

文章编号: 1001-0920(2012)04-0598-05

产品开发任务分配问题的多目标优化求解

武照云¹, 刘晓霞¹, 李 丽², 刘楠嶓¹

(1. 河南工业大学 机电工程学院, 郑州 450007; 2. 河南职业技术学院 汽车工程系, 郑州 450046)

摘要: 针对目前产品开发任务分配问题研究存在的不足, 给出了任务分配问题的数学描述和约束条件, 提出了任务分配模型中的相关矩阵, 并采用权重因子和极差变换法建立了多目标优化的目标函数. 针对任务分配过程的动态性和不确定性, 提出采用基于时序逻辑关系的动态分配蚁群算法进行优化计算, 并分析了该方法的优点, 给出了详细的算法步骤. 最后通过仿真实验验证了所提出方法的可行性和有效性.

关键词: 产品开发; 任务分配; 多目标优化; 蚁群算法

中图分类号: TP391

文献标识码: A

Multi-objective optimization for task assignment problem of product development

WU Zhao-yun¹, LIU Xiao-xia¹, LI Li², LIU Nan-bo¹

(1. School of Mechanical & Electrical Engineering, He'nan University of Technology, Zhengzhou 450007, China;

2. Automobile Engineering Department, He'nan Vocational & Technical College, Zhengzhou 450046, China.

Correspondent: WU Zhao-yun, E-mail: wzhaoyun@163.com)

Abstract: In view of the disadvantages of the current research on the task assignment problem of product development, the mathematical description and constraint conditions of the problem are presented. The related matrixes of the task assignment model are proposed. The objective function of multi-objective optimization is built based on weight factor and range transform method. According to the dynamic characteristic and uncertainty of task assignment process, the dynamic assignment oriented ant colony algorithm based on sequential logic is proposed. The advantages and steps of the algorithm are analyzed. Finally, a simulation test illustrates the feasibility and effectiveness of the proposed method.

Key words: product development; task assignment; multi-objective optimization; ant colony algorithm

1 引言

目前, 产品开发过程管理和规划已成为工程管理领域的热点问题, 对缩短产品开发时间、降低成本具有重要意义. 复杂的产品开发过程由一系列开发任务组成, 每个任务的完成效果都对整体效益产生影响. 在规划部署开发任务时, 应重点研究如何将各任务分配给合适的开发人员, 该问题是项目规划中需重点解决的关键环节. 目前, 国内外学者在任务分配与调度方面的研究大多集中在车间生产任务的调度方面, 对于产品开发任务分配的研究则较少^[1].

文献 [2] 给出了基于均衡-适度原则的任务分配数学模型; [3] 提出了根据优先级安排任务及分配资源的动态调度算法; [4] 讨论了开发人员的时间与能力最优利用问题, 提出了相应的数学规划法和启发式

方法; [5] 增加了员工对任务的感兴趣程度, 并给出了相应的任务分配算法; [6] 建立了并行产品开发过程中的任务分配模型, 并运用遗传算法求解; [7] 提出了基于多目标优化的产品协同开发任务调度理论, 并提出了基于混合微粒群算法的任务调度算法. 上述文献均从不同角度对任务分配问题进行了研究, 但在任务分配原则和数学模型方面只有 [6] 和 [7] 分析得比较全面, 其余文献均不够深入. [6] 的优化目标是缩短项目周期, 没有考虑开发成本与质量, 因此优化目标函数过于单一; [7] 进行了基于多目标优化的理论研究, 但其目标函数中忽略了由产品固有成本形成的市场价值, 因而不够完备.

本文重点探讨基于多目标优化的目标函数构造方法和基于时序逻辑关系的优化求解策略, 旨在解决

收稿日期: 2010-10-16; 修回日期: 2010-12-24.

基金项目: 河南省科技攻关计划项目(102102210127); 河南省教育厅自然科学研究计划项目(2008A460002).

