

基于项目拆分决策的多模式资源投入

陆志强, 宗保氏

(同济大学 机械与能源工程学院, 上海 201804)

摘要: 结合一类实际生产决策需求, 提出了基于项目拆分决策的多模式资源投入调度问题, 并以资源投入最小化为优化目标, 建立了问题的数学模型. 针对模型特点, 提出了包含项目拆分算法和多模式资源投入型项目调度算法的双层优化算法, 其中项目拆分算法通过将作业在不同子项目之间有效移动获得合理的拆分方案, 多模式资源投入型项目调度算法通过分析不同作业对时间约束和资源约束的影响来确定优先级规则, 进而得到最佳调度方案. 应用 PSPLIB 标准算例进行数据实验, 结果证明了算法的有效性和可靠性.

关键词: 项目拆分; 资源投入; 集成优化

中图分类号: F273

文献标志码: A

Multi-Mode Resource Investment Project Scheduling Problem Based on Project Splitting

LU Zhiqiang, ZONG Baoshi

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

Abstract: Combined with a class of actual production decision-making needs, a multi-mode project scheduling problem based on project splitting decision was proposed. In order to minimize the resource investment, an integrated optimization model including project splitting and multi-mode resource investment model was build. Considering the characteristics of the model, a double-layer optimization algorithm, including project splitting algorithm and multi-mode resource investment project scheduling algorithm, was presented. The project splitting algorithm obtained a reasonable resolution scheme by moving jobs among different sub-projects and the multi-mode resource-based project scheduling algorithm determined the priority rules by analyzing the influence of different jobs on time constraints and resource constraints. The data experiment was carried out by using the Project Scheduling Problem Library (PSPLIB) standard example. Results validated the algorithm.

Key words: project splitting; resource investment; integrated optimization

在飞机、轮船等大型工业品的装配过程中, 由于产品规模较大, 工艺操作繁琐等自身特点, 导致传统固定站位式生产方式效率低稳定性差, 从而无法满足市场需求, 因此由汽车精益生产方式衍变而来的具有高效率和高稳定性特点的移动式装配生产线逐渐成为大型工业品的主流生产方式. 以飞机移动装配线为例, 类似于汽车移动生产线, 装配线通常被划分为多个工位, 飞机依次缓慢通过所有工位, 按照装配计划完成全部装配作业. 然而由于飞机装配工期长, 线边资源特别是关键资源不再被固定在特定工位, 而是整个装配线共享所有资源^[1], 所以这类问题可以看作一类项目调度问题. 将整个飞机装配过程抽象成一个大项目, 装配线上每个工位需要完成的装配作业集合组成了飞机装配的一个子项目, 在装配线资源共享条件下对这些子项目进行调度, 可以看作并行执行的多项目调度问题. 进一步, 在实际生产中, 作业往往有多种执行模式, 不同的执行模式对应于不同的资源用量和加工时间, 而且由于飞机装配线每个工位含有大量种类繁多的装配作业, 所以合理地安排作业执行模式可以有效的节约资源缩短工期. 综上所述, 飞机移动装配线总装调度问题在理论上是一类具有作业多模式、约束复杂等特征, 且关联资源投入的大规模多项目协同调度问题. 这一类问题可以称为基于项目拆分决策的多模式资源投入调度问题 (multi-mode resource investment project scheduling problem based on project splitting, MRIPSP-PS). 目前尚未有针对 MRIPSP-PS 问题的直接研究, 与经典的 MRIPSP (multi-mode resource investment project scheduling problem) 问题相比,

收稿日期: 2017-11-22

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, 主要研究方向为生产系统计划与调度、物流与供应链建模.

E-mail: zhiqianglu@tongji.edu.cn

MRIPSP-PS 的决策不仅涵盖了作业调度、作业模式选择与资源投入等决策,同时还集成了项目拆分决策,并且这些决策相互影响,现有求解 MRIPSP 问题的方法也无法直接应用于求解 MRIPSP-PS 问题,因此对 MRIPSP-PS 问题的建模与算法研究,有重要的理论及实际意义。

显然,MRIPSP-PS 问题与 MRIPSP 问题具有相关性,所以现有的 MRIPSP 问题的算法对求解 MRIPSP-PS 问题具有一定参考意义.对 MRIPSP 问题的研究起始于单模式下的资源投入项目调度问题(Resource Investment Project Scheduling Problem, RIPSP).Mohring^[2]最早提出了 RIPSP 问题,并设计了精确算法对问题进行求解.通过改进 Mohring 的算法,Demeulemeester^[3]设计了一种称为 Minimum Bounding Algorithm(MBA)的精确算法,通过算例对比证明了 MBA 具有更高的求解效率.Rodrigues 和 Yamashita^[4]将启发式算法与类似于 MBA 的分支算法相结合,加快了求解速度.Rangaswamy^[5]详细地介绍了一个新的基于分支定界的算法,进一步提高了求解效率.然而精确算法仅能用于小规模问题,无法有效求解大规模问题.为了求解大规模问题的,不同学者提出了各种启发式算法.Kelly^[6]最早研究了启发式算法,算法包括优先规则(Priority Rule, PR)和调度生成机制(Schedule Generation Scheme, SGS)两部分.之后的学者,包括 Kurtulus^[7],Boctor^[8],Lawrence^[9],分别提出了不同的优先级规则.Afshar-Nadjafi^[10]引入了资源释放时间的概念,建立了一种新的多模式资源投入问题模型.Chen^[11]设计了 3 类规则,分别用于选择作业、作业的执行模式和作业的开始时间,并通过 3 类规则的不同组合来尝试求解多模式资源投入问题。

