

考虑物料配送的飞机移动生产线调度问题优化

胡鑫铭, 陆志强*

(同济大学 机械与能源工程学院, 上海, 201804)

摘 要: 以飞机移动生产线为实际背景, 将作业装配过程调度抽象为资源受限项目调度问题并进行了扩展, 引入物料配送与线边存储决策, 以及相关能力约束等实际因素, 建立了以装配总工期最小化为目标的数学模型。针对模型, 设计了一种以遗传算法为框架的启发式算法, 其中结合了解生成算法和局部优化搜索算法。在遗传算法较优的全局搜索能力下, 通过 SCRDS 算法综合作业顺序、资源约束、配送能力、线边空间等因素联合决策作业的开始时间、物料配送时间和物料在线边空间的存放区域, 并提出了两作业间物料摆放位置调整的局部优化算法对作业开始时间和物料配送时间进行再优化, 进一步缩短了装配总工期。利用标准算例库进行了数值实验, 实验结果证明了模型与算法的有效性。

关 键 词: 飞机移动生产线; 物料配送; 遗传算法; SCRDS 算法; 局部优化

中图分类号: TP29

文献标识码: A

文章编号:

飞机移动装配生产线是丰田生产方式/精益生产理论在飞机制造业的应用, 对于降低飞机的装配成本, 提升装配效率, 缩短装配工期具有重要意义。

在飞机整个装配过程中, 移动生产线上的飞机以平稳的速度从一个工作站移动到下一个并缓慢通过整个装配线, 从而完成飞机所需的所有装配作业, 而移动装配线速度的调整, 可以设定装配线的不同生产节拍。为了保证生产线按需连续稳定生产, 装配作业所需的物料及其准时配送也成为移动生产线正常运作的关键环节之一, 通常在装配线旁会专门配置相关空间作为装配所需物料的存放区域, 通过小车运送的方式将物料按时定量配送至相应的线边区域。事实上, 飞机移动生产线上架飞机的任意作业调度安排, 不仅必须遵循工艺顺序约束及资源约束, 还需同时考虑物料配送计划与物料的线边存储, 为了获得更加有效可行的调度计划, 其作业调度决策需要高度协调装配作业活动与物料配送活动。本文把飞机移动生产线装配作业调度问题抽象为一类特殊的

资源受限项目调度问题 (Resource-Constrained Project Scheduling Problem, RCPSP)。在这一问题中, 各装配活动可以看作是项目的作业, 装配所需的人力、设备工具、线边存储空间及物料配送能力等视为项目所需资源。其与一般资源受限项目调度问题的主要区别在于问题中引入了物料配送因素, 问题的决策不仅需要优化决策各装配作业的开始时间, 还必须同时决策安排装配所需物料的配送时间与物料存放位置。因此, 可认为是一类集作业调度与物料配送调度问题于一体的联合决策问题, 本文把这一问题称之为考虑物料配送的飞机移动生产线调度问题 (Aircraft Moving Assembly Line Scheduling Problem Considering Material Delivery, AMALSP-CMD)。综上所述, 飞机移动生产线中装配作业数量大、作业间的装配顺序复杂、资源共享、物料配送及存储能力受限等特点对装配作业的调度造成了极大困难, 本文正是以此为背景, 提出了 AMALSP-CMD 问题并对其进行优化研究。

现有文献已经对经典的 RCPSP 问题进行了

较为充分的研究,目前的求解方法基本可以分为精确算法、启发式算法和元启发式算法三类。Chaleshtarti 和 Shadrokh^[1]采用分枝定界法求解目标为最小化工期的累积资源受限项目调度问题(RCPSP-CU)。Berthold 等^[2]整合了整数规划、约束规划和满意度测试法,提出了一种分支定界机制,并对下界的确定做了优化。RCPSP 问题已被证明是一个 NP-hard 问题^[3],启发式算法的效率相对精确算法较高,在求解大规模问题时更加有效。Tormos 和 Lova^[4]在 RCPSP 中引入随机抽样和混合多路径算法对作业进行优化,同时证明了这类算法的高效性。元启发式算法是对启发式算法的改进,主要包括禁忌搜索算法、模拟退火算法、遗传算法等。Bukata 等^[5]采用并行禁忌搜索算法求解 RCPSP 最小工期。Bouleimen 和 Lecocq^[6]在传统的模拟退火算法中考虑解空间的特异性重新设计了搜索机制,对单模式和多模式的 RCPSP 问题进行求解。Alcaraz 和 Maroto^[7]使用遗传算法对项目调度中的资源进行分配,并提出了新的交叉算子,在广泛的样本试验中取得了良好的效果。Hartmann^[8]提出了一类自适应遗传算法,通过任务列表对染色体进行编码,同时在染色体上增加了一个判定串并行解码方式的基因,数值实验证明了这类算法可以通过有效选择合适的解码方式产生较优解。Merkle 等^[9]对蚁群算法进行优化,使用了双信息素组合评价方法搜索更优的项目调度计划。Kumar 和 Vidyarthi^[10]改进了传统的粒子群算法对 RCPSP 问题求解,提出了一种有效的粒子生成算子,避免了传统算法中粒子由于位置与速度的更新出现失效,提高了算法的效率。而针对飞机移动生产线的 RCPSP 问题,Wang 和 Lu^[11]分析了飞机移动装配线的多重约束,建立了整数规划模型并提出了基于 WRST 规则的启发式算法。Zheng 和 Xi^[12]对 3 种作业人数选择规则和 9 种作业优先顺序决策规则进行组合,构成了 27 种启发式算法,通过比较这些启发式算法在测试问题集中的表现选出最有效的解决此问题的方法。上述 RCPSP 问题的研究对于本文所研究的 AMALSP-CMD 问题中作业调度的建模与算法设计具有重要参考意义。而对于生产线的物料配送问题,Ge 等^[13]针对机械产品总装过程中物料的 JIT 配送问题进行了研究,把产品生产计划和节拍时间作为已知参数,考虑了配送设备的容积约束和工位线旁库存的面积限制,

