DOI: 10.3901/JME.2015.14.175

求解柔性作业车间调度问题的两级邻域搜索混合算法*

赵诗奎

(济南大学机械工程学院 济南 250022)

摘要: 针对柔性作业车间调度问题(Flexible job shop scheduling problem, FJSP), 以优化最大完工时间为目标, 提出一种融合两级邻域搜索和遗传算法的混合算法。基于通过利用机器空闲时间来减小最大完工时间的想法, 构造邻域结构, 对关键路径上的关键工序进行移动, 实现邻域搜索, 以改进当前解; 设计针对 FJSP 问题特点的两级邻域搜索方式, 第一级邻域搜索为跨机器移动工序, 将工序移动到除当前加工机器之外的其他可选机器上, 第二级邻域搜索为同机器移动工序, 将工序在当前加工机器上进行移动; 给出两级邻域搜索相应的保证可行解工序移动条件; 兼顾 FJSP 问题求解算法的全局搜索能力和局部搜索能力, 利用遗传算法实现全局搜索, 两级邻域搜索实现局部搜索; 采用国际通用的 FJSP 问题基准算例进行测试, 验证了所提方法的有效性。

关键词: 柔性作业车间调度问题; 两级邻域搜索; 邻域结构; 遗传算法

中图分类号: TP301

Bilevel Neighborhood Search Hybrid Algorithm for the Flexible Job Shop Scheduling Problem

ZHAO Shikui

(School of Mechanical Engineering, University of Jinan, Jinan 250022)

Abstract: For the flexible job shop scheduling problem (FJSP), in order to optimize the maximum completion time, a hybrid algorithm mixed with bilevel neighborhood search and genetic algorithm is proposed. The neighborhood structure is constructed by using machine idle time to reduce the maximum completion time. In order to improve the current solution, critical operations of the critical path are moved to achieve neighborhood search. The method of bilevel neighborhood search is designed according to the characteristics of FJSP. The first level neighborhood search is the cross-machine moving operation, and the operation is moved to other optional machines in addition to current processing machine. The second level neighborhood search is the same-machine moving operation, and the operation is moved on current processing machine. Operation moving conditions corresponding to the bilevel neighborhood search are given to ensure feasible solutions. Both of global search ability and local search ability of FJSP solving algorithm are considered, and to use genetic algorithm to achieve global search, bilevel neighborhood search to achieve local search. The internationally accepted FJSP benchmark examples are adopted to test the validity of the proposed method.

Key words: flexible job shop scheduling problem; bilevel neighborhood search; neighborhood structure; genetic algorithm

0 前言

高效的生产调度优化技术对于提高制造系统生产率和设备资源利用率, 缩短产品制造周期具有

十分重要的意义。随着加工技术、自动化技术的发展, 柔性制造系统和数控加工中心等带有一定柔性的生产系统逐渐出现, 具有柔性机器选择的柔性作业车间成为企业应对动态突变市场环境和机器故障等突发事件的有力工具, 柔性作业车间调度问题(Flexible job shop scheduling problem, FJSP)逐渐成为研究的焦点问题。FJSP 问题是传统作业车间调度问题(Job shop scheduling problem, JSP)的延伸, 它突破

* 国家自然科学基金(51405193)、山东省优秀中青年科学家科研奖励基金(BS2014ZZ013)和济南大学博士基金(XBS1427)资助项目。20141011 收到初稿, 20150420 收到修改稿

了机器唯一性约束, 工序可在多台机器上加工, 更加符合生产实际, 同时其解空间的规模进一步扩大, 也是更为复杂的 NP(Non-deterministic polynomial)难问题, 其研究具有重要的理论意义和工程应用价值。

近年来, 以智能算法为代表的求解方法在车间调度问题中得到了成功应用。纵观目前文献中对 FJSP 问题的智能求解算法, 获取高质量解, 求解能力强的高效算法, 总会融入基于邻域结构的邻域搜索技术。邻域结构是对一个给定解, 通过移动工序产生小的扰动, 获得另一个解的机制。MASTROLILLI 等^[1]较早研究了针对 FJSP 问题的有效邻域结构, 在此基础上提出了著名的 M&G 算法。其邻域结构的基本思想是在当前解中移走一个工序, 然后为该工序查找新的合适加工位置, 以改进当前解。在该思想的指导下, 近年来发展了多种更为有效的邻域结构。WANG 等^[2-3]研究了 FJSP 问题移动一个工序邻域结构, 以最大完工时间最小为优化目标, 分别采用分布估计算法和人工蜂群算法求解; 基于邻域结构, YUAN 等^[4]利用和声搜索算法求解 FJSP 问题; GAO 等^[5]将遗传算法和变邻域搜索算法混合求解 FJSP 问题, 定义了单个工序的移动条件; YUAN 等^[6]采用混合差分进化算法求解 FJSP 问题, 对邻域结构移动条件进行了凝练; 张国辉等^[7-8]扩展了文献[9]中的 N5 邻域结构, 交换工序块的块首和块尾相邻的 2 个工序; BOŽEJKO 等^[10]基于 N5 邻域结构, 构造了求解 FJSP 问题的高效双层元启发式算法; LI 等^[11]基于公共关键工序块构造邻域结构, 对块首工序、块内工序和块尾工序分别定义了不同的移动方式; HMIDA 等^[12]基于工序块的概念, 提出了 4 种针对机器选择和工序排序的邻域结构, 研究了求解 FJSP 问题的高效差分进化算法。已有的这些研究成果证明了邻域结构对求解 FJSP 问题的重要性。

