

# 带资源空窗期的资源投入型问题的建模与优化

陆志强, 周皓雪

(同济大学 机械与能源工程学院, 上海 201804)

**摘要:** 以飞机移动式装配线为背景, 在基本资源投入型问题的基础上考虑资源空窗期约束, 建立以最小化资源使用总成本为目标的数学模型. 针对该模型设计了一种构造启发式算法, 并提出了非关键任务优先级决策规则. 考虑空窗期约束特点, 以连续排入的两个非关键任务间结果最优的启发式规则来确定非关键任务位置, 并提出以非关键任务优先级和关键任务开始时间为双链表编码的遗传算法, 然后将启发式规则嵌套在遗传算法的解码和评估阶段. 最后通过数值实验比较启发式算法和遗传算法与 CPLEX 在求解该问题时的优劣, 证明了两种算法的有效性.

**关键词:** 资源投入; 资源空窗期; 启发式算法; 遗传算法

**中图分类号:** F273

**文献标志码:** A

## Modeling and Optimization of Resource Investment Problem with Resource Window

LU Zhiqiang, ZHOU Haoxue

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

**Abstract:** Resource investment problem with resource window constraint was considered in the context of aircraft mobile assembly line. A mathematical model was proposed to solve the problem with the objective of the total cost minimization of resource. Firstly, based on the characteristics of the resource window constraint, a constructive heuristic algorithm with non-critical activity priority decision rules was developed to solve small-scale problems. Secondly, a genetic algorithm, which was coded by a double-linked list including non-critical activity priority and critical activity start time and decoded by non-critical activity priority decision rules, was proposed for the large-scale problems. Finally, numerical experiments were carried out to compare the advantages and disadvantages between heuristic algorithm, genetic algorithm and CPLEX, and the effectiveness of the two proposed algorithms was proved.

**Key words:** resource investment; resource window; heuristic algorithm; genetic algorithm

基于精益思想的飞机移动式装配线已成为飞机装配生产的新模式, 具有装配效率高、质量稳定、可按需批量生产等特点, 近年来被世界各大航空制造企业竞相采用. 实际上, 对于每架飞机而言, 移动式装配线上的总装任务调度问题具有装配工序繁多、装配任务复杂、装配过程受到多重约束限制等特点. 调度过程可根据实际应用的目的抽象为资源受限项目调度问题(RCPSP)以及与之关联的资源投入型问题(RIP)两大类. 前者与装配线的规划、设计和决策相关, 而后者与装配线的运作决策关联. 对装配线的日常运作决策而言, 通常需要在给定工期的条件下, 优化装配线的各类资源配置及分配使用, 达到最小化资源投入总成本的目的, 即资源投入型问题. 对于一般装配线来说, 装配线复杂度较低, 工作可替代性强. 飞机移动式装配线的工序繁琐复杂、工作专业性强, 对于特殊人力资源要求极高. 同时, 整个装配周期较长, 关键技术人员稀少并且可能存在不可用时间段, 对于飞机装配生产线考虑资源利用区间是重要且必要的. 因此, 在基本 RIP 的基础上, 考虑到飞机装配生产中某些关键资源具有提前已知的不可用时间段, 即引入资源空窗期约束, 提出带资源空窗期的资源投入型问题.

资源投入型问题作为 RCPSP 的对偶问题, 最早由 Möhring<sup>[1]</sup>引入, 并且该问题被证明为 NP-hard 问题. 以 RCPSP 的算法求解为基础, 得到了求解该问题的图解精确算法. Drexel 等<sup>[2]</sup>基于拉格朗日松弛和列生成方法为 RIP 提出了两个下界算法, 并且通过算例实验与 Möhring<sup>[1]</sup>的算法进行了比较. Rodrigues 等<sup>[3]</sup>提出了一种改进型割平面法, 通过启

收稿日期: 2018-12-12

基金项目: 国家自然科学基金(61473211, 71171130)

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

发式算法得到初始解并不断更新低界,缩小解空间,从而提高计算效率.由于精确算法只能求解小规模问题,因此大量国内外学者针对大规模问题提出启发式算法与元启发式算法. Song 等<sup>[4]</sup>提出带局部搜索的进化算法,允许项目延期但是设有极高的延期惩罚. Myszkowski 等<sup>[5]</sup>在混合蚁群算法中嵌入启发式优先级规则对资源进行分配. Van Peteghem 等<sup>[6]</sup>提出了一种应用于 RIP 的人工免疫算法. He 等<sup>[7]</sup>设计了基于动态优先级规则的启发式算法,并对任务开始时间决策区间内的所有位置进行评测. Qi 等<sup>[8]</sup>提出了一种新的粒子群算法,在解码阶段设计了多种启发式规则以修复不可行解. 由于现有考虑资源特点的 RIP 相关研究稀缺,因此带资源空窗期的 RCPSP 的研究方法对本研究具有参考意义. 胡淑芳<sup>[9]</sup>针对资源时间窗和任务可拆分特性,设计了小规模单技能及多技能项目调度问题求解的分支定界算法. 廖广瑞等<sup>[10]</sup>考虑资源的多技能和时间窗属性,设计了基于优先规则的 Rollout 求解算法,并嵌入了启发式资源分配方法对资源进行快速分配. 蔡方中等<sup>[11]</sup>研究了基于时间窗的多项目资源分配问题,提出了带时间窗参数以及惩罚因子的多项目管理资源分配模型.

