

置换流水线车间调度问题的研究

刘莹^{1,3} 谷文祥^{2,3} 李向涛³

(东北师范大学人文学院 长春 130117)¹ (长春建筑学院基础教学部 长春 130607)²

(东北师范大学计算机科学与信息技术学院 长春 130117)³

摘 要 随着科学技术的发展和生产规模的不断扩大,置换流水线车间调度问题已经越来越受到广大学者的关注。目前,该领域已经出现了许多优化算法,其大大提高了生产的效率。尽管有一些文献已经进行了比较和综述,但是它们没有包括最新的求解方法和研究结果。因此,文中从一个崭新的视角对该问题进行综述,并详细地对比各种算法,使广大学者对其有一个更新、更全面的了解。

关键词 调度,置换流水线调度,元启发式,最大完成时间,延迟时间

中图法分类号 TP18 **文献标识码** A

Research on Permutation Flow-shop Scheduling Problem

LIU Ying^{1,3} GU Wen-xiang^{2,3} LI Xiang-tao³

(College of Humanities & Sciences of Northeast Normal University, Changchun 130117, China)¹

(Changchun Architecture & Civil Engineering College Department of Basic Subjects Teaching, Changchun 130607, China)²

(School of Computer Science and Information Technology, Northeast Normal University, Changchun 130117, China)³

Abstract As the science and technology develop and the production scale expands constantly, the research on permutation flow-shop scheduling problem has been more and more attention by many scholars. So far, there are a lot of optimization algorithms in the fields, and they greatly increase the production efficiency. There are some literatures which are compared and reviewed, but not including the research of the latest solving methods and results. From a new perspective on the problems in classification, we detailed the comparison of new algorithms, making scholars to have the latest and comprehensive understanding.

Keywords Scheduling, Permutation flow-shop, Meta-heuristics, Makespan, Tardiness

1 引言

早在 20 世纪初期,生产调度问题就已经被企业工程师和生产管理人员所关注。随着科学技术的不断发展,生产规模越来越大,生产的复杂性越来越高,市场的竞争也越来越激烈,因此,对企业的管理和生产过程监控的要求也越来越高。自从 1954 年 Johnson 发表了一篇开创性的文章以来,调度问题吸引了众多学者深入的研究与探讨^[1]。

生产调度是针对一项可以分解的生产任务,探讨在尽可能满足约束条件(如交货期、工艺路线和资源情况等)的前提下,通过下达生产指令,安排其组成部分(操作)使用哪些资源、其加工时间及加工顺序,以获得生产任务执行时间或成本的最优化^[2]。

按照加工模式,生产调度一般分为:单机调度问题、并行机调度问题、Job-shop 调度问题、Flow-shop 调度问题和 Open-shop 调度问题。Flow-shop 调度问题是许多流水线生

产调度问题的简化模型,与实际生产密切相关,是目前研究较为广泛的一种调度问题。置换流水线车间生产调度问题(Permutation Flow-shop Scheduling Problem, PFSP)是 flow shop 问题中一类典型的子问题,也是调度问题的研究热点。

目前,已经有许多国内外学者对 PFSP 进行了综述。2004 年, Framinan 等人对 PFSP 中的启发式进行了详细的分类^[3],并提出了启发式发展的 3 个阶段:指数发展(index development)、解的构造(solution construction)和解的改进(solution improvement)。2005 年, Ruiz 和 Maroto 回顾了前面的算法,在同一个 benchmark 上详细比较和分析了 25 种经典算法,并对各种方法进行了实验测试^[4]。2006 年, Gupta 和 Stafford 对 PFSP 过去 50 年的发展进行了综述,并指出了未来的研究方向^[5]。2007 年, Kalczynski 等人也对多种算法进行了分析,并对文献中的几种算法进行了比较^[6]。2012 年, Pan 和 Ruiz 将各种启发式算法分为简单启发式和复杂启发式两种,并通过实验对这两类启发式进行了对比和分析^[7]。

到稿日期:2013-01-29 返修日期:2013-05-10 本文受国家自然科学基金面上项目(61070084, 60473042),吉林省教育厅“十二五”科学技术研究项目(556)资助。

刘莹(1980—),女,博士生,主要研究方向为智能规划与规划识别、智能调度;谷文祥(1947—),男,教授,博士生导师,主要研究方向为智能规划与规划识别、形式语言与自动机、模糊数学及其应用, E-mail: gwx@nenu.edu.cn(通信作者);李向涛(1984—),男,博士生,主要研究方向为智能调度、智能计算等。

由于 PFSP 的实际应用性很强,许多国内外学者都致力于该方向的研究,并提出了大量的求解算法。总体说来,这些算法大致可以分为 3 类:精确方法(Exact Methods)、构造方法(Constructive Methods)和改进方法(Improvement Methods)^[8]。精确方法主要包括分支定界方法和混合整数线性规划方法。其计算时间过长,所以只适用于解决规模比较小的问题。构造方法主要是简单的启发式方法,该方法可以在较短的时间内生成一个近似的最优解。不过,构造方法求得的解的质量经常没有预期的好。改进方法主要是元启发式,如遗传算法、蚁群算法、粒子群算法和迭代搜索算法等,这些算法都是目前的研究热点。其中,元启发式算法(Meta-Heuristic Algorithm)是启发式算法的改进,它是随机算法与局部搜索算法相结合的产物。元启发式是一种基于直观或经验构造的算法,在可接受的花费(指计算时间、计算空间等)下给出待解决优化问题的每一实例的一个可行解,所以更适用于求解 PFSP 这类 NP-Hard 问题。

综合前面的研究现状,本文将从另一个崭新的角度对 PFSP 求解算法进行分类综述,即按照问题的求解目标进行分类,对比近年来用元启发式算法求解 PFSP 的情况,以便广大学者以一个新的视角去了解 PFSP。

2 PFSP 问题的描述与假设

通常,人们在求解问题时,首先要将问题进行描述,然后才能求解。同时,对于一些难以解决的问题,要先对其做一些前提假设,将问题化简,就可以得到一个比较理想的解,PFSP 也是如此。

2.1 PFSP 问题的描述

企业中的置换流水线车间生产调度问题(Permutation Flow-shop Scheduling Problem, PFSP)是一类典型的 NP-hard 生产调度问题^[9]。在流水线车间中,由于同一台机器上要加工的工件可能有多个,一个工件可能有多道工序要到多台机器上加工,不同工序的加工时间也可能不相同,因此安排工件在机器上不同的加工顺序对最后的结果会影响很大。

置换流水线车间调度问题(PFSP)是一个生产规划的过程,有着很强的工程意义,该问题可以具体描述如下^[10]:

n 个加工任务($N=J_1, J_2, \dots, J_n$)要在 m 台机器($M=M_1, M_2, \dots, M_m$)上加工,每项任务都包含一个操作集合 $J=\{O_{i1}, \dots, O_{jm}\}$,加工任务 J_j 在第 M_i 台机器上的加工时间是给定的,记为 $P_{i,j}(i=1, 2, \dots, m; j=1, 2, \dots, n)$ 。

