

基于离散粒子群算法的工作流任务调度研究

刘环宇,侯秀萍

(长春工业大学 计算机科学与工程学院,吉林 长春 130012)

摘 要:工作流系统可以实现企业的业务过程的自动化,好的调度算法可以提高整个工作流系统的执行效率。对工作流任务调度的研究具有重要的意义。离散粒子群优化算法是一种收敛速度快、调整参数少、易实现及理解的优化算法。结合工作流特点,通过置换因子的概念,将离散粒子群算法与工作流任务调度相结合,给出了基于离散粒子群算法的工作流任务调度的定义。并结合保险理赔处理进行实验,得到了良好的效果,验证了应用的有效性和可行性。

关键词:工作流任务调度;置换因子;离散粒子群优化算法

中图分类号:TP311

文献标识码:A

文章编号:1673-629X(2010)05-0088-04

Research on Task Scheduling of Workflow Based on Discrete Particle Swarm Optimization

LIU Huan-yu, HOU Xiu-ping

(College of Computer Science and Engineering, Changchun University of Technology, Changchun 130012, China)

Abstract: Workflow system enables business process automatic. Good scheduling algorithm can improve the entire workflow system performance. Workflow task scheduling research has important significance. Particle swarm optimization algorithm is an optimization algorithm of fast convergence, less parameters in adjusting, easy to implement and understand. Integrated with workflow features, through the concept of replacement factor, defined workflow scheduling algorithm based on discrete particle swarm and the definition is applied to task scheduling of workflow and the handling of insurance claims, obtained good results, verify the effectiveness and feasibility of the application.

Key words: workflow task scheduling; factors of permutation; discrete particle swarm optimization algorithm

0 引言

工作流的概念起源于制造业和办公自动化领域中过程思想^[1],这种过程的思想自工业化生产以来就已经存在,主要侧重于工作中的各项活动的流程性方面,通过改进流程以提高效率。工作流是一类能够完全或部分自动执行的经营过程,它根据一系列过程规则,文档、信息或任务能够在不同的执行者之间传递与执行^[2]。通过建立工作流模型,可以将企业业务过程形式化为计算机可以处理的信息,实现企业业务过程执行的流程化、自动化,进而提高企业的工作效率和竞争力。工作流领域中的“资源”是指能够执行某些任务的实体,这些实体在实际的企业应用环境中可能是操作人、组织单元、加工设备、或者软件组件^[3]。工作流中的任务是指组成企业业务过程的工作逻辑单元,它

可以作为一个整体被资源执行。任务调度的目标是实现资源与任务的优化配置。

目前,对工作流的任务调度的研究已经得到了学者们的关注^[4,5],粒子群优化算法具有收敛速度快、调整参数少、易实现、易理解等优点,学者们试图用它去解决组合优化问题,并已经取得一些成果^[6],故文中将粒子群优化(Particle Swarm Optimization, PSO)算法应用到工作流任务调度之中。

1 工作流中的任务调度

一个工作流包括一组任务,以及它们之间的执行顺序关系、过程及活动的启动、终止条件、对每个活动的描述,活动间的输入、输出数据。一个工作流描述实际的应用中的一个具体业务流程,完成一个实际业务目标^[7]。

工作流中任务调度的实质是:在一定约束条件下,根据调度策略,工作流引擎将任务分配到合适的资源上执行,以达到高效的完成企业的业务活动的目的。

收稿日期:2009-08-25;修回日期:2009-11-28

基金项目:吉林省科技发展计划重大科技攻关项目(20040305)

作者简介:刘环宇(1983-),女,吉林松原人,硕士研究生,研究方向为软件工程;侯秀萍,教授,研究方向为软件工程。

当有 workflow 实例到来时, workflow 引擎调用规则库中的业务规则与该实例相匹配,并按照规则将其化划分为一系列相关的活动,确定活动执行的先后顺序,即任务和任务的调度次序与触发条件。进一步的将各个任务划分到相应的任务类中。系统中的资源按照职能进行分类,每一类资源完成一类任务。当有多个 workflow 实例同时到来时,将发生同类任务对同类资源的竞争,需要应用一定的调度策略,对系统中的任务调度,提高系统的运行效率。工作流及任务调度形式化描述如下:

定义工作流 $WF = \langle RuleSet, WFSet, TaskSet, ResourceSet, Relation, Data \rangle$, 其中:

