

## 基于耦合约束的飞机维修作业调度优化研究

杨少华,王 瑛,刘 刚

(空军工程大学 装备管理与安全工程学院,西安 710051)

**摘 要:** 为提升维修效能,提出飞机维修作业调度模型与调度优化算法。在沿用柔性作业车间调度问题形式化与图形化描述的基础上,设定耦合约束构建维修作业调度模型,选取遗传算法对执行步骤进行设计,引入耦合算子重新调整工序排序,以避免染色体违背耦合约束无法解码的情况。分析结果表明,该算法能适应飞机维修作业调度的需求,并且在 Brandimarte 测试中展现出较好的优化性能。

**关键词:** 飞机维修作业;耦合约束;染色体编码;多部件系统;遗传算法

**中文引用格式:**杨少华,王 瑛,刘 刚. 基于耦合约束的飞机维修作业调度优化研究[J]. 计算机工程,2016,42(9):297-304.

**英文引用格式:**Yang Shaohua,Wang Ying,Liu Gang. Research on Aircraft Maintenance Work Scheduling Optimization Based on Coupling Constraint[J]. Computer Engineering,2016,42(9):297-304.

## Research on Aircraft Maintenance Work Scheduling Optimization Based on Coupling Constraint

YANG Shaohua,WANG Ying,LIU Gang

(Equipment Management and Safety Engineering College,Air Force Engineering University,Xi'an 710051,China)

**[Abstract]** To improve maintenance efficiency, an aircraft maintenance work scheduling model and an optimization algorithm are proposed. By using formal and graphical presentation of flexible job shop scheduling problem, the coupling constraint is set to construct maintenance work scheduling model, and the execution steps of Genetic Algorithm(GA) are set where the coupling operator is introduced to adjust the process sequence to avoid the situation that chromosome violates the coupling constraint. Experimental results show that the algorithm can satisfy the needs of aircraft maintenance work scheduling and present well optimization performance in Brandimarte experiment.

**[Key words]** aircraft maintenance work; coupling constraint; chromosome coding; multiple component system; Genetic Algorithm(GA)

**DOI:**10.3969/j.issn.1000-3428.2016.09.052

### 1 概述

在飞机维修作业工作中,维修作业调度作为工作开展依据,实现任务要素与资源要素的优化整合。要素之间错综复杂且持续变化的关系却使得以往维修工作的进行主要依靠人工调度与协调,存在效率低下、灵活不足与传承困难等缺陷。为此,作为维修效能提升的关键,探索调度优化方案的自动生成方式,具有十分重要的理论价值与实践意义。

基于专业路径、基于章节目录以及基于任务与区域的工作分解结构,代表任务要素与资源要素的不同划分方式,但在飞机维修众多要素的优化整合中,工艺约束始终构成维修作业调度的基础,需要以

此作为调度优化的基本依据。源于调度问题的普适性,工业领域中广泛研究的柔性作业车间调度问题同样参照工艺约束,其理论与应用成果可以为飞机维修作业调度所借鉴。与此同时,作为调度优化研究的关键,调度优化算法需要满足工业对于可行性、时效性与经济性的要求,智能优化算法成为算法设计的主流形式。

以禁忌搜索为代表的个体智能算法依据个体搜索信息,关注性能估计<sup>[1-2]</sup>、局部极值跳出机制<sup>[3-4]</sup>的设计,可以产生十分明确的搜索方向<sup>[5]</sup>,但是所依托的邻域结构<sup>[6-8]</sup>必须结合问题背景进行设定,需要坚实的数学分析<sup>[9]</sup>,实践中约束条件的变化也容易导致算法的重新设计,应用范围受到限制。与此不同,

**作者简介:**杨少华(1971-),男,博士研究生,主研方向为智能控制、遗传算法;王 瑛,教授、博士、博士生导师;刘 刚,博士。

**收稿日期:**2016-01-11 **修回日期:**2016-02-23 **E-mail:** lucky\_evermail@sina.cn

以遗传算法 (Genetic Algorithm, GA) 为代表的群体智能算法依据群体搜索信息, 关注搜索算子<sup>[10-11]</sup>、集中性与多样性搜索<sup>[12-13]</sup>、精英保留策略的设计<sup>[14]</sup>, 凭借独立于问题背景的编码与解码机制<sup>[15]</sup>, 可以产生广泛的适用设计形式, 也能够依据实践需求进行调整, 应用潜力十分广阔。

在借鉴丰硕成果的同时, 需要注意柔性作业车间调度问题的工艺约束普遍采用链式约束形式, 只适用于描述单部件系统中工序之间的约束关系。作为多部件系统的代表, 飞机维修作业中任务要素与资源要素对时间与空间的竞争, 导致维修作业不仅涉及相同任务中工序之间的链式约束, 也存在不同任务中工序之间的耦合约束。单一工序的作业需要若干邻接前序工序的铺垫, 也可以影响若干邻接后序工序作业的进行, 与柔性作业车间调度所设定的单一工序至多存在一道邻接前序工序与一道邻接后序工序的情形存在十分明显的差异。

为此, 本文借鉴调度优化算法研究成果, 扩展工艺约束的存在形式, 综合耦合约束与链式约束, 以满足调度的实践需求, 对飞机维修作业调度优化进行研究, 提出一种基于耦合约束的飞机维修作业调度优化算法。

