软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2015,26(3):574−583 [doi: 10.13328/j.cnki.jos.004767]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

面向实例方面处理的工作流动态调度优化方法*

文一凭[1], 刘建勋[1], 陈志刚[2]

[1](知识处理与网络化制造湖南省普通高校重点实验室(湖南科技大学),湖南 湘潭 411201)

[2](中南大学 信息科学与工程学院,湖南 长沙 410083)

通讯作者: 文一凭, E-mail: ypwen81@gmail.com

摘 要: 针对现实业务过程对实例方面处理的需求,建立面向实例方面处理的工作流动态调度优化模型,并提出了相应的优化方法.该方法利用蚁群优化算法的特点直接构建可行解,利用分组浪费时间与分组浪费费用的概念来设计启发式信息,同时优化最小化活动实例的总停留时间与总执行费用这两个目标函数,最终产生一组满足约束条件的 Pareto 优化调度方案.实验结果说明了算法的有效性.

关键词: 工作流;实例方面处理;动态调度;蚁群优化

中图法分类号: TP311

中文引用格式: 文一凭,刘建勋,陈志刚.面向实例方面处理的工作流动态调度优化方法.软件学报,2015,26(3):574−583. http://www.jos.org.cn/1000-9825/4767.htm

英文引用格式: Wen YP, Liu JX, Chen ZG. Instance aspect handling-oriented scheduling optimization in workflows. Ruan Jian Xue Bao/Journal of Software, 2015,26(3):574−583 (in Chinese). http://www.jos.org.cn/1000-9825/4767.htm

Instance Aspect Handling-Oriented Scheduling Optimization in Workflows

WEN Yi-Ping[1], LIU Jian-Xun[1], CHEN Zhi-Gang[2]

[1](Key Laboratory of Knowledge Processing and Networked Manufacture (Hunan University of Science and Technology), Xiangtan 411201, China)

[2](School of Information Science and Engineering, Central South University, Changsha 410083, China)

Abstract: To meet the needs of instance aspect handling in practical workflow applications, a model for instance aspect handling-oriented optimal scheduling of multiple activity instances is constructed. An algorithm for such scheduling optimization is presented correspondingly. It utlizes the theory of ant colony optimization to achieving the objectives of minimum acitity instances' total dwelling time and minimum acitity instances' total cost with constraints. The conception of wasted grouping time and wasted grouping cost are introduced according to the two optimization objectives, based on which the heuristic information for the ants are designed. The result of simulation experiment shows its effectiveness.

Key words: workflow; instance aspect handling; dynamic scheduling; ant colony optimization

工作流调度一直是工作流管理中的核心问题之一,合理的调度方案不仅可以提高工作流的执行效率,而且能够降低服务资源的使用成本.与一般的调度不同,面向实例方面处理的工作流调度需要在某个阶段,将来自于不同工作流实例的同一类型活动的多个并发实例进行合适的分组合并后,再将其分配给合适的执行者(人力资源或非人力资源)执行.这种实例方面处理可直接减少分配给执行者的任务数目,并有助于降低活动实例的执行成本,提高执行者的工作效率.但是,传统工作流所关注的是描述活动间控制约束关系的执行顺序等过程语义,是一种静态水平的约束描述,并没有描述与处理同一过程多个实例之间的动态垂直方面的关联或约束关系.因此,这种调度方式对传统工作流执行过程的控制机制提出了新的要求,现有工作流调度算法也大多无法根据需

* 基金项目: 国家自然科学基金(61272063, 61402167)
收稿时间: 2014-06-27; 修改时间: 2014-09-30; 定稿时间: 2014-11-21

© 中国科学院软件研究所 http://www.jos.org.cn

要对一组并发实例进行实例方面优化处理.

我们的前期工作已对支持实例方面处理的工作流调度控制方法展开了研究[2],该方法通过过程控制器、活动实例管理器 MShell、执行者资源管理器等组件及相关控制算法来控制实例方面处理过程.在此基础上,本文进一步研究如何将 Mshell 输入活动实例队列中的多个活动实例进行分组,以及如何为这些活动实例分组指派执行者并安排执行顺序的问题.从本质上说,该问题仍然是一个工作流调度问题.但不同的是,它不仅包含任务分配和任务排程这两个基本问题,还涉及如何将活动实例进行分组的决策问题.尽管分组决策会增加调度的难度,但并不影响其可调度性:

- 首先,分组决策问题本质上是一个组合优化问题,可借鉴现有处理组合优化问题的方法来求解;
- 其次,可以采用简化的方式来产生活动实例分组调度方案,例如,任务分配可先按照活动实例的到达顺序选取活动实例并产生活动实例分组,再将该活动实例分组随机分配给可用的执行者;任务排程则按照活动实例分组的产生次序依次进行处理.这种简化方式容易产生调度方案,但由于工作流运行时的不确定性,其结果并不一定是最优的.

