并行工程产品开发过程定量化建模与计划制订①

汪　峥，严洪森，刘霞岭，宋文忠
（东南大学自动化研究所，南京 210096）

摘要：由于现有的一些并行工程定量化模型中修改设计微循环的细节特征描述得还不够充分，本文提出了一种新的基于产品-工艺设计活动对网络的定量化模型来描述并行工程产品开发过程，并且给出了产品-工艺设计活动对平均持续时间与产品或工艺设计活动资源占用率的计算方法. 在此基础上，将并行工程产品开发过程的计划制订问题建模成一个有资源约束的项目调度问题. 与其它此类问题不同的是，在本文中，分配给产品开发项目的各类资源的数量不是事先给定的，而是与最优产品开发计划一起获得的，所以这是一个资源分配与计划制订的集成优化问题. 本文提出了一种新的基于分枝定界的算法来解决此问题并引入一个启发式规则来提高算法的搜索效率.

关键词：并行工程；过程建模；修改设计微循环；产品-工艺设计活动对网络；产品开发计划；有资源约束的项目调度问题；分枝定界算法

中图分类号：TB114.1　　**文献标识码**：A　　**文章编号**：1007-9807(2000)04-0046-14

0 引　言

并行工程(concurrent engineering)产品开发模式要求产品结构设计与相关的工艺过程设计并行进行，并且要求产品生命周期的各种因素在产品开发过程的一开始就要被考虑到[1]. 在并行工程产品开发模式中，由于实施了小组化工作方式和及时地进行产品设计结果的信息预发布，工艺设计人员能够尽早发现产品设计中的错误，因而由此导致的设计迭代是一个"修改设计微循环"[2,3]. 这是并行工程与传统的串行产品开发模式之间最显著的差别.

文献中已有许多种并行工程的模型，大致包括以下几类：(1) 定性模型，例如 Gantt 图模型[4]，IDEF3 模型[5]，多协作知识源范式模型[6]，集成多视图模型[7,8] 等；(2) Petri 网模型[2,3,9]，用于对产品开发过程进行描述、性能分析和仿真；(3) 定量化模型[10-15]，其研究重点在两个方面：

其一是上下游活动之间的重叠. Krishnan，Eppinger 和 Whitney 研究了两个成对的产品开发活动，把上游设计看作一个进化过程并分析了下游设计对这一进化过程的敏感性[10]. 在此基础上，他们研究了上下游活动重叠的策略及设计迭代的特征. Loch 和 Terwiesch 考虑了并行工程中的不确定性及上下游活动间的相互依赖性，据此讨论了如何确定上下游活动间的重叠程度及上下游设计相互交流的频繁程度[11]，并且进一步指出不确定因素解决得越早，产品开发项目将从活动重叠中受益越大[12].

其二是设计迭代的特征. Krishnan，Eppinger 和 Whitney，Loch 和 Terwiesch 及 Ha 和 Porteus 等着重研究了并行产品开发过程中进行设计评审的最优策略[10,11,13]，但他们只强调了上游设计修改对下游设计的影响而下游设计发现上游设计错误的能力则未被充分考虑. Smith 和 Eppinger 在设计结构矩阵(design structure matrix)及其推广工

① 收稿日期：1999-11-01；修订日期：2000-04-15.
基金项目：国家自然科学基金资助项目(69884001)；江苏省自然科学基金资助项目(BK95047506).
作者简介：汪 峥(1973-)，男，江苏江宁人，博士生.

作转换矩阵(work transformation matrix)的基础上，分别针对设计活动顺序和平行进行的情形，研究了设计迭代的特征，并进行了设计模态分析[14、15]. 但是，并行工程中的设计活动既不完全是顺序的又不完全是平行的，所以其设计迭代与 Smith 和 Eppinger 讨论的情形有很大不同.

为此，本文将提出另外一种基于产品 - 工艺设计活动对网络的并行工程产品开发过程定量化模型作为已有定量化模型的补充. 与文[10]—[15]相比，本文的不同之处在于：(1) 将研究修改设计微循环的细节，强调下游设计活动发现上游设计的错误以及由此导致的设计迭代过程；(2) 讨论上下游设计活动之间存在重叠的情形，而不是顺序或平行的情形.

建立并行工程产品开发过程定量化模型的目的在于据此制订一个合理的产品开发计划来安排和协调产品开发活动，以求在有限资源条件下以最短的时间和最好的质量开发出用户满意的产品. 本文研究的产品开发计划制订问题可这样描述：把有限的资源在适当的时间分配给有关的产品开发活动，使得整个产品开发项目的完成时间尽可能短，并使得平均资源利用率不低于一给定值.

这是一个具有资源约束的项目调度问题(resource-constrained project scheduling problem)，文[16]给出了此类问题近来研究工作的一个综述. 此类问题常被建模为 0-1 整数规划问题，大量的整数决策变量使问题变得困难. Demeulemeester 等提出了一种分枝定界算法来解决单项目多资源约束的项目调度问题[17]. Icmeli 等则通过放松对活动时间的整数要求来减少整数决策变量的数目，以使问题简化[18]. Leachman 等考虑了资源利用强度，引入了变强度活动的概念并提出了一种强度分配的启发式算法[19]. Cheng 等研究了如何用进化算法来解决这类问题[20]. Belhe 等则着重研究了产品开发项目中设计活动的调度方法[21,22]. 在通常的这类问题中，分配给项目的资源数量是已知的或事先确定的. 但在本文要解决的问题中，分配给产品开发项目的各类资源数量不是事先确定的，而是在制订产品开发计划的同时获得的. 因此这是一个资源分配与计划制订的集成优化问题，本文将给出一种新的基于分枝定界的算法来解决此问题，并且引入一个启发式规则来提高算法的搜索效率.

1　并行工程产品开发过程的定量化建模

1.1　产品 - 工艺设计活动对网络

并行工程要求产品生命周期内的各种因素在产品开发的一开始就被考虑到. 通常，产品生命周期包含多个阶段，如需求分析，产品设计，工艺设计，原形制造，测试与维护等. 为使问题简化，本文仅考虑产品设计与工艺设计两个阶段，它们并行进行，相互重叠并通过修改设计微循环相互影响. 给出如下定义：

定义 1　产品设计活动与其相关的工艺设计活动被修改设计微循环联系在一起形成产品 — 工艺设计活动对(见图 1).

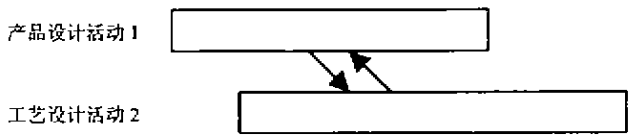

图 1　产品 - 工艺设计活动对

并行工程模式下产品开发过程可以这样来描述：首先进行产品结构总体方案设计和总体装配工艺设计，然后进行分系统结构方案设计和分系统装配工艺设计，最后是零部件结构与加工工艺设计. 这一过程可用由产品 - 工艺设计活动对构成的网络来描述.

定义 2　由产品 - 工艺设计活动对构成的网络称为产品 - 工艺设计活动对网络(见图 2).

图 2 表示的是某种汽车的传动系统开发过程的产品 — 工艺设计活动对网络模型. 其中，活动对 1 表示传动系统总体结构与装配工艺方案设计；活动对 2,3,4,5 分别表示传动轴分系统、速度

调节分系统，方向调节分系统和制动分系统的结构与装配工艺方案设计；活动对 6,7,8,9 分别表示上述四个分系统的零部件结构与工艺详细设计. 最后一个活动 10 表示产品开发项目的结束. 这里，假定此传动系统的开发过程是一个改型设计过程而非全新设计过程，产品结构设计的错误可完全被相应的装配或工艺设计所发现. 因而可进一步假定一但总体方案确定，就不再改变，亦即没有从分系统到总体的设计修改迭代过程. 同样，也假定没有从零部件到分系统或总体的设计修改迭代过程.

1.2 产品 - 工艺设计活动对平均持续时间及产品和工艺设计活动资源占用率的计算方法

给定一产品设计活动 1 及其相关的工艺设计活动 2，在并行工程模式下，它们的行为可描述如下(图 3). 假定活动 1 可被分解为 n 个子活动，记作 $(1,u)$，$u = 1,2,\cdots,n$. 当一个子活动完成时，其设计结果就传给活动 2 的设计者. 活动 2 也只能进行到一定阶段，因为活动 1 尚未全部完成. 因此，活动 2 也可分解为 n 个阶段 $(2,h)_1$，$h = 1,2,\cdots,n$ 并且它们与活动 1 的子活动一一对应. 活动 2 的每个阶段可进一步被分解为若干子活动，并令 m_h 表示第 h 个阶段的子活动数. 则活动 2 的子活动可记为 $(2,v)_2$，$v = 1,2,\cdots,m$，这里 $m = \sum_{h=1}^{n} m_h$. 当活动 2 的某个子活动结束时，产品设计活动 1 的错误有可能被发现. 若被发现，则正常的产品和工艺设计活动就停止，修改设计活动即可开始. 修改结束后，正常设计活动恢复.

