

文章编号:1001-9081(2006)09-2166-03

## 一种基于工作流图的时间资源分配策略

刘丹妮,陈秀寓

(东北大学 东软信息学院,辽宁 大连 116023)

(liudanni@neusoft.edu.cn)

**摘要:**研究了工作流中时间分配问题。在具体活动基础上增加标记活动,以记录所在路由分支的资源使用情况;将时间看作一种不可更新资源,并将时间资源分为私有资源、公有资源和双重资源三类;提出了一种路由结构的时间资源分析方法和分配算法,根据不同路由分支的特点及初始分配资源在实际执行过程中的使用情况,通过裂变与聚合进行时间资源再分配,保证整个工作流的顺利完成。通过实例说明工作流图中时间资源分配过程。

**关键词:**工作流;时间资源;松弛时间

**中图分类号:**TP302 **文献标识码:**A

## A strategy for time resource allocation based on workflow graph

LIU Dan-ni, CHEN Xiu-yu

(Neusoft Information Institute, Northeastern University, Dalian Liaoning 116023, China)

**Abstract:** Time allocation in workflow was studied. Firstly, labeled activities were added in order to make records of the resources usage on each route origin. Secondly, time was regarded as a kind of non-renewable resources, and was classified into three types: private, public and mixed. Thirdly, an approach for time resource analysis and allocation was proposed. In the beginning, time resource was pre-allocated according to the characteristics of different routing, then during the process of actual operation, time resource was reallocated according to time consuming situation by the method of splitting and converging. Finally, a case study illustrated the time resource allocation process in workflow graph.

**Key words:** workflow; TR(time resource); TSlack(time slack)

### 0 引言

近年来,在科学研究和实践中,工程的时序安排越来越多地受到学者关注。Brucker 对资源环境、活动特征和客观功能等进行分类,对资源约束的单一模式和多重模式应用启发式算法进行了活动执行序列的讨论<sup>[1]</sup>。Senkul 提出了模拟资源消耗和资源分配信息的语言,含有约束解决方案的进度表模型将得到的工作流描述转换成约束语言,生成满足资源分配约束的时刻表<sup>[2]</sup>。Hongchen 将时间工作流进行扩展,对工作流描述的资源一致性进行验证的同时考虑了时间约束条件<sup>[3]</sup>。依据业务实例在工作流四种路由结构的逗留时间分布,给出以资源单位时间消耗成本最小化为目标的求解最优资源数量的方法<sup>[4]</sup>。目前很多文献将时间作为约束条件分析资源模型中人员、设备、软件等资源的分配和调度情况<sup>[1-6]</sup>,但是将时间作为不可更新资源纳入资源范畴,与其他资源统一分配和回收,取消模型外部时间约束的分配策略几乎未见报道。本文将研究时间作为资源在四种路由结构中的分配情况,以及工作流图中活动、各种路由分支的时间分配策略。

工作流的过程模型由资源模型、组织模型、功能映射模型和约束规则组成。约束规则中时间约束是首要约束,如能将时间作为资源,并入资源模型,可减少约束规则描述的内容,简化工作流模型的建模过程。将活动需要的时间、人员、设备等资源分配给某活动,当时间资源耗尽时,其他资源的权限立即被回收。如果这些非时间资源闲置或其他需要此资源的活

动不紧急,则将这些资源和相应时间的权限一同对该活动开放。时间起到对其他资源权限的开放和关闭作用。

本文基于文献[7,8]的工作流模型,给出标记活动和具体活动的概念,通过对时间资源的特性分析,将时间资源分为私有资源、公有资源和双重资源三类。对四种结构中的活动和路由分支进行时间资源与松弛时间的界值确定。依据以上分析,给出时间资源的分配算法,详细描述了时间资源分配策略,以及松弛时间的裂变和聚合。最后通过实例演示了算法的分配过程。

### 1 工作流模型

在实际的工作流中,常有可跨越活动和可替换路径,对可跨越活动适当取舍和可替换路径适当选择,会提高工作流的执行效率。以文献[7,8]的工作流模型为基础,把时间纳入资源范畴,对活动、分支和路由结构进行时间资源分配。