建立了多目标数学模型并采用了混合粒子群算法求解。Fathi 等^[14-15]对汽车装配线物料配送中的多载量小车(Tow train)调度问题进行了研究,构建了最小化多载量小车出行次数与线边库存成本的多目标数学模型并设计了相应的优化算法。

综上所述,目前的研究大多将装配系统和物料配送系统拆分为两个相对独立的系统,对线边物料配送的研究大部分都是基于已制定好的生产计划,但实际生产线上常出现由于物料配送能力受限等原因导致装配作业中断的情况。事实上,生产线中各装配作业的开始时间与物料配送时间等因素是相互影响的,通过联合决策可以更有效地为作业装配和物料配送提供协同性计划,保证生产线平稳运行,具有较高的理论研究意义。Khayat 等^[16]在对车间内的作业调度问题的研究中综合考虑了机器资源与搬运小车的数量约束,建立了最小化作业加工完成时间为目标函数的数学规划模型并通过 CPLEX 软件求解。Aguirre 等^[17]对半导体制造中的加工与搬运问题进行了联合建模并设计了基于混合整数规划的顺序启发式算法。现阶段我国飞机装配车间的物料配送方式还停留在传统的“面向服务”模式,即工人根据装配计划到库房申请领料配送,然后进行装配作业,这种被动的物料供应模式容易导致物料管理混乱并产生生产浪费、装配延误等问题。因此采用主动的“面向装配”模式并将物料配送与装配调度相结合综合制订生产计划具有实际应用价值。本文所提出 AMALSP-CMD 问题,基于实际生产背景,综合考虑了飞机装配过程的作业顺序约束、资源约束、物料配送能力上限和物料在线边空间的存放等因素,建立了以总工期最小化为目标函数的数学模型,并提出了相应的算法对作业开始执行时间、物料配送时间和线边空间存储位置进行联合决策。

1 问题描述及数学模型

1.1 问题描述

飞机移动生产线调度问题,是通过优化决策各作业的执行时间、物料配送时间及其相应的线边存放空间,以达到最小化装配工期的目标。AMALSP-CMD 问题的假设与说明如下:

1) 飞机装配作业项目包含 n 项作业, $J = \{1, 2, \dots, n\}$, 任意 $j \in J$ 的执行工期为 t_j , 开始时间为 T_j^s , 完成时间 $T_j^f = T_j^s + t_j$ 。 P_j 表示作业 j

的紧前作业集合, 任意作业在其紧前作业全部完成后才能执行。增加作业 0 和作业 $n+1$ 表示虚作业, 虚作业执行时间为 0, 且不占用任何资源。

2) 可更新资源为 $R=\{1,2,\dots,K\}$, 任意时刻第 k 种资源总量为 R_k , 单位时间内作业 j 消耗资源 k 的数量用 r_{jk} 表示。

3) 将时间离散化, 时间集合为 $D=\{1,2,\dots,z\}$,

z 表示所有作业操作时间总和, 任意 $d \in D$ 表示离散的时间节点。

4) 将飞机上的作业空间离散化, 如图 1 所示, 对于任意一架飞机, 以飞机机尾作为零参考点, ψ 表示飞机的长度。飞机上的作业空间集合为 $\Gamma=\{1,2,\dots,\psi\}$, 任意 $l_j \in \Gamma$ 表示作业 j 在飞机上的固定装配位置。

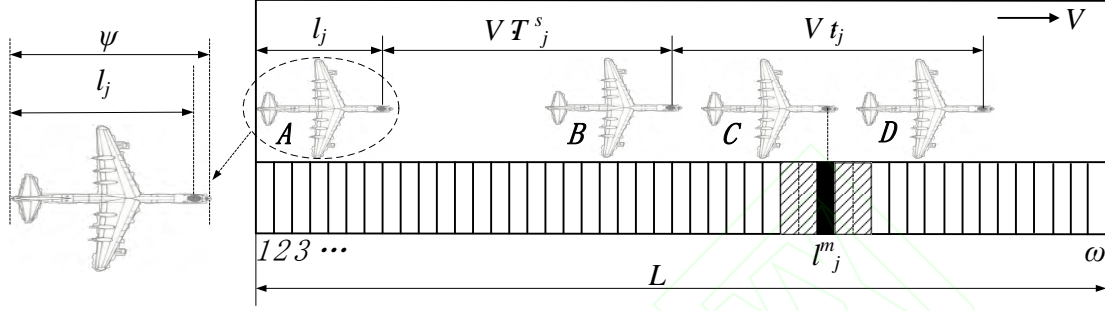


图 1 飞机移动生产线作业装配点及物料摆放位置示意图

Fig.1 Schematic diagram of positions of assembly and material storage on aircraft moving assembly line

5) 假定飞机移动生产线的移动速度为 V , 作业 j 在移动生产线上的装配位置和物料摆放位置可通过图 1 表示。离散化的线边空间集合为 $L=\{1,2,\dots,\omega\}$, $l \in L$ 。对于任意一架飞机, 图中状态 A 表示飞机进入移动生产线的起始位置。在时刻 T_j^s 飞机移动到状态 B 位置时作业 j 开始装配, 经过 t_j 时间段飞机移动到状态 D 所示位置时作业 j 装配完成。作业 j 开始执行到完成时间段内飞机在移动生产线上移动距离的中间位置, 即状态 C 所示位置, 定义该位置作业 j 的装配点所对应的线边空间位置为该作业所需物料的中心摆放位置 l_j^m , 用图中黑色部分表示。任意 $l_j^m \in L$, $l_j^m = \lceil l_j + V \cdot T_j^s + V \cdot t_j / 2 \rceil$, $\lceil \cdot \rceil$ 表示向上取整。

6) 线边空间的分配以将任意作业装配所需物料存放在距离装配中心位置较近区域为基本原则, 可允许物料占用 l_j^m 左右两侧各 m 个单位的线边空间, 用图 1 中阴影部分表示。