本文视活动或子活动的正常设计时间与正常设计工作量为同一概念. 令随机变量 d_1 和 d_2 分别表示产品设计活动 1 和工艺设计活动 2 的正常设计时间. 设 d_1 服从 β 分布. 令活动 1 的最乐观，最可能和最悲观完成时间分别为 a_1，c_1 和 b_1，令 $f_1(x)$ 表示活动 1 正常设计时间的分布密度函数，其参数为 p_1 和 q_1，则

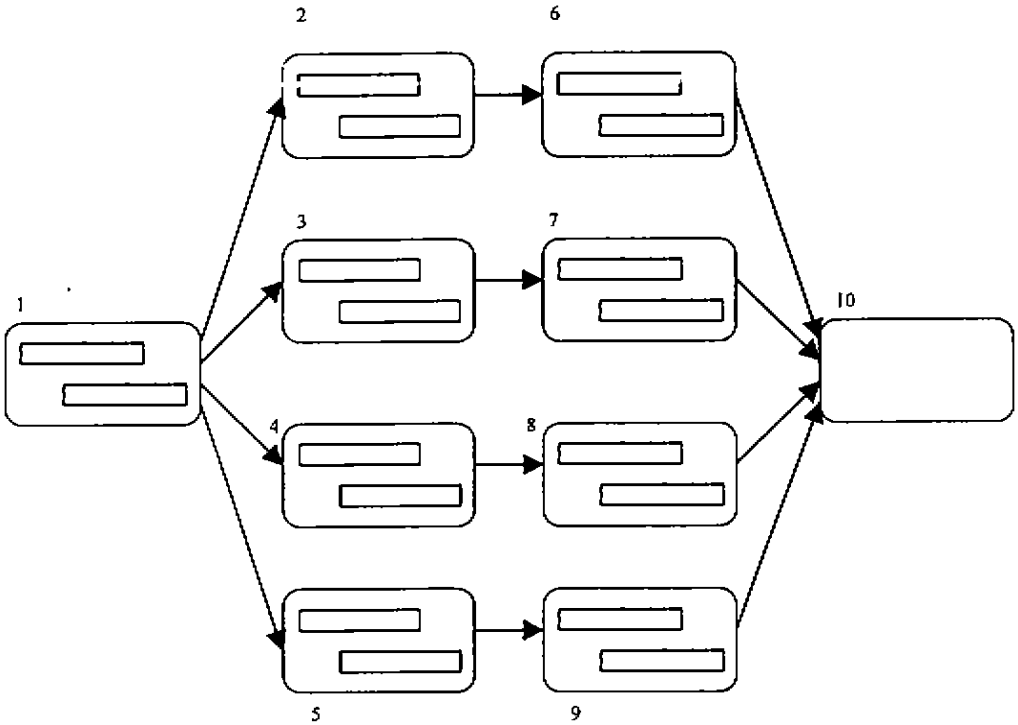

图 2 产品 - 工艺设计活动对网络

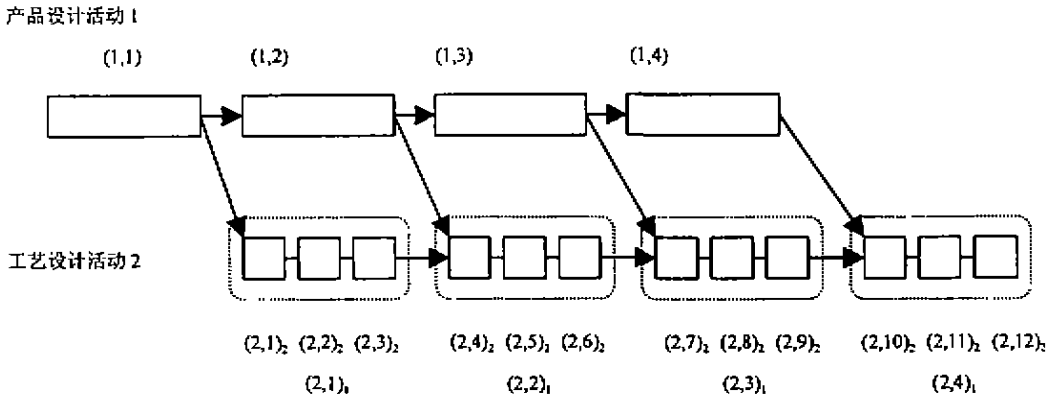

图 3　产品 - 工艺设计活动对的分解

$$f_1(x)=[(x-a_1)^{p_1-1}(b_1-x)^{q_1-1}]/[(b_1-a_1)^{p_1+q_1-1}\beta(p_1,q_1)],\quad a_1\leqslant x\leqslant b_1 \tag{1}$$

由(1)可见,q_1 可由 a_1、c_1、b_1 和 p_1 共同确定. 不考虑修改设计微循环,活动 1 正常设计时间均值为

$$\begin{aligned}\bar{d}_1&=Ed_1\\&=\int_{-\infty}^{+\infty}xf_1(x)\mathrm{d}x\\&=\int_{a_1}^{b_1}xf_1(x)\mathrm{d}x\\&=a_1+[(b_1-a_1)p_1]/(p_1+q_1)\end{aligned} \tag{2}$$

令子活动$(1,u)$的正常设计工作量为 $d_{1u}=\alpha_{1u}d_1$,$u=1,2,\cdots,N$,这里 $\alpha_{1u}\in[0,1]$ 并且 $\sum_{u=1}^{n}\alpha_{1u}=1$ 成立. 则易证随机变量 d_{1u} 服从参数为 $\alpha_{1u}a_1$、$\alpha_{1u}b_1$、p_1 和 q_1 的 β 分布,并且

$$\begin{aligned}\bar{d}_{1u}&=Ed_{1u}\\&=\alpha_{1u}a_1+[(\alpha_{1u}b_1-\alpha_{1u}a_1)p_1]/(p_1+q_1)\\&=\alpha_{1u}\bar{d}_1\end{aligned} \tag{3}$$

令 d'_{1u} 表示子活动$(1,u)$的修改时间. 设 d'_{1u} 服从具有参数 a'_{1u}、c'_{1u}、b'_{1u}、p'_{1u} 和 q'_{1u} 的 β 分布. 令 $a'_{1u}=0$. 这意味着在最乐观情况下,$(1,u)$的修改工作几乎不需花多少时间;令 $b'_{1u}=d_{1u}$,这意味着在最悲观情况下,$(1,u)$的修改工作要花与正常设计几乎相等的时间;令$(1,u)$的最可能修改时间 $c'_{1u}=\gamma_{1u}b_{1u}=\gamma_{1u}d_{1u}$,这里 $0\leqslant\gamma_{1u}\leqslant1$. 假定对所有子活动$(1,u)$,$u=1,2,\cdots,n$,$p'_{1u}=p'_1$,$q'_{1u}=q'_1$ 及 $\gamma_{1u}=\gamma_1$ 成立. 则易证子活动$(1,u)$的修改时间均值为

$$\bar{d}'_{1u}=E[E(d_{1u}|d_{1u}]=\bar{d}_{1u}p_1/(p'_1+q'_1) \tag{4}$$

假定 d_2 也服从 β 分布,则它的均值 $\bar{d}_2$,各子活动正常设计时间均值 $\bar{d}_{2v}$ 及子活动修改设计时间均值 $\bar{d}'_{2v}$,$v=1,2,\cdots,m$,也可用同样的方法得到.

现考虑修改设计微循环. 假设:(1)仅考虑发生于产品设计活动 1 并被发现于工艺设计活动 2 的错误;发生并被发现于同一活动 1(或 2)的错误的修改时间被认为是正常设计时间的一部分;(2)发生于活动 1 的某子活动$(1,i)$的错误能够也一定会在发生在$(1,i)$之后的活动 2 的某阶段(如$(2,k)_1$,$k=i,i+1,\cdots,n$)被发现;(3)在活动 2 的某一阶段$(2,k)_1$,仅有发生在它之前的活动 1 的子活动(如$(1,i)$,$i=1,2,\cdots,k$)的错误能被发现;(4)修改设计微循环仅发生于活动 2 的某子活动结束之后及下一子活动开始之前,且属于活动 2 同一阶段的各子活动发现活动 1 的错误的概率相等.