为进一步提高 FJSP 问题邻域结构的有效性, 本文对 FJSP 问题邻域结构进行了深入研究, 设计了基于邻域结构的两级邻域搜索方式: 跨机器移动工序实现第一级邻域搜索和同机器移动工序实现第二级邻域搜索。在此基础上, 设计混合算法对 FJSP 问题进行求解。采用国际通用的 FJSP 问题基准算例进行测试, 验证所提方法的有效性。

1 问题描述

FJSP 问题可以描述为 n 个工件在 m 台机器上加工, 按照工件的工艺路线要求, 每个工件由一道

或多道具有顺序约束的工序组成, 每道工序可能有多台可以选择的机器进行加工。加工过程中, 需要满足以下约束条件。

- (1) 同一时刻, 每道工序最多只能选择一台机器进行加工, 一旦开始加工不能中断, 直至加工完成。
- (2) 同一时刻, 每台机器最多只能加工一道工序。
- (3) 同一工件的工序之间存在先后顺序约束, 不同工件的工序之间不存在顺序约束。
- (4) 不同工件之间具有相同的加工优先等级。

本文以优化最大完工时间 $C_{\max}$ 为目标, 目标函数表示如式(1)所示

$$C_{\max} = \min(\max_{i=1}^n C_i) \quad (1)$$

式中, n 为工件总数目; C_i 为工件 J_i 的完工时间。

FJSP 问题包括机器选择和工序排序两个子问题, 调度问题研究的就是为每道工序选择一台加工机器, 并安排每台机器上的工序加工顺序, 从而确定每道工序的开工时间和完工时间, 以优化最大完工时间。如表 1 所示为一个 4×3 FJSP 问题实例, 包含 4 个工件和 3 台机器。当 FJSP 问题中每道工序确定了加工机器之后, 即转化为 JSP 问题。

表 1 4×3 FJSP 问题实例

工件	工序	工序 编号	加工机器及时间		
			M_1	M_2	M_3
J_1	O_{11}	1	2	3*	4
	O_{12}	2	5	6	3*
	O_{13}	3	2*	7	3
J_2	O_{21}	4	2	4	2*
	O_{22}	5	5	2*	3
	O_{23}	6	4*	5	3
J_3	O_{31}	7	5*	8	9
	O_{32}	8	1	2*	5
	O_{33}	9	7	6	5*
J_4	O_{41}	10	7	1*	8
	O_{42}	11	2*	2	7
	O_{43}	12	3*	7	3

注: *为加工时间所对应的机器为每道工序所选择的机器, 此时即对应一个 JSP 问题。

2 混合算法

2.1 两级邻域搜索

邻域搜索的关键在于邻域结构的设计, 邻域结构能够结合问题本身的特征信息, 有效地指导算法的搜索方向, 避免了算法搜索的盲目性, 进而提高算法搜索的效率和质量。对于 JSP 问题, 每道工序

只能在一台机器上加工, 所以, 只需要在工序所在的当前机器上查找空闲时间进行移动(同机器移动工序); 对于 FJSP 问题, 每道工序可能可以在多台机器上加工, 所以, 既可以在工序所在的当前机器上查找空闲时间进行移动(同机器移动工序), 又可以在其他可选机器上查找空闲时间进行移动(跨机器移动工序)。

在文献[13]中, 通过对作业车间调度问题不同解码方式所蕴含机理的分析, 包括半主动调度解码、主动调度解码和全主动调度解码^[14], 进而基于通过利用机器空闲时间来减小最大完工时间的想法, 设计邻域结构。本文在文献[13]针对 JSP 问题研究的基础上, 对基于空闲时间的邻域结构进行了进一步的深入研究, 将其扩展研究应用于求解 FJSP 问题, 设计了针对 FJSP 问题的两级邻域搜索方法, 并研究了相应的保证可行解工序移动条件。

在 FJSP 问题的调度甘特图中, 工序间无时间间隔的最长路径称为关键路径, 对应于析取图模型^[15]中, 从开始节点 0 到结束节点#的最长路径。 $L(u, v)$ 表示析取图中从工序 u 的节点到工序 v 的节点最长路径的长度, 调度的最大完工时间 $C_{\max}=L(0, \#)$ 。组成关键路径的工序定义为关键工序, 否则为非关键工序, 采用文献[16]中的关键路径查找方法。由于最大完工时间取决于关键路径的长度, 所以只对关键工序查找机器空闲时间, 然后进行移动, 实现邻域搜索。

2.1.1 第一级邻域搜索: 跨机器移动工序

跨机器移动工序邻域结构示意图如图 1 所示, 工序 w 为工序块上的一个关键工序, 相关的变量符号解释说明如下: $JP[w]$ 为工序 w 的工件前续工序; $JS[w]$ 为工序 w 的工件后续工序; $s^E(w)$ 为工序 w 的最早开工时间; $c^E(w)$ 为工序 w 的最早完工时间; $s^L(w)$ 为工序 w 的最晚开工时间; $c^L(w)$ 为工序 w 的最晚完工时间。

