

# 资源受限的飞机总装脉动生产线节拍转换调度建模

陆志强, 王浩宇

(同济大学 机械与能源工程学院, 上海 201804)

**摘要:** 以飞机总装脉动生产线节拍转换过程为实际生产背景, 提出基于项目拆分的资源受限项目节拍转换调度问题, 并建立了以最小化转换期时长为目标函数的数学模型。根据项目节拍转换调度问题的特点, 采用双层迭代算法进行求解。结果表明: 该算法有效缩短生产线节拍转换时间, 对不同项目组合的优化程度为 1.1%~23.4%。

**关键词:** 多项目调度; 项目拆分; 作业划分; 飞机总装脉动生产线; 禁忌搜索

中图分类号: TP29

文献标志码: A

## Modeling of Resource Constrained Project Takt Transition Scheduling Problem for Aircraft Pulsating Assembly Line

LU Zhiqiang, WANG Haoyu

(School of Mechanical Engineering, Tongji University, Shanghai 201804, China)

**Abstract:** A resource constrained project takt transition scheduling problem based on project splitting was proposed under the background of aircraft pulsating assembly line takt transition process, and a mathematical model was established with the objective function of minimizing total transition time. According to the feature of the problem, a two-level iterative algorithm was developed. It is shown that the algorithm can decrease the transition time effectively and the degree of optimization for different project portfolios is 1.1%~23.4%.

**Key words:** multi-project scheduling; project splitting; job allocation; aircraft pulsating assembly line; Tabu search

高, 已经广泛应用于大型飞机的装配过程<sup>[1]</sup>。在飞机总装脉动生产线的生产过程中, 装配线被划分为多个资源共享的装配单元, 总装过程被拆分为多个作业子集分配到各装配单元上, 每个作业子集包含多个作业, 这些作业子集可以视为飞机总装项目的子项目。飞机沿生产线方向依次缓慢通过各单元, 同时各单元按照装配计划完成装配作业。在一个装配周期内, 多架飞机在多个装配单元上同时进行装配。因此, 飞机总装脉动生产线的调度问题可抽象为一类多项目调度问题。此外, 当一类飞机装配项目生产结束时, 需要对生产线上的各类资源进行调整, 以适应新的装配项目。在原项目到新项目的转换过程中, 原类型飞机不断完工离开生产线, 新类型飞机逐渐进入各装配单元, 这种 2 个项目在装配线上共存的时期, 称为项目转换期。由于 2 个项目的资源需求和节拍时间的差异, 原有调度计划可能无法正常实施, 致使项目延期, 因此需要对项目转换期内的装配计划进行调整, 重新分配各装配单元对应的作业子集, 调整作业时间, 制定新的装配计划以优化转换过程, 缩短项目工期。目前针对这类问题的研究较少, 实际生产时多采用牺牲生产效率、降低装配线速度的方式适应新项目。

为了研究飞机总装脉动生产线的节拍转换过程, 提出了基于项目拆分的资源受限项目节拍转换调度问题(RCPTTSP-PS)。RCPTTSP-PS 在理论上是一类具有多项目并行调度、复杂约束并且考虑时空多维度的大规模多项目协同调度问题。目前尚没有对 RCPTTSP-PS 的直接研究, 但是对基础的多项目协同调度问题(RCMPSP)的研究已经十分透彻。学界常见的求解 RCMPSP 的方法有 2 种, 即 Goncalves 等<sup>[2]</sup>提出的组合方法和 Kurtulus 等<sup>[3]</sup>提出的独立方法, 前者是通过虚任务将多项目连接成单项目后求解, 后者是对每个项目给出各自的关键路

飞机总装脉动生产线因其产线平稳、生产柔性

收稿日期: 2021-02-09

基金项目: 国家自然科学基金(61473211)

第一作者: 陆志强(1968—), 男, 教授, 博士生导师, 工学博士, 主要研究方向为物流与供应链的建模和优化以及生产工

程等。E-mail: zhiqianglu@tongji.edu.cn



论文  
拓展  
介绍

径进行并行求解。Browning等<sup>[4]</sup>认为第二种方法更加贴近真实情况,将是多项目问题求解的方向。RCMPSP是一个NP难题<sup>[5]</sup>,无法使用精确算法求解,因此学界多采用启发式算法和元启发算法进行求解。优先级规则是启发式算法的主流,Browning等<sup>[4]</sup>对20种主流的优先级规则进行了评价,对于不同的项目特性给出了不同优先级规则使用的建议。Wang等<sup>[6]</sup>和Kurtulus<sup>[7]</sup>分别引入了随机作业时长和不同的项目延迟惩罚,并分析了相同的优先级规则在不同环境下的求解质量和鲁棒性。元启发算法相较于启发式算法,求解精度更高,但是迭代时间长,适用于静态项目求解。Kim等<sup>[8]</sup>提出了一种使用模糊逻辑控制器进行参数调整的混合遗传算法以求解RCMPSP;陈俊杰等<sup>[9]</sup>提出了以蚁群算法为基础的两阶段算法,对考虑柔性资源的RCMPSP效果显著;陈浩杰等<sup>[10]</sup>使用超启发式遗传规划算法,对多目标RCMPSP进化出最优的优先级规则。

对RCMPSP的扩展问题——基于项目拆分的资源受限项目调度问题(RCPMSP-PS),由陆志强等<sup>[11]</sup>首先提出并进行了详细阐述,根据项目拆分的特点,设计了双层迭代算法进行求解。朱宏伟等<sup>[12]</sup>在RCMPSP-PS的基础上,提出了考虑资源转移时间的RCMPSP-PS问题,通过自适应遗传算法有效避免了不合理的资源转移。宗保氏等<sup>[13]</sup>在RCMPSP-PS的基础上提出了考虑项目拆分的资源投入调度问题,通过遗传算法进行搜索,以优化项目过程的资源投入。在考虑新项目加入调度方面,Chen等<sup>[14]</sup>考虑了突发订单对生产线的干扰,并对基础计划利用优先级规则进行实时调整;Yassine等<sup>[15]</sup>对产品开发过程中具有优先级的项目迭代过程进行了建模分析,并利用遗传算法求解,最终给出了可供管理者使用的决策矩阵。Abrantes等<sup>[16]</sup>对产品开发过程也进行了研究,重点分析产品开发过程中新项目开始时的人力资源配置问题,通过跨国企业的人力资源配置模式分析,优化管理流程以快速响应产品开发的动态环境。