从现有文献可以看出,对于 RIP 的研究范围仍偏重于基本问题,而基本 RIP 通常设有大量假设条件,属于较为理想化的纯理论问题,因此 RIP 的研究成果难以与更为复杂的实际生产过程有效结合. 此外,从算法设计上来看,大部分启发式算法在任务开始时间的决策区间内取值时,计算效率不高,并且易限于局部考虑而忽略全局影响. 常见的搜索算法可分为以下两类:一类以资源量进行编码,但这样编码容易导致资源上、下界差距较大,搜索效率低,难以找到结果较好的可行解;一类以任务开始时间进行编码,但这类编码大多数未考虑问题特点,搜索的方向性不强,导致算法效率低. 基于此,在进行算法设计时重点考虑资源空窗期约束,首先提出求解 RIP 的启发式算法. 区别于基本 RIP,设计了非关键任务优先级、关键任务排入以及非关键任务调度的三阶段启发式算法,对模型进行求解. 通过进一步的分析发现,启发式算法处理大规模算例时具有局限性,因此在启发式算法的基础上设计了以非关键任务优先级和关键任务开始时间为双链表编码的遗传算法,并将启发式规则嵌套在遗传算法的解码和评估阶段,提高了求解的精确性. 最后,通过数值实验验证了启发式算法和遗传算法的有效性,同时证实了空

窗期约束的存在对于算法设计具有影响.

## 1 问题描述及数学模型

### 1.1 问题描述

假设飞机装配项目由若干项任务构成,任务之间存在着一定的时序关系,任务的执行需要消耗相应的资源. 记给定项目总工期为  $T$ , 包含  $J$  项任务, 每项任务  $j(j=1, 2, \dots, J)$  的执行时间为  $t_j$ ,  $p(j)$  为第  $j$  项任务的紧前任务集合,  $s(j)$  为第  $j$  项任务的紧后任务集合,  $t_{ES,j}$ ,  $t_{LS,j}$ ,  $t_{EF,j}$  和  $t_{LF,j}$  分别为任务  $j(j=1, 2, \dots, J)$  的最早开始时间、最晚开始时间、最早结束时间以及最晚结束时间,  $t_{TF,j}$  表示任务  $j$  开始时间的浮动长度, 即  $t_{TF,j} = t_{LS,j} - t_{ES,j}$ . 项目中使用的可更新资源种类为  $P$ , 每种资源的单位使用成本为  $C_p$  ( $p=1, 2, \dots, P$ ).  $r_{jp}$  表示任务  $j(j=1, 2, \dots, J)$  需要的单位资源  $p(p=1, 2, \dots, P)$  的数量,  $r_{j,\max}$  表示任务  $j$  需要的所有资源中需求量最大的资源值. 关键资源  $p_k$  在特定的一段或多段时间区域内不能工作, 即资源的空窗期约束.  $A_e$  代表需要用到关键资源  $p_k$  的任务集合. 对时间进行离散化处理, 模型的求解目标是优化各时刻  $t(t=1, 2, \dots, T)$  每种资源  $p(p=1, 2, \dots, P)$  的最大需求量  $R_p$  的总成本.

模型中决策变量包括:  $x_{jt}$  为 0、1 变量, 1 表示任务  $j(j=1, 2, \dots, J)$  在时刻  $t$  被执行, 否则取 0;  $z_{jm}$  为 0、1 变量, 1 表示任务  $j \in A_e$  在第  $m$  个非空窗期区间内完成, 否则取 0.

### 1.2 空窗期

在飞机装配的实际生产中, 关键资源的使用时间往往不是连续的, 某些时段可用, 某些时段不可用, 将这种约束称为资源空窗期约束. 参考刘振元等<sup>[12]</sup>相关论文, 对空窗期相关参数进行定义.

如图 1 所示, 假设有  $M$  个空窗期,  $m=1, 2, 3, \dots, M$ ,  $t_{b,m}$  和  $t_{e,m}$  分别为第  $m$  个空窗期的开始和结束时间. 在空窗期内资源可用数量为零, 在非空窗期内资源可用数量为常数  $R_0$ .  $l_m$  表示第  $m$  个空窗期的长

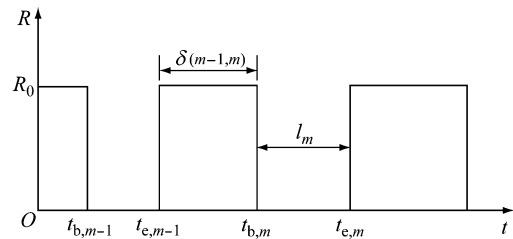


图 1 资源空窗期示意图

Fig. 1 Schematic diagram of resource window

度,易得 $l_m = t_{e,m} - t_{b,m}$ ;  $\delta_{(m-1,m)}$ 表示第 $m$ 个非空窗期的长度,易得 $\delta_{(m-1,m)} = t_{b,m} - t_{e,m-1}$ . 假设 $t_{b,0} = t_{e,0} = 0$ .

### 1.3 数学模型

目标函数为

$$\min \sum_{p=1}^P (C_p \max(R_{ip})) \quad (1)$$

约束为

$$R_{ip} = \sum_{j=1}^J r_{jp} x_{jt}, \quad t = 1, 2, \dots, T, p = 1, 2, \dots, P \quad (2)$$

$$\sum_{t=t_{ES,j}}^{t_{LF,j}} x_{jt} = t_j, \quad j = 1, 2, \dots, J \quad (3)$$

$$t_j x_{jt} \leq \sum_{k=1}^{t-1} x_{j^*k}, \quad \forall j^* \in p(j) \quad (4)$$

$$\sum_{m=1}^M z_{jm} t_{e,m-1} \leq \sum_{t=t_{EF,j}}^{t_{LF,j}} t x_{jt} \leq \sum_{m=1}^M z_{jm} t_{b,m}, \quad j \in A_e \quad (5)$$