然而现有的研究在应用于 MRIPSP-PS 问题时存在几点不足:①MRIPSP-PS 问题需要对项目进行拆分,需要研究如何确定合理的拆分方案;②文献中对多模式资源投入型项目调度问题的研究较少,需要更深入研究作业模式选择和作业优先规则,以及其中涉及的时间约束与资源约束的相互影响关系.因此,针对上述问题,本文提出了包含项目拆分算法和多模式资源投入型项目调度算法的双层优化算法,上层项目拆分算法将整个项目合理的拆分成多个子项目;下层多模式资源投入型项目调度算法,通过时间因素评估函数和资源因素评估函数,提出了一种新的优先级规则。

1 问题描述及数学模型

1.1 问题描述

以飞机移动装配线为应用背景,飞机总装项目可以用节点式网络 $G=(V, E)$ 表示,其中 V 为项目网络 G 中的节点集合,代表项目的作业集合, E 为项目网络中弧的集合,代表作业之间的优先关系.项目由编号为 $j(j=1, 2, \dots, J)$ 的 J 项作业组成,另外添加作业 0 和作业 $J+1$ 两个虚作业分别代表项目的开始作业和结束作业。

一架飞机所需的全部装配作业及优先关系如图 1 所示.飞机缓慢通过整个装配线,同时利用装配线两侧的装配资源,按照装配调度计划完成总装所需的全部装配作业任务。

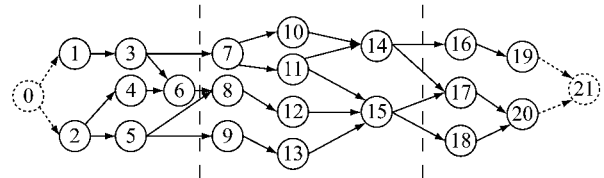


图 1 单架飞机总装作业

Fig.1 A single aircraft assembly operation

装配线对多架飞机同时进行装配,每隔一个装配节拍,一架飞机完成装配离开生产线,同时一架待装配飞机进入生产线.为了对装配线上的多架飞机进行调度建模,同时满足装配节拍的周期性调度,可以将装配线上各个工位对应为多个“虚拟大工位”,进而把一架飞机所包含的全部装配作业合理的分配给这些“虚拟大工位”,每个“虚拟大工位”对应一个装配作业子集合,称为子项目.例如以图 1 中的虚线为界,将飞机总装作业划分为 3 个子项目,划分后的示意图如图 2 所示。

在此情形下,一方面,对一架飞机来讲,它通过了整个装配线也就通过了所有子项目,完成所有需要的装配任务;另一方面,每隔一个装配节拍,装配线当前子项目位置上的飞机完成对应作业子集中所有任务后进入下一个子项目(或者装配完成离开生产线),同时后一架飞机进入当前子项目位置.由于各个子项目的装配过程在时间上是不同的,这多个子项目可以作为并行开展的多个项目共享整个装配线的资源,这样既实现了资源在项目内和项目间的共享,又使得资源的分配计划符合装配节拍的周期性要求.显然,装配线的资源投入量取决于子项目划分决策和划分后多个子项目的资源投入调度计划。

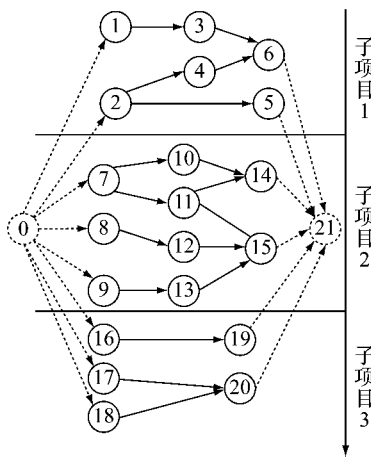


图 2 飞机总装作业子项目拆分

Fig.2 The sub projects splitting of aircraft assembly operation

1.2 数学模型

不同于单模式问题,在作业多模式条件下的资源投入调度问题中,作业 j 有 $m(m=1,2,\dots,M)$ 种执行模式,分别对应的执行时间为 d_{jm} ,全部作业共享 K 种可更新资源, $r_{jmk}(k=1,2,\dots,K)$ 表示作业 j 按照模式 m 需要第 k 中资源的数量。 P_j 表示作业 j 的紧前作业集合,作业 $h(h \in P_j)$ 表示作业 h 为作业 j 的紧前作业, S_j 表示作业 j 的紧后作业集合。按照从左到右的顺序,定义第 i 个“虚拟大工位”对应的作业任务集合记为子项目 $i(i=1,2,\dots,N)$ 。项目节拍周期记为 C ,对时间离散化处理,时间点集合 $T = \{t|t=1,2,\dots,C\}$,设 U 为一个足够大的正数。