置换流水线车间生产调度问题(PFSP)用 $n/m/P/C_{\max}$ 表示,即 n 个加工任务/ m 台机器/置换流水线调度/最大完成时间。

任务的加工顺序为 $\pi=\{\pi_1, \pi_2, \dots, \pi_n\}$,其中 n 项任务将依次通过 m 台机器。用 $C(\pi_j, m)$ 表示任务 π_j 在机器 m 上的完成时间。根据加工顺序 $\pi=\{\pi_1, \pi_2, \dots, \pi_n\}$,置换流水线车间生产调度问题的完成时间表示如下:

$$C(\pi, 1) = P_{\pi, 1}$$

$$C(\pi_j, 1) = C(\pi_{j-1}, 1) + P_{\pi, 1}, j=2, \dots, n$$

$$C(\pi_1, i) = C(\pi_1, i-1) + P_{\pi, i}, i=2, \dots, m$$

$$C\pi_j, i = \max(C(\pi_{j-1}, i), C(\pi_j, i-1)) + P_{\pi, i}, j=2, \dots, m$$

那么 Makespan 可以定义为: $C_{\max}(\pi) = C(\pi_n, m)$ 。其中, Makespan 是指最大完成时间。置换流水车间生产调度问题的

目标是找到最合适的排列 $\pi^*: C_{\max}(\pi^*) \leq C(\pi_n, m), \forall \pi \in \Pi$ 。

2.2 PFSP 问题的假设

PFSP 问题的求解目标是确定 n 个加工任务在每台机器上的最优加工顺序,使一个或多个性能指标达到最优。对该问题通常作如下假设^[2,11]:

- (1) 每个加工任务在机器上的加工顺序为 $1, 2, \dots, m$;
- (2) 每台机器同时只能进行一个加工任务;
- (3) 一个加工任务不能同时在不同的机器上进行;
- (4) 工序不能预定;
- (5) 工序的准备时间和加工顺序无关,且包含在加工时间中;
- (6) 加工任务在每台机器上的加工顺序相同,且是确定的;
- (7) 所有机器都是连续可用的;
- (8) 一个加工任务如果已经在台机器上执行,则不允许中断。

1979 年, Gupta 等又在 Johnson 的研究基础上针对任务、机器和操作策略提出了 21 条假设,详见文献[12]。

3 PFSP 问题求解算法的分类

置换流水线车间调度问题是当前调度问题最广泛研究的内容之一,其求解算法也有很多,基本上都是以最小化 Makespan、Total Flowtime 和 Tardiness 为目标,且已经被证明是 NP 完全问题^[9,13]。本文以 PFSP 问题求解的目标进行分类,并进行详细的介绍。

3.1 以 Makespan 为目标的求解算法

目前,在众多的 PFSP 问题求解算法中,以 Makespan 为目标进行求解的算法是最常见的,如 Palmer^[14]、Dannenbring^[15]、Taillard^[16] 和 Framinan^[17] 等,下面将介绍几种近年来以 Makespan 为目标的元启发式算法。

3.1.1 微分进化算法(Differential Evolution, DE)

微分进化算法是 1995 年由 Rainer Storn 和 Kenneth Price 作为一种演化策略提出的^[18]。DE 算法也是一种基于群体进化的启发式算法,其算法通用,原理简单有效,容易实现,且易于与其他算法混合,从而构造出具有更优性能的算法,所以被广泛应用于各个领域。

微分进化算法包含变异和交叉操作,但同时相较于遗传算法的选择操作,微分进化算法采用一对一的淘汰机制来更新种群。算法的原理采用对个体进行方向扰动,以达到对个体的函数值进行下降的目的,同其他进化算法一样,微分进化算法不利用函数的梯度信息,因此对函数的可导性甚至连续性没有要求,适用性很强。

微分进化算法在 PFSP 的问题求解中应用很多,如离散微分进化算法、混合微分进化算法、混合离散微分进化算法和自适应微分进化算法等。

2007 年, Pan 等人提出了一个简单新颖的离散微分进化算法(Discrete DE, DDE)^[19],该算法的解是基于离散任务排序的,其中的目标个体表示为任务序列 $\pi=\{\pi_1, \pi_2, \dots, \pi_n\}$ 。变异个体通过扰动目标群体中的前一代的最优解而获得,因此,微分进化就是以这个方式实现的。由于扰动是随机产生的,因此人们期望在被扰动群体中的每个个体都是可以相互区别的。计算变异个体的公式如下:

$$v_i = \begin{cases} F_k(\pi_g^{-1}), & \text{if } r < P_m \\ \pi_g^{-1}, & \text{otherwise} \end{cases}$$

式中, π_g^{-1} 是来自于目标群前一代中的最优解; P_m 是扰动概率值; F_k 是带有扰动强度 K 的扰动因子; 一致的随机数 $r \in [0, 1]$ 。在接下来的扰动阶段, 获得试验个体的公式如下:

$$u_i = \begin{cases} CR(v_i, \pi_i^{-1}), & \text{if } r < P_c \\ v_i, & \text{otherwise} \end{cases}$$

式中, CR 是交叉因子, P_c 是交叉概率值。最后, 选择出试验个体与目标个体之间适应度值的候选者, 公式如下:

$$\pi_i = \begin{cases} u_i, & \text{if } f(u_i) < f(\pi_i^{-1}) \\ \pi_i^{-1}, & \text{otherwise} \end{cases}$$

Pan 等人提出的这个 DDE 算法的主要思想就是在搜索过程中, 利用从目标群体的前一代中获得的最优解, 来实现 PFSP 问题。

2008 年, Bin Qian 等人提出了一个混合的微分进化算法 (Hybrid DE, HDE)^[20], 用于解决单目标和多目标的 PFSP 问题。由于微分进化算法具有连续的特点, 因此他们在文献 [21] 的基础上又提出了 largest-order-value (LOV) 规则, 将微分进化算法中的个体 $X_i = [x_{i,1}, x_{i,2}, \dots, x_{i,n}]$ 转换成任务序列因子 $\pi_i = [\pi_{i,1}, \pi_{i,2}, \dots, \pi_{i,n}]$ 。依照 LOV 规则, $X_i = [x_{i,1}, x_{i,2}, \dots, x_{i,n}]$ 首先进行降序排列来获得序列 $\varphi_i = [\varphi_{i,1}, \varphi_{i,2}, \dots, \varphi_{i,n}]$, 然后, 通过计算得到 $\pi_{i,\varphi_{i,k}} = k$ 。该混合算法是将基于微分进化搜索和局部搜索结合在一起, 最终可以收敛为全局最优。通过实验对比表明, HDE 具有鲁棒性和有效性。