综上所述,大多数多项目调度研究只考虑了周期内的并行调度,对于有新项目加入的多项目调度,也仅考虑了各项目独立的情况。RCPTTSP-PS不仅将项目拆分与作业调度同时纳入决策模型,还考虑了转换前后的2个装配项目对于资源、工期和节拍时间的差异,以及2个项目相互影响的复杂情况。现有算法难以求解RCPTTSP-PS,因此对RCPTTSP-PS进行建模与算法研究,有着重要的理论意义。

## 1 问题描述及数学模型

### 1.1 问题描述

RCPTTSP-PS以飞机总装脉动生产线的节拍转换过程为实际应用背景。在脉动生产线上,根据实际生产线长度 $L$ 、飞机长度 $L_A$ 、安全距离 $S_D$ ,装配线被分为 $N$ 个装配单元,总装项目也被拆分为 $N$ 个作业子集并分配给各个装配单元。由于新项目的进入和原项目的完工,整个转换过程在时间上被分为多个阶段。如图1所示,飞机沿装配线方向逐步完成装配,转换期前后,装配线上都仅有一种类型飞机,装配线稳定无变动,而转换期内,装配线由2种飞机(G0,G1)的作业子集构成,每过一个周期时间,飞机都会沿装配线方向移动一个装配单元,各装配单元对应的作业子集都会改变。在此过程中每个周期内调度计划均不同,周期内的资源需求和周期时间也均会发生变化。以转换期时长最小化为优化目标,建立数学模型并求解。

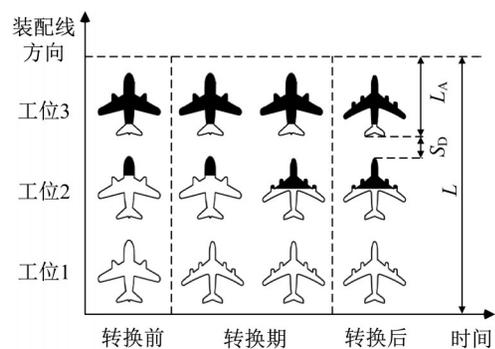


图1 飞机移动装配线转换过程

Fig.1 Aircraft moving assembly line transition process

为了描述拆分之后的时序约束关系,给出如下定义:

**定义1 真实时序约束。**对于在初始项目网络中存在时序约束的2个作业,若项目拆分后2个作业仍处于同一个作业子集内,则2个作业间的时序约束仍然存在,称该约束为真实时序约束。

**定义2 虚拟时序约束。**对于在初始项目网络中存在时序约束的2个作业,若项目拆分后2个作业不处于同一个作业子集内,则2个作业间不存在时序约束,仅在拆分时需要考虑,称该约束为虚拟时序约束。

**定义3 同源作业子集。**如果一系列作业子集拆分自同一架飞机的总装项目,就称这一系列作业子集为同源作业子集,也被称为同源子项目。

图2为转换过程中项目的拆分方案以及时序约

束关系。其中,数字代表装配作业,实线有向弧代表真实时序约束,虚线有向弧代表虚拟时序约束。在考虑了拆分的飞机装配过程中,不同周期内的同源作业子集间会有虚拟时序约束,同时同源作业子集间可以进行作业的移动以优化拆分方案。在调度时,只有真实时序约束会被考虑;在作业移动时,真实时序约束和虚拟时序约束都需要纳入考虑。

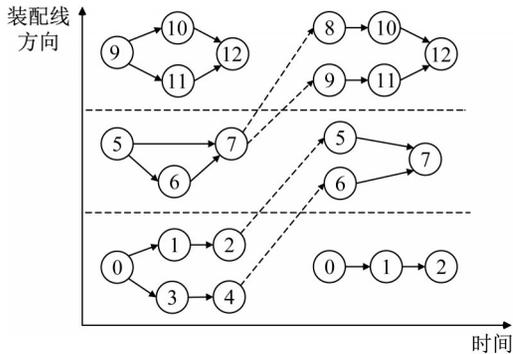


图2 转换过程拆分方式

Fig.2 Splitting method in transition process

此外,为了着重研究项目转换,给出如下假设:作业执行不可中断,当前周期内所有作业完工后才能进入下一周期,作业只有一种执行模式。

1.2 数学模型

根据以上问题描述,给出具体参数定义。模型中部分参数如表1所示。

表1 部分参数符号及说明

Tab.1 Some parameter symbols and their descriptions

参数	定义
$g$	装配项目, $g=0, 1$ , 代表项目 G0, G1
$n$	装配单元序号, $n=1, 2, \dots, N$
$m$	装配周期序号, $m=1, 2, \dots, M$
$v$	装配项目序号, $v=1, 2, \dots, V$
$j$	作业符号, $j=1, 2, \dots, J$
$k$	资源符号, $k=1, 2, \dots, K$
$d_j$	作业 $j$ 的执行时间
$r_{jk}$	作业 $j$ 对资源 $k$ 的需求量
$R_k$	资源 $k$ 的投入量
$P_j$	作业 $j$ 的紧前作业合集
$p$	作业 $j$ 的紧前作业, $p \in P_j$
$Q_{gv}$	装配项目 $g$ 的第 $n$ 个装配单元的作业集
$S_{mv}$	作业子集
$t_{jmv}$	作业子集 $S_{mv}$ 中作业 $j$ 的开始时间
$W$	一个足够大的正数
$T_m$	周期 $m$ 的周期时间
$C$	转换期总时长

(1) 决策变量

$x_{ijmv}$  为 0、1 变量,若装配周期  $m$  时装配单元  $n$  上

作业  $j$  的开始时间为  $t$ , 则  $x_{ijmv} = 1$ , 否则  $x_{ijmv} = 0$ 。

$y_{jmv}$  为 0、1 变量,若作业  $j$  属于作业子集  $S_{mv}$ , 则  $y_{jmv} = 1$ , 否则  $y_{jmv} = 0$ 。

