软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2015,26(3):562−573 [doi: 10.13328/j.cnki.jos.004766]
©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

基于社会关系的工作流任务分派策略研究*

余 阳[1], 王 颖[2], 刘醒梅[2], 陈 健[2]

[1](中山大学 软件学院,广东 广州 510006)

[2](中山大学 信息科学与技术学院,广东 广州 510006)

通讯作者: 余阳, E-mail: yuy@mail.sysu.edu.cn, http://ss.sysu.edu.cn/~yy/

摘 要: 在工作流管理系统中,任务分派策略对工作流系统的性能影响较大,而人力资源社会属性的不稳定也给任务分派带来了挑战.一般的任务分派策略还存在以下问题:分派时只考虑候选资源的个体属性,忽略了流程中其他资源对候选资源的影响;需要为候选资源预先设置能力指标,但预设指标很难与候选资源的实际情况吻合,错误的能力指标会导致将任务分派给不合适的资源,降低工作流系统的性能.为克服上述问题,基于不同的状态转移视角和奖励函数,提出了4种基于Q学习的任务分派算法.通过对比实验,论证了基于Q学习的任务分派算法在未预设资源能力的情况下仍能取得较好效果,且支持在任务分派过程中考虑社会关系的影响,使得平均案例完成时间进一步降低.

关键词: 工作流;任务分派;社会关系;Q学习

中图法分类号: TP311

中文引用格式: 余阳,王颖,刘醒梅,陈健.基于社会关系的工作流任务分派策略研究.软件学报,2015,26(3):562−573. http://www.jos.org.cn/1000-9825/4766.htm

英文引用格式: Yu Y, Wang Y, Liu XM, Chen J. Workflow task assignment strategy based on social context. Ruan Jian Xue Bao/Journal of Software, 2015,26(3):562−573 (in Chinese). http://www.jos.org.cn/1000-9825/4766.htm

Workflow Task Assignment Strategy Based on Social Context

YU Yang[1], WANG Ying[2], LIU Xing-Mei[2], CHEN Jian[2]

[1](School of Software, Sun Yat-Sen University, Guangzhou 510006, China)

[2](School of Information Science and Technology, Sun Yat-Sen University, Guangzhou 510006, China)

Abstract: Task assignment strategy has a great impact on the performance of the workflow management system. The instability of human resource brings challenges to task assignment. General task assignment strategies have some deficiencies. First, they only consider the individual attributes of candidate resources, ignoring the influences to the candidate resources from other resources in process. In addition, they need to setup a capability index of each resource in advance. However, it is hard to make the capability index fit the actual situation, and a wrong capability index will make the workflow engine assign the task to the unsuitable resource, degrading the performance of workflow management system. To overcome the above deficiencies, four Q-learning-based task assignment algorithms are proposed according to different state transition views and different reward functions. Simulation experiments show that Q-learning-based task assignment algorithms can work well even without setting up a capability index in advance. Also due to their support to consider the social relationship, the average time of case completion decreases.

Key words: workflow; task assignment; social relationship; Q-learning

随着计算机的普及、网络技术的发展,工作流技术已被广泛运用到物流、电子商务、医疗、行政办公等各种企业管理系统当中.它将业务过程从应用程序中分离出来,进行独立管理,使得软件更容易支持多变的业务流

* 基金项目: 国家自然科学基金(60873162); 广东省自然科学基金(S2012010009634); 广东省重大科技专项(2012A010800012); 广州市科技计划(12A12051586)

收稿时间: 2014-04-30; 修改时间: 2014-08-08; 定稿时间: 2014-11-21

© 中国科学院软件研究所 http://www.jos.org.cn

程,或是更高效地并行处理任务,从而提高工作流效率.工作流管理系统的核心是工作流引擎,它的主要任务之一是根据设定的工作流模型进行任务分派(task assignment),即按照一定的分派策略为任务选择执行资源.不同的资源具有不同的属性,同一项任务由不同的资源执行会有不同的效率;不同的分派策略会为任务挑选不同的资源,导致任务处理效率的差异,因此对工作流的执行性能有着重大影响.

在进行任务分派时,需要全面考虑资源的属性.在工作流管理系统中,根据人类的参与与否,资源分为人力资源与非人力资源.人力资源不但具有个体属性(如能力、兴趣、经验等),还具有社会及心理属性(如工作压力、合作者间熟悉程度等).资源的社会属性对知识型团队的协作效率影响尤为明显.目前的任务分派策略研究大多只考虑资源的个体属性,而较少考虑到资源社会属性的影响.另外,在关注性能的任务分派策略中,往往需要预先设定资源的能力数值,但人力资源的能力值很难被精确量化,从而影响任务分派的合理性.