## 2 飞机维修作业调度模型的构建

为便于借鉴研究成果, 本文沿用柔性作业车间调度问题的形式化与图形化描述进行模型构建。

### 2.1 形式化描述

设定任务集合  $J = \{J_i | i = 1, 2, \dots, n\}$ 、工序集合  $O = \{O_{ij} | i = 1, 2, \dots, n, j = 1, 2, \dots, n_i\}$ 、单元集合  $M = \{M_k | k = 1, 2, \dots, m\}$ , 对于任意工序  $v \in O$ ,  $s_v$  为作业开始时间,  $p_{v,\mu_v}$  为不同单元的作业处理时间 (相同单元简化为  $p_v$ ),  $e_v$  为作业结束时间。工序作业需要满足设定的约束条件:

(1) 链式约束, 对于任意工序  $v \in O$ , 存在唯一的  $\rho_v = J_i \in J$ , 表示  $v$  从属于任务  $J_i$ 。对于从属于  $J_i$  的工序  $v, u$ , 依照相同任务工序的工艺次序进行作业, 设  $v$  在  $u$  之前作业, 需满足  $s_u \geq s_v + p_{v,\mu_v}$ , 设  $JP[v]$ ,  $JS[v]$  为  $v$  的链式邻接前序与后序工序。

(2) 耦合约束, 对于任意从属于不同任务的工序  $v, u$ , 需要满足不同任务工序工艺的作业次序, 设  $v$  在  $u$  之前作业, 需满足  $s_u \geq s_v + p_{v,\mu_v}$ , 设  $OP[v]$ ,  $OS[v]$  为  $v$  的耦合邻接前序与后序工序。

(3) 单元约束, 对于任意工序  $v \in O$ , 存在唯一的  $\mu_v = M_k \in M$ , 表示  $v$  从属于单元  $M_k$ , 但  $M_k$  可在  $v$  的选择作业单元集合  $C_v$  中任意选取。对于从属于  $M_k$  的工序  $v, u$ , 依照排序次序进行作业, 设  $v$  在  $u$  之前作业, 需满足  $s_u \geq s_v + p_{v,\mu_v}$ , 设定  $MP[v]$ ,  $MS[v]$  为  $v$

的单元邻接前序与后序工序。

设定  $f_{M_k}$  为单元  $M_k$  的完工时间,  $f_{M_k} = \max \{s_v + p_{v,\mu_v} | \mu_v = M_k\}$ , 表示其由从属工序的最大完工时间决定。以符号  $\rightarrow$  表示工序之间的工艺约束,  $v \rightarrow u$  代表  $v$  在  $u$  之前作业, 飞机维修作业调度模型以最小化完工时间为调度目标, 形式化模型可以描述为:

$$\min \max_{v \in O} s_v + p_{v,\mu_v} \quad (1)$$

$$\text{s. t. } s_v \geq 0, v \in O \quad (2)$$

链式约束:

$$s_u - s_v \geq p_{v,\mu_v} \text{ if } v \rightarrow u, v \in J_i, u \in J_i \quad (3)$$

耦合约束:

$$s_u - s_v \geq p_{v,\mu_v} \text{ if } v \rightarrow u, v \in J_i, u \in J_j \quad (4)$$

单元约束:

$$s_u - s_v \geq p_{v,\mu_v} \cup s_v - s_u \geq p_{u,\mu_u} \text{ if } \mu_v = \mu_u \quad (5)$$

### 2.2 图形化描述

飞机维修作业调度模型的图形化描述采用析取图表示, 设析取图  $G = (V, A, E)$ , 依据元素进行描述形式设定:

(1) 工序节点集  $V = O \cup \{0, *\}$ , 对应工序集合  $O$ ,  $0, *$  表示虚拟的开始作业工序与结束作业工序。

(2) 工艺约束弧集  $A = \{(v, u) | v, u \in O, v \rightarrow u\} \cup \{(0, v) | v \in O\} \cup \{(u, *) | u \in O\}$ , 从工艺视角描述工序之间的约束关系, 对任意约束弧  $(v, u)$ ,  $v, u$  所属任务相同表示链式约束;  $v, u$  所属任务不同表示耦合约束;  $(0, v)$  表示虚拟开始作业工序与任务首工序的工艺约束;  $(u, *)$  表示任务尾工序与虚拟结束作业工序的工艺约束。

(3) 单元约束弧集  $E = \bigcup_{k=1}^m \{(u, v)(v, u) | \mu_u = \mu_v = M_k, u, v \in O\}$ , 对应从单元视角描述工序之间的约束关系,  $(u, v)(v, u)$  双向弧表示  $v, u$  的选择作业单元集合, 划定工序分配与工序排序的选择范围。

飞机维修作业调度优化, 需要在单元约束弧集  $E$  中进行取舍。针对工序分配, 任意工序只能在作业单元集合中选择唯一的单元约束双向弧, 表示工序从属的作业单元。针对工序排序, 在工序从属单元的双向弧中选择单向弧, 表示工序在从属单元中的作业次序。工序分配与工序排序将  $E$  转化为  $S = \bigcup_{k=1}^m \{(u, v) | \mu_u = \mu_v = M_k, u, v \in O\}$ , 代表生成的调度优化方案。设  $D_s = \{V, A \cup S\}$ , 若  $D_s$  中不存在循环路径, 表明  $S$  为可行调度方案, 反之为不可行调度方案。

在  $G$  中, 以工序节点权重表示对应工序的作业处理时间  $p_{v,\mu_v}$ , 设  $p_0 = p_* = 0, s_0 = 0, t_* = 0$ 。考虑到工序  $v$  可以存在若干耦合约束, 作业开始时间  $s_v = \max \{s_{JP[v]} + p_{JP[v]}, \max \{s_{OP[v]} + p_{OP[v]}\}, s_{MP[v]} + p_{MP[v]}\}$ , 作业结束时间  $e_v = s_v + p_v$ 。

