

项目制造环境下项目计划协同优化方法

孙清超*, 魏 静, 孙 伟, 臧含书

(大连理工大学 机械工程学院, 辽宁 大连 116024)

摘要: 为减少项目执行过程中不必要的等待时间、加快项目进程,提出了面向项目制造的项目计划协同优化方法.该优化方法涉及两个方面:建立项目计划信息之间的关联约束关系,以及关联约束基础上的优化解算.首先通过分析项目计划间信息关联,提出了基于产品信息和WBS的项目计划间接约束关系建立方式;进而结合不同计划模式的差异性,设计了一种用于项目计划协同优化的遗传算法.最后结合实例验证了优化方法的有效性及其合理性.

关键词: 项目制造;协同优化;间接约束关系;遗传算法

中图分类号: TH166;TP391 **文献标志码:** A

0 引言

项目制造针对的是一种面向订单设计(engineer-to-order, ETO)^[1~3]的生产类型,以大型复杂产品(如大型专用设备、飞机、舰船等)为开发和生产主体,以项目管理为中心,由技术准备(产品设计、工艺设计)、生产准备、产品生产、安装调试等相关活动所组成.与大批量生产、成批生产类型相比,项目制造具有根据客户要求组织设计、生产,产品可重复性低,客户化设计工作量大、生产准备周期长等特点^[4].

项目制造过程中不同阶段对应着不同类型的计划,如技术准备阶段对应产品设计计划、工艺设计计划、工具/工装设计计划等,生产准备及生产阶段对应主生产计划(MPS)、物料需求计划(MRP)和能力需求计划(CRP)等.项目制造过程中,不同类型的计划往往由不同的部门或组织管理,考虑到项目制造周期长、涉及部门广、过程管理复杂等特点^[4],很难对所有设计、工艺、采购、生产等任务统一管理.

针对项目计划优化问题,国内外已进行大量研究,但通常针对单一计划类型^[5~10],即项目计划中的各任务工作类型相同,鲜有文献针对不同

类型计划信息的协同优化问题进行研究.项目制造环境下,进行项目计划协同优化的前提是建立不同类型计划之间的约束关系,信息认知充分条件下,可以直接建立任务间的约束关系^[11].针对计划信息认知不充分或信息不透明的情况,本文提出了基于产品信息和WBS(work breakdown structure,工作分解结构)的项目计划间接约束方式.考虑到遗传算法搜索速度快、算法简单、容易取得相对最优解等特点^[7,12,13],选择应用遗传算法进行项目计划协同优化,但项目制造环境下的项目计划协同优化问题属于“间接约束基础上的协同优化”问题,相应遗传算法的基因表达结构、交叉及变异运算方式等与广泛应用的遗传算法都有所不同.

1 建立不同类型项目计划间的关联约束关系

1.1 面向项目制造的协同计划模型

项目制造环境下的任务关联约束关系模型如图1所示, $\{t_{a1}, t_{a2}, \dots, t_{ai}, \dots, t_{am}\}$ 为计划 J_a 中的任务, $\{t_{b1}, t_{b2}, \dots, t_{bj}, \dots, t_{bn}\}$ 为计划 J_b 中的任务,记 t_{ai} 与 t_{bj} 之间的依赖关系为 $r_{ai,bj} = (prop, type)$,

收稿日期: 2010-09-15; 修回日期: 2012-03-10.

基金项目: 国家自然科学基金资助项目(51005038).

作者简介: 孙清超*(1979-),男,博士,讲师, E-mail: qingchao@dlut.edu.cn.

其中 $prop$ 为 t_{ai} 与 t_{bj} 的关系属性, $type$ 为依赖关系类型. 根据协调理论, 任务间的依赖关系可以分为流(flow) 依赖、共享(sharing) 依赖和配合(fit) 依赖 3 种类型^[14]. 其中流依赖是指一个任务产生的资源被其他任务所使用, 如工艺设计需要依赖产品设计的结果; 共享依赖是指多个任务使用共同的资源, 如多个任务需要相同的人来执行, 需要参考相同的产品信息等; 配合依赖是指多个任务产生单个资源, 如多名工程师设计同一个产品的不同部件, 部件间存在配合关系. 根据任务间依赖关系, 可以得到计划 J_a 和计划 J_b 依赖关系矩阵 R_{ab} :

$$R_{ab} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix} \quad (1)$$

式中: R_{ab} 由 4 个分块矩阵 \mathbf{A} 、 \mathbf{B} 、 \mathbf{C} 、 \mathbf{D} 构成, 其中 \mathbf{A} 和 \mathbf{D} 分别为计划 J_a 和计划 J_b 内部各任务间的依赖关系, \mathbf{B} 为计划 J_a 对计划 J_b 的依赖关系, \mathbf{C} 为计划 J_b 对计划 J_a 的依赖关系.