为了解决以上问题,本文引入社会关系影响的内容,对其在任务分派问题的影响进行了分析.在此基础上,将社会关系对资源能力的影响以及任务分派问题进行建模,继而提出两种状态转移规则和 4 种奖励函数,并基于 Q 学习算法,结合社会关系影响提出新的任务分派算法以探索基于社会关系的任务分派策略,从而提高工作流管理系统的性能.本文通过多个仿真实验,分析了基于 Q 学习的任务分派算法的效果,验证了 Q 学习方法解决工作流任务分派问题的可行性,为解决工作流任务分派问题提供了一种新的思路.

本文第 1 节简单介绍任务分派及社会关系领域的研究成果.第 2 节描述基于 Q 学习的任务分派策略,并讨论对于工作流的任务分派问题以及 Q 学习算法的状态空间、动作空间、状态转移规则以及奖励函数的定义.第 3 节主要描述基于 Q 学习的任务分派算法以及如何在基于 Q 学习的任务分派算法中考虑社会关系的影响,并给出 Q 值表的迭代更新流程.第 4 节为实验部分,描述基于 Q 学习的任务分派算法相关的仿真实验设置与实验结果,对实验结果进行分析与总结.第 5 节总结本文的工作成果,并指出本文研究的可改进之处.

1 相关研究现状分析

1.1 任务分派研究现状

任务分派的研究领域主要分为两个方面:任务分派控制和任务分派规则描述.本文关注的是任务分派控制的研究,即任务分派运行时根据何种策略选择候选资源.文献[1]在任务分派时不仅考虑当前待分派任务的候选资源的情况,还考虑了其前置执行者对候选资源的影响,其中用到了最短处理时间、最短完成时间、平均工作负载和最短工作列表这 4 种传统、简单的任务分派算法,但其未能充分考虑任务分派过程中的负载均衡问题.文献[2]在任务分派时不仅基于当前待分派任务选择执行的资源,也基于整个案例流程进行考虑,根据历史的执行记录优先选择合作次数多的资源共同执行流程,也未能充分考虑资源的负载情况.文献[3–5]提出对任务进行动态分派,能够充分考虑各个资源分派时的情况,符合现实意义,本文主要研究的也是任务的动态分派情况.文献[6,7]都提出了从工作流日志中挖掘任务角色分配规则的方法,从而使任务分派的操作脱离人工依赖,实现自动化或半自动化.文献[8]使用隐马尔可夫模型(hidden Markov model)为给定流程的各个子任务选择最有可能的执行角色,为文本问题建模提供了思路.文献[9]提出了类 Apriori 算法,根据日志中的频繁模式为任务分派执行的资源,其任务分派指定到具体的单个资源上,而非选择一类角色.文献[8]从数据角度关注业务流程中任务的分派.文献[10]总结了 43 种可用于任务分派的资源模式,但除了最短队列(shortest queue)模式考虑了资源的工作负载外,其他模式都是关注于资源的角色权限、任务的生命周期,没有真正考虑资源的性能.文献[9]基于模糊理论与三角模糊数提出了 MCTA(multi-criteria task assignment)算法,在任务分派时能综合考虑任务属性、资源能力、社会关系,但没有说明社会关系如何影响资源的能力.文献[11]则增加了对资源工作负载的考虑,提出了 MMTA(multilevel model of task assignment)算法.文献[12]从多目标 QoS(quality of service)约束方面研究任务分派问题,使得流程在执行时能够满足多方面的 QoS 约束.值得一提的是,它指出资源的能力有时存在与其标称值不一样的情况,因此加入了信任可靠值.文献[13]通过预测案例过程未来的路由状况和任务处理时间,采用启发式算法选择较优的资源,文献[14–16]为任务分派问题提供了一个解决框架,但没有考虑多案例同时到达的情况,也没有考虑资源间的社会关系.文献[17]提出利用社会关系矩阵解决任务分派.文献[18,19]使用增强学习的方

© 中国科学院软件研究所 http://www.jos.org.cn

法解决工作流中的任务分派问题,实验结果表明:使用 Q 学习算法进行资源分配的平均案例完成时间比 FIFO, SLACX 和 SPT 算法短,对本文的研究有着较大的启发作用.

1.2 社会关系研究现状

目前,已经有一些研究表明,人的社会及心理属性对其工作效率有影响.文献[1]提出了社会关系影响因子的概念,并将其用于任务分派策略中优化工作流系统的性能,其对社会关系的建模的正确性影响着系统的优化效果.如果建模不正确,则得不到预期的结果.文献[20]发现了耶基斯-多德森定律(Yerkes-Dodson law),指出:人的工作表现与其受到的压力呈倒 U 型分布,工作压力过低或过高时工作表现较差,适中的工作压力才能使表现达到顶峰.但对倒 U 模型的衡量比较难.文献[21]提出了工作流系统中资源的两种工作负载定义方式,并使用线性回归分析的方法从工作日志中验证了资源的任务处理时间受其工作负载影响,一定程度上验证了此定律.文献[22]使用问卷调查的方式,总结出不同性别、不同岗位工作人员的工作压力差异,其中,男性工作压力大于女性,生产、行政及研发岗位压力居前.文献[23]论证了 3 类假设,指出:个人的社会关系网络结构对个人的工作绩效有影响、团队的社会关系网络结构对团队的绩效有影响.文献[24,25]根据工作流模型、角色-执行者映射关系,以前置任务者与后继任务者的关系建立基于工作流的社会关系网络,并分析工作流模型中最重要的执行者.