令 B_{ik}(或 $\overline{B}_{ik}$)表示子活动$(1,i)$的错误在$(2,k)_1$被发现(或未被发现)这一事件,$k=i,\cdots,n$. 于是概率

$$\begin{aligned}&P(\overline{B}_{ii}\overline{B}_{i,i+1}\cdots\overline{B}_{i,k-1}B_{ik})\\&=P(B_{ik}|\overline{B}_{ii}\overline{B}_{i,i+1}\cdots\overline{B}_{i,k-1})\cdot P(\overline{B}_{i,k-1}|\overline{B}_{ii}\overline{B}_{i,i+1}\cdots\overline{B}_{i,k-2})\cdots P(\overline{B}_{i,i+1}|\overline{B}_{ii})P(\overline{B}_{ii})\\&=P(B_{ik}|\overline{B}_{ii}\overline{B}_{i,i+1}\cdots\overline{B}_{i,k-1})[1-P(B_{i,k-1}|\overline{B}_{ii}\cdot\overline{B}_{i,i+1}\cdots\overline{B}_{i,k-2}]\cdots[1-P(\overline{B}_{i,i+1}|\overline{B}_{ii})]\cdot[1-P(B_{ii})]\end{aligned} \tag{5}$$

令 $P(B_{ii})=p$ 及条件概率 $P(B_{ih}|\overline{B}_{ii}\overline{B}_{i,i+1}\cdots\overline{B}_{i,h-1})=p$,$h=i+1,\cdots,n-1$. 由假设(2),可令 $P(B_{in}|\overline{B}_{ii}\overline{B}_{i,i+1}\cdots\overline{B}_{i,n-1})=1$. 可用矩阵 $P^{(1)}=$

$(p_{ik}^{(2)})_{n\times n}$ 来描述上述概率，这里

$$p_{ik}^{(1)}=\begin{cases}0, & \text{若 } k<i\\ P(B_{ii})=p, & \text{若 } k=i\\ P(\overline{B}_{ii}\overline{B}_{i,i+1}\cdots\overline{B}_{i,k-1}B_{ik})=(1-p)^{k-i}p, & \text{若 } k=i+1,\cdots,n-1\\ P(\overline{B}_{ii}\overline{B}_{i,i+1}\cdots\overline{B}_{i,n-1}B_{in})=(1-p)^{n-i}, & \text{若 } k=n\end{cases}\tag{6}$$

令 C_{ik} 表示由阶段 $(2,k)_1$ 的工艺设计人员发现发生于子活动 $(1,i)$ 的错误而导致的修改设计微循环发生这一事件. 令 C_{ik} 发生的概率为 $P(C_{ik})=P(\overline{B}_{11}\cdots\overline{B}_{1,i-1}B_{ik})=p_{ik}^{(1)}$. 若 $u<i$，则 C_{uk} 发生意味着发生于 $(1,i)$ 的错误也和发生于 $(1,u)$ 的错误一起被修改，从而 C_{ik} 不会发生. 于是，当 $2\leqslant i\leqslant k$ 时，C_{ik} 发生的概率为 $P(C_{ik})=[1-\sum_{u=1}^{i-1}P(C_{uk})]p_{ik}^{(1)}$. 而事件 $C_{1k},C_{2k},\cdots,C_{nk}$ 都不发生的概率为 $1-\sum_{u=1}^{k}P(C_{uk})$. 于是可得矩阵 $P^{(2)}=(p_{ik}^{(2)})_{n\times n}$，这里

$$p_{ik}^{(2)}=\begin{cases}P(C_{ik})=p_{ik}^{(1)} & \text{若 } i=1\\ P(C_{ik})=[1-\sum_{u=1}^{i-1}p_{uk}^{(2)}]p_{ik}^{(1)} & \text{若 } 2\leqslant i\leqslant k\\ 0 & \text{若 } k<i\end{cases}\tag{7}$$

用矩阵 $P^{(3)}=(p_{ij}^{(3)})_{n\times m}$ 描述在 $(2,j)_2$ 完成时工艺设计人员发现发生于 $(1,i)$ 的错误的概率，由假设(4)可得

$p_{ij}^{(3)}=p_{ik}^{(2)}/m_k$，当子活动 $(2,j)_2$ 属于阶段 $(2,k)_1$. (8)

现在给出计算活动对平均持续时间的方法.

步骤 1 令 $\overline{b}_{11}$ 表示子活动 $(1,1)$ 的开始时间均值，它也是活动对开始时间的均值. 令 t 表示当前时刻，$\overline{g}_1(t)$ 表示活动 1 在当前时刻已完成的正常设计工作量的均值. 首先，令 $t=\overline{b}_{11}+\overline{d}_{11}+\overline{d}_{21}$ 及 $\overline{g}_1(t)=\min\{\overline{d}_1,\overline{d}_{11}+\overline{d}_{21}\}$. 令 $\overline{g}_1[(2,j)_2]$ 表示当活动 2 的子活动 $(2,j)_2$ 结束时活动 1 已完成的正常设计工作量的均值，首先令 $j=1$ 及 $\overline{g}_1[(2,j)_2]=\min\{\overline{d}_1,\overline{d}_{11}+\overline{d}_{21}\}$.

步骤 2 设活动 2 的子活动 $(2,j)_2$ 属于阶段 $(2,k)_1$，并设发生于活动 1 的子活动 $(1,i)$ 的错误在 $(2,j)_2$ 被发现，修改设计微循环发生(此微循环记作 $\langle i,j\rangle$). 当 $(2,j)_2$ 结束时，子活动 $(1,k)$ 之后的已完成的产品正常设计工作量的均值为 $\overline{z}_{1j}=\overline{g}_1[(2,j)_2]-\sum_{u=1}^{k}\overline{d}_{1u}$，其修改时间均值 $\overline{z}'_{1j}$ 可由式(4)求得(以 $\overline{z}_{1j}$ 代替 $\overline{d}_{1u}$). 则在微循环 $\langle i,j\rangle$ 中，活动 1 的修改时间的均值为 $\overline{w}'_{1j}=\overline{d}'_{1i}+\overline{d}_{1,i+1}+\cdots+\overline{d}_{1k}+\overline{z}'_{1j}$. 活动 2 的修改时间的均值为 $\overline{w}'_{2j}=\overline{d}'_{2l}+\overline{d}_{2,l+1}+\cdots+\overline{d}_{2i}$，这里当时 $i=1$ 时，$l=1$；当 $i>1$ 时，$l=\sum_{h=1}^{i-1}m_h+1$. 令 $\overline{b}_{1u}$ 和 $\overline{f}_{1u}$ 分别表示子活动 $(1,u)$ 的修改工作的开始时间与结束时间均值，$u=i,\cdots,k$. 令 $\overline{b}'_{2v}$ 和 $\overline{f}'_{2v}$ 分别表示子活动 $(2,v)_2$ 的修改工作的开始时间与结束时间均值，$v=l,\cdots,j$. 则这些子活动的修改工作的开始与结束时间的均值可依如下方法计算：$\overline{b}'_{1i}=t$；$\overline{f}_{1u}=\overline{b}_{1u}+\overline{d}_{1u}$，$u=i,i+1,\cdots,k$；$\overline{b}'_{1,u-1}=\overline{f}_{1u}$，$u=i,i+1,\cdots,k-1$；$\overline{b}_{2v}=\overline{f}_{1i}$；$\overline{f}_{2v}=\overline{b}_{2v}+\overline{z}_{1j}$；若 $v=1$，则 $\overline{b}'_{2v}=\overline{f}'_{11}$，若 $v=\sum_{h=1}^{u}m_h+1$，$u=i-1,i,\cdots,k-1$ 且 $i>1$，则 $\overline{b}_{2v}=\max\{\overline{f}_{1,u-1},\overline{f}_{2,v-1}\}$，否则 $\overline{b}_{2v}=\overline{f}_{2,v-1}$；$\overline{f}_{2v}=\overline{b}_{2v}+\overline{d}_{2v}$，$v=l,l+1,\cdots,j$. 在子活动 $(2,j)_2$ 和 $(2,j+1)_2$ 之间(或当 $j=m$ 时在 $(2,m)_2$ 之后)，产品与工艺设计修改工作持续时间的均值分别为 $\overline{\delta}'_{1j}=\sum_{i=1}^{k}p_{ij}^{(3)}(\overline{f}'_{1i}-\overline{b}'_{1i})$ 和 $\overline{\delta}'_{2j}=\sum_{i=1}^{k}p_{ij}^{(3)}(\overline{f}'_{2i}-\overline{b}'_{1i})$；产品和工艺设计修改工作结束时间的均值分别为 $\overline{\xi}_{1j}=\overline{b}_{1i}+\overline{\delta}'_{1j}$ 和 $\overline{\xi}_{2j}=\overline{b}'_{1i}+\overline{\delta}'_{2j}$；产品和工艺设计实际修改时间均值分别为 $\overline{w}'_{1j}=\sum_{i=1}^{k}p_{ij}^{(3)}\overline{w}'_{1i}$ 和 $\overline{w}'_{2j}=\sum_{i=1}^{k}p_{ij}^{(3)}\overline{w}'_{2i}$. 若 $j<m$，转步骤 3；若 $j=m$，转步骤 4.