2011 年, Deng G. D. 和 Gu X. S. 提出了一种新颖的混合离散微分进化算法 (Hybrid Discrete DE, HDDE)^[22], 该算法以无闲置为约束条件, 以最小化 Makespan 为目标, 并在网络表示方法的基础上, 增添了加速方法, 可以将计算复杂度从 $O(mn^3)$ 减少到 $O(mn^2)$ 。HDDE 也是通过变异、交叉和选择 3 种操作来实现的。首先 HDDE 中的个体用任务序列 $\pi = (\pi(1), \pi(2), \dots, \pi(n))$ 来表示, 而群体目标中第 i 个个体、变异群体和试验个体分别用 π_i 、 V_i 和 U_i 来表示。变异个体可以通过下面的公式来获得:

$$V_i = \begin{cases} \text{insert}(\pi_g^{-1}), & \text{if } rand < p_m \\ \text{insert}(\pi_i^{-1}), & \text{otherwise} \end{cases}$$

式中, π_g^{-1} 表示在 $t-1$ 代中相对好的解 π_g , $\text{insert}()$ 表示随机插入操作, p_m 是与 π_g^{-1} 相关的插入变异概率, r 是一个随机整数, 且 $r \in [1, PS]$, $r \neq i$, PS 是种群的大小。 $rand$ 是一个一致的随机数, $rand \in [0, 1)$ 。如果 $rand$ 小于 p_m , 那么插入操作将在 π_g 上执行; 否则, 它将在一个随机的目标个体上执行。试验个体可以通过下面的公式来实现:

$$U_i = \begin{cases} CR(V_i, \pi_i^{-1}), & \text{if } rand < p_c \\ V_i, & \text{otherwise} \end{cases}$$

其含义同文献 [19]。HDDE 与其它算法相比较, 均表现出较优的性能。

3.1.2 遗传算法 (Genetic Algorithm, GA)

遗传算法是在复杂的搜索范围内用于解决全局优化问题的搜索启发式, 其基本思想是根据问题的目标函数构造一个适应度函数, 对一个由多个解 (每个解对应一个染色体) 构成的种群进行评估、遗传运算、选择, 经多代繁殖, 获得适应度值最好的个体作为问题的最优解^[23]。

1995 年, Chen 等人最早将 GA 算法应用于 PFSP, 他们

在初始种群中使用了几个启发式规则^[24]。1996 年, Murata 等人提出了一种采用两点交叉算子、移动变异和精英保留策略的混合 GA 算法^[25], 算法中还混合了局部搜索, 其性能要优于非混合的遗传算法以及禁忌搜索、模拟退火和局部搜索。1998 年, Reeves 和 Yamada 提出了另一个混合遗传算法^[26], 该算法的思想是多步交叉融合 (Multi-Step Crossover Fusion, MSXF), 它把交叉算子和局部搜索合并在一起, 使用其它的父代作为参考, MSXF 算子从一个父代执行了一个偏置的局部搜索。通过邻域结构和距离的测量定义, MSXF 算子和模拟退火算法的退火机制比较类似。实验结果表明, 该算法优于其它元启发式算法。2001 年, Ponnambalam 等人^[27] 使用 GPX (Generalized Position Crossover) 交叉算子以及类似移位的变异和随机的初始化解评估了一种遗传算法。2008 年, Marcelo Seido Nagano 等人在传统遗传算法的基础上, 提出了一种构造性遗传算法 CGA^[28], 其以最小化 Makespan 为目标, 并取得了很好的效果。

此外, 还有很多学者在此方面做了大量的研究工作, 如 Wang L.^[29], Aldowaisan^[30] 和 Ruiz^[31,32] 等。

3.1.3 混合粒子群算法 (Hybrid Particle Swarm Optimization, HPSO)

粒子群算法 (PSO) 最早是由 Kennedy 和 Eberhart 于 1995 年提出的优化算法^[33]。该算法的主要思想是: 首先初始化一群随机粒子 (随机解), 然后通过迭代寻找最优解。在每次迭代过程中, 粒子通过跟踪两个极值来更新自己的速度和位置。一个是粒子本身所找到的最优解, 即个体极值; 另一个是整个种群目前找到的最优解, 即全局极值。另外, 也可以用种群中的一部分作为粒子的邻居, 得到局部极值。PSO 算法简单, 容易实现, 不需要调整很多参数, 已经被广泛应用于各个领域。

标准粒子群的基本算法很简单, 主要有如下几个步骤^[23]:

- Step 1 初始化粒子群, 随机设定每个粒子的初始位置 X_i 和初始速度 V_i ;
- Step 2 计算各个粒子的适应度值;
- Step 3 对每个粒子, 比较它的适应度值和它经历过的最好位置 P_i 的适应度值, 如果比 P_i 好, 更新 P_i ;
- Step 4 对每个粒子, 比较它的适应度值和其邻域内或群体内经历过的最好位置 P_g 的适应度值, 如果比 P_g 好, 更新 P_g ;
- Step 5 调整粒子的速度 X_i 和位置 V_i ;
- Step 6 如果达到结束条件, 则结束; 否则转 Step 2。

假设搜索空间是 N 维, 群体中有 m 个粒子, 那么群体中的第 i 个粒子可以表示为一个 N 维的向量, $X_i = (x_{i1}, x_{i2}, \dots, x_{iN})$, $i=1, 2, \dots, m$ 。算法执行中, 粒子领域内搜索个体的最优位置记作 $P_g^i = (p_{g1}^i, p_{g2}^i, \dots, p_{gN}^i)$, g 为粒子领域内最优粒子的索引。

混合粒子群优化算法在置换流水线车间调度问题上的应用也十分广泛。2009 年, 吉林大学的张长胜博士在毕业论文中提出了多种求解调度问题的混合粒子群算法^[23], 如求解单目标调度问题的 SADPSO 算法、HPGA 算法、AHPPO 算法和 ATPPSO 算法, 还有求解多目标调度问题的 MDPSO 算法。

同年, Kuo I-H 等人提出了一种有效的基于混合粒子群优化模型的调度算法(HPSO)用于求解 FSSP^[34]。由于连续版本的 PSO 在连续空间中具有很好的搜索能力, 因此它是该算法的主要组成部件。此外, 为了改进 PSO 的局部搜索能力, 个体增强方案是该算法的另一个组成部件。同时, 算法还考虑了 PSO 时间变化系数的影响。

1) 首先 HPSO 要进行的是 RK(random-key)编码方案, 通过编码方案可以转换 RK 虚拟空间和 FSSP 解空间的位置。RK 虚拟空间的每一个位置都是一个实数向量, 而 FSSP 解空间的每一个位置都是一个整数向量。假设 RK 虚拟空间中有一个位置 A 表示为 $(\lambda_1, \lambda_2, \dots, \lambda_k)$, 其中, $\lambda_i (1 \leq i \leq k)$ 是任务 i 的对应权值, 然后按照任务索引升序排列这些权值, 排序任务的索引就是 FSSP 解空间的位置。

