

基于邻接矩阵的作业车间调度可行解判定方法

于丰顺¹, 赵诗奎^{1,2†}, 仵政源¹, 李彤¹

(1. 济南大学 机械工程学院, 济南 250022;

2. 济南大学 山东省金属关键构件表面处理与智能装备重点实验室, 济南 250022)

摘要: 针对作业车间调度问题中邻域结构的可行解判定问题, 提出一种基于邻接矩阵的可行解判定方法. 首先, 从析取图角度分析工序间的路径关系情况, 指出现有可行解判定方法的局限性, 进而设计基于邻接矩阵的可行解判定方法. 该方法不但能保证邻域移动可行性的精准判定, 而且能够避免可行解的遗漏, 进一步扩大整体的有效搜索空间. 此外, 为了提高邻接矩阵相关的计算效率, 提出一种基于拓扑排序片段的邻接矩阵双向缩减方法, 提高快速判定效率. 最后, 对该方法在邻域数目上与其他的可行解判定方法进行比较, 并融入混合算法对不同规模的基准算例进行测试求解, 从而验证该方法的有效性、基础意义和应用价值.

关键词: 作业车间调度; 邻域结构; 可行解; 邻接矩阵; 拓扑排序; 双向缩减

中图分类号: TP301 文献标志码: A

DOI: 10.13195/j.kzyjc.2025.0408

引用格式: 于丰顺, 赵诗奎, 仵政源, 等. 基于邻接矩阵的作业车间调度可行解判定方法 [J]. 控制与决策, 2025, 40(10): 3085-3095.

A feasible solution determination method for job shop scheduling based on adjacency matrix

YU Feng-shun¹, ZHAO Shi-kui^{1,2†}, WU Zheng-yuan¹, LI Tong¹

(1. School of Mechanical Engineering, University of Jinan, Jinan 250022, China; 2. Shandong Key Laboratory of Surface Engineering and Intelligent Equipment for Key Metal Components, University of Jinan, Jinan 250022, China)

Abstract: A feasible solution determination method based on an adjacency matrix is proposed for neighborhood structures in job shop scheduling problems. First, the path relationships among operations are analyzed from the perspective of disjunctive graphs, and the limitations of existing determination methods are identified. Based on this analysis, an adjacency-matrix-based method is designed to ensure the accurate determination of neighborhood move feasibility while avoiding the omission of feasible solutions, which substantially broadens the valid search space. In addition, to improve the computational efficiency of the adjacency matrix, a dual-directional pruning method for the adjacency matrix is proposed based on segments of topological sorting, accelerating the feasibility checking process. Finally, the proposed method is compared with other determination methods in terms of the number of feasible neighborhoods, and benchmark instances of various scales are solved using a hybrid algorithm to validate its effectiveness. The results confirm the foundational significance and practical value of the proposed method.

Keywords: job shop scheduling; neighborhood structure; feasible solution; adjacency matrix; topological sorting; dual-directional pruning

0 引言

作业车间调度问题 (JSP) 是一种典型的 NP 难组合优化问题, 也可看作一种排序问题, 其实质是对不同的工作任务安排一个可执行的顺序和时间, 使预定的目标最优化^[1]. JSP 不仅在实际生产中应用广泛, 也是许多复杂车间调度问题 (如柔性作业调度、

流水作业调度) 重要的研究基础.

基于一定邻域结构的局部搜索算法是一种求解 JSP 快速有效的方法, 局部搜索算法的特点是从一个解出发, 根据一定的规则自我进化, 从而探索最优解. 邻域结构是对一个给定解产生小的扰动获得邻域解的机制, 它是局部搜索的核心, 极大影响算法的搜索

收稿日期: 2025-04-18; 录用日期: 2025-06-25.

基金项目: 国家自然科学基金项目 (52275490).

责任编委: 李新宇.

†通信作者. E-mail: me_zhaosk@ujn.edu.cn.

效率和解的质量. JSP 中较为常用的是基于关键路径的工序块邻域结构, Dell'Amico 等^[2]提出了块的概念以及 N4 邻域结构, 即在保证邻域解可行的基础上, 将块内的工序尽可能地向块首或块尾移动; 基于 N4 邻域结构, Nowicki 等^[3]提出了 N5 邻域结构, 将关键块的块首与块尾工序交换; Balas 等^[4]提出的 N6 邻域结构是将关键块的内部工序移动到块首和块尾; Zhang 等^[5]在此基础上提出的 N7 邻域结构考虑了将关键块的块首和块尾工序移动到块内的移动方式; 潘全科等^[6]在差分进化算法中提出了一种基于块结构的插入和交换邻域结构; 赵诗奎等^[7]通过分析加工机器上关键工序前后的空闲时间, 提出了一种新型邻域结构; Xie 等^[8]进一步扩展了 N7 邻域结构, 将关键块上的工序移动到关键块外, 能搜索更广阔的解空间; 薛玲玲^[9]提出了一种基于块结构邻域搜索的遗传算法; 巴志勇等^[10]分析了 N7 邻域结构关键块中工序无效移动的情况, 设计了一种 3 对工序精确联动的邻域结构; Gui 等^[11]分析了工序移动是否为可行解的充要条件, 从工序路径关系上提出新的邻域结构.

上述邻域结构都是求解 JSP 较为有效的方法, 但对于部分邻域移动, 它们生成的邻域解不一定是可行解, 不可行解会增大算法的搜索范围, 导致计算时间的大幅度增加, 邻域解的可行性判定影响着算法整体求解质量和效率. 针对这一现象, 学者们各自提出了可行解的快速判断方法. Dell'Amico 等^[2]从工序最大完工时间角度提出了可行解判断条件, 从而保证 N4 邻域移动的可行性; Balas 等^[4]从工序头尾长的角度提出了一种判断可行解的约束条件, 后续一些学者在此基础上进行了一些拓展, 如文献 [5, 7-8] 等; Sun 等^[12]提出了一种基于图论的启发式判别算法. 但这些约束条件大多为充分不必要条件, 无法将所有的可行解包括在内, 容易遗漏一部分有潜在提升可能的解^[11]. 除了基本的局部搜索算法外, 将其他智能优化算法应用于 JSP 已成为当前研究领域的一个热点, 如近年来新兴的近端策略优化算法^[13]、多目标进化算法^[14]等其他优秀算法, 它们为解决 JSP 及其衍生问题提供了重要方向.

为对邻域解的可行性进行高效且精准地判定, 本文对邻域搜索过程中判断可行解的条件进行研究, 从拓扑排序和邻接矩阵的角度出发, 结合现有的可行解判定方法和 JSP 领域的相关知识, 提出一种判定工序移动可行解的方法. 该方法不仅能对工序移动的可行性进行有效判定, 而且判定范围包含全部的邻域解, 可避免部分潜力解的遗漏. 同时, 针对该

方法计算邻接矩阵时耗时较多的问题, 设计一种基于拓扑排序片段的双向缩减方法. 最后, 通过多个实验测试验证本文所提出算法的有效性, 并与其他算法在基准算例上进行对比分析.

1 问题模型

1.1 问题描述

JSP 可以描述为: n 个工件 $J = \{J_1, J_2, \dots, J_n\}$ 在 m 台机器 $M = \{M_1, M_2, \dots, M_m\}$ 上加工, 每个工件 J_i 由 m 个连续工序 $\{O_{i,1}, O_{i,2}, \dots, O_{i,m}\}$ 组成, 相应的加工机器集合为 $\{\sigma_{i,1}, \sigma_{i,2}, \dots, \sigma_{i,m}\}$, 工序 $O_{i,j}$ 在机器 $\sigma_{i,j}$ 上的加工时间为 $p_{i,j}$, 工件加工过程需要满足一定的约束条件, 包括如下 4 项:

- 1) 同一工件的工序之间存在先后顺序约束;
- 2) 一个工件同一时刻最多只能在一台机器上加工;
- 3) 一台机器同一时刻最多只能加工一个工件;
- 4) 工序在机器上一旦开始加工不能中断.