7) 飞机装配作业 j 所需的物料主要包括四类: 形状不规则、数量少的大型结构件; 需求量大、体积小的通用标准件; 体积中等、数量多的装配件和复杂昂贵的系统件。大型结构件由于线边空间容量不足, 采用标准料架直接进行配送; 通用标准件 (如装配所需的螺母等紧固件) 由于需求量较高且体积小, 通常在各工位线边单独划定存放区域, 通过定期补货的方式配送, 因此本

文重点考虑装配件和系统件两类物料的配送与存放。装配件和系统件均使用标准料箱由小车配送至线边空间并在装配作业结束后统一回收空料箱的方式进行供应, 配送完成时间 $T_j^p \leq T_j^s$, 配送提前期 $t_j^p \leq p_{\max}$, 物料占用线边空间时长为 $T_j^o = T_j^f - T_j^p$ 。作业 j 对物料的需求为 v_j 。

8) 每个单位线边空间容量为 S_u , 假定任意时刻物料综合配送能力的上限为 D_{\max} , 记项目总工期为 T 。

本文研究问题的三类决策变量:

x_{jd} : 0,1 变量, 当其为 1 时表示第 j 个作业在时刻 d 开始执行, 否则为 0。

y_{jd} : 0,1 变量, 当其为 1 时表示第 j 个作业所需物料在时刻 d 完成配送, 否则为 0。

v_{jdl} : 非负整数变量, 表示作业 j 在时刻 d 在线边空间 l 存放的料箱数量, 取值范围为 $\{0, 1, 2S_u\}$ 。

1.2 数学模型

AMALSP-CMD 的数学模型如下:

$$\min Z = T \quad (1)$$

s.t.

$$\sum_{d=1}^z x_{jd} = 1 \quad \forall j \in J \quad (2)$$

$$\sum_{d=1}^z y_{jd} = 1 \quad \forall j \in J \quad (3)$$

$$\sum_{d=1}^z d \cdot x_{jd} \geq \sum_{d=1}^z (d + t_a) \cdot x_{ad} \quad a \in P_j, \forall j \in J \quad (4)$$

$$\sum_{j=1}^n r_{jk} \cdot \sum_{\tau=d-t_j+1}^d x_{j\tau} \leq R_k \quad \forall k \in K, \forall d \in D \quad (5)$$

$$\sum_{j=1}^n v_{jdl} \leq S_u \quad \forall d \in D, \forall l \in L \quad (6)$$

$$t_j^p = \sum_{d=1}^z (x_{jd} - y_{jd}) \cdot d \quad \forall j \in J \quad (7)$$

$$0 \leq t_j^p \leq p \quad \forall j \in J \quad (8)$$

$$l_j^m = \left\lceil l_j + V \cdot \sum_{d=1}^z x_{jd} \cdot d + V \cdot t_j / 2 \right\rceil \quad \forall j \in J \quad (9)$$

$$\sum_{l_j^m-m}^{l_j^m+m} v_{jdl} = v_j \cdot \sum_{\tau=d-t_j-t_j^p+1}^d y_{j\tau} \quad \forall j \in J, \forall d \in D \quad (10)$$

$$v_{j\tau_1 l} = v_{j\tau_2 l} \quad \forall j \in J, \forall l \in L, \forall \tau_1, \tau_2 \in T_j^0 \quad (11)$$

$$\sum_{j=1}^n (v_j \cdot y_{jd}) \leq D_{\max} \quad \forall d \in D \quad (12)$$

$$d \cdot \sum_{d=1}^z x_{jd} + t_j \leq T \quad \forall j \in J \quad (13)$$

$$x_{jd} \in \{0,1\}, \quad y_{jd} \in \{0,1\} \quad \forall j \in J, \forall d \in i \quad (14)$$

$$v_{jdl} = \{0,1,2 \dots S_u\} \quad \forall j \in J, \forall d \in D, \forall l \in L \quad (15)$$

式(1)为目标函数,表示要求飞机装配总工期最小化;式(2)表示作业一旦开始不允许抢占,必须连续执行直到完成;式(3)表示任意作业装配所需的物料必须一次性配送至线边空间;式(4)定义了作业间的时序约束关系,一项作业必须在所有的紧前作业全部完成后才能执行;式(5)为资源约束,任意时刻所有执行的作业对任意资源的需求不得超过该资源的供应上限;式(6)表示任意单位线边空间存放的物料不得超过其上限;式(7)定义了作业所需物料的配送提前期;式(8)表示任意作业物料的配送时间应早于其装配时间且配送提前期不超过上限值;式(9)定义了某项作业的物料摆放中心位置与装配位置及时间的线性关系;式(10)和(11)表示任意作业的物料应存放在该作业装配时对应线边空间的中间摆放位置且不得超出左右一定范围;式(12)表示任意时刻所有安排配送的物料数量不得超过物料综合配送能力上限;式(13)定义了项目的完成时间;式(14)、(15)定义了决策变量的可行域。

2 算法设计

针对 AMALSP-CMD 的特点,本文设计了一类以遗传算法为框架的启发式算法(THBG),其

中结合了解生成算法和局部优化搜索算法。该算法的基本框架与主要思想为:通过遗传算法较强的全局搜索能力搜寻较优的作业执行顺序,为每一条包含作业执行顺序的染色体设计有效的算法(SCRDS)对资源、配送能力和线边空间进行合理分配,从而获得各项作业的开始执行时间、物料配送时间和物料在线边空间的存放位置的初始解。在此基础上,基于上述初始解调度结果,通过考虑两作业间物料存放空间调整的局部优化算法对线边空间进行再分配,使得更多的作业能并行执行,进一步缩短装配总工期。以下分别对遗传算法框架、SCRDS 生成算法以及局部优化算法进行阐述。

2.1 遗传算法框架

遗传算法采用作业列表表达方式(Job List Representation)编码,染色体 $I = \{\lambda\}$ 由作业列表 λ 组成,每一个基因位表示一个作业序号,则每一条染色体代表一个作业顺序列表。为了提高算法效率,本文借鉴了文献[8]中提出的针对作业列表部分染色体的初始化方式与交叉、变异算子,在染色体内部镶嵌了作业间的紧前紧后关系确保初始解及产生的后代染色体均为可行解。染色体的初始化方式如图 2 所示。