综上,目前大部分研究都只考虑资源的个体属性,没有考虑到其他资源带来的影响.并且,传统的一些任务分派算法缺少对资源负载情况的充分考虑.针对以上问题,本文提出了基于社会关系的 Q 学习任务分派算法.

2 任务分派问题建模

为了研究基于社会关系的任务分派对工作流系统性能的影响,首先要对社会关系的影响进行建模.本文将同一案例中的工作传递关系看作资源的社会关系.这种社会关系在实际中很可能影响资源处理任务时的能力表现.文献[1]提出了社会关系影响因子的概念,并将其用于任务分派策略中优化工作流系统的性能.实验结果表明:在任务分派算法中引入社会关系影响因子后,可以减少系统的案例完成时间和提高系统的吞吐量.然而,有一个较大的因素影响着社会关系影响因子的优化效果——社会关系影响因子的建模正确性.如果社会关系影响因子的模型不符合现实情况,那么会得不到预想的优化效果.

针对这一难点,本文基于 Q 学习方法给出考虑社会关系的任务分派问题模型.Q 学习算法在平行机调度领域已有一定研究,使用 Q 学习算法解决任务分派问题还有以下优点:(1) 在不知道状态转移函数、奖励函数的准确定义下,仍能取得较好效果;(2) 关注的是长期利益而不是短期利益.

2.1 状态空间与动作空间

要给定基于 Q 学习的任务分派问题的模型,首先要将任务分派问题建模成马尔可夫决策过程.根据马尔可夫决策过程得知,我们需要定义任务分派问题中的状态空间、动作空间、状态转移函数以及奖励函数.

定义 2.1(状态空间). 使用 $s\subseteq T*Workload$ 表示任务分派问题中的状态空间,其中,

- T 表示工作流模型中的任务集合;
- $Workload$ 表示工作流系统中全部资源的负载情况集合.

对于某一具体状态 $s_\tau\in s$,令 $s_\tau=(t,Workload_\tau)$,其中,

- $t\in T$,表示工作流模型中的某一任务;
- τ表示系统的某一决策时刻,即有一个就绪工作项生成,需要对其进行任务分派的时刻;
- $Workload_\tau\in Workload$,$Workload_\tau=\{r.w|r\in R\}$,表示所有资源的工作列表的一种可能状况.也就是说,$Workload$ 表示的是所有资源的工作列表集合的所有可能状况.

定义 2.2(动作空间). 使用 $A=R$ 表示任务分派问题中的动作空间,其中,$A=R$ 表示工作流模型中的资源集合.对于某一状态 $s\in S$,使用 $a(s)=candidates(s.t)$表示状态 s 的可选动作,其中,$s.t$ 表示状态 s 所要分派的任务.

结合定义 2.1 与定义 2.2 可以看出:在某一状态执行某一动作是指在某种系统的工作负载下(具体表现为所有资源的工作列表情况),将某一就绪工作项分派给一名该任务的候选执行者,符合任务分派的意义.

© 中国科学院软件研究所 http://www.jos.org.cn

2.2 状态转移与奖励函数

根据状态转移规则可知,工作流中的任务分派问题有两种状态转移视角:一种是类似于平行机调度视角,忽略任务与任务之间的顺序,更关注系统中资源负载的变化,将所有处理中的案例当作一个整体看待,本文称整体视角;另一种是更关注工作流任务的先后次序,每次状态转移都针对同一案例,本文称为案例视角.

2.2.1 基于整体的状态转移视角

这里使用图 1 说明何为整体视角.假设在分派 *case*3 的 *T*2 之前,引擎之前分派的就绪工作项是 *case*2 的 *T*2.若在分派 *case*3 的 *T*2 时,状态为 $s'=s_\tau=(T2,Workload_\tau)$,那么整体视角下的上一状态就是 $s=s_{\tau-1}=(T2,Workload_{\tau-1})$.值得注意的是:

(1) s_τ中的 *T*2 是指 *case*3 的 *T*2,而 $s_{\tau-1}$ 中的 *T*2 则是指 *case*2 的 *T*2;

(2) $Workload_{\tau-1}$ 中的某个在处理工作项变为了 $Workload_\tau$中的就绪工作项,在整体视角中,资源工作负载的变化是连锁的.