蚁群优化(ant colony optimization,简称 ACO)等元启发式算法已被证明能够很好地解决不确定性调度问题.同时,与微粒群优化(particle swarm optimization,简称 PSO)等算法相比,ACO 算法可利用信息素与启发式信息来直接构建活动实例分组调度问题的可行解,从而更好地发挥算法的优化能力.因此,本文基于 ACO 算法提出了一种面向实例方面处理的工作流调度优化方法(简称 PACO-TC 算法),通过以最小化活动实例总停留时间及最小化活动实例总执行费用为优化目标,利用 ACO 算法的特点直接构建可行解,并针对优化目标的特点,提出了分组浪费时间与分组浪费费用的概念来设计启发式信息,最终产生一组满足约束条件的 Pareto 优化调度方案.

本文第 1 节描述面向实例方面处理的工作流调度优化问题.第 2 节描述其相应的优化模型.第 3 节介绍 PACO-TC 算法.第 4 节描述实验结果.第 5 节介绍相关研究.第 6 节进行总结.

1 问题描述

1.1 控制机制描述

为实现在工作流系统运行时对多个活动实例的动态实例方面处理,可以通过对传统工作流执行服务的执行机制进行适当的扩展来实现.另外,活动实例被创建后,必须控制其活动实例的分组合并过程,并控制何时触发实例方面处理区中的后继活动.为了做到这两点,首先需要在工作流过程定义时能够明确无误的将相关信息传递给工作流引擎,例如哪些活动可进行实例方面处理、根据哪些数据将活动实例进行实例方面处理等.此外,还需要一种控制机制来控制处理过程.定义 1 描述了支持实例方面处理的工作流模型:

定义 1. 支持实例方面处理的工作流是一类包含可进行实例方面处理的工作流活动的特殊工作流过程,可描述为一个五元组$\langle Activities,Transitions,BPAs,Rdata,DOMs\rangle$,其中,

(1) *Activities* 是一组工作流活动的集合;

(2) *Transitions* 为一组转换条件的集合,它描述了活动之间的连接关系与执行的先后顺序;

(3) *BPAs* 是一组实例方面处理区的集合.实例方面处理区由一个或多个可实例方面处理的工作流活动所组成;

(4) *Rdata* 是工作流相关数据的集合,它包括活动的输入/输出数据等;

(5) *DOMs* 是与可实例方面处理的工作流活动相对应的数据操作模型集.

图 1 描述了相应控制机制的基本结构,实例方面调度控制机制主要包含过程控制器、BPA 控制器、活动实例管理器、执行者资源管理器等组件,其中,

(1) 过程控制器主要负责根据业务过程模型定义及过程实例执行的需要产生或销毁 BPA 控制器;

(2) BPA 控制器与实例方面处理区相对应,负责产生或销毁活动实例管理器;

(3) 活动实例管理器与可实例方面处理活动相对应,具体完成对活动实例分组合并执行过程的控制.它主要由输入活动实例队列集、输出活动实例队列、分组调度控制组件及数据操作管理组件构成,由

© 中国科学院软件研究所 http://www.jos.org.cn

分组调度控制组件负责对输入活动实例队列中的活动实例进行分组调度,并调用数据操作管理组件完成相关数据操作,分组调度所产生的新活动实例将依次存放到输出活动实例队列中;

(4) 执行者资源管理器管理与维护和活动实例执行相关的各类执行者的数目与状态等信息;

(5) 相关控制数据主要是实例方面调度过程中所涉及的控制数据.

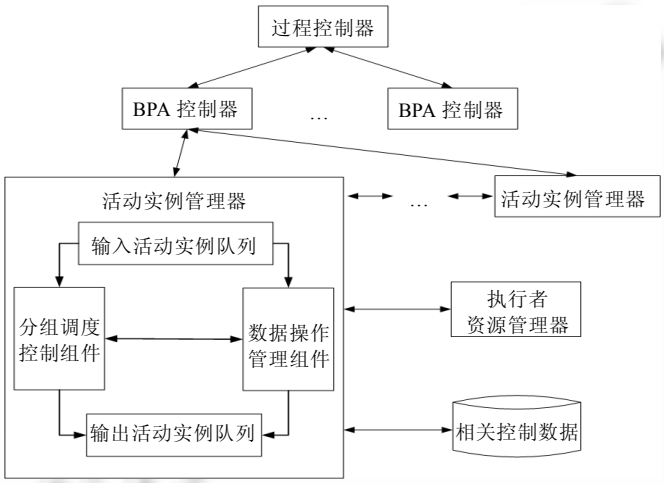


Fig.1 Basic structure of instance aspect handling control mechanism

图 1 实例方面调度控制机制基本结构

1.2 问题描述

本文研究的面向实例方面处理的工作流调度优化问题可描述如下:活动实例管理器的某输入活动实例队列中存在活动实例 $i(i=1,2,\ldots,M)$,M 为活动实例总数;系统中当前有执行者 k 可为其服务($k=1,2,\ldots,N$),N 为执行者总数.需要将这些活动实例进行分组,并为这些活动实例分组指派执行者及安排执行者上各个分组的执行顺序,从而达到一定的调度性能指标.

