

# 应用萤火虫算法求解置换流水线问题\*

周季华, 叶春明

(上海理工大学 管理学院, 上海 200093)

**摘要:** 为了更好地研究生产调度问题, 针对置换流水线调度问题, 提出了一种新颖的群智能优化算法, 即萤火虫算法。萤火虫算法模拟了萤火虫通过发光进行信息交流的这种行为特征从而发展演变为一种启发式算法; 并分析了萤火虫算法的仿生原理和数学模型。应用 MATLAB 软件, 对 CAR1 问题在不同的扰动下对算法进行了仿真测试, 并将萤火虫算法和微粒群算法相比较, 仿真结果表明了萤火虫算法优化生产调度问题的可行性和有效性。该算法有待进一步的深入研究。

**关键词:** 置换流水线调度问题; 群智能优化; 萤火虫算法; 仿生原理; 数学模型

**中图分类号:** TP301.6      **文献标志码:** A      **文章编号:** 1001-3695(2013)01-0152-03

**doi:**10.3969/j.issn.1001-3695.2013.01.038

## Using firefly algorithm to solve permutation flow-shop scheduling problem

ZHOU Ji-hua, YE Chun-ming

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

**Abstract:** In order to further study production scheduling problem, this paper proposed a novel group intelligent optimization algorithm named the firefly algorithm for PFSP. FA was a meta-heuristic approach based on the behavior of the flashing characteristics of fireflies. This paper analyzed the bionic principle and mathematical model. Tested the FA through CAR1 problem under the different disturbances by MATLAB. Compared to PSO, the results indicate that the firefly algorithm has better feasibility and validity for solving production scheduling problem.

**Key words:** permutation flow-shop scheduling problem (PFSP); swarm intelligence optimization; firefly algorithm (FA); bionic principle; mathematical model

### 0 引言

流水车间调度问题(flow-shop scheduling problem, FSP), 又称为同序作业调度问题, 是与城市不对称情况下的旅行商问题(TSP)难度相当的同一类型 NP 完全问题中最困难的问题之一。它把许多实际流水线生产调度问题简化成模型, 也是目前车间调度研究中最广泛的一类典型调度问题, 在离散制造工业和流程工业中都具有广泛的应用, 具有一定的代表性。

自从 Johnson<sup>[1]</sup>于 1954 年发表第一篇关于流水车间调度问题的文章以来, 许多学者开始关注流水车间调度问题, 提出了许多解决的方法。寻求最优解的常用方法是整数规划和分枝定界法。由于流水车间调度问题是 NP 完全问题, 对于一些大规模或者中等规模的问题, 整数规划和分枝定界方法并不是很有效, 因此学者们相继提出了一些启发式算法。一般的智能搜索算法, 如神经网络<sup>[2]</sup>、模拟退火算法<sup>[3]</sup>、蚁群算法<sup>[4]</sup>、微粒群算法<sup>[5]</sup>、遗传算法<sup>[6]</sup>等, 已被应用于解决 flow-shop 问题。

萤火虫算法是一种比较新颖的算法, 在生产调度方面的研究刚刚起步, 成果较少。国内文献已有对萤火虫算法的介绍和通过典型的函数优化对算法进行仿真测试, 但是目前这方面的文献还是比较少的。萤火虫算法是模拟自然界中萤火虫成虫发光的生物学特性发展而来的, 也是基于群体搜索的随机优化

算法<sup>[7]</sup>。此算法由剑桥学者 Yang<sup>[8]</sup>提出。本文分析了萤火虫算法的仿生原理, 通过在不同干扰下对经典 CAR1 问题的测试以及对比微粒群算法和萤火虫算法仿真结果, 验证了该算法是可行和有效的。

### 1 置换 flow-shop 问题数学模型表示

考虑最小化最大完成时间的经典置换流水线的调度问题。记  $P_{i,j}$  为工件  $i$  在机器  $j$  上的加工时间(假设各工件的加工准备时间为零或已包含在加工时间  $P_{i,j}$  内),  $\pi = (j_1, j_2, \dots, j_n)$  为所有工件的一个排序,  $\Pi$  为所有排序的集合,  $C(j_i, k)$  为工件  $j$  在机器  $k$  的加工完毕时间。不失一般性, 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为<sup>[9]</sup>:

$$C(j_1, 1) = P_{j_1, 1} \tag{1}$$

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

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

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

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

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

**收稿日期:** 2012-05-17; **修回日期:** 2012-06-28      **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资助项目(10YJA630187); 上海教育委员会科研创新资助项目(12ZS133)