表 1 所示为一个 4×3 的 JSP 实例, 由 4 个工件与 3 个机器共 12 道工序组成, 后续的邻域移动分析及方法解释均基于该实例进行说明.

表1 4×3 的 JSP 实例

工件号	工序号	加工机器及加工时间		
		M_1	M_2	M_3
J_1	$O_{1,1}$			3
	$O_{1,2}$		3	
	$O_{1,3}$	4		
J_2	$O_{2,1}$		2	
	$O_{2,2}$			3
	$O_{2,3}$	3		
J_3	$O_{3,1}$	3		
	$O_{3,2}$		3	
	$O_{3,3}$			4
J_4	$O_{4,1}$		2	
	$O_{4,2}$	4		
	$O_{4,3}$			1

本文以最小化最大完工时间 C_{\max} 为目标, 研究如何安排每台机器上的工序加工顺序, 以减小最大完工时间.

目标函数表示为

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

其中: i 为工件 J_i 的工件号, n 为工件总数, C_i 为工件 J_i 的完工时间.

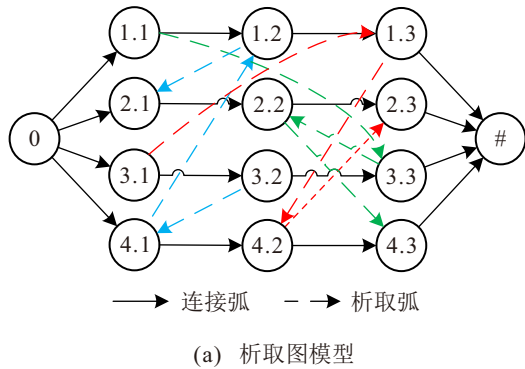
后续用到的相关符号说明如表 2 所示.

表2 符号表示及说明

符号	描述
MP[x]	工序x的机器前序工序
JP[x]	工序x的工件前序工序
MS[x]	工序x的机器后序工序
JS[x]	工序x的工件后序工序
DP[x]	工序x在拓扑排序中直接前序工序, DP[x] = JP[x]或MP[x]
DS[x]	工序x在拓扑排序中直接后序工序, DS[x] = JS[x]或MS[x]
P_x	工序x的加工时间
$l(x)$	工序x的最大弧数
F_G	析取图G的一个拓扑排序
$f_G(x)$	工序x在 F_G 中从左到右的位置
$L(0, x)$	工序x的头长, 即0时刻开始到工序x的最早开工时间长度
$L(x, \#)$	工序x的尾长, 即工序x的最晚开工时间开始到结束的时间长度

1.2 析取图模型

析取图模型是描述 JSP 并对其进行理论分析的有



力工具, JSP 的析取图模型定义为 $G = \langle N, A, E \rangle$. 其中: N 为所有工件的工序节点以及 2 个虚拟节点 (开始节点和结束节点, 分别用 0 和 # 表示) 组成的集合; A 为同一工件相邻两道工序间的连接弧集合, 对应于工件的工艺路线约束; E 为同一机器两道工序间的析取弧集合, 对应工件的机器路线约束. 图 G 中, 路径 $P = \{o_1, o_2, \dots, o_n\}$ 表示从节点 o_1 到 o_n 的一条有向通路, 由一系列相邻节点间的有向弧 $(o_i, o_{i+1}) (1 \leq i < n)$ 组成, $L(u, v)$ 表示 G 中从工序 u 的节点到工序 v 的节点最长路径的长度, 从开始节点到结束节点的最长路径称为关键路径, 对应调度的最大完工时间 $C_{max} = L(0, \#)$.

图 1(a) 为表 1 实例一个可行调度解对应的析取图模型, 图 1(b) 为该调度解对应的甘特图, 由黑色箭头连接的工序为关键工序, 所有的关键工序组成关键路径. 在同一机器上由两个及两个以上紧密相连的关键工序组成的工序块称为关键工序块.

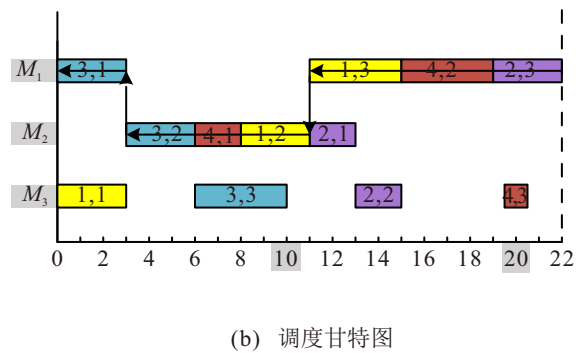


图1 调度解的析取图模型和甘特图

关于析取图的一个重要定理^[11]如下.

定理 1 在作业车间调度问题中, 产生不可行解的充要条件是解的析取图中存在闭环 (有向回路).

1.3 拓扑排序

根据定理 1, 当析取图 G 中不包含闭环时对应一个 JSP 可行解, 而拓扑排序是检查有向图中是否存在闭环的方法之一, 拓扑排序是一种用于有向无环图 (DAG) 的排序方法, 是动态构造 JSP 邻域解的一个重要工具. 对于 JSP 的一个可行调度解而言, 其拓扑排序是一个由所有工序构成的有序集合, 它反映了该调度解中所有工序的调度顺序^[15]. 在调度解的拓扑排序中, 当且仅当析取图 G 为有向无环图时存在拓扑排序. 在 JSP 中, 拓扑排序存在以下重要性质和定理^[15-17]:

性质 1 给定一个 DAG 图 G , 若存在从节点 u 到 v 的路径, 则 $f_G(u) < f_G(v)$.

性质 2 给定一个 DAG 图 G , 若节点 $u \neq v$, 则 $f_G(u) \neq f_G(v)$.

性质 3 一个 DAG 图有多个合法的拓扑排序, 拓扑排序并不唯一.

定理 2 给定一个 DAG 图 G 和其拓扑排序 F_G , 若 $f_G(u) < f_G(v)$, 则图 G 中不存在从 v 到 u 的路径.

由于一个调度解可能存在多个拓扑排序, 本文采用一种基于工序最大弧数^[16]的计算方法实现拓扑排序的统一构造. 路径的弧数定义为该路径上相邻节点间的有向弧的数量总和, 对于工序 $\forall x \in N$, 其最大弧数 $l(x)$ 表示从 0 到 x 的所有路径的弧数最大值^[15-16], 计算公式为

$$l(x) = \max \{l(\text{JP}[x]), l(\text{MP}[x])\} + 1. \quad (2)$$

表 3 为图 1 对应的调度解中各个工序的最大弧数.

表3 工序的最大弧数

工序	$O_{1.1}$	$O_{1.2}$	$O_{1.3}$	$O_{2.1}$	$O_{2.2}$	$O_{2.3}$	$O_{3.1}$	$O_{3.2}$	$O_{3.3}$	$O_{4.1}$	$O_{4.2}$	$O_{4.3}$
最大弧数	1	4	5	5	6	7	1	2	3	3	6	7

将析取图 G 中所有的节点,按最大弧数由小到大排列得到的节点序列(若两个节点的最大弧数相等,则表示两者不存在路径关系,可随机确定两者之间的顺序,此种情况并不影响后续的判定),是调度解的一个合法的拓扑排序^[15].按表3的最大弧数从小到大排序确定该调度解的一个拓扑排序为 $F_G = \{O_{3.1}, O_{1.1}, O_{3.2}, O_{4.1}, O_{3.3}, O_{1.2}, O_{1.3}, O_{2.1}, O_{4.2}, O_{2.2}, O_{2.3}, O_{4.3}\}$.