属于工序 w 的机器空闲时间为: 从 $JP[w]$ 的最早完工时间 $c^E(JP[w])$, 到 $JS[w]$ 的最晚开工时间 $s^L(JS[w])$, 时间段区间 $[c^E(JP[w]), s^L(JS[w])]$ 内, 工序 w 可选加工机器的空闲时间。在时间段区间 $[c^E(JP[w]), s^L(JS[w])]$ 内, 能够最大限度地查找出属于工序 w 的机器空闲时间。

属于工序 w 的机器空闲时间可以分为两类:

① 除工序 w 当前加工机器之外的其他可选机器的空闲时间; ② 工序 w 当前加工机器的空闲时间。

跨机器移动工序是对第①类机器空闲时间的利用, 同机器移动工序是对第②类机器空闲时间的利用。

对工序块的关键工序移动操作如下: 工序 x 和 y 是两个相邻工序, x 和 y 所在机器是工序 w 的可选加工机器, 但不是工序 w 的当前加工机器, 工序 x 和 y 所在机器对工序 w 的加工时间为 $p_w(p_w=2)$, 关键工序 w 跨机器移动到工序 x 和 y 之间后, 是否可能减小最大完工时间, 与工序 x 和 y 之间可用空闲时间的大小有关。只有当工序 x 和 y 之间的空闲时间段与属于工序 w 的机器空闲时间查找区间 $[c^E(JP[w]), s^L(JS[w])]$ 的交集大于 p_w 时, 移动工序 w 到工序 x 和 y 之间, 才有可能减小最大完工时间^[15]。

由图 1 可知

$$[c^E(x), s^L(y)] \cap [c^E(JP[w]), s^L(JS[w])] = 3 > p_w = 2$$

可以将关键工序 w 移动到工序 x 和 y 之间, 即关键工序 w 的跨机器移动条件为

$$[c^E(x), s^L(y)] \cap [c^E(JP[w]), s^L(JS[w])] > p_w \quad (2)$$

图 1 跨机器移动工序邻域结构示意图

当工序 x 和 y 之间的空闲时间段与属于工序 w 的机器空闲时间查找区间 $[c^E(JP[w]), s^L(JS[w])]$ 的交集不大于 p_w 时, 即

$$[c^E(x), s^L(y)] \cap [c^E(JP[w]), s^L(JS[w])] \leq p_w$$

关键工序 w 移动到工序 x 和 y 之间后, 只能产生新的关键路径, 不能减小最大完工时间, 因此, 不能移动工序 w 。如图 1 所示, 工序 a, b, e, f 所在机器对工序 w 的加工时间 $p_w=2$, 对于工序 a 和 b

$$[c^E(a), s^L(b)] \cap [c^E(JP[w]), s^L(JS[w])] = 2 = p_w = 2$$

对于工序 e 和 f

$$[c^E(e), s^L(f)] \cap [c^E(JP[w]), s^L(JS[w])] = 1 < p_w = 2$$

将关键工序 w 移动到工序 a 和 b 之间, 或者 e 和 f 之间, 都不能减小最大完工时间, 因此, 不能移动

工序 w 。

此外,之所以将跨机器移动工序作为第一级邻域搜索,首先,对于 FJSP 问题,包含机器选择和工序排序两个子问题,其中,机器选择的质量是实现工序排序优化的前提和基础,相关的分析研究见文献[17],而跨机器移动工序包含了机器选择问题,同机器移动工序仅仅是排序优化问题;其次,从公式 2 可以看出,跨机器移动工序是对比较大的机器空闲时间($>p_w$)的利用,而接下来的同机器移动工序则涉及了更小的机器空闲时间的利用,先利用大的机器空闲时间,再利用小的机器空闲时间,符合调度优化的一般思想。

2.1.2 第二级邻域搜索: 同机器移动工序

在跨机器移动工序,进行第一级邻域搜索之后,为进一步提升算法搜索性能,再同机器移动工序,对工序排序进行优化,进行第二级邻域搜索。FJSP 问题同机器移动工序邻域结构示意图如图 2 所示,对工序块的关键工序邻域移动操作与 JSP 问题^[13]相同: 工序 x 和 y 是两个相邻工序, x 和 y 所在机器是工序 w 的当前加工机器。与跨机器移动工序机器空闲时间的判断条件不同,关键工序 w 移动到工序 x 和 y 之间,是否可能减小最大完工时间,只要当工序 x 和 y 之间的空闲时间段与属于工序 w 的机器空闲时间区间 $[c^E(JP[w]), s^L(JS[w])]$ 的交集不为 0 时,也即工序 w 的可用机器空闲时间不为 0,移动工序 w 到工序 x 和 y 之间,就有可能减小最大完工时间。关键工序 w 的同机器移动条件为

$$[c^E(x), s^L(y)] \cap [c^E(JP[w]), s^L(JS[w])] \neq 0 \quad (3)$$

图 2 同机器移动工序邻域结构示意图

由于同一工序块各关键工序之间是紧密连接的,不存在空闲时间,所以,当关键工序 w 为块内工序时,包括前移和后移,分别从块首工序开始向前查找和从块尾工序开始向后查找空闲时间进行移动。

