

# 机床产品协同任务规划与管理的研究

孟秀丽<sup>1</sup>, 韩向东<sup>1</sup>, 曹杰<sup>2</sup>

(1. 南京财经大学工业工程系, 南京 210003; 2. 南京财经大学电子商务系, 南京 210003)

**摘要:** 研究并实现了机床产品协同设计任务规划与管理功能。采用按功能划分和按结构划分相结合的方式对机床产品协同设计的任务进行分解。通过对结构设计矩阵递归实施独立操作和定耦与归一操作完成设计任务的规划。研究影响任务分配的因素及其具体分配算法, 提出了机床层次式递归化任务分解模型, 最后给出了任务规划功能模块的网络实现流程并开发了相应的原型系统。

**关键词:** 机床; 协同设计; 任务规划; 任务分配

## Research on Collaborative Task Schedule and Management of Machine Tool

MENG Xiuli<sup>1</sup>, HAN Xiangdong<sup>1</sup>, CAO Jie<sup>2</sup>

(1. Department of Industrial Engineering, Nanjing University of Finance and Economics, Nanjing 210003;

2. Department of Electronic Business, Nanjing University of Finance and Economics, Nanjing 210003)

**【Abstract】** Task schedule and management of machine tool's collaborative design are studied and carried out. Task is decomposed by the means which combine function decomposition and structure decomposition. Unaided manipulation and coupling manipulation are used recursively in DSM to schedule task. Influencing factor of task assignment and the concrete arithmetic are studied. The layered and recursive task decomposition model of machine tool is proposed. In the end, implementation flow based on network of task schedule system is proposed and a corresponding prototype system is partially developed.

**【Key words】** Machine tool; Collaborative design; Task schedule; Task decomposition

机床产品协同设计是以一组任务的串行、并发和交叉耦合方式进行的, 设计过程复杂, 因此必须对机床产品协同设计过程进行有效的任务规划和控制。国内外学者已对其进行了不少研究<sup>[1,2]</sup>, 但针对机床产品协同设计任务规划的研究还不多见, 因此本文从任务规划入手, 对任务分解、任务重组和任务分配的方法进行了研究, 提出了机床产品开发任务分解模式及流程。

### 1 协同设计任务重组

#### 1.1 设计结构矩阵的建立

机床任务分解就是按照一定的要求将机床产品设计任务分解成多个子任务并确定各子任务之间的关系, 进而将子任务分解为若干设计活动, 由多个协同设计小组共同完成。机床产品设计信息量大、类型多、变化不规则, 而且涉及多个领域, 因此采用按功能划分和按结构划分相结合的方式对其任务进行分解。设计任务可以采用任务树型结构来描述, 总任务为最高层次, 然后分解出若干子任务, 子任务还可以进一步分解。

任务分解完成以后, 下面将应用矩阵分析的方法来判断任务分解的合理性, 并对设计过程活动重新进行组合。

设计结构矩阵(Design Structure Matrix, DSM)用来表达设计过程之间的信息联系, 设任务集合为  $(T_1, T_2, \dots, T_n)$ , 用一个  $n \times n$  的矩阵  $A$  表示设计结构矩阵:  $A = (a_{ij})_{n \times n}$ 。该矩阵的行和列对应着设计的各项任务, 主对角线元素标志着设计活动本身, 其它元素用以表示活动间联系的存在性。其中  $a_{ij} = 1 (i, j = 1, 2, \dots, n)$  表示任务  $T_j$  的进行需要任务  $T_i$  的

信息,  $a_{ij} = 0 (i \neq j)$  表示任务  $T_i$  和任务  $T_j$  之间不存在信息联系, 并令  $a_{ii} = 0$ 。

这种布尔矩阵以 0 和 1 表示设计过程的联系, 但在实际应用中, 各个设计过程间信息交流的程度不同, 其联系有强有弱, 采用布尔量无法反映信息交流的强弱。而模糊集理论描述这种不确定性的最有效工具, 所以, 采用模糊理论来表示设计结构矩阵中的元素, 通过增加的信息将布尔矩阵转化为数字矩阵。在机床产品协同设计中, 任务间相互作用程度的度量的取值方法为: 基于经验的模糊变量集合 {强, 较强, 中, 较弱, 弱}, 对应取值 {1, 0.75, 0.5, 0.25, 0}。用模糊变量值表示设计结构矩阵中的元素, 从而将布尔矩阵转化为数字矩阵, 就构成了模糊设计结构矩阵。