设工序  $u$  至  $v$  的最长路径及其距离为  $p(u, v)$ ,  $l(u, v)$ , 飞机维修作业调度优化的形式化模型可以

转化为在单元约束弧集  $E$  中选择最小化  $l(0, *)$  的调度优化方案  $S$ 。为便于应用研究成果,本文对飞机

维修作业调度模型与柔性作业车间调度模型进行对比,联系与区别如图 1 所示。

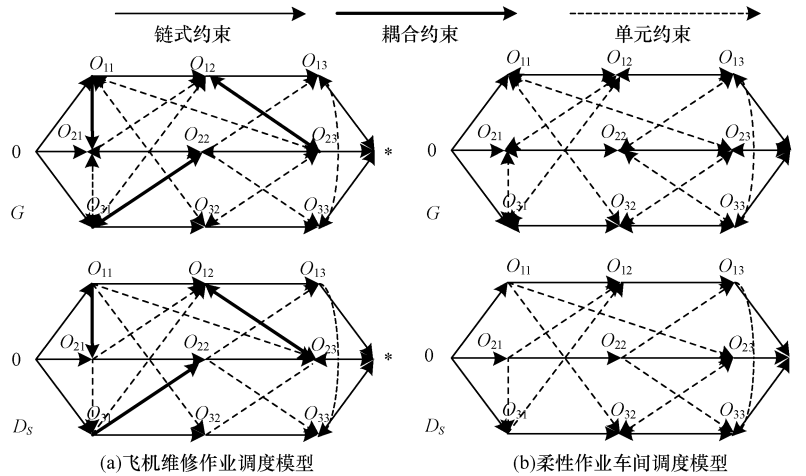


图 1 模型间的联系与区别

### 3 单目标调度优化算法设计

作为飞机维修作业调度优化的核心,调度优化算法的设计决定优化性能与效率。依据智能优化算法分析,本文选取群体智能算法中应用广泛的遗传算法为设计框架,围绕算法执行步骤进行设定。

#### 3.1 染色体编码机制的设计

飞机维修作业调度模型需要同时考虑工序分配与工序排序,为此,借鉴柔性作业车间调度问题的编码方式,染色体编码机制可以采用分段设计的形式。

##### 3.1.1 工序分配部分编码机制的设计

工序分配部分,染色体由  $\sum_{i=1}^n |J_i|$  位基因组成,基因依照  $J_i$  的次序进行排列,在  $J_i$  所属的基因序列中,依次依照  $O_{ij}$  的次序进行排列。对于任意基因  $v$ ,  $v$  所代表的工序固定不变,  $v$  的取值范围为其代表工序的选择作业单元集合。工序分配部分染色体的编码形式如图 2 所示。

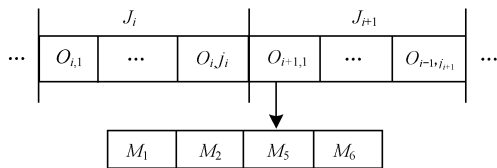


图 2 工序分配部分染色体的编码形式

##### 3.1.2 工序排序部分编码机制的设计

工序排序部分,染色体同样由  $\sum_{i=1}^n |J_i|$  位基因组成,基因以  $J_i$  的序号  $i$  的形式表述,序号  $i$  在染色体中出现的次数由  $n_i = |J_i|$  决定。对于任意基因  $v$ ,  $v$  所代表的工序动态变化,设  $v$  的取值为  $i$ ,代表其所属任务为  $J_i$ ,且截止到  $v$ ,  $i$  在染色体中出现的次数为

$j$ ,代表  $v$  对应的工序为  $O_{ij}$ 。工序排序部分染色体的编码形式如图 3 所示。

工序排序部分编码	5	5	2	1	3	3
所属任务	$J_5$	$J_5$	$J_2$	$J_1$	$J_3$	$J_3$
代表工序	$O_{51}$	$O_{52}$	$O_{21}$	$O_{11}$	$O_{31}$	$O_{32}$

图 3 工序排序部分染色体的编码形式

#### 3.2 调度优化方案的初始化

优良的初始调度方案可以显著促进算法的优化性能与优化效率,为保证收敛概率,需要提升种群的多样性,为此,本文设计均衡生成与随机生成对调度优化方案的初始化进行设定。

设定规模为  $n$  的任务分配数组  $JobA$ ,  $JobA(i)$  代表  $J_i$  中已获得分配的工序数量,令  $JobA(i) = 0$ ; 设定规模为  $m$  的单元负载数组  $ProcessM$ ,  $ProcessM(k)$  代表  $M_k$  中工序的累积作业时间,令  $ProcessM(k) = 0$ ; 设定待分配工序链表  $AllocationL$ , 令链表为空。

##### 3.2.1 均衡生成方式

单元之间的负载均衡在一定程度上促进完工时间的最小化,为此,本文设计调度优化方案的均衡生成方式:

(1) 依次对  $J_i$  中尚未分配的首工序  $O_{ij}$  进行检验,设  $O_{ij}$  的耦合邻接前序工序为  $O_{st}$ ,若  $JobA(s) \geq t$ ,说明  $O_{ij}$  可以进行分配,将  $J_i$  的序号  $i$  移动至  $AllocationL$  尾部;反之,说明  $O_{ij}$  尚未满足分配条件。

(2) 为均衡各任务的分配,以  $(|J_i| - JobA(i)) / |J_i|$  为选择概率对  $AllocationL$  中待分配的工序进行选择,设选定的工序为  $O_{ij}$ ,  $JobA(i) + 1$ ,清空  $AllocationL$ 。

(3)为均衡各单元的负载,以  $1/(p_{O_{ij},M_k} + Process M(k))$  为选择概率对  $O_{ij}$  的选择作业单元进行选择,设选定的单元为  $M_k, ProcessM(k) + p_{O_{ij},M_k}$ 。