当工序 w 为块首工序时, $JP[w]$ 若存在,则也必为关键工序,对于同一台机器上,工序块前面的两个相邻工序 x 和 y , 由于 $c^E(JP[w])=s^L(w)$ 和 $s^L(y)<s^L(w)$,

所以 $s^L(y)<c^E(JP[w])$, 工序 x 和 y 之间的空闲时间满足 $[c^E(x), s^L(y)] \cap [c^E(JP[w]), s^L(JS[w])]=0$, 不存在工序 w 的可用空闲时间,所以只存在从块尾工序开始向后查找空闲时间的后移操作。

当工序 w 为块尾工序时, $JS[w]$ 若存在,则也必为关键工序,对于同一台机器上,工序块后面的两个相邻工序 e 和 f , 由于 $s^L(JS[w])=c^E(w)$ 和 $c^E(w)<c^E(e)$, 所以 $s^L(JS[w])<c^E(e)$, 工序 e 和 f 之间的空闲时间满足 $[c^E(e), s^L(f)] \cap [c^E(JP[w]), s^L(JS[w])]=0$, 不存在工序 w 的可用空闲时间,所以只存在从块首工序开始向前查找空闲时间的前移操作。

2.1.3 可行解保障技术

对关键工序进行移动时,面临的另一个问题是: 移动关键工序后导致死锁,产生不可行解,因此,可行解保障技术也是邻域结构设计的重要研究内容。FJSP 问题是从传统 JSP 问题引伸发展而来,相对 FJSP 问题, JSP 问题的基础理论研究更加成熟,因此,可以基于已有传统 JSP 问题的工序移动条件,进而推导出 FJSP 问题相关的保证可行解工序移动条件。文献[18-19]针对传统 JSP 问题,分别提出了两个关键工序进行位置相对移动时,保证可行解的关键工序移动条件,其中,文献[18]定理的适用范围扩展到了同一台机器上任意两个关键工序。在文献[13]中,将定理的适用范围扩展到了同一台机器上的任意两个工序,取消了关键工序条件限制,本文对文献[13]中定理的适用范围进一步扩展应用到了 FJSP 问题。

(1) 同机器移动工序保证可行解条件。在传统 JSP 问题中,工件的各道工序分别在不同的机器上加工,在工件的加工过程中,最多只经过一台机器一次。与传统 JSP 问题的不同之处在于,在 FJSP 中,同一工件的多道工序可以在同一台机器上加工,在工件的加工过程中,可能多次经过同一台机器。因此,对于一个 FJSP 问题的可行解,在同一台机器上对工序进行前移和后移时,基于文献[13]中的定理 1 和定理 2,可以得出 FJSP 保证可行解的工序移动条件。

推论 1 对于一个 FJSP 问题可行解,同一台机器上的任意两个工序为 u 和 v , 假设 u 在 v 前加工,如果满足下列条件之一:① 工序 u 为工件的末道工序, $JS[u]$ 不存在;② $JS[u]$ 存在, $L(v, \#) \geq L(JS[u], \#)$, 且 $v \neq JS[u]$; 那么移动工序 u 至工序 v 之后仍然是可行解。

推论 2 对于一个 FJSP 问题可行解, 同一台机器上的任意两个工序为 u 和 v , 假设 u 在 v 前加工, 如果满足下列条件之一: ① 工序 v 为工件的首道工序, $JP[v]$ 不存在; ② $JP[v]$ 存在, $L(0, u) + p_u \geq L(0, JP[v]) + p_{JP[v]}$, 且 $u \neq JP[v]$; 那么移动工序 v 至 u 之前仍然是可行解。

(2) 跨机器移动工序保证可行解条件。对于一个 FJSP 问题可行解, 推论 1 和推论 2 给出了在同一台机器上对工序进行后移和前移时, 保证可行解的工序移动条件。由于 FJSP 工序可能可以在多台机器上加工, 因此, 基于空闲时间对工序进行邻域搜索时, 工序不仅可以在同一台机器上进行移动, 而且可以移动到其他可选机器上, 此时同样需要保证解的可行性。

同机器移动工序包括对工序的前移和后移操作, 跨机器移动工序是将工序移动到其他可选机器的相邻两道工序之间, 三种工序移动方式如图 3 所示。上述推论 1 给出的是同一台机器上, 工序往后移动, 移至某工序之后, 保证可行解的工序移动条件; 推论 2 给出的是同一台机器上, 工序往前移动, 移至某工序之前, 保证可行解的工序移动条件。对于 FJSP 问题, 当某工序 w 移动到其他机器, 将工序 w 插入到该机器上相邻的两道工序 x 和 y 之间时, 对于前面的工序 x 而言, 工序 w 移动到了工序 x 之后, 对于后面的工序 y 而言, 工序 w 移动到了工序 y 之前, 因此, 综合推论 1 和推论 2, 可以推导出 FJSP 问题工序跨机器移动时保证可行解的工序移动条件。

图3 三种工序移动方式

推论 3 对于一个可行解, 同一台机器上的任意两个相邻工序为 x 和 y , 假设 x 在 y 前加工, 工序 x 和 y 所在机器是其他机器上工序 w 的可选加工机器, 如果同时满足下列两个条件: ① 工序 w 为