本文研究的执行者是实质参与活动实例执行的实体,它可以是人力资源,也可以是非人力资源,如生产设备、应用程序等.一般而言,活动实例在执行时可能需要一个或多个执行者,而不同的执行者可能具有不同的执行能力,并且在完成不同难度的任务时所需要的执行时间也可能不一样.另外,工作流系统的目标就是尽可能快捷的任务,工作流实例方面处理的目标也在于提高处理效率、节省时间与执行费用.在时间方面,工作流实例的停留时间是一个重要优化指标[2].如果活动实例在实例方面处理过程中停留时间过长,将直接影响工作流实例的停留时间,进而影响工作流性能及用户对业务过程处理的满意度.

因此,本文主要考虑以下两个优化目标:

1) 最小化各活动实例停留时间的总和,其中,活动实例停留时间是指活动实例从其到达输入活动实例队列时刻到其执行结束时刻之间的时间间隔;

2) 最小化各活动实例执行费用的总和,其中,活动实例执行费用是指执行者在执行该活动实例时所耗费的费用.

2 面向实例方面处理的工作流调度优化模型

2.1 基本假设

在工作流管理系统中,工作流活动或任务只有在相应资源的数量、种类、能力得到保证的前提下才能被执行.此外,资源执行任务的时间与任务的要求、资源能力与偏好等因素有关.任务的要求描述了任务的工作量及

© 中国科学院软件研究所 http://www.jos.org.cn

要求完成的质量方面的信息,资源能力描述了资源具有的技能及技能水平等信息,资源偏好描述了资源在选择任务执行时的倾向[3].由于实例方面处理的特殊性,执行者可视为一种特殊的具有最大任务量约束的资源;同时,为简化问题讨论,可暂不考虑执行者的偏好等因素对执行时间的影响.因此,本文做出如下假设:

(1) 每个活动实例 i 具有不同的任务量 w_i 及执行难度 d_i;活动实例 i 从到达输入活动实例队列到当前时刻已耗费的等待时间为 t_i.其中,w_i 描述了任务的工作量大小,d_i 描述了任务要求完成的质量需求;

(2) 每个执行者 k 具有不同的执行能力 ew_k;bt_k 为执行者 k 从当前时刻到其可用时刻之间的时间间隔大小,若执行者 k 当前为空闲状态,则 bt_k=0.其中,执行者的可用时刻是指该执行者可参与执行本次活动实例分组调度所分配的活动实例分组的时刻;

(3) 活动实例分组 j 的执行难度 gd_j 取决于 j 中活动实例的最大执行难度;任务量 gw_j 为活动实例分组 j 中各活动实例的任务量总和;

(4) 各活动实例分组的任务量不能超过最大任务量 W.其中,W 体现了执行者的最大任务量约束;

(5) 各执行者在其可用时刻到达后可无中断的执行.若活动实例分组 j 在执行者 k 上的执行顺序为 q,则活动实例分组 j 中各活动实例从执行者 k 可用到其执行结束所需要的时间 wt_j^k 可描述如下:

$$wt_j^k = bt_k + \sum_{q=1}^{m} et_{kq} \tag{1}$$

其中,et_{kq} 为执行者 k 上执行顺序为 q 的活动实例分组在被执行时所需要的执行时间;

(6) 活动实例分组的执行时间与任务量的大小、执行难度成正比,与执行者的执行能力成反比.即执行者 k 执行活动实例分组 j 所需要的执行时间为 $gd_j \cdot gw_j / ew_k$;

(7) 执行者 k 执行活动实例分组 j 所耗费的执行费用为 $ew_k \cdot f(gw_j)$.其中,$f(gw_j)$是活动实例分组任务量 gw_j 的函数,它反映了任务量对执行费用的影响.一般而言,gw_j 值越大,单位任务量的执行费用 $f(gw_j)/gw_j$ 相对越小.

2.2 优化模型

基于上面的假设,设活动实例停留时间总和为 f_1,活动实例执行费用总和为 f_2,可建立优化模型如下:

$$F=\min(f_1,f_2) \tag{2}$$

$$\text{s.t. } f_1 = \sum_{i=1}^{M}\left(\sum_{j=1}^{M}\sum_{k=1}^{N}\sum_{m=1}^{M} E_{ijkm} \cdot \left(bt_k + \sum_{q=1}^{m} et_{kq}\right) + t_i\right) \tag{3}$$

$$f_2 = \sum_{j=1}^{M}\sum_{k=1}^{N}\sum_{m=1}^{M}\left(\frac{\sum_{i=1}^{M} E_{ijkm} \cdot ew_k \cdot f(gw_j)}{\max\left\{1, \sum_{i=1}^{M} E_{ijkm}\right\}}\right) \tag{4}$$

$$et_{kq} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{M} E_{ijkq} \cdot gd_j \cdot gw_j \Big/ ew_k}{\max\left\{1, \sum_{i=1}^{M}\sum_{j=1}^{M} E_{ijkq}\right\}}, 1 \leqslant k \leqslant N; 1 \leqslant q \leqslant M \tag{5}$$