$$t_j x_{jt} - t_j x_{j(t+1)} + \sum_{k=t+2}^T x_{jk} \leq t_j, \quad j = 1, 2, \dots, J, \quad t = 1, 2, \dots, T \quad (6)$$

$$t x_{jt} \leq T, \quad t = 1, 2, \dots, T \quad (7)$$

目标函数(1)表示该问题是优化整个周期内各资源使用的最大总成本. 约束条件(2)表明时刻 $t$ 资源 $p$ 的使用量;式(3)表示任务在整个周期内的执行时间满足该任务工期约束;任务之间应该满足约束(4),即满足一定的优先关系;式(5)说明需要关键资源的任务必须满足空窗期约束;式(6)说明任务一旦开始,在完成之前不允许被中断;式(7)说明所有任务的最晚结束时间不应大于项目给定工期.

## 2 启发式算法设计

启发式算法的设计思路为:基于关键路径法(CPM)获得项目中的关键链,确定关键任务和非关键任务. 将关键任务纳入调度计划,决策关键任务排入方式为连续排入或预留时间间隔排入;在此基础上安排非关键任务,提出四种规则确定非关键任务优先级,然后以最小资源需求量为目标,确定非关键任务的决策区间,再通过局部规则判断基准在决策区间内的取值,最终确定非关键任务调度位置. 通过以上分析可知,该启发式算法可根据决策重点分为三个阶段.

### 2.1 第一阶段:生成非关键任务优先级列表

吴怡薇等<sup>[13]</sup>在求解资源水平问题时提出采用资源用量规则和自由度规则来确定非关键任务优先级. 本研究在基本 RIP 上考虑空窗期约束,为了减少

空窗期对任务的交互影响,增加工期规则以及最早开始时间(EST)规则,使可能对调度结果产生更大影响的任务拥有更多可排空间,尽早排入.

#### (1) 工期规则

任务工期越长,与已排定任务重叠度越高,在有空窗期的条件下,与空窗期重叠的可能性也越大. RIP 目标的本质就是通过调度使得任务间重叠度降低,所以 $t_j$ 越长,非关键任务 $j$ 的优先级越高.

#### (2) EST 规则

一般而言,在 RIP 中任务调度顺序与各任务在项目网络图中位置不完全一致. EST 规则可作为前三种规则的补充,避免优先级相同的情况出现,因此可定义 $t_{ES,j}$ 越小,非关键任务 $j$ 的优先级越高.

任务的工期对开始时间决策区间的大小影响很大,而任务的资源用量直接影响资源投入总成本,故工期规则及资源用量规则作为优先规则考虑. 具体操作时,按照上述四种规则顺序依次比较非关键任务 $j$ 的 $t_j$ 、 $r_{j,\max}$ 、 $t_{TF,j}$ 以及 $t_{ES,j}$ ,最终确定非关键任务优先级列表 $L_p$ .

第一阶段算法步骤如下所示:

**步骤 1** 建立已安排任务集合 $S = \emptyset$ ,未安排任务集合 $S' = J$ .

**步骤 2** 确定需要关键资源 $p_k$ 的任务集合 $A_e$ .

**步骤 3** 计算每种资源的最大需求量的总和

$$r_{\max}, r_{\max} = \sum_{p=1}^P \max r_{jp}.$$

**步骤 4** 根据 CPM 获得任务 $j \in J$ 的 $t_{ES,j}$ 、 $t_{LS,j}$ 、 $t_{EF,j}$ 、 $t_{LF,j}$ 和 $t_{TF,j}$ .

**步骤 5** 确定关键任务为集合 $C_r$ ,余下任务即为非关键任务.

**步骤 6** 生成非关键任务优先级列表 $L_p$ .

### 2.2 第二阶段:关键任务排入方式的确定

#### 2.2.1 相关关键任务 $j_0^*$

由第 2.1 节获得非关键任务优先级列表 $L_p$ ,设列表 $L_p$ 中第 1 个非关键任务为 $j_0$ ,令 $\Delta = t_{ES,j_0^*} - t_{ES,j_0}$ ,称使得 $\Delta$ 取 0 或最小正数的 $j_0^*$ 为相关关键任务. 如图 2 所示,任务 2、5、7、8、9 为关键任务,任务 1 为列表 $L_p$ 中第 1 个非关键任务,任务 7 为相关关键任务.

#### 2.2.2 关键任务调整基准

调整关键链时,关键链中相关关键任务 $j_0^*$ 之前的关键任务不做调整, $j_0^*$ 按调整区间点集进行位置的改变,而 $j_0^*$ 之后的关键任务将随着 $j_0^*$ 的调整顺次右移,最终确定所有关键任务的位置. 调整相关关

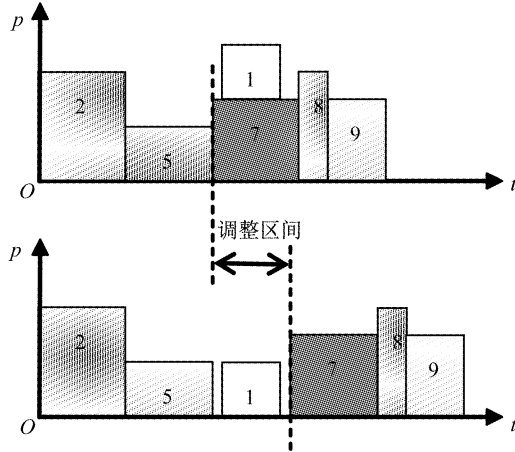


图 2 相关关键任务调整

Fig. 2 Related critical activity adjustment

键任务  $j_0^*$  的位置可使优先级最高的非关键任务拥有更多的可排空间,可最大限度降低资源用量峰值。

第二阶段算法步骤如下所示:

**步骤 1** 将集合  $C_r$  内的所有关键任务连续排入。

**步骤 2** 选择列表  $L_p$  中第 1 个非关键任务,记为  $j_0$ ;根据  $\Delta = t_{ES,j^*} - t_{ES,j_0}$  计算得到相关关键任务  $j_0^*$ 。

**步骤 3** 将  $j_0^*$  之前的每个关键任务按最早开始时间排入,更新集合  $S, S'$ 。

**步骤 4**  $j_0^*$  的最早开始时间记为  $t_{ES,j_0^*}$ ,令  $t = t_{ES,j_0^*}$ 。

**步骤 5**  $j_0^*$  的开始时间为  $t$ 。

**步骤 6** 将  $j_0^*$  之后的关键任务按照最早开始时间规则排入,若在任务工期约束和空窗期约束下有解,则将  $t$  记入  $j_0^*$  的开始时间  $t_{ST,j_0^*}$  的调整区间点集  $S_{change,j_0^*}$ ;否则,转入步骤 8。

**步骤 7**  $t = t + 1$ ;返回步骤 5。

**步骤 8** 生成  $j_0^*$  的开始时间  $t_{ST,j_0^*}$  的调整区间点集  $S_{change,j_0^*}$ 。

## 2.3 第三阶段:非关键任务的调度决策

### 2.3.1 相关任务与相关距离 $d_{ij}$

能够影响任务  $j$  的开始时间  $t_{ST,j}$  的决策区间的任务为和任务  $j$  具有时序关系的任务,称该任务为任务  $j$  的相关任务。定义  $R_e = \{i \in S | i < j\}$  为对任务  $j$  进行调度时已经完成调度的相关任务。对于两个相关任务  $i, j$  而言,定义相关距离  $d_{ij}$  为任务  $i$  到任务  $j$  的最长路径的长度,易得  $d_{ij} = -d_{ji}$ 。

### 2.3.2 任务 $j \in J$ 开始时间 $t_{ST,j}$ 可排区间点集

对任意任务  $j$  的  $t_{ST,j}$  可排区间  $[t'_{ES,j}, t'_{LS,j}]$  应满足下述条件:

$$t'_{ES,j} = \max\{t_{ES,j}, \max_{i \in R_e}(t_{ST,i} + d_{ij})\}, d_{ij} > 0 \quad (8)$$

$$t'_{LS,j} = \min\{t_{LS,j}, \min_{i \in R_e}(t_{ST,i} + d_{ij})\}, d_{ij} < 0 \quad (9)$$

若任务  $i, j$  不相关,即不存在  $d_{ij}$ ,则  $t'_{ES,j} = t_{ES,j}$ ; 否则,  $t'_{LS,j} = t_{LS,j}$ 。

由于存在空窗期约束,因此对任意任务  $j \in A_e$  的可排区间点集需剔除处于空窗期区间内的不可行解。假定关键资源  $p_k$  的可用区间为  $[t_{e,m}, t_{b,m}]$ ,则对任意任务  $j \in A_e$  的  $t_{ST,j}$  可排区间  $[t'_{ES,j}, t'_{LS,j}]$ ,需与  $[t_{e,m}, t_{b,m} + 1 - t_j]$  取交集,即满足

$$\sum_{m=1}^M t_{e,m} \leq \sum_{t=t_{ST,j}}^{t_{TF,j}} t \leq \sum_{m=1}^M t_{b,m} \quad (10)$$

从而形成最终的  $t_{ST,j}$  可排区间离散点集

$$S_{move,j} = \{t_{ST,j1}, t_{ST,j2}, \dots, t_{ST,jn}\} \quad (11)$$

### 2.3.3 任务 $j \in J$ 开始时间 $t_{ST,j}$ 可变区间点集

将任务  $j$  按照  $S_{move,j} = \{t_{ST,j1}, t_{ST,j2}, \dots, t_{ST,jn}\}$  依次排入,更新资源列表  $R_{tp}$ ,计算每一个排入点所对应的  $r'_{max}$ ,将  $r'_{max}$  的最小值记为  $r'_{min}$ ,将所有取得  $r'_{min}$  的  $t_{ST,j}$  记为任务  $j$  的开始时间  $t_{ST,j}$  的可变区间点集

$$S_{change,j} = \{t'_{ST,j1}, t'_{ST,j2}, \dots, t'_{ST,jn}\} \quad (12)$$

任务  $j$  开始时间  $t_{ST,j}$  的可变区间点集即为任务  $j$  开始时间  $t_{ST,j}$  的决策区间。

### 2.3.4 局部最优判断基准

对于任意非关键任务  $j$  而言,易得  $S_{change,j} \subseteq S_{move,j}$ ,而  $S_{change,j}$  中的点实际上为只考虑任务  $j$  的资源总量最小解集,显然集合内的所有点未必都能对应全局下的最小资源总成本。

根据优先级排入非关键任务,待排任务的位置确定受后序任务的影响。因此,通过考虑任务  $j$  的调度位置对后序任务的排入影响来进一步确定  $t_{ST,j}$ ,即使得连续排入的两个任务间结果最优。

第三阶段算法步骤如下所示:

**步骤 1**  $j_0^*$  按照调整区间点集  $S_{change,j_0^*}$  排入;将  $j_0^*$  之后的关键任务按照最早开始时间规则排入,更新集合  $S, S'$ 。

**步骤 2** 生成当前任一时刻  $t$  的资源列表  $R_{tp}$ 。

**步骤 3** 按照优先级顺序选择非关键任务优先级列表  $L_p$  中任务  $j$ ,获得其对应的  $R_e = \{i \in S | i < j\}$ ,对于任意任务  $i \in R_e$  求出相关距离  $d_{ij}$ ,继而得出  $t_{ST,j}$  可排区间点集  $S_{move,j}$ 。

**步骤 4** 将任务  $j$  按照  $S_{move,j}$  依次排入,更新资源列表  $R'_{tp}$ ,得到任务  $j$  开始时间  $t_{ST,j}$  的可变区间点集  $S_{change,j}$ 。若  $r'_{min} \leq r_{max}$ ,更新集合  $S, S'$ ,取点集  $S_{change,j}$  的第一个点排入,返回步骤 3;若  $r'_{min} > r_{max}$ ,

则转入步骤 5.