(4)依照染色体编码方式,将选定的  $O_{ij}$  与  $M_k$  在对应的基因中进行设定,以循环步骤(1)~步骤(4)的方式形成完整的染色体。

### 3.2.2 随机生成方式

为使种群均匀分散在调度方案空间中,保持种群的多样性,本文设定调度优化方案的随机生成方式:

(1)依次对  $J_i$  中尚未分配的首工序  $O_{ij}$  进行检验,设  $O_{ij}$  的耦合邻接前序工序为  $O_{st}$ ,若  $JobA(s) \geq t$ ,说明  $O_{ij}$  可以进行分配,将  $J_i$  的序号  $i$  移动至  $AllocationL$  尾部;反之,说明  $O_{ij}$  尚未满足分配条件。

(2)对  $AllocationL$  中待分配的工序随机进行选择,设选定的工序为  $O_{ij}, JobA(i) + 1$ ,清空  $AllocationL$ 。

(3)对  $O_{ij}$  的选择作业单元随机进行选择,设选定的单元为  $M_k, ProcessM(k) + p_{O_{ij},M_k}$ 。

(4)依照染色体编码机制,将选定的  $O_{ij}$  与  $M_k$  在对应的基因中进行设定,以循环步骤(1)~步骤(4)的方式形成完整的染色体。

### 3.3 交叉算子的设计

交叉算子模拟生物代际的遗传方式,延续父代染色体的优良性状,需要分别对染色体工序分配部分与工序排序部分进行设计。

#### 3.3.1 工序分配部分交叉算子的设计

工序分配部分,染色体中基因代表的工序固定不变,交叉算子只能针对相同基因进行设计,本文采用均匀交叉操作作为交叉算子,如图4所示。

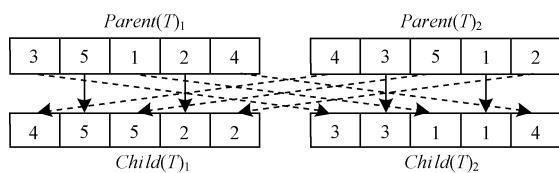


图4 工序分配部分染色体的交叉算子

(1)在区间  $[0, \sum_{i=1}^n |J_i|]$  中,随机生成整数  $N$ ,并在工序分配部分染色体中随机选择  $N$  位基因。

(2)在父代染色体  $Parent(T)_1$  与  $Parent(T)_2$  中,将选定的  $N$  位基因的数值,分别复制到对方子代染色体  $Child(T)_2$  与  $Child(T)_1$  对应的基因中。

(3)在父代染色体  $Parent(T)_1$  与  $Parent(T)_2$  中,将未经复制的  $\sum_{i=1}^n |J_i| - N$  位基因的数值,分别复制到各自子代染色体  $Child(T)_1$  和  $Child(T)_2$  对应的基因中。

#### 3.3.2 工序排序部分交叉算子的设计

在工序排序部分,染色体中基因代表工序的排列次序,交叉算子需要针对不同基因进行设计,本文采用 POX 交叉操作作为交叉算子,如图5所示。

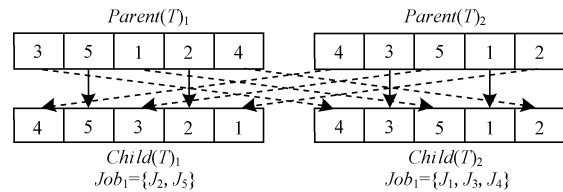


图5 工序排序部分染色体的交叉算子

(1)将工件集随机划分为互补的子集  $Job_1$  和  $Job_2$ 。

(2)在父代染色体  $Parent(T)_1$  中,选择属于  $Job_1$  的基因,保持其次序对位复制到子代染色体  $Child(T)_1$  中;在父代染色体  $Parent(T)_2$  中,选择属于  $Job_2$  的基因,保持其次序对位复制到子代染色体  $Child(T)_2$  中。

(3)在父代染色体  $Parent(T)_1$  中,选择属于  $Job_1$  的基因,保持其次序填补到子代染色体  $Child(T)_2$  空缺的基因中;在父代染色体  $Parent(T)_2$  中,选择属于  $Job_2$  的基因,保持其次序填补到子代染色体  $Child(T)_1$  空缺的基因中。

### 3.4 变异算子的设计

变异算子模拟生物代际的基因突变,改善种群多样性并以一定概率跳出局部极值,需要分别对染色体工序分配部分与工序排序部分进行设计。

#### 3.4.1 工序分配部分变异算子的设计

工序分配部分,染色体中基因代表的工序相对固定,变异算子只能针对相同基因进行设计:

(1)在区间  $[0, \sum_{i=1}^n |J_i|]$  中,随机生成整数  $N$ ,并在工序分配部分染色体中随机选择  $N$  位基因。

(2)依次对选定的基因进行变异操作,设基因  $v$  代表工序  $O_{ij}$ ,在  $O_{ij}$  的选择作业单元中,以  $1/p_{O_{ij},M_k}$  为概率随机选择作业单元。

#### 3.4.2 工序排序部分变异算子的设计

在工序排序部分,染色体中基因代表工序的排列次序,变异算子采用嵌入操作与交换操作进行设计:

(1)在区间  $[0, \sum_{i=1}^n |J_i|]$  中,随机生成整数  $N$ ,并在工序排序部分染色体中随机选择  $N$  位基因。

(2)依次对选定的基因进行变异操作,选定的基因  $v$  在嵌入操作与交换操作中随机选择变异方式,并在染色体中随机选择嵌入或交换位置。

### 3.5 耦合算子的设计

本文在染色体工序排序部分编码机制的设计中,采用柔性作业车间调度问题的编码方式,只能保证链式约束的满足,在经过交叉与变异操作之后,染色体可能违背耦合约束导致无法解码。为此,需要引入适应耦合约束的调整机制,本文以耦合算子的形式表述。