如果从析取图 G 的拓扑排序中任意截取一段排序片段,则由于截取片段中的各节点位置顺序没有发生变化,从原拓扑排序截取的子拓扑排序各节点依然包含原析取图的路径关系.建立调度解的拓扑排序,主要利用拓扑排序的性质和定理,将当前调度解析取图中所有工序的路径关系并按一定的顺序放进排序集合中,为下一步邻接矩阵的判断和缩减建立基础.

2 已有可行解条件研究

关于邻域解可行性的判定一直是邻域结构的重要组成部分,不论是哪种判定方法,其本质都是对工序满

足其加工顺序约束要求的具体体现. N1 邻域结构和 N5 邻域结构由于只交换相邻的关键工序,相关工序开始时间顺延或提前,这样的移动方式不会产生不可行解. Dell' Amico 等提出 N4 邻域结构时,从工序的最大完工时间角度限制了不可行解的产生,而 Balas 等在 N6 邻域结构中,进一步从头长和尾长的角度分别判断工序的前移和后移,下面两个定理为 Balas 等^[4]提出的可行解的判定条件.

定理 3 如果工序 u 、 v 以及 v 的工件后序工序 $JS[v]$ 都在一个关键路径中,并且满足 $L(0,u)+p_u \geq L(0,JP[v])+p_{JP[v]}$,则将工序 v 移到工序 u 之前将产生非循环的完全选择.

定理 4 如果工序 u 、 v 以及 u 的工件前序工序 $JS[u]$ 都在一个关键路径中,并且满足 $L(v,\#) \geq L(JS[u],\#)$,则将工序 u 移到工序 v 之后将产生非循环的完全选择.

以工序 u 移到工序 v 之后为例, u 在 v 之前加工且两者在同一机器上,但 $JS[u]$ 与 v 之间的关系是未知的, u 、 v 、 $JS[u]$ 可能会存在如图2所示的3种情况.

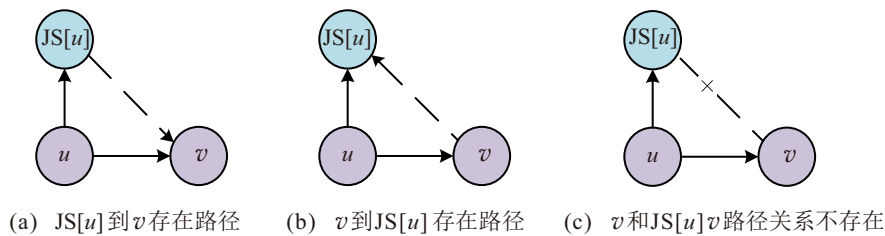


图2 定理4中工序 u 、 v 、 $JS[u]$ 可能存在的3种路径关系

1) 如图2(a)所示的第1种情况,存在 $JS[u]$ 到 v 的路径关系,那么 $JS[u]$ 应该在 v 之前加工,此时 $L(v,\#) < L(JS[u],\#)$,不满足定理4;

2) 如图2(b)所示的第2种情况,存在 v 到 $JS[u]$ 的路径关系,那么 v 应该在 $JS[u]$ 之前加工,此时 $L(v,\#) \geq L(JS[u],\#)$,满足定理4;

3) 如图2(c)所示的第3种情况,不存在 v 和 $JS[u]$ 的路径关系,那么 v 和 $JS[u]$ 的加工顺序不会被限制,此时两者的尾长大小也无法比较,定理4的条件只有部分能满足这种情况.

实际上,上述3种情况中,只有第1种生成的解是不可行解,第2种和第3种情况下均为可行解,但Balas的判定条件排除了第3种条件的部分可能性,从而导致可行解被错误地判定为不可行解.因此,这

种判定条件属于充分但不必要条件,存在部分可行解遗漏的风险.

后续很多学者的工作在上述 Balas 的条件上进行了拓展改进. Zhang 等^[5]在 N7 邻域结构中将两个关键工序 u 、 v 的范围扩展到同一机器上;赵诗奎等^[7]进一步扩展了该条件,使此条件可应用于同机器上的任意工序,取消了关键工序限制;Xie 等^[8]在 N8 邻域结构中收紧了该判定条件的约束范围,提高了邻域结构的紧凑性和搜索效率;Dauzère-Pérès 等^[18]以移动后的相关工序头尾长作为约束进行一次判断,不区分前移和后移;Grabowski 等^[19]针对关键块内工序的邻域移动,将目标工序的约束改为其两个前序(后序)工序之一的头尾长大小约束,其可行解判定条件相比上述定理3和定理4更加严格; Nasiri

等^[20]扩展了文献 [19] 的条件, 采用一个距离函数进一步细化判定的范围, 辅助判断邻域解的可行性; Tamssaouet 等^[21]通过改进广度优先搜索 (BFS) 算法寻找工序的搜索层, 再以一个充分条件判定邻域解的可行性, 并将算法参数化灵活调整其精确性. 目前, 针对 JSP 中邻域移动的可行性判定方法, 大部分以移动的工序头尾长约束其原析取图的路径关系为主, 没有从根本上改善图 2 的特殊情况. Gui 等^[11]在他的邻域结构中, 从析取图模型的角度分析了不可行解产生的原因, 并阐述了工序移动产生可行解充要条件的两个命题.

命题 1 在同一台机器上有两个工序 u 和 v , 且 u 在 v 之前加工. 当通过将 v 移到 u 之前来生成邻域解时, 该邻域解可行的充要条件是在当前解的析取图中不存在从 u 到 $JS[v]$ 的路径.

命题 2 当通过将 u 移到 v 之后来生成邻域解时, 该邻域解可行的充要条件是在当前解的析取图中不存在从 $JS[u]$ 到 v 的路径.

受上述研究启发, 本文同样从析取图模型的角度出发, 在文献 [11] 的充要条件的基础上进一步拓展, 摒弃原有的头尾长计算判断方法, 结合拓扑排序的知识和邻接矩阵判断邻接关系的特性, 直接判断两个移动工序的路径关系, 从而判断邻域移动的可行性.

3 基于邻接矩阵的可行解判定方法

3.1 邻接矩阵和可达矩阵

析取图 G 节点之间的邻接关系可用邻接矩阵表示, 而有向图的可达性用可达矩阵表示, 两者都是 0-1 矩阵. 邻接矩阵表示节点之间的直接关系, 其计算方法如下:

$$A_{ij} = \begin{cases} 1, & (i, j) \in G; \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

其中: A_{ij} 为邻接矩阵 A 中的第 i 行、第 j 列元素, (i, j) 为节点 i 到 j 的连接弧或析取弧. 可达矩阵表示节点之间直接或间接关系, 在析取图 G 中用于描述从任意节点到另一节点是否存在通路, 反映了各工序间的路径关系. 通过邻接矩阵计算可达矩阵有多种方法, 本文采用 Warshall 算法^[22]计算. 在第 1.3 节中已经将析取图中所有工序的路径关系压缩进一个拓扑排序中, 在本节中直接以这个拓扑排序构建析取图 G 的可达矩阵, 图 3 为图 1 根据调度解的拓扑排序 F_G 生成的一个邻接矩阵和可达矩阵, 为了确保简洁美观, 在矩阵中省去 0.