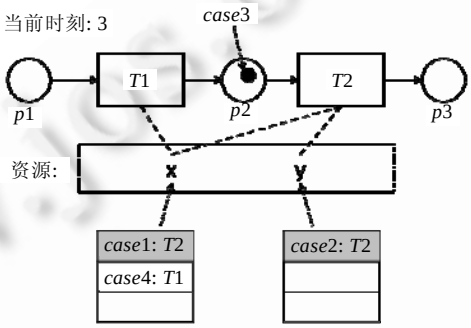


Fig.1 Workflow system at sometime

图 1 某一运行时刻的工作流系统

为方便叙述,称整体视角为 *V*1.基于整体视角,本文定义了针对任务的奖励函数 *V*1_*r*1(关注短期利益)与针对案例的奖励函数 *V*1_*r*2(关注长期利益).

定义 2.3(奖励函数 *V*1_*r*1). 设当前决策时刻为τ,时间戳为 ts_τ:

$$V1_r1=\frac{1}{ts_\tau-ts_{\tau-1}+1} \tag{1}$$

定义 2.4(奖励函数 *V*1_*r*2). 设当前决策时刻为τ,时间戳为 ts_τ,此时案例完成总数为 $cnum_\tau$:

$$V1_r2=\frac{cnum_\tau-cnum_{\tau-1}}{ts_\tau-ts_{\tau-1}+1} \tag{2}$$

奖励函数 *V*1_*r*1 的实质是单位时间内的任务完成数.由于每个案例都是由一定数量的任务组成,系统的任务完成得越快,也暗示着系统的案例完成得越快,平均每个案例的完成时间越少.所以,奖励函数 *V*1_*r*1 理论上能导向减少平均案例完成时间的局面.

奖励函数 *V*1_*r*2 的实质则是单位时间内的案例完成数.单位时间内的案例完成数越多,表示案例完成得越快,那么平均每个案例完成的时间也就越少.所以,奖励函数 *V*1_*r*2 理论上也能导向平均案例完成时间的减少.

2.2.2 基于案例的状态转移视角

同样使用图 1 说明何为案例视角,同样假设在分派 *case*3 的 *T*2 之前,引擎之前分派的就绪工作项是 *case*2 的 *T*2.若在分派 *case*3 的 *T*2 时,状态为 $s'=s_\tau=(T2,Workload_\tau)$,那么案例视角下的上一状态就是 $s=(T1,workload')$.值得注意的是:

(1) *s* 中的 *T*2 与 *s*′中的 *T*1 都是属于 *case*3 的,而且 *T*2 与 *T*1 是满足工作传递关系的.在案例视角下,除了并行任务外,都会满足 $s.t \trianglerighteq s'.t$;若是并行任务,则会触发连续的决策时刻;

© 中国科学院软件研究所 http://www.jos.org.cn

(2) 无法得知状态 s 与状态 s' 之间经过了多少个决策时刻,因为状态 s 与状态 s' 之间可能有多个新案例到达或其他案例的任务完成,而在案例视角下,状态转移是基于单个案例的,其他案例的变化是无法感知的.

为方便叙述,称案例视角为 $V2$.基于案例视角,本文定义了针对任务的奖励函数 $V2_r1$(关注短期利益)与针对案例的奖励函数 $V2_r2$(关注长期利益).

定义 2.5(奖励函数 $V2_r1$). 设当前决策时刻的时间戳为 ts',同一案例的上一次决策时刻的时间戳为 ts:

$$V2_r1=\begin{cases}0, & ts'=ts\\ \dfrac{1}{ts'-ts}, & ts'\neq ts\end{cases} \tag{3}$$

定义 2.6(奖励函数 $V2_r2$). 设 t 为当前需要分派的任务,σ为当前需要决策的案例,$\sigma.CCT$ 为该案例的完成时间:

$$V2_r2=\begin{cases}\dfrac{1}{\sigma.CCT}, & t \text{ is None}\\ 0, & \text{else}\end{cases} \tag{4}$$

奖励函数 $V2_r1$ 的实质也是单位时间的任务完成数,奖励值越大,表示任务完成得越快.在单个案例中,若该案例的各个任务的完成时间越短,那么该案例的总完成时间也会越短.奖励函数 $V2_r1$ 通过尽量降低每个案例的完成时间而达到减少平均案例时间的目标.

奖励函数 $V2_r2$ 的实质是吞吐量,其采用案例完成时间的倒数作为奖励值.奖励值越大,表示该案例的完成时间越少.因此,奖励函数 $V2_r2$ 也能通过尽量降低每个案例的完成时间而达成减少平均案例完成时间的目标.

第 2.2 节提到的 4 个奖励函数中:$V1_r1$ 与 $V2_r1$ 的短期利益性体现在其是通过减少任务完成时间而达到减少案例完成时间的目标,没有考虑社会关系及任务重复执行带来的影响;而奖励函数 $V1_r2$ 与 $V2_r2$ 则是直接采用案例完成时间作为目标.以上两种因素的影响会自然反映到奖励函数当中,因此,奖励函数 $V1_r1$,$V2_r1$ 是关注短期利益的,奖励函数 $V1_r2$,$V2_r2$ 是关注长期利益的.