$$gd_j = \max\left\{\sum_{i=1}^{M}\sum_{k=1}^{N}\sum_{m=1}^{M} E_{ijkm} \cdot d_i\right\}, 1 \leqslant j \leqslant M \tag{6}$$

$$gw_j = \sum_{i=1}^{M}\sum_{k=1}^{N}\sum_{m=1}^{M} E_{ijkm} \cdot w_i, 1 \leqslant j \leqslant M \tag{7}$$

$$\sum_{i=1}^{M} w_i \cdot E_{ijkm} \leqslant W, 1 \leqslant k \leqslant N; 1 \leqslant i \leqslant M \tag{8}$$

$$\sum_{j=1}^{M}\sum_{k=1}^{N}\sum_{m=1}^{M} E_{ijkm} = 1, 1 \leqslant i \leqslant M \tag{9}$$

© 中国科学院软件研究所 http://www.jos.org.cn

$$E_{ijkm} \in \{0,1\}, 1 \leqslant i,m \leqslant M; 1 \leqslant k \leqslant N \tag{10}$$

其中,

- 公式(2)描述了问题的目标函数;
- 公式(3)描述了活动实例的停留时间总和;
- 公式(4)描述了执行者的执行开销总和;
- 公式(5)描述了执行者 k 上执行顺序为 q 的活动实例分组在被执行时所需要的执行时间;
- 公式(6)描述了活动实例分组 j 的执行难度;
- 公式(7)描述了活动实例分组 j 的任务量大小;
- 公式(8)用于约束各活动实例分组的任务量不能超过最大任务量 W;
- 公式(9)用于约束各活动实例均属于某个分组且只能被某一个执行者执行;
- 公式(10)描述了活动实例分组与执行者分配情况:若 $E_{ijkm}=1$,则表示第 i 个活动实例被分配在活动实例分组 j,该分组将由执行者 e_k 执行,且执行序号为 m;反之,$E_{ijkm}=0$.

3 PACO-TC 算法描述

蚁群优化是一种基于群体智能的构建性元启发算法,其中,PACO(pareto-based ant colony optimization)算法是由 Doerner 等人提出的一种多目标蚁群优化算法[4],它易于实现且具有较好的优化效果.因此,本文针对上述优化模型,基于 PACO 算法提出面向实例方面处理的工作流调度时间费用优化方法——PACO-TC 算法.

3.1 可行解的构造

PACO-TC 算法以活动实例作为蚂蚁的起点开始搜索,采用概率选择的方法构建解.首先,将各活动实例随机分配给不同的蚂蚁,蚂蚁将当前最早可用的执行者作为当前执行者,并根据所分配的活动实例在执行者上构建一个分组作为当前分组;然后,蚂蚁不断按照状态转移概率从候选列表中选择一个活动实例放入当前分组,并更新候选列表,如果候选列表为空,蚂蚁重新选择当前最早可用的执行者作为当前执行者,并先构建一个空的分组作为当前分组,再随机选择一个未被调度的活动实例加入当前分组.重复此过程,直到活动实例全部被调度.

蚂蚁从候选列表中选择活动实例 j 的操作依据伪随机比例规则进行,具体选择方法可描述如下:

$$j=\begin{cases}\arg\max\limits_{j\in\Omega_u}\left\{\left[\sum\limits_{k=1}^{2}(p_k\cdot\tau_{u,j}^k)\right]^{\alpha}\cdot[\eta_{u,j}]^{\beta}\right\}, & q\leqslant q_0, \sum\limits_{k=1}^{2}p_k=1\\ j', & q>q_0\end{cases} \tag{11}$$

$$P_{u,j}=\begin{cases}\dfrac{\left[\sum\limits_{k=1}^{2}(p_k\cdot\tau_{u,j}^k)\right]^{\alpha}\cdot[\eta_{u,j}]^{\beta}}{\sum\limits_{h\in\Omega_u}\left[\sum\limits_{k=1}^{2}(p_k\cdot\tau_{u,h}^k)\right]^{\alpha}\cdot[\eta_{u,h}]^{\beta}}, & j\in\Omega_u, \sum\limits_{k=1}^{2}p_k=1\\ 0, & j\notin\Omega_u\end{cases} \tag{12}$$

$$\tau_{u,j}^k=\frac{\sum\limits_{i\in G_u}\tau_{i,j}^k}{|G_u|}, j\in\Omega_u, k=1,2 \tag{13}$$

其中,p_k 是不同目标的信息素权重值,$0\leqslant p_k\leqslant 1$;$\alpha$与$\beta$这两个参数分别体现了信息素与启发式信息在蚂蚁选择路径中的相对重要性;$|G_u|$表示当前分组 G_u 中的活动实例数目;q 是在[0,1)区间上均匀分布的一个随机数,q_0($0\leqslant q_0<1$)是预先定义好的一个常数.若 $q\leqslant q_0$,则蚂蚁直接按照公式(11)选择信息素与启发式信息综合值最大的活动实例 j;否则,按照公式(12)的选择概率选取活动实例 j'.由于活动实例分组是动态构建的,分组 G_u 中所包含的元素不断变化,按照公式(13)来间接利用信息素 $\tau_{i,j}^k$,从而间接得到分组 G_u 与活动实例关于第 k 个目标的信息素值 $\tau_{i,j}^k$,该值描述了对于第 k 个优化目标,活动实例 i 与 j 在同一分组中的期望.