步骤 3 若 $j<m$，令 $t=\overline{\xi}_{2j}$. 令

$$\overline{g}_1(t)=\begin{cases}\overline{g}_1[(2,j)_2] & \text{当 } \overline{\xi}_{1j}>\overline{\xi}_{2j}\\ \min\{\overline{d}_1,\overline{g}_1[(2,j)_2]+(\overline{\xi}_{2j}-\overline{\xi}_{1j})\} & \text{当 } \overline{\xi}_{1j}\leqslant\overline{\xi}_{2j}\end{cases}\tag{9}$$

若 $j+1 \neq \sum_{h=1}^{k} m_h + 1$ 或者若 $j+1 = \sum_{h=1}^{k} m_h + 1$ 且 $\bar{g}_1(t) \geqslant \sum_{a=1}^{k-1} \bar{d}_{1a}$，则活动 2 的子活动 $(2,j+1)_2$ 可在 $(2,j)_2$ 的修改完成后立即开始. 当 $(2,j+1)_2$ 结束时，活动 1 已完成的正常设计工作量的均值为

$$\bar{g}_1[(2,j+1)_2] = \begin{cases} \min\{\bar{d}_1, \bar{g}_1(t) + \max[0, \bar{d}_{2,j+1} - (\bar{\xi}_{1j} - \bar{\xi}_{2j})]\} & \text{当 } \bar{\xi}_{1j} > \bar{\xi}_{2j} \\ \min\{\bar{d}_1, \bar{g}_1(t) + \bar{d}_{2,j+1}\} & \text{当 } \bar{\xi}_{1j} \leqslant \bar{\xi}_{2j} \end{cases} \tag{10}$$

个产品开发项目的最早完成时间，并且平均资源利用率不低于一给定值. 首先给出资源利用率的定义.

若 $j+1 = \sum_{h=1}^{k} m_h + 1$ 且 $\bar{g}_1(t) < \sum_{a=1}^{k+1} \bar{d}_{1a}$，则 $(2,j+1)_2$ 必须在子活动 $(1,k+1)$ 完成后才能开始. 因此可令 $t \leftarrow t + \left[\sum_{a=1}^{k-1} \bar{d}_{1a} - \bar{g}_1(t)\right]$ 及 $\bar{g}_1(t) = \sum_{a=1}^{k+1} \bar{d}_{1a}$. 当 $(2,j+1)_2$ 结束时，活动 1 已完成的正常设计工作量均值为

$$\bar{g}_1[(2,j+1)_2] = \min\{\bar{d}_1, \bar{g}_1(t) + \bar{d}_{2,j+1}\} \tag{11}$$

当 $(2,j+1)_2$ 结束时，令 $t \leftarrow t + \bar{d}_{2,j+1}$，令 $j \leftarrow j+1$ 并转步骤 2.

步骤 4　当 $j = m$ 时，令 $t = \bar{\xi}_{2m}$ 并结束计算. 于是 $\bar{\xi}_{2m}$ 即为活动对的结束时间. 令 $\bar{s}$ 表示活动对的平均持续时间，于是 $\bar{s} = \bar{\xi}_{2m} - \bar{b}_{11}$. 产品设计活动 1 和工艺设计活动 2 的实际平均工作时间（包括正常设计时间和修改设计时间）分别为 $\bar{\zeta}_1 = \sum_{a=1}^{n} \bar{d}_{1a} - \sum_{j=1}^{m} \bar{w}_{1j}$ 和 $\bar{\zeta}_2 = \sum_{v=1}^{m} \bar{d}_{2v} + \sum_{j=1}^{m} \bar{w}_{2j}$

正在进行的活动需占用一定的资源，当活动停止时，即释放所占用的资源. 我们可给出如下定义：

定义 3　产品设计活动（或工艺设计活动）占用资源的时间均值（亦即实际平均工作时间 $\bar{\zeta}_1$ 或 $\bar{\zeta}_2$）与活动对平均持续时间（亦即 $\bar{s}$）之比称为产品设计活动（或工艺设计活动）在整个活动对时间范围内的资源占用率. 于是产品设计活动 1 的资源占用率为 $\rho_1 = \bar{\zeta}_1/\bar{s}$，工艺设计活动 2 的资源占用率为 $\rho_2 = \bar{\zeta}_2/\bar{s}$.

2　资源分配与产品开发计划制订集成优化问题的数学模型及算法

2.1　资源分配与产品开发计划制订集成优化问题的描述

本文研究的资源分配与产品开发计划制订集成优化问题的目标是：在有限资源约束下，求整个产品开发项目的最早完成时间，并且平均资源利用率不低于一给定值. 首先给出资源利用率的定义.

定义 4　分配给产品开发项目的第 k 类资源的利用率 $\eta_k(r)$ 及所有资源平均利用率 $\bar{\eta}(r)$ 可分别定义为

$$\eta_k(r) = \sum_{i \in P} (r_{i1}^k \rho_{i1} + r_{i2}^k \rho_{i2}) \bar{s}_i / r_k f(r), \quad k = 1, \cdots, \lambda$$

$$\bar{\eta}(r) = \sum_{k=1}^{\lambda} \sum_{i \in P} (r_{i1}^k \rho_{i1} + r_{i2}^k \rho_{i2}) \bar{s}_i / \sum_{k=1}^{\lambda} r_k f(r).$$

这里，λ 表示资源类型数；r_k 表示分配给此产品开发任务的第 k 类资源的数量；$f(r) = f(r_1, r_2, \cdots, r_\lambda)$ 表示项目最早完成时间，它由 $r = (r_1, r_2, \cdots, r_\lambda)$ 决定；P 表示产品 - 工艺设计活动对的集合；r_{i1}^k 和 r_{i2}^k 分别表示活动对 i 中的产品设计活动和工艺设计活动对第 k 类资源的最大需求量，这里“最大”意味着若分配的资源数大于 r_{i1}^k 或 r_{i2}^k，产品开发的进度不会加快，反之则产品开发进度会变慢；ρ_{i1} 和 ρ_{i2} 分别表示活动对 i 中产品设计活动和工艺设计活动的资源占用率；$\bar{s}_i$ 表示活动对 i 的平均持续时间（或工作量）.

并行工程产品开发过程的资源分配与计划制订的集成优化问题可描述如下：

$$P1: \quad \min \quad f(r) \tag{12}$$

$$\text{s.t.} \quad f(r) \leqslant \Psi \tag{13}$$

$$\bar{\eta}(r) \geqslant \bar{\eta} \tag{14}$$

这里，$r = (r_1, r_2, \cdots, r_\lambda)$，$f(r)$ 与 $\bar{\eta}(r)$ 的意义与定义 4 中相同，$r_1, r_2, \cdots, r_\lambda$ 均为非负整数；Ψ 表示规定的交付期（due date）；$\bar{\eta}$ 表示给定的平均资源利用率的最低值.

下面给出一种基于分枝定界的算法来解决问题 $P1$，此算法要在由整数点构成的搜索域中寻找一最优点. 对于搜索域中某些整数点，有必要计算由它决定的项目最早完成时间. 因此，我们首先研究在资源数量给定的情况下，如何求项目最早完成时间这一子问题.

2.2 在资源数量给定的情况下求产品开发项目的最早完成时间的算法

在资源数量给定的情况下求产品开发项目的最早完成时间的子问题可描述如下

$$\text{P2:}\quad \min\quad y=\max_{i\in P}\{f_i\} \tag{15}$$

$$\text{s.t.}\quad \int_{b_i}^{f_i}x_i(t)\mathrm{d}t=\bar{s}_i \tag{16}$$

$$f_i\leqslant b_j,(i\,|\,j)\in H \tag{17}$$

$$\sum_{i\in P}(r_{i1}^k\rho_{i1}+r_{i2}^k\rho_{i2})x_i(t)\leqslant r_k,\quad k=1,2,\cdots,\lambda \tag{18}$$

$$0\leqslant x_i(t)\leqslant 1$$

$$t\geqslant 0,i,j\in P \tag{19}$$

这里，P，r_{i1}^k，r_{i2}^k，ρ_{i1}，ρ_{i2}，$\bar{s}_i$，r_k 和 λ 与定义 4 中意义相同；H 表示活动对之间的前后顺序关系的集合，$(i\,|\,j)$ 表示 i 是 j 的紧前活动对；b_i 和 f_i 分别表示活动对 i 的开始与结束时间；t 表示当前时刻；$x_i(t)$ 表示在时刻 t 活动对 i 所要求的资源数量的满足程度.

解决问题 P2 的算法是一个数学规划与启发式规则相结合的算法. 在引入算法之前先给出一些定义.

定义 5 在当前时刻 t，所有紧前活动对均已完成但其本身尚未完成的活动对 i 的集合记作 A_t；其紧前活动对尚未全部完成的活动对的集合记作 N_t；所有已完成的活动对的集合记作 M_t.

定义 6 在当前时刻 t，活动对 i 的已完成工作量记作 $\omega_i(t)$，并且 $\omega_i(t)=\int_{b_i}^{t}x_i(\tau)\mathrm{d}\tau$.