工件的末道工序, $JS[w]$ 不存在, 或者 $JS[w]$ 存在, $L(x, \#) \geq L(JS[w], \#)$, 且 $x \neq JS[w]$; ② 工序 w 为工件的首道工序, $JP[w]$ 不存在, 或者 $JP[w]$ 存在, $L(0, y) + p_y \geq L(0, JP[w]) + p_{JP[w]}$, 且 $y \neq JP[w]$; 那么移动工序 w 至工序 x 和 y 之间仍然是可行解。

对于推论 3, 当工序 w 移动到其他机器的首道工序之前或者末道工序之后加工时, 即工序 x 或工序 y 不存在时, 分别只需要满足推论 3 的条件②或①即可。上述 3 个推论的证明过程可以参考文献[13]中给出的详细证明过程, 基本思想都是证明移动工序后, 新产生的有向弧不可能出现在有向回路中。

2.2 遗传算法实现

2.2.1 编码与解码

FJSP 问题包括机器选择和工序排序两个子问题, 首先要为每道工序选择一台加工机器, 然后安排每台机器上的工序加工顺序, 从而确定每道工序的开工时间和完工时间。遗传算法染色体采用双链结构编码方式, 以表 1 所示 FJSP 问题为例, 其编码结构如图 4 所示。机器选择链表示各工序从可选机器集中选择的机器, 每个基因为机器号, 工序顺序链采用 JSP 问题中基于工序的编码方式, 每个基因为工件号, 根据位置顺序解释是第几道工序。

图4 遗传算法染色体编码方式

采用主动调度的解码方式, 根据机器选择链所确定的每道工序的加工机器及相应的加工时间, 按照工序顺序链所确定的工序加工顺序依次进行解码, 在不推迟其他已安排工序开工时间的条件下, 将工序插入到对应机器上最早可行的时刻加工。

2.2.2 选择操作

选择操作可以使优良的个体得以更大的概率生存, 本文算法采用锦标赛选择方法, 在父代种群中随机选取 2 个个体, 令其中适应度值好的个体进入下一代, 同时, 辅以精英保留策略。

2.2.3 交叉操作

交叉操作可以使群体中优良个体的基因特性在一定程度上得以保持。机器选择链采用基于工件的交叉和两点交叉混合的方式, 工序顺序链采用线性

次序交叉^[20]和优先操作交叉^[21]混合的方式。基于工件的交叉如图 5 所示, 随机选择 2 个工件, 将父代个体两个工件的机器号互换。两点交叉是随机生成两个随机数, 交换两随机数位置之间的基因片段。

图 5 机器选择链基于工件的交叉示意图

线性次序交叉如图 6 所示, 随机生成两个随机位置, 交换两父代个体随机位置之间的基因片段, 在原父代个体中删除从另一父代个体交换过来的基因, 然后从第一个基因位置起依次在两交叉位置外填入剩余基因。优先操作交叉随机划分工件集为两个非空集合, 其中一个集合中工件的位置在两父代个体中保持不变, 交换两父代个体另一集合中工件的顺序。

图 6 工序顺序链线性次序交叉示意图

2.2.4 变异操作

变异操作可以维持遗传算法种群的多样性, 防止出现早熟收敛现象。机器选择链采用单点变异策略, 在机器选择链中, 随机选择一个位置, 在该位置工序所对应的可选机器集中选择一个与当前机器号不同的机器, 替换当前机器。工序顺序链采用互换变异方式, 在工序顺序链中, 随机选择两个不同位置, 将其基因值互换。

2.3 混合算法设计

单一算法存在全局搜索能力或局部搜索能力差的问题, 将不同特点的算法混合使用, 可以避免单一算法的不足和缺陷。本文综合考虑 FJSP 问题求解算法的全局搜索能力和局部搜索能力, 利用遗传算法实现全局搜索, 基于邻域结构的两级邻域搜索实现局部搜索, 设计求解 FJSP 问题的混合智能优化算法, 混合算法框架如图 7 所示。对选择、交叉、变异三种遗传操作产生的群体中的个体解码之后,

再利用本文所提出的基于邻域结构的两级邻域搜索进一步改进个体。邻域搜索时, 对关键路径上的所有工序, 按照随机次序逐个选取。对每个工序, 首先跨机器移动进行第一级邻域搜索, 然后同机器移动进行第二级邻域搜索。

图 7 求解 FJSP 问题的混合算法框架图

3 算法测试

为了检验基于两级邻域搜索混合算法求解 FJSP 问题的性能, 利用 Visual Basic 语言编程, 对 FJSP 问题基准算例进行测试, 计算机中央处理器的主频为 2.60 GHz, 内存为 2.0 GB。算法参数设置如下: 每个算例种群规模 $PopSize$ 分别如求解结果统计表 2 所示, 交叉概率为 0.8, 变异概率为 0.1, 最大代数 100 代。每个算例求解 20 次, 每次求解过程中, 当最优值连续 10 代不更新时, 结束运行。