$z_{pjmv}$  为 0、1 变量,表示作业的真实时序约束,若作业  $p$  是作业  $j$  的紧前作业,并且拆分后在同一个作业子集  $S_{mv}$  中,则  $z_{pjmv} = 1$ , 否则  $z_{pjmv} = 0$ 。

(2) 目标函数

$$\min C = \sum_{m=1}^M T_m \quad (1)$$

(3) 约束条件

$$g = \begin{cases} 0, & n > m \\ 1, & \text{其他} \end{cases} \quad (2)$$

$$(g-1)N + v - m + n = 0 \quad (3)$$

$$\bigcup_{m=1}^v S_{mv} = \bigcup_{n=N+1-v}^N Q_{gv}, g=0, \forall v \quad (4)$$

$$\bigcup_{m=v}^M S_{mv} = \bigcup_{n=1}^{N-v} Q_{gv}, g=1, \forall v \quad (5)$$

$$\sum_{v=1}^{\max\{M-m, N-n\}} y_{p(m+v)(n+v)} \leq W(1-y_{jmv}), m < M, n < N, j \in S_{mv}, p \in P_j \quad (6)$$

$$z_{pjmv} = y_{pmm} y_{jmv}, \forall m, n, j \in S_{mv}, p \in P_j \quad (7)$$

$$t_{jmv} = t x_{ijmv}, \forall m, n, j \in S_{mv} \quad (8)$$

$$(t_{pmm} + d_p) z_{pjmv} \leq t_{jmv}, \forall m, n, j \in S_{mv}, p \in P_j \quad (9)$$

$$\sum_{t=0}^{T_m} x_{ijmv} = 1, \forall m, n, j \in S_{mv} \quad (10)$$

$$t_{jmv} + d_j \leq T_m, \forall m, n, j \in S_{mv} \quad (11)$$

$$\sum_{n=1}^N r_{jk} y_{jmv} \sum_{\max\{0, t-d_j\}}^t x_{ijmv} \leq R_k, \forall t, k, m, j \in S_{mv} \quad (12)$$

式(1)是目标函数,即要求转换过程的总时间最小。式(2)和式(3)表达了4个参数  $g, v, m, n$  之间的关系,如图3所示。图3中,一个圆角矩形表示一个作业子集,该作业子集所属的飞机类型  $g$  与项目序号  $v$  用于确定作业子集在生产计划中的位置,具有相同  $g, v$  的作业子集属于同一飞机,互为同源作业子集,图3中双向箭头表示同源关系。作业子集对应装配周期  $m$  与装配单元  $n$ ,用以确定作业子集的位置,  $m$  相同的作业在同一周期内装配,共享线边资源。

式(4)和式(5)保证每一组同源作业子集完成了所有需要完成的作业;式(6)表示所有作业的紧前作业不能分配到该作业所在作业子集的后序同源作业子集中;式(7)表示真实时序约束  $z_{pjmv}$  与项目拆分决策变量  $y_{jmv}$  的关系,当作业与其紧前作业被同时分配到同一个作业子集时,真实时序约束生效;式(8)表

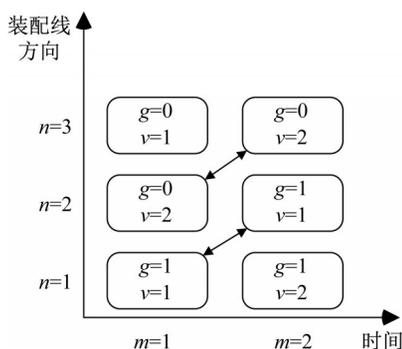


图3 作业子集序号关系

Fig.3 Relationship between sub-projects' sequence numbers

示作业开始时间与决策变量 $x_{ijm}$ 的关系;式(9)限制作业子集内的优先次序关系,作业必须在其紧前作业完工后才能开始;式(10)表示每个作业只开始一次,并且开始后不能中断;式(11)表示每个作业都要在节拍时间内完成;式(12)表示资源约束,即每个时刻所使用的资源不能超过资源上限。

## 2 算法框架

针对所研究的问题,采用双层迭代算法。上层使用双重禁忌搜索(DTS)算法对作业拆分方案进行搜索,在作业子集之间进行作业的移动以搜索最优的拆分决策;下层使用基于最小最晚结束时间(MIN LFT)的优先级规则并结合串行调度生成机制进行调度,将每个作业子集的完工时间及资源使用比反馈给上层的双重禁忌搜索算法,通过不断的循环迭代使目标函数最小化。

### 2.1 双层迭代算法整体步骤

**步骤1** 根据非转换期的项目拆分方案,依照进入装配线的顺序进行项目组合,构造原始拆分方案,并利用最小最晚结束时间优先级规则进行串行调度,求解初始转换期总时长 $C_0$ ,令 $C_{\text{best}} = C_0$ 。

**步骤2** 设置总迭代次数为 $E$ ,初始迭代次数 $e = 1$ 。

**步骤3** 计算各作业子集的完工时长 $C_{mm}$ ,利用概率密度函数 $\psi(m, n, i)$ 计算作业移动的起终点项目的选取概率,其中 $i$ 表示同源作业子集之间的距离。

**步骤4** 判断是否有满足特赦条件的起终点的选取概率,如果有,转步骤6。

**步骤5** 采用偏移随机抽样(RBRS)算法和相对禁忌表确定作业移动的起终点项目序号。

**步骤6** 利用作业选取规则选取移动作业。

**步骤7** 查询移动后的项目关键链是否在绝对禁忌表中,若在则重新选取移动作业。如果所有作业移动后的项目关键链均在绝对禁忌表中,就转步骤5重新选择起终点对。

**步骤8** 采用最小最晚结束时间优先级规则对各周期内的拆分方案进行串行调度,计算每个周期的完工时间 $T_m$ 、各周期内资源利用率 $f_{bm}$ 以及当前迭代的转换期总时长 $C_e$ ,若 $C_e < C_{\text{best}}$ ,则 $C_{\text{best}} = C_e$ 。