MRIPSP-PS 问题要求在满足作业优先关系约束和节拍周期约束的前提下,确定作业的所属子项目和作业的开始时间,与基本 RIPSP 问题一样,MRIPSP-PS 问题的目标函数也是项目资源投入最小化。

项目资源投入记为 A ,具体决策变量如下: Y_{ij} 为 $0,1$ 变量,作业 j 属于子项目 i 为 1 ,否则为 0 ; x_{jmt} 为 $0,1$ 变量,作业 j 在选择模式且在时刻完成为 1 ,否则为 0 。

目标函数:

$$\min A = \sum_{k=1}^K \tau w_k R_k \tag{1}$$

约束条件:

$$\sum_{s=i+1}^N Y_{sj} \leq U(1 - Y_{ij}), \forall i \in N, \forall j \in J, \forall h \in P_j, \forall t \in T \tag{2}$$

$$\sum_{m=1}^M \sum_{t=1}^C x_{jmt} = 1, \forall j \in J \tag{3}$$

$$\sum_{m=1}^M \sum_{t=1}^C t x_{jmt} - \sum_{m=1}^M \sum_{t=1}^C (t - d_{jm}) x_{jmt} \leq U(2 - Y_{ij} - Y_{jh}), \forall i \in N, \forall j \in J, \forall h \in P_j \tag{4}$$

$$\sum_{i=1}^N Y_{ij} = 1, \forall j \in J \tag{5}$$

$$\sum_{j=1}^J \sum_{m=1}^M r_{jmk} \cdot \sum_{e=t}^{t+d_{jm}-1} x_{jme} \leq R_k, \forall k \in K, \forall t \in T \tag{6}$$

$$x_{jmt} \leq C, \forall j \in J, t \in T \tag{7}$$

其中,式(1)表示问题的目标函数为最小化资源投入;式(2)表示在拆分决策中,任意作业的紧前作业不能被分配到该作业所在子项目的后续子项目中;式(3)表示一项作业只能在一种模式下被执行一次;式(4)表示优先关系约束,任意一项作业开始时间必须大于所有实际有效紧前作业的完成时间;式(5)表示任意一项作业能且仅能被分配到某一个子项目中;式(6)表示资源实际用量,项目所需资源量必须大于或等于任意时刻所有在执行作业的资源使用总量;式(7)表示工期约束,任意作业必须在项目工期内完成,不能延期。

2 PSS-JRTS 双层迭代算法

根据模型的决策变量,需要做出两种决策,针对决策变量 Y_{ij} 的决策称为拆分决策,针对决策变量 x_{jmt} 的决策称为多模式资源投入项目调度决策。基于项目拆分决策的多模式资源投入问题不仅要决策作业的所属节拍,还要决策作业的执行模式及开始时间,因此本文提出一种双层迭代算法,PSS-JRTS 算法 (project split scheme, joint resource and time scheme),上层 PSS(project split scheme)算法包含项目初始拆分策略和作业移动调整策略,通过对不断有效调整拆分结果获得合理的拆分方案,下层 JRTS(joint resource and time scheme)算法通过分析安排不同作业时对项目时间约束和资源约束的影响来确定作业的优先级规则。

2.1 算法整体步骤

算法的整体结构:上层 PSS 算法划分作业的所属子项目,下层 JRTS 算法确定作业的执行模式及开始时间来生成调度方案,具体步骤如下:

(1)调用上层 PSS 算法的项目初始拆分策略,将项目中所有作业划分到 N 个子项目中,初始化项目拆分结果 Y_{ij} ,设置拆分调整迭代次数 $I_{iter} = 100, z$

表示当前迭代次数,初始化为零,项目资源投入记为 A_{best} ,初始化为无穷大.

(2)调用下层 JRTS 算法对当前拆分方案求解,更新 x_{jmu} ,得到当前方案的资源投入 A' .若 $A' < A_{\text{best}}$,更新 $A_{\text{best}} = A'$.

(3)若 $z < I_{\text{iter}}$,调用上层 PSS 算法的拆分调整策略,获得新的拆分方案 Y' , $z = z + 1$,并转入(2);否则,转入(4).

(4)输出 A_{best} .

2.2 上层 PSS 算法

为了获得一个合理的拆分方案,本文采用一种先将项目作初始拆分,再不断调整拆分结果的启发式算法.项目初始拆分策略将项目拆分为 N 个子项目,使得每个作业都属于且仅属于某一个子项目.在每次迭代中,通过将某个子项目中的一个作业移动到其它子项目的方式来调整拆分结果,获得不同的拆分方案.

2.2.1 项目初始拆分策略

初始拆分策略是要在满足作业优先关系的其条件下将作业划分到 N 个子项目中,本文采用了基于 CPM(critical path method)的初始拆分方案^[12].对未拆分的初始项目,由 CPM 生成满足优先关系的调度方案,得到每个作业的结束时间 $t_{\hat{f}_j}$,其中,作业的工期均采用作业所有模式的平均工期计算.为了将项目拆分成 N 个可行的子项目,按照关键链工期长度 $t_{\hat{f}_{j+1}}$ 将项目拆分成 N 份.定义时间截断区间长度 L 为

$$L = \frac{t_{\hat{f}_{j+1}}}{N} \tag{8}$$

对每一个作业,按照以下方案分配到子项目中:

$$Y_{ij} = \left\{ \begin{array}{l} 1, (i-1)L \leq t_{\hat{f}_j} < iL, \\ \forall j \in J, i = 1, 2, \dots, N-1 \\ 1, (i-1)L \leq t_{\hat{f}_j} < t_{\hat{f}_{j+1}}, i = N \end{array} \right\} \tag{9}$$

由于 CPM 得到每个作业的结束时间 $t_{\hat{f}_j}$ 均小于其紧前作业的结束时间,这样的拆分方案保证了任意作业的紧前作业不会被分配到该作业所在子项目的后续子项目中,从而使得拆分方案可行.