首先, 采用文献[22]中的 3 个 FJSP 算例和 BRANDIMARTE^[23]所设计的 FJSP 算例进行测试。测试试验结果如表 2 所示, 将 C_{max} 与每个问题实例的最优解下界 LB 从相对误差 Re 的角度进行对比分析。将测试结果与相关文献中的结果进行对比, 包括文献[1]中著名的 M&G 算法、文献[5]中的 hGA 算法、文献[24]中的 eGA 算法、文献[2]中的 BEDA 算法、文献[25]中的 KBVNS 算法。本文算法对 8×8 、 10×10 、 15×10 算例测试结果的相对误差都为 0, 由于部分文献算法没有对该 3 个算例进行测试, 所以只对 10 个 mk 算例的相对误差求平均值 $Av(Re)$, 本文算法的 $Av(Re)$ 为 15.1, 整体性能优于 M&G 算法、

表 2 mk FJSP 算例求解结果统计

问题	$n \times m$	LB	M&G		hGA		eGA		BEDA		KBVNS		本文算法				
			文献[1]		文献[5]		文献[24]		文献[2]		文献[25]		$PopSize$	$C_{\max}$	Re	$Av(C_{\max})$	$Av(t)$
			$C_{\max}$	Re	$C_{\max}$	Re	$C_{\max}$	Re	$C_{\max}$	Re	$C_{\max}$	Re					
mk01	10×6	36	40	11.11	40	11.11	40	11.11	40	11.11	40	11.11	50	40	11.11	40	2
mk02	10×6	24	26	8.33	26	8.33	26	8.33	26	8.33	26	8.33	50	26	8.33	26.55	8.1
mk03	15×8	204	204	0	204	0	204	0	204	0	204	0	50	204	0	204	0.7
mk04	15×8	48	60	25	60	25	60	25	60	25	60	25	50	60	25	60.5	10.6
mk05	15×4	168	173	2.98	172	2.38	173	2.98	172	2.38	173	2.98	100	172	2.38	172.9	35.7
mk06	10×15	33	58	75.76	58	75.76	58	75.76	60	81.82	58	75.76	100	58	75.76	59.9	141.8
mk07	20×5	133	144	8.27	139	4.51	144	8.27	139	4.51	139	4.51	100	139	4.51	140.2	58.4
mk08	20×10	523	523	0	523	0	523	0	523	0	523	0	50	523	0	523	0.4
mk09	20×10	299	307	2.68	307	2.68	307	2.68	307	2.68	307	2.68	50	307	2.68	307	17.6
mk10	20×15	165	198	20	197	19.39	198	20	206	24.85	199	20.61	50	200	21.21	204.6	123.8
8×8	8×8	14	—	—	14	—	—	—	14	—	14	—	50	14	—	14	0.4
10×10	10×10	7	—	—	7	—	—	—	7	—	7	—	50	7	—	7	0.4
15×10	15×10	11	—	—	11	—	—	—	11	—	11	—	50	11	—	11	2.9
$Av(Re)$ (mk01~mk10)				15.41	14.92		15.41		16.07		15.1		15.1				

eGA 算法和 BEDA 算法, 与 hGA 算法、KBVNS 算法相当。在个别算例的求解上, 与其余 5 种算法相比, 除 mk10 算例外, 本文算法的求解结果均取得了所列算法的最好解, 验证了本文算法的有效性, 图 8 为本文算法求得的 mk10 算例的调度甘特图。

然后, 采用由 Barnes 和 Chambers 设计的 21 个 Bcdata FJSP 算例进行测试, 测试试验结果如表 3 所示。将测试结果与文献[1]中的 M&G 算法、文献[5]中的 hGA 算法、文献[24]中的 eGA 算法、文献[12]中的 CDDs 算法求解结果进行对比。本文算法的 $Av(Re)$ 为 22.5, 整体性能优于其余 4 种算法。在个别算例的求解上, 与其余 4 种算法相比, 21 个算例中有 17 个算例, 本文算法的求解结果均取得了

所列算法的最好解。将本文算法求解 Bcdata FJSP 基准算例的优于解和差于解数目与文献中 4 种算法的求解结果进行统计分析, N_1 为本文算法的优于解数目, N_2 为差于解数目, 统计结果如表 4 所示。与文献[1]中的 M&G 算法相比, 优于解数目 N_1 和差于解数目 N_2 都为 4, 与文献[12]中的 CDDs 算法相比, 优于解数目 N_1 和差于解数目 N_2 分别为 5 和 4, 基本相当。与其余 2 种算法相比, 优于解数目明显比差于解数目多, 对于算例 setb4xy, 本文算法求得的 $C_{\max}=912$, 其余 4 种算法均为 $C_{\max}=916$, 求解结果优于其余 4 种算法, 验证了本文算法的有效性, 图 9 为本文算法求得的 setb4xy 算例的调度甘特图。

图 8 mk10 算例的调度甘特图($C_{\max}=200$)