**步骤9** 将起终点对加入相对禁忌表。

**步骤10** 对移动后的拆分方案计算关键路径长度,判断是否要加入绝对禁忌表。

**步骤11**  $e = e + 1$ ,若 $e < E$ ,则转步骤3。

**步骤12** 输出 $C_{\text{best}}$ 。

### 2.2 基于双重禁忌搜索算法的作业移动

提出了一种双重禁忌搜索算法,加入了相对禁忌表和绝对禁忌表来处理不同的禁忌情况。相对禁忌表即为常规禁忌搜索算法中的禁忌表,可持续更替。绝对禁忌表则存储无法达到历史最优的解的特征,在搜索过程中不断增加。相比传统元启发算法,该算法通过绝对禁忌表增加约束,缩减解集的搜索空间,利用相对禁忌表和启发式规则来控制解的搜索方向,两者相结合进行最优解的搜索。

#### 2.2.1 作业起终点项目的选择

首先计算出当前每个作业子集 $S_{mm}$ 的完工时间 $C_{mm}$ ,通过 $C_{mm}$ 计算起点与终点的权重比 $\rho_{mm}$ 。 $\rho_{mm}$ 的计算式如下所示:

$$\rho_{mm} = \begin{cases} C_{mm} - \min\{C_{ms}\}, s \in N, \text{作为起点} \\ \max\{C_{ms}\} - C_{mm}, s \in N, \text{作为终点} \end{cases}$$

由于不同的作业子集间可能拥有同一个项目的继承关系,因此仅有同源作业子集间可以进行作业移动。同源作业子集有如下性质:

**性质1** 对于一系列同源作业子集 $S_{mm}$ ,根据式(1),有 $m, n$ 之差为常数。定义 $S_{mm}$ 的同源作业子集为 $S_{(m+i)(n+i)}, i \in \mathbf{Z}, i \neq 0$ 。

在性质1的基础上引入 $i$ 作为作业移动起终点作业子集在其整装项目中的序号差,以确保选取的起终点作业子集为同源作业子集。设概率密度函数

$$\psi(m, n, i) = \frac{(\rho_{mm}\rho_{(m+i)(n+i)} + \epsilon)^\alpha}{\sum_{m=1}^M \sum_{n=1}^N \sum_i (\rho_{mm}\rho_{(m+i)(n+i)} + \epsilon)^\alpha}$$

式中: $\epsilon > 0$ 为参数,用于保证 $\rho_{mm} = 0$ 的项目也有一定概率被选择; $\alpha$ 为用于控制过程随机性的参数。

$\psi(m, n, i)$  的值决定作业子集  $S_{mm}$  被选中起点并移动  $i$  个单位的概率。因同源作业子集间存在虚拟时序约束,因此每次移动时仅移动一个单位,  $i = \pm 1$ , 表示作业移动的方向与距离。

### 2.2.2 相对禁忌框架设计

#### (1) 相对禁忌表

使用相对禁忌表的目的是跳出局部最优,避免搜索选入循环。本研究中相对禁忌表记录的是记录

之前若干次移动所选取的起终点作业子集序号  $S_{mm}, S_{(m+i)(n+i)}$ 。

#### (2) 相对禁忌长度

禁忌对象为起终点作业子集序号,禁忌对象的总数相对较少。如图 4 所示,  $N=3$  时可选取的起终点对仅有 4 对,  $N=4$  时有 12 对,  $N=5$  时有 24 对。拆分数量对于可移动的起终点影响很大,因此禁忌长度  $L_{\text{Tabu}}$  的选取也很关键。本研究中取  $L_{\text{Tabu}} = N - 2$ 。

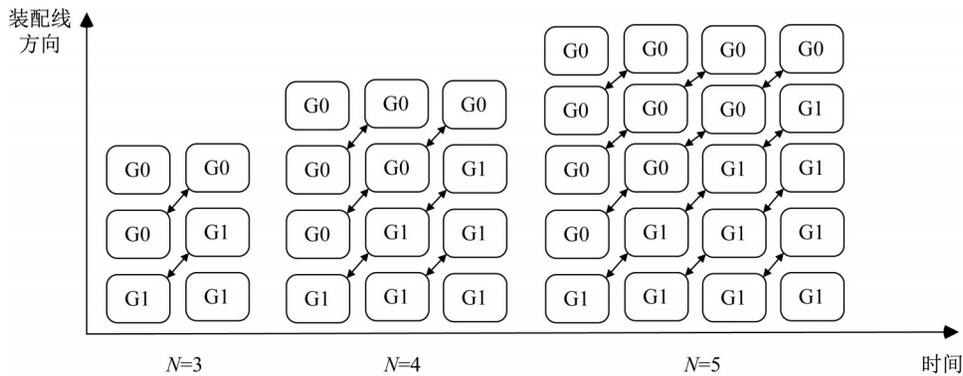


图 4 不同拆分数量下转换期

Fig.4 Transition process of different splitting numbers

#### (3) 特赦规则

对于完工时间最长的起点  $S_{mm}$ , 若其可移动的终点  $S_{(m+i)(n+i)}$  的完工时间为  $(m+i)$  周期内的最小值, 且  $m$  和  $(m+i)$  周期内的资源利用率分别为全周期内最高和最低时, 对  $S_{mm}, S_{(m+i)(n+i)}$  特赦。选取  $S_{mm}, S_{(m+i)(n+i)}$  为调整的起终点。若存在多个可特赦起终点项目对, 则随机选取一对起终点。

### 2.2.3 作业选取

**定义 4** 作业移动。对一个作业  $j \in S_{mm}$ , 将其从  $S_{mm}$  移动至  $S_{(m+1)(n+1)}$ , 称为作业右移; 将其从  $S_{mm}$  移动至  $S_{(m-1)(n-1)}$ , 称为作业左移。

**定义 5** 松弛约束。作业  $j$  与作业  $q$  在移动前存在真实时序约束, 在移动后真实时序约束转化为虚拟时序约束, 称为松弛约束。

**定义 6** 绑定约束。作业  $j$  与作业  $q$  在移动前存在虚拟时序约束, 在移动后虚拟时序约束转化为真实时序约束, 称为绑定约束。

作业移动过程如图 5 所示。当目标作业移动时, 目标作业与起点作业子集的其他作业松弛约束, 真实时序约束变少, 因此作业子集的关键链长度变短, 对应作业子集工期变短; 目标作业与终点作业子集的其他作业绑定约束, 真实时序约束增加, 因此作业子集的关键链长度变长, 对应作业子集工期