#### 1.2 任务重组步骤

设计结构矩阵反映了设计活动之间存在的联系, 通过对设计结构矩阵进行规划就可以达到对其所反映的设计过程进行分析和重组的目的。

首先看以下 2 个定义:

**定义 1** 设计结构矩阵中空行或空列(该行或该列的所有元素均为 0)所对应的活动称为独立任务, 这些活动可以独立进行。

**基金项目:** 江苏省高技术研究基金资助项目(BG2003020); 江苏省科技厅软科学基金资助项目(BR2004503)

**作者简介:** 孟秀丽(1975 -), 女, 博士、讲师, 主研方向: 协同设计, 网络化制造等; 韩向东, 教授; 曹杰, 博士、副教授

**收稿日期:** 2005-11-01 **E-mail:** mengxiuli2001@163.com

其中,空列表示该任务的进行不需要任何信息输入,故该任务可首先进行,排在前面;空行表示该任务的进行不向其它活动输出信息,故可以放在最后执行。

**定义2** 在模糊设计结构矩阵A中, $a_{ij} \in A, a_{ji} \in A, a_{ij} \neq 0, a_{ji} \neq 0$ ,这样的元素构成的集合称为耦合集,任务 $t_i$ 与 $t_j$ 称为耦合任务集,耦合任务集包括具有相互依赖关系的两个或多个任务,它表示了设计过程中形成信息回路的一类任务的集合。

耦合任务集是由在并行任务网络中构成强连通子集的任务组成的集合,对于机床产品这样一个复杂的设计项目来说,在整个项目的完成过程中存在多次范围不同的反复,所以也就存在多个耦合任务子集。

通过上述对耦合任务和独立任务的定义可知,整个任务分解与归划操作是上述独立操作和定耦与归一操作的递归过程,其程序实现流程如图1所示。首先进行独立操作,将空列表示的独立活动排在前面,将空行表示的活动排在后面,然后进行定耦和归一操作。

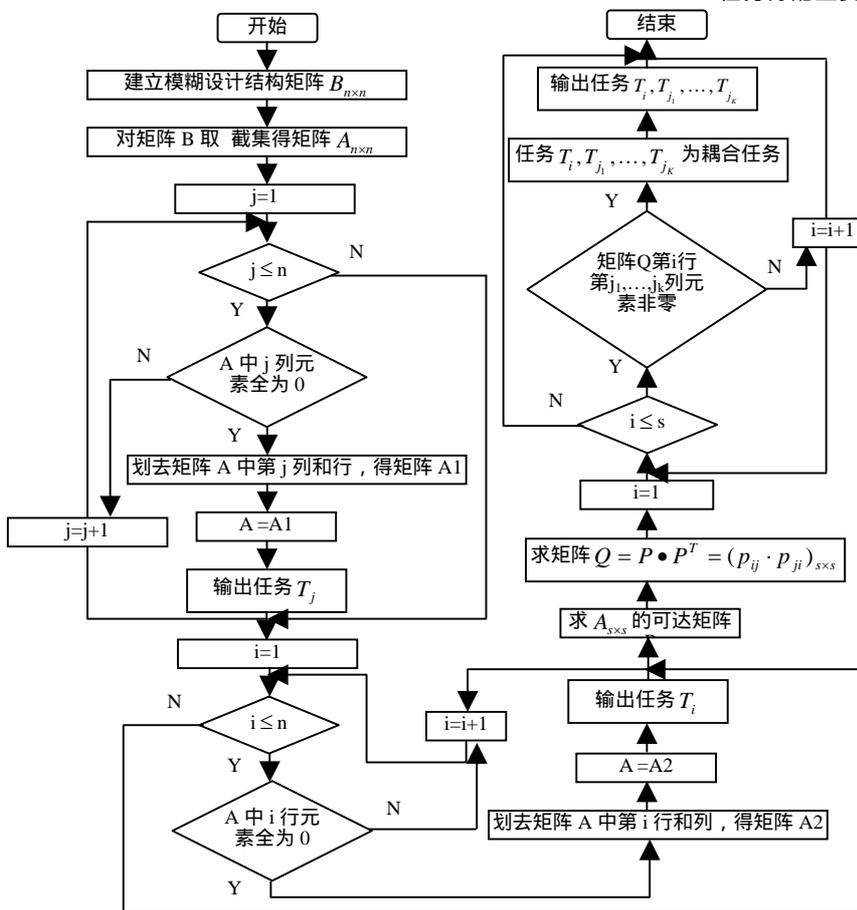


图1 任务分解规划算法流程

定耦操作的实质是耦合任务集的识别,即确定设计活动中耦合任务集的过程。根据耦合任务集的定义,耦合任务集对应于网络图中的强连通子集,耦合任务集的识别可以转化为设计过程图中强连通子集。为简化运算,对模糊设计结构矩阵A取其 $\lambda$  ( $0 \leq \lambda \leq 1$ , $\lambda$ 值视具体情况而定, $\lambda$ 实际上对应着活动间依赖程度的大小)截矩阵,得到一个简单的布尔矩阵,并对该矩阵实施分解算法,识别出其中的耦合任务集,耦合任务集的识别过程如下:

首先,用Warshall方法<sup>[3]</sup>从邻接矩阵A求可达矩阵P:

其次,求矩阵

$$Q = P \cdot P^T = (q_{ij}) \quad q_{ij} = p_{ij} \cdot p_{ij}^T = p_{ij} \cdot p_{ji}$$

若矩阵Q的第i行的非零元素在第 $j_1, j_2, \dots, j_k$ 列,则结点 $v_i, v_{j_1}, v_{j_2}, \dots, v_{j_k}$ 对应的任务构成的集合为耦合任务集。

对整个机床产品协同设计过程而言,任务间的信息交流并不受上述算法的影响。将各个耦合任务集看作子过程,则耦合任务集之间的信息流量就是其中各个相关任务信息流量的累加和。

另外,若耦合任务集的耦合度比较高,且由不同的小组执行,则频繁的信息交流会增加协同的难度,这时,就可以将耦合度高的子过程进行合并,对耦合过程集进行构造,形成新的子过程并交给一个协同小组完成,从而将耦合的各个活动与外界的联系转化为该新的子过程与外界的联系,以降低小组间的信息交流,这一过程称为耦合任务的归一操作。

## 2 协同设计任务分配

任务分配主要分2步来实现:一是将子任务分配给合适的小组;二是给每一个子任务确定合适的开始和结束时间,即任务的人员分配和时间分配。

### 2.1 设计任务的人员分配

协同设计的任务分配有许多影响因素,如成员的能力、任务的难度、成员的兴趣和工作负荷、任务的种类和数量,因此任务的分配要综合考虑各种因素。下面首先对问题的特征从数学上进行描述。

设某一任务集为 $T = \{T_1, T_2, \dots, T_n\}$ ,

人员集合为 $Man = \{E_1, E_2, \dots, E_m\}$ 。

任务分配问题就是寻找从T到Man的影射,将合适的任务分配给合适的人,则影响任务人员分配的因素主要有如下4个:

- (1) 人员综合能力矩阵 $P = (p_{ij})_{m \times n}$ ,用来定量化描述开发人员对各个任务的胜任程度。其中, $0 \leq p_{ij} \leq 1$ , $p_{ij}$ 的值越大,能力越强,在不引起资源冲突的情况下,应该将任务j分配给人员i。
- (2) 协同人员繁忙系数矩阵 $R = [r_1, r_2, \dots, r_n]$ ,其中, $0 \leq r_i \leq 1$ , $r_i$ 表示协同人员的繁忙系数。
- (3) 人员兴趣矩阵 $Q = (q_{ij})_{m \times n}$ ,其中 $0 \leq q_{ij} \leq 1$ ,每个协同小组以一个矢量 $\{q_{i1}, q_{i2}, \dots, q_{in}\}$ 描述对任务的选择倾向,且 $\sum_{j=1}^n q_{ij} = 1$ 。
- (4) 任务紧急程度矩阵 $E = [e_1, e_2, \dots, e_n]$ ,其中, $0 \leq e_j \leq 1$ , $e_j$ 为第j项任务的紧急程度系数。

具体的任务分配算法如下:

(1) 确定任务分配影响因素的系数

(1) 确定任务分配影响因素的系数

(1) 确定任务分配影响因素的系数

由于影响设计任务分配的诸多因素均为动态非量化的概念，带有不同程度的模糊性，因此给确定任务分配的影响因素带来了困难。本文采用模糊系数的确定方法，把影响分配的每个因素按其性质和程度细分为若干等级，再确定每个因素的各个等级系数。