作者简介: 武照云(1981—), 男, 副教授, 博士, 从事产品开发过程规划、数字化设计的研究; 刘晓霞(1975—), 女, 副教授, 博士, 从事作业车间优化调度的研究.

任务分配数学模型及优化求解的一些关键问题。

2 任务分配问题描述与约束条件

本文的研究对象是非耦合产品开发过程中的任务分配问题。该问题描述如下: 产品开发过程中包括 n 个设计任务, 无设计反馈。现有 m 个开发人员, 其对各任务的胜任情况都不相同, 而各自的工作效率、消耗成本和完成质量也不相同。每个任务只能由一名人员完成, 其完成时间、成本和质量也随着人员的不同而不同。要解决的问题是如何分配开发任务才能使项目在整体收益(综合时间、成本和质量)上达到最优。该问题需满足以下约束条件:

1) 一个人同一时刻只允许参与一项任务, 在不冲突的情况下, 允许一个人参与多项任务;

2) 任务执行完毕后, 它所占用的资源(开发人员)应立即释放;

3) 任务间的关系和执行顺序应按设计结构矩阵(DSM)进行。

3 任务分配问题的数学模型

3.1 任务分配模型的相关矩阵

为了对任务分配问题进行计算与优化, 必须对相关特征进行描述, 需引入以下矩阵变量:

1) 设计结构矩阵 $DSM_{n \times n}$ 。当任务 i 需要任务 j 提供信息时, $dsm_{ij} = 1$; 否则, $dsm_{ij} = 0$ 。

2) 人员胜任矩阵 $PBM_{m \times n}$ 。用来描述开发人员对各任务的胜任情况, 当人员 i 能够胜任任务 j 时, $pbm_{ij} = 1$; 否则, $pbm_{ij} = 0$ 。

3) 人员效率矩阵 $PEM_{m \times n}$ 。 pem_{ij} 表示人员 i 完成任务 j 时的工作效率, $pem_{ij} \in [0, 1]$ 。

4) 人员成本矩阵 PCM_m 。 pcm_i 表示开发人员 i 在单位时间内所消耗的成本。

5) 人员质量矩阵 $PQM_{m \times n}$ 。 pqm_{ij} 表示开发人员 i 完成任务 j 时能够达到的质量指标, $pqm_{ij} \in [0, 1]$ 。

6) 任务质量权重矩阵 TQM_n 。 tqm_i 表示任务 i 的完成质量对产品总体质量的影响权重, 且满足

$$\sum tqm_i = 1.$$

7) 任务工作量矩阵 TTM_n 。 ttm_i 表示将任务 i 分配给 $pem_{ki} = 1$ 的人员时所需要的时间。

8) 任务分配矩阵 $TAM_{m \times n}$ 。当开发人员 i 被分配至任务 j 时, $tam_{ij} = 1$; 否则, $tam_{ij} = 0$ 。

3.2 优化求解的目标函数

根据以上矩阵, 可以得到任务执行的实际周期矩阵 FTM_n 。 ftm_i 表示任务 i 实际执行时间, 即

$$ftm_i = \frac{ttm_i}{\sum_{k=1}^n (pem_{ki} \cdot tam_{ki})}, \quad i = 1, 2, \dots, n. \quad (1)$$

按文献[7]的方法, 根据 DSM 中的时序约束和 FTM 中的实际执行时间, 可以得到任务开始时刻矩阵 STM。根据 STM 和 FTM, 可计算出整个产品开发过程所需的时间

$$T = \max(stm_i + ftm_i) - \min(stm_i), \quad i = 1, 2, \dots, n. \quad (2)$$

根据 TAM 和 PCM, 可以计算出整个产品开发过程所需的人员成本

$$C = \sum_{j=1}^n \left(\sum_{i=1}^m (tam_{ij} \cdot pcm_i \cdot ftm_j) \right). \quad (3)$$

根据 TAM, TQM 和 PQM, 可以计算出整个产品开发过程的预计质量指标值

$$Q = \sum_{j=1}^n \left(\sum_{i=1}^m tqm_{ij} (tam_{ij} \cdot pqm_{ij}) \right). \quad (4)$$

