

启发式规则与 GA 结合的优化方法求解工作流动态调度优化问题^{*})

肖志娇 常会友 衣 杨

(中山大学信息科学与技术学院计算机科学系 广州 510275)

摘 要 调度是 workflow 管理系统的核心问题,是保证 workflow 正确运行的关键。在 workflow 环境下,动态调度要比静态调度更切合实际。本文在总结前人工作的基础上,提出了一系列工作流动态调度的启发式规则,并以最小化任务总拖期时间和最大化任务总提前时间为目标,建立了工作流动态调度问题模型。采用启发式规则与遗传算法相结合的优化方法求解工作流动态调度优化问题。仿真结果说明了优化方法的可行性和有效性,同时比较了该方法与多种静态调度方法,进而说明了该方法的优越性。

关键词 工作流,动态调度,启发式,遗传算法

An Optimization Method of Workflow Dynamic Scheduling Based on Heuristic GA

XIAO Zhi-Jiao CHANG Hui-You YI Yang

(School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510275)

Abstract Scheduling of workflows is an important issue in workflow management. Many researches have been done in this area, but most of them are confined to static method of workflow scheduling. Actually, dynamic scheduling methods are more reasonable than static ones. Based on existed researches, some heuristic rules of workflow dynamic scheduling are presented. An optimization method based on Genetic Algorithm (GA) is proposed to schedule workflows dynamically. Examples are given to illustrate the feasibility and validity of the method. Experiment results show that the method has good performance and outperformed static methods.

Keywords Workflow, Dynamic scheduling, Heuristic, Genetic algorithm

1 引言

workflow 管理系统的目标是在正确的时间将正确的任务分配给正确的资源执行。工作流调度就是要找到一个满足业务逻辑约束的工作流任务执行的正确序列^[1]。

调度在现实环境中几乎无处不在。调度问题是最有名的组合优化问题之一,它一直是优化领域研究的热点。研究人员针对不同领域的调度问题提出了各种优化方法。但工作流调度问题与其它领域的调度问题相比,存在许多不同^[2,3]。

首先,工作流是动态的,它具有较多的不确定性。这些不确定性,在传统的调度问题中是不存在的。

(1)工作流的不确定性:工作流实例的到达是不确定的;工作流执行路径和任务执行次数是不确定的。

(2)资源的不确定性:由于工作流执行过程有时可能会持续较长的时间,可用的资源、数量和组成在这段时间里都有可能发生变化;许多因素将对任务的分配产生影响,例如资源的能力,任务的难度,资源当前的负荷,资源的兴趣等,而这些因素往往都是动态变化的;任务与资源间是多对多的关系。

其次,工作流结构通常包含并行分支,各个工作流的实例也会并发执行,这也是传统的调度中没有的情况。

以自然选择和遗传理论为基础的遗传算法是由 Michigan 大学的 Holland 教授及其学生于 1975 年创建的。它是将

生物进化过程中适者生存规则与群体内部染色体的随机信息交换机制相结合的搜索算法。近年来,遗传算法已经被成功地应用于解决多个不同领域的调度问题,因此,本文将利用遗传算法来求解工作流动态调度问题。

本文总结了工作流调度中的一些启发式的规则,设计了一个启发式规则与遗传算法结合的、全局最优的工作流动态调度算法。本文的内容组织如下:第 2 节介绍工作流调度问题的相关知识及已有的研究;第 3 节对工作流动态调度优化问题进行描述,提出启发式规则,并建立问题的数学模型;第 4 节结合启发式规则,采用遗传算法对问题进行求解;第 5 节使用实例说明该优化方法的可行性和有效性,并比较了本文方法与以往的静态分配方法、逐个任务分配方法等其他工作流任务分配方法;最后总结全文。

2 工作流调度

逻辑^[4]、Petri 网^[5]等都被用来对工作流进行调度。但这些研究大多数是只考虑了满足时间和结构约束的工作流静态调度。Senkul^[1]等人提出了一种考虑工作流资源约束的预调度的方法。该方法也属于一种静态调度方法,它无法处理工作流的动态特性,也没有考虑系统中可能存在的多个并发工作流的情况。Baggio^[3]也采用静态调度的方法,使用猜测的方式来处理工作流的不确定性。段永强^[2]等人提出了一种基