$$\mathbf{B} = \begin{pmatrix} r_{a1,b1} & r_{a1,b2} & \cdots & r_{a1,bn} \\ r_{a2,b1} & r_{a2,b2} & \cdots & r_{a2,bn} \\ \vdots & \vdots & & \vdots \\ r_{am,b1} & r_{am,b2} & \cdots & r_{am,bn} \end{pmatrix}$$

$$\mathbf{C} = \begin{pmatrix} r_{b1,a1} & r_{b1,a2} & \cdots & r_{b1,am} \\ r_{b2,a1} & r_{b2,a2} & \cdots & r_{b2,am} \\ \vdots & \vdots & & \vdots \\ r_{bn,a1} & r_{bn,a2} & \cdots & r_{bn,am} \end{pmatrix}$$

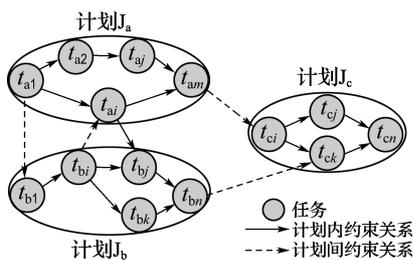


图 1 项目制造环境下的协同计划模型

Fig. 1 Collaborative plan model of project manufacturing

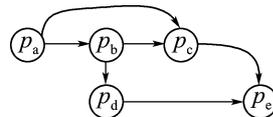
1.2 基于产品信息和 WBS 建立“间接约束关系”

由于不同的工作类型往往涉及不同的责任部门, 人力、设备的共享与冲突并不是主要的问题, 在整合不同类型的项目计划时主要考虑产品信息及相应的工作单元间的依赖关系.

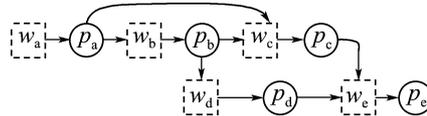
产品实现过程的不同阶段对应不同类型的产品信息, 从功能需求、概念结构到产品设计信息、工艺路线、工艺信息、工装信息到物料信息、生产信息等, 各类信息之间具有紧密的协作关系, 如图 2(a) 所示. 记 $r_{p_{ij}}$ 为产品信息 p_i 对 p_j 的依赖关系, $\{p_1, p_2, \dots, p_n\}$ 为相互关联的产品信息, 则产品信息依赖关系可以表示为如式(2) 所示的矩阵 R_p :

$$R_p = \begin{pmatrix} r_{p11} & r_{p12} & \cdots & r_{p1n} \\ r_{p21} & r_{p22} & \cdots & r_{p2n} \\ \vdots & \vdots & & \vdots \\ r_{pn1} & r_{pn2} & \cdots & r_{pnm} \end{pmatrix} \quad (2)$$

图 2(b) 描述了由于产品信息依赖导致的相应工作单元间的依赖关系. 其中工作单元 w_a 与 w_b 的协作关系属于流依赖; w_b 与 w_c 的关系中既存在流依赖, 也存在由于共享 p_a 而导致的共享依赖; w_c 与 w_d 之间既存在由于共享 p_b 而导致的共享依赖, 也存在由于 p_c 、 p_d 共同作为产生 p_e 的前提条件而导致的配合依赖.



(a) 产品信息间的关系



(b) 工作单元间的关系

图 2 产品信息间与工作单元间关系

Fig. 2 Relationship of product information and work cells

记 c_{ij} 为产品信息 p_i 与工作单元 w_j 的关系, 则产品信息集合 $\{p_1, p_2, \dots, p_n\}$ 与工作单元集合 $\{w_1, w_2, \dots, w_m\}$ 之间的关系可以表示为

$$\mathbf{C} = \begin{matrix} & p_1 & p_2 & \cdots & p_n \\ \begin{matrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{matrix} & \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mn} \end{pmatrix} \end{matrix} \quad (3)$$