图 2(a)为一包含虚作业共 8 个作业的项目网络图,首先安排虚作业 0 进入作业列表,将未安排的作业中紧前作业已全部执行完成的作业加入待选集合 H ,此时待选作业集合 $H = \{1,2,3\}$;然后在 H 中等概率随机挑选一个作业为下一个执行作业,例如选择作业 3;此时更新 $H = \{2,3\}$,如图 2(b)所示。下一步在新的候选作业集合 H 中再随机挑选一个作业进行安排,依次直至所有作业被安排完。染色体的交叉方式为子代染色体保留部分父代 I^F 的染色体,交换部分的染色体继承父代 I^M 剔除 I^F 中重复的作业后按序保留下的作业列表,如将染色体 $\lambda^F = (0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7)$ 与 $\lambda^M = (0 \ 2 \ 3 \ 1 \ 5 \ 4 \ 6 \ 7)$ 第 4 到 6 基因位的片段进行交叉得到子代染色体 $\lambda^S = (0 \ 1 \ 2 \ 3 \ 5 \ 4 \ 6 \ 7)$ 与 $\lambda^D = (0 \ 2 \ 3 \ 1 \ 4 \ 5 \ 6 \ 7)$ 。变异算子通过交换或调整同一候选作业集合 H 中的作业顺序完成,例如对图 2(a)候选集合 $H = \{1,2,3\}$ 中的作业执行顺序进行重组将父代染色体 $\lambda = (0 \ 1 \ 2)$ 变异为 $\lambda_m = (0 \ 2 \ 3)$ 。

设定遗传算法的种群大小为 M ,迭代次数为

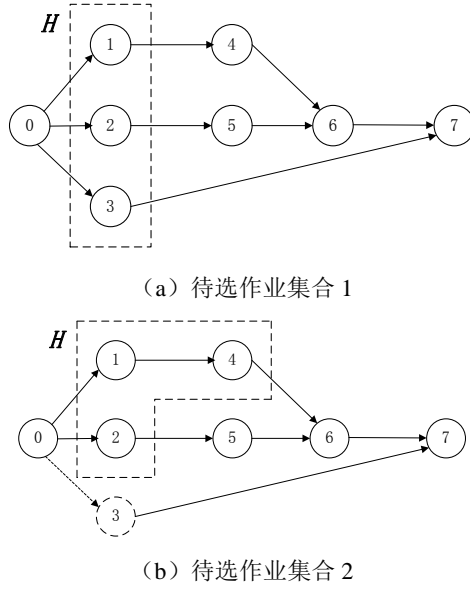


图2 染色体初始化示意图

Fig.2 Schematic diagram of chromosome initialization

N , 适应度函数 fit 为项目工期的倒数, λ 表示当前代数。算法框架如图 3 所示。其中 SCRDS 算法和局部优化算法分别在 2.2 和 2.3 中详细介绍。

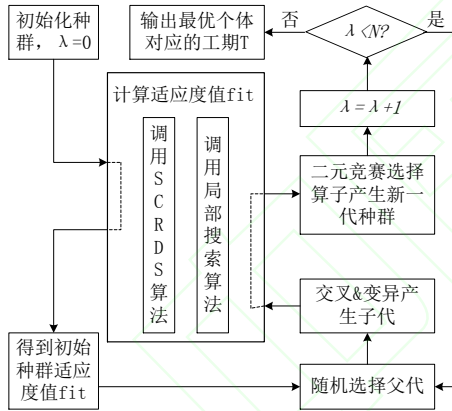


图3 遗传算法框架

Fig.3 Genetic algorithm framework

2.2 基于作业顺序列表的解生成算法 (SCRDS)

在作业执行顺序确定的情况下, 综合考虑作业的资源约束、物料配送能力约束和线边空间容量约束为各项作业安排开始执行时间、物料配送时间和物料存放空间。对于列表中的每一个作业, 采用 JIT 配送方式。算法的设计以串行进度生成机制为核心, 依次判定选定作业的各类资源、配送能力和线边空间容量在其紧前作业最晚完成时刻是否满足。在资源满足条件下, 若在给定的最大配送提前期内某时刻配送能力满足且线边空间容量充足, 可通过提前配送策略将该作业所需物

料早于装配开始时间配送并以暂存的形式存放在相应线边空间处, 此时可以得到作业的开始执行时间, 物料配送时间以及物料相应的线边存放位置。任意资源、配送能力或线边空间容量不足均将作业开始时间延期一单位执行。SCRDS 算法的具体步骤如下:

Step1: 获取作业安排序列 $\{j_1, j_2 \dots j_n\}$, 令 $i=1, k=1$ 。

Step2: 选取作业 j_i , 将其紧前作业的最晚完成时间 $\max\{T_{pj}^F\}$ 作为拟开始安排时间 T_j^{ST} , 则 j_i 的结束时间 $T_j^{FT} = T_j^{ST} + t_j$, 装配所需物料拟配送时间 $T_j^{DT} = T_j^{ST}$ 。

Step3: 若 j_i 满足当 $d \in [T_j^{ST}, T_j^{FT}]$, $r_{jk} \leq R_k$, 转 Step4, 否则转 Step11。

Step4: 若 j_i 所需物料满足 $d = T_j^{ST}$, $v_j \leq D_d^C$, 转 Step6, 否则转 Step5。

Step5: 作业 j_i 的物料配送时间 $T_j^{DT} = T_j^{DT} - 1$, 若 $T_j^{ST} - T_j^{DT} \leq p_{\max}$ 且 $T_j^{DT} \geq 0$, 转 Step4, 否则转 Step11。