**作者简介:** 周季华(1986-), 女, 山东人, 硕士研究生, 主要研究方向为生产调度(zhouqian104@163.com); 叶春明(1964-), 男, 安徽人, 副院长, 教授, 博士, 主要研究方向为工业工程。

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

### 2 萤火虫算法基本原理

#### 2.1 算法的仿生原理

萤火虫优化算法是基于自然界中萤火虫的发光行为特点而提出的一种启发式优化算法,但是算法中舍弃了萤火虫发光等一些生物学特点,仅仅利用其发光特性来根据其搜索区域寻找伙伴,并向邻域结构内位置较优的萤火虫移动,从而来实现位置进化<sup>[7]</sup>。

在萤火虫算法中,萤火虫相互吸引取决于吸引度和亮度两大要素。萤火虫算法基于三个假设<sup>[10]</sup>:

a)所有的萤火虫都是无性别的,因此一只萤火虫可以被其他任意一只萤火虫所吸引。萤火虫的亮度与自身所在位置的目标值有关,亮度越高说明该萤火虫所处的位置越好。

b)吸引度的大小与它们自身亮度成正比,越亮的萤火虫拥有越高的吸引力。对于任意两只萤火虫,亮度高的萤火虫能够吸引亮度弱的,但是亮度会随着距离的增加而减弱。

c)如果所有萤火虫的亮度一样,它将随机移动,亮度与目标函数相关。

萤火虫算法是模拟萤火虫群体行为的一种优化算法。其仿生原理是:用搜索空间中的点模拟自然界中的萤火虫个体,将搜索和优化过程模拟成萤火虫个体的吸引和移动过程;将求解问题的目标函数度量成个体所处位置的优劣;将个体的优胜劣汰过程类比为搜索和优化过程中用好的可行解取代较差可行解的迭代过程<sup>[7]</sup>。

#### 2.2 算法数学表述与分析

萤火虫算法包含两个要素,即亮度和吸引度。亮度体现了萤火虫所处位置的优劣并决定其移动方向,吸引度决定了萤火虫移动的距离。通过亮度和吸引度的不断更新,从而实现目标优化。从数学角度对萤火虫算法的优化机理进行如下描述<sup>[7]</sup>:

a)萤火虫的相对荧光亮度为

$$I = I_0 \times e^{-\gamma r_{ij}} \tag{7}$$

其中: $I_0$ 为萤火虫的最大萤光亮度,即自身( $r=0$ 处)萤光亮度,与目标函数值相关,目标函数值越优自身亮度越高; $\gamma$ 为光强吸收系数,因为荧光会随着距离的增加和传播媒介的吸收逐渐减弱,所以设置光强吸收系数以体现此特性,可设为常数; $r_{ij}$ 为萤火虫*i*与*j*之间的空间距离。

b)萤火虫的吸引度为

$$\beta = \beta_0 \times e^{-\gamma r_{ij}^2} \tag{8}$$

其中: $\beta_0$ 为最大吸引度,即光源处( $r=0$ 处)的吸引度, $\gamma, r_{ij}$ 意义同上。

c)萤火虫*i*被吸引向萤火虫*j*移动的位置更新由式(9)决定

$$x_i = x_i + \beta \times (x_j - x_i) + \alpha \times (\text{rand} - 1/2) \tag{9}$$

其中: $x_i, x_j$ 为萤火虫*i*和*j*所处的空间位置; $\alpha$ 为步长因子,是 $[0,1]$ 上的常数; $\text{rand}$ 为 $[0,1]$ 上服从均匀分布的随机因子。

算法实现优化的过程是:先将萤火虫群体随机散布在解空间,根据所处的空间位置计算每个萤火虫的亮度,通过比较,亮度高的萤火虫可以吸引亮度低的萤火虫向其移动,移动的距离

主要由吸引度的大小决定。为了加大搜索区域,避免过早陷入局部最优,在位置更新过程中增加了扰动项 $\alpha \times (\text{rand} - 1/2)$ ,根据式(9)来计算更新后的位置。这样通过多次移动后,所有个体都将聚集在亮度最高的萤火虫的位置上,从而实现寻优。