首先，建立因素集，因素集为影响任务分配的各个因素的集合，表示为

$$U = \{u_1, u_2, u_3, u_4\} = \{q, p, r, e\}$$

其中  $q$  表示协同人员的兴趣， $p$  表示协同设计人员综合能力， $r$  表示协同人员繁忙系数， $e$  表示任务紧急程度。

然后，建立因素等级集。集合中的各因素具有不同程度的模糊性，所以每个因素按其结果的影响程度细分为若干等级，组成因素的等级集，即

$$u_i = \{u_{i1}, u_{i2}, \dots, u_{in_i}\}$$

式中  $u_{ij}$  ( $i = 1, 2, \dots, m; j = 1, 2, \dots, n_i$ ) 为第  $i$  个因素的等级  $j$  个等级， $n_i$  为该因素的模糊等级数，在此定义  $n_i = 6$ ，并定义各因素的等级系数，如表 1 所示。

表 1 各因素的等级系数

评语 因素	1级	2级	3级	4级	5级	6级
P	1.0	0.8	0.6	0.4	0.2	0.1
Q	0.9	0.7	0.5	0.3	0.1	0.0
R	1.0	0.8	0.6	0.4	0.2	0.0
E	1.0	0.8	0.5	0.4	0.2	0.1

### (2) 确定趋向矩阵

根据上述任务影响因素，定义趋向矩阵为

$$TR = (tr_{ij})_{m \times n} = \begin{bmatrix} p_{11} - r_1 + e_1 + q_{11} & p_{12} - r_1 + e_2 + q_{12} & \dots & p_{1n} - r_1 + e_n + q_{1n} \\ p_{21} - r_2 + e_1 + q_{21} & p_{22} - r_2 + e_2 + q_{22} & \dots & p_{2n} - r_2 + e_n + q_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ p_{m1} - r_m + e_1 + q_{m1} & p_{m2} - r_m + e_2 + q_{m2} & \dots & p_{mn} - r_m + e_n + q_{mn} \end{bmatrix}$$

### (3) 求任务执行矩阵

任务的执行模式是各种各样的，一个任务可以由一个人来完成，也可以由掌握该技术的若干个人组成的团队组合来完成，由执行矩阵  $D = [d_1 \ d_2 \ \dots \ d_n]$  来表示任务的执行模式，其中  $d_j$  表示协同人员  $d_j$  执行任务  $j$ 。任务执行矩阵采用以下方法求出<sup>[4]</sup>：

根据趋向矩阵  $TR$ ，生成优先矩阵  $O = (o_{ij})_{m \times n}$ ， $O$  的第  $j$  列第 1 个元素是  $x|tr_{xj} = \text{Max}(tr_{ij})$ ， $1 \leq i \leq m$ ，第 2 个元素是矩阵  $TR$  中  $j$  列次最大值  $tr_{ij}$  对应的  $i$  值，以此类推求出矩阵  $O$  的  $j$  列中所有的元素值。

如果  $m \geq n$ ，一个协同小组只能执行一个设计任务，否则一个协同小组可以执行多个设计任务。根据这一原则和趋向矩阵  $TR$  决定执行矩阵  $D = (d_{ij})_{1 \times n}$ 。

## 2.2 设计任务的时间分配

设一个项目有  $n$  个设计任务组成，则任意设计任务  $i$  包括 3 个时间参数，分别为：任务触发时刻  $t_{si}$ 、任务完成时刻  $t_{ei}$ 、任务完成周期  $t_i$ 。另外定义整个项目的完成周期为  $T$ 。任务时间分配的目的就是给出每个子任务开始执行的时间，即求出任务的触发时刻，使整个项目任务完成的时间最短，所以优化目标为  $\min(T)$ 。

分配到某一协同人员的子任务的触发时刻既与任务网络

图中它的前序任务集中各子任务的结束时间有关，也与其所在任务子集中的前序任务的结束时间有关，因此定义子任务的触发时刻为

$$t_{si} = \begin{cases} 0, & \text{若 } pre(i) = \\ \max t_{ej}, & \text{pre}(i) \end{cases}$$

其中， $pre(i)$  为子任务  $i$  在任务网络中及其所在任务子集中的所有前序任务的集合，则  $t_i = t_{ei} - t_{si}$ 。

由上述分析可知，若某个子任务没有前序子任务，则其开始时间取为 0，所以，该触发时刻加上完成周期后得到了任务的完成时刻，即  $t_{ei} = t_{si} + t_i$ 。所有任务中完成时刻值最大的任务为最晚完成的任务，其完成时刻在数值上等于整个项目的完成周期  $T$ ，即  $T = \max\{t_{ei}\}$  ( $i = 1, 2, \dots, n$ )。