该任务分配问题属于多目标优化问题, 优化目标为 $\min T, \min C, \max Q$ 。然而, 不可能存在一个分配策略使上述目标都满足。为此, 可采用引入权重的优化目标函数

$$\min F(Z) = T_w + C_w + Q_w. \quad (5)$$

式中各个指标的单位不同、量纲不同、数量级也不同, 不便于分析计算。因此, 本文采用极差变换方法对各指标进行标准化处理, 处理后的目标函数可表示为

$$\begin{aligned} \min F(Z) = & w_T \frac{T - T_{\min}}{T_{\max} - T_{\min}} + w_C \frac{C - C_{\min}}{C_{\max} - C_{\min}} + \\ & w_Q \frac{Q_{\max} - Q}{Q_{\max} - Q_{\min}}; \\ \text{s.t. } & g_1(Z) = T_{\min} - T \leq 0, \\ & g_2(Z) = T - T_{\max} \leq 0, \\ & g_3(Z) = C_{\min} - C \leq 0, \\ & g_4(Z) = C - C_{\max} \leq 0, \\ & g_5(Z) = Q_{\min} - Q \leq 0, \\ & g_6(Z) = Q - Q_{\max} \leq 0, \\ & h(Z) = w_T + w_C + w_Q - 1 = 0. \end{aligned} \quad (6)$$

式中: w_T, w_C, w_Q 分别为指标 T, C, Q 的权重, 其值可采用 AHP 法或熵值法确定^[8]; $T_{\min}, T_{\max}, C_{\max}, Q_{\min}, Q_{\max}$ 分别为满足要求的各种分配方案中时间、成本和质量的最小值和最大值, 其实质是该任务分配问题分别对时间 T , 成本 C 和质量 Q 的单目标优化极小值与极大值, 所以这 6 个极值的求解便构成了对各自指标的单目标优化问题, 也需采用智能算法或进化算法等进行计算。可见, 该问题的多目标优化需以单目标优化为前提和基础。因此, 任务分配问题的多目标优化求解过程如下:

- 1) 先分别对上述 6 个指标值进行单目标优化;
- 2) 得到式 (6) 后, 再进行多目标优化计算.

4 任务分配模型的优化求解

4.1 优化算法概述

在任务分配问题的优化求解策略上, 以往学者大多采用静态分配的方法, 即任务分配次序是固定的, 并不遵循任务执行的时序关系, 也不考虑 DSM 中的执行约束和资源约束, 分配结束后再进行冲突检测和处理. 显然, 该算法原理简单, 但必然产生大量不满足要求的劣解, 同时也使解空间变得异常庞大, 导致运算时间长、收敛缓慢.

针对上述问题, 本文提出基于时序逻辑关系的动态分配蚁群算法. 该算法以蚁群算法为基础, 并严格按照由任务时序关系确定的次序进行分配. 任务时序关系由 DSM 中的执行约束和任务执行时间决定, DSM 中的执行约束是确定的, 但任务实际执行时间却是未知的, 只有在任务-人员分配后才能确定, 从而每一个任务的分配结果都会对后续任务的时序关系产生影响. 因此, 该算法在完成每一个任务分配时, 都会根据该分配结果实时调整后序任务的时序关系, 进而动态确定后续任务的分配次序. 由于在算法中融入了任务时序关系和资源约束, 该算法产生的解都能满足要求, 同时减小了算法的搜索空间, 使运算时间大大减少, 收敛速度加快.

4.2 求解流程与算法步骤

由 3.2 节分析可知, 任务分配模型的优化求解流程如图 1 所示.

