathbest$ 中.

步骤 3 然后对初始化的水滴与第一次迭代得到的最优个体进行交叉操作, 这样有利于局部搜索.

步骤 4 根据智能水滴算法的公式, 对每个水滴更新其速度和泥土量, 对满足一定条件的水滴进行相应的变异和进化操作, 同时更新每条路径的泥土量.

步骤 5 水滴循环寻优, 对每个水滴进行变邻域搜索算法中的交换操作, 完成互换操作后比较适应度值, 保留最小的适应度值将其保存为水滴当前的位置和适应度值, 每次迭代后找出种群的最优解, 然后比较当前迭代中适应度值的最优解和上一代的最优解, 如果当前得到适应度值优于上一代的, 则用

当前的替代上一代的。

步骤6 重复执行步骤5直到达到最大迭代次数,最后输出 Record 及 indi_record. 算法结束。

具体流程如图1所示。

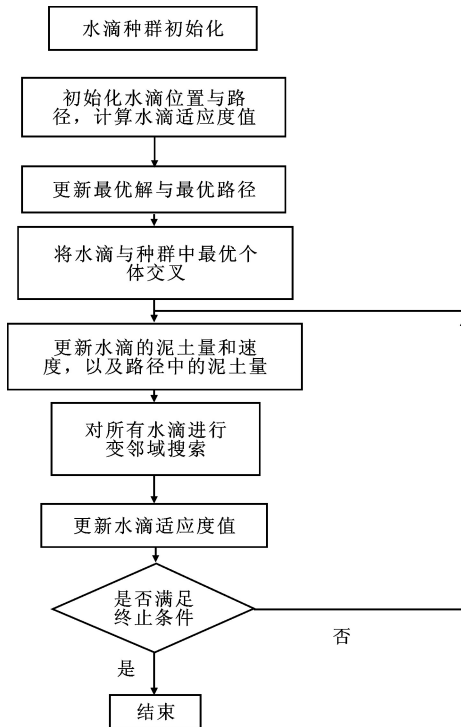


图1 IWD的算法流程

4 改进的智能水滴算法在流水线调度中的应用

为了验证改进后智能水滴算法的有效性和可行性,本节采用经典的基准数据 CAR 类求解流水线调度问题,并分别与原始智能水滴算法和粒子群算法进行比较. 算法编程采用 MATLAB 7.0 的编译软件版本,操作系统为 Windows7. 种群大小为 20, 最大迭代次数为 100, 每个实例都随机独立执行 20 次,图2、图3和图4分别为 PSO,原始 IWD 和改进后 IWD 求解 CAR4 问题时的收敛效果. 比较结果表明,原始 IWD 在求解 FSP 时容易陷入局部最优,故很难得到最优解. 虽然 PSO 和改进 IWD 算法最终都能得到相同的最优解,但是在收敛的速度和平均最优解上,改进的智能水滴算法要优于 PSO.

由图2和图3可知,原始 IWD 算法比 PSO 算法更易早熟,原始 IWD 平均 10.6 代收敛,但基本上得不到最优解,总体而言,其效果劣于 PSO 算法.