Step6: 计算时刻 T_j^{ST} 作业 j_i 的物料摆放中心位置 l_j^m , 标记为 l_c 。若满足 $d \in [T_j^{DT}, T_j^{FT}]$, $v_j \leq v_{l_c}$, 则将 j_i 的物料存放于线边单元 l_c 中, $v_{l_c} = v_{l_c} - v_j$, 转 Step12, 否则先将空间 l_c 放满, $v_j = v_j - v_{l_c}$, 转 Step7。

Step7: 从 l_c 向左侧线边空间搜索 k 个单元, 若 $d \in [T_j^{DT}, T_j^{FT}]$, $v_j \leq v_{l_c-k}$, 则将作业 j_i 的剩余物料存放在线边单元 $l_c - k$ 内, $v_{l_c-k} = v_{l_c-k} - v_j$, 转 Step12, 否则转 Step8。

Step8: 从 l_c 向右侧线边空间搜索 k 个单元, 若 $d \in [T_j^{DT}, T_j^{FT}]$, $v_j \leq v_{l_c+k}$, 则将作业 j_i 的剩余物料存放在线边单元 $l_c + k$ 内, $v_{l_c+k} = v_{l_c+k} - v_j$, 转 Step12, 否则转 Step9。

Step9: 若 $d \in [T_j^{DT}, T_j^{FT}]$, $v_{l_c-k} + v_{l_c+k} \geq v_j$, 则将作业 j_i 物料先放满线边 $l_c - k$ 的空间, $v_j = v_j - v_{l_c-k}$, 余下物料存放在 $l_c + k$ 中, $v_{l_c+k} = v_{l_c+k} - v_j$, 转 Step12, 否则, 先将 $l_c - k$ 和 $l_c + k$ 空间放满, $v_j = v_j - (v_{l_c-k} + v_{l_c+k})$, 转 Step10。

Step10: 若 $k < m$, $k = k + 1$, 转 Step7, 否则转 Step11。

Step11: $T_j^{ST} = T_j^{ST} + 1$, $T_j^{DT} = T_j^{ST}$, 转 Step3。

Step12: 为作业 j_i 安排开始执行时间 $T_j^S = T_j^{ST}$, 物料配送时间 $T_j^D = T_j^{DT}$ 。

Step13: 更新资源 R_k 、配送能力 D^C 和线边空间容量 v_l 。

Step14: 若 $j_i = j_n$, 转 Step15, 否则 $i = i + 1$, 转 Step2。

Step15: 输出调度计划 S , 作业 j_n 的完成时间即为项目工期 D , 算法结束。

2.3 基于物料摆放位置调整的局部优化搜索算法

在 SCRDS 算法安排作业物料摆放位置的过程中, 为了使作业尽早开始执行, 在满足资源约束和配送能力约束的前提下, 作业会最大限度地占用周边的空间。因此可能会出现作业 i 占用了作业 j 的线边空间, 导致作业 j 所需物料没有足够的存放空间必须延后执行的情况发生。本文设计了针对此类情况的两作业物料摆放位置调整的局部优化算法, 通过撤销作业 i 对作业 j 物料摆

放位置的占用或改变作业 i 物料的摆放位置, 可能使得作业 j 能够提前执行, 缩短项目工期。局部优化的机理如图 4 所示。

对于一个包含 7 作业的项目, l 轴表示线边空间单元, t 表示时间轴, S_u 表示单位线边空间的最大容量, 图中方块代表作业的物料需求量。假定允许物料存放在中心摆放位置左右各一格的线边空间单元内, 对于作业 3 的物料 (设中心摆放位置 $l_j^m = 2$), 若将其拆分为两部分分开存放于线边单元 1 和单元 3 中, 则作业 4 在不延期的情况下所需物料在线边单元 3 中有足够的空间可以存放。因此作业 4 和作业 3 可以并行执行, 同时后续作业也能提前执行缩短整个项目的工期。

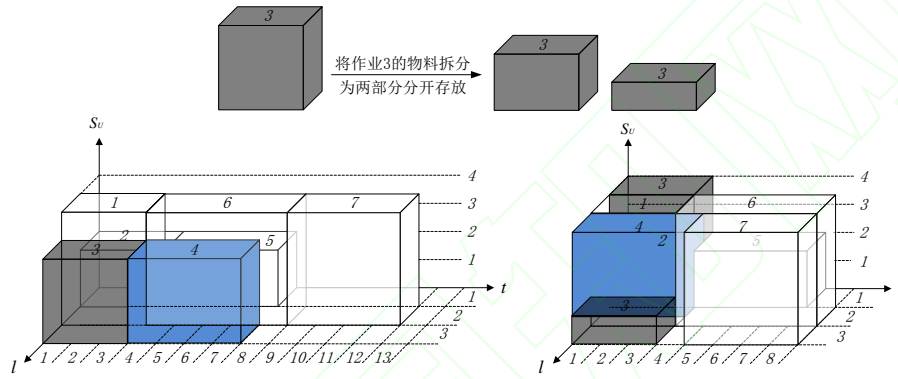


图 4 基于物料摆放位置调整的局部优化机理

Fig.4 Local optimization mechanism based on material positions adjusting

假定作业 i 的开始时间为 T_i^S , 完成时间为 T_i^F , 物料摆放中心位置为 l_i^m , 作业 j 的开始时间为 T_j^S , 完成时间为 T_j^F , 物料摆放中心位置为 l_j^m , 所有作业均允许物料占用摆放中心位置左右两侧各 m 个单位的线边空间。若作业 j 没有被延期执行, 作业 i 开始时间为 $T_i^{S'}$, 完成时间为 $T_i^{F'}$, 作业 j 开始时间为 $T_j^{S'}$, 完成时间为 $T_j^{F'}$ 。

性质: 在资源约束和配送能力满足的前提下, 若作业 i 和作业 j 满足以下四个条件, 则采用局部算法可以优化项目工期。

(1) $l_j^m \in [l_i^m - m, l_i^m + m]$ 且 $i \notin P_j$, 表示作业 i 与作业 j 的物料线边摆放空间重叠且作业 i 不是作业 j 的紧前作业。