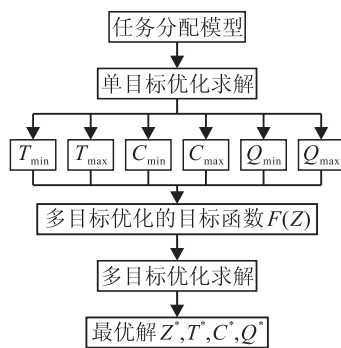


图 1 优化求解流程图

由图 1 可知, 整个流程分为 2 部分: 1) 单目标优化求解; 2) 多目标优化求解. 在单目标优化和多目标优化过程中, 均采用本文提出的基于时序逻辑关系的动态分配蚁群算法, 其区别仅在于各自的目标函数不同. 限于篇幅, 这里仅给出多目标优化时的算法步骤.

首先引入以下变量: NC 为迭代次数; u 为蚁群数量; n 为任务数量; m 为人员数量; $\text{PAM}_{m \times 1}$ 为人员可用状态矩阵; $\text{TaskState}_{n \times 1}$ 为任务执行状态矩阵;

$\text{ToWork}_{n \times 1}$ 为任务剩余时间矩阵; $\text{Info}_{m \times n}$ 为信息素矩阵; $\text{Path}_{u \times n}$ 为蚁群分配方案矩阵; $\text{Result}_{u \times 5}$ 为蚁群分配结果矩阵; $\text{PathBest}_{1 \times n}$ 为全局最优方案向量; $Y_{1 \times 4}$ 为全局最优解向量; $\text{RCourse}_{\text{NC} \times 4}$ 为迭代过程矩阵.

Step 1: 初始化变量. 设定 NC, u 等参数, 迭代次数 $\text{nc} = 1$, 生成任务-人员周期矩阵 FTM' , 有

$$\begin{aligned} \text{ftm}'(i, j) &= \text{ttm}(j) / \text{pem}(i, j), \\ i &= 1, 2, \dots, m, j = 1, 2, \dots, n. \end{aligned} \quad (7)$$

Step 2: 开始第 nc 次迭代计算. Path 和 Result 均置为零矩阵, $k = 1$.

Step 3: 遍历蚁群. 对于蚂蚁 k , PAM 置为 1 矩阵, TaskState 中所有元素置为 'wait', ToWork 置为零矩阵, 时间变量 $T = 0$, 成本变量 $C = 0$, 质量变量 $Q = 0$.

Step 4: 蚂蚁分配. 遍历 TaskState , 若满足:

状态 = 'wait' and 无前继任务 or

所有前继任务状态 = 'complete',

则更新该任务 T_i 状态, 即 $\text{taskstate}(i) = \text{'ctive'}$. T_i 进入执行状态, 可以对其进行人员分配. 蚂蚁 k 在分配前, 先根据 PBM 和 PAM 确定可选择分配集 allowed_k . 如果 allowed_k 非空, 则计算启发式因子 η , 再结合 Info 与节点选择规则, 确定分配的人员 E_j , 并执行如下操作: 更新 Path ; 令 $\text{pam}(j) = 0$ (将人员状态置为不可用); 计算任务 T_i 的执行时间 $t_i = \text{ftm}'(j, i)$; 计算 T_i 的执行成本 $c_i = \text{pcm}(j) \cdot \text{ftm}(j, i)$; 计算 T_i 的实际质量指标值 $q_i = \text{pqm}(j, i) \cdot \text{tqm}(i)$; 更新 ToWork 中对应任务所需的完成时间 $\text{towork}(i) = t_i$; 成本与质量值累加, 即 $C = C + c_i$, $Q = Q + q_i$. 如果 allowed_k 为空, 则该蚂蚁无法进行分配操作, 转 Step 7.

Step 5: 触发事件. 在 TaskState 中遍历 active 状态的任务, 找到时间最短者 T_l . 以 T_l 执行完毕为事件触发点, 执行如下操作: 修改 TaskState 中 T_l 的状态

$$\text{taskstate}(l) = \text{'complete'};$$

时间变量累加

$$T = T + \text{towork}(l);$$

修改其他 active 状态任务的剩余时间

$$\text{towork}(i) = \text{towork}(i) - \text{towork}(l);$$

重置 T_l 的剩余时间

$$\text{towork}(l) = 0;$$

修改 PAM 中对应人员状态以释放资源 $\text{pam}(l) = 1$.