**步骤 5** 将任务  $j$  在列表  $L_p$  中的上一任务按照可变区间点集  $S_{\text{change},j}$  依次排入, 返回步骤 3; 若上一任务按照可变区间点集  $S_{\text{change},j} = \emptyset$ , 则令  $r_{\text{max}} = r'_{\text{min}}$ , 转入步骤 1.

**步骤 6** 直至已安排任务集合  $S=J$ , 未安排任务集合  $S'=\emptyset$ . 输出此时  $r_{\text{max}}$ , 记入  $\text{result}[]$ . 按照  $j_0^*$  的开始时间  $t_{\text{ST},j_0^*}$  的调整区间点集  $S_{\text{change},j_0^*}$  依次移动, 重复步骤 1~6, 直至  $S_{\text{change},j_0^*} = \emptyset$ , 转入步骤 7.

**步骤 7**  $\text{result}[]$  中的最小值为最终结果, 输出该结果, 算法结束.

### 3 遗传算法设计

分析启发式算法的设计过程, 发现非关键任务的优先级以及关键任务的开始时间会对最终结果产生较大的影响, 而启发式算法中并未对这两项进行更多可能性的尝试. 因此, 本研究遗传算法的上层算法中, 对于决策项目非关键任务的优先级和关键任务的开始时间, 下层解码算法采用第 2 节中提出的启发式规则, 决策各非关键任务的开始、结束时间, 并以此方案的目标函数值作为遗传算法的适应度值, 返回到上层遗传算法, 再不断复制、交叉、变异, 通过循环迭代的方式, 得到近似最优解.

#### 3.1 编码

对于有  $J$  项任务的项目, 假设其中关键任务数量为  $a$ , 非关键任务数量为  $b$ , 易得  $a+b=J$ . 每组编码的第一层对应各非关键任务的调度优先级列表, 长度为  $b$ . 若编码第  $k$  位为  $j$ , 则代表任务编号为  $j$  的任务调度优先级为  $k$ . 第二层对应项目中各关键任务开始时间, 长度为  $a$ , 对于关键任务 1 假定开始时间为定值  $t_{\text{ST},1}=1$ , 后续关键任务开始时间在开始时间的可排区间点集内随机确定, 如图 3 所示.

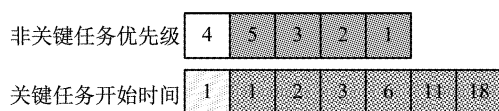


图 3 编码示例

Fig.3 Coding example

#### 3.2 遗传操作

##### 3.2.1 初始种群

初始种群由两部分组成: 一是按照第 2.1 节中启发式算法非关键任务的优先级列表以及第 2.2 节中连续排入的关键任务作为一个初始解; 二是在可

行范围内随机生成.

##### 3.2.2 选择

运用随机联赛选择算子, 适应值小的进入下一代种群. 适应值函数为

$$f(s) = \sum_{p=1}^P (C_p \max R_{tp}) \quad (13)$$

##### 3.2.3 交叉

本研究采用单点交叉的方法. 由于遗传算法采用双链表编码模式, 对非关键任务优先级以及关键任务开始时间进行编码, 交叉操作后可能出现不可解, 因此需要进行额外操作消除不可行解. 对于非关键任务链表需要消除交叉后可能出现的重复现象, 对于关键任务开始时间链表需要保证交叉后的时间在该任务开始时间的可行范围之内, 具体过程如图 4 所示.

(1) 对非关键任务优先级的链表进行交叉操作. 对染色体  $C_2$ , 选择与染色体  $C_1$  中  $x$  位置之前的基因不相同的基因链, 与染色体  $C_1$  中  $x$  位置之后的基因段互换, 生成新的染色体  $S_g$ ; 对染色体  $C_1$ , 同理生成新的染色体  $D$ .

(2) 对关键任务开始时间的链表进行交叉操作. 对染色体  $C_2$ , 选择与染色体  $C_1$  中  $x$  位置之前的基因链, 与染色体  $C_1$  中  $x$  位置之后的基因段互换, 若互换后的开始时间点处在区间  $[t_{\text{ES}}, t_{\text{LS}}]$  内, 则直接互换; 若互换后的开始时间点小于  $t_{\text{ES}}$ , 则互换后的基因取值为  $t_{\text{ES}}$ ; 若互换后的开始时间点大于  $t_{\text{LS}}$ , 则互换后的基因取值为  $t_{\text{LS}}$ ; 生成新的染色体  $S_g$ ; 对染色体  $C_1$ , 同理生成新的染色体  $D$ .