© 中国科学院软件研究所 http://www.jos.org.cn

3.2 启发式信息的构造

对于活动实例分组而言,一个好的分组方案应尽量使执行难度接近的活动实例在同一分组中,并尽量使分组任务量接近最大任务量.也就是说:如果分组中各活动实例执行难度差异太大,会造成时间上的浪费;分组任务量与最大任务量差异太大,会造成执行费用上的浪费.为此,本文提出分组浪费时间和分组浪费费用的概念,具体定义如下:

定义 2. 分组浪费时间是指由于分组中活动实例执行难度间的差异所造成的时间上的浪费.假设活动实例 i 的执行难度为 d_i,当前分组为活动实例集 G_u,gd_u 为其执行难度(即,G_u 中各活动实例的最大执行难度),其分组浪费时间可计算如下:

$$GWT_u = \sum_{i \in G_u} w_i \cdot (gd_u - d_i) \tag{14}$$

定义 3. 分组浪费费用是指由于活动实例分组的任务量没有达到最大任务量所造成的执行费用的浪费.假设当前分组为活动实例集 G_u,W 为最大任务量,活动实例 j 的任务量为 w_j,其分组浪费费用的计算公式如下:

$$GWC_u = W - \sum_{i \in G_u} w_i \tag{15}$$

因此,在从候选列表中选择活动实例时,应引导蚂蚁向能降低当前分组浪费费用与当前分组浪费空间的方向移动.假设 $G_{u'}$为当前分组 G_u 加入活动实例 $j(j \in \Omega_u)$后所得的活动实例集,综合这两方面的启发式信息 $\eta_{u,j}$ 可计算如下:

$$\eta_{u,j} = \begin{cases} \Delta t_{u,j} + \Delta c_{u,j} + 1, & \Delta t_{u,j} + \Delta c_{u,j} \geqslant 0 \\ 1, & \Delta t_{u,j} + \Delta c_{u,j} < 0 \end{cases} \tag{16}$$

$$\Delta t_{u,j} = GWT_u - GWT_{u'} \tag{17}$$

$$\Delta c_{u,j} = GWC_u - GWC_{u'} \tag{18}$$

3.3 储备集的更新

当活动实例全部被调度后,蚂蚁便完成了一次搜索并得到了一个可行解.在第 t 次迭代时,当所有蚂蚁都完成一次搜索后,假设这些可行解构成的一个解集合为 $S_{pop}(t)$.为保证最优解的多样性,需要根据 $S_{pop}(t)$中的非支配解对储备集 Ar_s 进行更新,具体步骤如下:

(1) 求 $S_{pop}(t)$中的非支配解,找出当前蚁群 $S_{pop}(t)$的非支配解集 $S_{NDS}(t)$;

(2) 将 $S_{NDS}(t)$加入储备集 Ar_s,删除其中的被支配解,完成对 Ar_s 的更新;

(3) 为避免解的数量过多导致的效率问题,设储备集 Ar_s 的规模为 N_s,当 Ar_s 中的个体数量大于 N_s,则随机选出 N_s 个非支配解保留在 Ar_s 中,其余个体全部删除.

3.4 信息素的更新

为增加蚂蚁探索其他路径的可能性,PACO-TC 算法在蚂蚁每选择一个活动实例 j 加入到当前分组 G_u 后,便更新与当前分组相关的信息素值.对蚂蚁所走过的路径上信息素浓度进行一次局部更新,以减小该路径对后来蚂蚁的吸引力.信息素的局部更新公式如下:

$$\tau_{i,j}^k = (1-\rho) \cdot \tau_{i,j}^k + \rho \cdot \tau_0, i \in G_u; k = 1,2 \tag{19}$$

其中,$\rho(0<\rho\leqslant 1)$为局部信息素挥发率,$(1-\rho) \cdot \tau_{i,j}^k$ 代表原有信息素的挥发;τ_0 为初始信息素.

另外,为引导蚂蚁向最优解的方向搜索,在所有蚂蚁完成一次搜索后,以当前储备集 Ar_s 中的元素在各目标函数的最优解与次优解作为全局最优经验,对各信息素进行全局更新.信息素的全局更新公式如下:

$$\tau_{i,j}^k = (1-\gamma) \cdot \tau_{i,j}^k + \gamma \cdot \Delta\tau_{i,j}^k, k = 1,2 \tag{20}$$

$$\Delta\tau_{i,j}^k = \begin{cases} \dfrac{q_k}{f_k(S)}, & S \in S_{best,best'}^k(t) \\ 0, & S \notin S_{best,best'}^k(t) \end{cases}, k = 1,2 \tag{21}$$

© 中国科学院软件研究所 http://www.jos.org.cn

其中:γ为全局信息素挥发率,$(1-\gamma)\cdot\tau_{i,j}^{k}$ 代表原有信息素的挥发;q_k 为调整参数;$f_k(S)$为可行解 S 在优化目标 k 上的目标函数值;$S_{best,best'}^{k}(t)$ 是由当前储备集 Ar_s 在第 k 个优化目标上的最优解与次优解(second-best solution)所组成的集合.