因为协同设计子任务前后关联，所以子任务之间通过部分重叠的方式串在一起，对于任务网络中相邻的两个设计任务  $i$  和  $i-1$ ，两任务间的时间关系可以用并行度来描述。定义并行度为

$$C_i = 1 - (t_{si} - t_{s(i-1)}) / (t_{e(i-1)} - t_{s(i-1)})$$

由上述分析可知，与每个任务的完成时刻相关的两个因素是任务的执行时间和任务的并行度。下面根据文献[4]的研究成果来求解上述时间模型。

协同设计任务间存在耦合关系，这种耦合关系通过耦合矩阵  $CP$  表示  $CP = (cp_{ij})_{n \times n}$ ，其中  $cp_{ij}$  表示任务  $i$  和  $j$  之间的信息交互值。各子任务的耦合度可以通过子任务间的交互信息值来测定

$$cpd_{ij} = \begin{cases} cp_{ji} / (cp_{ji} + cp_{ij}) & (cp_{ji} < cp_{ij}) \\ 0 & (cp_{ji} \geq cp_{ij}) \end{cases} \quad (1)$$

协同设计并行度是从时间角度测定子任务间的联系，而耦合度是从交互信息度量子任务间的关系，二者虽然含义不同，但是从不同的角度对同一过程的数值描述，所以数值应该是相等的，即

$$cpd_{i+1} = C_{i+1}$$

代入二者的定义得

$$1 - (t_{s(i+1)} - t_{si}) / (t_{ei} - t_{si}) =$$

$$cp_{(i+1),i} / (cp_{i,(i+1)} + cp_{(i+1),i})$$

经过计算得

$$t_{s(i+1)} = (1 - cpd_{i+1}) \times t_i + t_{si} \quad (2)$$

由式(2)可以获得耦合关系的下一个子任务的起始时刻，从而计算出整个项目所需的执行时间。

## 3 机床产品协同设计任务分解模式及流程

参照机床厂多年实践经验，并根据机床的结构、功能以及各部分的连接关系，将机床划分为原动机、主轴变速箱、主轴箱、进给箱、床身、立柱、刀架、尾架、底板、溜板工作台等功能模块。因此，机床产品协同设计过程由一系列功能环节构成，每个功能环节由多个任务组成，整个开发过程可以表示为一个任务流。任务的执行在时序上有先后顺序，在逻辑上有一定的层次，在子任务的关联上有串联和并联关系，并且子任务之间在时间跨度上存在串行和并行的关系。机床协同设计任务的分解是自上向下细化任务，自底向上逐步实施，从而形成层次式递归化的任务流程分解模型，如图 2 所示。

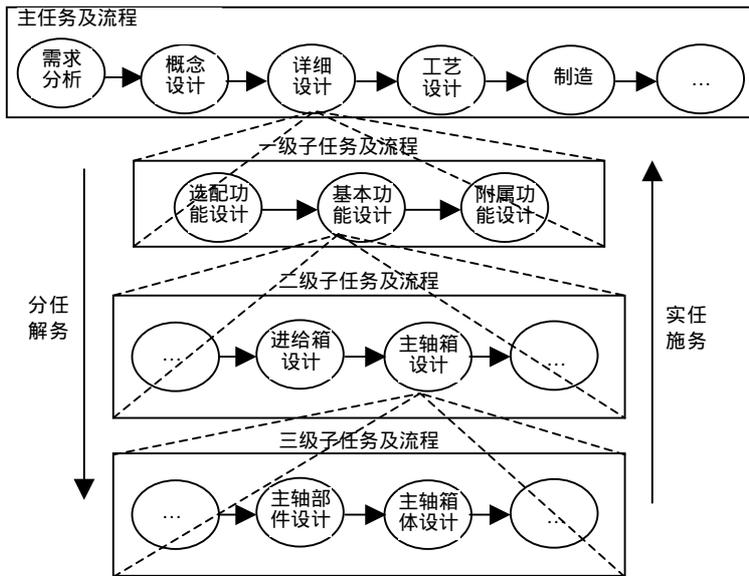


图2 机床层次式递归化任务分解模型及流程

该任务流程分解模型是一种基于任务动态分解的分层设计流程模型。该模型中产品设计过程的所有设计任务不要求预先全部定义好，除了最底层的叶任务之外，根任务和所有子任务都对应着一层设计流，构成该层设计流的所有环节都是该任务的子任务，任务的执行从叶任务开始，并从最底层的设计流开始逐层向上执行完毕。