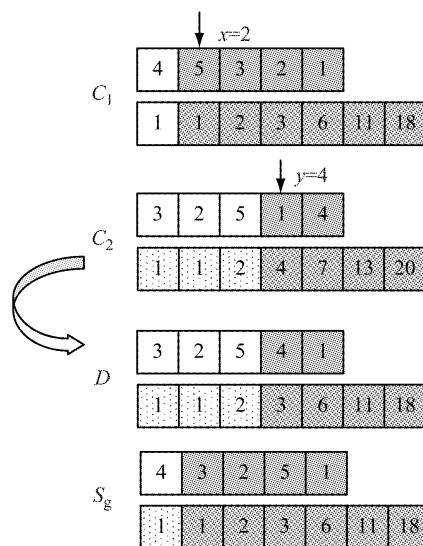


图 4 交叉操作示例

Fig.4 Cross example

##### 3.2.4 变异

本研究采用基本位变异的方法, 分以下两步

进行:

(1) 对  $k=1,2,\dots,b$ , 在  $[1,b]$  之间随机产生两个数  $x,y$ , 交换  $x,y$  两位置所对应的任务的优先级, 生成新的列表.

(2) 对  $k=2,3,\dots,a$ , 在  $[2,a]$  之间随机产生一个数  $z$ , 改变位置  $z$  所对应任务的开始时间, 并重新生成位置  $z$  后序所有关键任务的开始时间. 具体过程如图 5 所示.

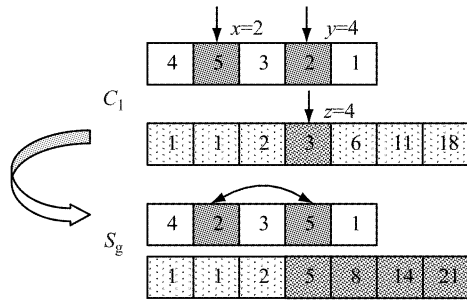


图 5 变异操作示例

Fig.5 Variation example

### 3.3 解码

利用上述遗传算法, 得到了非关键任务优先级列表以及关键任务的开始时间, 接下来对所得到的染色体根据第 2 节中介绍的启发式规则, 进行相应链表的解码操作.

## 4 数值实验

实验中所用测试算例基于测试问题库 PSPLIB 中的算例进行改造, 对于未提供的参数, 采用在一定大小范围内通过 random 函数随机产生自然数的方式来确定, 其中工期上限  $T$  取资源不受限下基于最早开始时间调度得到的任务最晚完成时间的 1.2 倍. 空窗期数  $M=1$ , 空窗期开始时间和结束时间随机生成, 长度不超过  $0.1T$ . 遗传算法设定的种群规模  $N=100$ , 最大迭代次数  $\lambda_{\max}=50$ , 交叉概率  $P_c=0.8$ , 变异概率  $P_m=0.1$ . 测试平台为 Intel Core i5 处理器, 2.20 GHz 主频, 4.00G 内存.

表 1~3 显示了小规模问题的算例结果, 算例规模依次为 10、20、30 项任务; 每组包含五个算例,  $C$  列、 $H$  列和  $G$  列分别表示 CPLEX 以及启发式算法和遗传算法在处理本组五个算例时得到的最好解的平均值取整. 对于包含 60 和 90 项任务的大规模问题, 启发式算法和遗传算法在求解速度上的差异较为明显, 而 CPLEX 在规定运算时间内无法求得结果, 故对于大规模问题分别对提出的两种算法进行

求解精度和求解速度的比较. 每个算例的空窗期约束在首次随机生成后统一设定, 保证比较的合理性. 表 4~5 为大规模问题的算例结果,  $v$  列表示每组算例在相应算法下各运行 10 次后得到的最小值的平均值,  $w$  列表示求得对应结果的平均时间, 单位为 s.

表 1~5 中:

$$g_1 = \frac{H-C}{H} \times 100\% \quad (14)$$

$$g_2 = \frac{G-C}{G} \times 100\% \quad (15)$$

$$g_3 = \frac{H-G}{H} \times 100\% \quad (16)$$

表 1 10 项任务实验结果

Tab.1 Results of 10 jobs

编号	$C$	$H$	$g_1/\%$	$G$	$g_2/\%$	$g_3/\%$
1	38.0	39.0	1.66	38.0	0.51	2.63
2	42.0	44.0	4.95	42.0	1.11	4.76
3	58.0	59.0	2.69	59.0	1.84	0.00
4	55.0	58.0	6.01	56.0	1.50	3.57
5	62.0	65.0	4.71	63.0	1.20	3.17
6	54.0	56.0	3.60	56.0	2.79	0.00
7	63.0	66.0	4.94	64.0	0.83	3.13
8	70.0	74.0	6.01	72.0	2.94	2.78
9	78.0	80.0	2.10	79.0	1.22	1.27
10	66.0	69.0	3.84	68.0	1.77	1.47
均值	58.6	61.0	4.05	59.7	1.57	2.28

表 2 20 项任务实验结果

Tab.2 Results of 20 jobs

编号	$C$	$H$	$g_1/\%$	$G$	$g_2/\%$	$g_3/\%$
1	54.0	58.0	6.62	56.0	3.69	2.84
2	57.0	60.0	4.48	59.0	2.22	2.26
3	52.0	55.0	7.13	54.0	5.35	1.67
4	51.0	55.0	7.62	53.0	3.72	3.82
5	71.0	75.0	6.52	72.0	2.25	4.15
6	67.0	72.0	6.51	69.0	2.83	3.57
7	56.0	59.0	4.77	57.0	2.47	2.28
8	55.0	59.0	6.05	57.0	3.28	2.77
9	50.0	54.0	6.27	52.0	3.18	3.08
10	62.0	64.0	5.19	63.0	3.01	2.19
均值	57.5	61.1	6.12	59.2	3.20	2.86

从表 1~3 中  $g_1$  项和  $g_2$  项可以发现, 对任务数为 10、20、30 的小规模算例, 所设计的启发式算法与 CPLEX 求解得到的最优解分别相差 4.05%、6.12%、10.85%, 遗传算法求得的结果与 CPLEX 求得的最优解分别相差 1.57%、3.20%、6.73%. 可以证实所提出的启发式算法和遗传算法在一定误差范围内均能求解出较好的结果, 遗传算法求解精度更高; 随着算例规模的增大, 启发式算法在求解精度上的局限性越发明显; 小规模算例中启发式算法与遗传算法的  $g_3$  值相差 3% 左右.