^{*})国家自然科学基金项目(60573159)、广东省自然科学基金(05100302)。肖志娇 博士研究生,主要从事工作流、软计算、模型分析与优化等研究;常会友 博士,教授,博士生导师,主要研究领域为协同理论与技术、工作流、信息系统集成;衣 杨 博士,副教授,硕士生导师,主要研究领域为系统优化与建模、智能优化及软计算。

于资源约束的动态任务调度方法。该方法首先从当前待执行任务中挑选最优的可执行任务的集合,然后在考虑资源兴趣的情况下,根据资源能力和任务难度对任务进行分配。

工作流调度包含任务分配和任务排程两个部分。工作流任务分配,即给工作项分配资源^[6],是工作流调度的核心问题和关键环节。工作流任务分配有直接的静态分配和间接的动态分配两种形式。直接的静态分配在建模时就为任务指定具体的执行资源。这种方法过程模型与资源模型的耦合度过高,导致系统不灵活,资源利用率不高。间接的动态分配在建模时仅给出可用资源的描述,运行时根据实际情况,选取合适的资源来执行任务,从而降低了模型间的耦合度,比较灵活,能充分利用资源。目前多数工作流管理系统均采用后者。

截止目前,工作流任务分配一般采用简化的方式获得分配结果,例如选取最小负荷、最近可用的资源等,显然其结果不是最优解;文^[7,8]采用基于综合评价的方法,但是由于它逐个、孤立的为待分配资源的任务选取合适的执行资源,而在某一时刻,系统中可能存在多个待分配资源的任务,所以该方法同样无法实现全局范围内的最优。

任务排程,即安排在同一资源上的各个任务的执行顺序。排程也是现实中广泛存在的问题。常用的排程规则有^[9]:先进先出(FIFO)、后进先出(LIFO)、最短处理时间(SPT)、最早截至期限(EDD)等。也有一些专家学者提出了一些复杂的排程原则,对于工作流任务的排程均具有借鉴意义。

在工作流调度中,任务分配和任务排程这两个方面是紧密联系的^[9]。它们相互影响,进而对优化调度的结果共同产生影响。因此,采用传统的方法对它们分别单独进行优化是无法取得优化效果的。本文设计了一种启发式规则与遗传算法结合的工作流动态调度优化算法。以任务的拖期时间总和最小以及任务的超前时间总和最大为目标,将工作流调度涉及的两个方面统一起来进行同步优化。

3 工作流动态调度优化规则及数学模型

3.1 工作流动态调度问题的描述

本文研究的工作流调度问题可描述如下:工作流管理系统中有待执行的任务 j ($j=1, \dots, m$), m 为待执行任务总数,可分配资源 i ($i=1, \dots, n$), n 为资源总数。总的资源集合为 R , $RT_j \in R$ 是可分配给任务 j 的资源集合。问题的目标是从 RT_j 中挑取合适的资源 r_j ($r_j=1, \dots, n$), 执行任务 j , 使任务的拖期时间总和最小, 以及任务的超前时间总和最大。

3.2 工作流动态调度问题的优化规则

工作流任务的时间裕度反映了其拖期时间和提前时间。任务 j 的时间裕度用 ts_j 表示。

规则1 任务 j 的时间裕度为任务的期望完成时间减去其执行时间与资源的可用时间之和,即

$$ts_j = te_j - t_{ij} - tr_i \quad (1)$$

其中, te_j 是任务 j 的期望完成时间; t_{ij} 是资源 i 执行任务 j 的执行时间; tr_i 是资源 i 的可用时间; i 是执行任务 j 的资源, $i \in RT_j$ 。

不同的资源拥有不同的技能,在同一技能上通常也具有不同的水平,因此不同资源完成不同任务所需时间不同。资源 i 执行任务 j 的执行时间 t_{ij} 主要由任务的要求、资源的能力、资源的偏好等因素确定。任务的要求描述该任务的工作量及要求完成质量方面的信息。资源能力描述资源具备某项技能的水平,执行某项任务的效率等方面的信息。资源偏好