利用图 3 的可达矩阵可以直接查找该调度解的

	$O_{3,1}$	$O_{1,1}$	$O_{3,2}$	$O_{4,1}$	$O_{3,3}$	$O_{1,2}$	$O_{1,3}$	$O_{2,1}$	$O_{4,2}$	$O_{2,2}$	$O_{2,3}$	$O_{4,3}$
$O_{3,1}$			1				1					
$O_{1,1}$					1	1						
$O_{3,2}$				1	1							
$O_{4,1}$						1			1			
$O_{3,3}$											1	
$O_{1,2}$							1	1				
$O_{1,3}$									1			
$O_{2,1}$										1		
$O_{4,2}$											1	1
$O_{2,2}$											1	1
$O_{2,3}$												
$O_{4,3}$												

(a) 调度解的邻接矩阵

	$O_{3,1}$	$O_{1,1}$	$O_{3,2}$	$O_{4,1}$	$O_{3,3}$	$O_{1,2}$	$O_{1,3}$	$O_{2,1}$	$O_{4,2}$	$O_{2,2}$	$O_{2,3}$	$O_{4,3}$
$O_{3,1}$			1	1	1	1	1	1	1	1	1	1
$O_{1,1}$					1	1	1	1	1	1	1	1
$O_{3,2}$				1	1	1	1	1	1	1	1	1
$O_{4,1}$						1	1	1	1	1	1	1
$O_{3,3}$										1	1	1
$O_{1,2}$							1	1	1	1	1	1
$O_{1,3}$									1		1	1
$O_{2,1}$										1	1	1
$O_{4,2}$											1	1
$O_{2,2}$											1	1
$O_{2,3}$												
$O_{4,3}$												

(b) 调度解的可达矩阵

图3 调度解的邻接矩阵和可达矩阵

工序路径关系状况. 例如, 将机器 M_2 上的工序 $O_{3,2}$ 移到 $O_{2,1}$ 之后, 如果使用第 2 节提到的 Balas 的定理 4 判定此次移动的可行性, 则其定理中将移动相关的工序皆限制为关键工序, 而此次移动工序 $O_{2,1}$ 不在关键路径中, 并不适用于此处判断. 但赵诗奎等^[7]扩展了定理 4, 将此条件应用于同机器上的任意工序, 其有关工序后移的定理如下.

定理 5 对于一个可行解 s , 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, 则移动工序 u 至工序 v 之后仍然是可行解: 工序 u 为工件的末道工序, $JS[u]$ 不存在或 $JS[u]$ 存在且 $L(v, \#) \geq L(JS[u], \#)$.

若以定理 5 的条件判断此次移动, $u = O_{3,2}$, $JS[u] = O_{3,3}$, $v = O_{2,1}$, 则根据图 1(b) 的调度甘特图有, $L(v, \#) = 7$, $L(JS[u], \#) = 20 - 11 = 9$, 显然不满足不等式条件, 将会把此次移动直接判定为不可行解, 但由图 3(b) 可知, 工序 $O_{3,2}$ 到工序 $O_{2,1}$ 的值为 1, 然而工序 $O_{3,2}$ 的工件后序工序 $O_{3,3}$ 到 $O_{2,1}$ 的值为 0, 代表两者仅存在同机器上的路径关系, 不存在其他的工序约束, 移动后析取图中不会出现闭环, 因此该移动可行.

图 4 所示为工序 $O_{3,2}$ 移到 $O_{2,1}$ 之后的调度甘特图, 最大完工时间 C_{\max} 从 22 减少到 20, 说明此次移

动为一次有效的移动,进一步验证了第2节中有关使用头尾长判定的分析.

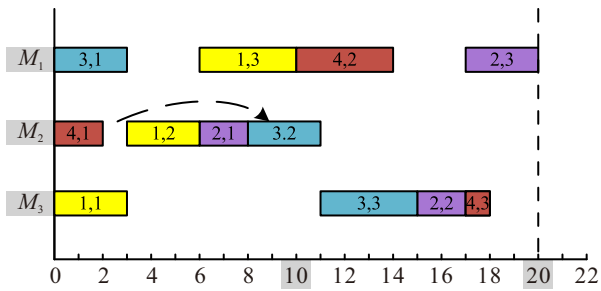


图4 工序 $O_{3.2}$ 移到 $O_{2.1}$ 之后的调度甘特图

在利用矩阵判定解的可行性过程中,完整的计算邻接矩阵将消耗大量计算资源,但实际上,仅需获取移动相关工序的路径关系即可满足判定需求.因此,本文提出一种基于邻接矩阵的双向缩减方法,以提高可行解判定的计算效率.

3.2 邻接矩阵的双向缩减

3.2.1 双向缩减的理论分析

邻接矩阵根据拓扑排序计算得到,它的大小影响可达矩阵的计算效率,从而进一步影响可行解的判定效率.邻接矩阵的缩减实际上是拓扑排序的缩减,它的理论依据要从拓扑排序的性质和析取图中的路径关系出发.在作业车间调度问题中,每个工序最多只有两个前序工序 $JP[x]$ 、 $MP[x]$ 和两个后序工序 $JS[x]$ 、 $MS[x]$,它的路径关系都是这些前、后序工序的关系延伸.根据拓扑排序的性质1和定理2,在移动工序 u 、 v 之前,由于 u 与 v 同机器,一定存在从 u 到 v 的同机器路径,但不一定存在从 u 到 v 的其他路径.

图5所示为工序移动前后路径发生的变化,图5(a)为移动之前 u 、 v 相关的工序和路径(弧)关系,图5(b)为 v 移到 u 之前相关的工序和路径关系,图5(c)为 u 移到 v 之后相关的工序和路径关系,黑色实线箭头代表工序移动前存在的路径,黑色虚线箭头代表工序移动前可能存在的路径,红色实线箭头代表工序移动后新生成的路径.在图5(a)中,存在 $u \rightarrow MS[u] \rightarrow \dots \rightarrow MP[v] \rightarrow v$ 路径.工序 v 前移时,如图5(b)所示,会断开 $MP[v] \rightarrow v$ 的路径,生成新的 $v \rightarrow u$ 的路径, u 的两个后序工序 $JS[u]$ 、 $MS[u]$ 到 v 的工件前序工序 $JP[v]$ 的路径是未知的,此时,如果移动之前存在这个路径,则析取图会产生闭环.工序 u 后移时,如图5(c)所示,会断开 $u \rightarrow MS[u]$ 的路径,生成新的 $v \rightarrow u$ 的路径, u 的工件后序工序 $JS[u]$ 到 v 的两个前序工序 $JP[v]$ 、 $MP[v]$ 的路径也是未知的.此时,如果移动之前存在这个路径,则析取图会

产生闭环.

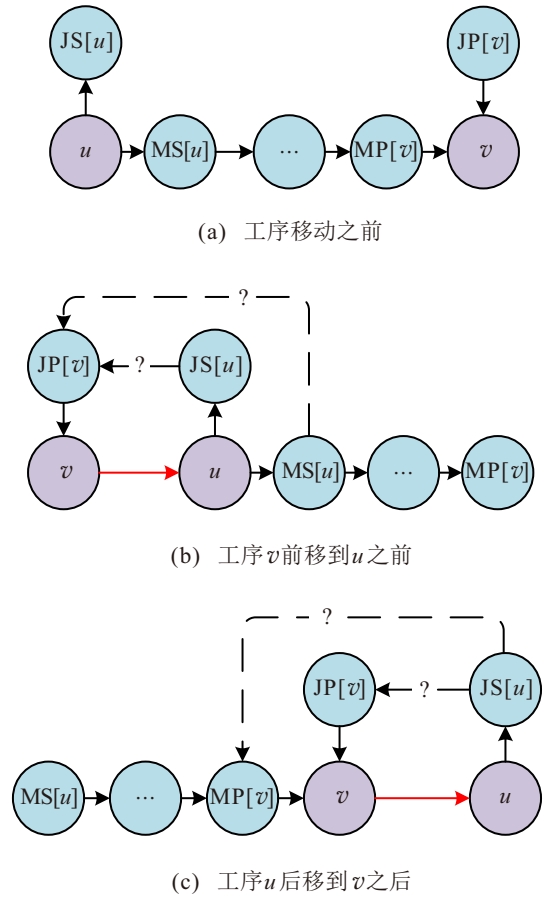


图5 移动前后路径(弧)的变化

通过上述分析,由于导致析取图闭环的工序 $JS[u]$ 、 $MS[u]$ 、 $JP[v]$ 、 $MP[v]$ 都在 u 到 v 的拓扑排序片段内,除此片段外的其他工序不会影响可行解的判定,本节重点研究在 u 到 v 拓扑排序片段内是否还有缩减的可能.