定义 7 活动对 i 的初始时差(initial slack)记作 $\mu_i(b_i)$，并且 $\mu_i(b_i)=LS_i-b_i$. 这里，LS_i 为不考虑资源约束时，按照与通常的活动网络理论相同的方法计算出的产品 - 工艺设计活动对网络中活动对 i 的最晚开始时间. 活动对 i 在时刻 t 的时差(slack)记作 $\mu_i(t)$，并且 $\mu_i(t)=\mu_i(b_i)-\int_{b_i}^{t}[1-x_i(\tau)]\mathrm{d}\tau$.

解子问题 P2 的过程可被分解为若干阶段. 在每一阶段，需要确定正在进行的活动对的资源满足程度. 为此引入一条启发式规则，即集合 A_t 内的活动对中具有较小时差者在分配资源时有优先权. 当资源满足程度确定后，找出 A_t 中最先完成的活动对，然后进入下一阶段并更新 A_t. 集合 A_t 内正在进行的活动对的资源满足程度的确定问题可用下述线性规划模型 P3 来描述.

$$\text{P3:}\quad \min\quad \sum_{i\in P}[1-x_i(t)]a_i(t)/\mu_i(t) \tag{20}$$

$$\text{s.t.}\quad \sum_{i\in P}(r_{i1}^k\rho_{i1}+r_{i2}^k\rho_{i2})a_i(t)x_i(t)\leqslant r_k,\quad k=1,2,\cdots,\lambda \tag{21}$$

$$0\leqslant x_i(t)\leqslant 1,i\in P \tag{22}$$

这里，P，r_{i1}^k，r_{i2}^k，ρ_{i1}，ρ_{i2}，r_k，λ 和 $x_i(t)$ 与在问题 P2 中的含义相同；$\mu_i(t)$ 表示活动对 i 在时刻 t 的时差；若 $i\in A_t$，则 $a_i(t)=0$，若 $i\notin A_t$，则 $a_i(t)\neq 0$. 由目标函数(20)可见，为了使目标函数最小，时差 $\mu_i(t)$ 越小，相应的资源满足程度 $x_i(t)$ 就应越大. 这正体现了时差小的活动对在分配资源时优先的原则.

解决问题 P2 的算法可描述如下：

算法 1.

步骤 1 令 ε 为一小正数. 令叠代次数 $k=1$，把要求的交付期作为第一次叠代的交付期，即 $\Psi_1=\Psi$. 在产品 - 工艺设计活动对网络的基础上，利用与活动网络理论相同的方法，计算不考虑资源约束时产品开发项目的最早完成时间 y_0. 令 $\Psi_{\max}^*$ 表示产品开发项目不能在它之前完成的交付期的当前最大值，并首先令 $\Psi^*\max=y_0$. 令 $y_{\min}^*$ 表示产品开发项目最早完成时间的当前最小值，并首先令 $y_{\min}^*=\Psi$.

步骤 2 令第 k 次叠代要求的交付期为 Ψ_k. 若 $y_0>\Psi_k$，项目不能在交付期 Ψ_k 之前完成，转步骤 7. 若 $y_0\leqslant\Psi_k$，利用与活动网络理论相同的方法，计算第 k 次叠代中所有活动对 $i\in P$ 的最早开始时间 ES_{ik}，最早结束时间 EF_{ik}，最晚开始时间 LS_{ik} 和最晚结束时间 LF_{ik}.

步骤 3 令当前时间 $t=0$. 把第一个产品 - 工艺设计活动对放入集合 A_t，即 $A_t=\{1\}$，令 $N_t=P-\{1\}$ 及 $M_t=\varnothing$，并令 $b_1=ES_{1k}=t=0$，$\omega_1(t)=0$ 及 $\mu_1(t)=LS_{1k}-ES_{1k}$.

步骤 4 解问题 P3 获得当前时刻 t 的资源满足程度 $x_i(t)$，$i\in P$.

步骤 5 求集合 A_t 中最早完成的活动对. 首先求出 $\Delta\tilde{t}=\min\{\Delta\tilde{t}\ |\Delta\tilde{t}=[\bar{s}_i-\omega_i(t)]/x_i(t),i\in A_t\}$，令 $\tilde{t}=t+\Delta\tilde{t}$，然后求 $I_{\tilde{t}}=\{i\,|\Delta\tilde{t}_i=\Delta\tilde{t},i\in A_t\}$. $\forall\, i\in I_{\tilde{t}}$，令 $f_i=\tilde{t}$ 及 $\omega_i(\tilde{t})=\bar{s}_i$. 更新已

完成活动对的集合为 $M_{\tilde{t}}=M_t\cup I_{\tilde{t}}$. 令 $N_{\tilde{t}}=\{i|i\in N_t,P_i\subseteq M_{\tilde{t}}\}$, 这里 P_i 为活动对 i 的紧前活动对集合. $\forall i\in N_{\tilde{t}}$, 令 $b_i=\tilde{t}$, $\omega_i(\tilde{t})=0$, $\mu_i(\tilde{t})=\mu_i(b_i)=LS_{ik}-b_i=LS_{ik}-\tilde{t}$. $\forall i\in A_t-I_t$, 更新活动对已完成工作量 $\omega_i(\tilde{t})$, $\omega_i(t)+x_i(t)\Delta\tilde{t}$, 更新活动对时差为 $\mu_i(\tilde{t})=\mu_i(t)-[1-x_i(t)]\Delta\tilde{t}$, 然后更新未开始活动对集合为 $N_{\tilde{t}}=N_t-N_{\tilde{t}}$ 及正在进行的活动对的集合为 $A_{\tilde{t}}=(A_t-I_{\tilde{t}})\cup N_{\tilde{t}}$.

步骤6　若 $\exists i\in A_{\tilde{t}}$, $\mu_i(\tilde{t})<0$, 则产品开发项目不能在交付期 Ψ_k 之前完成, 转步骤7. 否则, 若 $M_{\tilde{t}}=P$, 则产品开发项目能在交付期 Ψ_k 之前完成, 转步骤8. 若 $M_{\tilde{t}}\neq P$, 令 $t=\tilde{t}$ 并转步骤4.

步骤7　若 $k=1$, 转步骤9, 因为可以得出结论:在有资源约束时, 产品开发项目不能在要求的交付期 Ψ 之前完成. 若 $k>1$, 则令 $k\leftarrow k+1$, 然后令 $\Psi_k=(\Psi_{k-1}+y^*_{\min})/2$ 并转步骤2. 特别地, 若 $k>1$ 且 $\Psi_k>\Psi^*_{\max}$, 还需令 $\Psi^*_{\max}=\Psi_k$.

步骤8　令 $y_k=\tilde{t}$, 这里 y_k 表示有资源约束时, 第 k 次叠代中项目最早完成时间. 若 $|y_k-y^*_{\min}|<\varepsilon$, 转步骤9. 若 $|y_k-y^*_{\min}|\geqslant\varepsilon$, 令 $k\leftarrow k-1$ 然后令 $\Psi_k=(y_{k-1}+\Psi^*_{\max})/2$ 并转步骤2. 特别地, 若 $k>1$ 且 $y_k<y^*_{\min}$, 还需令 $y^*_{\min}=y_k$.

步骤9　若 $k=1$, 产品开发项目不能在要求的交付期 Ψ 之前完成. 若 $k>1$, 产品开发项目能够在要求的交付期 Ψ 之前完成并且最后一次叠代中项目最早完成时间即为P2的解, 即 $y=y_k$.

因此, 在问题P1中, 对于给定的 $r_1,r_2,\cdots,r_\lambda$, 可根据算法1求得 $y=f(r)=f(r_1,r_2,\cdots,r_\lambda)$. 同时, 每一个活动对的开始与结束时间也可得到. 由定义4可求得各类资源的利用率和所有资源的平均利用率.

2.3　解资源分配与计划制订集成优化问题的算法

接下来将引入解决问题P1的算法, 这是一种分枝定界算法. 在此之前须给出几个定义.

定义8　令 $r^l_k=\mathrm{ceil}\left[\sum_{i\in P}(r^l_{i1}\rho_{i1}+r^l_{i2}\rho_{i2})\bar{s}_i/\Psi\right]$, $k=1,2,\cdots,\lambda$, 这里函数 ceil(·) 意味着求不小于括号内数的最小整数. 可见, 若 $\exists k, r_k<r^l_k$, 则产品开发项目不能在交付期 Ψ 之前完成. 如果不考虑资源约束, 则可计算出相应的产品开发项目的最早完成时间 f^*. 令 $r^*_k=\max_{t\in[0,f^*]}\{\sum_{i\in P}(r^*_{i1}+r^*_{i2})\theta_i(t)\}$, $k=1,2,\cdots,\lambda$, 这里, 若时刻 t 活动对 i 在进行, 则 $\theta_i(t)=1$, 否则 $\theta_i(t)=0$. 于是 $\forall k, k=1,2,\cdots,\lambda, r^*_k$ 是最大资源需求量. 因此, 问题P1的最优解必然在下面的区域内: $U=[r^l_1,r^*_1]\times[r^l_2,r^*_2]\times\cdots\times[r^l_\lambda,r^*_\lambda]$, 它是分枝定界算法的搜索域, $r=(r_1,r_2,\cdots,r_\lambda)$ 是搜索域内的一个点, 称为节点. 特别地, 若 $\exists r_k$, 使得 $r_k=r^l_k$ 或 $r_k=r^*_k$ 成立, 则 $r=(r_1,r_2,\cdots,r_\lambda)$ 可称为边界点.