2.2.2 拆分调整策略

按照初始拆分策略将项目拆分为 N 个子项目后,通过在不同子项目间移动作业来调整拆分结果,获得新的拆分方案.每次移动需要确定两个变量,包括从哪一个子项目中移出作业和确定此子项目中的可移动的作业.为了获得一个较优的拆分方案,使目标函数最小化,那么应该避免在拆分方案中出现各

个子项目中作业个数分布差异过大的情况.本文设计了一种考虑子项目中作业个数分布情况的优先规则来决策每次迭代中用于移出作业的子项目.同时,根据不同子项目中作业的可移动条件来确定可移动作业的集合,最终从可移动作业的集合选定要移动的作业.

用 m_i 表示子项目 i 的作业个数, α_i 表示选择子项目 i 作为移出作业子项目的概率:

$$\alpha_i = \frac{m_i}{\sum_{i=1}^N m_i} \tag{10}$$

采用轮盘赌的方式选定子项目来移出作业.这样,拥有较多作业的子项目被选中的概率越高,同时轮盘赌的选择方式带来一定的随机性.对于选定的子项目,其中的作业必须满足一定的条件时才可以被移动到其他子项目中.如图 2 所示,用 n 表示子项目序号,从上到下序号为 $n=1, 2, 3, \dots, N$.用 \hat{P}_j, \hat{S}_j 分别代表作业 j 的有效紧前作业集合和有效紧后作业集合,并取图 2 中的三个子项目中的子项目 2 为例(图 3),子项目对应的可移动作业集用 F_n 表示.以下分析当图 3 为不同子项目 n 时,可移动作业集 F_n 应满足的条件.

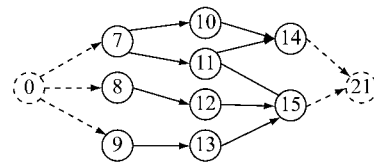


图 3 子项目 n 中可移动作业

Fig.3 The moving jobs of sub project n

(1)若 $n=1$,图 3 表示第一个子项目,此时子项目的作业只能向后移动到子项目 2 中,不能前移.图 3 中作业 14 和作业 15 可以移动,而作业 7—13 都存在有效紧后作业,不能直接移动.即子项目 1 中的作业需满足紧后作业为空时才能移动,则有

$$F_1 = \{j \mid \hat{S}_j = \emptyset \cap Y_{1j} = 1, \forall j \in J\} \tag{11}$$

(2)若 $1 < n < N$,那么图 3 既不是第一个子项目也不是最后一个子项目,而是类似于图 2 中的子项目 2.此时子项目中的作业既可以前移到子项目 $n-1$ 中,也能向后移动到子项目 $n+1$ 中.其中,图 3 中作业 7、8、9 可以向前一个子项目移动,作业 14、15 可以向后一个子项目移动,而作业 10—13 不能移动.子项目 $n(1 < n < N)$ 的可移动作业集 F_n 为

$$F_n = \{j \mid (\hat{P}_j = \emptyset \cup \hat{S}_j = \emptyset) \cap Y_{nj} = 1, \forall j \in J\} \tag{12}$$

(3)若 $n=N$,此时图 3 表示最后一个子项目,

子项目中作业只能前移到子项目 $n-1$ 中,不能向后移动.图 3 中作业 7、8、9 可以前移,其他作业都不能移动,即子项目 N 中的可移动作业集 F_N 为

$$F_N = \{j \mid \dot{P}_j = \phi \cap Y_{N_j} = 1, \forall j \in J\} \quad (13)$$

当选定子项目来移出作业,本文按照每次从可移动作业集 F_n 中移动一个作业到其他子项目的方式来调整项目拆分方案,得到新的拆分结果 Y_{ij} .

2.3 下层 JRTS 算法

通过上层 PSS 算法,将项目拆分为多个并行的子项目,如图 2 所示,下层 JRTS 算法需要确定作业的执行模式以及开始时间,即决策变量 x_{jmt} . 本文采用启发式算法^[6],包含进度生成机制和优先级规则两部分.其中进度生成机制为串型进度生成机制(serial schedule generation scheme, SSGS),而传统的优先级规则涉及到选择每一次要排的作业以及这个作业的开始时间和执行模式.选择不同的作业、不同的执行模式和不同的开始时间,会对调度计划产生不同的影响,主要包括对时间约束的影响和对资源约束的影响.本文提出一种同时考虑资源与时间调度(JRTS, joint resource and time scheme)的函数,分别分析作业安排对两种约束的影响,设计了一种有效的优先级规则,同时决策要排的作业以及相应的执行模式和开始时间.