### 3 萤火虫算法流程

a)初始化萤火虫位置,计算萤火虫的目标函数值作为各自最大萤光亮度 $I_0$ 。

b)由式(7)(8)计算群体中萤火虫的亮度*I*和吸引度 $\beta$ ,根据相对亮度决定萤火虫的移动方向。

c)根据式(9)更新萤火虫的空间位置,对处在最佳位置的萤火虫进行随机扰动。

d)根据更新后萤火虫的位置,重新计算萤火虫的亮度。

e)当满足最大搜索次数则转f);否则,搜索次数增加1,转c),进行下一次搜索。

f)输出全局最优解和个体最优解。

算法流程如图1所示。

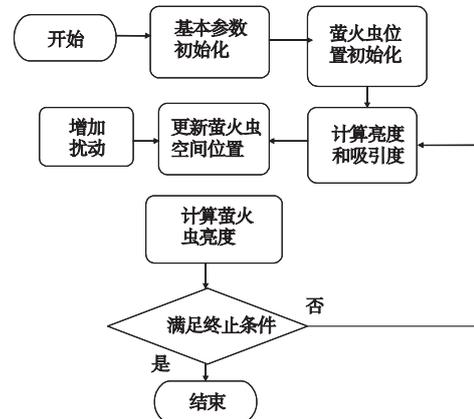


图1 萤火虫算法流程

### 4 仿真实验

本文采用经典 Car1 问题进行实验仿真测试。Car1 问题描述了 10 个工件在五台机器上的加工过程,加工顺序相同。

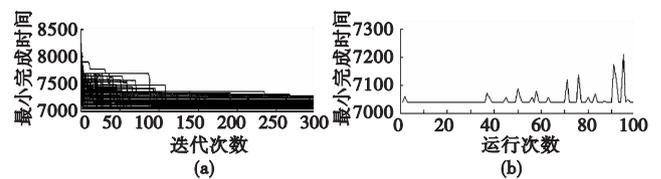


图2 扰动为0.1%时FA对Car1问题寻优结果

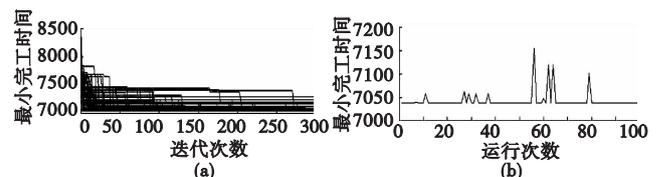


图3 扰动为0.3%时FA对Car1问题寻优结果

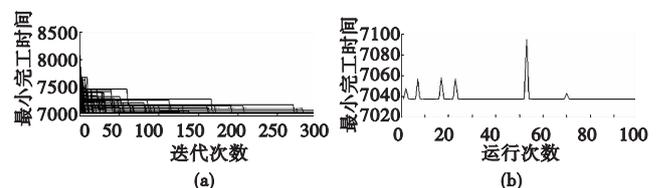


图4 扰动为0.5%时FA对Car1问题寻优结果

采用 MATLAB 编写算法程序,参数设置:萤火虫数目  $m = 40$ ,最大吸引度  $\beta_0 = 1.0$ ,光强吸收系数  $\gamma = 1.0$ ,步长因子  $\alpha = 0.2$ ,最大迭代次数  $\max T = 300$ 。为了验证算法有效性,算法独立运行 100 次。图 2~5 中的(a)表示每次迭代的最优值;图 2~5 中的(b)表示独立运行 100 次,最优值的分布情况。不同扰动下,FA 对 Car1 问题寻优结果如表 1 所示。

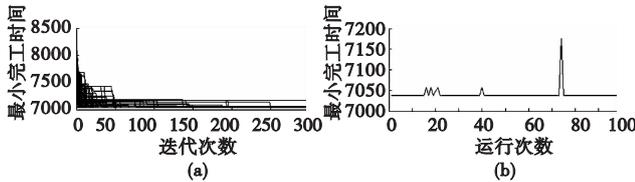


图5 扰动为0.7%时FA对Car1问题寻优结果

表 1 不同扰动下 FA 对 Car1 问题寻优结果

扰动/%	运行次数	最优结果	最优结果 击中次数	最好与 最坏差值	寻优率/%
0.1	100	7 038	82	172	82
0.3	100	7 038	89	116	89
0.5	100	7 038	94	57	94
0.7	100	7 038	94	137	94

从表 1 可以看出,在不同扰动下,最优结果都是一样的,但最优结果击中的次数是不同的。随着扰动的增加,寻优率基本上是呈增大趋势的,扰动为 0.5% 时,最优结果的波动幅度是最小的且寻优率最大。因此,在 FA 测试 Car1 问题时,当扰动为 0.5% 时,得到最佳情况。

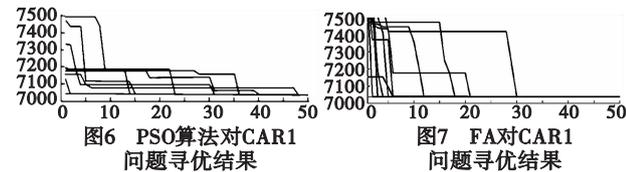


图6 PSO算法对CAR1问题寻优结果

图7 FA对CAR1问题寻优结果

图 6、7 为分别采用基本 PSO 算法和 FA 算法对 CAR1 问题进行仿真测试,两种算法都独立运行了 10 次,每次迭代 50 代。从图中可以得到独立运行 10 次达到最优解 7 038 的迭代次数。从平均来看,PSO 算法在 26 代左右可以达到最优解,FA 在 11 代左右就可以达到最优解。结果表明,虽然都是独立运行 10 次,两种算法都可以全部击中理论最优值(7 038),但 FA 效率明显要高于基本 PSO 算法。