*in* 和 *out* 分别表示工作流的入口和出口。活动节点形式化地描述为:  $node(i) = (DUR, E, L, opt.)$ ; 这里,  $E = (Ebf, Ebs, Ewf, Ews)$ ,  $L = (Lbf, Lbs, Lwf, Lws)$ ,  $opt. \in \{0, 1\}$ ,  $opt. = 1$  表示 *i* 是可跨越活动,  $opt. = 0$  时表示 *i* 不是可跨越活动。*DUR* 是建模阶段活动的执行延迟; *opt.* 标记可跨越活动; *E/L* 表示此活动的最早/最迟结束时间; *b/w* 表示若出现条件分支,则选择最优/最差分支运行; *f/s* 表示跨过/保留所有可跨越活动,并且,选择最优/最差可替换路径。

文中对该模型进行扩展,在并发结构和选择结构的分叉

收稿日期:2006-01-22; 修订日期:2006-04-03

作者简介:刘丹妮(1975-),女,黑龙江哈尔滨人,讲师,硕士,主要研究方向:工作流建模技术、工作流可达性分析方法、工作流资源分配策略; 陈秀寓(1977-),女,讲师,硕士,主要研究方向:工作流模型建立方法、工作流静态验证与动态验证技术。

汇合处以及各分支末尾处分别增加活动以标记时间资源。

## 2 时间资源分配策略

### 2.1 基本概念

对文献[8]的工作流程图进行扩展,在具有实际含义的具体活动基础上增加标记活动。

**定义 1** 标记活动:是一种只起到标记作用且没有实际含义的活动,它不被执行,只记录它所在分支的时间分配情况。例如:并发结构和条件结构的分叉点(*and\_split*, *or\_split*)、汇合点(*and\_join*, *or\_join*)以及顺序结构的末尾节点。

**定义 2** 具体活动:工作流程图中有实际含义,可被执行的活动称为具体活动(*activity*, 简称 *act*)。

资源分为可更新资源、不可更新资源和双重约束资源。时间的不可再生性决定了时间属于不可更新资源。作为一种资源,时间有其特殊性:

1) 可消耗性。时间也有折旧,例如有  $DUR(t) = 6$ ,则分配给事件  $t$  的时间资源  $s$  为 6,当  $t$  执行结束的时间为  $h = 6$ ,则  $t$  的执行使得  $s$  的折旧率达到了 100%;否则,  $(6 - h)$  作为  $t$  的松弛时间,并入  $t$  所在分支。

2) 可裂变。时间具有非原子性。活动、分支或路由结构执行未结束时出现资源耗尽,可向上一级申请资源支持,上级结构将松弛时间分裂出一部分分配给请求者的过程称为裂变。

3) 可聚合。活动、分支或路由结构执行结束时,剩余资源的权限提交给上一级并与上级资源合并的过程称为聚合。

时间资源分为三种:私有资源、公有资源和双重资源。

**定义 3** 私有资源:私有资源用来描述路由结构、分支或活动  $k$  的独有资源,记为  $TR(k)$ 。该资源一旦分配给  $k$ ,就被  $k$  独占,  $k$  执行结束时的松弛时间记为  $TSlack(k)$ ,可按照资源分配策略进行聚合。

**定义 4** 公有资源:公有资源用来描述 workflow 实例  $WF$  的松弛时间,记为  $TSlack(WF)$ 。所有活动实例或结构分支在资源不足时,都有权限按照裂变规则请求裂变  $TSlack(WF)$ 。

**定义 5** 双重资源:双重资源用来描述路由结构中任一分支  $\alpha$  的松弛时间,记为  $TSlack(\alpha)$ 。它具有私有资源和公有资源的双重特征。 $\alpha$  中的任一活动  $k$  未结束时出现  $TR(k)$  耗尽,可申请裂变  $TSlack(\alpha)$ 。同时,只有  $\alpha$  中的活动有权申请裂变  $TSlack(\alpha)$ ,  $\alpha$  分支执行结束时,  $TSlack(\alpha)$  剩余资源失效或者聚合,具体情况依据  $\alpha$  所在路由结构和资源分配策略而定。 $\alpha$  结束时的剩余资源记在该分支的标记活动  $k$  上,记为  $TSlack(k)$ 。