2) 个体增强方案(Individual Enhancement Scheme, IE)

个体增强方案是要在 FSSP 解空间中完全搜索一个具有长度为 2 的海明距离的个体邻域, 目的是找到一个较优的解。IE 方案的主要思想就是要交换个体的两个任务的顺序, 并比较交换前后的 Makespan 的值。如果交换后的解, 则保持这个改变, 并更新该个体的排序位置, 具体算法详见文献^[34]。

3) PSO 组件

PSO 组件是 HPSO 算法的主要组成部分, 在 RK 虚拟空间中, 每个粒子基于下面这两个公式来移动它们的位置, 并且 PSO 通过 RK 编码方案的转换来评价 FSSP 解中的每一个粒子的解。

$$V_{id} = \omega \times V_{id} + C_1 \times \text{Rand}() \times (P_{id}^{best} - P_{id}) + C_2 \times \text{Rand}() \times (P_{gid}^{best} - P_{id})$$

$$P_{id} = P_{id} + V_{id}$$

式中, V_{id} 是粒子的一步移动距离; ω 是惯性权重, 可以增加或减少粒子一步移动的距离; C_1 和 C_2 是学习因子; $\text{Rand}()$ 可以生成 0-1 之间的随机数; P_{id} 是第 id 个粒子的位置; P_{id}^{best} 是 P_{id} 的最佳经验; P_{gid}^{best} 是第 gid 个粒子的最佳经验。

最后, Kuo I-H 等人将 HPSO 算法与 NPSO、GA 算法进行了比较。实验证明, HPSO 具有较高的收敛速度, 可以更快地找到最优区域。此外, 利用 PSO 对 PFSP 问题进行研究的学者还有: Lian^[35]、Liu^[36]、Liao^[37]等。

3.1.4 蚁群算法 ACO(Ant Colony Optimization)

1998 年 Stutzle 把 ACO 应用到 PFSP 问题上, 他提出了称为 MMAS(Max-Min Ant System)的算法^[38], 后来 Ying 等人在 benchmark 问题上做了测试, 证实该方法非常有效。2004 年, Rajendran 和 Ziegler 提出了两种 ACO 算法来解决 PFSP 问题^[39]。第 1 种算法称为 M-MMAS, 该算法通过合并总结规则和新的局部搜索技巧, 扩展了 Stutzle 的 MMAS 算法。第 2 种算法称为 PACO, 是 M-MMAS 的发展版本。2010 年, Ahmadizar 和 Barzinpour 发展了 ACA 算法^[40], 将其应用于 PFSP 问题中, 并求解最小化的 Makespan。

2012 年, Fardin Ahmadizar 提出了一种新的蚁群算法, 称为 NACA^[41]。该算法是在初始序列的基础上来初始化信息素路径的, 换言之, 这种新的办法与大多数的 ACO 不同, 即在 NACA 开始的时候不会将相等的值分配给所有的信息素

路径, 并且信息素路径的强度会在最低和最高的界限下动态修改。在信息素路径的基础上迭代地应用 Pseudo-stochastic 规则, 一只人工蚂蚁可以构造出一个完整的解。然后, 通过执行局部搜索来改进解的质量。下面介绍 NACA 的通用框架:

- Step 1 设置参数, 生成种子解和初始信息素路径。
- Step 2 当终止条件不满足时, 执行下面的操作:
 - 2.1 对于蚁群中的每一只蚂蚁:
 - a. 重复应用转变规则, 构造一个解;
 - b. 应用局部搜索改进解的质量;
 - c. 对于更新已经改进的解, 直到生成最佳解为止。
 - 2.2 根据全局更新规则, 修改信息素路径;
 - 2.3 更新最低和最高的路径界限值, 用于限制信息素路径。
- Step 3 返回找到的最佳解。

这种新的 ACO 算法是为 PFSP 算法中的 Makespan 标准而提出的。首先, 一个相当好的解可在忽略不计的计算时间里生成, 然后, 信息素路径的强度以该解为基础进行初始化。解的质量取决于一只人工蚂蚁的改进, 而人工蚂蚁的改进是基于任务索引的局部搜索过程来完成的, 该过程结合了选择一个任务插入到另一个任务序列中的阈值概率。一旦所有人工蚂蚁都已经生成了各自的解, 就应用全局更新规则来修改信息素路径。经实验表明, 与一些著名的 ACO 算法相比, 该算法可以产生更好的解。

3.1.5 迭代贪婪算法(Iterated Greedy, IG)

2007 年, Rubén Ruiz 和 Thomas Stützle 提出了一种简单有效的迭代贪婪算法, 用于解决 PFSP 问题^[42]。该算法迭代地应用两个阶段: 毁坏和构造。在毁坏阶段, 对作业序列进行毁坏处理, 随机地、无替换地从作业序列中删除若干作业。在构造阶段, 利用 NEH¹⁾ 的基本原理, 将删除的作业依次重新插入到可能的最好位置, 这样就重新获得了一个完整的作业序列, 然后与毁坏前的作业序列进行比较, 将满足条件的作业序列作为当前解, 进入下一轮迭代, 直到条件满足为止。经过实验对比表明, 该算法不用局部搜索都具有良好的性能。

用于解决最小化 Makespan 的方法还有很多, 如刘长平等提出的萤火虫算法就是一种非常新颖有效的方法^[43]。

3.2 以 Total Flowtime 为目标的求解算法

Total Flowtime 是所有任务在 m 台机器上完成时间的总和。近年来, 以最小化 Total Flowtime 为目标的 PFSP 问题备受关注, 目前已经提出了许多种算法。在 PFSP 中, Total Flowtime 的最小化问题与 Makespan 的一样, 都被证明是 NP-Hard 问题。通常情况下, 应用元启发式解决该问题的方法似乎更为有效, 下面介绍几种以最小化 Total Flowtime 为目标的元启发式算法。

3.2.1 混合启发式方法(Composite Heuristics)

2009 年, Li 等人提出了一种有效的混合启发式算法^[44], 用于处理 Total Flowtime 最小化问题。该算法以 FPE-R (Forward Pair-Wise Exchange-Restart) 和带有迭代方法的 FPE(Forward Pair-Wise Exchange)为基础, 提出了 3 个执行效果出色的混合启发式 ICH1、ICH2 和 ICH3。其主要思想

¹⁾ NEH 算法是 Nawaz, Ensore 和 Ham 在 1983 年提出的一种启发式算法, 几十年来, 经过许多学者的比较, 证明了 NEH 启发式是最具有竞争性的算法, 目前有许多算法都是在此基础上提出来的。