记 $r_{w_{ij}}$ 为工作单元 w_i 对 w_j 的工作依赖, 产品信息集合 I_i 、 O_i 分别对应于 w_i 的输入和输出, I_j 、

O_j 分别对应于 ω_j 的输入和输出, I_k, O_k 分别对应于相关工作单元 ω_k 的输入和输出, 分别用 f, s, t 表示工作单元间的流依赖、共享依赖和配合依赖, 则

$$r_{wij} = \begin{cases} f; O_i \cap I_j \neq \emptyset \\ s; O_k \cap I_i \neq \emptyset \text{ 且 } O_k \cap I_j \neq \emptyset \\ t; O_i \cap I_k \neq \emptyset \text{ 且 } O_j \cap I_k \neq \emptyset \end{cases} \quad (4)$$

如果属于不同计划的任务 t_i, t_j 分别对应于工作单元 ω_i, ω_j , 在忽略任务间由于共享人力、设备等资源导致的依赖关系情况下, 任务 t_i, t_j 间的依赖关系 r_{ij} 等同于工作单元 ω_i, ω_j 间的依赖关系 r_{wij} , 即

$$r_{ij} = r_{wij} \quad (5)$$

2 一种用于项目计划协同优化的遗传算法

项目制造环境下的项目计划协同优化属于“间接约束基础上的协同优化”. “间接约束”即结合项目计划、WBS 和产品信息之间的内在关系建立项目计划间接约束关系; “协同优化”指的是根据间接约束关系, 考虑计划模式的差异, 通过协同优化各类型项目计划, 最终达到减少不必要的等待时间, 加快项目总体进程的目的. 因此, 协同优化遗传算法与通常的遗传算法^[12,13] 具有很大差异, 主要表现在种群初始化方式、适应函数和交叉运算方式等方面. 该算法采用“确定式”选择算子和基于次序的变异算子^[12], 本文不针对选择和变异算子进行详细论述.

2.1 初始化种群

从项目的整个生命周期进行分析, 项目制造中的项目计划 J 是由不同类型的子计划 J_i 整合而成的, 每个子计划 J_i 都由一系列相互联系的任务所组成. 构建待优化任务序列的过程始于一个空的任务序列 T_i , 首先需要设置对应的计划模式信息, 然后依次随机选择各子计划中的任务 t_j 添加到 T_i 中, 直至将所有的任务添加完毕. 在将 t_j 添加到 T_i 之前, 需要判断是否满足了将 t_j 添加到 T_i 中的条件: (1) 如果计划模式为前向模式 (forward, F), 添加 t_j 之前需要判断 t_j 的前置任务是否都添加到 T_i 中; (2) 如果计划模式为后向模式 (backward, B), 添加 t_j 之前需要判断 t_j 的后续

任务是否都添加到 T_i 中. 按照该方式可以得到由 pop_size 个任务序列构成的种群, 每个任务序列如图 3 所示.

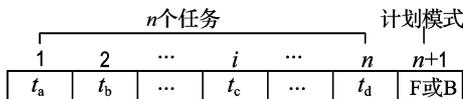


图3 待优化任务序列

Fig. 3 Sequence of tasks to be solved

2.2 适应函数的确定

t_i 为计划 J_a 中的任务, t_j 为计划 J_b 中的任务, 如果 t_i 为 t_j 的前置任务中最晚完成的任务, 则 t_j 在满足执行条件下的不必要的等待时间 $TW_{ij} = ES_j - EF_i$ (ES_j 为 t_j 的计划开始时间, EF_i 为 t_i 的计划完成时间). 项目计划协同优化的目的在于减少任务在满足执行条件下的不必要的等待时间, 所以确定适应函数如下:

$$f(I_i) = \max TW - \sum TW_{ij} \quad (6)$$

式中: $\max TW$ 为 $\sum TW_{ij}$ 的可能最大值, $\max TW \approx n \sum d_{kl}$, n 为项目中的子计划类型数, d_{kl} 为工期. $f(I_i)$ 越大, 表示不必要的等待时间 $\sum TW_{ij}$ 越小.