Step 6: 判断蚂蚁 k 是否满足分配终止条件. 遍历 TaskState , 若所有元素均为 'complete', 则转 Step 7; 否则, 转 Step 4.

Step 7: 保存蚂蚁 k 的分配结果. 如果蚂蚁 k 成功

完成分配, 则计算目标函数值, 并且将成功标识“1”, T, C, Q, Z 存入 **Result** 中; 若分配失败, 则存入失败标识“0”。

Step 8: 判断蚁群中所有蚂蚁是否都已完成分配。若 $k < u$, 则令 $k = k + 1$, 转 **Step 3**; 若 $k = u$, 则转 **Step 9**。

Step 9: 更新全局最优解。找出本次迭代的最优解向量 Y' 及对应的最优方案向量 $PathBest'$, 将 Y' 中各值存入 **RCourse** 中。若 $Y'(4) < Y(4)$, 则更新全局最优解向量及全局最优方案向量

$$Y = Y', PathBest = PathBest'$$

Step 10: 更新信息素。根据信息素更新规则及最大-最小蚂蚁系统(MMAS)方法对信息素矩阵进行 **Info** 更新, 即

$$Info(i, j) = (1 - \rho) \cdot Info(i, j) + \Delta\tau_{ij}^{best}. \quad (8)$$

Step 11: 判断算法迭代是否结束。若 $nc < NC$, 则令 $nc = nc + 1$, 转 **Step 2**; 若 $nc = NC$, 则迭代结束, 输出计算结果。

5 仿真实验

以图2所示某产品开发过程的任务网络为例, 验

证本文方法的可行性和有效性。

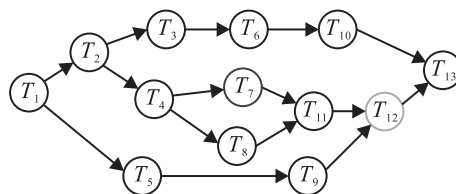


图2 仿真实验任务网络

图2中: 任务数量 $n = 13$, 人员数量 $m = 20$ 。通过模糊层次分析法并结合专家经验, 对所涉及的模型数据进行评估和量化。限于篇幅, 将 **PEM**, **PCM** 与 **TQM** 合并表示, 如表1所示 (T 为任务, E 为人员), 其他矩阵数据略。

在优化求解中, 选取适当的算法参数是一个很关键的问题。信息素权重 α 代表信息素的重要程度; 启发因子权重 β 则代表了启发式信息的重要程度; 信息素挥发系数 ρ 表示信息素的挥发程度, 其大小会影响算法的搜索效果和收敛速度。本文通过大量实验确定相应的参数, 图3是上述参数取不同值时对单目标优化 (T_{min}) 计算结果的影响曲线。