2.3.1 时间因素评估函数

时间因素评估函数用于评价作业的安排方案对时间约束的影响.根据 CPM 可以得到作业 j 的最早及最晚开始时间,分别对最短工期模式(shortest duration mode)和最长工期模式(longest duration mode)得到的变量做出如下定义: t_{SEST_j} 为最早开始时间(最短工期模式); t_{SLST_j} 为最晚开始时间(最短工期模式); t_{SLFT_j} 为最晚结束时间(最短工期模式); t_{LIFT_j} 为最晚结束时间(最长工期模式).如图 4 所示,作业 j 的执行时间应在时间窗 $[t_{SEST_j}, t_{SLFT_j}]$ 内,且当位于时间窗的前半段 $[t_{SEST_j}, t_{LIFT_j}]$ 内时,因为不超过最长工期模式下的最晚结束时间,所以作业 j 的后续作业均可以选择所有的执行模式执行而不会延期.当作业 j 的执行时间位于 $[t_{LIFT_j}, t_{SLFT_j}]$ 内时,超过了最长工期模式下的最晚结束时间,此时若作业 j 的后续作业均按照最长工期模式执行,必然无法在项目总工期内完工,引起延期.极端情况为作业 j 的完工时间为 t_{SLFT_j} ,此时作业 j 的后续作业只能选择最短工期模式执行,否则必然引起延期.

综合考虑作业 j 的执行时间处在不同位置是对后续作业的影响,当作业 j 以模式 m 在 t 时刻开始

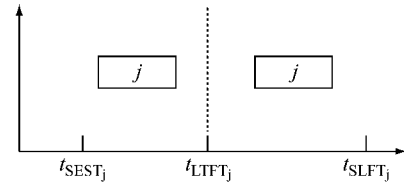


图 4 作业 j 的可排区间

Fig. 4 The interval of the job j

的时间因素评估函数为

$$f_{\text{time}}(j, m, t) = \frac{\text{Max}(t_{\dot{f}_j}(j, m, t) - t_{LIFT_j}, 0)}{t_{SLFT_j} - t_{LIFT_j}} \quad (14)$$

式(14)中 $t_{\dot{f}_j}(j, m, t) = t + d_{jm}$,表示作业 j 以模式 m 在 t 时刻开始的结束时间,分母用于标准化最大值为 1.当作业 j 在 t_{LIFT_j} 结束时, $f_{\text{time}}(j, m, t)$ 为 0;随着结束时间的推迟, $f_{\text{time}}(j, m, t)$ 会逐渐增大,且当作业在 t_{SLFT_j} 时刻结束时, $f_{\text{time}}(j, m, t)$ 取最大值 1;若 $f_{\text{time}}(j, m, t)$ 大于 1 时,说明后续作业必然引起延期,表示作业 j 以模式 m 在 t 时刻开始不可行,且时间因素评估函数的值越小越好.

2.3.2 资源因素评估函数

资源因素评估函数用于评价作业安排对资源约束的影响.对于一个局部进度计划, R_n 表示时刻 t 时资源 k 的消耗量; R_{ke} 表示时刻 e 时资源 k 的消耗量.项目中资源 k 的投入成本 I_k 可以表示为

$$I_k = \text{Max}R_{ke} \quad (15)$$

所有资源的总投入为 $A = \sum_k I_k$.假设一个局部进度计划(partial schedule),若安排新的作业 j 以模式 m 在 t 时刻开始,则需要的额外资源投入成本为

$$E_{\text{extra}}(j, m, t) = \sum_k \text{Max}(0, \text{Max}\{R_{ke} + r_{jmk} \mid t \leq e < t + d_{jm}\} - I_k) \quad (16)$$

假设此时的资源投入上限为 R_{RIL} ,则资源因素评估函数为

$$f_{\text{resource}}(j, m, t) = \frac{E_{\text{extra}}(j, m, t)}{R_{RIL} - A} \quad \forall j, \forall m \quad (17)$$

同样的,式(17)中分母用于标准化函数的最大值为 1.安排作业 j 以模式 m 在 t 时刻开始,若不增加额外的资源投入, $f_{\text{resource}}(j, m, t)$ 的值为零;若增加额外的资源投入, $f_{\text{resource}}(j, m, t)$ 的值增大,且在达到资源投入上限 RIL 时,取得最大值 1;若 $f_{\text{resource}}(j, m, t)$ 大于 1,表示此时资源约束被破坏,作业 j 以模式 m 在 t 时刻开始不可行.类似于时间因素评估函数,资源因素评估函数越小越好.

2.3.3 时间-资源综合评估函数

$f_{\text{time}}(j, m, t)$ 和 $f_{\text{resource}}(j, m, t)$ 分别表示安排活

动时面临的时间影响和资源影响. 将两个因素结合考虑, 作业 j 以模式 m 在 t 时刻开始的时间-资源综合评估函数为

$$f_{\text{total}}(j, m, t) = \begin{cases} 2 & \text{condition} \\ \frac{f_{\text{time}}(j, m, t) + f_{\text{resource}}(j, m, t)}{2} & \text{otherwise} \end{cases} \quad (18)$$

在式(18)中, condition 条件为 $1 < f_{\text{time}}(j, m, t)$ or $f_{\text{resource}}(j, m, t) > 1$ 表示存在约束被破坏. 若资源约束和工期约束均未被破坏, 则 $f_{\text{total}}(j, m, t)$ 取时间因素评估函数和资源因数评估函数的平均值, 易知此时 $f_{\text{total}}(j, m, t)$ 的取值范围是 $[0, 1]$; 若某一个约束被破坏, 即作业 j 以模式 m 在 t 时刻开始不可行时, 将 $f_{\text{total}}(j, m, t)$ 赋值为 2. 应用式(18), 选择 j, m, t 的步骤:

(1) 对每个待选择的作业, 选定它对应于最小综合评估函数值的模式和开始时间, 这表示若选择此作业时的最优选择.