$TR(k) \in [TR_{min}(k), TR_{max}(k)]$ ,  $k$  为活动或分支。  
 $TR_{min}(k) = DUR_{min}(k)$ ,  $TR_{max}(k) = DUR_{max}(k)$ 。  
 $TSlack(k) \in [TSlack_{min}(k), TSlack_{max}(k)]$ 。

任意分支  $\beta$  的标记活动  $j$  描述为  $(TR_{min}(\beta), TR_{max}(\beta), TSlack_{min}(\beta), TSlack_{max}(\beta))$ 。

### 2.2 各种路由结构时间资源分析

#### 1) 顺序结构

顺序结构中只有一个分支  $\alpha$ ,按活动  $x_k (x_k \in (x_i, \dots, x_j))$  的执行时间分配私有资源  $TR(x_k)$ ,活动执行时出现资源不足时,请求裂变上层资源。因为  $DUR_{min}(k) = DUR_{max}(k) = DUR(k)$ ,所以有下述关系存在:

$$TR(x_k) = DUR(x_k);$$

$$TSlack(\alpha) = TR(\alpha) - \sum_{k=1}^j DUR(x_k)$$

#### 2) 并发结构

并发结构中存在多个并发分支,各分支分配相同的资源,由于活动实例不同,各分支的松弛时间会有不同(如图 1),每个分支的  $TR$  和  $TSlack$  记录在该分支末尾的标记活动上。以  $\alpha$  分支为例,资源分配情况如下:

$$TR_{min}(\alpha) = Ebf(j) - Ebf(i) - DUR(j);$$

$$TR_{max}(\alpha) = Ews(j) - Ebf(i) - DUR(j);$$

$$TSlack_{min}(\alpha) = TR_{min}(\alpha) - \sum_{k=1}^n DUR(a_k);$$

$$TSlack_{max}(\alpha) = TR_{max}(\alpha) - \sum_{k=1}^n DUR(a_k);$$

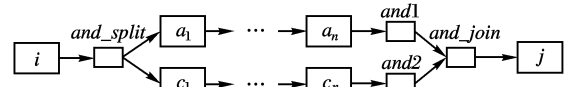


图 1 并发结构

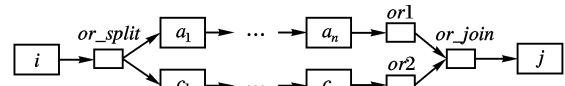


图 2 条件结构

#### 3) 条件结构

条件结构中存在多个分支,各分支资源分配情况由活动决定,松弛时间不尽相同。如图 2。

$$TR_{min}(\alpha) = Ebf(a_n) - Ebf(a_1) + DUR(a_1);$$

$$TR_{max}(\alpha) = Ews(a_n) - Ebf(a_1) + DUR(a_1);$$

$$TSlack_{min}(\alpha) = TR_{min}(\alpha) - \sum_{k=1}^n DUR(a_k);$$

$$TSlack_{max}(\alpha) = TR_{max}(\alpha) - \sum_{k=1}^n DUR(a_k);$$

#### 4) 循环结构

循环分支  $\alpha$  的最少执行次数为 0,最大执行次数由该分支的最大概率循环次数  $\lambda$  决定,  $\lambda$  根据经验确定。

$$TR_{min}(\alpha) = 0; TR_{max}(\alpha) = \lambda \sum_{k=1}^n DUR(i_k);$$

$$TSlack_{min}(\alpha) = 0; TSlack_{max}(\alpha) = 0$$

### 2.3 工作流图的时间分配策略

从 workflow 入口开始,依次访问每一个节点,按活动执行时间为其分配资源。遇到并发结构、选择结构或循环结构的分叉点时,找到相应的汇合点,分别依据各分支的特点,按照顺序结构分配时间资源。活动执行过程中资源不足或剩余,可向所在分支申请裂变或聚合双重资源。结构分支执行过程中资源缺乏或过剩,向上一层结构请求裂变或聚合双重资源和公有资源。