设定规模为  $n$  的工件排序数组  $SequenceT$ ,  $SequenceT(i)$  代表  $J_i$  中已获得调整的工序数量,令  $SequenceT(i) = 0$ ; 设定待排序工序链表  $SequenceL$ , 令链表为空。具体步骤如下:

(1) 对父代染色体  $Parent(T)$  中待调整的首位基因  $v$  进行检验, 设  $v$  所代表工序  $O_{ij}$  的耦合邻接前序工序为  $O_{st}$ , 若  $SequenceT(s) \geq t$ , 说明  $O_{ij}$  满足调整条件, 将  $v$  移动至子代染色体  $Child(T)$  的尾部,  $SequenceT(i) + 1$ ; 反之, 将  $v$  移动至待排序工序链表  $SequenceL$  的尾部。

(2) 依次检验待排序维修工序链表  $SequenceL$  中的基因, 若  $v$  所代表工序  $O_{ij}$  满足调整条件, 将  $v$  从  $SequenceL$  中移除至子代染色体  $Child(T)$  的尾部,  $SequenceT(i) + 1$ 。

(3) 循环执行步骤(1)、步骤(2), 直至父代染色体  $Parent(T)$  中基因调整完毕且  $SequenceL$  为空。

耦合算子设计反映耦合约束对染色体工序排序的制约, 是连接维修作业调度与柔性作业车间调度的桥梁。

### 3.6 染色体解码机制的设计

染色体解码机制是编码机制的逆运算, 但不同的解码方式可以产生不同的调度优化方案。完工时间属于正则性目标, 最优调度方案存在于活动调度中, 为此, 染色体解码需要设计排除半活动调度的机制。

设定工序分配单元矩阵  $AllocationM$ , 依次对工序分配部分染色体基因进行解码,  $AllocationM(i, j)$  代表  $O_{ij}$  选定的单元。设定工序分配作业时间矩阵  $AllocationP$ , 依据  $AllocationM(i, j)$ ,  $AllocationP(i, j)$  代表  $O_{ij}$  在选定单元中的作业时间。生成活动调度方案的关键在于工序排序部分染色体的解码步骤如下:

(1) 计算工序排序部分染色体首位基因  $v$  所代表的工序  $O_{ij}$ , 从染色体中移除  $v$ , 检验  $O_{ij}$  是否为所属任务首工序以及所属单元首工序, 并且没有耦合邻接前序工序。

(2) 若满足,  $s_{O_{ij}} = 0$ , 计算其作业结束时间。

(3) 若不满足, 计算  $e_{JP[O_{ij}]}$ ,  $e_{OP[O_{ij}]}$  以及  $O_{ij}$  所属单元的可嵌入最早闲置时间区间  $[ps_i, pe_i]$ ,  $s_{O_{ij}} = \max\{e_{JP[O_{ij}]}, e_{OP[O_{ij}]}, ps_i\}$ , 计算其作业结束时间。

(4) 循环执行步骤(1)~步骤(3), 直至工序排序部分染色体为空。

### 3.7 选择算子设计

选择算子模拟自然选择, 为促进优良性状的延续, 本文设定迭代周期为  $T$  时的选择范围同时包含子代染色体  $Child(T)$  与父代染色体  $Parent(T)$ , 并采用锦标赛方式作为提升收敛效率的选择算子。步骤如下:

(1) 随机生成正整数  $K$ ,  $K$  的选择依调度规模有

所不同。

(2) 从父代染色体  $Parent(T)$  与子代染色体  $Child(T)$  中, 随机选择  $K$  条染色体进行适应度比较, 选定适应度最高的染色体进入  $T + 1$  周期的迭代过程。

(3) 循环执行步骤(1)、步骤(2), 直至染色体数量达到设定的种群规模。

与此同时, 为防止性能退化, 本文引入精英保留策略, 在算法框架中设定历史最优调度方案, 实时更新历次搜索进程中发现的最优调度方案。

### 3.8 终止条件的设计

本文采用群体智能优化算法常见的终止条件进行设定, 如下:

(1) 迭代次数达到设定值。

(2) 历史最优调度方案无改进迭代次数达到设定值。

综合算法执行步骤的设定, 本文设计的单目标调度优化算法流程如图 6 所示。

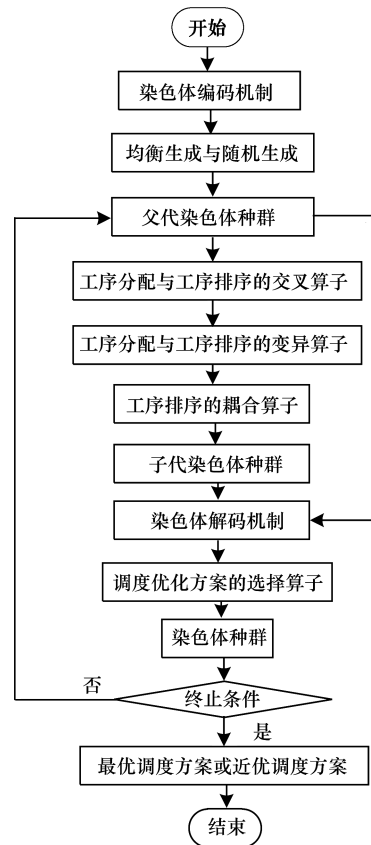


图 6 单目标调度优化算法流程

## 4 案例分析

### 4.1 飞机维修作业调度案例分析

为验证单目标调度优化算法的适用性, 本文参照飞机维修作业流程构建形式化模型, 工件集、工序集以及链式约束与耦合约束如图 7 所示, 单元与工序的作业时间如图 8 所示, 单位为 min。