2.3 “F/B 混合交叉”运算方式

交叉运算作为最主要的遗传操作, 其意义在于使两个配对的染色体按某种方式交换部分基因, 从而形成新的个体. 考虑到不同个体对应的计划模式信息有所差异, 本文提出一种新的交叉运算方式——“F/B (forward/backward) 混合交叉”: 假设 $F = \{g_1^f, g_2^f, \dots, g_{n+1}^f\}$ 和 $M = \{g_1^m, g_2^m, \dots, g_{n+1}^m\}$ 为选择进行交叉的个体, F (father)、 M (mother) 中的前 n 位基因对应任务编号, 基因 g_{n+1}^f 和 g_{n+1}^m 对应计划制定模式信息 (F 或 B); $S = \{g_1^s, g_2^s, \dots, g_{n+1}^s\}$ 和 $D = \{g_1^d, g_2^d, \dots, g_{n+1}^d\}$ 为交叉后产生的新个体, 其中 S (son) 和 D (daughter) 分别从 F 和 M 继承计划模式信息. “F/B 混合交叉”运算方式的伪代码如下:

```

随机产生一个交叉点  $k(1 \leq k < n)$ 
// 生成 S
if( $g_{n+1}^f = F$ ) // 如果 F 对应的计划模式为 F
    
```

```

for( $i \leq k$ )// 确定 S 的前  $k$  位基因
     $g_i^s = g_i^f$ 
for( $k < i \leq n$ )// 确定 S 的后  $n - k$  位基因
    for( $1 \leq l \leq n$ )
        if( $g_l^m \notin S$ )
             $g_i^s = g_l^m$ 
            break
 $g_{n+1}^s = F$ // S 的计划模式为 F
else if( $g_{n+1}^f = B$ )// 如果 F 的计划模式为 B
    for( $n+1 \geq i > k$ )// 确定 S 的后  $n - k + 1$  位基因
         $g_i^s = g_i^f$ 
    for( $k \geq i \geq 1$ )// 确定 S 的前  $k$  位基因
        for( $n \geq l \geq 1$ )
            if( $g_l^m \notin S$ )
                 $g_i^s = g_l^m$ 
                break
// 同理, 根据 M 的计划模式信息生成 D
判断新的个体 S 和 D 是否满足任务依赖关系
    
```

如图 4 所示, 将个体 $F = \{4, 2, 1, 3, 5, 6, 7, F\}$ 和 $M = \{5, 1, 3, 2, 4, 7, 6, B\}$ 按照上述方法交叉后, 则可以得到两个新的个体: $S = \{4, 2, 1, 5, 3, 7, 6, F\}$ 和 $D = \{1, 3, 5, 2, 4, 7, 6, B\}$.

	$k=3$			$n=7$				
F	4	2	1	3	5	6	7	F
M	5	1	3	2	4	7	6	B
S	4	2	1	5	3	7	6	F
D	1	3	5	2	4	7	6	B

图 4 交叉运算示例

Fig. 4 Example of crossover operation

2.4 算法复杂度分析

本文算法是对通常遗传算法的改进, 以弥补通常遗传算法在多计划模式协同优化方面的不足. 改进算法的时间复杂度为 $O'(tmn) + O''(tm)$, 其中 $O'(tmn)$ 为优化运算时间, $O'(tmn) \approx O(tm)$ ($O(tm)$ 为通常遗传算法的时间复杂度), $O''(tm)$ 为构建初始种群时间, $O''(tm) = kO(tm)$, 其中 $k \approx (n_{r1} + n_{r2}) / n_{r1}$, n_{r1} 、 n_{r2} 分别为计划内、计划间的约束关系系数, 如第 3 章所述示例中, $n_{r1} = 27$, $n_{r2} = 30$, 则 $k \approx 2$.

3 应用实例

以某企业开发某轧管机组为例分析项目计划

协同优化问题. 该轧管机组由 50 多个关键子系统、部件组成, 其中轧管机传动装置由轧管机齿轮机座、轧管机主减速器等 6 个关键部件组成, 研制周期约 1 a, 涉及产品设计、工艺设计、铸热、锻热、机加、热处理、焊接、装配等类型的工作, 主要任务如表 1~6 所示.

结合表 1~6 中的数据, 应用 1.1、1.2 中介绍的方法, 可以建立各类型任务之间的关约束关系, 如 $r_{11,22} = f$, $r_{15,16} = s$ 等.