3.5 算法主要步骤

算法 1. PACO-TC.

输入:蚂蚁群体规模 N_a,储备集规模 N_s,最大迭代次数 $T_{\max}$,待调度的活动实例数目 M,执行者数目 N;

输出:与储备集 Ar_s 对应的最优调度方案集.

(1) 初始化各蚂蚁的信息素等信息,设置相关参数;

(2) 各蚂蚁按可行解构造策略开始搜索;

(3) 所有蚂蚁完成一次搜索后,转步骤(5);

(4) 各蚂蚁按信息素及启发式信息的指引,从候选列表中选择活动实例,然后按公式(19)更新局部信息素;

(5) 评价各蚂蚁所构建的解的质量,并按储备集更新策略维护 Ar_s,

(6) 按公式(20)更新全局信息素;

(7) 若达到终止条件,则输出与 Ar_s 对应的最优调度方案集;否则,转步骤(2).

4 实验结果

4.1 实验设计

某公司网上定制建筑材料颜色的业务过程可用图 2 进行描述,主要包括接收订单、喷漆、送货、收款等一系列处理步骤.由于单个喷漆设备可同时为多箱同一颜色的建筑材料喷漆,而在喷不同颜色时要更换部分组件,为节省时间和资源,在喷漆时需要将不同客户的订单按建筑材料的颜色组合在一起进行加工处理.因此,这是一个典型的需要在业务过程某些阶段进行实例方面处理的业务过程.

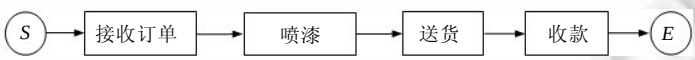


Fig.2 Process for customizing colors of construction material online

图 2 某公司网上定制建筑材料颜色的业务过程

为了测试算法的有效性,我们对图 2 中的业务过程进行了仿真.仿真时,由于仅需对加工同种颜色建筑材料的喷漆活动实例进行调度,可不考虑建筑材料的颜色差异.具体仿真参数设定如下:

(1) 订单的到达服从参数为λ的泊松分布;

(2) 喷漆活动实例的任务量为订单中建筑材料的箱数,是 1~10 之间随机产生的整数;执行难度取决于订单中建筑材料的型号,为 1~3 之间随机产生的整数;

(3) 喷漆设备(执行者)一次最多可同时喷 30 箱材料,即喷漆活动实例分组的最大任务量为 30;

(4) 喷漆设备的加工能力(执行能力)为[5,6]之间随机产生的随机数,且各喷漆设备均为空闲状态.

(5) 喷漆设备 k 对 x 箱材料进行加工的费用(执行费用)为 $ew_k\cdot\omega\cdot x$,其中,ew_k 为喷漆设备 k 的加工能力;ω 的取值与 x 的大小有关,可用下式描述:

$$\omega=\begin{cases}0.4, & 1\leqslant\omega<10\\ 0.35, & 10\leqslant\omega<20\\ 0.3, & 20\leqslant\omega\leqslant 30\end{cases}\tag{22}$$

同时,为了比较不同算法间的性能,实验中采用了如下评价指标:

(1) 超体积(hypervolume)指标[5]

该指标是目前流行的一种解集综合性能评价方法,解集越接近真实的 Pareto 前端,且拥有相对均匀并且广

© 中国科学院软件研究所 http://www.jos.org.cn

泛的分布时,该解集的超体积值相对越大.

在计算解集的超体积指标时,必须选择一个参考点 X^*,该点必须被解集中的所有非支配解所支配.设 $X^*=(r_1, r_2,\ldots,r_k)$,则解集 S 的超体积值为 S 中的所有解与 X^*在目标空间中所围成的超立方体的体积,可描述如下:

$$H(S)=L\left(\bigcup_{X\in S}[f_1(X),r_1]\times[f_2(X),r_2]\times\ldots\times[f_k(X),r_k]\right) \tag{23}$$

在实验时,我们按照文献[6]的方法选取参考点,具体描述如下:

$$r_1=\max_T+\sigma(\max_T-\min_T) \tag{24}$$

$$r_2=\max_C+\sigma(\max_C-\min_C) \tag{25}$$

其中,$\max_T$ 与 $\min_T$ 分别表示测试算法所得到的最大和最小的活动实例停留时间总和;$\max_C$ 与 $\min_C$ 分别表示最大和最小的活动实例执行开销总和;σ为大于 0 的参数,实验时,令 σ=0.01.

(2) 运行时间

算法在种群规模和最大迭代次数相同时运行一次所需要的时间.