表 3 Bcdata FJSP 算例求解结果统计

问题	$n \times m$	LB	M&G 文献[1]		hGA 文献[5]		eGA 文献[24]		CDDS 文献[12]		本文算法				
			$C_{\max}$	Re	$C_{\max}$	Re	$C_{\max}$	Re	$C_{\max}$	Re	PopSize	$C_{\max}$	Re	$Av(C_{\max})$	$Av(t)$
mt10c1	10×11	655	928	41.68	927	41.53	927	41.53	928	41.68	300	927	41.53	929.05	47
mt10cc	10×12	655	910	38.93	910	38.93	910	38.93	910	38.93	200	910	38.93	916.65	67.8
mt10x	10×11	655	918	40.15	918	40.15	918	40.15	918	40.15	200	918	40.15	924.75	44.3
mt10xx	10×12	655	918	40.15	918	40.15	918	40.15	918	40.15	200	918	40.15	925.15	41.9
mt10xxx	10×13	655	918	40.15	918	40.15	918	40.15	918	40.15	200	918	40.15	923.5	31.6
mt10xy	10×12	655	906	38.32	905	38.17	905	38.17	906	38.32	200	905	38.17	907.45	41.9
mt10xyz	10×13	655	847	29.31	849	29.62	847	29.31	849	29.62	200	847	29.31	860.7	90.3
setb4c9	15×11	857	919	7.23	914	6.65	914	6.65	919	7.23	200	914	6.65	923.75	32.7
setb4cc	15×12	857	909	6.07	914	6.65	909	6.07	909	6.07	200	909	6.07	916.8	70.6
setb4x	15×11	846	925	9.34	925	9.34	925	9.34	925	9.34	200	925	9.34	936.2	88.2
setb4xx	15×12	846	925	9.34	925	9.34	925	9.34	925	9.34	200	925	9.34	934.2	56.9
setb4xxx	15×13	846	925	9.34	925	9.34	925	9.34	925	9.34	200	925	9.34	936.05	37.3
setb4xy	15×12	845	916	8.4	916	8.4	916	8.4	916	8.4	300	912	7.93	924.15	47.7
setb4xyz	15×13	838	905	8	905	8	905	8	905	8	200	905	8	919.55	58.1
seti5c12	15×16	1027	1174	14.31	1175	14.41	1174	14.31	1174	14.31	200	1174	14.31	1185.25	101.4
seti5cc	15×17	955	1136	18.95	1138	19.16	1136	18.95	1136	18.95	200	1136	18.95	1151.45	77.5
seti5x	15×16	955	1201	25.76	1204	26.07	1209	26.6	1201	25.76	200	1204	26.07	1226.45	172.1
seti5xx	15×17	955	1199	25.55	1202	25.86	1204	26.07	1199	25.55	200	1200	25.65	1221.4	151.4
seti5xxx	15×18	955	1197	25.34	1204	26.07	1204	26.07	1197	25.34	300	1199	25.55	1211.45	190.5
seti5xy	15×17	955	1136	18.95	1136	18.95	1136	18.95	1136	18.95	200	1136	18.95	1152.6	138
seti5xyz	15×18	955	1125	17.8	1126	17.91	1125	17.8	1125	17.8	100	1127	18.01	1140.25	97.6
$Av(Re)$			22.53		22.61		22.58		22.54		22.5

图 9 setb4xy 算例的调度甘特图($C_{\max}=912$)

表 4 Bcdata FJSP 算例优于解和差于解数目对比分析

文献	M&G 文献[1]	hGA 文献[5]	eGA 文献[24]	CDDS 文献[12]
优于解数目	4	7	4	5
差于解数目	4	1	1	4

4 结论

(1) 针对最小化最大完工时间的 FJSP 问题, 提

出了一种基于邻域结构的两级邻域搜索方法, 并与遗传算法融合, 构造了一种混合求解算法, 对基准算例的测试结果验证了所提方法的有效性。

(2) 验证了基于通过利用机器空闲时间来减小最大完工时间方法的有效性, 可以避免算法搜索的盲目性。

(3) 保证可行解的工序移动条件为进一步研究 FJSP 问题新的邻域结构和工序邻域移动搜索方式,

提供了理论基础。

(4) 基于邻域结构的两级邻域搜索方法可以作为求解 FJSP 问题的一项共性关键技术, 通过与其他不断涌现的不同机理的智能算法进行融合, 算法的求解性能还存在进一步的提升空间。