#### 4 协同任务规划系统的运行实例

依据上述理论分析，建立了协同任务规划的体系结构并开发了原型系统<sup>[5]</sup>。从图3可以看出，该结构与面向机床产品的协同设计支持环境的体系结构相对应，是典型的基于Browser/Server模式的三层分布式体系结构。

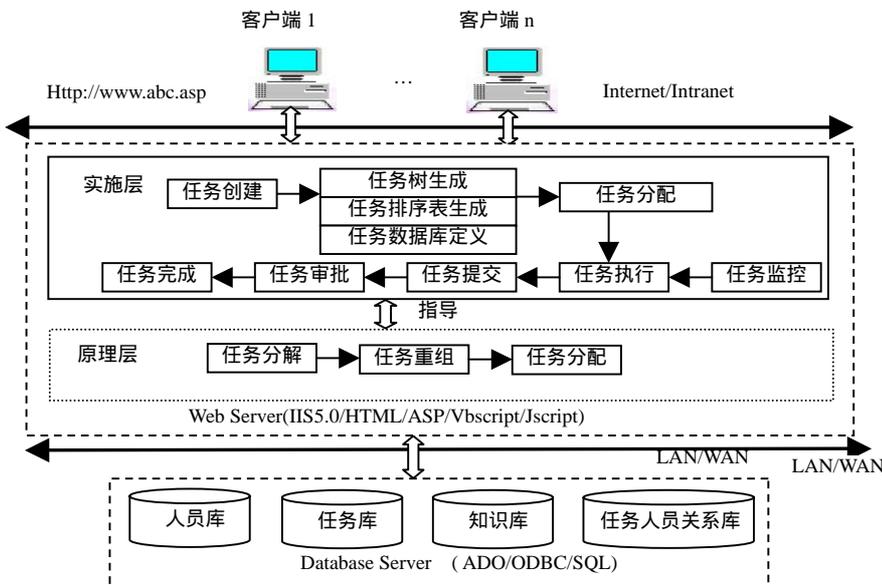


图3 协同任务规划功能模块的体系结构

第1层是客户层，用户通过浏览器调用所需的模块完成任务的规划和管理。

第2层是Web服务器层，用于存放任务创建、任务进度表生成、任务执行、任务监控、任务提交、任务审批等功能模块，这些功能模块的实现是以上面研究的任务分解、任务重组和任务分配原理与方法为指导的。作为中间层，它负责对用户需求的响应和反馈从数据库中查询的结果。

第3层是数据库服务器，用于存储任务数据、人员数据、机床产品数据、任务人员关系数据等。

系统开发采用ASP与Visual C++语言，应用服务器上的任务创建、任务重组和分配等功能模块都使用VC++6.0编程语言编写后制成ActiveX控件，通过ASP和ActiveX控件实现与客户端的交互。系统采用数据库技术实现机床协同设计任务规划流程的表达与控制。

原型系统的部分运行界面图略，工作流程如下

用户通过浏览器登录协同任务规划系统主窗口，创建设计任务，进行任务分解，生成设计任务树，根据任务重组原理生成任务排序列表，将任务树和相关任务信息存入任务数据库中。用户还可以根据需要对任务进行添加、删除和修改等操作。然后，根据任务分配原理将各个子任务分配给各个参与协同人员。协同人员接受任务并开始执行任务，将任务结果提交给审批者进行审批，若结果符合要求则任务完成。

#### 5 结论

本文研究了机床产品协同设计任务的具体分解方法，采用结构设计矩阵来判断任务分解的合理性并对任务进行重组，给出耦合任务集的解耦方法。研究了任务的人员分配和时间分配方法，确保将子任务在合适的时间分配给合适的协同小组。最后以机床产品为实际对象，提出了其层次式递归化任务分解模型，并给出了任务规划功能模块的网络实现流程。

#### 参考文献

- 1 杨波, 黄克正, 孙红卫. 面向并行工程的任务分配与规划[J]. 计算机集成制造系统, 2002, 8(7): 542-543.
- 2 周锐, 郁鼎文, 张玉峰. 基于BOM的任务分解求解策略[J]. 机械科学与技术, 2003, 22(2): 315-317.
- 3 肖位枢. 图论及其算法[M]. 北京: 航空工业出版社, 1993, 19-29.
- 4 周雄辉, 李祥, 阮雪榆. 注塑产品与模具协同设计任务规划算法研究[J]. 机械工程学报, 2003, 39(2): 113-117.
- 5 孟秀丽. 面向机床产品的协同设计支持环境关键技术的研究[D]. 南京: 东南大学, 2005.