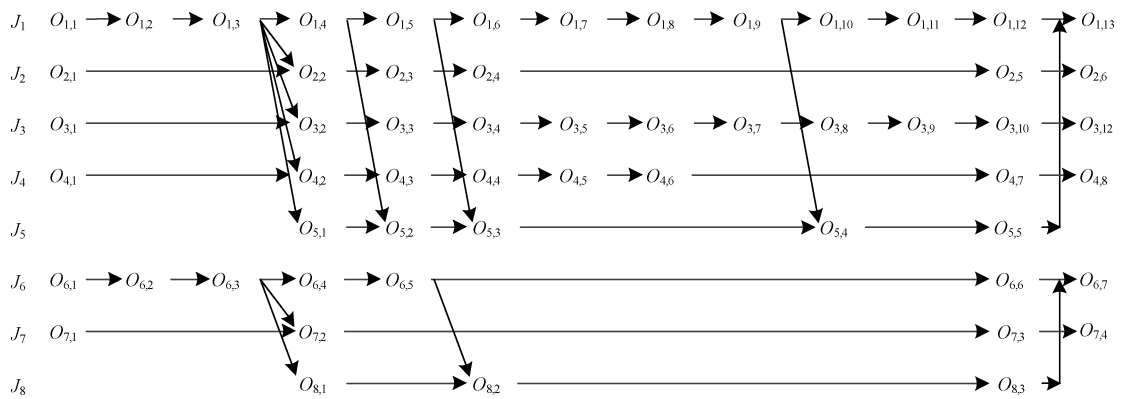


图7 工件集、工序集以及链式约束与耦合约束

	M <sub>1</sub>	M <sub>2</sub>	M <sub>3</sub>	M <sub>4</sub>	M <sub>5</sub>	M <sub>6</sub>		M <sub>7</sub>	M <sub>8</sub>		M <sub>9</sub>	M <sub>10</sub>	M <sub>11</sub>		M <sub>12</sub>	M <sub>13</sub>		M <sub>14</sub>	M <sub>15</sub>
O <sub>1,1</sub>	30	30	-	-	-	-	O <sub>2,1</sub>	30	-	O <sub>3,1</sub>	30	-	-	O <sub>4,1</sub>	30	-	O <sub>5,1</sub>	25	28
O <sub>1,2</sub>	23	29	14	30	20	22	O <sub>2,2</sub>	40	39	O <sub>3,2</sub>	14	20	13	O <sub>4,2</sub>	14	13	O <sub>5,2</sub>	32	27
O <sub>1,3</sub>	60	70	49	62	50	70	O <sub>2,3</sub>	43	49	O <sub>3,3</sub>	-	11	14	O <sub>4,3</sub>	12	15	O <sub>5,3</sub>	25	31
O <sub>1,4</sub>	35	46	37	25	-	28	O <sub>2,4</sub>	16	19	O <sub>3,4</sub>	20	15	18	O <sub>4,4</sub>	36	27	O <sub>5,4</sub>	24	21
O <sub>1,5</sub>	46	37	42	35	50	25	O <sub>2,5</sub>	26	29	O <sub>3,5</sub>	14	-	21	O <sub>4,5</sub>	24	21	O <sub>5,5</sub>	12	15
O <sub>1,6</sub>	55	62	42	-	57	56	O <sub>2,6</sub>	60	-	O <sub>3,6</sub>	12	14	11	O <sub>4,6</sub>	15	20	O <sub>8,1</sub>	24	25
O <sub>1,7</sub>	40	-	26	37	31	29				O <sub>3,7</sub>	-	22	25	O <sub>4,7</sub>	23	17	O <sub>8,2</sub>	36	39
O <sub>1,8</sub>	-	35	17	42	32	41				O <sub>3,8</sub>	12	11	-	O <sub>4,8</sub>	60	-	O <sub>8,3</sub>	12	14
O <sub>1,9</sub>	30	21	34	26	-	35				O <sub>3,9</sub>	12	14	15						
O <sub>1,10</sub>	52	43	38	48	50	52				O <sub>3,10</sub>	37	26	40						
O <sub>1,11</sub>	93	-	85	80	94	83				O <sub>3,11</sub>	60	-	-						
O <sub>1,12</sub>	31	26	22	30	25	27				O <sub>7,1</sub>	24	21	32						
O <sub>1,13</sub>	60	60	-	-	-	-				O <sub>7,2</sub>	14	13	15						
O <sub>6,1</sub>	24	35	31	24	19	21				O <sub>7,3</sub>	32	28	25						
O <sub>6,2</sub>	14	13	16	10	12	11				O <sub>7,4</sub>	35	21	26						
O <sub>6,3</sub>	21	23	25	20	24	22													
O <sub>6,4</sub>	14	21	15	23	13	17													
O <sub>6,5</sub>	25	21	26	21	24	22													
O <sub>6,6</sub>	31	25	36	32	27	25													
O <sub>6,7</sub>	37	31	32	36	31	38													

图8 单元与工序的作业时间

在形式化模型的基础上,本文采用 Java 编程,在主频 Intel(R) Core(TM) i3-2310M CPU 2.10 GHz,内存 2.00 GB 的计算机中运行程序,设定种群规模为  $2 \times (n + m)$ ,选择算子中锦标赛随机选择 2 种染色体进行对比,设定迭代次数为 100,历史最优调度方案无改进迭代次数为 30。依据单目标调度优化算法,需要确定调度优化方案的初始化形式,为此,对均衡生成与随机生成采用不同比例分别进行 10 次计算,实验情况如表 1 所示。