描述资源在选择任务执行时的倾向。偏好是一个主观的因素,但也可以从资源目前的负荷、以往的选择等几个方面对其进行一些定量的评价。

规则2 资源 i 执行任务 j 的执行时间为任务的工作量与执行任务资源的能力之比。考虑到若任务符合资源的偏好,则资源执行速度提高的缘故,将 t_{ij} 计算如下:

$$t_{ij} = \alpha \cdot (1 - p_{ij}) \cdot (d_j / c_{ij}) \quad (2)$$

其中, d_j 表示任务 j 的工作量; c_{ij} 表示资源 i 完成任务 j 的能力; p_{ij} 表示资源 i 对任务 j 的偏好,其值大于0 小于1,值越大则越偏好; α 为调整系数; i 是执行任务 j 的资源, $i \in RT_j$ 。

规则3 资源 i 的可用时间 tr_i 取决于当前时间、资源的空闲时间等,即

$$tr_i = \begin{cases} tc + \sum_{k=1}^e t_{ik}, & tv_i < tc \\ tv_i + \sum_{k=1}^e t_{ik}, & tv_i \geq tc \end{cases} \quad (3)$$

其中, tv_i 是资源 i 空闲时间; tc 是当前时间; $\sum_{k=1}^e t_{ik}$ 是本次分配中已经分配给资源 i 的任务的执行时间总和; e 是已分配给资源 i 的任务数。

为了对任务分配和任务排程进行统一优化求解,引入虚资源的概念。设有资源 i 可参与 s 个任务的执行,则加入 $(s-1)$ 个该资源的虚资源,将资源和虚资源统一表示为 $i.k$ ($k=0, \dots, s-1$), 其中 $i.0$ 即为资源 i 。需要注意的是,首先要确定 s 是哪个数量级的,然后确定 k 为几位小数。资源在执行任务时,按 $i.k$ 值由小到大,依次执行分配给资源 $i.k$ 的任务。那么,对资源的可用时间的计算规则3 进行修正,成为规则3.1。

规则3.1 资源 $i.k$ 的可用时间 tr_{ik} 取决于当前时间、资源的空闲时间和分配给资源 $i.h$ ($h < k$) 的任务的执行时间,即

$$tr_{ik} = \begin{cases} tc + \sum_{h=0}^{k-1} t_{ih}, & tv_i < tc \\ tv_i + \sum_{h=0}^{k-1} t_{ih}, & tv_i \geq tc \end{cases} \quad (4)$$

其中, tv_i 是资源 i 空闲时间; tc 是当前时间; t_{ih} 是分配给资源 $i.h$ 的任务的执行时间。

前面讨论的都是一个任务的执行只需要一个资源参与的情况。若任务需要多个资源的参与,则将资源组成团队参与任务的执行。团队 wt_g 是多个协同工作的资源的集合。团队有几种工作模式:同时同地工作、同时不同地工作、不同同时同地工作、不同时不同地工作。异地工作的团队间的协作主要取决于工作流系统对分布式的支持。这里我们只考虑同时与不同时工作的团队的区别。团队成员必须同时工作时,如多人会议,团队资源的可用时间取决于团队成员中最后可用资源的可用时间。团队资源执行任务的时间与团队内各组成资源的能力、偏好有关,同时还取决于团队中各资源间的合作能力。团队成员无须同时工作时,如协同设计,通常可将任务划分为相互关联的成员资源各自的工作,看作是一个子工作流程,只是此时资源执行任务的时间与其他任务所选取的资源相关。这里,我们只考虑团队资源同时工作的情况。

规则4 团队 wt_g 执行任务 j 的执行时间 $t_{wt_g j}$ 取决于 wt_g 中各成员资源的能力和偏好与任务要求的匹配度,及资源间的合作能力,将 $t_{wt_g j}$ 计算如下:

$$t_{wt_g j} = \sum_{h=1}^{|wt_g|} (\lceil a_{wt_g h} \cdot (1 - p_{wt_g h}) \cdot$$

$$(d_{wt_{gh}}/c_{wt_{gh}}) \cdot s_{wt_{gh}} \cdot \alpha_{wt_{gh}} \quad (5)$$

其中, $\alpha_{wt_{gh}}$ 为调整系数; $p_{wt_{gh}}$ 表示资源 wt_{gh} 对任务 j 的偏好, 其值大于 0 小于 1, 值越大则越偏好; $d_{wt_{gh}}$ 表示任务 j 由资源 wt_{gh} 所执行部分的工作量; $c_{wt_{gh}}$ 表示资源 wt_{gh} 完成任务 j 的能力; $s_{wt_{gh}}$ 是对资源 wt_{gh} 与 wt_g 中其它资源合作能力的综合评价; $\alpha_{wt_{gh}}$ 是资源 wt_{gh} 所执行部分所占任务 j 的比重, $\sum_{h=1}^{|wt_g|} \alpha_{wt_{gh}} = 1$; $|wt_g|$ 表示 wt_g 中成员资源的数量; wt_{gh} 是作为成员加入 wt_g 中参与任务 j 执行的第 h 个资源, $wt_{gh} \in wt_g$ 。