参 考 文 献

- [1] MASTROLILLI M, GAMBARDELLA L M. Effective neighborhood functions for the flexible job shop problem[J]. *Journal of Scheduling*, 2000, 3(1): 3-20.
- [2] WANG L, WANG S Y, XU Y, et al. A bi-population based estimation of distribution algorithm for the flexible job-shop scheduling problem[J]. *Computers & Industrial Engineering*, 2012, 62(4): 917-926.
- [3] WANG L, ZHOU G, XU Y, et al. An effective artificial bee colony algorithm for the flexible job-shop scheduling problem[J]. *International Journal of Advanced Manufacturing Technology*, 2012, 60(1-4): 303-315.
- [4] YUAN Y, XU H, YANG J D. A hybrid harmony search algorithm for the flexible job shop scheduling problem[J]. *Applied Soft Computing*, 2013, 13(7): 3259-3272.
- [5] GAO J, SUN L Y, GEN M. A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems[J]. *Computers & Operations Research*, 2008, 35(9): 2892-2907.
- [6] YUAN Y, XU H. Flexible job shop scheduling using hybrid differential evolution algorithms[J]. *Computers & Industrial Engineering*, 2013, 65(2): 246-260.
- [7] 张国辉. 柔性作业车间调度方法研究[D]. 武汉: 华中科技大学, 2009.
ZHANG Guohui. Research on methods for flexible job shop scheduling problems[D]. Wuhan: Huazhong University of Science and Technology, 2009.
- [8] GAO J, GEN M, SUN L Y, et al. A hybrid of genetic algorithm and bottleneck shifting for multiobjective flexible job shop scheduling problems[J]. *Computers & Industrial Engineering*, 2007, 53(1): 149-162.
- [9] NOWICKI E, SMUTNICKI C. A fast taboo search algorithm for the job shop problem[J]. *Management Science*, 1996, 42(6): 797-813.
- [10] BOŻEJKO W, UCHROŃSKI M, WODECKI M. Parallel hybrid metaheuristics for the flexible job shop problem[J]. *Computers & Industrial Engineering*, 2010, 59(2): 323-333.
- [11] LI J Q, PAN Q K, SUGANTHAN P N, et al. A hybrid tabu search algorithm with an efficient neighborhood structure for the flexible job shop scheduling problem[J]. *International Journal of Advanced Manufacturing Technology*, 2011, 52(5-8): 683-697.
- [12] HMIDA A B, HAOUARI M, HUGUET M, et al. Discrepancy search for the flexible job shop scheduling problem[J]. *Computers & Operations Research*, 2010, 37(12): 2192-2201.
- [13] 赵诗奎, 方水良, 顾新建. 作业车间调度的空闲时间邻域搜索遗传算法[J]. *计算机集成制造系统*, 2014, 20(8): 1930-1940.
ZHAO Shikui, FANG Shuiliang, GU Xinjian. Idle time neighborhood search genetic algorithm for job shop scheduling[J]. *Computer Integrated Manufacturing Systems*, 2014, 20(8): 1930-1940.
- [14] 张超勇, 管在林, 刘琼, 等. 一种新调度类型及其在作业车间调度中的应用[J]. *机械工程学报*, 2008, 44(10): 24-31.
ZHANG Chaoyong, GUAN Zailin, LIU Qiong, et al. New scheduling type applied to solving job-shop scheduling problem[J]. *Chinese Journal of Mechanical Engineering*, 2008, 44(10): 24-31.
- [15] BALAS E. Machine sequencing via disjunctive graphs: An implicit enumeration algorithm[J]. *Operations Research*, 1969, 17(6): 941-957.
- [16] 王磊, 黄文奇. 求解工件车间调度问题的一种新的邻域搜索算法[J]. *计算机学报*, 2005, 28(5): 809-816.
WANG Lei, HUANG Wenqi. A new local search algorithm for job shop scheduling problem[J]. *Chinese Journal of Computers*, 2005, 28(5): 809-816.
- [17] 赵诗奎, 方水良, 顾新建. 基于极限调度完工时间最小化的机器选择及 FJSP 求解[J]. *计算机集成制造系统*, 2014, 20(4): 854-865.
ZHAO Shikui, FANG Shuiliang, GU Xinjian. Machine selection and FJSP solution based on limit scheduling completion time minimization[J]. *Computer Integrated Manufacturing Systems*, 2014, 20(4): 854-865.
- [18] ZHANG C Y, LI P G, GUAN Z L, et al. A tabu search algorithm with a new neighborhood structure for the job shop scheduling problem[J]. *Computers & Operations Research*, 2007, 34(11): 3229-3242.
- [19] BALAS E, VAZACOPOULOS A. Guided local search

- with shifting bottleneck for job shop scheduling[J]. Management Science, 1998, 44(2): 262-275.
- [20] 王凌. 车间调度及其遗传算法[M]. 北京: 清华大学出版社, 2003.
- WANG Ling. Job shop scheduling and its genetic algorithm[M]. Beijing: Tsinghua University Press, 2003.
- [21] 张超勇, 饶运清, 刘向军, 等. 基于 POX 交叉的遗传算法求解 Job-Shop 调度问题[J]. 中国机械工程, 2004, 15(23): 2149-2153.
- ZHANG Chaoyong, RAO Yunqing, LIU Xiangjun, et al. An improved genetic algorithm for the job shop scheduling problem[J]. China Mechanical Engineering, 2004, 15(23): 2149-2153.
- [22] XIA W J, WU Z M. An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems[J]. Computers & Industrial Engineering, 2005, 48: 409-425.
- [23] BRANDIMARTE P. Routing and scheduling in a flexible job shop by tabu search[J]. Annals of Operations Research, 1993, 41: 157-183.
- [24] ZHANG G H, GAO L, SHI Y. An effective genetic algorithm for the flexible job-shop scheduling problem[J]. Expert Systems with Applications, 2011, 38(4): 3563-3573.
- [25] KARIMI H, RAHMATI S H A, ZANDIEH M. An efficient knowledge-based algorithm for the flexible job shop scheduling problem[J]. Knowledge-Based Systems, 2012, 36: 236-244.
-
- 作者简介: 赵诗奎(通信作者), 男, 1984年出生, 博士, 讲师。主要研究方向为车间生产调度、智能优化算法、云制造等。
- E-mail: zhaoshikui@126.com