表 1 均衡生成与随机生成比例对调度算法的影响

序号	均衡生成比例/%	随机生成比例/%	平均最优调度方案的完工时间/min	平均收敛迭代次数
1	100	0	449.31	17.53
2	80	20	449.28	23.84
3	60	40	449.24	28.31
4	40	60	449.25	33.29
5	20	80	449.20	37.26
6	0	100	449.23	39.63

综合均衡生成与随机生成的不同比例,调度优化方案的不同初始化策略没有对调度优化算法的优化性能产生相对明显的影响,但是却可以改变调度优化算法的优化效率。观察不同初始化策略,随机生成比例在 [40%, 100%] 区间时,平均最优调度方案的性能指标出现波动,可以认为在此区间内随机生成方式对优化性能的影响减弱,然而,平均收敛迭代次数却会随均衡生成比例的减小而增大,且递增趋势逐渐放缓。为此,综合优化性能与收敛速率的影响,本文选定 (60%, 40%) 作为均衡方式与随机方式的选择比例。调度优化算法寻找到的最优调度方案所形成的甘特图如图 9 所示。

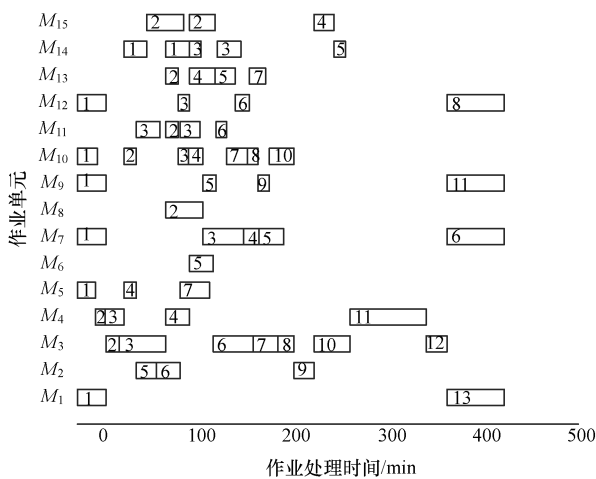


图 9 最优调度方案

最优调度优化方案的获得表明本文所设计的调度优化算法能够适用于飞机维修作业调度优化问题,但是,鉴于耦合约束研究尚未得到普遍关注,并不足以收集到足够与精准的案例数据以支撑调度优化算法性能的对。为此,单目标调度优化算法性能还有待验证。

#### 4.2 柔性作业车间调度案例分析

参照飞机维修作业调度的形式化与图形化描述,柔性作业车间调度问题可以视为维修作业调度模型的简化形式,单目标调度优化算法同样适用于柔性作业车间调度优化。基于不同的实验目标,Brandimarte, Hurink, Dauzère-Pérès, Chamber, Kacem 分别设计的测试数据共同组成柔性作业车间调度问题的测试资源,十分便于调度优化算法性能的对。

为此,本文选定 Brandimarte 测试数据进行试验,在相同配置的计算机中运行程序,同样设定种群规模为  $2 \times (n + m)$ ,选择算子中锦标赛随机选择 2 种染色体进行对比,迭代次数为 100,历史最优调度方案无改进迭代次数为 30,以 (60%, 40%) 作为均衡方式与随机方式的选择比例,分别进行 10 次计算,实验情况如表 2 所示。

表 2 Brandimarte 测试数据的完工时间比较

案例	$n \times m$	LB, UB	文献 [12] 已知最优完工时间 /min	本文最优完工时间 /min	本文平均完工时间 /min
MK01	10 × 6	36, 42	37	39	39.43
MK02	10 × 6	24, 32	26	26	26.12
MK03	15 × 8	204, 211	204	206	207.36
MK04	15 × 8	48, 81	60	60	60.11
MK05	15 × 4	168, 186	172	180	182.84
MK06	10 × 15	33, 86	57	58	58.00
MK07	20 × 5	133, 157	139	140	140.03
MK08	20 × 10	523, 523	523	523	523.58
MK09	20 × 10	299, 369	307	307	307.00
MK10	20 × 15	165, 296	198	198	200.43

观察实验情况,本文设计的单目标调度优化算法在 Brandimarte 测试数据的应用中可以获得或接近已知最优调度方案,并且调度优化的平均性能波动相对不明显。鉴于已知最优调度方案是多种调度优化算法共同作用的结果,单一调度优化算法通常只在若干案例中取得优势。为此,单目标调度优化算法的优化性能得到验证。

#### 5 结束语

为提升维修效能,探索调度优化方案的自动生成方式,本文对飞机维修作业调度问题的模型构建与单目标调度优化算法设计进行研究。实验结果表明,引入工序排序适应性调整机制的单目标调度优化算法可以满足飞机维修作业调度的需要,并具有良好的优化性能。今后将采用作业单元综合集成资源要素,并细化各类资源要素,以改进调度模型与算法,使其更加符合调度实践的需要。

#### 参考文献

- [1] Nowicki E. Some New Tools to Solve the Job Shop Problem[J]. European Journal of Operational Research, 2002, 93(2): 49-68.
- [2] Wang Shengyao. An Estimation of Distribution Algorithm for the Multi-objective Flexible Job-shop Scheduling Problem[C]//Proceedings of IEEE Symposium on Computational Intelligence in Scheduling. Washington D. C., USA: IEEE Press, 2013: 1-16.
- [3] Panos M P, Oleg V. Solving Job Shop Scheduling Problems Utilizing the Properties of Backbone and Big Valley[J]. Computer Optimal Application, 2010, 47(3): 61-76.