表1 仿真实验初始数据

PEM	T ₁	T ₂	T ₃	T ₄	T ₅	T ₆	T ₇	T ₈	T ₉	T ₁₀	T ₁₁	T ₁₂	T ₁₃	PCM
E ₁	0.8	0	0.2	0.2	0	0	0.5	0	0.3	0	0.6	0	0.5	40
E ₂	0.6	0.5	0	0.4	0.7	0.3	0	0.6	0	0.7	0	0.5	0	35
E ₃	0.7	0.5	0.5	0.5	0	0	0.6	0.9	0	0.4	0	0.3	0	45
E ₄	0	0.8	0	0.5	0	0	0.7	0.8	0	0.6	0.2	0	0.7	50
E ₅	0.9	0	0	0	0.8	0.6	0	0	0.6	0	0.9	0	0.9	70
E ₆	0	0.3	0.7	0	0.7	0.8	0	0	0.4	0	0.3	0.5	0.2	60
E ₇	0	0.7	0.6	0.6	0.9	0.2	0.8	0	0	0.8	0.5	0.3	0	65
E ₈	0	0.6	0.8	0.7	0	0	0.5	0.4	0	0	0	0.9	0	47
E ₉	0	0.5	0	0.9	0.5	0	0	0.6	0.5	0.5	0	0.2	0	60
E ₁₀	0.3	0	0.7	0	0.6	0.5	0	0	0.8	0.6	0	0	0.6	55
E ₁₁	0	0.3	0.9	0.2	0.7	0	0.4	0.2	0	0	0.1	0	0.5	36
E ₁₂	0	0.8	0.5	0	0	0	0.7	0	0.9	0	0.4	0.5	0	46
E ₁₃	0.6	0.6	0.2	0.6	0	0.5	0	0	0.6	0.6	0.6	0.6	0.3	45
E ₁₄	0	0	0	0	0.3	0.7	0	0.3	0.7	0.7	0	0	0	65
E ₁₅	0.2	0	0.7	0	0.2	0.3	0	0.5	0	0	0.3	0	0	50
E ₁₆	0.4	0	0	0.2	0	0	0.9	0	0.2	0	0	0	0	40
E ₁₇	0.7	0	0	0.4	0	0	0.8	0	0.4	0.3	0	0.3	0.8	43
E ₁₈	0	0.9	0	0	0	0	0.7	0	0	0.5	0	0	0	55
E ₁₉	0	0.7	0	0	0	0.8	0	0.8	0.6	0	0.6	0	0.2	61
E ₂₀	0	0	0	0.6	0	0.7	0	0.5	0	0	0	0.5	0.6	36
TQM	0.06	0.07	0.07	0.08	0.09	0.07	0.08	0.07	0.08	0.09	0.09	0.06	0.09	

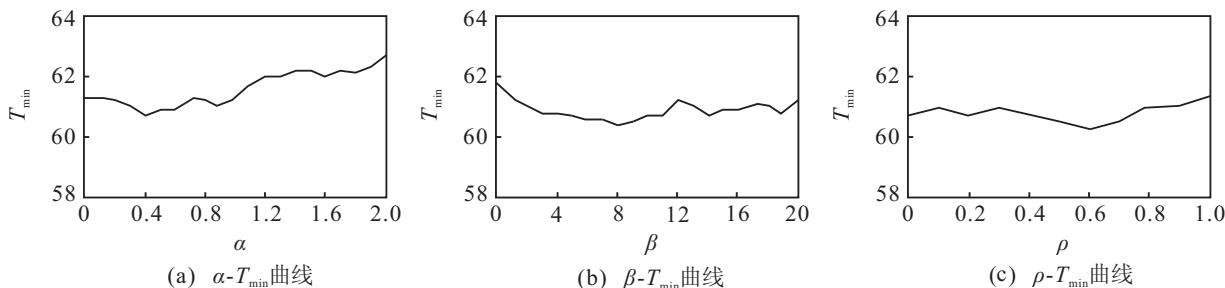


图3 不同参数取值对计算结果的影响

从图 3 可以看出, $\alpha = 0.4, \beta = 8, \rho = 0.6$ 是比较合理的选择。

取蚁群数量 $u = 60$, 运用 Matlab 7.0 编制算法程序, 实现其求解过程。首先对各单目标进行优化求解, 算法程序运行 50 次, 每次运行迭代 500 次, 所得计算结果如表 2 所示; 然后进行多目标优化求解, 取权重值 $w_T = 0.5, w_C = 0.2, w_Q = 0.3$, 可得目标函数表达式如下:

$$\min Z = 0.5 \frac{T - 60.135}{213.691} + 0.2 \frac{C - 5301.3}{18515.7} + 0.3 \frac{0.864 - Q}{0.645} \quad (9)$$

表 2 单目标优化的计算结果

优化目标	最优解	达到最优解所需的代数平均值
T_{\min}	60.135	345
T_{\max}	273.826	353
C_{\min}	5301.3	337
C_{\max}	23817	346
Q_{\min}	0.223	352
Q_{\max}	0.868	361