(2) 对于所有确定了模式和开始时间的待选择作业, 选择最大综合评估函数值的作业. 这样, 综合评估函数值最大的活动最先排, 对后续作业的影响最下, 避免安排后续作业时使得解变差, 即

$$(j^*, m^*, t^*) = \arg \text{Max}_j (\arg \text{Min}_{m, t} f_{\text{total}}(j, m, t)) \quad (19)$$

在串行调度阶段, 进度生成机制按照每一步排入一个作业的方式, 将所有作业排入进度计划中. 其中, 每一步需要决策将要排入调度计划的作业和作业的执行模式与作业的开始时间. 应用式(19), 可以综合时间约束和资源约束, 选择合适的作业和作业的执行模式与作业的开始时间.

2.3.4 JRTS 算法步骤

JRTS 算法采用先设定一个较低的资源投入上限 R_{RIL} , 在此 R_{RIL} 下, 应用串行进度生成机制和由式(19)确定的优先级规则对拆分后的多个子项目进行调度求解. 若根据式(19)选定的 j, m, t 使得式(19)中 $f_{\text{total}}(j, m, t) > 1$, 即不可行时, 不断增加资源投入上限 R_{RIL} , 并重新进行调度求解, 直到求出可行的最小 R_{RIL} , 具体步骤如下:

(1) 初始化已排作业集合 $S_{\text{AS}} = \emptyset$, 可排作业集合 $S_{\text{CS}} = \emptyset$, 待排作业集合 $S_{\text{WS}} = J$, 初始化 R_n 均为零, 设定初始资源投入上限 $R_{\text{RIL}} = 0$.

(2) 更新可排作业集合, 调用式(19)选择 j, m, t .

(3) 判断是否可行. 若 $f_{\text{total}}(j, m, t) > 1$, 即不可

行时, $R_{\text{RIL}} = R_{\text{RIL}} + 1$, 并转入(1); 若 $f_{\text{total}}(j, m, t) \leq 1$, 则转入(4).

(4) 将作业 j 以模式 m 安排在 t 时刻开始, 并更新 R_k , 更新 S_{AS} , S_{CS} 和 S_{WS} .

(5) 若待排作业集合 $S_{\text{WS}} = \emptyset$, 表示作业已排完, 则转入(6); 否则, 转入(2).

(6) 输出资源投入最小值 $A = R_{\text{RIL}}$.

3 数据实验

选用 PSPLIB 标准算例库中的算例进行算法测试, 运用 PyCharm(Python 3.5.3) 开发环境编程, 仿真测试平台为 Intel Core i7-4790 处理器, 12G 内存. 现有文献中关于 MRIPSP-PS 的研究较少, 为了验证本文算法的有效性, 选择多模式资源首相项目调度问题 MRCPSP (Multi-Mode Resource-Constrained Project Scheduling Problem) 问题中常用的优先规则构建对比算法^[13], 其优先规则通常包含两类: ① 优先级规则用于选择每次要排的作业; ② 优先级规则用于选择作业的执行模式和开始时间.

3.1 基于优先级规则的启发式算法

基于优先级规则的对比较算法通过串行进度生成机制生成调度计划, 每次应用作业优先级规则选定要排的作业, 再调用作业模式和开始时间优先级规则将作业排入调度计划中. 具体规则如表 1 所示:

表 1 两类优先级规则

Tab.1 Two kinds of priority rules	
第 1 类规则	第 2 类规则
最小松弛时间 最晚完成时间	最小额外投入

表 1 中的规则中最小松弛时间规则表示 R_{MST} ; 最晚完成时间规则表示 R_{MLTT} ; 最小额外投入规则表示 R_{MEI} ; 同时上述三种规则可以组成 MST-MEI 和 MLTT-MEI 两种启发式算法, 本文将分别针对不同规模算例与两种对比算法比较.

3.2 数据对比

分别使用 PSPLIB 标准算例库中 J10, J20 和 J30 这 3 种不同规模的算例, 同时设定拆分数量 N 分别为 2 和 3, 对比求解质量. 在 PSBLIB 算例库中, 作业的资源消耗分为可更新资源和不可更新资源, 实验将所有资源都视为可更新资源. 在表 2~表 6 中, V_{JRTS} 一栏表示本文算法所得解, V_{MST} 一栏表示 MST-MEI 算法所得解, $G_1 = \frac{V_{\text{MST}} - V_{\text{JRTS}}}{V_{\text{JRTS}}} \times 100\%$,

V_{MLFT} 表示 MLTT-MEI 算法所得解, $G_2 = \frac{V_{MLFT} - V_{JRIS}}{V_{JRIS}} \times 100\%$.