变长。

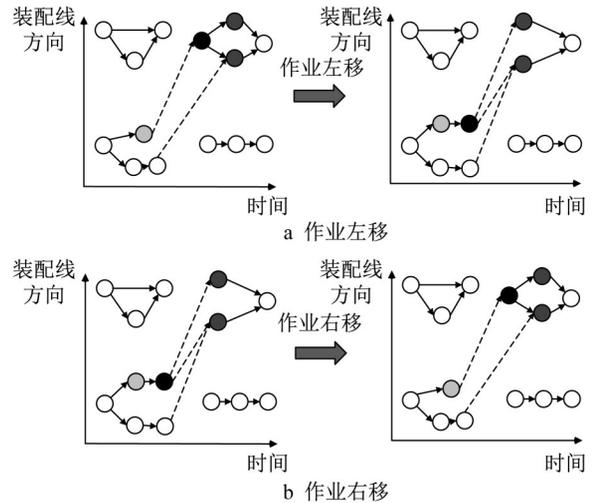


图 5 作业左(右)移分析

Fig.5 Analysis of job moving left (right)

作业移动的具体作业选取与时序约束的改变如下:

(1) 当作业左移时, 只有无真实紧前时序约束的作业  $j$  可以被选中, 参数  $\gamma_{1j}$  表示目标作业在移动前的真实紧后作业约束数量, 参数  $\gamma_{2j}$  表示目标作业在移动后的真实紧前作业约束数量,  $\Delta\gamma_j = \gamma_{1j} - \gamma_{2j}$ 。

(2) 当作业右移时, 只有无真实紧后时序约束的

作业 $j$ 可以被选中,参数 $\gamma_{1j}$ 表示目标作业在移动前的真实紧前作业约束数量,参数 $\gamma_{2j}$ 表示目标作业在移动后的真实紧后作业约束数量, $\Delta\gamma_j = \gamma_{1j} - \gamma_{2j}$ 。

此外,作业是在项目转换期内不同周期间进行作业移动,起点周期内调度作业减少,资源约束松弛,对应周期的整体工期变小;终点周期内的调度作业增多,资源约束收紧,对应周期的整体工期变大。因此,在选择了作业移动的起终点后,将结合真实时序约束与资源约束2个方面来选取作业进行移动。作业选取的权重参数 $\delta_j = \Delta\gamma_j/J + u_j$ ,其中 $\Delta\gamma_j/J$ 表示解绑的时序约束数量与总作业数的比值, $u_j = r_{j1}(1-f_{1m}) + \dots + r_{jk}(1-f_{km})$ 为资源评估函数,其具体含义为作业 $j$ 在周期 $m$ 内使用的资源与周期 $m$ 所使用的全部资源的比值, $f_{km} = \sum_{j=1}^{j \in S_{mm}} r_{jk}d_j/R_kT_m$ 为周期 $m$ 资源 $k$ 的使用率。 $\delta_j$ 将作业 $j$ 移动的真实时序约束变化与资源变化同时纳入考虑,默认选取 $\delta_j$ 最大的作业进行移动。当存在 $\delta_j$ 相同的作业时,添加补充规则:作业左移时,选取序号最小的作业;作业右移时,选取序号最大的作业。

#### 2.2.4 绝对禁忌规则

##### (1) 绝对禁忌表

绝对禁忌表的目的是筛选所有不可能成为最优解的可行解的特征,以此来减少解的搜索空间。本研究中,绝对禁忌表的禁忌对象为拆分方案的关键链。每次移动后,需计算移动后拆分方案的关键链,若关键链长度大于历史最优解 $C_{best}$ 时,将该关键链包含进绝对禁忌表中。绝对禁忌表有如下性质:

**性质2** 包含绝对禁忌表中关键链的拆分方案均不是最优解。

**证明** 绝对禁忌表中的关键链所对应的解不是唯一的,所有包括这条关键链的项目拆分方案,其完工时间均大于关键链长度。又因为关键链的时长大于历史最优解 $C_{best}$ ,因此拆分方案完工时间 $C > C_{best}$ 无法成为历史最优解。

因为绝对禁忌表的性质2,绝对禁忌表不受限于禁忌长度,所以绝对禁忌表不会解除禁忌。

##### (2) 绝对禁忌规则

当作业移动后,若移动后产生的新拆分方案中包含绝对禁忌表中的关键链,则新方案不被接受,返回作业选取步骤顺次按权重 $\delta_j$ 选取下一个可移动作业。如果所有移动方案均在绝对禁忌表中,就跳过本次移动,选取新的项目调整起终点。

## 3 数值实验

### 3.1 实例验证

为了验证算法和模型的可行性,选用PSPLIB提供的算例作为某工厂所装配的飞机项目G0、G1,使用Python3.7.0编程实现算法。数据测试平台选用i5-4210U处理器,2.39 GHz主频,8 G内存。项目G0、G1各包括30个作业及2个虚拟作业,4种完成项目所需的可更新资源,具体数据如表2所示。某工厂需要将飞机移动生产线从G0转换至G1,计算转换的总工期并使得总时间最小化。采用所提出的模型及算法,对于G0、G1确定的拆分数量 $N=3$ ,算法参数设定 $\alpha=1, \epsilon=1, L_{Tabu}=1, E=100$ 。

为了进一步验证本研究模型和算法的可行性,以同样的转换过程,与不进行重拆分的调度方案初始解进行对比。给定各类资源上限均为15,初始解的具体拆分方案如表3所示,本研究算法优化的重拆分方案如表4所示,2种拆分方案的对比结果如表5所示。可以看出,采用本研究方案的项目时长显著小于不经过调整的方案。在实际生产中,使用本研究方案在转换过程中进行变节拍生产,可以提高生产线的生产效率。

### 3.2 算法分析

选取参数 $\epsilon=1$ 以确保 $\psi(m, n, i)$ 不为零,初始拆分方案依文献[10]进行计算,因为初始拆分方案已经是项目的最优解,所以取迭代次数 $E=100, N=5, L_{Tabu}=3$ 。对控制移动方向的随机参数 $\alpha$ 进行敏感性分析,选取标准算例库中的3种作业规模J30、J60、J90的各40个项目进实验。3种算例下,比较差值百分比的均值 $\overline{GAP}$ 。图6是参数 $\alpha$ 的敏感性分析。可以看出,当 $\alpha=1$ 时, $\overline{GAP}$ 值最小,所以实验时取 $\alpha=1$ 。