(上接第 129 页)收敛更快,而产生的任务调度序列也具有更小的适应度值,最重要的是调度解完全符合云存储系统需要,不会产生无效解。

下一步计划从理论建模上进一步分析限制解空间之后对调度算法的影响,并在此基础上从候选解的产生、筛选等方面进一步改进,使得算法能更快地逼近最优解,获取较小适应度的解。

参考文献:

[1] HAYES B. Cloud computing [J]. *Communications of the ACM*, 2008, 51(7):9-11.  
 [2] LIN G, DASMALCHI G, ZHU J. Cloud computing and IT as a service: opportunities and challenges [C]//Proc of the 6th IEEE International Conference on Web Services. Los Alamitos: IEEE Computer Society, 2008:1-5.  
 [3] NAMJOSHI J, GUPTA A. Service oriented architecture for cloud based travel reservation software as a service [C]//Proc of the 2009

5 结束语

鉴于生产调度的理论和实际意义,研究生产调度的优化算法一直是很多学者研究的热点,并取得了很多成果。本文提出了应用萤火虫算法来求解置换 flow-shop 型生产调度优化问题,通过对经典的 CAR1 问题仿真测试,以及与基本 PSO 算法相比较,得出萤火虫算法求解生产调度问题是可行和有效的。生产调度的研究已有多年,但由于生产调度问题是极其复杂的 NP-hard 问题,因此求解此类问题一直是难点,还有待进一步的研究。国内外对于萤火虫算法应用到生产调度中的研究是很少的,本文对其作了初步的研究,还存在很多问题,还需进行大量的研究工作。

参考文献:

[1] JOHNSON S M. Optimal two and three-stage production schedules with setup times included [J]. *Naval Research Logistics Quarterly*, 1954, 1(1):61-68.  
 [2] 于海斌,薛劲松,王浩波,等.一种基于神经网络的生产调度方法 [J]. *自动化学报*, 1999, 25(4):449-456.  
 [3] 宋锦和.基于模拟退火算法的生产调度问题 [J]. *长春工程学院学报*, 2004, 5(1):61-63.  
 [4] 姜桦,李莉,乔非,等.蚁群算法在生产调度中的应用 [J]. *计算机工程*, 2005, 31(5):76-78.  
 [5] 王万良,唐宁.微粒群算法的研究现状与展望 [J]. *浙江工业大学学报*, 2007, 35(2):136-141.  
 [6] 曹全军,初红艳,费仁元.启发式算法和遗传算法在生产调度中的应用 [J]. *中国机械工程*, 2006, 17(Z2):211-214.  
 [7] 刘长平,叶春明.一种新颖的仿生群智能优化算法:萤火虫算法 [J]. *计算机应用研究*, 2011, 28(9):3295-3297.  
 [8] YANG Xin-she. Nature-inspired metaheuristic algorithms [M]. [S. l.]:Luniver Press, 2008:83-96.  
 [9] 陈子皓,叶春明,寇明顺.应用新型量子粒子群优化算法求解 PF-SP 问题 [J]. *技术与创新管理*, 2012, 33(2):162-165.  
 [10] YANG Xin-she. Firefly algorithms for multimodal optimization [C]//Proc of the 5th International Conference on Stochastic Algorithms: Foundations and Applications. Berlin: Springer-Verlag, 2009:169-178.

IEEE International Conference on Cloud Computing. Washington DC: IEEE Computer Society, 2009:147-150.  
 [4] ARMBRUST M, FOX A, GRIFFITH R, et al. Above the clouds: a Berkeley view of cloud computing [EB/OL]. 2009. <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html>.  
 [5] TAYAL S. Tasks scheduling optimization for the cloud computing systems [J]. *International Journal of Advanced Engineering Sciences and Technologies*, 2011, 5(2):111-115.  
 [6] HANG Ruay-shiung, HEN Po-hung. Complete and fragmented replica selection and retrieval in data grids [J]. *Future Generation Computer Systems*, 2007, 23(4):536-546.  
 [7] RAHMAN R M, ALHAJJ R, BARKER K. Replica selection strategies in data grid [J]. *Journal of Parallel and Distributed Computing*, 2008, 68(12):1561-1574.  
 [8] CHAUHAN S S, JOSHI R C. QoS guided heuristic algorithms for grid task scheduling [J]. *International Journal of Computer Applications*, 2010, 2(9):24-31.