3 基于社会关系的任务分派算法

基于 Q 学习的任务分派算法在不需要预设资源能力的情况下能够取得好效果,而且支持在任务分派中考虑社会关系的影响,探索基于社会关系的任务分派策略,从而提高工作流管理系统的性能.

3.1 Q学习任务分派算法

为实现 Q 学习任务分派算法,还需定义评估函数 $Q(s,a)$及其实际估计函数 $\hat{Q}(s,a)$,将状态-动作对(s,a)映射成 Q 值,以便于迭代逼近 $Q(s,a)$.

根据马尔科夫决策过程引入评估函数 $Q(s,a)$,它表示在状态 s 使用动作 a 所获得的最大折算累积奖励值:

$$Q(s,a)\equiv r(s,a)+\gamma\max Q_{a'}(\delta(s,a),a') \tag{5}$$

其中,$r(s,a)$表示在状态 s 使用动作 a 所获得的奖励值;$\delta(s,a)$表示在状态 s 使用动作 a 所获得的下一状态;$0\leqslant\gamma<1$,表示在未来某时刻的奖励值会通过指数级的折算反映到 t 时刻的动作上.

在本文的定义中,有:

$$(s,a)=((t,workload),a)=(t,workload,a)\subseteq T\times Workload\times R \tag{6}$$

其中,T 和 R 都是有限集,可以映射到多维表格中,所以关键在于如何处理 $Workload$ 集合.考虑到执行者的工作列表长度能在一定程度上反映执行者的工作负载,所以本文根据执行者的工作列表长度,将执行者的工作负载概化为空闲($FREE$)、低负载(LOW)、高负载($HIGH$)这 3 个等级,计算方式如下:

定义 3.1(工作负载概化函数). 设 $t\in T$ 为待分派的任务,对于资源 $r\in candidates(t)$,资源 r 的工作负载由函数 $workload(r)$确定:

© 中国科学院软件研究所 http://www.jos.org.cn

$$workload(r)=\begin{cases}FREE, & |r.wl|=0\\ LOW, & |r.wl|\leqslant avg_workload(t)\\ HIGH, & |r.wl|>avg_workload(t)\end{cases} \tag{7}$$

其中,函数 $avg_workload(t)$表示任务 t 的候选执行者的平均工作列表长度.

由于任务分派时只需从候选执行者中进行选择,所以定义 3.1 中的平均工作列表长度也只考虑了任务的候选执行者.工作列表为空的执行者在接到任务后能立即开始处理,不用叠加任务的排队时间,所以本文将工作列表为空的状态列为 *FREE* 等级,其余再按照高于平均值或低于平均值分为 *HIGH* 和 *LOW* 两级.

至此,我们能够将 $\hat{Q}(s,a)$ 映射成一个大表:

$$\hat{Q}(s,a)=Q_Table(t,r,workload(r)) \tag{8}$$

如:$Q_Table(t1,r1,FREE)$表示将任务 $t1$ 分派给执行者 $r1$ 且 $r1$ 的负载水平为空闲时的 Q 值;$Q_Table(t1,r2,LOW)$表示将任务 $t1$ 分派给执行者 $r2$ 且 $r2$ 的负债水平为低的 Q 值.

算法 3.1 描述了使用整体视角的 Q 学习任务分派算法的主要步骤.算法第 4 行中是根据 Q 值选择执行者的操作.在一般的情况下:迭代初期,选择受 Q 值的影响较少、偏向于随机地探索各种状态-动作对(不一定选择 Q 值最高的作为执行者);随着迭代次数的增加,选择受 Q 值影响的权重越来越大,偏向与选择 Q 值最高的执行者.算法的第 7 行可使用定义 2-3 或定义 2-4 作为奖励函数.算法第 9 行是对 Q 值表进行迭代更新的过程描述.

算法 3.1. 基于整体视角的 Q 学习任务分派算法.

1. 初始化 *Q_Table* 的每个值为 0
2. 就绪工作项产生,获得当前状态 s
3. 一直重复直到系统停止:
4. 根据 $\hat{Q}(s,a)$ 的值选择执行者 r_{pick},$a\leftarrow r_{pick}$,并将任务 t 分派给 r_{pick}
5. 时钟推进至新的就绪工作项产生或某在处理工作项完成
6. 获得新状态 s'
7. 接收到回报 r
8. 按照下式更新 *Q_Table* 表项:
9. $\hat{Q}_n(s,a)\leftarrow(1-\alpha_n)\hat{Q}_{n-1}(s,a)+\alpha_n[r+\gamma\max_{a'}\hat{Q}_{n-1}(s',a')]$
10. $s\leftarrow s'$

算法 3.2 则描述了使用案例视角的 Q 学习任务分派算法的主要步骤.由于都是基于 Q 学习算法的,其框架与算法 3.1 类似,但注意以下几点不同:算法第 5 行,在根据 Q 值选择完执行者后,需要将对应的状态-动作对保存到案例σ的属性当中,以便进行之后的 Q 值表更新;算法第 8 行,在到达新状态后,需要根据案例σ'读取上一次的状态-动作对,这里注意,案例σ'与案例σ可能是不同的案例,这是与算法 3.1 的最大不同;算法第 9 行,可以使用定义 2.5 或定义 2.6 作为奖励函数.