$Branch(or\_split)$  是条件结构中 *or\_split* 节点后继分支的集合,令  $Branch(or\_split) = \{\eta, \varphi\}$ ;  $Branch(and\_split)$  是并发结构中 *and\_split* 节点后继分支的集合,令  $Branch(and\_split) = \{\psi, \omega\}$ 。时间资源分配算法详细描述如下:

依次遍历每一个活动节点  $k$ ,

IF ( $k == act$ ) THEN

按照顺序结构的分配策略为  $k$  分配  $TR(k)$ ,  $k$  在执行过程中不断消耗  $TR(k)$ ,

当  $TR(k) = 0$  时,  $k$  未执行完,则请求裂变  $k$  所在分支  $\alpha$  的  $TSlack(\alpha)$ , 令  $TSlack(\alpha) = TSlack(\alpha) -$

$TR'(k), TR'(k)$  是根据实际情况确定的  $k$  所需要的后续资源。

当  $k$  执行结束时,  $TR(k) > 0$ , 则将剩余资源  $TSlack(k)$  与  $k$  所在分支  $\alpha$  的  $TSlack(\alpha)$  聚合, 令  $TSlack(\alpha) = TSlack(\alpha) + TSlack(k)$ 。

END IF

IF ( $k = or\_split$ ) THEN

找到与其对应的汇合点  $or\_join$ ;

对于任意  $\alpha \in Branch(or\_split)$ , 按照条件结构的分配规则, 找到各分支的  $TR_{min}(\alpha)$  和  $TR_{max}(\alpha)$ 。

$TR(\eta) = TR(\varphi) = Max(TR_{max}(\eta), TR_{max}(\varphi))$ ;

$TR(\varphi) = Max(TR_{max}(\eta), TR_{max}(\varphi))$ ;

$TSlack(\eta) = TD(\eta) - TR_{max}(\eta)$ ;

$TSlack(\varphi) = TD(\varphi) - TR_{max}(\varphi)$

$\alpha$  分支一旦执行, 其他非  $\alpha$  分支的  $TR$  和  $TSlack$  资源消亡。

$\alpha$  分支执行结束时, 剩余资源的权限作为该分支所在结构的剩余资源向上一层路由结构聚合,  $TSlack(or\_join) = TSlack(or1)$ ,  $or1$  是  $\alpha$  分支的标记活动。

END IF

IF ( $k = and\_split$ ) THEN

找到相对应的汇合点  $and\_join$ , 对于任意  $\alpha \in Branch(and\_split)$ , 按照并发结构的分配规则, 找到各分支的  $TR_{min}(\alpha)$  和  $TR_{max}(\alpha)$

$TR(\psi) = TR(\omega) = Max(TR_{max}(\psi), TR_{max}(\omega))$ ;

$TR(\omega) = Max(TR_{max}(\psi), TR_{max}(\omega))$ ;

$TSlack(\psi) = TD(\psi) - TR_{max}(\psi)$ ;

$TSlack(\omega) = TD(\omega) - TR_{max}(\omega)$ 。

并发结构各分支均执行结束时,  $TSlack(and\_join) = MAX(TSlack(and1), TSlack(and2))$ ,  $and1$  和  $and2$  分别是  $\psi$  和  $\omega$  的标记活动。  $TSlack(and\_join)$  作为并发结构的松弛资源向上层路由结构聚合。

END IF

IF ( $k = circle$ ) THEN

根据实际情况确定循环分支  $\alpha$  的最大概率执行次数  $\lambda$ , 依据循环结构的分配规则, 令  $TR(\alpha) = TR_{max}(\alpha)$ ,  $TSlack(\alpha) = 0$ 。