- [ 4 ] Mohammad A A, Jamal S. A Clustering-based Modified Variable Neighborhood Search Algorithm for a Dynamic Job Shop Scheduling Problem [ J ]. International Journal of Advanced Manufacturing Technology, 2014, 70 ( 3 ) : 1955-1961.
- [ 5 ] Nowicki E, Smutnicki C. An Advanced Tabu Search Algorithm for the Job Shop Problem [ J ]. Journal of Scheduling, 2005, 8 ( 2 ) : 145-159.
- [ 6 ] Balas E, Vazacopoulos A. Guided Local Search with Shifting Bottleneck for Job Shop Scheduling [ J ]. Management Science, 1998, 44 ( 2 ) : 262-275.
- [ 7 ] Nasiri M M, Kianfar F. A GA/TS Algorithm for the Stage Shop Scheduling Problem [ J ]. Computers & Industrial Engineering, 2011, 61 ( 1 ) : 161-171.
- [ 8 ] Nasiri M M, Kianfar F. A Guided Tabu Search/Path Relinking Algorithm for the Job Shop Problem [ J ]. International Journal of Advanced Manufacturing Technology, 2012 ( 58 ) : 1105-1113.
- [ 9 ] 刘刚, 王瑛, 张发. 基于关键路径求解作业车间调度问题的收敛性分析 [ J ]. 计算机集成制造系统, 2014, 20 ( 5 ) : 1078-1087.
- [ 10 ] Ziaee M. A Heuristic Algorithm for Solving Flexible Job Shop Scheduling Problem [ J ]. International Journal of Advanced Manufacturing, 2014, 71 ( 1 ) : 519-528.
- [ 11 ] Chang Hao-Chin, Liu Tung-Kuan. Optimization of Distributed Manufacturing Flexible Job Shop Scheduling by Using Hybrid Genetic Algorithms [ J ]. Journal of Intelligent Manufacturing, 2015, 26 ( 4 ) : 1623-1637.
- [ 12 ] Imen D, Kinza N M, Assia L. A New Genetic Algorithm for Flexible Job-shop Scheduling Problems [ J ]. Journal of Mechanical Science and Technology, 2015, 29 ( 3 ) : 1273-1281.
- [ 13 ] Maroua N, Abdelghani B, Abderezak J, et al. An Effective and Distributed Particle Swarm Optimization Algorithm for Flexible Job-shop Scheduling Problem [ J ]. Journal of Intelligent Manufacturing, 2015, 26 ( 2 ) : 1324-1341.
- [ 14 ] Ling Wang, Gang Zhou, Ye Xu, et al. An Effective Artificial Bee Colony Algorithm for the Flexible Job-shop Scheduling Problem [ J ]. International Journal of Advanced Manufacturing, 2012, 60 ( 1 ) : 303-315.
- [ 15 ] Mariano F, Ana C O, Fernando T. A Memetic Algorithm Based on a NSGAI Scheme for the Flexible Job-shop Scheduling Problem [ J ]. Annals of Operations Research, 2010, 181 ( 1 ) : 745-765.

编辑 刘冰

(上接第296页)

- [ 12 ] Daubechies I, Debrise M, de Mol C. An Iterative Thresholding Algorithm for Linear Inverse Problems with a Sparsity Constraint [ J ]. Communications on Pure & Applied Mathematics, 2004, 57 ( 11 ) : 1413-1457.
- [ 13 ] Lin Zhouchen, Ganesh A, Wright J, et al. Fast Convex Optimization Algorithms for Exact Recovery of a Corrupted Low-rank Matrix [ J ]. Journal of the Marine Biological Association of the UK, 2009, 56 ( 3 ) : 707-722.
- [ 14 ] Toh K C. An Accelerated Proximal Gradient Algorithm for Nuclear Norm Regularized Linear Least Squares Problems [ J ]. Pacific Journal of Optimization, 2010, 6 ( 3 ) : 615-640.
- [ 15 ] Liang Xiao, Ren Xiang, Zhang Zhengdong, et al. Repairing Sparse Low-rank Texture [ C ] // Proceedings of ECCV'12. Berlin, Germany: Springer, 2012: 482-495.
- [ 16 ] 郑勋焯, 陈保毕. 音乐信号小波阈值去噪模型及其应用 [ J ]. 长江大学学报: 自然版, 2015, 12 ( 1 ) : 5-9.
- [ 17 ] Hyvarinen A, Hurri J, Hoyer P O. Independent Component Analysis [ EB/OL ]. [ 2015-08-17 ]. [http://www.researchgate.net/publication/227042328\\_Independent\\_Component\\_Analysis?citationList=incoming](http://www.researchgate.net/publication/227042328_Independent_Component_Analysis?citationList=incoming).
- [ 18 ] 孔薇, 杨杰, 周越. 基于独立成分分析的强背景噪声去噪方法 [ J ]. 上海交通大学学报, 2004, 38 ( 12 ) : 1957-1961.
- [ 19 ] Karhunen J. Independent Component Analysis ( ICA ) and Blind Source Separation ( BSS ) [ EB/OL ]. ( 2011-02-09 ). <http://research.ics.aalto.fi/ica/>.
- [ 20 ] Kabal P. An Examination and Interpretation of ITU-R BS. 1387: Perceptual Evaluation of Audio Quality [ EB/OL ]. ( 2002-05-01 ). <http://www.mp3-tech.org/programmer/docs/kabalr2002.pdf>.
- [ 21 ] Thiede T, Treurniet W C, Bitto R, et al. PEAQ—The ITU Standard for Objective Measurement of Perceived Audio Quality [ J ]. Journal of the Audio Engineering Society, 2000, 48 ( 1/2 ) : 3-29.

编辑 陆燕菲