定义9　若节点 r 不满足问题P1的约束条件(13), 则称之为不可行节点. 若节点 r 满足约束条件(13)但不满足约束条件(14), 则称之为弱可行节点. 若节点 r 同时满足约束条件(13)和(14), 则称之为强可行节点.

定义10　设 $r=(r_1,r_2,\cdots,r_\lambda)$ 是搜索域 U 内的一个节点, 则 $SN(r)=\{r_k|r_k=(r_1,\cdots,r_k-1,\cdots,r_\lambda),k=1,2,\cdots,\lambda\}$ 可被称为节点 r 的子节点集. $SN(r)$ 中的每个元素称为节点 r 的子节点. 节点 r 称为集合 $SN(r)$ 中所有元素的父节点.

定义11　若 $\forall k, r'_k\leqslant r_k$ 成立, 则称 $r'\leqslant r$. 这里 $k=1,2,\cdots,\lambda$; $r'=(r'_1,r'_2,\cdots,r'_\lambda)$; $r=(r_1,r_2,\cdots,r_\lambda)$. 并设 $LN(r)=\{r'|r'\leqslant r\}$.

定理1　设 $r'\leqslant r$ 成立. 若 r 是一不可行节点, 则 r' 也是一不可行节点. 若 r 和 r' 均满足约束条件(13), 且 $f(r)$ 和 $f(r')$ 分别是由 r 和 r' 决定的产品开发项目的最小完成时间, 则 $f(r')\geqslant f(r)$ 成立.

证明　显然, 向量 r' 中的每个元素都不大于向量 r 中相应的元素. 资源越少, 产品开发的进度越慢. 所以, 定理1的结论成立. 特别地, 若 $r'\in SN(r)$, 则定理1的结论成立.　　证毕.

定义12　无须继续搜索其子节点的节点称为死节点.

定理2　设 r^* 为当前最优强可行节点, $f^*=f(r^*)$ 为由 r^* 决定的产品开发项目的最小完成时间. 若某节点 $r\in U$ 满足下面三个条件中之任意一个, 则无须搜索其子节点, 因而它是一死节点. 条件(1): r 是一不可行节点; 条件(2): $f(r)\geqslant f^*$; 条件(3): $f(r)<f^*$ 成立并且 r 是一个强可行节点.

证明(1) 若 r 是一不可行节点，则由定理 1 可知，r 的所有子节点均为不可行节点，因而无须继续搜索这些子节点，即 r 为一死节点. (2) 若 $f(r) \geqslant f^*$ 成立，则根据定理 1，由 r 的子节点决定的产品开发项目的最小完成时间不会小于 $f(r)$，显然更不会小于 f^*，即子节点不会比 r^* 更优. 因此无须继续搜索其子节点，即 r 为一死节点. (3) 若 $f(r) < f^*$ 成立且 r 是一个强可行节点，则 r 成为当前最优强可行节点. 根据定理 1，即使 $SN(r)$ 或 $LN(r)$ 中有强可行节点，但由 r 的子节点决定的产品开发项目的最小完成时间不会小于 $f(r)$，也就是说子节点不会比 $f(r)$ 更优. 因此无须继续搜索其子节点，即 r 为一死节点. 证毕.

显然，由定义 12 及定理 1 和 2 可导出下面的推论.

推论 1 若 r 是死节点，则集合 $LN(r)$ 内的节点全都是死节点；若集合 $SN(r)$ 内的节点全都是死节点，则 r 是死节点.

定义 13 分枝定界算法最先搜索的节点称为初始节点. 从初始节点开始，算法根据一定的规则在搜索域中搜索到一系列的节点，这些节点按被搜索到的先后次序形成了搜索路径. 在节点 r 的所有父节点中，在搜索路径中距 r 最近者称为 r 的最近父节点.

由于通常的分枝定界算法的搜索效率不太高，所以本文引入一个启发式规则来提高其搜索效率. 令 V_1 表示相应的项目完成时间和平均资源利用率已计算过的节点的集合. 设 $\hat{r} \in V_1$ 并令 $f(\hat{r})$ 表示其相应的项目完成时间. 令 $V_r = \{\hat{r} \mid \hat{r} \in V_1 \text{ 且 } r \leqslant \hat{r}\}$ 及 $\hat{f}_r = \max\{f(\hat{r}), \hat{r} \in V_r\}$. 不失一般性，假设项目开始时间为零. 则相应于 r 的平均资源利用率的上界可定义如下.

定义 14 相应于 r 的第 k 类资源利用率的上界及平均资源利用率的上界分别为

$$\eta_k(r) = \sum_{i \in P} (r_{i1}^k \rho_{i1} + r_{i2}^k \rho_{i2}) \bar{s}_i / r_k \hat{f}_r,$$

$$k = 1, \cdots, \lambda \text{ 和}$$

$$\eta(r) = \sum_{k=1}^{\lambda} \sum_{i \in P} (r_{i1}^k \rho_{i1} + r_{i2}^k \rho_{i2}) \bar{s}_i / \sum_{k=1}^{\lambda} r_k \hat{f}_r$$

这里 $P, r_{i1}^k, r_{i2}^k, \rho_{i1}, \rho_{i2}, \bar{s}_i, r_k$ 和 λ 与定义 4 中意义相同，并且 $r = \{r_1, r_2, \cdots, r_\lambda\}$. 令 V_2 表示相应的各类资源利用率的上界和平均资源利用率的上界已经计算过的节点的集合.

定理 3 若 $\hat{\eta}(r) < \tilde{\eta}$，则 $\bar{\eta}(r) < \tilde{\eta}$ 成立，这里 $\bar{\eta}(r)$ 是由节点 r 决定的平均资源利用率.

证明 若 $r \leqslant r$，则由定理 1，$f(r) \geqslant f(r)$ 成立. 由于 $\hat{f}_r = \max\{f(\hat{r}), \hat{r} \in V_r\}$，则 $f(r) \geqslant \hat{f}_r$ 成立. 根据定义 4 和 14，$\bar{\eta}(r) \leqslant \hat{\eta}(r)$ 成立，于是 $\bar{\eta}(r) < \tilde{\eta}$. 证毕.

根据定理 3，可引入如下启发式规则.

规则 1 对于一节点 $r \in U$，若 $\hat{\mu}(r) < \tilde{\eta}$，则不必计算 $f(r)$，而可直接搜索其子节点.

根据规则 1，计算项目完成时间的次数会大大减少. 解决问题 P1 的分枝定界算法如下：

算法 2

步骤 1 根据定义 8 决定搜索域 U，令 $r_0 = (r_1^1, r_2^1, \cdots, r_\lambda^1)$ 表示初始节点. 以 DN 表示已知死节点集并首先令 $DN = \varnothing$. 以 $r_{cn} = (r_{(cn,1)1}, r_{(cn,2)1}, \cdots, r_{(cn,\lambda)1})$ 表示当前节点并首先令 $r_{cn} = r_0$. 令 $f^* = M$，这里 M 为一充分大正数.

步骤 2 若 r_0 为一不可行节点，则此问题无可行解，令 $DN \leftarrow DN \cup \{r_0\} \cup LN(r_0)$ 并转步骤 6 以停止搜索. 若 r_0 是一强可行解，则它即为最优解，令 $r^* = r_0$ 及 $f^* = f(r_0)$，令 $DN \leftarrow DN \cup \{r_0\} \cup LN(r_0)$ 并转步骤 6. 若 r_0 是一弱可行解，则根据算法 1 计算 $f(r_0)$ 和 $\bar{\eta}(r_0)$，令 $V_1 = \{r_0\}$ 及 $V_2 = \varnothing$ 并转步骤 3.

步骤 3 令 $SN(r_{cn}) = \{r_{(sn,j)} \mid r_{(sn,j)} = (r_{(cn,1)}, \cdots, r_{(cn,j)} - 1, \cdots, r_{(cn,\lambda)}), j = 1, 2, \cdots, \lambda\}$ 表示 r_{cn} 的子节点的集合. 若 $r_{cn} \in V_1$，则选择一子节点 $r_{(sn,k)} \in SN(r_{cn}) - DN$，其对应的第 k 类资源利用率在各类资源利用率中最低；若 $r_{cn} \in V_2$，则选择一子节点 $r_{(sn,k)} \in SN(r_{cn}) - DN$，其对应的第 k 类资源利用率的上界在各类资源利用率的上界中最低. 令 $r_{cn} \leftarrow r_{(sn,k)}$.