要判断 u 、 v 的移动产生的移动是否可行,只需要在移动之前判断除 u 到 v 的同机器路径外是否还存在 u 到 v 的其他路径导致析取图的闭环.对于工序前移,只需要判断 u 的两个后序工序到 $JP[v]$ 是否存在路径,反映到拓扑排序中为:截取 $MS[u]$ 或 $JS[u]$ 到 $JP[v]$ 的片段,在此片段之外的工序对可行解的判定不会发挥作用,因此在矩阵中可以舍弃.

同理,对于工序后移,只需要判断 u 的工件后序工序 $JS[u]$ 到 v 的两个前序工序是否存在路径,即:截取原拓扑排序中 $JS[u]$ 到 $JP[v]$ 或 $MP[v]$ 的片段,在矩阵中可以舍弃除此片段之外的工序.

3.2.2 双向缩减方法

基于上述理论分析,本文提出一种邻接矩阵的双向缩减方法.前文提到除 u 到 v 拓扑排序片段之外的其他工序不会影响可行解的判定,故在本节中直接对此片段进行缩减,并将此片段称为中间工序集

合 $F_G^{uv}, F_G^{uv} = \{x \in V | f_G(u) < f_G(x) < f_G(v)\}$.

若要工序 v 移到 u 之前, 缩减过程如图 6 所示, 则首先删除 F_G^{uv} 中 $JP[v]$ 之后到 v 的所有工序. 在 F_G^{uv} 中寻找距离 u 最近的 u 的直接后序工序 $DS[u]$, 并且 $DS[u]$ 在拓扑排序中的位置要在 $JP[v]$ 之前, 即有 $F_G^{uv}(DS[u]) < F_G^{uv}(JP[v])$, 否则, 说明不存在 $DS[u]$ 到 $JP[v]$ 的路径, 该移动为可行解. 随后进一步删除 F_G^{uv} 中 u 之后 (包括 u) 到 $DS[u]$ 之前的所有工序, 此时缩减后的新的中间工序集合 $F_G^{uv'} = \{x \in N | F_G^{uv}(DS[u]) \leq F_G^{uv}(x) \leq F_G^{uv}(JP[v])\}$.

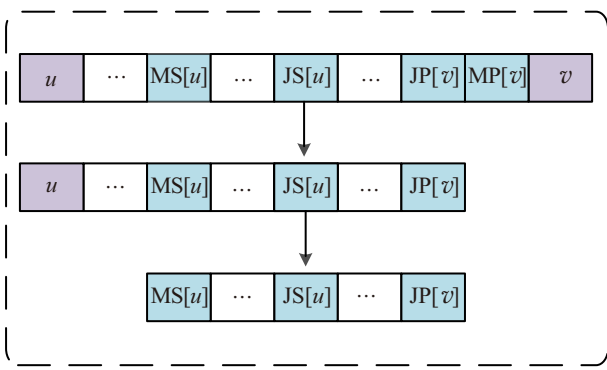


图6 工序前移时双向缩减过程

若要将工序 u 移到 v 之后, 缩减过程如图 7 所示, 则首先删除中间工序集合 F_G^{uv} 中 u 到 $JS[u]$ 的所有工序. 在中间工序集合 F_G^{uv} 中寻找距离 v 最近的 v 的直接前序工序 $DP[v]$, 且 $DP[v]$ 在拓扑排序中的位置要在 $JS[u]$ 之后, 即 $F_G^{uv}(JS[u]) < F_G^{uv}(DP[v])$, 否则, 说明不存在 $JS[u]$ 到 $DP[v]$ 的路径, 该移动为可行解. 随后进一步删除 F_G^{uv} 中 $DP[v]$ 之后到 v 之前 (包括 v) 的所有工序, 此时新的中间工序集合 $F_G^{uv'} = \{x \in N | F_G^{uv}(JS[u]) \leq F_G^{uv}(x) \leq F_G^{uv}(DP[v])\}$.

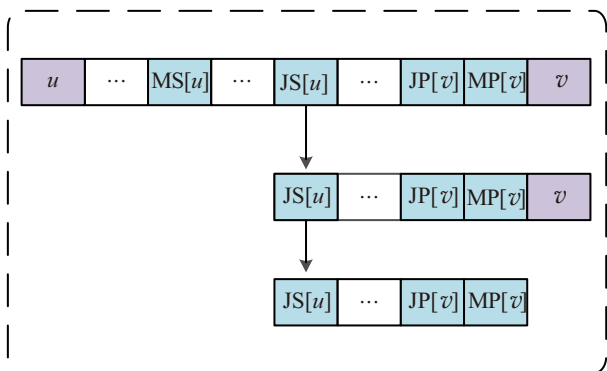


图7 工序后移时双向缩减过程

依据 F_G^{uv} 计算邻接矩阵, 邻接矩阵和可达矩阵的阶数等于 F_G^{uv} 内部工序的总数, 中间工序集合缩减后尽可能地减少了其内部工序数目的大小, 间接提升了整体的计算效率.

3.3 计算可达矩阵判定

根据定理 1, 从析取图的角度分析, 工序移动可行性的判断可以转为析取图上路径关系的判断, 通过拓扑排序可以将析取图的路径关系反映在可达矩阵中. 要判定移动的工序 u, v 是否可行, 仅需在经过缩减的矩阵中判断是否存在 u 到 v 的其他路径, 具体过程如下:

若将 v 移到 u 之前生成邻域解, 则在可达矩阵 R 中分别判断 $JS[u]$ 和 $MS[u]$ 到 $JP[v]$ 的路径关系 (如果 $JS[u]$ 或 $MS[u]$ 不在可达矩阵中, 则只需判断另一个). 若都不存在路径, 即 $R_{[JS[u]][JP[v]]} = 0, R_{[MS[u]][JP[v]]} = 0$, 则该移动为可行解; 若其中之一不为 0, 则该移动为不可行解.

若要将 u 移到 v 之后生成邻域解, 则在可达矩阵 R 中分别判断 $JS[u]$ 到 $JP[v]$ 和 $MP[v]$ 的路径关系 (如果 $JP[u]$ 或 $MP[u]$ 不在可达矩阵中, 则只需判断另一个). 若都不存在路径, 即 $R_{[JS[u]][JP[v]]} = 0, R_{[JS[u]][MP[v]]} = 0$, 则该移动为可行解; 若其中之一不为 0, 则该移动为不可行解.

例 1 按照上述流程, 将图 1 的调度解机器 M_2 上的工序 $O_{3.2}$ 移到工序 $O_{2.1}$ 之后, 对产生的图 3 中邻接矩阵和可达矩阵进行缩减, 此时 $u = O_{3.2}, DS[u] = JS[u] = O_{3.3}, v = O_{2.1}, DP[v] = MP[v] = O_{1.2}, F_G^{uv} = \{O_{3.3}, O_{1.2}\}$, 而 $MS[u] = O_{4.1}, JP[v] = 0$ 皆不在缩减后的矩阵中, 所以仅需判断 $JS[u] = O_{3.3}$ 到 $MP[v] = O_{1.2}$ 的路径情况即可判断工序 $O_{3.2}$ 到工序 $O_{2.1}$ 的路径关系.