RuleSet 为系统的规则库,由有限条规则构成。

WFSet 为系统中的工作流实例集合。

wf_i 为工作流实例,为 k 个任务的集合,对应于业务过程的一次执行。满足 $wf_i \in WFSet$, 且 $1 < i < n$, n 为系统中工作流实例的数目。 $task_{ik} \in wf_i$ 为 wf_i 中的第 k 个任务,对应于第 k 个任务类中的任务个体。

TaskSet 为系统中的任务类集合。

其中 $taskClass_k = \langle tId, tdescription \rangle$, $taskClass_k \in TaskSet$ 为系统中第 k 类任务, tId 为任务类号, $tdescription$ 为任务的描述。任务个体 $task_{ik} = \langle tId, wfId, ftime \rangle$, 其中 $task_{ik} \in taskClass_k$ 且 $task_{ik} \in wf_i$, tId 为任务类号, $wfId$ 为工作流实例号, $ftime$ 为任务的完成时间。

ResourceSet 为 workflow 系统中资源类集合,包含系统中所有的资源类。

$resource_k = \langle reId, description \rangle$, $resource_k \in ResourceSet$, 为系统中的第 k 类资源,其中 $reId$ 为资源类的编号, $description$ 为资源类的描述。资源个体 $reobj_{km} = \langle reId, reobjId, state, taskLsit \rangle$, 其中 $reobj_{km} \in resource_k$, $reId$ 为所属资源类号, $reobjId$ 为资源编号, $state$ 为资源状态, $taskLsit$ 为任务的等待队列。

$Relation = \{ \langle task_{12}, reobj_{21} \rangle, \langle task_{32}, reobj_{23} \rangle \dots \langle task_{ik}, reobj_{km} \rangle \}$ 为任务与资源的关联集合,描述任务的资源配备信息。

Data 为工作流中的相关数据,包括资源的状态信息,过程的流转信息以及需要由应用程序处理的业务数据。

某一时刻系统中存在 m 类资源 $ResourceSet = \{ resource_1, resource_2, \dots, resource_m \}$, n 个工作流实例 $WFSet = \{ wf_1, wf_2, \dots, wf_n \}$, 工作流实例 wf_n 根据匹配的业务规则划分为 k 个任务 $wf_n = \{ task_{n1}, task_{n2}, \dots, task_{nk} \}$, 并且确定每个任务的执行时间,对于每一个任务类 $taskClass_i (0 < i < k)$ 包含 k 个任务 $task_{1k}$,

$task_{2k}, \dots, task_{nk}$ 。当任务类中的任务数大于对应资源类中的资源数目时,将发生资源竞争,需要对工作流实例 wf_1, wf_2, \dots, wf_n 执行次序进行调度,生成最优调度序列 $\langle wf_{n-3}, wf_n, wf_1 \rangle$, 使 n 个过程实例的执行总时间(CT)最小。定义 StartTime 为调度序列中第一个实例执行的开始时间,EndTime 为最后一个实例的执行结束时间, n 个工作流实例的执行时间为 $CT = EndTime - StartTime$ 。

2 基于离散粒子群算法的工作流任务调度

传统的粒子群算法(PSO)主要用于解决连续空间上的问题,要将该算法应用在离散问题上,就需要对粒子群算法进行改造,参考文献[8]的定义,文中将工作流和离散粒子群算法相结合,基本定义如下:

(1) 粒子的位置 X 定义为序列 $\langle wf_1, wf_2, \dots, wf_n \rangle$ 表示 n 个工作流实例的一个调度序列。

(2) 置换因子 (p_i, p_k) , 其中 p_i 为序列 $\langle wf_1, wf_2, \dots, wf_n \rangle$ 中第 i 个位置。 p_k 表示序列中的第 k 个位置, (p_i, p_k) 表示交换序列中 p_i, p_k 位置上的元素。

(3) 粒子的速度定义为置换因子集合 $V = \{ (p_1, p_k), (p_4, p_{k-1}), \dots, (p_3, p_{k+1}) \}$, 置换集合的加法定义为两个置换集合的合集。

(4) 位置与速度的加法定义为置换因子集合对调度序列 $X_i = \langle wf_3, wf_8, \dots, wf_n \rangle$ 的作用,结果为新的调度序列 $X_{i+1} = \langle wf_n, wf_2, \dots, wf_{n-1} \rangle$, 如: $X_i = \langle wf_2, wf_3, wf_1, wf_4, wf_5 \rangle$, $V_i = \{ (p_1, p_3), (p_2, p_5), (p_3, p_4), (p_4, p_2), (p_5, p_1) \}$, $X_{i+1} = X_i + V_i = \langle wf_5, wf_3, wf_4, wf_2, wf_1 \rangle$ 。