4.2 PACO-TC算法的性能测试

本文选取 2 个多目标微粒群优化算法与 PACO-TC 算法进行性能比较.这些算法分别是基于 *Sigma* 值的多目标微粒群优化算法(MOPSO based on the Sigma method,简称 SMOPSO)[7]、基于时变惯性权重和学习因子的多目标微粒群优化算法(MOPSO with time variant inertia and acceleration coefficients,简称 TV-MOPSO)[8].实验中,算法采用 Matlab 7.8 编程,微机配有 Intel Core CPU(3.2GHz),2GB 内存和 Windows 7 操作系统.

PACO-TC 算法具体设置如下:选择概率 q_0=0.4,α=1,β=3,局部信息素挥发率ρ=0.1,全局信息素挥发率γ=0.2,初始信息素 τ_0=1.为了便于比较,各算法的种群规模(设为 100)和最大迭代次数(设为 200)均相等,SMOPSO,TV-MOPSO 算法的其他参数设置按照原文作者进行设置.

考虑到算法的随机性,实验时采用 30 次实验所获得的平均覆盖率及平均超体积值进行比较.通过将不同实例规模的测试数据分别进行实验,PACO-TC 与 SMOPSO,TV-MOPSO 算法的平均超体积值及在迭代 200 次时平均运行时间比较结果如图 3 与图 4 所示.其中,实例规模表示调度时喷漆活动实例的数目,执行者数目均为 4.

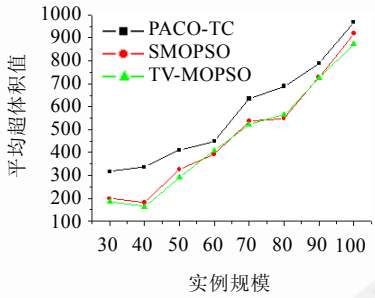


Fig.3 Average hypervolume of PACO-TC, SMOPSO and TV-MOPSO

图 3 PACO-TC 与 SMOPSO,TV-MOPSO 算法的平均超体积值

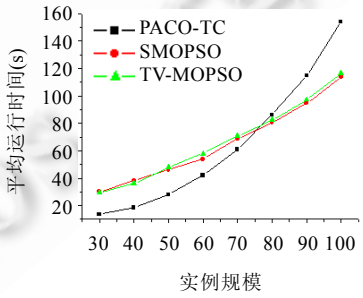


Fig.4 Average execution time of PACO-TC, SMOPSO and TV-MOPSO

图 4 PACO-TC 与 SMOPSO,TV-MOPSO 算法的平均运行时间

从图 3 可知:从总体来说,PACO-TC 算法在不同实例规模下所获得的解的质量均优于 SMOPSO,TV-MOPSO 算法;但从图 4 可知:由于 PACO-TC 算法直接构造可行解且包含了许多启发式信息,当实例规模增加时,PACO-TC 算法的时间增加幅度较 SMOPSO,TV-MOPSO 算法更加明显.

5 相关研究

在现代制造、电子商务、电子政务等工作流实际应用中,存在着许多需要在某个阶段对过程或活动实例进

© 中国科学院软件研究所 http://www.jos.org.cn

行实例方面处理的情形.但传统工作流没有描述与处理同一过程多个实例之间动态垂直方面的关联或约束关系,从而导致无法根据需要对一组并发实例进行实例方面优化处理.

针对这一不足,我们已对支持实例方面处理的工作流调度与控制机制展开了一系列研究[1,9,10].目前,这些研究已引起相关学者的关注与兴趣.例如,对于工作流成批处理这一实例方面处理的特殊情形,文献[11]针对现实应用中工作流过程的多个实例需要同步执行,以提高处理效率等需求.在我们的基础上,进一步研究了成批活动的过程建模与执行方法.文献[12]借鉴了我们将待处理队列中多个活动实例进行分组批处理的思想,并针对如何在资源受限情况下减少过程实例的执行时间问题,提出了一种根据实例间的相似特性将实例进行动态分组并分配合适数目的资源的方法,以减少因待处理的实例过多而产生的延时.

一些学者也针对多个工作流过程实例之间的共性关系开展了一些研究工作.例如:文献[13]针对工作流管理系统中多个并行执行的工作流之间可能存在的资源方面特性,通过分析并行工作流中各活动间的时间关系和资源约束,提出了一种对并行工作流进行时间约束动态校验的方法;文献[14]提出了将面向方面思想应用于工作流的数据校验与安全,以提高工作流的模块性,并支持服务合成;文献[15]针对过程中活动间不同活动语义的上下文,对活动多实例的活动属性进行了统一的形式化描述,提出了活动多实例控制体 Shell,用于控制活动多实例的分配和提交;文献[16]提出了一种基于染色 Petri 网理论的主从流程实例分解、同步和协调的解决方案,以实现对工作流系统中的流程实例进行并发处理和共享处理环节.