(2) $T_j^S > \max\{T_i^{F'}, T_j^F\}$, 表示作业 j 被延期执行。

(3) $T_i^{S'} \leq T_j^{S'} \leq T_i^{F'}$, 表示若不存在延期, 作业 i 和作业 j 之间可以存在时间上的重叠, 有并行执行的可能。

(4) 当 $T_i^{S'} \leq t \leq \max\{T_i^{F'}, T_j^F\}$, $S_{left}[l_i^m - m, l_i^m + m] - S_{uj}^m \geq S_{left}[l_i^m - m, l_i^m + m] \cap S_{left}[l_j^m - m, l_j^m + m]$,

表示可以通过调整部分作业 i 的物料摆放位置使得作业 j 的物料能够在不拖期的情况下有足够的空间存放。

证明:

(1) 对于作业 $1 \sim i-1$, 其开始和结束时间不受影响。

(2) 对于作业 i 与作业 j 之间的作业, 由于只移动了作业 i 和作业 j 的物料, 对其它作业的物料摆放没有任何改变, 不会影响其开始和完成时间。

(3) 对于作业 $j+1 \sim n$, 由于 $T_j^{F'} \leq T_j^F$, 因此 $T_{j+1}^{F'} \leq T_{j+1}^{F'} \cdots T_n^{F'} \leq T_n^F$, 使得总工期必定不会增加。

局部优化算法的具体步骤如下:

Step1: 获取作业安排序列 $\{j_1, j_2 \cdots j_n\}$, 记优化前调度结果为 S , 工期为 D 。

Step2: 令 $i=1$, $k=2$, 选择初始作业 1 为 $j_i = j_1$, 初始作业 2 为 $j_k = j_2$ 。

Step3: 若 j_i 和 j_k 满足可执行局部优化的四个

条件, 则标记 j_i 为作业 1, j_k 为作业 2, 否则转 Step7。

Step4: 撤销 j_i 、 j_k 及 $j_{k+1} \sim j_n$ 的调度计划, 并将作业 1 的部分物料移至 $l_i - l_i \cap l_j$ 的线边空间单元, 将作业 2 的物料存放在 $l_i \cap l_j$ 线边空间单元。

Step5: 继续执行原来的调度计划, 将作业 $j_{k+1} \sim j_n$ 全部安排完, 得到新的调度计划 S' , 工期为 D' 。

Step6: 若 $D' < D$, 转入 Step7; 否则撤销 Step4 和 Step5 并转 Step7。

Step7: 若 $i \leq n$ 且 $k < n$, $k = k + 1$ 并转 Step3; 若 $i < n$, $k = n$, 则 $i = i + 1$, $k = i + 1$ 并转 Step3; 若 $i = n$ 且 $k = n$, 转 Step8。

Step8: 输出新的调度计划 S' 和新的项目工期 D' , 算法结束。

3 数值实验

本文基于 PSPLIB 问题库中的算例, 根据 AMALSP-CMD 特点对数据进行扩充, 为作业数量为 10、20、30、60、90、120、480 和 1200

八类规模的数据, 分别生成 10 组算例。其中作业间的顺序约束、作业操作时间和所需各类资源量使用 PSPLIB 问题库中的原数据, 同时为算例中的每个作业 j 随机生成 $l_j \in (1, 8)$ 的装配位置和 $v_j \in (5, 10)$ 的物料需求量, 并设定飞机移动速度 $V = 0.1$, 物料可占用左右两侧各 2 个单位的线边空间, 单位时间综合配送能力上限 $D_{\max} = 25$, 允许物料最大配送提前期 $p_{\max} = 3$ 。通过 C# (VS2010) 对算法编程进行测试与分析, 实验平台为 Intel i7-4790 处理器, 3.60GHz 主频, 8G 内存。

3.1 与 CPLEX 结果对比

为了验证 THBG 算法的精确性, 通过 CPLEX 软件对数学模型进行求解, 将求得的解与 THBG 算法得到的结果进行对比。由于 CPLEX 不适用于求解大规模算例, 因此表 1 给出了作业规模分别为 10、20、30 各 10 组不同算例的求解结果。表中 Dur 表示项目的工期, T 表示 CPU 求解时间, GAP 表示 THBG 算法得到的工期与 CPLEX 求得精确解的精度偏差, 计算公式为

$$GAP = \frac{Dur_{(THBG)} - Dur_{(CPLEX)}}{Dur_{(CPLEX)}} \times 100\%。$$

表 1 小规模算例数值实验结果

Table 1 Numerical experiment results of small-scale examples

算例	10job			20job			30job		
	Dur/T	CPLEX	THBG	GAP	CPLEX	THBG	GAP	CPLEX	THBG
1	16/0.58	16/0.54	0.00%	24/66.67	24/0.83	0.00%	43/810	43/1.52	0.00%
2	23/1.43	23/0.53	0.00%	24/6.14	24/0.83	0.00%	53/677	53/1.54	0.00%
3	13/1.61	13/0.53	0.00%	33/126.88	35/0.83	5.71%	37/3478	37/1.46	0.00%
4	34/2.77	34/0.52	0.00%	36/23.06	36/0.84	0.00%	41/9478	42/1.58	2.44%
5	20/1.11	20/0.50	0.00%	24/13.29	24/0.83	0.00%	46/25007	47/1.49	2.17%
6	31/0.93	31/0.54	0.00%	18/3.63	18/0.85	0.00%	42/3811	42/1.71	0.00%
7	20/0.92	20/0.52	0.00%	24/21.36	24/0.84	0.00%	61/14967	62/1.48	1.64%
8	22/1.87	22/0.52	0.00%	27/7.28	27/0.82	0.00%	LB=52/—	57/1.47	9.62%*
9	22/0.58	22/0.49	0.00%	22/3.58	22/0.83	0.00%	LB=51/—	57/1.48	11.76%*
10	20/1.05	20/0.51	0.00%	22/17.50	23/0.85	4.35%	61/32927	61/1.50	0.00%
AVG	22.1/1.29	22.1/0.52	0.00%	25.4/28.94	25.7/0.83	1.01%	48/11394	50.1/1.52	0.78%