同样, 算法程序运行 50 次, 每次运行迭代 500 次, 可得到目标函数最优解为 0.096, 对应的总时间 $T = 76.436$ 天, 总成本 $C = 7085.3$ 元, 质量指标值 $Q = 0.781$ 。任务-人员分配结果如表 3 所示。

表 3 多目标优化的任务-人员分配结果

任务	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}	T_{12}	T_{13}
人员	E_{17}	E_{18}	E_3	E_{13}	E_2	E_5	E_{17}	E_4	E_{19}	E_2	E_1	E_8	E_5

应用本文方法与基于静态分配的混合微粒群算法(混合 PSO)针对多目标优化的求解进行对比实验, 实验结果如表 4 所示。

表 4 算法性能对比实验结果

算 法	最优解	达到最优解的代数平均值	达到最优解的时间平均值
混合 PSO	0.096	382	51.66
本文算法	0.096	335	42.72

由表 4 可以看出, 本文算法具有较高的搜索较优解的能力, 经较少的次数便可以找到最优解, 节约了运算时间。

6 结 论

本文研究了产品开发任务分配的多目标优化问题, 阐述了任务分配问题的本质和约束条件, 建立了任务分配的数学模型和多目标优化函数, 提出了基于时序逻辑关系的动态分配蚁群算法, 并给出了详细的

算法步骤。算例表明, 采用所提出的算法可以对产品开发任务分配问题进行有效规划, 对同类问题的多目标优化具有一定的参考意义。

参考文献(References)

- [1] Wang J. Constraint-based schedule repair for product development projects with time-limited constraints[J]. Int J of Production Economics, 2005(95): 399-414.
- [2] 杨波, 黄可正, 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): 542-545. (Yang B, Huang K Z, Sun H W. Assignment and planning of task supporting concurrent engineering[J]. Computer Integrated Manufacturing Systems, 2002, 8(7): 542-545.)
- [3] 黄洪钟, 李丽, 刘伟. 并行工程中设计任务的动态调度[J]. 机械工程学报, 2002, 38(增): 164-167. (Huang H Z, Li L, Liu W. Dynamic scheduling of design tasks in concurrent engineering[J]. Chinese J of Mechanical Engineering, 2002, 38(S): 164-167.)
- [4] Bassett M. Assigning projects to optimize the utilization of employees' time and expertise[J]. Computers & Chemical Engineering, 2000(24): 1013-1021.
- [5] 陈传波, 赵伟伟. 一种自主 workflow 任务分配策略[J]. 华中科技大学学报: 自然科学版, 2005, 33(6): 20-22. (Chen C B, Zhao W W. Strategy for a task assignment of workflow system[J]. J of Huazhong University of Science and Technology: Nature Science Edition, 2005, 33(6): 20-22.)
- [6] 李玉家, 胡宗武, 金焯. 并行产品开发过程中的任务分配问题研究[J]. 中国机械工程, 2002, 13(7): 46-49. (Li Y J, Hu Z W, Jin Y. Research on task distribution in concurrent product development[J]. China Mechanical Engineering, 2002, 13(7): 46-49.)
- [7] 蒋增强. 产品协同开发过程管理的关键技术研究[D]. 合肥: 合肥工业大学机械与汽车工程学院, 2006. (Jiang Z Q. Research on key techniques for process management of product cooperative development[D]. Hefei: School of Mechanical and Automotive Engineering, Hefei University of Technology, 2006.)
- [8] 董景峰, 王刚, 吕民, 等. 基于改进蚁群算法的多供应商选择问题求解[J]. 计算机集成制造系统, 2007, 13(8): 1639-1644. (Dong J F, Wang G, Lv M, et al. Multi-supplier selection problem solution based on improved ant colony algorithm[J]. Computer Integrated Manufacturing Systems, 2007, 13(8): 1639-1644.)