规则 5 团队 wt_g 的可用时间 tr_{wt_g} 取决于团队中最后可用的资源的可用时间, 即

$$tr_{wt_g} = \max(tr_{wt_{gh}}), h=1, \dots, |wt_g| \quad (6)$$

其中, $tr_{wt_{gh}}$ 是资源 wt_{gh} 的可用时间, $|wt_g|$ 表示 wt_g 中资源的数量; wt_{gh} 是作为成员加入 wt_g 中参与任务 j 执行的资源, $wt_{gh} \in wt_g$ 。

3.3 工作流动态调度问题的数学模型

定义 1 拖期矩阵 a 大小为 $m \times 1$ 的矩阵。

$$a_j = \begin{cases} 1, & ts_j < 0 \\ 0, & ts_j \geq 0 \end{cases} \quad (7)$$

定义 2 超前矩阵 b 大小为 $m \times 1$ 的矩阵。

$$b_j = \begin{cases} 1, & ts_j < 0 \\ 1, & ts_j \geq 0 \end{cases} \quad (8)$$

工作流系统的目标就是尽可能快捷地完成任务^[9]。因此, 优化拖期工作是工作流调度优化的目标^[3]。这里采用两级优化目标, 目标函数可描述如下, 分别表示最小化拖期时间总和和最大化提前时间总和:

$$\min \sum_{j=1}^m (a_j \cdot w_j \cdot \theta_j \cdot |ts_j|) \quad (9)$$

$$\max \sum_{j=1}^m (b_j \cdot w_j \cdot \theta_j \cdot ts_j) \quad (10)$$

其中, w_j 为任务 j 所属项目 P_j 的重要度, 它是根据统一评价指标体系, 为工作流系统中存在的各个项目赋予的重要度, 评价指标包括项目收益、项目拖期惩罚等; θ_j 为任务 j 对所属项目的影晌度, 根据任务在项目中所处位置、任务执行时间占流程总时间的比例等因素确定^[2]; ts_j 为任务 j 的时间裕度, $|ts_j|$ 为任务 j 的时间裕度的绝对值。

则工作流动态调度优化问题的总体数学模型可描述如下:

$$\text{WTAP; } \min f_1(r_j) = \min \sum_{j=1}^m (a_j \cdot w_j \cdot \theta_j \cdot |ts_j|) \quad (11)$$

$$\max f_2(r_j) = \max \sum_{j=1}^m (b_j \cdot w_j \cdot \theta_j \cdot ts_j) \quad (12)$$

$$\text{s.t. } r_j \neq r_k, \text{ if } j \neq k \quad (13)$$

$$ts_j = te_j - tr_{r_j} - tr_j \quad (14)$$

$$t_{ij} = \alpha \cdot (1 - p_{ij}) \cdot (d_j / c_{ij}) \quad (15)$$

$$tr_{wt_{gh}} = \sum_{h=1}^{|wt_g|} ([\alpha_{wt_{gh}} \cdot (1 - p_{wt_{gh}}) \cdot (d_{wt_{gh}} / c_{wt_{gh}}) \cdot s_{wt_{gh}}] \cdot \sigma_{wt_{gh}}) \quad (16)$$

$$tr_{wt_g} = \begin{cases} tc + \sum_{h=0}^{k-1} t_h, & tv_i < tc \\ tv_i + \sum_{h=0}^{k-1} t_h, & tv_i \geq tc \end{cases} \quad (17)$$