针对以上原始 IWD 算法在求解 FSP 问题中既不能得到很好的求解效果且易早熟的问题,图4为 IWD 加入变邻域搜索以及交叉操作后的改进算法,

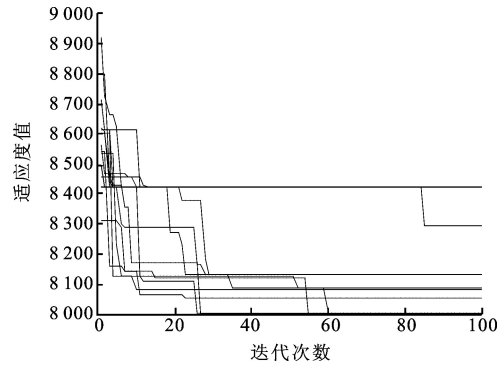


图2 PSO 算法的 CAR4 寻优结果

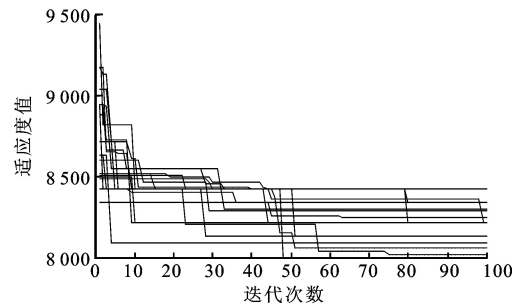


图3 原始 IWD 的 CAR4 寻优结果

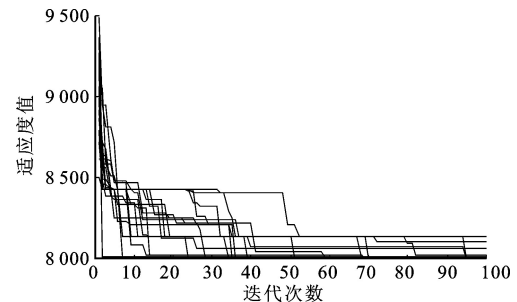


图4 改进 IWD 的 CAR4 寻优结果

用于求解 CAR4 问题时独立运行 20 次的效果收敛图,从图中可以看出,其 20 次运行中 12 次均能达到最优解,而 PSO 算法在 20 次运行中仅有 7 次达到最优解. 故改进 IWD 的优化性明显比 IWD 算法和 PSO 算法好,收敛速度更快并且具有很好的鲁棒性.

5 仿真及测试分析

最初运用原始智能水滴算法求出 CAR8 问题的最优排序 π^* (无干扰时的原计划最优排序)^[13], π^* 为 [7,3,8,5,2,1,6,4], 最优解为 8 366, 接着将新工件的到达加入到 CAR8 数据中 (一般设新工件数 ≥ 1). 最后,运用改进的智能水滴算法求解新方案,从而得到目标函数的最优解^[14-15].

本文将增加新工件(A1-A2)为干扰工件. 加工时间如下:

$$A1=[210,789,123,454,678,267,567,779];$$

$$A2=[427,630,145,120,520,483,852,963];$$

$$A3=[620,258,789,142,345,753,110,654];$$

$$A4=[142,41,876,578,679,420,132,251].$$

如果采用的计算机硬件不同,一般来讲计算时间也会不一样. 所以我们在本文中不把数据的计算时间作为考虑. 本文采用 Windows7 的操作系统和 MATLAB 7.0 的编译软件版本,所需求解的是在干扰下的 $Fm/prmu/C_{\max} + \alpha \sum \Delta_j(\sigma)$ 的调度问题,利用改进智能水滴算法分别对干扰工件 A1,A2,A3 和 A4 在不同加权系数值下进行测试,所得结果如表 1~4 所示.

表 1 不同加权系数的单工件干扰测试

干扰工件	α	最优解	调度排序
A1	0.1	8 930.7	9-7-3-8-5-2-1-4-6
	0.2	9 184.4	9-7-3-8-5-2-1-4-6
	0.5	9 778.5	7-3-8-5-2-1-9-4-6
	0.8	9 778.5	7-3-8-5-2-1-9-4-6
	1.0	9 880	7-3-8-5-2-1-6-4-9
	1.5	9 880	7-3-8-5-2-1-6-4-9
	3	9 880	7-3-8-5-2-1-6-4-9
	5	9 880	7-3-8-5-2-1-6-4-9
	8	9 880	7-3-8-5-2-1-6-4-9