表 2 J10 实验结果

Tab.2 The results of J10

算例	N=2					N=3				
	V_{JRIS}	V_{MST}	G_1	V_{MLFT}	G_2	V_{JRIS}	V_{MST}	G_1	V_{MLFT}	G_2
1	44	44	0.00	45	2.27	38	40	5.26	39	2.63
2	49	52	6.12	50	2.04	36	36	0.00	36	0.00
3	56	58	3.57	57	1.79	45	46	2.22	45	0.00
4	58	59	1.72	59	1.72	52	55	5.77	54	3.85
5	49	49	0.00	51	4.08	37	38	2.70	38	2.70
6	58	60	3.45	58	0.00	50	52	4.00	50	0.00
7	56	59	5.36	57	1.79	50	52	4.00	51	2.00
8	57	57	0.00	58	1.75	38	38	0.00	38	0.00
9	42	46	9.52	42	0.00	50	54	8.00	52	4.00
10	54	56	3.70	56	3.70	52	53	1.92	52	0.00
平均	52.3	54.0	3.35	53.3	1.91	44.8	46.4	3.57	45.5	1.56

表 3 J20 实验结果

Tab.3 The results of J20

算例	N=2					N=3				
	V_{JRIS}	V_{MST}	G_1	V_{MLFT}	G_2	V_{JRIS}	V_{MST}	G_1	V_{MLFT}	G_2
1	63	66	4.76	65	3.17	65	66	1.54	66	1.54
2	66	71	7.58	71	7.58	63	67	6.35	66	4.76
3	71	76	7.04	75	5.63	58	60	3.45	60	3.45
4	63	63	0.00	63	0.00	59	62	5.08	59	0.00
5	62	70	12.90	67	8.06	62	63	1.61	62	0.00
6	64	66	3.13	66	3.13	67	69	2.99	69	2.99
7	69	75	8.70	69	0.00	59	61	3.39	61	3.39
8	73	74	1.37	74	1.37	57	61	7.02	60	5.26
9	63	70	11.11	67	6.35	57	63	10.53	61	7.02
10	64	66	3.13	65	1.56	68	70	2.94	69	1.47
平均	65.8	69.7	5.93	68.2	3.69	61.5	64.2	4.39	63.3	2.93

为了在更大的规模下对比算法的有效性,在 PSPLIB 问题库的基础上,将 30 个作业的算例串联,分别构造出包含 60 个作业和 90 个作业的大规模算例,进一步对算法的性能进行对比,实验结果如表 5~

表 6 所示.

从表 2~表 4 的数据实验结果可以看出,在小规模的问题下,本文算法领先于 MST-MEI 算法和 MLTT-MEI 算法,说明本文算法可以发挥资源评价函数和时间评价函数的综合优势,使作业按照较优的方式安排.同时,两种对比算法,尤其是 MLTT-MEI 算法的结果在小规模问题中非常接近本文算法,这也说明本文选择最晚作业结束时间规则作为对比算法是正确的.而且针对不同的拆分数量 N ,实验结果显示本文算法都得到更优的解,证明了其稳健性.

表 4 J30 实验结果

Tab.4 The results of J30

算例	N=2					N=3				
	V_{JRIS}	V_{MST}	G_1	V_{MLFT}	G_2	V_{JRIS}	V_{MST}	G_1	V_{MLFT}	G_2
1	81	84	3.70	85	4.94	69	72	4.35	71	2.90
2	74	80	8.11	78	5.41	77	84	9.09	82	6.49
3	76	78	2.63	78	2.63	79	82	3.80	82	3.80
4	77	80	3.90	79	2.60	78	85	8.97	82	5.13
5	76	81	6.58	79	3.95	68	70	2.94	70	2.94
6	86	90	4.65	89	3.49	76	81	6.58	80	5.26
7	85	90	5.88	91	7.06	80	82	2.50	85	6.25
8	85	88	3.53	87	2.35	80	83	3.75	82	2.50
9	74	80	8.11	80	8.11	81	83	2.47	83	2.47
10	79	85	7.59	84	6.33	80	85	6.25	83	3.75
平均	79.3	83.6	5.42	83.0	4.67	76.8	80.7	5.08	80.0	4.17

在图 5 的图例中,MST_N2 和 MST_N3 分别表示在拆分个数分别为 2 和 3 时,每组算例平均领先对比算法的百分比.在问题规模从 10 增加到 30 时,本文算法的 GAP 从 2%左右提高到 4%以上,而且对比不同的对比算法都展现出有效的竞争力.同时,在不同拆分个数下,本文算法都能取得稳定的领先.在 60 和 90 个作业的大规模问题下,本文算法领先对比算法的 GAP 稳定在 4%左右,而且对比不同的