$$tr_{wt_g} = \max(tr_{wt_{gh}}), h=1, \dots, |wt_g| \quad (18)$$

$$a_j = \begin{cases} 1, & ts_j < 0 \\ 0, & ts_j \geq 0 \end{cases} \quad (19)$$

$$b_j = \begin{cases} 0, & ts_j \leq 0 \\ 1, & ts_j > 0 \end{cases} \quad (20)$$

其中, 式(11)和式(12)是问题的两级目标函数; 式(13)是由于虚资源的引入, 需要保证一资源一任务的约束条件; 式(14)是当前调度方案下, 任务 j 的时间裕度的求解公式; 式(15)是资源 i 执行任务 j 所需时间的求解公式; 式(16)是资源团队 wt_g 执行任务 j 所需时间的求解公式; 式(17)是资源可用时间的求解公式; 式(18)是资源团队 wt_g 可用时间的求解公式; 式(19)和式(20)分别为拖期系数矩阵和超前系数矩阵。

不同的应用领域, 不同的工作流会有不同的约束。这些约束条件大致可以分为两种: 第一种是硬约束。硬约束是要求一定要满足的约束。这类约束有可以分为绝对约束和相对约束。绝对约束是对单个对象的约束, 如任务 j 必须由资源 i 执行, 考虑到虚资源, 则可表示为 $\lfloor r_j \rfloor = i$ 。相对约束是对多个对象间的约束, 如任务 j 和 k 必须由同一资源执行, 即 $\lfloor r_j \rfloor = \lfloor r_k \rfloor$ 。另一种是软约束。软约束通常是一些商业规则, 如任务 j 和 k 由同一资源执行, 则执行时间减少为原来的 90%。在此无法对约束条件一一进行形式化描述。应针对具体的问题, 建立具体的约束模型。

4 启发式规则与 GA 结合的工作流动态调度优化算法

4.1 编码

编码采用基于任务的编码方式。由于一个任务可能需要一个团队资源的参与, 而组成团队的资源成员可能有多个, 因此需要进行优化选择。假设一个任务所需的不同资源的个数是固定的, 设任务 j 需要 n_j 个资源的参与, 则染色体长度为 $\sum_{j=1}^m n_j$, 其中 m 为待分配的任务的总数。分配给第 j 个任务的资源是第 $\sum_{h=1}^{j-1} n_h + 1$ 到 $\sum_{h=1}^j n_h$ 个基因位的取值。

4.2 适应函数

采用基于排序的适应值分配方法来评价每个染色体的价值。首先计算各个染色体的目标函数值; 然后根据第一级目标函数值由小到大对个体进行排序。如果某些个体具有相同的目标函数值, 则它们根据第二级目标函数值由大到小进行排序。如果某些个体最终无法排序, 则让它们随机排序。设 r_k 是个体 x_k 的顺序 ($k=1, 2, \dots, np$), 则基于顺序的适值函数如下:

$$f(x_k) = 100 * (f_{1np} - f_{1k}) + 10 * (f_{2k} - f_{2np}) + f(x_{np}) \quad (21)$$

$$f(x_{np}) = 1 \quad (22)$$

4.3 遗传算子

选择采用二进制竞争选择策略。变异算子采用基本位变异。交叉算子采用改进的熔合算子^[10]。由于不同的问题具有不同的约束条件, 就会由于交叉和变异而产生一些不可行解。对于此类不可行解, 应针对具体的约束采用具体的处理方法。

4.4 算法步骤

具体算法步骤如下:

步骤 1. 根据 4.1 进行预处理;

步骤 2. 初始化: NP 为初始种群大小, NG 为迭代次数, P_m 为变异概率, 生成初始种群 A_0 ;

步骤 3. 计算种群各个个体的适值 f ;

步骤 4. 采用二进制竞争选择策略从种群中选出两个父染色体 P_1 和 P_2 ;

- 步骤 5. 对选出的两个染色体进行交叉生成子染色体 C;
- 步骤 6. 以 P_m 概率对 C 进行变异;
- 步骤 7. 如果 C 不可行, 则进行修补, 否则继续下一步;
- 步骤 8. 如果 C 与种群中任意解相同, 则继续下一步, 否则, 从种群中随机选出一个适应值劣于 C 的个体并用 C 替代, 并令 $i=i+1$;
- 步骤 9. 重复步骤 4 到 8 直到 $i=NG$.