表 3 30 项任务实验结果

Tab.3 Results of 30 jobs

编号	C	H	$g_1/\%$	G	$g_2/\%$	$g_3/\%$
1	39.0	42.0	10.14	41.0	7.45	2.53
2	38.0	41.0	10.02	40.0	5.70	4.11
3	37.0	40.0	9.94	39.0	5.50	4.36
4	40.0	44.0	9.69	42.0	5.08	4.46
5	62.0	69.0	12.24	67.0	8.68	3.25
6	64.0	72.0	11.64	69.0	7.40	4.02
7	62.0	68.0	9.97	66.0	6.24	3.48
8	67.0	74.0	11.31	71.0	6.34	4.73
9	56.0	62.0	12.84	60.0	7.60	4.76
10	55.0	61.0	10.69	59.0	7.29	3.34
均值	52.0	57.3	10.85	55.4	6.73	3.90

表 4 60 项任务实验结果

Tab.4 Results of 60 jobs

编号	启发式算法		遗传算法		$g_3/\%$
	$v$	$w$	$v$	$w$	
1	69.0	34.58	62.0	57.04	10.14
2	73.0	32.76	64.0	43.72	12.33
3	69.0	20.03	62.0	45.96	10.14
4	70.0	35.65	63.0	44.15	10.00
5	68.0	30.71	61.0	47.91	10.29
6	70.0	27.66	62.0	43.27	11.43
7	85.0	27.78	74.0	46.40	12.94
8	62.0	24.05	57.0	59.88	8.06
9	72.0	31.66	64.0	57.43	11.11
10	79.0	30.53	73.0	43.20	7.59
均值	71.7	29.54	64.2	48.89	11.87

表 5 90 项任务实验结果

Tab.5 Results of 90 jobs

编号	启发式算法		遗传算法		$g_3/\%$
	$v$	$w$	$v$	$w$	
1	86.0	57.10	72.0	115.47	16.28
2	84.0	62.39	75.0	126.73	10.71
3	89.0	58.66	75.0	120.27	15.73
4	79.0	60.21	68.0	111.97	13.92
5	84.0	62.07	75.0	127.11	10.71
6	89.0	59.63	75.0	128.90	15.73
7	68.0	62.09	61.0	121.89	10.29
8	83.0	63.00	74.0	113.43	10.84
9	102.0	60.05	91.0	101.03	10.78
10	87.0	60.84	77.0	100.97	11.49
均值	85.1	60.60	74.3	116.78	13.57

对于任务数为 60 和 90 的大规模算例而言,通过表 4、表 5 中的  $g_3$  项结果可以看出,两种算法的  $g_3$  值分别为 11.87%、13.57%,相差较大,虽无法与精确结果作比较,但可以说明求解大规模问题时,与启发式算法相比,采用遗传算法可以明显提升求解精度. 比较两表中的  $w$  列结果发现,任务数为 60 时两种算法的平均求解时间分别为 29.54、48.89 s,任务数为 90 时两种算法的平均求解时间分别为 60.60、116.78 s,启发式算法的求解速度更快. 所设

计的启发式算法的时间复杂性为  $O(n^2)$ ,由于提出的遗传算法在解码和评估阶段嵌套了相同的启发式规则,随着算例规模的增大,两者在求解时间上的差异也随之变大. 对于大规模算例,遗传算法与启发式算法相比具有较高的精度,但求解速度较慢. 由此表明,提出的两种算法在不同的层面上具有不同的应用价值.

## 5 总结与展望

针对飞机移动式装配线上存在的关键资源带有空窗期约束这一特性,提出以连续排入的两个非关键任务间结果最优的启发式规则,分别设计了构造启发式算法和改进遗传算法求解带资源空窗期的 RIP,并通过数值实验验证了所提出算法的求解精度和求解速度在不同层面上的有效性. 小规模算例中,两种算法所得解与最优解间的平均差值在 11% 以内;大规模算例中,遗传算法的求解精度较高,启发式算法的求解速度较快. 将飞机移动式装配线抽象为项目调度资源投入问题的研究对于提升我国飞机制造业水平、带动经济发展和科技进步具有重要意义.

## 参考文献:

- [1] MÖHRING R H. Minimizing costs of resource requirements in project networks subject to a fixed completion time [J]. Operations Research, 1984, 32(1): 89.
- [2] DREXL A, KIMMS A. Optimization guided lower and upper bounds for the resource investment problem [J]. Operational Research Society, 2001, 52(3): 340.
- [3] RODRIGUES S B, YAMASHITA D S. An exact algorithm for minimizing resource availability costs in project scheduling [J]. European Journal of Operational Research, 2010, 206(3): 562.
- [4] SONG Y, LIU J, WIMMERS M O, et al. A differential evolution algorithm with local search for resource investment project scheduling problems [C] // Evolutionary Computation. Sendai: IEEE, 2015: 1725-1731.
- [5] MYŠKOWSKI P B, SKOWROŃSKI M E, OLECH L P, et al. Hybrid ant colony optimization in solving multi-skill resource-constrained project scheduling problem [J]. Soft Computing, 2015, 19(12): 3599.
- [6] VAN PETEGHEM V, VAN HOUCKE M. An artificial immune system algorithm for the resource availability cost problem [J]. Flexible Services and Manufacturing Journal, 2013, 25 (1/2): 122.
- [7] HE L H, ZHANG L Y. Dynamic priority rule-based forward-backward heuristic algorithm for resource leveling problem in construction project [J]. Journal of the Operational Research