表 2 不同加权系数的两工件干扰测试

干扰工件	α	最优解	调度排序
A12	0.1	10 030	10-7-9-3-8-1-5-2-4-6
	0.2	10 571	10-7-3-8-5-2-1-9-4-6
	0.5	11 413	7-3-8-5-2-9-6-4-1-10
	0.8	11 715	7-3-8-5-2-1-4-6-10-9
	1.0	11 867	7-3-8-5-2-1-4-6-9-10
	1.5	10 852	7-3-8-5-2-1-6-4-10-9
	3	10 852	7-3-8-5-2-1-6-4-10-9
	5	10 852	7-3-8-5-2-1-6-4-10-9
	8	10 852	7-3-8-5-2-1-6-4-10-9

表 3 不同加权系数的三工件干扰测试

干扰工件	α	最优解	调度排序
A134	0.1	10 730	10-7-3-8-5-2-9-4-6-1-11
	0.2	11 348	10-7-3-8-5-1-2-6-4-11-9
	0.5	12 067	7-3-8-5-2-9-6-4-1-10-11
	0.8	12 331	7-3-8-5-2-1-6-11-4-10-9
	1.0	12 083	7-3-8-5-2-1-4-6-11-9-10
	1.5	12 756	7-3-8-5-2-1-4-6-9-11-10
	3	11 352	7-3-8-5-2-1-6-4-11-9-10
	5	11 352	7-3-8-5-2-1-6-4-11-9-10
	8	11 352	7-3-8-5-2-1-6-4-11-9-10

表 4 不同加权系数的四工件干扰测试

干扰工件	α	最优解	调度排序
A1234	0.1	11 395	10-7-9-8-3-5-2-12-1-6-11-4
	0.2	11 733	7-3-8-5-2-1-10-4-6-9-12-11
	0.5	12 461	7-3-8-5-2-1-4-10-6-9-11-12
	0.8	13 643	7-3-8-5-2-9-4-6-11-1-10-12
	1.0	14 253	7-3-8-1-5-2-6-4-10-12-9-11
	1.5	14 433	7-3-8-5-1-2-6-4-9-10-11-12
	3	15 526	7-3-8-5-2-1-9-4-6-10-12-11
	5	15 156	7-3-8-5-2-1-4-6-11-10-12-9
	8	11 757	7-3-8-5-2-1-6-4-10-9-12-11

由上表可知,改进 IWD 算法均可以在有效时间内得到 CAR8 问题加干扰的最优解,并且改进的智能水滴算法的搜索效率更有效,最优解可以在较少的迭代规模搜索中得到. 然而考虑干扰管理的流水线调度时,加权系数 α 的取值会影响目标函数的大小. 当 α 取值大于某个数时,最优解为固定值,而且最优工件排序也是固定的原计划最优排序加上新工件的排序. 由于干扰管理模型的特殊性,为了防止出现干扰管理的作用失效,所以有必要限制加权系数 α 的取值范围,使其目标函数既不过分偏重于干扰时间差和,也不会忽视原目标函数 $C_{\max}(6)$.

6 结束语

本文通过对智能水滴算法的改进,并与成熟的 PSO 算法^[16]求解流水线调度问题的比较,从而证实了改进后 IWD 算法的可行性及有效性,取得了比较满意的优化效果. 由于加入干扰管理的生产调度问题是一种更符合实际生产环境的模型,对以后的生产环境的研究具有一定的意义. 但是利用改进的 IWD 算法求解更大规模复杂调度问题时,其鲁棒性和搜索效率有待提高.

参考文献:

- [1] 黄伟婷. 自适应遗传算法在流水车间优化调度中的应用[J]. 计算机与现代化, 2010(9): 155-157.
- [2] 刘延风,刘三阳. 改进微粒群优化求解置换流水车间调度问题[J]. 计算机集成制造系统, 2009, 15(10): 1968-1985.
- [3] Shah_Hosseini H. Problem solving by intelligent water drops[C]// Evolutionary Computation, 2007. CEC IEEE Congress on. Singapore,IEEE, 2007: 3226-3231.
- [4] Shah_Hosseini H. Intelligent water drops algorithm: A new optimization method for solving the multiple knapsack problem[J]. International Journal of Intelligent Computing and Cybernetics, 2008, 1(2): 193-212.