END IF

### 3 实例分析

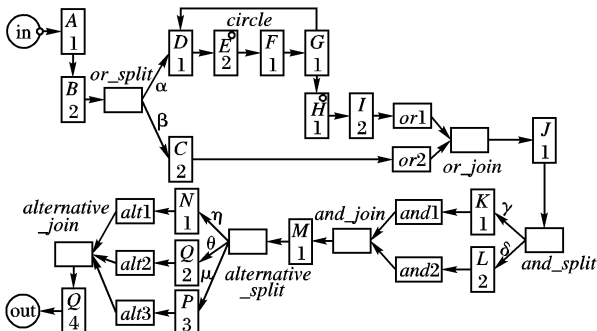


图3 一个生产销售实例

如图3, A 表示销售部接到客户订单, B 表示销售部查看库存量, 根据库存量选择不同的分支, 能够满足客户需求时,

执行活动 C (进入库房点货), 否则与客户沟通延期发货 (活动 D), 并告知销售部经理 (活动 E)、告知销售小组组长 (活动 F)、通知生产部门经理 (活动 G) 和生产小组组长 (活动 H), 然后组织生产 (活动 I), 其中 E 和 H 是可跨越的。两条条件分支汇合后为客户开出发货单 (活动 J), 告知客户准备发货 (K) 与货物装箱 (L) 并发执行, 填写销售登记表 (M) 后选择空运 (N)、陆运 (O) 或海运 (P) 方式运输, 最后与客户结账 (Q)。  $or1, or2, and1, and2, alt1, alt2$  和  $alt3$  分别是各分支标记节点。  $or\_split, or\_join, and\_split, and\_join, alternative\_split$  和  $alternative\_join$  分别是各路由结构的分叉和汇合节点。对具体活动的时间描述如表 1 所示。

表 1 活动节点的时间描述

node	活动的时间描述
A	(1, (1, 1, 1, 1), (1, 1, 1, 1), 0)
B	(2, (3, 3, 3, 3), (3, 3, 3, 3), 0)
C	(2, (5, 5, 5, 5), (5, 5, 8, 11), 0)
D	(1, (4, 4, 4, 4), (1, -2, 4, 4), 0)
E	(2, (4, 6, 4, 6), (1, 0, 4, 6), 1)
F	(1, (5, 7, 5, 7), (2, 1, 5, 7), 0)
G	(1, (6, 8, 6, 8), (3, 2, 6, 8), 0)
H	(1, (6, 9, 6, 9), (3, 3, 6, 9), 1)
I	(2, (8, 11, 8, 11), (5, 5, 8, 11), 0)
J	(1, (6, 6, 9, 12), (6, 6, 9, 12), 0)
K	(1, (7, 7, 10, 13), (8, 8, 11, 14), 0)
L	(2, (8, 8, 11, 14), (8, 8, 11, 14), 0)
M	(1, (9, 9, 12, 15), (9, 9, 12, 15), 0)
N	(1, (10, 10, 13, 16), (10, 12, 13, 18), 0)
O	(2, (11, 11, 14, 17), (10, 12, 13, 18), 0)
P	(3, (12, 12, 15, 18), (10, 12, 13, 18), 0)
Q	(3, (14, 16, 17, 22), (14, 16, 17, 22), 0)

从 in 开始遍历每一个活动, 依据活动执行时间为具体活动分配时间资源:  $TR(A) = 1, TR(B) = 2, \dots$ 。下面重点讨论各路由结构分配情况:

访问到  $or\_split$  节点, 找到与其对应的  $or\_join$ , 用标记活动  $or1$  和  $or2$  分别标记该条件结构中的两个分支  $\alpha$  和  $\beta$ ; 访问到  $and\_split$  活动, 找到对应的  $and\_join$ , 用  $and1$  和  $and2$  分别标记并发分支  $\gamma$  和  $\delta$ ; 访问到  $alternative\_split$  节点, 找到对应的  $alternative\_join$ , 用  $alt1, alt2$  和  $alt3$  分别标记各替换分支  $\theta, \eta$  和  $\mu$ 。标记活动记录所在分支可能分配拥有的最小资源、最大资源、实际分配的时间资源和松弛时间。标记活动分别描述如下:

$or1: (5, 8, 8, 0); or2: (2, 2, 2, 6); and1: (1, 1, 1, 1); and2: (2, 2, 2, 0); alt1: (1, 1, 1, 2); alt2: (2, 2, 2, 1); alt3: (3, 3, 3, 0); cicle: (3, 4, 4, 0)$

在实际执行过程中, 根据资源消耗情况可对双重资源或公有资源裂变和聚合。执行到  $or\_split$  节点时, 假设经过条件判断进入  $\beta$  分支。C 活动分配的资源是 2, 由于突发事件, 2 单位时间消耗完时, C 未执行结束。因为  $TSlack(\beta) = 6$ , 所以可向  $\beta$  分支申请裂变一定量的资源。申请数量如果大于 6, 多余部分要向  $TSlack(WF)$  申请。如果 C 活动执行完毕只消耗 1 单位, 剩余的 1 单位松弛时间与  $TSlack(\beta)$  聚合, 此时  $or2$  标记的  $TSlack(\beta) = 7$ ,  $\beta$  分支结束时,  $TSlack(\beta)$  向上与  $TSlack(or\_join)$  聚合, 最后与  $TSlack(WF)$  聚合。

(下转第 2171 页)

$N]$ ,按照变异概率取代抗体上原有的值,从而生成新抗体群  $D(t)$ 。若产生的解不满足组播树条件,则重新进行克隆变异操作。

(6) 克隆选择

对父代抗体群  $A(t)$  和变异后的克隆抗体群  $D(t)$  进行克隆选择操作,利用亲和度函数  $f(T)$  选出亲和度最高且互不相同的  $N$  个抗体组成新抗体群  $A(t + 1)$ 。

(7) 结束条件

计算  $A(t + 1)$  中每个抗体的亲和度,若新一代抗体中存在最优个体,则算法终止;否则转到(2)继续操作。

根据文献[6]证明,免疫多克隆策略算法的种群序列  $\{A(a), n \geq 0\}$  是有限齐次马尔可夫链,并且该算法是以概率 1 收敛的。

该算法的时间复杂度与抗体群规模  $N$ ,免疫克隆规模  $N_c$ ,算法的迭代次数  $t$  有关,所以算法的复杂度为  $O(NN_c t)$ 。

4 仿真实验及结果分析

本文采用 Matlab 7.0 作为仿真工具,分别采用文献[3]提出的改进遗传算法、标准遗传算法以及本文提出的基于免疫多克隆策略的路由算法。为了验证仿真结果,随机生成完全图的网络拓扑  $G$ ,节点数为 8 个,网络边的延迟代价为一个范围的随机整数,并且每个节点的度约束指定为  $[2, n/2]$  之间的一个随机整数,网络拓扑如图 1 所示。

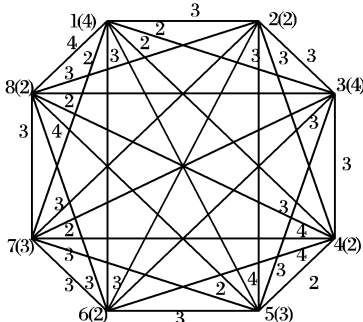


图 1 节点为 8 的网络

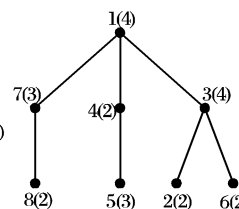


图 2 应用层组播树

各算法最大进化代数  $G_c = 400$ ,编码长度为 8。克隆算法的抗体群规模  $N = 60$ ,克隆系数  $\beta = 200$ ,由计算可得克隆后抗体群规模  $N_c = 237$ ,多克隆交叉和变异概率为 0.1。对于遗传算法染色体群规模取  $N = 235$ (约等于克隆算法中克隆后

的抗体群规模  $N_c$ ),且每次计算开始时遗传算法的初始群由克隆算法的初始抗体群复制得到,以使三种算法的初始数量和分布上处在同一水平上。遗传算法的突变概率为 0.05,交叉概率为 0.9。