γ确定了延迟回报和立即回报的相对比例:如果γ=0,那么只考虑立即回报;当γ被设置为接近 1 的值,未来的回报相对于立即回报有更大的重要程度[26].在任务分派中,我们更关注长期利益,因此在算法 3.1 与算法 3.2 中,长期率系数γ均取 0.9,使用较高的折算率,使奖励的影响效果更长远.

使用三元组从 Q 值表读取 Q 值的操作,复杂度可以当作 $O(1)$.基于 Q 学习的任务分派算法在每次进行决策(即每次迭代过程,算法 3.1 与算法 3.2 的循环体部分)的时候,只需要遍历所有的候选执行者即可,所以基于 Q 学习的任务分派算法的时间复杂度是 $O(n)$,其中,n 为候选执行者的规模.

算法 3.2. 基于案例视角的 Q 学习任务分派算法.

1. 初始化 *Q_Table* 的每个值为 0
2. 就绪工作项产生,获得当前状态 s
3. 一直重复直到系统停止:
4. 根据 $\hat{Q}(s,a)$ 的值选择执行者 r_{pick},$a\leftarrow r_{pick}$,并将任务 t 分派给 r_{pick}

© 中国科学院软件研究所 http://www.jos.org.cn

5. 根据对应的案例σ,保存(s,a)
6. 时钟推进至新的就绪工作项产生或某在处理工作项完成
7. 获得新状态 s'
8. 根据新的案例σ',读取(s,a)
9. 接收到回报 r
10. 按照下式更新 Q_Table 表项:
11. $\hat{Q}_n(s,a) \leftarrow (1-\alpha_n)\hat{Q}_{n-1}(s,a) + \alpha_n[r + \gamma \max_{a'} \hat{Q}_{n-1}(s',a')]$
12. $s \leftarrow s', \sigma \leftarrow \sigma'$

3.2 加入社会关系的Q学习任务分派算法

在基于 Q 学习的任务分派算法中,考虑社会关系的影响比较容易,只需要修改状态空间的定义即可.

定义 3.2(考虑社会关系的状态空间). 使用 $S_{sc} \subseteq T \times R \times Workload$ 表示任务分派问题中的状态空间,其中,

- T 表示工作流模型中的任务集合;
- R 表示工作流模型中的资源集合;
- $Workload$ 表示工作流系统中全部资源的负载情况集合.

定义 3.2 是对定义 2.1 的扩展,在其基础上加入了前置任务执行者的概念.例如,对于某一状态 $s=(t2,r_{ex},workload) \in S_{sc}$,其表示刚刚完成工作项的执行者是 r_{ex},现在要对任务 $t2$ 进行分派.

由于状态空间被修改了,所以还要修改 $\hat{Q}_n(s,a)$ 函数:

$$\hat{Q}(s,a) = Q_Table(t, r_{ex}, r, workload(r)) .$$

对 $\hat{Q}_n(s,a)$ 函数的修改,实际是在 Q 值表上增加前置任务执行者一个维度.可以这么理解:将原来的一张由(任务,执行者)二元组对应的表格,分开几张由(任务,前置任务执行者,执行者)三元组对应的表格.

除上述两处修改外,加入考虑社会关系不需要对动作空间、状态转移规则、算法 3.1、算法 3.2 进行修改.

4 仿真实验

4.1 Q学习任务分派算法实验

4.1.1 实验目的与方案

本节希望通过仿真实验检验第 2 节提出的几个Q学习任务分派算法的性能,以验证其解决任务分派问题的可行性.*SWL*(最短工作列表分派算法)和 *SCT*(最短完成时间分派算法)被用于实验以验证 Q 学习算法的性能[1]. Q 学习任务分派算法相关的仿真实验均采用如图 2 所示的工作流模型.工作流流程由 5 个任务组成,其中,$T3$ 与 $T4$ 是并行任务,$T3$ 需要按照一定概率 $Q(1-P_{T3})$重复进行,且概率 P_{T3} 与 $T3$ 的执行者有关.此工作流流程包含了工作流中顺序、并行、选择、重复这 4 种基本路由结构.