(5) 粒子位置与位置的减法操作 $X_j - X_i$ 定义为 X_i 向 X_j 的转换,结果为置换因子集合。如: $X_i = \langle wf_2, wf_3, wf_1, wf_4, wf_5 \rangle$, $X_j = \langle wf_5, wf_3, wf_4, wf_2, wf_1 \rangle$, X_i 要变换为 X_j 执行减法操作 $X_j - X_i = \{ (p_1, p_3), (p_2, p_5), (p_3, p_4), (p_4, p_2), (p_5, p_1) \}$ 。

(6) c_1, c_2 与置换因子集合的乘积为集合中因子截取比例。若置换集合中的置换因子的数目为 m , $m * c_1$ 向上取整, $c_2 * m$ 向下取整且 $c_1 + c_2 = 1$, 如: $c_1 = 0.5$, 对于置换集合 $\{ (p_1, p_3), (p_2, p_5), (p_3, p_4), (p_4, p_2), (p_5, p_1) \}$ $m = 5$, $m * c_1 = 0.25$, 向下取整为 2, 截取置换集合中的两个置换因子 (p_1, p_3) 和 (p_2, p_5) 。

速度和位置的更新公式为:

$$V_i(t+1) = \omega V_i(t) + c_1(P_i - X_i(t)) + c_2(P_g - X_i(t))$$

$$X_i(t+1) = X_i(t) + V_i(t+1)$$

其中 P_i 为局部最优解, P_g 为全局最优解, c_1, c_2 为 (0, 1) 之间的数, ω 为惯性权重, 迭代过程按 $\omega = \omega_{\max} - L(\omega_{\max} - \omega_{\min})/L_{\max}$ 递减, L 为迭代次数, L_{\max} 为最大迭代次数。

适应度函数:

$$\min(CT) = \min(\text{EndTime} - \text{StartTime})$$

算法描述如下:

(1) 根据业务规则, 将 n 个工作流实例 wf_1, wf_2, \dots, wf_n 中的每一个 wf_n 划分为 k 个任务, 确定任务间的逻辑关系和触发条件, 并把 $n * k$ 个任务划分到相应的任务类中。

(2) 初始化粒子数 n , 初始化粒子 i 的位置为 $X_i = \langle x_1, x_2, \dots, x_n \rangle$ 。表示 n 个工作流实例的一个执行序列, 随机生成置换因子集合即速度, 粒子 i 的局部最优解为 X_i 本身, 全局最优解为 n 个局部最优解中适应度值最小的调度序列。

(3) 对于每个粒子计算其适应度函数值。若优于其局部最优解, 更新局部最优。根据每个粒子的局部最优解更新全局最优解。

(4) 对于每一个粒子根据位置和速度更新公式, 计算新的位置向量和速度向量。

(5) 若达到最大迭代次数或完成调度目标, 则转(6), 否则执行(3), (4)。

(6) 得到调度方案, 分配任务到资源队列。

3 算例分析

文中以保险理赔为例, 验证基于离散粒子群算法的工作流调度的可行性。

Petri 网是一种图形化的建模工具, 由于模型对带有并发性、异步性、分布式、非确定性、并行性系统的有力描述, 已成为目前最有前景的建模工具, 尤其适合于描述离散事件动态过程^[9]。故文中应用 Petri 网对保险理赔的一般流程进行建模, 结果如图 1 所示。

险情发生后, 用户通过拨打保险公司热线报案。确认在理赔范围内的案件, 保险公司将按以下流程进行处理, 如图 1 所示。其中: T1 用户报案, T2 现场勘察, T3 损失评估, T4 客户资料收集, T5 用户获得回执, T6 用户补交单据, T7 计算理赔成本, T8 核算理赔

金额及单证保险费, T9 经理审批通过, T10 交上级部门审批通过, T11 理赔部门交财务部门划款, T12 用户获取保险金。

设某一时刻有 5 个保险理赔案例到来。每个案例对应工作流中的一个工作流实例, 案例的每个处理过程, 对应于工作流实例中的一个任务, 每一个任务对应于一个任务类。每一个处理过程需要一个资源来执行, 每一个资源属于一个资源类。

初始化工作流如下:

$WFSet = \{wf_1, wf_2, wf_3, wf_4, wf_5\}$, wf_1 对应于医疗保险, wf_2 对应农业保险, wf_3 对应于财产保险, wf_4 对应于车险, wf_5 对应于人身意外保险。

$$\begin{aligned} wf_1 &= \{\text{task}_{11}, \text{task}_{12}, \text{task}_{13}, \text{task}_{14}\}, \\ wf_2 &= \{\text{task}_{21}, \text{task}_{22}, \text{task}_{23}, \text{task}_{24}\}, \\ wf_3 &= \{\text{task}_{31}, \text{task}_{32}, \text{task}_{33}, \text{task}_{34}\}, \\ wf_4 &= \{\text{task}_{41}, \text{task}_{42}, \text{task}_{43}, \text{task}_{44}\}, \\ wf_5 &= \{\text{task}_{51}, \text{task}_{52}, \text{task}_{53}, \text{task}_{54}\} \end{aligned}$$

其中 task_{ij} ($1 < i < 5, 1 < j < 4$) 表示第 i 个工作流实例的第 j 个任务, 由资源类 resource_i 中的资源个体执行, 对应的每个任务的完成时间如下(单位为小时), 矩阵的每一行对应一个工作流实例 wf , 每一列对应于一类任务:

$$\begin{bmatrix} 2 & 3 & 1.5 & 0.5 \\ 4 & 3 & 1.5 & 0.5 \\ 3 & 3.5 & 2 & 0.6 \\ 2.5 & 2 & 1 & 0.4 \\ 3 & 3.5 & 1.5 & 0.6 \end{bmatrix}$$

$\text{resource}_1 = \langle 1, \{\text{ReObj}_1, \text{ReObj}_2, \text{ReObj}_3\}, \text{查勘员} \rangle$, $\text{resource}_2 = \langle 2, \{\text{ReObj}_1, \text{ReObj}_2\}, \text{理算员} \rangle$, $\text{resource}_3 = \langle 3, \{\text{ReObj}_1, \text{ReObj}_2\}, \text{核赔员} \rangle$, $\text{resource}_4 = \langle 4, \{\text{ReObj}_1, \text{ReObj}_2\}, \text{财务人员} \rangle$ 。

初始化粒子群算法的各个参数为, 粒子数为 $n = 10$, $c_1 = c_2 = 0.5$, $\omega_{\max} = 0.9$, $\omega_{\min} = 0.4$, $L_{\max} = 10$ 置换集合中的置换因子最大数目为 7, 初始化粒子 10 个粒子分别为 $\langle 12345 \rangle$, $\langle 12435 \rangle$, $\langle 12543 \rangle$, $\langle 13425 \rangle$, $\langle 14523 \rangle$, $\langle 23415 \rangle$, $\langle 23514 \rangle$, $\langle 34512 \rangle$, $\langle 24513 \rangle$, $\langle 13524 \rangle$, $\langle 12345 \rangle$ 为调度序列 $\langle wf_1, wf_2, wf_3, wf_4, wf_5 \rangle$ 的简化形式。

粒子迭代结果如表 1; 实验结果如表 2。

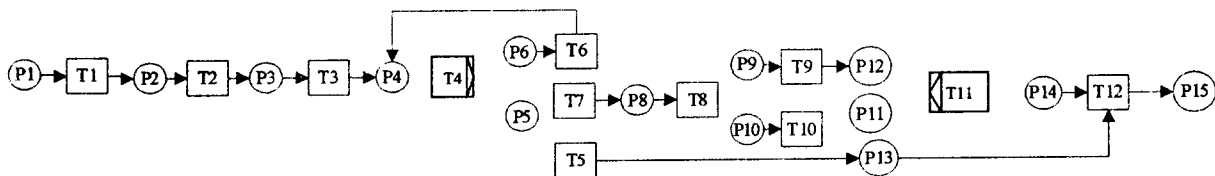


图 1 保险理赔处理流程

表 1 粒子迭代结果

迭代次数	粒子 1	粒子 2	粒子 3	粒子 4	粒子 5	粒子 6	粒子 7	粒子 8	粒子 9	粒子 10	全局最优解	执行时间(h)
1	23145	25143	35412	52134	25143	32541	34125	43251	41325	45231	32541	12.9
2	21453	41325	34251	54321	31245	23415	14325	21543	45231	13425	41325	13
3	15324	13524	15423	35241	23154	32541	32154	41523	13524	14352	32541	12.9
4	43152	35412	54123	34251	32415	23415	21453	53412	25413	15324	35421	13
5	41523	23514	15324	14325	13524	31542	42135	35142	52431	42315	23514	12.9
6	43215	15324	45213	15432	32415	23541	15432	53214	25431	15234	53214	12.9
7	32541	32451	32154	51243	35412	14532	14253	54321	53412	32541	32541	12.9
8	23514	14523	15234	34251	43152	23541	53241	12543	14253	21543	23514	12.9
9	32514	43125	23154	12453	13542	34521	41352	32451	53421	13524	32512	12.9
10	23514	24513	13254	24153	14523	24531	23541	52413	25413	21534	23541	12.9