5 仿真实验及结果分析

表 1 任务及其可用资源

任务	可用资源
1(5)	1(14), 15(15)
2(17)	2(18), 6(18), 8(12), 11(10), 12(18), 14(17), 17(13), 20(17)
3(13)	1(14), 2(7), 3(6), 4(7), 5(11), 6(15), 7(7), 8(17), 9(12), 10(8), 11(15), 12(11), 13(9), 14(14), 15(13), 18(16), 20(20)
4(3)	9(11)
5(5)	9(18), 10(4), 19(20)
6(13)	5(6), 13(6), 14(18), 18(15), 20(3)
7(13)	4(1), 13(18), 17(4), 20(6)
8(8)	1(14), 2(6), 5(10), 6(2), 7(20), 9(12), 10(9), 12(11), 13(7), 15(9), 16(5), 18(12), 20(16)
9(12)	1(11), 2(13), 4(5), 8(8), 13(16), 14(14)
10(10)	4(10), 10(12), 14(16), 15(8)

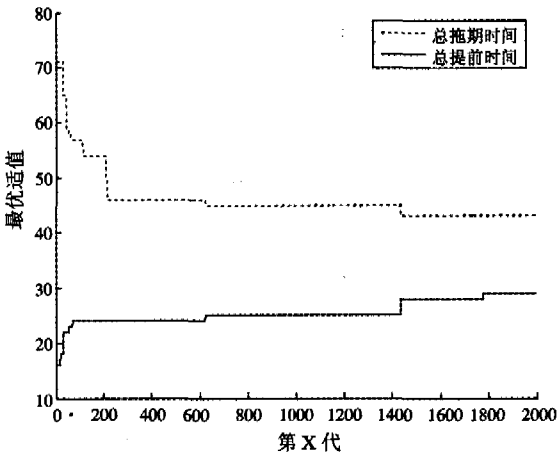


图 1 各代种群的最优适应值的走势图

本文在实际运行系统中随机选取一个时刻的状态作为实例来验证本文方法的有效性和优越性。表 1 是该时刻待分配的任务及其可用资源。其中任务一栏括弧中的数字表示任务的预期完成时间, 可用资源一栏括弧中的数字表示该资源执行该任务所需时间。在这里各个任务均只需一个资源参与其

执行。为了方便计算和表达清晰, 将各个时间参数均转化为相对时间刻度。资源可用时间为 $tr=[5, 4, 7, 8, -2, 13, 11, 11, 12, 9, 0, 1, 8, 6, -3, -3, 1, 4, 5, 14]$ 。w 和 u 均为 1。

算法采用 matlab 实现。系统在 P4 2.4G/512M RAM 微机上运行。求得的最优解为 $r_i=\{1, 11, 2, 9, 10, 5, 17, 16, 4, 15\}$, 总拖期时间为 43, 总提前时间为 29。其中初始种群大小为 10, 最大迭代次数为 2000, 变异概率为 0.2。图 1 是算法求解迭代过程中各代种群的目标函数最优值的走势曲线。由图 1 可以看出, 该算法具有很好的进化特性, 能够达到或逼近全局最优解。

图 2 比较了采用不同调度方法求解该问题取得的最优值。由于在任务分配方面, 对于本文的目标函数, EET(选取最快可能完成的资源)比其他如最近可用、最短执行时间等方案显然要更优, 因此只与 EET 方法比较。在排程方面, 比较了 FIFO(先来先分配)和 EDD(最早截至期限)两种策略。

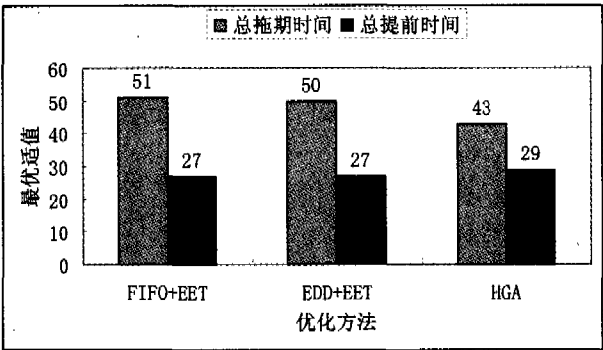


图 2 3 种方法求解最优适应值比较

表 2 也是这些方法求解不同问题的部分算例的比较。所有这些数据表明本文提出的调度方法要优于其它调度方法。在表 2 中同样也可以看到本文方法在求解不同问题时, 具有良好的性能。