注: CPLEX 栏中数据, 如 16/0.58 分别表示 CPLEX 求得的精确解和 CPU 时间, LB=52/—表示 CPLEX 在可接受时间(36000s)内无法求出问题可行解, 记录其低界值 LB=52; THBG 栏中数据, 如 16/0.54 分别表示 THBG 算法得到的解和 CPU 时间; GAP 栏带 “*” 表示 CPLEX 得不到可行解时用低界计算 GAP 值。

通过与 CPLEX 实验对比可以发现在小规模问题中, THBG 算法在大多数算例中都能获得较好的解。在 10job 作业规模算例下均能得到最优解, 当作业规模为 20 和 30 时, 与最优

解差值的均值分别为 1.01% 和 0.78%, 证明了 THBG 算法的有效性。随着作业规模增加, CPLEX 求解所需的时间迅速上升, 在 30job 作业规模下, 平均计算时间达到了 11394s, 甚至

无法在可接受时间内得到理想的低界。而 THBG 算法不仅能在较短时间内得到优质解，且求解时间稳定，说明 THBG 算法在求解时间上也具有巨大的优势。

3.2 大规模数据算法效果对比

为了测试本文提出的算法在求解大规模数据的效果，选取作业规模为 30、60、90、120、480 和 1200 的算例，并为每个算例增加三类对比实验，R-NMO 表示作业物料只能存放在摆放中心位置不允许占用两侧线边空间规则，H-TD 表示传统的先安排生产计划后考虑物流配送的两阶段决策算法，H-SCRDS 表示基于 SCRDS 的启发式算法。其中，H-TD 算法的具

体规则为：第一阶段：在不考虑物料配送能力与存储决策的情况下仅针对资源约束和作业间的顺序约束做出生产作业计划。第二阶段：对第一阶段做出的生产作业计划进行物流配送的安排。这类决策方式可能会出现厂内物流配送能力不足或线边空间容量无法满足原有生产作业计划的情况发生，此时需要将部分作业延期执行得到新的生产作业计划。在其余条件相同情况下，将上述三种算法的解与本文设计的 THBG 算法得到的项目工期进行对比，数值实验结果如表 2 和图 5 所示。表中 Dur 表示各个数据规模下 10 组算例得到结果的均值，GAP 表示不同算法得到工期结果的差值百分比。

表 2 不同算法工期结果和优化比例

Table 2 Durations and optimal proportion of different algorithms

Job Size	Dur				GAP		
	R-NMO	H-TD	H-SCRDS	THBG	GAP _{R-NMO}	GAP _{H-TD}	GAP _{L-SCRDS}
	(A)	(B)	(C)	(D)	(A-D)/D	(B-D)/D	(C-D)/D
30	54.86	56.24	50.17	50.10	9.79%	12.38%	0.14%
60	74.70	73.45	61.89	61.40	21.81%	19.19%	0.80%
90	91.70	93.65	76.03	75.00	22.80%	23.78%	1.37%
120	137.30	156.05	111.53	108.90	26.48%	41.75%	2.42%
480	395.48	393.37	305.24	296.37	33.44%	32.73%	2.99%
1200	981.50	1033.62	739.00	709.44	38.35%	45.70%	4.17%

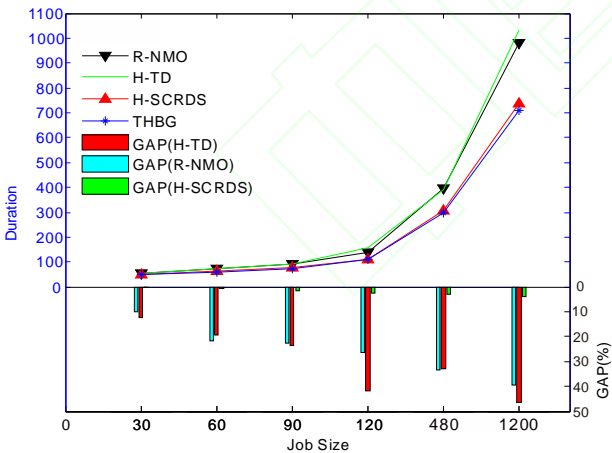


图 5 不同算法计算工期结果比较

Fig.5 Comparison of durations of different algorithms

对上述数值实验结果分析可以得出以下结论：

1) 本文设计的 SCRDS 算法相比于物料不能占用两侧线边空间的规则对项目工期结果有很大的优化，SCRDS 算法对飞机移动生产线边空间进行了有效地分配与利用，提高了线边空间的使用

效率。在项目作业规模为 30 时，相比于 R-NMO 规则所得结果的优化比例为 9.79%，当项目的作业规模为 60、90 和 120 时，优化比例分别为 21.81%、22.80%和 26.48%，均超过 20%，作业规模为 1200 时，平均优化比例可以达到 38.35%。

2) THBG 算法在各个规模的问题上得到的结果都优于 H-TD 算法得到的结果。当作业规模为 1200 时，THBG 算法得到的结果优化比例最高达到了 70.70%，均值达到 45.70%，说明将 AMALSP-CMD 中的资源、物料配送能力等约束综合考虑对装配计划进行决策相比传统的两阶段调度方法具有优势。

3) 问题规模较小时，局部优化算法效果不明显，只有少数项目可以实现工期优化。随着问题规模的增大，局部优化算法效果逐渐变好。当问题规模为 1200 时，平均优化比例超过 4%。由于 SCRDS 算法对线边空间的利用率已经很高，在作业规模较小时，很难找到满足条件的作业通过微

调物料摆放位置使其提前执行, 因此该局部优化算法针对大规模算例效果较好。

4 结 论

1) 以飞机移动生产线为实际背景, 将其抽象为资源受限项目调度问题, 并引入了物料配送能力约束和线边空间物料存放的约束, 对 AMALSP-CMD 问题建立了相应的数学模型