步骤 4 若 $r_{cn} \in DN$，则 r_{cn} 是一已知死节点，转步骤 5. 若 $r_{cn} \notin DN$，根据推论 1 判断 r_{cn} 是否为一新的死节点. 若 r_{cn} 为一新的死节点，令 $DN \leftarrow DN \cup \{r_{cn}\} \cup LN(r_{cn})$ 并转步骤 5. 若根据推论 1 不能判断 r_{cn} 是否为一新的死节点，则判断是否 $r_{cn} \in V_1 \cup V_2$. 若 $r_{cn} \in V_1 \cup V_2$，转步骤 3. 若 $r_{cn} \notin V_1 \cup V_2$，根据定义 14 计算由 r_{cn} 决定的各类资源利用率的上界(即 $\hat{\eta}_i(r_{cn}), j = 1, 2, \cdots, \lambda$) 及平均资源利用率的上界 $\hat{\eta}(r_{cn})$. 根据规则 1，若

$\hat{\eta}(r_{cn})<\eta$,令 $V_2 \leftarrow V_2 \cup \{r_{cn}\}$ 并转步骤 3 继续搜索 r_{cn} 的子节点. 若 $\hat{\eta}(r_{cn}) \geqslant \tilde{\eta}$,根据算法 1 计算 $f(r_{cn})$ 和 $\bar{\eta}(r_{cn})$. 若 r_{cn} 是一不可行节点,则由定理 2 的条件 1 可知 r_{cn} 为一新的死节点,令 $DN \leftarrow DN \cup \{r_{cn}\} \cup LN(r_{cn})$ 并转步骤 5. 若 r_{cn} 至少是弱可行解,令 $V_1 \leftarrow V_1 \cup \{r_{cn}\}$. 在这种情况下,若 $f(r_{cn}) \geqslant f^*$,由定理 2 的条件 2 可知,r_{cn} 是一新的死节点,令 $DN \leftarrow DN \cup \{r_{cn}\} \cup LN(r_{cn})$ 并转步骤 5. 若 $f(r_{cn})<f^*$ 且 r_{cn} 为一强可行解,则由定理 2 的条件 3 可知,r_{cn} 是一新的当前最优解,也是一新的死节点,于是令 $r^* = r_{cn}$,$f^* = f(r_{cn})$,及 $DN \leftarrow DN \cup \{r_{cn}\} \cup LN(r_{cn})$ 并转步骤 5. 若 $f(r_{cn})<f^*$ 且 r_{cn} 为一弱可行解,则由定理 2 可知,r_{cn} 不是死节点,于是转步骤 3 继续搜索 r_{cn} 的子节点.

步骤 5　令 r_{fn} 表示当前节点 r_{cn} 在搜索路径中的最近父节点. 若 r_{cn} 为一死节点且 $r_{cn} = r_0$,转步骤 6. 若 r_{cn} 为一死节点且 $r_{cn} \neq r_0$ 或若 r_{cn} 为一边界点且 $r_{cn} \neq r_0$,令 $r_{cn} \leftarrow r_{fn}$ 并转步骤 4. 这意味着若当前节点为死节点或边界点,则须回溯至其最近父节点. 特别地,若当前死节点为初始节点,则转步骤 6,结束搜索.

步骤 6　若 $r_{cn} = r_0$ 且 $r_0 \in DN$,则结束搜索. 若 $f^* = M$,则问题 P1 无强可行解. 若 $f^* < M$,则当前的 r^* 为最优解,f^* 为产品开发项目的最早完成时间,平均资源利用率高于 $\tilde{\eta}$.

由算法 1 可知,每个产品 - 工艺设计活动对的开始和结束时间是和整个项目的最早完成时间同时获得的. 因此,一旦问题 $P1$ 的最优解和产品开发项目的最小完成时间被求得,则每个活动对相应的开始与结束时间也就同时求得,产品开发计划也就制订出来了.

3　算例

本节将给出一算例来说明活动对平均持续时间,产品和工艺设计活动资源占用率的计算方法以及解资源分配和产品计划制订集成优化问题的算法.

3.1　计算活动对平均持续时间及产品和工艺设计活动的资源占用率

例 1. 图 2 表示并行工程环境下某种汽车传动系统的开发过程. 这是一个由 9 个产品 - 工艺设计活动对(1-9)和一个虚活动(10)构成的网络. 以活动对 1 为例. 假定产品设计活动 1 可分解为四个子活动,工艺设计活动 2 可分解为四个阶段且每个阶段又可分解为三个子活动. 因此活动 2 可分解为 12 个子活动(见图 3). 活动 1,活动 2 的参数列于表 1,各子活动的参数列于表 2.

表 1　活动对 1 的产品设计活动 1 和工艺设计活动 2 的参数

活动号	a	c	b	p	p'	γ
活动 1	20.0	22.0	24.0	5.0	10.0	0.4
活动 2	18.0	20.0	23.0	6.0	11.0	0.5

a,c 和 b 分别表示正常设计的最乐观、最可能和最悲观完成时间;p 和 p' 分别表示正常设计完成时间和修改设计完成时间所服从的 β 分布密度函数的参数;γ 表示最可能修改时间与最悲观修改时间之间的系数.

表 2　子活动参数

活动号	子活动号	活动 1 中的紧前子活动	活动 2 中的紧前子活动	子活动工作量在活动工作量中所占比例
活动 1	(1,1)			0.25
活动 1	(1,2)	(1,1)		0.30
活动 1	(1,3)	(1,2)		0.25
活动 1	(1,4)	(1,3)		0.20
活动 2	(2,1)	(1,1)		0.08
活动 2	(2,2)		(2,1)$_2$	0.06
活动 2	(2,3)		(2,2)$_2$	0.08

续表 2

活动号	子活动号	活动 1 中的紧前子活动	活动 2 中的紧前子活动	子活动工作量在活动工作量中所占比例
活动 2	(2,4)	(1,2)	$(2,3)_2$	0.09
活动 2	(2,5)		$(2,4)_2$	0.09
活动 2	(2,6)		$(2,5)_2$	0.11
活动 2	(2,7)	(1,3)	$(2,6)_2$	0.08
活动 2	(2,8)		$(2,7)_2$	0.11
活动 2	(2,9)		$(2,8)_2$	0.10
活动 2	(2,10)	(1,4)	$(2,9)_2$	0.06
活动 2	(2,11)		$(2,10)_2$	0.07
活动 2	(2,12)		$(2,11)_2$	0.07

假设活动 1 的子活动$(1,i)$,$i=1,2,3,4$ 的错误未在阶段$(2,s)_1$,$s=i,\cdots,k-1$ 被发现而在阶段$(2,k)_1$ 被发现的条件概率为 $p=0.8$,这里 $k=i,\cdots,4$. 则由式(6),(7) 和(8) 可得描述发生于活动 1 的子活动$(1,i)$ 的错误在活动 2 的子活动$(2,j)_2$ 被发现的概率的矩阵 $P^{(3)}$ 为

$$P^{(3)}=\begin{bmatrix}0.2667 & 0.2667 & 0.2667 & 0.0533 & 0.0533 & 0.0533 & 0.0107 & 0.0107 & 0.0107 & 0.0027 & 0.0027 & 0.0027\\ 0.0000 & 0.0000 & 0.0000 & 0.2240 & 0.2240 & 0.2240 & 0.0516 & 0.0516 & 0.0516 & 0.0132 & 0.0132 & 0.0132\\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.2168 & 0.2168 & 0.2168 & 0.0635 & 0.0635 & 0.0635\\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.2539 & 0.2539 & 0.2539\end{bmatrix}$$

根据 2.2 节中给出的方法(特别是步骤 1 — 4) 可得活动对 1 及其它各活动对平均持续时间,产品和工艺设计活动的实际平均工作时间及资源占用率的计算结果(见表 3).

表 3 9 个活动对的计算结果

活动对号	活动对平均持续时间	产品设计活动实际平均工作时间	产品设计活动资源占用率	工艺设计活动实际平均工作时间	工艺设计活动资源占用率
1	45.068 738	35.145 983	0.779 831	28.209 343	0.625 918
2	34.563 318	29.302 972	0.847 805	24.271 489	0.702 232
3	38.387 808	33.064 729	0.861 334	21.390 047	0.557 209
4	29.311 759	23.994 574	0.818 599	16.573 367	0.565 417
5	35.823 075	29.628 894	0.827 090	24.405 410	0.681 276
6	31.892 277	25.591 749	0.802 443	20.080 892	0.629 647
7	32.625 015	24.588 989	0.753 685	24.300 146	0.744 832
8	36.383 771	30.588 868	0.840 728	22.362 725	0.614 635
9	44.701 982	36.869 740	0.820 316	27.695 150	0.619 551

3.2 资源分配及产品开发计划的制订

例 1 中各产品设计活动和工艺设计活动的所需的资源类型和数量列于表 4 中.