是 Total Flowtime 的计算分为一个不变的子序列和一个改变的序列,改变的部分才需要重新计算,而不变的部分可以继续其父辈的值。这样,计算部分序列要比计算全部序列节省时间,节省时间的多少与优化的目标密切相关。经过实验表明,ICH3 的计算效果是最好的,但是它的总体性能要低于 ICH2 和 ICH3,而 ICH2 的效率介于 ICH1 和 ICH3 之间。

3.2.2 混合遗传算法(Hybrid Genetic Algorithm)

遗传算法也可以很好地解决 PFSP 的最小化 Total Flowtime 问题。2009 年,Zhang 等人提出了一个混合遗传算法 HGA(Hybrid Genetic Algorithm)^[45]。该算法是在初始种群中通过适当的启发式生成一个染色体,即人工染色体,而其它的染色体则是随机生成的。这个人工染色体是通过 WSMGS (Weighted Simple Mining Gene Structure)构造出来的,同时会生成一个新的交叉算子,通过交换父辈的染色体信息,可以生成较优的后代染色体,即较优的解。另外,算法中还运用了两个有效的启发式来改进每一代中生成的染色体。Zhang 等人以基准实例进行测试,HGA 算法获得了 115 个最优解,其中有 92 个是新发现的。

同年,Lin-Yu Tseng 等人提出了一种混合遗传局部搜索算法^[46],该算法将遗传算法与两种局部搜索算法结合起来,用于解决 PFSP 的最小化 Total Flowtime 问题。这两种局部搜索算法在搜索过程中各自扮演着不同的角色:插入搜索负责搜索小的邻域,而带有 Cut-and-Repair 的插入搜索则负责搜索大的邻域。此外,基于正交阵列的交叉操作可以提高遗传算法的强化能力。实验结果表明,这两种搜索算法的优势,使得这种混合基因算法非常具有竞争力。

3.2.3 迭代局部搜索算法(Iterated Local Search)

迭代局部搜索算法是一种元启发式,它提供了一个简单而有效的框架来改进局部搜索方法的性能,目前已经成功地应用于优化问题。2009 年,Xingye Dong 等人首次将迭代局部搜索算法应用于以最小化 Total Flowtime 为目标的 PFSP 中,并提出了 ILS(Iterated Local search)算法^[8]。通过实验测试表明,该算法具有很好的效果。

最近,Xingye Dong 等人提出了一种新的迭代搜索算法 MRSILS(multi-restart ILS)^[47],该算法允许从局部最优抽取的一组解中生成重启解(Restart Solution),并在保持重启解质量的同时扩展搜索空间。经过几次迭代后,算法若从已知最佳解中生成的重启解没有改进,则很难跳出局部最优。因此,算法应该通过简单的干扰来进行一次大的跳跃。文献^[47]将 MRSILS 算法与 6 个最先进的元启发式进行了比较,结果表明 MRSILS 比其中 5 个性能优越,比剩下的 1 个略好一点。经过计算,该算法的时间复杂度仅为 $O(mn^3)$,易于实现。

3.2.4 蚁群算法(Ant Colony Optimization)

蚁群算法不但可以解决以最小化 Makespan 为目标的 PFSP 问题,还可以很好地解决以最小化 Total Flowtime 为目标的 PFSP 问题。Rajendran 等人就将蚁群算法应用其中,并取得了良好的效果。他们在文献^[48]中提出了两种蚁群算法,分别是 ACO1 和 ACO2。ACO1 对 Stuetzle^[38]提出的算法进行了扩展,并运用了基于任务索引的局部搜索的新技术。ACO2 是一种全新的方法,通过实验表明,该算法在处理 PFSP 问题上要优于已有的几个著名的蚁群算法。

用于解决 Total Flowtime 最小化的方法还有很多,如微分进化算法、模拟退火算法等。最近由 Tasgetiren 等人提出的一种离散人工蜂群算法^[49]也取得了很好的效果。

3.3 以最小化 Tardiness 为目标的求解算法

在工业生产中,以最小化产品的交货延误时间为目标的 PFSP 问题越来越受到关注。延误时间简单地说就是一个任务的完成时间超出了它的最后期限。形式化定义如下: d_j 为任务 j 的最后期限, $j \in N$,任务 j 的延误时间定义为 $T_j = \max\{C_j - d_j, 0\}$,其中 C_j 为任务 j 的完工时间。总的延误时间定义为 $\sum_{j=1}^n T_j$ ^[50]。经过众多学者的研究表明,以最小化延误时间为目标的 PFSP 也为 NP-hard 问题。

自 1978 年 Gelders 和 Sambbandam^[51]首次以 tardiness 为目标求解调度问题以来,还有很多学者对该问题进行了深入的研究,如 Kim^[52]、Armntano^[53]、Hasijat 和 Rajendran^[54]等。经过多年的研究与发展,以最小化 Tardiness 为目标的 PFSP 问题大致可分为以最小化 Total Tardiness、Mean Tardiness、Weighted Tardiness 和 Total Weighted Tardiness 等为目标。

2008 年,E. Vallada 等人对作业调度中以最小化延误时间为目标的问题进行了详细的综述^[55],并在一个统一的标准下,以 350 个任务、50 台机器和 540 个问题作为数据集,对 40 种不同的启发式和元启发式进行测试分析,比较其性能的优劣,给出了合理的评价。接下来,我们将给出几种应用元启发式方法来求解以最小化 Total Tardiness 为目标的 PFSP 问题。

3.3.1 模拟退火算法(Simulated Annealing)

2004 年,Hasija 和 Rajendran 应用模拟退火算法^[56]来处理最小化 Total tardiness 问题。该算法通过分派规则产生一个初始解,然后应用模拟退火算法中的两个扰动方案来对其进行改进,一个是与文献^[57]类似的随机微扰方案,另一个是基于任务交换的微扰方案。实验结果表明,该算法与文献^[57]相比可以获得更好的解。

3.3.2 微分进化算法(Differential Evolution)

在 PSFP 问题中,微分进化算法不仅可以解决以最小化 Makespan 和 Total Flowtime 为目标的问题,还可以求解以最小化 Tardiness 为目标的问题,Onwubolu 等人提出的微分进化算法^[58]得到了实现。首先,该算法的初始种群由随机获得的向量参数组成,然后在转换成实际值后应用突变机制,将突变数组转换到离散值来评估,最后选择新的突变因子来代替当前的突变因子。经过实验对比表明,在以 Total Tardiness 为目标的情况下,微分进化算法的性能要比文献^[59]中提出的基因算法更为优秀。

3.3.3 可变贪婪算法(Variable Greedy Algorithm)