总结 调度问题是一个复杂的组合优化问题, 常具有 NP 完全性, 算法复杂性呈指数增长。复杂 workflow 环境下的调度问题就更加复杂了。workflow 调度问题一直是 workflow 领域研究的热点问题。好的调度方法能有效地提高 workflow 管理系统的性能。本文在总结前人工作的基础上, 研究了 workflow 动态调度问题。以任务的拖期时间总和最小, 以及任务的超前时间总和最大为目标, 建立了该问题的数学模型, 并设计了一种启发式规则与遗传算法结合的求解算法, 并通过仿真实验说明了本文方法的有效性和优越性。本文中的启发式算法对于其它领域的调度优化问题也有着相当重要的启发意义和参考价值。

表 2 不同方法求解不同问题部分算例比较

方法 问题	FIFO+EET		EDD+EET		HGA($np=10, ng=2000, p_m=0.2$)			
	拖期时间	提前时间	拖期时间	提前时间	拖期时间	提前时间	达优率	执行时间(s)
1	1	14	1	14	1	14	100%	6.5663
2	3	7	2	7	2	7	100%	6.3712
3	0	24	0	29	0	30	100%	8.0534
4	40	7	36	3	35	7	100%	9.367
5	16	2	16	2	16	4	95%	10.5794

(下转第 191 页)

进行筛选立即临近 Voronoi 多边形,以 L 为公共边的 Voronoi 多边形 $\{P_2, P_3\}$ 首先排除,没有 L 上的公共点的 Voronoi 多边形 $\{P_7\}$ 排除,余下 $\{P_5, P_8\}$,作为两个求 Voronoi 距离的方向。

(4)将 P_5 和 P_8 作为新的起点,重复步骤(2),(3)直到到达 P_2 的 Voronoi 多边形。两个方向得到的 Voronoi 多边形分别为 $\{P_1, P_5, P_4, P_{16}, P_3, P_2\}$ 和 $\{P_1, P_8, P_9, P_{10}, P_{11}, P_{12}, P_{13}, P_2\}$ 。

(5)计算 Voronoi 距离,按照我们定义的 Voronoi 距离,求得 P_1 到 P_2 两条路径中的较短的路径为 $\{P_1, P_5, P_4, P_{16}, P_3, P_2\}$,Voronoi 距离为 5。

如果起点与障碍物没有公共边,例如求 P_6 到 P_2 的 Voronoi 距离,则需要将 P_6 与 P_2 求连线,作为在未与障碍物相遇之前的搜索方向。按照循障碍物求距离法, P_6 到 P_2 的 Voronoi 距离为 7。

5 基于 Voronoi 距离的空间聚类分析实例

我们将基于 Voronoi 距离的聚类分析方法应用于银行业务点的选址分析。为了验证该方法的作用,我们只考虑区域的人口数量和平均收入水平,以楼栋为生长点,单栋楼中的人口和收入水平为权进行加权 Voronoi 图生成。研究区域中共有楼栋 1393 栋,有河流为障碍物,河流被桥梁分段。



图 6 K-平均值聚类结果

图 6 表示不考虑障碍物的情况下 K-平均值聚类的实验结果,聚类产生 5 个中心,其中第 2 和 5 簇跨越河流。图 7 表示考虑障碍物的情况下,生成有障碍物的加权 Voronoi 图,按照 Voronoi 距离来进行聚类的实验结果,图 6 中的第 2、5 簇被河流分割,形成第 2、7、5、6 四个簇,第 3 簇由于有桥梁存在,所以变化不大。另外,簇中心发生了移动,簇 1 和簇 4 的

中心向左边移动。通过对楼栋数据进行分析,发现第 1、4 簇的楼栋加权 Voronoi 多边形从左至右有由小变大的趋势,由于我们采用的 Voronoi 距离对 Voronoi 多边形大小不加以考虑而导致中心的移动。从实验可以看出,通过 Voronoi 图能够很好解决有实体障碍物的空间数据挖掘中的空间聚类分析。