- Society, 2013, 64(8): 1106.
- [8] QI J J, LIU Y J, JIANG P, *et al.* Schedule generation scheme for solving multi-mode resource availability cost problem by modified particle swarm optimization [J]. Journal of Scheduling, 2015, 18(3): 285.
- [9] 胡淑芳. 考虑资源技能和时间窗特性的任务可拆分项目调度[D]. 武汉: 华中科技大学, 2012.
- HU Shufang. Preemptive project scheduling with resources of multi-skill and time-windows [D]. Wuhan: Huazhong University of Science and Technology, 2012.
- [10] 廖广瑞, 刘振元, 毕阳. 多技能资源时间窗约束下项目调度研究[C]//中国控制与决策会议. 长沙: 出版者不详, 2014: 4885-4891.
- LIAO Guangrui, LIU Zhenyuan, BI Yang. Project scheduling with time window constraints on multi-skill resources[C]//Chinese Control and Decision Conference. Changsha: [s. n.], 2014: 4885-4891.
- [11] 蔡方中, 胡丹, 叶雷宏. 基于时间窗和关键链的多项目资源分配问题的研究[J]. 科技管理研究, 2013, 33(13): 229.
- QI Fangzhong, HU Dan, YE Leihong. Study on resource allocation of multi-project based on time window and critical chain[J]. Science and Technology Management Research, 2013, 33(13): 229.
- [12] 刘振元, 黄亚健. 资源多时间窗约束下的项目调度[J]. 系统工程, 2014(10): 126.
- LIU Zhenyuan, HUANG Yajian. Project scheduling with constraints of multiple time windows on resources[J]. Systems Engineering, 2014(10): 126.
- [13] 吴怡薇, 陆志强. 飞机移动装配线资源水平问题的建模研究[J]. 工业工程与管理, 2017, 22(1): 95.
- WU Yiwei, LU Zhiqiang. Modeling resource leveling for aircraft moving assembly line[J]. Industrial Engineering and Management, 2017, 22(1): 95.

#### (上接第 1469 页)

- [21] SIMONS-MORTON B G, KLAUER S G, OUIMET M C, *et al.* Naturalistic teenage driving study: findings and lessons learned [J]. Journal of Safety Research, 2015, 54: 41.
- [22] TOLEDO T, MUSICANT O, LOTAN T. In-vehicle data recorders for monitoring and feedback on drivers' behavior[J]. Transportation Research, Part C: Emerging Technologies, 2008, 16(3): 320.
- [23] ALJAAFREH A, ALSHABATAT N, AL-DIN M S N. Driving style recognition using fuzzy logic [C] // 2012 IEEE International Conference on Vehicular Electronics and Safety (ICVES 2012). [S. l.]: IEEE, 2012: 460-463.
- [24] REYMOND G, KEMENY A, DROULEZ J, *et al.* Role of lateral acceleration in curve driving: driver model and experiments on a real vehicle and a driving simulator[J]. Human Factors, 2001, 43(3): 483.
- [25] CHOI J, TAY R, KIM S, *et al.* Turning movements, vehicle offsets and ageing drivers driving behaviour at channelized and unchannelized intersections [J]. Accident Analysis & Prevention, 2017, 108: 227.
- [26] MARTINEZ C M, HEUCKE M, WANG F Y, *et al.* Driving style recognition for intelligent vehicle control and advanced driver assistance: a survey [J]. IEEE Transactions on Intelligent Transportation Systems, 2018, 19(3): 666.
- [27] KARGINOVA N, BYTTNER S, SVENSSON M. Data-driven methods for classification of driving styles in buses [R]. Detroit: SAE, 2012.
- [28] Mapbox. Open source routing machine[EB/OL]. [2018-06-15]. <http://project-osrm.org/>.
- [29] 白旭, 曾胜男, 杜豫川, 等. 城市快速路自由流车速研究[J]. 交通与运输, 2006(增2): 28.
- BAI Xu, ZENG Shengnan, DU Yuchuan, *et al.* The research of urban expressway free-flow velocity [J]. Traffic & Transportation, 2006(S2): 28.
- [30] MACQUEEN J. Some methods for classification and analysis of multivariate observations [C] // Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. Berkeley: University of California Press, 1967: 281-297.
- [31] KAUFMAN L, ROUSSEEUW P J. Finding groups in data: an introduction to cluster analysis[M]. New York: John Wiley & Sons, 2009.
- [32] 上海违章查询. 上海交通违章行为排行[EB/OL]. [2018-06-30]. [https://weizhang.zuzuche.com/wz\\_action\\_renking\\_list.php?page=1&city\\_id=0&province\\_id=310000](https://weizhang.zuzuche.com/wz_action_renking_list.php?page=1&city_id=0&province_id=310000).
- Shanghai traffic violation query. Shanghai traffic violation ranking [EB/OL]. [2018-06-30]. [https://weizhang.zuzuche.com/wz\\_action\\_renking\\_list.php?page=1&city\\_id=0&province\\_id=310000](https://weizhang.zuzuche.com/wz_action_renking_list.php?page=1&city_id=0&province_id=310000).
- [33] CHEN K T, CHEN H Y W. Driving style clustering using naturalistic driving data[J]. Transportation Research Record, 2019, 2673(6): 176.