表 1 工作类型 A

Tab. 1 Work type A

名称	工期	输入信息	输出信息
t_1	5		p_1, p_2, p_3
t_2	5	p_1	p_4
t_3	10	p_2	p_5
t_4	15	p_3	p_6
t_5	7	p_4, p_5	$p_7, p_8, p_9, p_{10}, p_{11}$
t_6	5	p_7	p_{15}, p_{16}
t_7	6	p_6	p_{12}, p_{13}, p_{14}
t_8	8	p_{12}	p_{18}, p_{19}
t_9	8	p_8	
t_{10}	8	p_9	
t_{11}	10	p_{11}	p_{17}
t_{12}	8	p_{13}	p_{20}, p_{21}
t_{13}	9	p_{14}	p_{22}
t_{22}	10	p_{17}	p_{25}
t_{23}	13	p_{25}	p_{33}

表 2 工作类型 B

Tab. 2 Work type B

名称	工期	输入信息	输出信息
t_{14}	12	p_{15}	p_{23}, p_{24}
t_{15}	17	p_{23}	p_{29}
t_{16}	20	p_{24}	p_{31}
t_{17}	17	p_{16}	p_{30}
t_{18}	18	p_{30}, p_{31}	p_{32}, p_{42}
t_{19}	21	p_{18}	p_{28}
t_{20}	12	p_{19}	p_{27}
t_{21}	13	p_{27}, p_{28}	
t_{24}	21	p_{20}	p_{43}, p_{47}
t_{25}	26	p_{21}	p_{44}, p_{48}
t_{26}	10	p_{22}	p_{25}, p_{26}
t_{27}	17	p_{25}	p_{45}
t_{28}	11	p_{26}	p_{46}

表 3 工作类型 C
Tab. 3 Work type C

名称	工期	输入信息	输出信息
t_{29}	90	p_{29}	p_{34}
t_{30}	20	p_{34}	p_{53}
t_{31}	60	p_{42}	p_{35}
t_{32}	23	p_{32}, p_{35}	p_{54}
t_{33}	129	p_{47}, p_{48}	
t_{34}	109	p_{49}	p_{58}
t_{35}	60	p_{50}	p_{59}
t_{36}	35	p_{51}	p_{37}
t_{37}	30	p_{37}, p_{38}, p_{39}	p_{56}, p_{57}
t_{38}	25		p_{38}, p_{52}
t_{39}	20	p_{41}	p_{39}

表 4 工作类型 D
Tab. 4 Work type D

名称	工期	输入信息	输出信息
t_{40}	36	p_{43}	p_{49}
t_{41}	20	p_{44}	p_{50}
t_{42}	90	p_{45}	p_{51}
t_{43}	49	p_{46}	p_{41}, p_{52}

表 5 工作类型 E
Tab. 5 Work type E

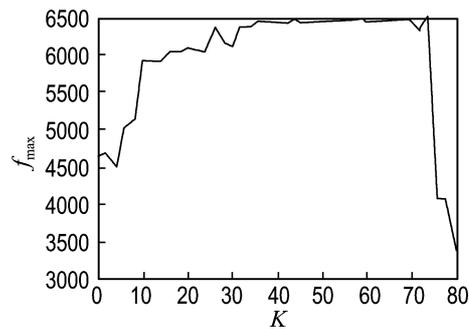
名称	工期	输入信息	输出信息
t_{44}	94	p_{33}	p_{36}
t_{45}	60	p_{36}	p_{55}

表 6 工作类型 F
Tab. 6 Work type F

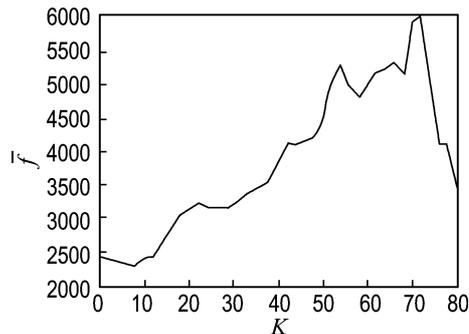
名称	工期	输入信息	输出信息
t_{46}	23	$p_{53}, p_{55}, p_{54}, p_{56}, p_{58}, p_{59}$	p_{40}
t_{47}	10	p_{40}	

遗传算法程序采用 Java 语言编写. 采用 2.1 所述方法初始化种群; 采用 2.3 所述方法进行交叉运算; 适应度函数如式(6)所示. 设定初始种群规模为 500, 交叉概率为 0.60, 变异概率为 0.02, 则得到图 5 所示的运算结果.

从图 5(a)中可以看出, 第 31 代种群以后, 最大适应度值稳定在 6 500 左右, 相对于初始任务序列, 优化后不必要的等待时间大大减少.



(a) 各代最大适应度值



(b) 各代平均适应度值

图 5 遗传算法运算结果

Fig. 5 Computing results of genetic algorithm

该遗传算法未考虑资源约束条件, 假定各任务的资源需求都能得到满足, 因此 74 代种群后, 通过交叉、变异等运算改变各任务先后顺序, 使适应度值出现了急剧下降的情况. 实际应用过程中, 项目管理人员需要结合资源负荷、任务优先级等其他因素综合判断优化结果的合理性, 并经过多次优化、调整确定最优任务序列.