图 7 Voronoi 图聚类结果

总结 借助 Voronoi 图能够明晰地表达侧向临近关系,将 Voronoi 图应用于空间数据挖掘的聚类分析。通过构建有障碍物的空间 Voronoi 图,使用 Voronoi 距离,采用循障碍物求距离法,能够方便地解决有实体障碍物的空间聚类分析。该方法也可以在空间距离分析中应用。在生成加权 Voronoi 图的方法,以及在 Voronoi 多边形大小差距较大时求 Voronoi 距离仍需要进一步研究。

参考文献

- 1 陈军. Voronoi 动态空间数据模型[M]. 北京:测绘出版社,2002
- 2 Goodchild M F, Yang Shiren. A hierarchical data structure for global information systems[J]. Computer Vision and Geographic Image Processing, 1992, 54(1): 31~44
- 3 Green P J, Sibson R. Computing dirichlet tessellations in the plane[M]. Computer Journal, 1997, 21: 168~173
- 4 Zhao Xuesheng, Chen Jun. QTM-based algorithm for generating Voronoi-diagram of spherical objects[C]. In: Proceeding of the 3-rd ISPRS Workshop on Dynamic and Multi-dimensional GIS, 2001. 391~195
- 5 Robert J R. Delaunay triangulation and Voronoi diagram on the surface of a sphere[J]. ACM Transaction on Mathematical Software, 1997, 23(3): 416~434
- 6 Zhao Renliang, Chen Jun, Li Zhilin. Define and describe k-order spatial adjacency with Voronoi distance[C]. In: Proceeding of Dynamic and Multi-Dimensional GIS, 1999. 77~82
- 7 Gold C M, Yang w. Spatial data management tools based on the Voronoi dynamic data model(VORDLL 1.0) User's Manual[M]. 1994
- 8 陈军. Voronoi 动态空间数据模型[M]. 北京:测绘出版社,2002
- 9 段永强, 曹健, 张申生. workflow 系统中的动态任务调度. 中国机械工程, 2002, 13(3): 233~235, 241
- 10 段永强, 曹健, 张申生. workflow 系统中的动态任务调度. 中国机械工程, 2002, 13(3): 233~235, 241
- 11 Tramontina G B, Wainer J, Ellis C. Applying scheduling techniques to minimize the number of late jobs in workflow systems. In: Proceedings of 2004 ACM Symposium on Applied Computing, Nicosia, Cyprus, 2004. 3. 1396~1403
- 12 Davulcu H, Kifer M, Ramakrishnan C R, et al. Ramakrishnan. Logic based modeling and analysis of workflows. In: Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, 1998. 25~33
- 13 Julia S, de Oliveira F F. A p-time hybrid Petri net model for the scheduling problem of workflow management systems. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, vol. 5. 2004. 4947~4952
- 14 欧阳昱. 基于授权约束的工作流任务指派算法. 计算机工程与应用, 2004, 36: 25~27
- 15 Shen Minxin, Tzeng G-H, Liu D-R. Multi-criteria task assignment in workflow management systems. In: Proceedings of the 36th Hawaii International Conference on System Sciences, 2003. 9
- 16 Lee K M. Adaptive resource scheduling for workflows considering competence and preference. In: Negoita M Gh, et al. eds. KES 2004, LNAI 3214, Springer-Verlag, Berlin, Heidelberg, 2004. 723~730
- 17 van der Aalst W M P, van Hee K. Workflow Management: Models, Methods, and Systems. MIT Press, 2002
- 18 Harper P R, de Senna V, Vieira I T, et al. A genetic algorithm for the project assignment problem. Computers & Operations Research, 2005, 32: 1255~1265

(上接第 160 页)

参考文献

- 1 Senkul P, Toroslu I H. An architecture for workflow scheduling under resource allocation constraints. Information System, 2005, 30: 399~422
- 2 段永强, 曹健, 张申生. workflow 系统中的动态任务调度. 中国机械工程, 2002, 13(3): 233~235, 241
- 3 Tramontina G B, Wainer J, Ellis C. Applying scheduling techniques to minimize the number of late jobs in workflow systems. In: Proceedings of 2004 ACM Symposium on Applied Computing, Nicosia, Cyprus, 2004. 3. 1396~1403
- 4 Davulcu H, Kifer M, Ramakrishnan C R, et al. Ramakrishnan. Logic based modeling and analysis of workflows. In: Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, 1998. 25~33
- 5 Julia S, de Oliveira F F. A p-time hybrid Petri net model for the