2008 年,J. M. Framinan 等人结合迭代贪婪算法和变邻域搜索算法(Variable Neighborhood Search)的特点,提出了一种可变的贪婪算法^[60]。该方法开始于一个随机解,然后通过增强贪婪算法的能力逐步逼近,从而改进这个解。如果发现了一个较好的解,则重复上面的操作来改进它。如果通过贪婪算法没有找到较好的解,则重新找一个随机解进行实验。该算法应用变邻域算法来克服局部最优,并基于放松的贪婪算法通过连续的迭代来增加候选解的数量。在算法的计算过程中,除了需要一些计算时间外,不需要任何参数,因此,在处理不同问题时,不需要做任何的调整。实验表明,该算法是当

时最好的可变量法。

3.3.4 带有路径重连的遗传算法(Genetic algorithms with path relinking)

2010年,E. Vallada 等人针对最小化延误时间问题提出了带有路径重连的遗传算法^[61]。经过大量的计算与统计实验说明,该算法相比最近的一些其它算法表现出了优越的性能。

路径重连是一个搜索技术,它的目标是探索搜索空间或找到连接两个优秀解的路径。在文献^[61]中,作者将路径重连技术应用于探索种群中两个个体之间的搜索空间。在调度问题中,所有任务进行互换移动首先要找到一个解,即起始点,所要达到的目标,即终止点。用这种方式,基于互换移动的路径重连技术就可以完成基于邻域插入的局部搜索过程。

每次移动执行后,所获得的解都会越来越远离起始点,而越来越接近终止点。每次移动都会获得一个延误值,最后,经过所有移动,将所有延误值中最小的值返回。执行路径重连的两个个体可以选择来自当前的种群,也可以选择来自精英解集合。作者提出了4种不同的方案来应用路径重连技术,取得了良好的效果。

4 PFSP 调度问题各种求解算法的比较

求解 PFSP 调度问题有很多种方法,本文以算法的求解目标进行分类汇总,将前面所介绍的元启发式算法进行列表比较,以给广大学者直观的印象,详细的比较情况见表1。

表1 各种算法的对比与评价

年份	作者	目标	算法	评价
2007	Pan 等 ^[19]	Makespan	离散微分进化算法(DDE)	基于离散任务排序
2008	Bin Qian 等 ^[20]	Makespan	混合微分进化算法(HDE)	基于微分进化算法和局部搜索算法
2011	Deng G. D. 等 ^[22]	Makespan	混合离散微分进化算法(HDDE)	在网络表示方法的基础上,又增添了加速方法
2009	Kuo I-H 等 ^[34]	Makespan	HPOS 算法	考虑了时间变化系数,改进了 PSO 的局部搜索能力
2012	Fardin Ahmadiar ^[41]	Makespan	NACA 算法	基于任务索引的局部搜索过程
2007	Rubén Ruiz 等 ^[42]	Makespan	IG 算法	迭代地应用两个阶段:毁坏和构造
2009	Xingye Dong 等 ^[8]	Total Flowtime	迭代局部搜索算法 ILS	算法结构简单,并将几种优化方法结合在一起
2009	Xiao Ping Li 等 ^[44]	Total Flowtime	混合启发式算法:ICH1、ICH2、ICH3	以 FP-ER 和带有迭代方法 FPE 为基础
2009	Yi Zhang 等 ^[45]	Total Flowtime	混合遗传算法 HGA	在遗传算法中,应用启发式来改进解的质量
2009	Lin-Yu Tseng 等 ^[46]	Total Flowtime	混合遗传局部搜索算法	将遗传算法与两种局部搜索算法结合起来
2013	Xingye Dong 等 ^[47]	Total Flowtime	迭代搜索算法 MRSILS	从局部最优抽取的一组解中生成重启解
2005	C. Rajendran 等 ^[48]	Total Flowtime	ACO1 与 ACO2	基于任务索引的局部搜索过程
2004	Hasija 等 ^[56]	Total Tardiness	模拟退火算法 SA	通过派工法则给出初始解并应用局部搜索算法
2006	Onwubolu 等 ^[58]	Total Tardiness	微分进化算法	初始种群随机生成
2008	J. M. Framinan 等 ^[60]	Total Tardiness	可变贪婪算法	应用变邻域与贪婪算法
2010	E. Vallada 等 ^[61]	Total Tardiness	遗传算法	应用路径重连技术

通过列表比较可以得出,目前较为流行的求解方法就是将多种启发式或元启发式结合使用,如微分进化算法和局部搜索算法的结合、遗传算法与两种局部搜索算法的结合等,在取长补短、互利互惠的同时,使算法的性能达到最理想的状态,从而提高工厂车间的生产效率。

结束语 生产调度问题具有很强的实际应用背景,在制造业、企业管理、交通运输、航空航天和现代化的柔性制造系统等众多领域都有着广泛的应用。其中的 PFSP 更是在流水线生产中起着至关重要的作用,因此,国内外有许多学者都致力于该问题的研究,并提出了多种有效的算法。本文从一个崭新的角度,即求解目标,对目前的一些求解算法进行分类对比、分析评价,使广大学者对其有一个更新、更全面的了解,也希望可以为初涉该领域的学者提供些许帮助。

参考文献

- [1] Johnson S M. Optimal two-and three-stage production schedules with setup times included[M]. Naval Research Logistics Quarterly, 1954; 61-68
- [2] 高守玮. 一类 Flow shop 生产调度问题及其优化方法之分析与研究[D]. 上海: 上海交通大学, 2007
- [3] Framinan J M, Gupta J N D, Leisten R. A Review and Classification of Heuristics for Permutation Flow-Shop Scheduling with Makespan Objective[J]. Journal of the Operational Research Society, 2004, 55(12): 1243-1255
- [4] Ruiz R, Maroto C. A Comprehensive Review and Evaluation of Permutation Flowshop Heuristics[J]. European Journal of Operational Research, 2005, 165: 479-494
- [5] Gupta J N D, Stafford E F Jr. Flowshop Scheduling Research after Five Decades[J]. European Journal of Operational Research, 2006, 169: 699-711
- [6] Kalczynski P J, Karbunowski J. On the NEH Heuristic for Minimizing the Makespan in Permutation Flowshops[J]. OMEGA: The International Journal of Management Sciences, 2007, 35: 53-60
- [7] Pan Q-K, Ruiz R. A Comprehensive Review and Evaluation of Permutation Flowshop Heuristics to Minimize Flowtime[J]. Computer & Operations Research, 2013, 40(7): 117-128
- [8] Dong Xing-ye, Huang Hou-kuan, Chen Ping. An iterated local search algorithm for the permutation flowshop problem with Total Flowtime criterion [J]. Computers & Operations Research, 2009, 36: 1664-1669
- [9] Kan R, A H G. Machine Scheduling Problems, Classification, Complexity and Computations 1976[M]// Martinus Nijhoff, The Hague, The Netherlands, 1976: 180
- [10] Pinedo M. Scheduling: Theory, Algorithms and Systems(second ed)[M]. New Jersey: Prentice Hall, 2002: 1118-1122
- [11] Baker K R. Introduction to Sequencing and Scheduling[M]. John Wiley & Sons 1974, New York, 1974: 305
- [12] Gupta J N D. A review of flowshop scheduling research[M]// Ritzman L P, Krajewski L J, Berry W L, et al., eds. Disaggregation Problems in Manufacturing and Service Organizations. Martinus Nijhoff, The Hague, 1979: 363-388
- [13] Garey M R, Johnson, et al. The complexity of flowshop and job-