由图 3 可知, 缩减后的邻接矩阵节点全为 0, 新的可达矩阵也必然全为 0, 说明不存在任何路径关系, 移动工序 $O_{1.1}$ 到工序 $O_{3.3}$ 之后为可行解. 缩减后的邻接矩阵从 12 阶降为 2 阶, 显著减少了矩阵的计算量.

邻接矩阵的计算是影响整体判定效率的关键所在, 尤其对于工序的大范围移动, 由单次移动生成的矩阵可能极其庞大, 因此, 在判定方法中可以尝试与其他的充分但不必要约束条件相结合, 首先应用这些约束条件判断, 减少邻接矩阵判断路径关系的使用次数, 如文献 [4] 中的定理 3 和定理 4. 它的判断速度较快, 但可能会漏掉一部分有潜在提升可能的解. 将其与基于邻接矩阵判断的方法相结合, 减少矩阵的非必要使用, 提升整体的计算速度.

图 8 为通过邻接矩阵判定和其他约束条件混合的判定流程图.

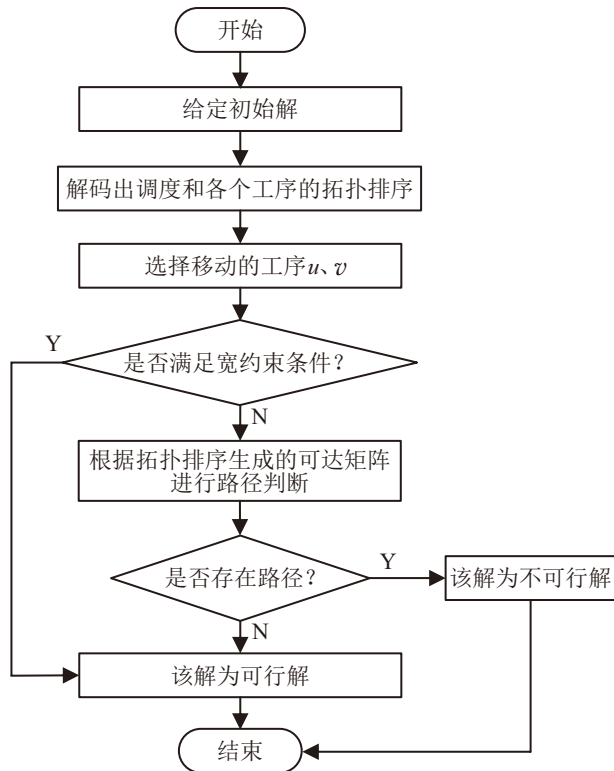


图8 混合判定可行解流程

3.4 混合算法设计

为了验证本文提出的判定方法的有效性,以遗传禁忌混合搜索算法为载体,在混合算法中融入基于邻接矩阵的可行解判定方法,使用相同的N8邻域结构^[8]进行实验测试.它的邻域解移动范围跨度大,对于可行解判定而言有较高要求,并且验证方法可行性后可以普适于其他插入型邻域结构.

在保持其他算法结构和参数不变的前提下进行对比实验,具体的混合算法设计如下.

step 1: 参数设置. 设定种群规模 P_s 、交叉概率 P_c 、变异概率 P_m 和禁忌搜索算法未改善迭代次数 $TSIterate$ 以及算法最大迭代次数 $MaxIterate$.

step 2: 初始化种群. 随机生成一个初始种群 P .

step 3: 评估适应度. 解码计算种群 P 中所有个体的适应度值.

step 4: 遗传操作. 轮盘赌选择两个父代个体 f_1 和 f_2 , 采用 POX 交叉算子^[23] 产生一个种群规模为 P_s 的新种群 P' , 按变异概率选择执行完交叉操作后的个体, 随机选择互换变异或逆序变异生成新个体.

step 5: 禁忌搜索. 对种群 P' 中每个个体执行禁忌搜索, 使用本文提出的混合判断方法判断解的可行性. 解的移动属性作为禁忌对象, 禁忌长度设置为 $L = 10 + n/m$, 选择评价价值最小且没被禁忌的邻域解作为候选解. 持续搜索得到每个个体的局部最优解, 直到最优解未改善迭代次数达到 $TSIterate$.

step 6: 更新种群. 合并初始种群 P 与新种群 P' , 选择适应度值最小的前 P_s 个解作为新种群 P .

step 7: 终止搜索. 如果当前最优解的适应度值达到下界, 或算法最大迭代次数超过 $MaxIterate$, 则输出当前最优解; 否则, 返回 step 4.

4 实验测试

本节通过实验测试验证基于邻接矩阵的作业车间调度问题可行解判定方法的有效性. 实验环境为 64 位 Windows 10 操作系统, Intel® Core™ i5-10500U CPU @ 3.1 GHz 处理器, 16.0 GB 内存. 编程环境为 Matlab 2022b, 每个算例独立求解 10 次, 取平均值作为结果.

4.1 可行解判定方法的有效性验证

为了验证本文方法的实际效果, 首先对可行解判定方法进行有效性验证. 将本文提出的可行解判定方法与其他文献方法进行比较, 使用 4 个不同规模的 LA 算例, 采用统一的邻域结构比较不同方法生成可行的邻域解数目. 如表 4 所示, 本文方法生成的可行解平均个数明显多于其他方法, 扩大了同一个体的搜索解空间, 充分展示了该方法的有效性.

表4 不同方法生成的可行的邻域解数目

算例	文献[4]	文献[18]	文献[7]	文献[20]	文献[8]	本文方法
LA06	17.4	21.4	36.8	46.6	41.4	61
LA11	25.2	36.6	55.4	69.4	64.2	92.4
LA16	30.4	35.4	51.8	64.2	57.6	81.2
LA26	51	82.8	114.6	138	125.6	194.2

本文提出的矩阵双向缩减方法是提高判定算法效率的重要组成部分, 对缩减前后不同规模的 LA 算例的耗时进行比较分析, 结果如表 5 所示. 通过实验可以观察到在加工机器数目相同的情况下, 工件数越多的算例因缩减矩阵所提升的时间就越多, 由于矩阵大部分缩减的都是同机器上的无关工序, 两者呈正相关, 且不管哪种规模的算例效果都很明显, 验证了双向缩减方法对优化耗时的提升作用.

表5 缩减前后耗时对比

算例	规模	缩减前耗时 / ms	缩减后耗时 / ms	耗时提升百分比 / %
LA01-05	10×5	2.81	1.59	43.42
LA06-10	15×5	5.84	3.1	46.92
LA11-15	20×5	27.54	8.96	67.47
LA16-20	10×10	10.28	5.41	47.37
LA21-25	15×10	32.09	10.21	68.18
LA26-30	20×10	70.23	18.05	74.29
LA31-35	30×10	337.38	56.83	83.15
LA36-40	15×15	63.23	22.97	63.67

4.2 混合算法测试

为了验证本文提出的混合算法的实际效果, 本文采用遗传禁忌混合算法对 FT、LA、ORB 等 53 个不同规模的基准算例进行测试, 经过多次试验, 具体参数设置为: 种群为 300, 遗传算法交叉概率为 0.9, 变异概率为 0.1, 终止条件设置为循环代数 300 代或最优解连续 30 代不更新. 禁忌搜索终止条件设置为循环代数 1000 代或最优解连续 60 代不更新. LB_{best} 为目前最优值, C_{max} 为本文算法最优值, C_{avg} 为

本文算法求解平均值, RE为相对误差, $RE = 100 \times (C_{max} - LB_{best}) / LB_{best}$, MRE为测试算例的平均相对误差, T_{avg} 为算法平均求解时间.