2) 针对物料的 JIT 配送策略及线边空间物料的摆放设计了以遗传算法为框架的启发式算法, 提高了线边空间利用率, 减少了因物料存放空间不足无法配送导致的作业延期, 优化了项目总工期。

3) 通过数值实验对算法进行了验证, 结果证明了本文提出算法的有效性, 与传统两阶段决策算法相比将物料配送及其在线边空间的存放与装配中的资源等约束进行综合考虑具有较高的理论与实际意义。

参考文献 (References)

- [1] Chaleshtarti A S, Shadrokh S. Branch and Bound Algorithms for Resource Constrained Project Scheduling Problem Subject to Cumulative Resources[C]//International Conference on Information Management, Innovation Management and Industrial Engineering. IEEE, 2011:147-152.
- [2] Berthold T, Heinz S, Lübbecke M E, et al. A Constraint Integer Programming Approach for Resource-Constrained Project Scheduling[M]. Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. Springer Berlin Heidelberg, 2010:313-317.
- [3] Blazewicz J, Lenstra J K, Kan A H G R. Scheduling subject to resource constraints: classification and complexity[J]. Discrete Applied Mathematics, 1983, 5(1):11-24.
- [4] Tormos P, Lova A. An efficient multi-pass heuristic for project scheduling with constrained resources[J]. International Journal of Production Research, 2003, 41(5):1071-1086.
- [5] Bukata L, Šúcha P, Hanzálek Z. Solving the Resource Constrained Project Scheduling Problem using the parallel Tabu Search designed for the CUDA platform[J]. Journal of Parallel & Distributed Computing, 2014, 77(11):58-68.
- [6] Bouleimen K, Lecocq H. A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version[J]. European Journal of Operational Research, 2003, 149(2):268-281.
- [7] Alcaraz J, Maroto C. A Robust Genetic Algorithm for Resource Allocation in Project Scheduling[J]. Annals of Operations Research, 2013, 102(1):83-109.
- [8] Hartmann S. A self-adapting genetic algorithm for project scheduling under resource constraints[J]. Naval Research Logistics, 2002, 49(5):433-448.
- [9] Merkle D, Middendorf M, Schneck H. Ant colony optimization for resource-constrained project scheduling[J]. Evolutionary Computation IEEE Transactions on, 2002, 6(4):333-346.
- [10] Kumar N, Vidyarthi D P. A model for resource-constrained project scheduling using adaptive PSO[J]. Soft Computing, 2015, 20(4):1565-1580.
- [11] 王琰, 陆志强. 基于多重约束的飞机移动装配线作业调度优化[J]. 工业工程与管理, 2011, 16(6):115-120.
Wang Y, Lu Z Q. Job Scheduling Optimization of Aircraft Moving Assembly Line under Multiple Constraints[J]. Industrial Engineering & Management, 2011, 16(6):115-120(in Chinese).
- [12] 郑倩, 奚立峰. 飞机移动生产线作业调度问题的启发式算法[J]. 工业工程与管理, 2015, 20(2):116-121.
Zheng Q, Xi L F. Heuristics For Aircraft Moving Assembly Line Scheduling Problem[J]. Industrial Engineering & Management, 2015, 20(2):116-121(in Chinese).
- [13] 葛茂根, 刘明周, 钱芳, 等. 基于 JIT 的多目标总装准时物料配送方法研究[J]. 中国机械工程, 2011, 22(23):2834-2838.
Ge M G, Liu M Z, Qian F, et al. Research on Multi-objective Method on Main Assembly Material Delivery Based on JIT[J]. China Mechanical Engineering, 2011, 22(23):2834-2838(in Chinese).
- [14] Fathi M, Alvarez M J, Mehraban F H, et al. A Multiobjective Optimization Algorithm to Solve the Part Feeding Problem in Mixed-Model Assembly Lines[J]. Mathematical Problems in Engineering, 2014, 11(1):809-812.
- [15] Fathi M, Rodríguez V, Fontes D B M M, et al. A modified particle swarm optimisation algorithm to solve the part feeding problem at assembly lines[J]. International Journal of Production Research, 2016(3):1-16.
- [16] Khayat G E, Langevin A, Riopel D. Integrated production and material handling scheduling using mathematical programming and constraint programming[J]. European Journal of Operational Research, 2006, 175(3):1818-1832.
- [17] Aguirre A M, Mández C A, Castro P M. A hybrid scheduling approach for automated flowshops with material handling and time constraints[J]. International Journal of Production Research, 2014, 52(9):2788-2806.

作者简介:

胡鑫铭 男, 硕士研究生。主要研究方向: 生产调度的建模与优化。

E-mail: xmh@tongji.edu.cn

陆志强 男, 教授, 博士生导师。主要研究方向:

物流与供应链建模与优化、生产工程等, 通讯作者。

Tel.: 021—69589485

E-mail: zhiqianglu@tongji.edu.cn

Optimization of aircraft moving assembly line scheduling problem considering material delivery

HU Xinming, LU Zhiqiang *

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

Abstract: This paper abstracted the scheduling of assembly process as a resource-constrained project scheduling problem on the background of aircraft moving assembly line with the decisions about capability of material delivery and the storage of line-side material being introduced. An integrating mathematical model with the objective of minimizing the makespan was established. We proposed a heuristic algorithm based on genetic algorithm framework, combining with solution generation algorithm and local search algorithm. With the global searching advantages of genetic algorithm, a joint decision of start time, material delivery time and material storage position for each job was made taking into account job sequence, resource constraints, delivery capability, line-side space and other factors through SCRDS algorithm. On this basis, a local optimization algorithm aiming at adjusting line-side material positions between two jobs was proposed to optimize the duration further. We programmed the algorithm and the results of the numerical experiments proved the validity of the model and algorithm.

Keywords: aircraft moving assembly line; material delivery; genetic algorithm; SCRDS algorithm; local optimization

Received: 2016-12-12; Accepted: 2017-01-20; Published online:

URL:

Foundation item: National Natural Science Foundation of China (61473211, 71171130)

* Corresponding author. Tel.: 021—69589485 E-mail: zhiqianglu@tongji.edu.cn