- shop scheduling[J]. *Mathematics of Operations Research*, 1976, 1:117-129
- [14] Palmer D.S. Sequencing jobs through a multistage process in the minimum total time; A quick method of obtaining a near-optimum[J]. *Operational Research Quarterly*, 1965, 16:101-107
- [15] Dannenbring D.G. An evaluation of flow shop sequencing heuristics[J]. *Management Science*, 1977, 23(11):1174-1182
- [16] Taillard E. Some efficient heuristic methods for the flowshop sequencing problems[J]. *European Journal of Operational Research*, 1990, 47:65-74
- [17] Framinan J M, Leisten R, Ruiz-Usano R. Efficient heuristics for flowshop sequencing with the objectives of Makespan and flow-time minimization[J]. *European Journal of Operational Research*, 2002, 141:559-569
- [18] Storn R, Price K. Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces [R]. Technical Report TR-95-012. ICSI, 1995
- [19] Pan Q-K, Tasgetiren M F, Liang Y-C. A discrete differential evolution algorithm for the permutation flowshop scheduling problem[C]//Proceedings of the genetic and evolutionary computation conference 2007 (GECCO'07). London, England, UK, 2007:126-33
- [20] Qian B, Wang L. A hybrid differential evolution method for permutation flow-shop scheduling[J]. *The International Journal of Advanced Manufacturing Technology*, 2008, 38:757-777
- [21] Bean J. Genetic algorithms and random keys for sequencing and optimization[J]. *ORSA Journal on Computing*, 1994, 6(2):154-160
- [22] Deng G L, Gu X S. A hybrid discrete differential evolution algorithm for the no-idle permutation flow shop scheduling problem with Makespan criterion[J]. *Computers & Operations Research*, 2012, 39:2152-2160
- [23] 张长胜. 求解规划、聚类 and 调度问题的混合粒子群算法研究[D]. 长春: 吉林大学, 2009
- [24] Chen C-L, Vempati V S, Aljaber N. An application of genetic algorithms for flow shop problems[J]. *European Journal of Operational Research*, 1995, 80:389-396
- [25] Murata T, Ishibuchi H, Tanaka H. Genetic algorithms for flow shop scheduling problems[J]. *Computers and Industrial Engineering*, 1996, 30:1061-1071
- [26] Reeves C R, Yamada T. Genetic Algorithms, Path Relinking and the Flowshop Sequencing Problem[J]. *Evolutionary Computation Journal*, 1998, 6(1):230-234
- [27] Ponnambalam S G, Aravindan P, Chandrasekaran S. Constructive and improvement flow shop scheduling heuristics; An extensive evaluation. *Production[J]. Planning and Control*, 2001, 12(4):335-344
- [28] Nagano N S, Ruiz R, Luiz Antonio Nogueira Lorena, A constructive genetic algorithm for permutation flowshop scheduling[J]. *Computers & Industrial Engineering*, 2008, 55:195-207
- [29] Wang L, Zheng D Z. An Effective Hybrid Heuristic for Flow Shop Scheduling[J]. *International Journal of Advanced Manufacturing Technology*, 2003, 21(1):38-44
- [30] Aldowaisan T, Allahverdi A. New heuristics for no-wait flowshops to minimize Makespan[J]. *Computers and Operations Research*, 2003, 30:1219-1231
- [31] Ruiz R, Maroto C, Alcaraz J. New Genetic Algorithms for the Permutation Flowshop Scheduling Problem[C]//Proceedings of The Fifth Metaheuristic International Conference. Kyoto, 2003: 63
- [32] Ruiz R, Maroto C, Alcaraz J. Two New Robust Genetic Algorithms for the Flowshop Scheduling Problem[J]. *OMEGA, the International Journal of Management Science*, 2005, 34:461-476
- [33] Kennedy J, Eberhart R. Particle Swarm Optimization[C]//Proceedings of the 1995 IEEE International Conference on Neural Networks. 1995:1942-1948
- [34] Kuo I-H, Horng S-J, et al. An efficient flow-shop scheduling algorithm based on a hybrid particle swarm optimization model[J]. *Expert Systems with Applications*, 2009, 36:7027-7032
- [35] Lian Z, Gu X, Jiao B. A Similar Particle Swarm Optimization Algorithm for Permutation Flowshop Scheduling to Minimize Makespan[J]. *Applied mathematics and computation*, 2006, 175(1):773-785
- [36] Liu B, Wang L, Jin Y H. An Effective PSO-Based Memetic Algorithm for Flow Shop Scheduling[J]. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 2007, 37(1):18-27
- [37] Liao C J, Tseng C T, Luarn P. A Discrete Version of Particle Swarm Optimization for Flowshop Scheduling Problems[J]. *Computers and Operations Research*, 2007, 34(10):3099-3111
- [38] Stutzle T. An ant approach for the flow shop problem[A]//Proceedings of the 6th European congress on intelligent techniques and soft computing [C]. Aachen: Verlag Mainz, 1998, 3:1560-1564
- [39] Rajendran C, Ziegler H. Ant-colony algorithms for permutation flowshop scheduling to minimize Makespan/Total Flowtime of jobs[J]. *European Journal of Operational Research*, 2004, 155:426-438
- [40] Ahmadizar F, Barzinpour F. A hybrid algorithm to minimize Makespan for the permutation flow shop scheduling problem[J]. *International Journal of Computational Intelligence Systems*, 2010, 3(6):853-861
- [41] Ahmadizar F. A new ant colony algorithm for Makespan minimization in permutation flow shops[J]. *Computers & industrial engineering*, 2012(63):355-361
- [42] Ruiz R, Stützle T. A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem[J]. *European Journal of Operational Research*, 2007, 177(3):2033-2049
- [43] 刘长平, 叶春明. 置换流水车间调度问题的萤火虫算法求解[J]. *工业工程与管理*, 2012, 3(17):56-59
- [44] Li Xiao-ping, Wang Qian, Wu Cheng. Efficient composite heuristics for Total Flowtime minimization in permutation flow shops[J]. *Omega*, 2009, 37:155-164
- [45] Zhang Yi, Li Xiao-ping, Wang Qian. Hybrid genetic algorithm for permutation flowshop scheduling problems with Total Flowtime minimization[J]. *European Journal of Operational Research*, 2009, 196:869-876
- [46] Tseng Lin-yu, Lin Ya-tai. A hybrid genetic local search algorithm for the permutation flowshop scheduling problem[J]. *European Journal of Operational Research*, 2009, 198:84-92