- [5] Kamkar I, Akbarzadeh-T M R, Yaghoobi M. Intelligent water drops a new optimization algorithm for solving the Vehicle Routing Problem[C]// Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on. Barcelona; University of Pompeu Fabra, IEEE, 2010: 4142-4146.
- [6] Rayapudi S R. An intelligent water drop algorithm for solving economic load dispatch problem[J]. International Journal of Electronics Engineering, 2011, 5(1): 43-49.
- [7] Yu G, Qi X. Disruption management: framework, models and applications[M]. Singapore: World Scientific Publishing Co. Pte. Ltd, 2004.
- [8] Qi X, Bard J F, Yu G. Disruption management for machine scheduling: the of SPT schedules[J]. International Journal of Production Economics, 2006, 103(1): 166-184.
- [9] Lee C Y, Leng J Y, Yu G. Two machine scheduling under disruptions with transportation considerations [J]. Journal of Scheduling, 2006, 9(1): 35-48.
- [10] 潘逢山,叶春明. 生产调度干扰管理模型构建及智能算法研究[J]. 工业工程与管理, 2012, 17(3):85-89.
- [11] 张洪滨. 智能水滴算法及其在通信中的应用[J]. 数据通信, 2012(3):19-21.
- [12] 刘敏,张超勇,张国军,等. 基于混合粒子群优化算法的置换流水车间调度研究[J]. 中国机械工程, 2011, 22(17): 2048-2053.
- [13] 盛晓华,叶春明. 蝙蝠算法在 PFSP 调度问题中的应用研究[J]. 工业工程, 2013, 16(1): 119-124.
- [14] Hoang D C, Kumar R, Panda S K. Optimal data aggregation tree in wireless sensor networks based on intelligent water drops algorithm[J]. IET Wireless Sensor Systems, 2012, 2(3): 282-292.
- [15] Nagalakshmi P, Harish Y, Kumar R K, et al. Combined economic and emission dispatch using intelligent water drops-continuous optimization algorithm[C]// Recent Advancements in Electrical, Electronics and Control Engineering (ICONRAEECE), 2011 International Conference on. India: Divakasi, IEEE, 2011: 168-173.
- [16] 沈海洋. 基于遗传 PSO 的无线传感网络覆盖优化算法研究[J]. 微电子学与计算机, 2013, 30(3): 148-151.

作者简介:

刘欣女,(1990-),硕士研究生.研究方向为生产调度及智能算法.

(上接第41页)

缝的提取作了关键的预处理.最终也成功运用于裂缝检测系统中.

此外,对于公式(1)中的阈值 δ 值暂时还是手动选取,需不断尝试,才能得到最佳值,这就使得算法不够自动化,有待进一步研究改进.

参考文献:

- [1] 何斌. Visual C++ 数字图像处理[M]. 2版. 北京:人民邮电出版社,2002.
- [2] 李严. 多尺度几何分析的图像去噪方法综述[J]. 计算机工程与应用,2011,47(34):168-173.
- [3] 王兴建. 基于多级去噪模型的路面裂缝检测方法[J].

计算机应用,2010,30(6):1606-1609.

- [4] 蔡念. 基于小波神经网络的图像去噪算法[J]. 生物物理学报,2005,25(1):78-81.
- [5] 翟继双. 基于特征的模糊神经网络遥感图像目标分类识别[J]. 遥感学报,2009,13(1):67-73.
- [6] 朱美龙. 小波神经网络在遥感溢油图像去噪中的应用[D]. 大连:大连海事大学, 2011.

作者简介:

罗佳女,(1981-),硕士.研究方向为图像处理与模式识别.

刘大刚男,(1979-),博士,副教授.研究方向为隧道与地下工程.