6 结束语

本文在前期工作的基础上进一步研究了面向实例方面处理的工作流调度优化问题,建立了相关优化模型,并基于蚁群优化原理提出了相应的调度优化算法.该算法有助于提高支持实例方面处理的工作流调度过程的有效性,并降低或节约活动实例的总停留时间与总执行费用.

References:

[1] Wen YP, Chen ZG, Liu JX. Dynamic workflow scheduling approach supporting instance aspect handling. Computer Integrated Manufacturing Systems, 2013,17(8):1842−1848 (in Chinese with English abstract).

[2] Liu S, Fan YS, Lin HP. Dwelling time probability density distribution of instances in a workflow model. Computers & Industrial Engineering, 2009,57(3):874−879. [doi: 10.1016/j.cie.2009.03.002]

[3] Russell N, van der Aalst WMP, ter Hofstede AHM, Edmond D. Workflow resource patterns: Identification, representation and tool support. In: Pastor O, *et al.*, eds. Proc. of the CAiSE 2005. LNCS 3520, Berlin: Springer-Verlag, 2005. 216−232. [doi: 10.1007/11431855_16]

[4] Doerner K, Gutjahr WJ, Hartl RF, Strauss C, stummer C. Pareto ant colony optimization: A metaheuristic approach to multiobjective portfolio selection. Annals of Operations Research, 2004,131:79−99. [doi: 10.1023/B:ANOR.0000039513.99038.c6]

[5] Zitzler E, Thiele L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. IEEE Trans. on Evolutionary Computation, 1999,3(4):257−271. [doi: 10.1109/4235.797969]

[6] Knowles J. ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. IEEE Trans. on Evolutionary Computation, 2005,10(1):50−66. [doi: 10.1109/TEVC.2005.851274]

[7] Mostaghim S, Teich J. Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO). In: Proc. of the 2003 IEEE Swarm Intelligence Symp. IEEE, 2003. 26−33. [doi: 10.1109/SIS.2003.1202243]

[8] Praveen KT, Sanghamitra B, Sankar KP. Multi-Objective particle swarm optimization with time variant inertia and acceleration coefficients. Information Sciences, 2007,177(22):5033−5049. [doi: 10.1016/j.ins.2007.06.018]

[9] Liu JX, Hu JM. Dynamic batch processing in workflows: Model and implementation. Future Generation Computer Systems, 2007, 23(3):338−347. [doi: 10.1016/j.future.2006.06.003]

[10] Liu JX, Wen YP, Li T, Zhang XY. A data-operation model based on partial vector space for batch processing in workflow. Concurrency and Computation: Practice and Experience, 2011,17(8):1633−1639.

© 中国科学院软件研究所 http://www.jos.org.cn

[11] Pufahl L, Weske M. Batch activities in process modeling and execution. In: Basu S, *et al*., eds. Proc. of the 11th Int'l Conf. on Service Oriented Computing (ICSOC 2013). Berlin: Springer-Verlag, 2013. 283−297. [doi: 10.1007/978-3-642-45005-1_20]

[12] Pflug J, Rinderle-Ma S. Dynamic instance queuing in process-aware information systems. In: Proc. of the 28th Annual ACM Symp. on Applied Computing (SAC 2013). New York: ACM Press, 2013. 1426−1433. [doi: 10.1145/2480362.2480629]

[13] Li HC, Yang Y. Dynamic checking of temporal constraints for concurrent workflows. Electronic Commerce and Application, 2005, 4:124−142. [doi: 10.1016/j.elerap.2004.09.003]

[14] Charfi A, Mezini M. Aspect-Oriented workflow languages. In: Meersman R, Tari Z, eds. Proc. of the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE. Berlin: Springer-Verlag, 2006. 183−200. [doi: 10.1007/11914853_12]

[15] Sun RZ, Shi ML. Schedule of activity instances in workflow management system. Ruan Jian Xue Bao/Journal of Software, 2005, 16(3):400−406 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/16/400.htm

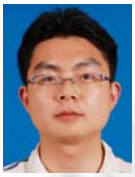
[16] Lu HH, Min LJ, Wang YS. Approach to master-slave workflow system and its Petri-net modeling. Journal on Communications, 2010,31(1):92−99 (in Chinese with English abstract).

附中文参考文献:

[1] 文一凭,陈志刚,刘建勋.支持实例方面处理的工作流调度控制方法.计算机集成制造系统,2013,17(8):1842−1848.

[15] 孙瑞志,史美林.工作流活动多实例的调度控制.软件学报,2005,16(3):400−406. http://www.jos.org.cn/1000-9825/16/400.htm

[16] 卢捍华,闵丽娟,王亚石.工作流主从实例处理方法及其 Petri 网建模.通信学报,2010,31(1):92−99.



文一凭(1981－),男,湖南祁阳人,博士,讲师,CCF 会员,主要研究领域为云计算,工作流与业务过程管理,大数据.

刘建勋(1970－),男,博士,教授,博士生导师,CCF 会员,主要研究领域为服务计算,工作流技术与 BPM.

陈志刚(1964－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为软件工程,智能计算与分布式处理.

© 中国科学院软件研究所 http://www.jos.org.cn