表 4 产品设计活动和工艺设计活动的所需的资源类型和数量

活动对号	活动类型	资源 1	资源 2	资源 3	资源 4
1	产品设计活动	5.0			
1	工艺设计活动		3.0	2.0	2.0
2	产品设计活动	3.0			
2	工艺设计活动		1.0	1.0	1.0
3	产品设计活动	3.0			
3	工艺设计活动		1.0	1.0	1.0

活动对号	活动类型	资源 1	资源 2	资源 3	资源 4
4	产品设计活动	3.0			
4	工艺设计活动		1.0	1.0	1.0
5	产品设计活动	3.0			
5	工艺设计活动		1.0	1.0	1.0
6	产品设计活动	4.0			
6	工艺设计活动		2.0	1.0	1.0
7	产品设计活动	4.0			
7	工艺设计活动		2.0	1.0	1.0
8	产品设计活动	4.0			
8	工艺设计活动		2.0	1.0	1.0
9	产品设计活动	4.0			
9	工艺设计活动		2.0	1.0	1.0

这里，资源 1 为产品结构设计人员；资源 2 为工艺设计人员；资源 3 为可维护性设计人员；资源 4 为可靠性设计人员.

设此产品开发项目的交付期为 200.0，即 $T = 200.0$. 并设要求的平均资源利用率的最低值为 $\eta = 0.80$. 根据定义 8，可得搜索域为 $U = [5, 16] \times [2,8] \times [2,4] \times [2,4]$. 从初始点 $r_0 = (16,8,4,4)$ 开始，根据算法 2 搜索最优节点. 搜索域中共有 $12 \times 7 \times 3 \times 3 = 756$ 个整数点，计算了其中 23 个点对应的项目完成时间和平均资源利用率后，可得最优解如下，它是这 23 个点中的第 5 个点.

$r^* = (7,3,2,2), f^*$

$= f(r^*) = 163.072\ 943,$

$\eta_1(r^*) = 0.870\ 304,$

$\eta_2(r^*) = 0.736\ 166,$

$\eta_3(r^*) = 0.728\ 195, \eta_4(r^*) = 0.728\ 195.$

$\bar{\eta}(r^*) = 0.800\ 957$

由算法 1 可知，每个活动对的开始及结束时间都可同时求得，于是就获得了产品开发计划(见表 5).

若应用了规则 1，则在搜索域内的 756 个整数点中，仅有 23 个点，其相应的项目完成时间和平均资源利用率需要计算. 程序用 C 语言编写，在一台具有 32M 内存的 Pentium 133 计算机上运行了约 3 秒. 若不应用规则 1，则需要计算 624 个点相应的项目完成时间和平均资源利用率，其中第 23 个点为最优解，在同样的计算机上运行所用时间约 8 秒. 两种情况下求得的结果相同，但应用了规则 1 会使搜索效率大大提高.

表 5　汽车传动系统产品开发计划

活动对号	开始时间	结束时间
1	0.000 000	45.068 738
2	45.068 738	115.455 160
3	45.068 738	83.702 466
4	45.068 738	81.981 395
5	45.068 738	80.891 812
6	115.455 160	147.347 437
7	83.702 466	163.072 943
8	81.981 395	161.300 816
9	80.891 812	159.314 527

4　结论

现有的关于并行工程定量化建模的研究还不够充分，特别是在修改设计微循环的细节特征的描述方面. 本文提出了另一种并行工程的定量化模型，产品 - 工艺设计活动对网络模型，作为已有的定量化模型的必要补充. 这种定量化模型能够描述修改设计微循环的特征，适合于数学处理. 本文首先给出了活动对平均持续时间以及产品和工艺设计活动资源占用率的计算方法. 在此基础上，研究了资源分配与产品开发计划制订的集成优化问题. 这是一种有资源约束的项目调度问题. 与通常的这类问题不同的是，在本文的问题中，分配给产品开发项目的资源数量不是事先确定的，而是在寻找最优计划的过程中逐步确定的.

本文给出了一个新的基于分枝定界的算法来解决此问题，并引入一个启发式规则来提高算法的搜索效率. 算例表明，本文给出的算法及启发式规则是非常有效的.

参 考 文 献：

[1] Winner R I, Pennell J P, Bertrand H E, Slusarczuck M M. Role of concurrent engineering in weapon system acquisition[R]. Final Rept. (AD-A203615/0/HDM), 1988

[2] 严洪森. 并行工程的 Petri 网建模[J]. 东南大学学报, 1995, 25(5A): 159-164

[3] Yan H S, Jiang Z J. Agile concurrent engineering[J]. Integrated Manufacturing Systems, 1999, 10(2): 103-112

[4] Sprague R A, Singh K J, Wood R T. Concurrent engineering in product development[J]. IEEE Design & Test of Computers, 1991, 8(1): 6-13

[5] 吴祚宝，熊光楞，常天庆，彭 毅. 并行工程管理系统研究[J]. 高技术通讯, 1996, 6(5): 21-25

[6] Lu S C Y. Knowledge processing for concurrent engineering: an evolving challenge in CIM research[J]. Robotics and Computer-Integrated Manufacturing, 1990, 7(3): 263-277

[7] 张玉云，熊光楞，李伯虎. 并行工程方法、技术与实践[J]. 自动化学报, 1996, 22(6): 745-753

[8] 彭 毅，吴祚宝，张珂殊，熊光楞. 并行工程产品开发过程的建模方法学[J]. 系统仿真学报, 1996, 8(3): 14-18

[9] 李文波，吴冲锋，王意冈. Petri 网分析企业组织效率的探讨[J]. 管理科学学报, 1999, 2(2): 49-56

[10] Krishnan V, Eppinger S D, Whitney D E. A model-based framework to overlap product development activities[J]. Management Science, 1997, 43(4): 437-451

[11] Loch C H, Terwiesch C. Communication and uncertainty in concurrent engineering[J]. Management Science, 1998, 44(8): 1032-1048

[12] Terwiesch C, Loch C H. Measuring the effectiveness of overlapping development activities[J]. Management Science, 1999, 45(4): 155-165

[13] Ha A Y, Porteus E L. Optimal timing of reviews in concurrent design for manufacturability[J]. Management Science, 1995, 41(9): 1431-1447

[14] Smith R P, Eppinger S D. A predictive model of sequential iteration in engineering design[J]. Management Science, 1998, 43(8): 1104-1120

[15] Smith R P, Eppinger S D. Identifying controlling features of engineering design iteration[J]. Management Science, 1997, 13(3): 276-293

[16] Herroelen W, De Reyck B, Demeulemeester E. Resource-constrained scheduling: a survey of recent developments [J]. Computers and Operations Research, 1998, 25(4): 279-302

[17] Demeulemeester E, Herroelen W. A branch-and-bound procedure for the multiple resource-constrained project scheduling problem[J]. Management Science, 1992, 38(12): 1803-1818

[18] Icmeli O, Rom W O. Solving the resource constrained project scheduling problem with optimization subroutine library[J]. Computers and Operations Research, 1996, 23(8): 801-817

[19] Leachman R C, Dincerler A, Kim S Y. Resource-constrained scheduling of projects with variable-intensity activities [J]. IIE Transactions, 1990, 22(1): 31-40

[20] Cheng R, Gen M. An evolution programme for the resource-constrained project scheduling problem[J]. Int. J. Computer Integrated Manufacturing, 1998, 11(3): 274-287

[21] Belbe U, Kusiak A. Dynamic scheduling of design activities with resource constraints[J]. IEEE Transactions on Systems, Man and Cybernetics—Part A: Systems and Humans, 1997, 27(1): 105-111

[22] Belhe U D, Kusiak A. Scheduling design activities with a pull system approach[J]. IEEE Transactions on Robotics and Automation, 1996, 12(1): 15-21

Quantitative modeling and planning of the product development process in concurrent engineering

WANG Zheng, YAN Hong-sen, LIU Xia-ling, SONG Wen-zhong
Research Institute of Automation, Southeast University, Nanjing 210096, China

Abstract: Since in some proposed quantitative models the detail feature of the mini-circulation of design revision in product development process has not been described sufficiently, a new type of quantitative model based on the network of product-process design activity pairs is proposed in this paper to describe the product development process in concurrent engineering. And the method of computing the mean duration of the product-process design activity pair and the resource occupation rate of product or process design activity is proposed. Based on that, the problem of the planning of the product development process in concurrent engineering is modeled as a resource-constrained project scheduling problem. Different from other resource-constrained project scheduling problems, in this paper, the number of the resources allocated for the product development project is not pre-determined, but is obtained along with the optimal product development plan. Therefore, this is an integrated optimization problem of the resource allocation and the planning of the product development process. A new algorithm based on branch-and-bound approach is proposed in this paper for solving the problem and a heuristic rule is introduced to improve the searching efficiency of the algorithm.

Key words: concurrent engineering; process modeling; mini-circulation of design revision; network of product-process design activity pairs; product development plan; resource-constrained project scheduling problem; branch-and-bound algorithm