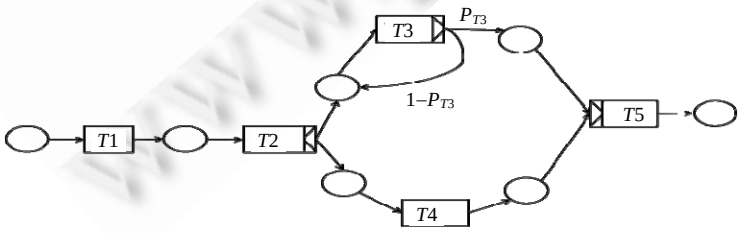


Fig.2 Workflow model for Q-learning-based task assignment experiment

图 2 用于 Q 学习任务分派算法实验的工作流模型

资源的能力(即任务执行时间,单位分钟)有 3 种配置方案(见表 1、表 2 与图 3).在职能设置中,资源-任务对的情况将一对一、一对多、多对一的情况都包含了.配置 1(见表 1)表示一般的能力设置,同类任务的资源能力

© 中国科学院软件研究所 http://www.jos.org.cn

有存在差异的情况,也有不存在差异的情况,但资源之间没有社会关系影响.配置 2(见表 2)中,同类任务的候选资源能力都一样,以便于体现社会关系影响的意义.图 3 是仿真程序中使用的带社会关系影响的资源能力设置文件的片段,是在配置 2 上加入社会关系影响得到的.社会关系影响的设置原则是:待分派任务的候选资源的能力受到前置执行者的影响.此处的设置符合现实意义:若前置执行者是本人,则本人的当前任务处理时间会减少 20%;若前置执行者不是本人,则本人的当前任务处理时间会等概率地不变或增加 20%.在任务 *T*3 的重复概率方面,分为相同重复概率(*R*3 与 *R*4 均为 0.2 概率需要重复执行)和不同重复概率(*R*3 为 0.2,*R*4 为 0.1)两种情况.

Table 1 First experimental resource configuration for Q-learning task assignment algorithm

表 1 Q 学习任务分派算法实验资源能力配置 1

	*T*1	*T*2	*T*3	*T*4	*T*5
*R*1	10	\	\	\	5
*R*2	5	\	\	\	5
*R*3	\	10	10	\	5
*R*4	\	\	15	\	\
*R*5	\	\	\	15	\

Table 2 Second experimental resource configuration for Q-learning task assignment algorithm

表 2 Q 学习任务分派算法实验资源能力配置 2

	*T*1	*T*2	*T*3	*T*4	*T*5
*R*1	10	\	\	\	5
*R*2	10	\	\	\	5
*R*3	\	10	10	\	5
*R*4	\	\	10	\	\
*R*5	\	\	\	10	\

```
{
    "task1":{"all":{"r1":10, "r2":10}},
    "task2":{"r1":{"r3":12},"r2":{"r3":10}},
    "task3":{"r3":{"r3":8, "r4":10}, "r4":{"r3":12, "r4":8}},
    "task4":{"r3":{"r5":10}},
    "task5":{"r3r5":{"r1":6,"r2":5,"r3":4},"r5r3":{"r1":6,"r2":5,"r3":4},"r4r5":{"r1":5,"r2":5,
}
```

Fig.3 The resource configuration based on the social relationships

图 3 带社会关系影响的资源能力配置

Q 学习任务分派算法实验按照如下进行:

(1) 在一般的配置下,初步对 2 种不同状态转移视角的各种 Q 学习任务分派算法进行评估,选择效果较好的状态转移视角进行改进;

(2) 测试 Q 学习任务分派算法能否应用社会关系提高工作流性能;

(3) 测试 Q 学习任务分派算法不用预设资源能力的特性.

为此,本文拟定了 3 个实验方案(见表 3).在仿真实验中,仿真单位时间设为 1 分钟.案例到达的概率服从二项分布,平均每 60 分钟到达 1 个案例.在每个对比实验中,各算法均采用同一案例到达队列进行 500 次仿真,其中队列包含 1 000 个案例,每次仿真以 1 000 个案例的平均案例完成时间作为评价标准.为了使 Q 学习遍历所有动作-状态对,Q 学习任务分派算法的前 250 次运行采用完全随机的分派方式(但会迭代更新 *Q* 值),而后 250 次采用确定的分派方式(一定选取 *Q* 值最高的候选执行者).资源的实际任务执行时间等于其相应任务的能力设置.仿真实验均使用 Python 2.7 实现.

Table 3 Plan and objective of Q-learning task assignment algorithm

表 3 Q 学习任务分派算法实验的方案与目的

	资源能力设置	*T*3 重复概率	实验目的
实验 1	配置 1	不同	横向对比不同状态转移视角的 Q 学习任务分派算法
实验 2	配置 2	不同	观察 Q 学习中奖励函数 *r*2 是否真的存在关注长期利益的特性
实验 3	配置 2、配置 3	相同	观察加入社会关系的 Q 学习算法对性能的影响

© 中国科学院软件研究所 http://www.jos.org.cn

4.1.2 仿真结果及分析

本节涉及的算法种类较多,为了方便描述,本文使用标签组合的方式对算法进行命名,命名规则见表 4.例如: *SCT_SC* 表示考虑社会关系影响的 SCT 分派算法;*Q_Table_V*1_*r*1 表示基于整体视角使用基于任务的奖励函数与 *Q* 值表的 Q 学习分派算法.