表 5 J60 实验结果

Tab.5 The results of J60

算例	N=2					N=3				
	V_{JRIS}	V_{MST}	G_1	V_{MLFT}	G_2	V_{JRIS}	V_{MST}	G_1	V_{MLFT}	G_2
1	103	107	3.88%	106	2.91%	90	96	6.67%	94	4.44%
2	96	105	9.38%	102	6.25%	83	85	2.41%	86	3.61%
3	89	93	4.49%	94	5.62%	90	96	6.67%	93	3.33%
4	123	132	7.32%	130	5.69%	105	108	2.86%	109	3.81%
5	104	110	5.77%	111	6.73%	89	93	4.49%	93	4.49%
6	121	126	4.13%	129	6.61%	110	115	4.55%	114	3.64%
7	91	96	5.49%	97	6.59%	95	99	4.21%	100	5.26%
8	123	127	3.25%	130	5.69%	117	121	3.42%	119	1.71%
9	100	109	9.00%	107	7.00%	94	99	5.32%	98	4.26%
10	116	124	6.90%	119	2.59%	111	113	1.80%	114	2.70%
平均	106.6	112.9	5.91%	112.5	5.53%	98.4	102.5	4.17%	102	3.66%

表 6 J90 实验结果
Tab.6 The results of J90

算例	N=2					N=3				
	V _{JRTS}	V _{MST}	G ₁ /%	V _{MLFT}	G ₂ /%	V _{JRTS}	V _{MST}	G ₁ /%	V _{MLFT}	G ₂ /%
1	106	109	2.83	108	1.89	121	128	5.79	126	4.13
2	130	137	5.38	133	2.31	126	130	3.17	129	2.38
3	128	136	6.25	135	5.47	102	110	7.84	108	5.88
4	115	120	4.35	120	4.35	108	111	2.78	111	2.78
5	120	124	3.33	123	2.50	125	130	4.00	128	2.40
6	113	119	5.31	115	1.77	115	122	6.09	119	3.48
7	118	124	5.08	120	1.69	112	116	3.57	116	3.57
8	106	110	3.77	108	1.89	103	110	6.80	109	5.83
9	126	131	3.97	133	5.56	111	115	3.60	114	2.70
10	131	138	5.34	138	5.34	109	113	3.67	113	3.67
平均	119.3	124.8	4.61	123.3	3.35	113.2	118.5	4.68	117.3	3.62

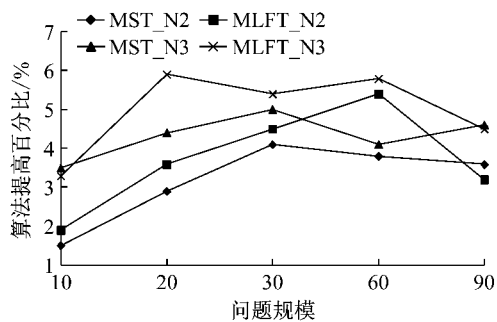


图 5 不同规模算法提高百分比

Fig.5 The increasing percentage of different scale

拆分个数和不同的对比算法,本文算法均能保持不低于 3% 的领先。

4 总结与展望

研究了飞机移动装配线作业调度的作业执行多模式问题,建立了包含项目拆分和多模式资源投入项目调度的数学模型,提出了合理的项目拆分策略和基于时间-资源综合评估规则的启发式算法。通过数据实验,PSS-JTRS 算法可以发挥资源评价函数和时间评价函数的优势,能够比其他启发式算法更好的对问题进行求解。在以后的研究中,可以考虑依次启发式算法为基础,进一步加入遗传算法、模拟退火算法等元启发式算法,使算法加入搜索机制,对飞机移动装配线作业调度问题进行更系统的优化。

参考文献:

[1] MASTOR A A. An experimental investigation and comparative evaluation of production line balancing techniques [J]. Management Science, 1970, 16(11): 728.
[2] MORING H. R. Minimizing costs of resource requirements in

project networks subject to a fixed completion time [J]. Operations Research, 1984, 32: 89.

- [3] DEMEULEMEESTER E. Minimizing resource availability costs in time-limited project networks [J]. Management Science, 1995, 41: 1590.
[4] RODRIGUES SB, YAMASHITA DS. An exact algorithm for minimizing resource availability costs in project scheduling [J]. European Journal of Operational Research, 2010, 206: 562.
[5] RANGASWAMY. Multiple resource planning and allocation in resource-constrained project networks [D]. Boulder: University of Colorado, 1999: 3037.
[6] KELLEY J. E. The critical path method: resources planning and scheduling [C]//Industrial Scheduling. Englewood Cliffs: Prentice-Hall, 1963: 347-365.
[7] KURTULUS I, Davis E. Multi-project scheduling: categorization of heuristic rules performance [J]. Management Science, 1982, 28: 161.
[8] BOCTOR. F. Some efficient multi-heuristic procedures for resource-constrained project scheduling [J]. European Journal of Operational Research, 1990, 49: 3.
[9] LAWRENCE. S, Morton. T. Resource-constrained multi-project scheduling with tardy costs: comparing myopic, bottleneck, and resource pricing heuristics [J]. European Journal of Operational Research, 1993, 64: 168.
[10] AFSHAR-NADJAFI, B. Multi-mode resource availability cost problem with recruitment and release dates for resources [J]. Applied Mathematical Modelling, 2014, 38(21-22), 5347.
[11] CHEN L, LI X, CAI Z. Heuristic methods for minimizing resource availability costs in multi-mode project scheduling [C]//Conference Proceedings-IEEE International Conference on Systems, Man and Cybernetics. Seoul: IEEE, 2012: 809-813.
[12] 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.
[13] KOLISCH R. Project scheduling under resource constraints [M]. Heidelberg: Physica-Verlag HD, 1995.