### 3.3 算法比较

由图4分析可知,计算规模随着拆分数量的增加呈幂函数增长,常规的精确算法难以求解,故选取已有的项目拆分问题启发式算法<sup>[10]</sup>结合项目节拍转换过程,将遗传算法与本研究算法进行对比。引用标准算例库PSPLIB中的3种作业规模J30、J60、J90,在 $N=3, 4, 5$ 的情况下,分别随机选取5对项目进行项目转换,参数设定 $\alpha=1, E=100, L_{Tabu}=N-2$ 。

表6~8列出了3种规模下3种拆分数目结果的对比。 $GAP_{GA}$ 与 $GAP_{BASE}$ 的计算式分别如下所示:

表 2 项目数据  
Tab.2 Project data

G0项目信息							G1项目信息						
作业	持续时间	紧后作业	资源类型				作业	持续时间	紧后作业	资源类型			
			一	二	三	四				一	二	三	四
1	0	2,3,4	0	0	0	0	1	0	2,3,4	0	0	0	0
2	6	11,12,15	0	0	7	0	2	3	10,31	0	0	9	0
3	4	5,8,16	0	0	1	0	3	9	5,14,22	0	7	0	0
4	2	27	0	0	0	2	4	4	6,7	9	0	0	2
5	1	6,12,14	0	0	6	0	5	10	8,15	10	3	3	10
6	5	7	2	0	0	0	6	9	9,19	0	0	0	4
7	1	9,28	0	0	0	6	7	2	31	2	6	1	0
8	3	10,20	0	9	0	0	8	4	11,16,17	8	0	8	4
9	5	12,21	0	8	0	0	9	4	12	0	0	4	0
10	6	17,25	8	0	0	0	10	10	13,19	0	0	8	0
11	6	25,28	0	0	7	0	11	7	13	10	0	0	0
12	4	25,29	7	0	0	0	12	7	14,26	0	2	5	6
13	1	22,23,31	8	0	0	0	13	1	29,30	3	0	0	0
14	6	17	0	1	0	0	14	6	28	0	10	8	0
15	5	24	0	0	0	2	15	7	18	0	1	5	1
16	3	18	0	0	5	0	16	1	20	0	1	0	5
17	3	26	0	2	0	0	17	1	19,21	3	0	0	0
18	2	19,23	7	0	0	0	18	4	31	10	6	1	0
19	3	22	0	8	0	0	19	6	29	9	2	5	9
20	3	29	0	0	5	0	20	6	30	0	6	0	3
21	7	31	0	0	0	2	21	5	23	6	0	4	0
22	4	24	0	0	0	1	22	6	24,27	5	0	0	0
23	8	24	0	0	0	5	23	6	24,26	0	6	0	8
24	1	29	0	0	0	5	24	10	25	0	8	0	5
25	8	30	0	9	0	0	25	4	28,29	0	4	0	9
26	8	28	0	5	0	0	26	7	27	9	0	5	0
27	5	30	0	9	0	0	27	3	28	0	9	0	5
28	1	30	0	4	0	0	28	6	30	2	8	3	8
29	5	32	0	4	0	0	29	9	32	0	0	0	7
30	2	32	0	9	0	0	30	1	32	0	0	0	4
31	1	32	0	0	0	10	31	10	32	5	0	0	7
32	0		0	0	0	0	32	0		0	0	0	0

表 3 初始拆分方案

Tab.3 Initial splitting plan

作业子集序号	G0拆分方案	G1拆分方案
一	1,2,3,4,5,6,7,8,10,11,14,16,18	1,2,3,4,5,8,11
二	9,12,13,15,17,19,20,23,25,26,27,28,30	6,7,9,10,12,13,14,15,16,17,18,20,21,22
三	21,22,24,29,31,32	19,23,24,25,26,27,28,29,30,31,32

表 4 转换期拆分方案

Tab.4 Transition splitting plan

装配单元序号	第一周期		第二周期	
	项目类型	作业序号	项目类型	作业序号
一	G1	1,2,3,5,6,7,8,9,11,14,16	G1	1,2,3,4,5,8,11
二	G0	9,12,13,15,17,21,22,23,31	G1	4,10,12,13,15,17,18,20,21,22
三	G0	21,22,24,29,31,32	G0	19,20,24,25,26,27,28,29,30,32

表 5 转换期节拍时间对比

Tab.5 Cycle time comparison during transition period

装配方案	节拍时间	转换期时长
转换期拆分方案	34/30	64
初始拆分方案	39/43	82

$$GAP_{GA} = \frac{C_{GA} - C_{DTS}}{C_{DTS}}$$

$$GAP_{BASE} = \frac{C_{BASE} - C_{DTS}}{C_{DTS}}$$

式中:C为转换期总时长;GA表示遗传算法,BASE

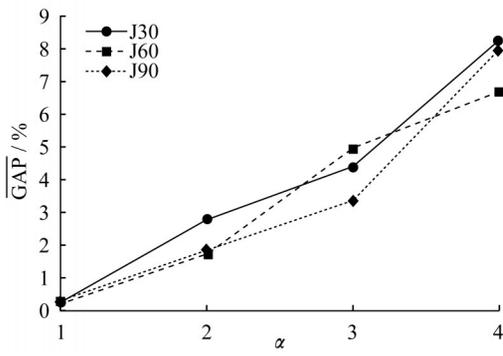


图6  $\alpha$ 对调度结果的影响

Fig.6 Effect of  $\alpha$  on scheduling results

表示初始拆分方案,DTS表示本研究方案。表6~8中, $t$ 为求解时间。对于GA,取迭代次数100,初始种群数10,交叉概率0.8,变异概率0.2,对作业所属项目进行编码。

通过表6~8可以得出,在求解小中大3种规模的算例时,本研究算法平均质量均优于对比算法GA,但对于部分项目,如J30、 $N=4$ 、第4组项目时,本研究算法的求解效果不如对比算法。与对比算法差别较小的原因可能是两者均在初始拆分方案的基础上进行进一步搜索,因此搜索的方向相似。从