(下转第 22 页)

- ber-Physical Systems Approach[M]. Lee & Seshia, 2011
- [15] Mark-Oliver S, Minyoung K, Carolyn T. Toward distributed declarative control of networked cyber-physical systems[C]//7th International Conference on Ubiquitous Intelligence and Computing. Xi'an, 2010; 397-413
- [16] Jorge R B G, Sergio T K. Architecture for sensor networks in cyber-physical system [C] // Proc. IEEE Latin-American Conference on Communications. Bogota, 2010; 1-6
- [17] Vicaire P A, Hoque E, Xie Zhi-heng, et al. Bundle: A Group Based Programming Abstraction for Cyber Physical Systems[C]// Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems. 2010; 32-41
- [18] Lee E A, Seshia S A. Introduction to Embedded Systems, A Cyber-Physical Systems Approach[M]. Lee & Seshia, 2011
- [19] 胡雅菲, 李方敏, 刘新华. CPS 网络体系结构及关键技术[J]. 计算机研究与发展, 2010, 47(Suppl.); 304-311
- [20] Baheti R, Gill H. Cyber-physical systems[M]// The Impact of Control Technology Washington D C, USA; IEEE, 2011; 161-166
- [21] Bestavros A, Kfoury A, Lapets A, et al. Safe compositional network sketches; formal framework[C]// Proceedings of the 13th ACM International Conference on Hybrid Systems; Computation and Control. New York, USA; ACM, 2010; 231-241
- [22] Kim K H. Desirable advances in cyber-physical system software engineering[C]// Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing. California, USA; IEEE, 2010; 2-4
- [23] Dillon T, Potdar V, Singh J, et al. Cyber-physical systems; Providing Quality of Service(QoS) in a heterogeneous systems-of-systems environment[C]// Proceedings of 5th IEEE International Digital Ecosystems and Technologies Conference(DEST). Daejeon, USA; IEEE, 2011; 330-335
- [24] Wan Kai-yu, Hughes D, Man L L, et al. Composition Challenges and Approaches for Cyber Physical Systems[C]// 2010 IEEE International Conference on Networked Embedded Systems for Enterprise Applications. Suzhou, 2010; 1-7
- [25] Ma Long-hua, Yuan Teng-kai, Xia Feng, et al. A high-confidence cyber-physical alarm system; Design and implementation[C]// 2010 IEEE/ACM International Conference on Cyber, Physical and Social Computing. Stockholm, 2010; 516-520
- [26] Kim K H. Challenges and future directions of cyber-physical system software[C]// 34th Annual IEEE International Computer Software and Applications Conference. Seoul, 2010; 10-13
- [27] Dragomirescu D. Cyber-Physical Systems for aeronautic applications[R]. Presentation Report, 2010. University of Toulouse, France 2010
- [28] Singh J, Hussain O, Chang E, et al. Event Handling for Distributed Real-Time Cyber-Physical Systems[C]// IEEE 15th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing. China, 2012; 23-30
- [29] Huang Jian, Zhang Yan-sheng, Ling Yen I, et al. Real-time service-oriented distributed governance[C]// Proc. 2010 6th World Congress on Services. Miami, 2010; 479-484
- [30] Huang Huang-ming, Tidwell T, Gill C, et al. Cyber-Physical Systems for Real-Time Hybrid Structural Testing; A Case Study [C]// Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems. Stockholm, 2010; 69-78
- [31] Derler P, Lee E A, Sangiovanni-Vincentelli A. Modeling Cyber-Physical Systems[C]// Proceedings of the IEEE special issue on CPS. December 2011
- [32] Lakshmanan K, Niz D, Rajkumar R, et al. Resource allocation in distributed mixed-criticality Cyber-Physical Systems[C]// Proc. of International Conference on Distributed Computing Systems. 2010

(上接第 7 页)

- [47] Dong Xing-ye, Chen Ping, Huang Hou-kuan, et al. A multi-restart iterated local search algorithm for the permutation flow shop problem minimizing total flow time[J]. Computers & Operations research, 2013, 40; 627-632
- [48] Rajendran C, Ziegler G. Two ant-colony algorithms for minimizing Total Flowtime in permutation flowshops[J]. Computers & Industrial Engineering, 2005, 48; 789-797
- [49] Tasgetiren M, Pan Q-K, Suganthan P, et al. A discrete artificial bee colony algorithm for the Total Flowtime minimization in permutation flow shops[J]. Information Sciences, 2011, 181; 3459-3475
- [50] Pinedo M. Scheduling: theory, algorithms and systems (2ed) [M]. Englewood Cliffs, NJ; Prentice-Hall, 2002
- [51] Gelders L F, Sambandam N. Four simple heuristics for scheduling a flowshop[J]. International Journal of Production Research, 1978, 16; 221-231
- [52] Kim Y D. Heuristics for flow shop scheduling problems minimizing meantardiness[J]. Journal of the Operational Research Society, 1993, 44; 19-28
- [53] Armentano V A, Ronconi D. Tabu search for total tardiness minimization in flowshop scheduling problems[J]. Computers & Operations Research, 1999, 26; 219-235
- [54] Hasiya S, Rajendran C. Scheduling in flowshops to minimize total tardiness of jobs[J]. International Journal of Production Research, 2004, 42; 2289-2301
- [55] Vallada E, et al. Minimising total tardiness in the m-machine flowshop problem; A review and evaluation of heuristics and metaheuristics[J]. Computers & Operations Research, 2008, 35; 1350-1373
- [56] Hasiya S, Rajendran C. Scheduling in flowshops to minimize total tardiness of jobs[J]. International Journal of Production Research, 2004, 42(2); 289-301
- [57] Parthasarathy S, Rajendran C. Scheduling to minimize mean tardiness and weighted mean tardiness in flowshop and flowline-based manufacturing cell[J]. Computers and Industrial Engineering, 1998, 34; 531-46
- [58] Onwubolu G, Davendra D. Scheduling flow shops using differential evolution algorithm[J]. European Journal of Operational Research, 2006, 171; 674-92
- [59] Onwubolu G C, Mutingi M. Genetic algorithm for minimizing tardiness in flow-shop scheduling[J]. Production Planning and Control, 1999, 10; 462-71
- [60] Framinan J M, Leisten R. Total tardiness minimization in permutation flow shops: a simple approach based on a variable greedy algorithm[J]. International journal of production Research, 2008, 22; 6479-6498
- [61] Vallada E, Ruiz R. Genetic algorithms with path relinking for the minimum tardiness permutation flowshop problem[J]. Omega, 2010, 38; 57-67