对于测试的 53 个 JSP 典型算例, 测试结果如表 6 所示, 除了 LA 系列部分较难求解的算例, 其他绝大部分均成功地计算到最优解, 并且在计算耗时上也有较好的表现. 对于较难求解的算例, 除规模较大的 LA29 和 LA36 外, 其他困难算例的 RE 均小于 0.5.

表6 53个典型算例的求解结果

问题	规模	LB_{best}	C_{max}	C_{avg}	RE	T_{avg}	问题	规模	LB_{best}	C_{max}	C_{avg}	RE	T_{avg}
LA01	10×5	666	666	666	0	1.46	LA28	20×10	1216	1216	1216.6	0	292.69
LA02	10×5	655	655	655	0	5.74	LA29	20×10	1152	1166	1176.2	1.22	722.41
LA03	10×5	597	597	597	0	11.32	LA30	20×10	1355	1355	1355	0	98.43
LA04	10×5	590	590	590	0	5.7	LA31	30×10	1784	1784	1784	0	84
LA05	10×5	593	593	593	0	1.23	LA32	30×10	1850	1850	1850	0	33.91
LA06	15×5	926	926	926	0	1.87	LA33	30×10	1719	1719	1719	0	47.73
LA07	15×5	890	890	890	0	3.31	LA34	30×10	1721	1721	1721	0	92.27
LA08	15×5	863	863	863	0	1.81	LA35	30×10	1888	1888	1888	0	86.56
LA09	15×5	951	951	951	0	1.78	LA36	15×15	1268	1281	1288.5	1.03	168.38
LA10	15×5	958	958	958	0	2.56	LA37	15×15	1397	1397	1410.2	0	593.78
LA11	20×5	1222	1222	1222	0	3.19	LA38	15×15	1196	1202	1212	0.50	711.53
LA12	20×5	1039	1039	1039	0	2.68	LA39	15×15	1233	1238	1239	0.40	289.04
LA13	20×5	1150	1150	1150	0	2.87	LA40	15×15	1222	1228	1229.6	0.49	287.43
LA14	20×5	1292	1292	1292	0	7.98	ORB01	10×10	1059	1059	1074.2	0	93.95
LA15	20×5	1207	1207	1207	0	9.2	ORB02	10×10	888	888	889.2	0	45.24
LA16	10×10	945	945	945.8	0	54.09	ORB03	10×10	1005	1005	1024.4	0	84.61
LA17	10×10	784	784	784.6	0	36.56	ORB04	10×10	1005	1005	1013.4	0	57.68
LA18	10×10	848	848	848	0	13.68	ORB05	10×10	887	887	888.8	0	59.82
LA19	10×10	842	842	843.4	0	20.17	ORB06	10×10	1010	1010	1016	0	91.11
LA20	10×10	902	902	906.5	0	43.19	ORB07	10×10	397	397	398	0	25.08
LA21	15×10	1046	1046	1059.7	0	150.46	ORB08	10×10	899	899	914.8	0	104.91
LA22	15×10	927	927	932.6	0	184.26	ORB09	10×10	934	934	942.6	0	86.77
LA23	15×10	1032	1032	1032	0	27.11	ORB10	10×10	944	944	944	0	70.35
LA24	15×10	935	938	941.6	0.32	239.41	FT06	6×6	55	55	55	0	1.31
LA25	15×10	977	977	980.6	0	219.79	FT10	10×10	930	930	931.1	0	116.28
LA26	20×10	1218	1218	1218	0	129.36	FT20	20×5	1165	1165	1172	0	195.44
LA27	20×10	1235	1237	1239.2	0.16	218.7							

针对 LA21 ~ LA29 和 LA36 ~ LA40 共 14 个较难求解的算例, 将本文的结果与文献 [24, 9, 25-26] 等近几年与邻域结构有关的文献分别进行比较, 测试结果如图 9 和表 7 所示. 在图 9 中, 对比数据经归一化处理, 数值越接近 0 (颜色越浅) 表示越接近当前算例最优值. 由于文献 [9] 并未给出全部的计算结果, 在热力图中只选取了几个代表性算例. 由上述图表可知, 在求解质量上, 本文算法的平均相对误差 MRE 为 0.294, 略差于文献 [26] 的 0.195, 优于文献 [9, 24-25] 的 MRE: 0.58、0.92 和 1.742, 并且在求到最优解的个数上也明显优于这 3 个文献的方法, 验

证了本文算法的有效性.

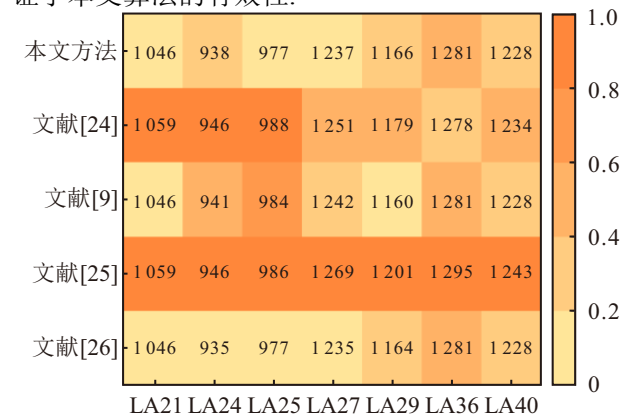


图9 不同算法结果对比热力图

表7 不同算法结果对比

问题	规模	LB	本文方法		文献[24]		文献[9]		文献[25]		文献[26]	
			C_{\max}	RE	C_{\max}	RE	C_{\max}	RE	C_{\max}	RE	C_{\max}	RE
LA21	15×10	1046	1046	0	1059	1.24	1046	0	1059	1.24	1046	0
LA22	15×10	927	927	0	932	0.54	—	—	935	0.86	927	0
LA23	15×10	1032	1032	0	1032	0	—	—	1032	0	1032	0
LA24	15×10	935	938	0.32	946	1.18	941	0.64	946	1.17	935	0
LA25	15×10	977	977	0	988	1.13	984	0.72	986	0.92	977	0
LA26	20×10	1218	1218	0	1218	0	—	—	1218	0	1218	0
LA27	20×10	1235	1237	0.16	1251	1.30	1242	0.49	1269	2.75	1235	0
LA28	20×10	1216	1216	0	1216	0	—	—	1239	1.89	1216	0
LA29	20×10	1152	1166	1.22	1179	2.34	1160	0.69	1201	4.25	1164	1.04
LA36	15×15	1268	1281	1.03	1278	0.79	1281	1.03	1295	2.12	1281	1.03
LA37	15×15	1397	1397	0	1419	1.57	—	—	1415	1.28	1397	0
LA38	15×15	1196	1202	0.50	1204	0.67	—	—	1246	4.18	1198	0.17
LA39	15×15	1233	1238	0.40	1247	1.14	—	—	1258	2.02	1233	0
LA40	15×15	1222	1228	0.49	1234	0.98	1228	0.49	1243	1.71	1228	0.49
MRE	—	—	—	0.294	—	0.92	—	0.58	—	1.742	—	0.195

5 结论

本文提出了一种基于邻接矩阵判断工序移动可行性的方法,与现有方法相比,本文的独特之处是将数学中的拓扑排序和可达矩阵等相关知识与工序移动的可行解判定研究相结合,实现了邻域解的快速判定,为该方面的研究提供了一种新的思路.此外,本文设计了一种邻接矩阵的双向缩减方法,显著减小了矩阵的大小,并通过与传统的判定方法相结合,进一步提高了整体的计算效率.通过与多种算法对不同规模 JSP 基准算例的测试结果对比,验证了所提出算法的稳定性和有效性.当前工作仅对工序移动的可行性判定进行了改进,而关于混合算法全局搜索性能与局部搜索性能的高效协同研究是下一步的重点工作.