表6 J30规模数值结果

Tab.6 Numerical results of J30

规模	拆分数	组别	$C_{DTS}$	$C_{GA}$	$C_{BASE}$	$t_{BASE}/s$	$t_{GA}/s$	$GAP_{GA}/\%$	$GAP_{BASE}/\%$
J30	N=3	1	77.0	79.0	95.0	26.34	16.04	2.6	23.4
		2	96.0	98.0	114.0	33.73	32.16	2.1	18.8
		3	108.0	110.0	119.0	17.90	18.88	1.9	10.2
		4	92.0	90.0	102.0	27.29	18.34	-2.2	10.9
		5	109.0	109.0	121.0	37.88	37.31	0.0	11.0
		平均值	96.4	97.2	110.2	28.63	24.55	0.9	14.8
	N=4	1	101.0	102.0	104.0	26.59	21.16	1.0	3.0
		2	125.0	125.0	128.0	35.94	41.09	0.0	2.4
		3	161.0	162.0	175.0	35.68	36.12	0.6	8.7
		4	135.0	132.0	144.0	30.14	22.13	-2.2	6.7
		5	142.0	143.0	161.0	34.67	34.48	0.7	13.4
		平均值	132.8	132.8	142.4	32.60	31.00	0.0	6.8
	N=5	1	130.0	132.0	137.0	43.33	42.34	1.5	5.4
		2	183.0	186.0	199.0	48.95	47.70	1.6	8.7
		3	199.0	203.0	224.0	40.70	40.72	2.0	12.6
4		170.0	171.0	176.0	43.09	40.70	0.6	3.5	
5		185.0	186.0	189.0	44.37	38.03	0.5	2.2	
	平均值	173.4	175.6	185.0	44.09	41.90	1.3	6.5	

表7 J60规模数值结果

Tab.7 Numerical results of J60

规模	拆分数	组别	$C_{DTS}$	$C_{GA}$	$C_{BASE}$	$t_{BASE}/s$	$t_{GA}/s$	$GAP_{GA}/\%$	$GAP_{BASE}/\%$
J60	N=3	1	115.0	118.0	128.0	104.43	113.38	2.6	6.7
		2	110.0	113.0	127.0	94.47	104.47	2.7	15.5
		3	130.0	129.0	143.0	142.46	169.18	-0.8	10.0
		4	129.0	127.0	146.0	131.16	138.92	-1.6	13.2
		5	140.0	140.0	158.0	147.61	150.72	0.0	12.9
		平均值	124.8	125.4	140.4	124.03	135.33	0.6	12.6
	N=4	1	158.0	162.0	181.0	171.16	178.92	2.5	14.6
		2	158.0	159.0	176.0	210.93	215.62	0.6	11.4
		3	140.0	143.0	158.0	141.11	153.23	2.1	12.9
		4	206.0	203.0	228.0	173.05	172.20	-1.5	10.7
		5	170.0	174.0	196.0	155.11	147.61	2.4	15.3
		平均值	166.4	168.2	187.8	170.27	173.51	1.2	13.0
	N=5	1	241.0	240.0	257.0	182.27	194.56	-0.4	6.6
		2	190.0	192.0	211.0	318.96	295.06	1.1	11.1
		3	203.0	206.0	224.0	336.94	373.64	1.5	10.3
4		282.0	291.0	310.0	240.91	244.76	3.2	9.9	
5		216.0	214.0	246.0	149.76	167.08	-0.9	13.9	
	平均值	226.4	228.6	249.6	245.77	277.01	0.9	10.4	

表 8 J90 规模数值结果  
Tab.8 Numerical results of J90

规模	拆分数	组别	$C_{DTS}$	$C_{GA}$	$C_{BASE}$	$t_{BASE}/s$	$t_{GA}/s$	$GAP_{GA}/\%$	$GAP_{BASE}/\%$
J90	N=3	1	167.0	165.0	178.0	277.30	318.42	-1.2	6.6
		2	119.0	123.0	141.0	281.14	323.94	3.4	18.5
		3	176.0	180.0	183.0	221.10	286.85	2.3	4.0
		4	181.0	180.0	183.0	258.41	293.92	-0.6	1.1
		5	116.0	118.0	126.0	279.63	334.77	1.7	8.6
		平均值	151.8	153.2	162.2	263.51	311.58	1.1	7.8
	N=4	1	171.0	169.0	186.0	292.21	308.02	-1.2	8.8
		2	215.0	214.0	221.0	539.95	592.14	-0.5	2.8
		3	174.0	177.0	184.0	276.43	283.23	1.7	5.7
		4	218.0	221.0	232.0	312.68	328.12	1.4	6.4
		5	245.0	247.0	275.0	364.12	361.91	0.8	12.2
		平均值	204.6	205.6	219.6	357.06	374.68	0.5	7.2
	N=5	1	204.0	207.0	234.0	749.72	1011.83	1.5	14.7
		2	248.0	252.0	273.0	546.91	649.41	1.6	10.1
		3	269.0	268.0	291.0	523.56	538.41	-0.4	8.2
4		279.0	281.0	290.0	804.11	885.42	0.7	3.9	
5		318.0	320.0	330.0	538.79	591.23	0.6	3.8	
平均值		263.6	265.6	283.6	632.61	735.25	0.8	8.1	

整体上看,本研究算法与对比算法之间的GAP稳定在3%以内,并且平均求解质量优于GA约1%,这是由于双重禁忌搜索限制了搜索空间,提高了搜索质量。从与初始拆分方案的GAP值上来看,本研究算法对于初始拆分方案的优化程度平均在10%左右,但是不同的项目间波动较大,最高值可达23.4%,最低值仅有1.1%,这是由于本研究算法是在已有的项目拆分方案上进行进一步优化,可优化区间主要来自于转换的两项目间的差异。对于相似的2个项目,初始拆分方案是类似的,在转换期间对资源的需求和节拍时间的要求都相似,因此优化的空间有限。对于差异性较大的项目,本研究算法的优化空间更大。在不同规模的算例下,拆分数并未对优化结果产生显著的影响,但是从约束角度分析,随着拆分数增大,被松弛的真实时序约束增多,作业调度的自由度就更高。当拆分数与作业数相等时,该问题将退化为一个没有时序约束的项目调度问题。因此,随着拆分数的增大,每个周期的节拍时间有减小的趋势。在算法的运算速度方面,随着项目规模和拆分数的增加,本研究算法和GA的求解时间都在增大。在项目规模较小时,GA的求解时间略小于本研究算法,但是当项目的规模数达到J60和J90时,本研究算法的求解时间显著小于对比算法,这说明本研究算法对大规模项目的适应性较好,而对于小规模算法求解速度较差。