仿真结果图 2 为采用免疫克隆算法得到的应用层组播树。通过进行 200 次仿真取平均值,得到三种算法的性能比较,如表 1 所示。从表中可以看出基于免疫多克隆策略的组播路由算法比遗传算法具有更快的收敛速度,并且解的性能要比其他两种算法高。

表 1 算法性能比较

	$G(50)$	$G(100)$	$G(150)$	$G(200)$	$G(250)$	$G(300)$	$G(350)$
本文算法	33	29	25	23	19	17	17
改进 GA	35	32	28	25	20	18	18
标准 GA	36	33	29	27	24	21	20

5 结语

应用层组播是目前网络研究领域的研究热点。本文对基于树的应用层组播路由算法进行了研究,建立了网络模型,提出基于免疫多克隆策略的带宽约束的应用层组播路由问题。本算法克服了遗传算法中的“早熟”现象,取得比遗传算法更好的结果,同时,该算法具有更快的收敛速度。实验结果证明,本算法具有较好的性能。

参考文献:

[1] DE CAATRO LN, VON ZUBEN FJ. Learning and Optimization using the clonal selection principle[J]. IEEE Transactions on Evolutionary computation, 2002, 6(3): 239 - 251.

[2] ZHU Y, WU MY, SHU W. Comparison study and evaluation of overlay multicast networks[A]. Multimedia and Expo. 2003. ICME'03 [C], 2003, (3): 493 - 496.

[3] 徐磊,章兢. 广义最小生成树的遗传算法求解及其应用[J]. 系统工程与电子技术, 2004, 26(3): 390 - 392.

[4] 潘耘,余镇危,王励成. 求解应用层组播路由问题的遗传算法[J]. 小型微型计算机系统, 2005, 26(1): 55 - 58.

[5] 余波,王东. 基于加权选择函数的应用层组播路由算法[J]. 计算机工程, 2005, 31(18): 105 - 107.

[6] 刘若辰,杜害峰,焦李成. 免疫多克隆策略[J]. 计算机研究与发展, 2004, 41(4): 571 - 576.

(上接第 2168 页)

4 结语

本文将时间纳入资源范畴,定为不可更新资源。通过对路由结构的分析,提出时间资源分配算法,在建模阶段进行资源分配,在执行过程中实现资源的动态裂变与聚合。时间是一种特殊资源,供活动消耗,同时决定着其他资源权限的开放与关闭。

参考文献:

[1] BRUCKER P, DREXL A, MOHRING R, et al. Resource-constrained project scheduling: Notation, classification, model, and methods[J]. European Journal of Operational Research, 1999, 112(3): 3 - 41.

[2] SENKUL P, TOROSLU IH. An architecture for workflow scheduling under resource allocation constraints [J]. Information System,

2005, 30(5): 399 - 422.

[3] LI H, YANG Y, CHEN TY. Resource constraints analysis of workflow specifications[J]. The Journal of Systems and Software, 2004, 73(2): 271 - 285.

[4] 刘胜,范玉顺,尹朝万. 基于工作流模型的资源配置优化方法[J]. 计算机集成制造系统, 2005, 11(9): 1272 - 1278.

[5] 李伟平,范玉顺. 基于工作流的资源受限项目调度研究[J]. 清华大学学报(自然科学版), 2004, 10(44): 1384 - 1388.

[6] 蔡方中,王正肖,孙永军. 虚拟企业运营过程中项目调度和执行策略[J]. 计算机工程, 2002, 5(28): 70 - 72.

[7] EDER J, PANAGOS E, POZEWAUNING H, et al. Time management in workflow systems [A]. ABRAMOWICZ W, ORLOWSKA ME, ed. Proceedings of the 3rd International Conference on Business Information Systems[C]. Berlin: Springer-Verlag, 1999. 265 - 280.

[8] 唐达,刘丹妮. 一种工作流时间截止期限的动态验证方法[J]. 计算机集成制造系统——CIMS, 2004, 9(10): 1154 - 1159.