表 2 实验结果

	最优解	最差解	平均解
处理时间(单位 h)	12.9	14.1	13.29

选取第十次迭代获得最优调度序列 $< wf_2, wf_3, wf_5, wf_4, wf_1 >$ 为最终的调度方案,资源如下:

Relation = { (task₁₁, reobj₁₂), (task₁₂, reobj₂₂), (task₁₃, reobj₃₂), (task₁₄, reobj₄₂), (task₂₁, reobj₁₁), (task₂₂, reobj₂₁), (task₂₃, reobj₃₁), (task₂₄, reobj₄₁), (task₃₁, reobj₁₂), (task₃₂, reobj₂₁), (task₃₃, reobj₃₁), (task₃₄, reobj₄₂), (task₄₁, reobj₁₃), (task₄₂, reobj₂₁), (task₄₃, reobj₃₁), (task₄₄, reobj₄₁), (task₅₁, reobj₁₃), (task₅₂, reobj₂₂), (task₅₃, reobj₂₂), (task₅₄, reobj₄₁) }。

4 结束语

离散粒子群算法具迭代速度快,易于实现的优点,工作流任务的调度是工作流研究的重点内容。文中将二者相结合,给出了工作流的定义,将离散粒子群算法应用工作流任务调度的过程中,并通过实验证明了应用的有效性及其合理性。

参考文献:

[1] Georgakopoulos D, Homick M. An Overview of Workflow

Management: From Process Modeling to Workflow Automation Infrastructure [J]. Distribute and Parallel Databases, 1995(3):119-153.

[2] Workflow Management Coalition. Workflow management coalition terminology and glossary[R]. Brussels: Workflow Management Coalition, 1996.

[3] 吴立峰,金 烨. 工作流系统的资源管理问题[J]. 计算机工程,2007(10):58-63.

[4] 肖郑进,何钦铭,陈 奇. 模糊环境中工作流任务分配的多级模型[J]. 计算机研究与发展,2007,44(2):302-309.

[5] 刘丹妮,陈秀寓. 一种基于工作流图的时间资源分配策略[J]. 计算机应用,2006,26(9):2166-2171.

[6] Cagnina L, Esquivel S, Gallard I R. Particle Swarm Optimization for Sequencing Problems: A Case Study[C]// In: Proc of the Congress on Evolutionary Computation. Oregon, Portland:[s. n.], 2004:536-541.

[7] 周 燕,姜 浩. 面向动态工作流的 Web 服务组合模型研究[J]. 计算机技术与发展,2008,18(1):63-66.

[8] 宋佩华,蒋联源,欧启忠. 基于离散粒子群优化算法求解矩形件排样问题[J]. 计算机应用与软件,2008,25(1):238-240.

[9] 乐晓波,李京京,唐贤瑛. 基于 Petri net 建模的资源调度的蚁群算法[J]. 计算机技术与发展,2006,16(1):44-46.

(上接第 87 页)

[9] Poli R, Kennedy J, Blackwell T. Particle swarm optimization [J]. Swarm Intell,2007(1):33-57.

[10] 黄辉先,陈资滨. 一种改进的粒子群优化算法[J]. 系统仿真学报,2007,19(21):4922-4925.

[11] 吕振肃,侯志荣. 自适应变异的粒子群优化算法[J]. 电子学报,2004,32(3):416-420.

[12] Marques L, Nunes U, de Almeida A T. Particle swarm-based olfactory guided search[J]. Auto Robot,2006,20:277-287.

[13] 陆宗骥,金登南. Visual C++ .NET 图像处理编程[M]. 北

京:清华大学出版社,2006.

[14] EI-Gallad A, EI-Hawary M, Sallam A, et al. Enhancing the particle swarm optimizer via proper parameters selection[C]// IEEE CCECE02 Proceedings. Canadian: IEEE Press, 2002: 792-797.

[15] 彭 宇,彭喜元,刘兆庆. 微粒群算法参数效能的统计分析[J]. 电子学报,2004,32(2):209-213.

[16] 王 凌,刘 波. 微粒群优化与调度算法[M]. 北京:清华大学出版社,2008:45-50.