## 4 结语

以RCPTTSP-PS为研究对象,建立了离散时间下的数学模型,提出了双重禁忌搜索算法,并结合双层迭代算法对问题进行求解。在项目转换阶段,考虑了项目移动对起终点项目的共同影响,提出了起终点选取函数以及作业选取权重函数。同时,为了限制搜索空间,提出了相对禁忌规则和绝对禁忌规则,有助于避免算法陷入局部最优,并利用绝对禁忌规则限制搜索空间,提高了搜索速度。在数值实验中,本研究算法的求解质量略优于对比算法,与对比算法的GAP值在1%左右。在求解速度方面,本研究算法在大规模项目的求解速度明显优于对比算法。因此,本研究算法对于飞机脉动生产线这类大规模项目调度问题具有良好的效果。同时,相较于初始拆分方案,本研究算法对于转换期时长的优化程度在1.1%~23.4%之间,与初始拆分方案的GAP值方差较大,这是由不同项目网络之间的差异引起的。在未来的研究中,可以对RCPTTSP-PS中不同项目的关联性进行进一步研究,有助于提高生产线节拍转换过程的时间优化,对于生产线上的实际应用也有一定帮助。

### 作者贡献声明:

陆志强:提出研究选题,设计研究思路和论文框架。

王浩宇:设计研究思路,完成实验设计并实施,分析数据,撰写论文。

## 参考文献:

- [1] 潘志豪,郭宇,查珊珊,等.基于混合优化算法的飞机总装脉动生产线平衡问题[J].计算机集成制造系统,2018,24(10):2436.  
PAN Zhihao, GUO Yu, ZHA Shanshan, *et al.* Aircraft pulsating assembly line balancing problem based on hybrid algorithm [J]. Computer Integrated Manufacturing Systems, 2018,24(10):2436.
- [2] GONCALVES J F, MENDES J J, RESENDE M G. A genetic algorithm for the resource constrained multi-project scheduling problem [J]. European Journal of Operational Research, 2008,189(3):1171.
- [3] KURTULUS I, DAVIS E W. Multi-project scheduling: categorization of heuristic rules performance [J]. Management Science, 1982, 28(2):161.
- [4] BROWNING T R, YASSINE A A. Resource-constrained multi-project scheduling: priority rule performance revisited[J]. International Journal of Production Economics, 2010, 126(2):212.
- [5] LENSTRA J K, KAN RINNOOY A H G. Complexity of scheduling under precedence constraints [J]. Operations Research,1978,26(1):22.
- [6] WANG Yanting, HE Zhengwen, KERKHOVE L P, *et al.* On the performance of priority rules for the stochastic resource constrained multi-project scheduling problem [J]. Computers & Industrial Engineering, 2017, 114:223.
- [7] KURTULUS I. Multiproject scheduling: analysis of scheduling strategies under unequal delay penalties [J]. Journal of Operations Management, 1985, 5(3): 291.
- [8] KIM K W, YUN Y S, YOON J M, *et al.* Hybrid genetic algorithm with adaptive abilities for resource-constrained multiple project scheduling [J]. Computers in Industry, 2005, 56(2):143.
- [9] 陈俊杰,同淑荣,叶正梗,等.资源受限多项目调度问题的两阶段算法[J].控制与决策,2020,35(8):224.  
CHEN Junjie, TONG Shurong, YE Zhengeng, *et al.* Two-stage algorithm for resource-constrained multi-project scheduling problem [J]. Control and Decision, 2020, 35(8):224.
- [10] 陈浩杰,丁国富,张剑,等.求解资源受限多项目调度的改进遗传规划算法[J].中国机械工程,2021,32(10):1213.  
CHEN Haojie, DING Guofu, ZHANG Jian, *et al.* Improved genetic programming algorithm for resource constrained multi-project scheduling problem [J]. China Mechanical Engineering, 2021,32(10):1213.
- [11] 陆志强,杨超.基于项目网络拆分决策的多项目协同调度问题建模[J].上海交通大学学报,2017,51(2):193.  
LU Zhiqiang, YANG Chao. Modeling of resource constrained multi-project scheduling problem based on project splitting [J]. Journal of Shanghai Jiaotong University, 2017,51(2):193.
- [12] 朱宏伟,陆志强.考虑资源转移时间的项目可拆分资源受限多项目调度问题[J].计算机集成制造系统,2019,25(3):62.  
ZHU Hongwei, LU Zhiqiang. Modeling and improved algorithm for resource constrained multi-project scheduling problem based on project splitting with resource transfer time [J]. Computer Integrated Manufacturing Systems, 2019, 25(3):62.
- [13] 宗保氏,陆志强.项目拆分与资源投入调度问题的集成优化[J].上海交通大学学报,2018,52(7):793.  
ZONG Baoshi, LU Zhiqiang. Integrated optimization of project splitting and resource investment project scheduling [J]. Journal of Shanghai Jiaotong University, 2018,52(7):793.
- [14] CHEN Haojie, DING Guofu, ZHANG Jian, *et al.* Research on priority rules for the stochastic resource constrained multi-project scheduling problem with new project arrival [J]. Computers & Industrial Engineering, 2019, 137:106060.
- [15] YASSINE A A, MOSTAFA O, BROWNING T R. Scheduling multiple, resource-constrained, iterative, product development projects with genetic algorithms [J]. Computers & Industrial Engineering, 2017, 107:39.
- [16] ABRANTES R, FIGUEIREDO J. Resource management process framework for dynamic NPD portfolios [J]. International Journal of Project Management, 2015, 33(6): 1274.