参考文献 (References)

- [1] 吴贝贝,张宏立,王聪,等.基于正态云模型的状态转移算法求解多目标柔性作业车间调度问题[J].控制与决策,2021,36(5):1181-1190.
(Wu B B, Zhang H L, Wang C, et al. State transition algorithm based on normal cloud model for solving multi-objective flexible job shop scheduling problem[J]. Control and Decision, 2021, 36(5): 1181-1190.)
- [2] Dell'Amico M, Trubian M. Applying tabu search to the job-shop scheduling problem[J]. *Annals of Operations Research*, 1993, 41(3): 231-252.
- [3] Nowicki E, Smutnicki C. A fast taboo search algorithm for the job shop problem[J]. *Management Science*, 1996, 42(6): 797-813.
- [4] Balas E, Vazacopoulos A. Guided local search with shifting bottleneck for job shop scheduling[J]. *Management Science*, 1998, 44(2): 262-275.
- [5] 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.
- [6] 潘全科,王凌,高亮,等.基于差分进化与块结构邻域的作业车间调度优化[J].*机械工程学报*,2010,46(22):182-188.
(Pan Q K, Wang L, Gao L, et al. Differential evolution algorithm based on blocks on critical path for job shop scheduling problems[J]. *Journal of Mechanical Engineering*, 2010, 46(22): 182-188.)
- [7] 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 S K, Fang S L, Gu X J. Idle time neighborhood search genetic algorithm for job shop scheduling[J]. *Computer Integrated Manufacturing Systems*, 2014, 20(8): 1930-1940.)
- [8] Xie J, Li X, Gao L, et al. A new neighbourhood structure for job shop scheduling problems[J]. *International Journal of Production Research*, 2023, 61(7): 2147-2161.
- [9] 薛玲玲.作业车间调度的块结构邻域搜索遗传算法[J].*计算机集成制造系统*,2021,27(10):2848-2857.
(Xue L L. Block structure neighborhood search genetic algorithm for job-shop scheduling[J]. *Computer Integrated Manufacturing Systems*, 2021, 27(10): 2848-2857.)
- [10] 巴智勇,袁逸萍,裴国庆,等.作业车间调度的多工序精确联动邻域结构混合进化算法[J].*计算机集成制造系统*,2024,30(2):537-552.
(Ba Z Y, Yuan Y P, Pei G Q, et al. Hybrid evolutionary algorithm with multi-operation precise joint movement neighborhood structure for job shop scheduling

- problem[J]. *Computer Integrated Manufacturing Systems*, 2024, 30(2): 537-552.)
- [11] Gui L, Li X, Gao L, et al. Necessary and sufficient conditions for feasible neighbourhood solutions in the local search of the job-shop scheduling problem[J]. *Chinese Journal of Mechanical Engineering*, 2023, 36(4): 157-172.
- [12] Sun L, Huang Z, Zhang H, et al. Infeasibility test algorithm and fast repair algorithm of job shop scheduling problem[J]. *Journal of Southeast University (English Edition)*, 2011, 27(1): 88-91.
- [13] 王艳红, 付威通, 张俊, 等. 基于改进近端策略优化算法的柔性作业车间调度[J]. *控制与决策*, 2025, 40(6): 1883-1891.
(Wang Y H, Fu W T, Zhang J, et al. Flexible job-shop scheduling based on improved proximal policy optimization algorithm[J]. *Control and Decision*, 2025, 40(6): 1883-1891.)
- [14] 罗聪, 龚文引. 混合分解多目标进化算法求解绿色置换流水车间调度问题[J]. *控制与决策*, 2024, 39(8): 2737-2745.
(Luo C, Gong W Y. A hybrid multi-objective evolutionary algorithm based on decomposition for green permutation flow-shop scheduling problem[J]. *Control and Decision*, 2024, 39(8): 2737-2745.)
- [15] 黄学文, 王达, 王强. 作业车间调度的拓扑排序及其在邻域解动态构造中的应用[J]. *系统工程理论与实践*, 2024, 44(5): 1680-1698.
(Huang X W, Wang D, Wang Q. Topological order for job-shop scheduling and its application in dynamically generating neighbouring solutions[J]. *Systems Engineering — Theory & Practice*, 2024, 44(5): 1680-1698.)
- [16] Braune R, Zäpfel G, Affenzeller M. Enhancing local search algorithms for job shops with min-sum objectives by approximate move evaluation[J]. *Journal of Scheduling*, 2013, 16: 495-518.
- [17] Marchetti-Spaccamela A, Nanni U, Rohnert H. Maintaining a topological order under edge insertions[J]. *Information Processing Letters*, 1996, 59(1): 53-58.
- [18] Dauzère-Pérès S, Paulli J. An integrated approach for modeling and solving the general multiprocessor job-shop scheduling problem using tabu search[J]. *Annals of Operations Research*, 1997, 70(0): 281-306.
- [19] Grabowski J, Wodecki M. A very fast tabu search algorithm for job shop problem[C]. *Metaheuristic optimization via memory and evolution: Tabu search and scatter search*. New York: Springer, 2005: 117-144.
- [20] Nasiri M M, Kianfar F. A new neighborhood for the job shop scheduling problem[J]. *Advanced Materials Research*, 2012, 433: 1540-1544.
- [21] Tamssaouet K, Dauzère-Pérès S. A general efficient neighborhood structure framework for the job-shop and flexible job-shop scheduling problems[J]. *European Journal of Operational Research*, 2023, 311(2): 455-471.
- [22] 叶红. 可达矩阵的 Warshall 算法实现[J]. *安徽大学学报: 自然科学版*, 2011, 35(4): 31-35.
(Ye H. Reach ability matrix by Warshall algorithm[J]. *Journal of Anhui University: Natural Science Edition*, 2011, 35(4): 31-35.)
- [23] 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(23): 83-87.
(Zhang C Y, Rao Y Q, Liu X J, et al. An improved genetic algorithm for the job shop scheduling problem[J]. *China Mechanical Engineering*, 2004(23): 83-87.)
- [24] Mahmud S, Abbasi A, Chakraborty R K, et al. Multi-operator communication based differential evolution with sequential tabu search approach for job shop scheduling problems[J]. *Applied Soft Computing*, 2021, 108: 107470.
- [25] Viana M S, Morandin Junior O, Contreras R C. A modified genetic algorithm with local search strategies and multi-crossover operator for job shop scheduling problem[J]. *Sensors*, 2020, 20(18): 5440.
- [26] Peng C, Wu G, Liao T W, et al. Research on multi-agent genetic algorithm based on tabu search for the job shop scheduling problem[J]. *PloS One*, 2019, 14(9): 1-19.

作者简介

于丰顺 (1999-), 男, 硕士生, 主要研究方向为作业车间生产调度、智能优化算法, E-mail: 1553161219@qq.com;

赵诗奎 (1984-), 男, 教授, 博士, 博士生导师, 主要研究方向为运筹优化、车间生产调度、智能优化算法, E-mail: me_zhaosk@ujn.edu.cn;

仵政源 (2001-), 男, 硕士生, 主要研究方向为装配车间生产调度、智能优化算法, E-mail: 1426095214@qq.com;

李彤 (2001-), 男, 硕士生, 主要研究方向为绿色节能调度、智能优化算法, E-mail: 1564778434@qq.com.