4 结 论

(1) 在结合协调理论研究项目计划、WBS、产品信息内在联系的基础上, 提出了基于 WBS 和产品信息的间接关联约束关系建立方法;

(2) 项目制造环境下的项目计划协同优化属于“间接约束基础上的协同优化”, 针对其“间接约束”及“协同优化”的特点, 设计了一种新型遗传算法, 该算法保留了项目计划模式的差异性, 对于其他“项目计划协同优化”问题具有一定借鉴意义.

参考文献:

[1] CHEN Chin-sheng. Concurrent engineer-to-order operation in the manufacturing engineering contracting

- industries [J]. **International Journal of Industrial and Systems Engineering**, 2006, **1**(1/2):37-58
- [2] LI Xiao-ping, XU Xiao-fei, ZHAN De-chen. Multilevel hierarchical production planning architecture for engineer-to-order enterprises [J]. **Journal of Harbin Institute of Technology (New Series)**, 2002, **9**(3):254-258
- [3] EARL C, SONG Dong-ping, HICKS C. Planning complex engineer-to-order products [M] // GOGU G, COUTELLIER D, CHEDMAIL P, eds. **Recent Advances in Integrated Design and Manufacturing in Mechanical Engineering**. Dordrecht: Kluwer, 2003: 463-472
- [4] 陈心力, 江志斌, 胡宗武, 等. 改进的项目制造方案的物料需求计划运算逻辑[J]. 工业工程与管理, 2003, **8**(4):67-70
- [5] 程 序, 吴 澄. 大规模项目调度问题的分解和协调优化方法[J]. 清华大学学报(自然科学版), 2003, **49**(1):153-156
- [6] 潘晓英, 焦李成. 项目优化调度的多智能体社会进化算法[J]. 计算机研究与发展, 2008, **45**(6):998-1003
- [7] GONCALVES J F, MENDES J J M, RESENDE M G C. A genetic algorithm for the resource constrained multi-project scheduling problem [J]. **European Journal of Operational Research**, 2008, **189**(3):1171-1190
- [8] VAN PETEGHEM V, VANHOUCHE M. A genetic algorithm for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problem [J]. **European Journal of Operational Research**, 2010, **201**(2):409-418
- [9] PAN N H, HSAIO P W, CHEN K Y. A study of project scheduling optimization using Tabu Search algorithm [J]. **Engineering Applications of Artificial Intelligence**, 2008, **21**(7):1101-1112
- [10] CHIANG Chuan-wen, HUANG Yu-qing, WANG Wen-yen. Ant colony optimization with parameter adaptation for multi-mode resource-constrained project scheduling [J]. **Journal of Intelligent & Fuzzy Systems**, 2008, **19**(4):345-358
- [11] 梅成刚, 肖人彬. 复杂设计过程的信息建模与重组 [J]. 机械科学与技术, 2004, **23**(1):105-109
- [12] 席裕庚, 柴天佑. 遗传算法综述[J]. 控制理论与应用, 1996, **13**(6):697-708
- [13] ALCARAZ J, MAROTO C. A robust genetic algorithm for resource allocation in project scheduling [J]. **Annals of Operations Research**, 2001, **102**(1-4):83-109
- [14] 孙清超, 郭 钢. 基于 DSM 的多产品开发项目协同规划方法研究[J]. 机械科学与技术, 2009, **28**(9): 1203-1207

Collaborative optimization method of project plans for project manufacturing environment

SUN Qing-chao*, WEI Jing, SUN Wei, ZANG Han-shu

(School of Mechanical Engineering, Dalian University of Technology, Dalian 116024, China)

Abstract: To reduce unnecessary waiting time during tasks executing process and accelerate project process in project manufacturing, the collaborative optimization method of project plans is presented. The optimization method is divided into two aspects: project plans associating and project plans optimizing calculation under the condition that all kinds of plans are associated. Firstly, the association method of project plans based on product data and WBS is put forward by analyzing the information relationship among project plans. Additionally, considering differences among multiple project plan models, a kind of genetic algorithm for collaborative optimizing tasks list is designed. Finally, the example is provided and the case study results reveal that these methods are valid and practical.

Key words: project manufacturing; collaborative optimization; indirect constraint relation; genetic algorithm