Table 4 Name tag and its significance of task assignment algorithm

表 4 任务分派算法名称标签及其意义

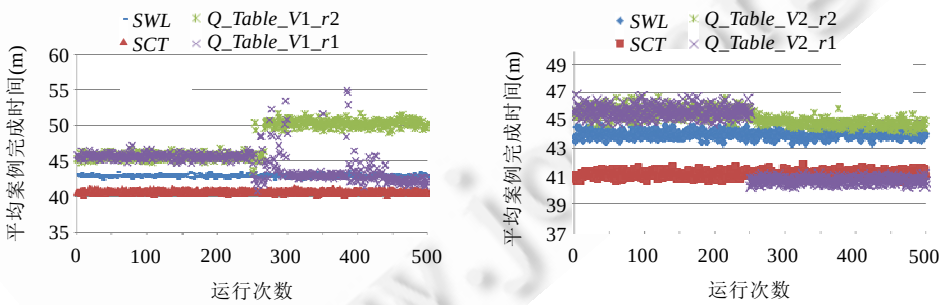
标签字样	意义
随机	随机任务分派算法
SWL	最短工作列表分派算法
SCT	最短完成时间分派算法
Q_Table	使用 *Q* 值表的 Q 学习分派算法
_*V*1	基于整体视角
_*V*2	基于案例视角
_*r*1	使用基于任务的奖励函数,关注短期利益
_*r*2	使用关注案例的奖励函数,关注长期利益
_*SC*	考虑社会关系影响

- 实验 1

实验 1 的目标是考察两种状态转移视角的 Q 学习任务分派算法的性能,结果如图 4 所示.

图 4(a)展示了基于整体视角的 Q 学习任务分派算法的仿真结果.从图中的结果看来,基于整体视角的 Q 学习任务分派算法效果并不好:采用关注短期利益的 *r*1 奖励函数只与 SWL 算法效果相当,效果存在不稳定的情况;关注长期利益的 *r*2 函数的奖励具有延迟性,因此,采用关注长期利益的 *r*2 效果反而比随机算法效果还差(*Q_Table* 算法的前 250 次仿真为随机分派).

图 4(b)展示了基于案例视角的 Q 学习任务分派算法的性能,可以看出,效果比基于整体视角的要好.在没有预设资源能力的情况下,*Q_Table_V*2_*r*1 算法的效果要比需要预设能力的 SCT 算法还要好,而且表现很稳定.理论上最好的 *Q_Table_V*2_*r*2 算法没有达到预期的效果,仅稍微优于随机分派算法,差于 *SWL* 算法.但是在往后的实验中,*Q_Table_V*2_*r*2 算法都有比较好的效果.



(a) 基于整体视角的 Q 学习任务分派算法效果 (b) 基于案例视角的 Q 学习任务分派算法效果

Fig.4

图 4

实验 1 的结果说明:在本文的状态空间定义下,基于案例视角的 Q 学习任务分派算法效果比基于整体视角的好,且 *Q_Table_V*2 算法能优于一般任务分派算法.

- 实验 2

图 5 表明,使用奖励函数 *r*2 的算法都得到了比 SCT 算法要好的效果.

© 中国科学院软件研究所 http://www.jos.org.cn

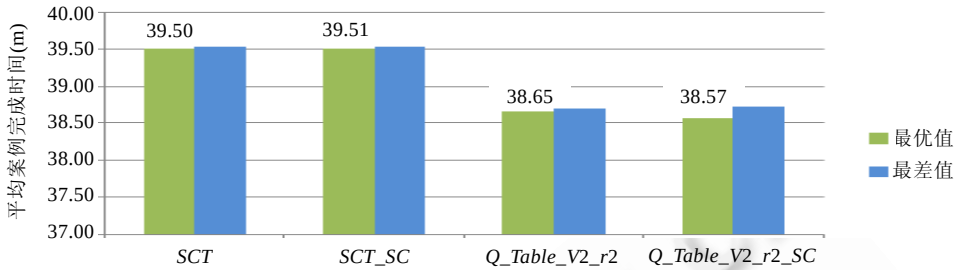


Fig.5 Simulation result of different repetition probability without social relationships

图 5 无社会关系影响,不同重复概率设置下的仿真结果

从表 3 的实验 2 设置中可以看出:对于任务 *T*3 来说,尽管 *R*3,*R*4 的处理时间是一样的,但 *R*4 只能处理 *T*3 这一种任务,*R*3 却可以处理多种任务,随时预留能力范围广的资源以便灵活调度,是一种任务分派策略的优化方向,所以在一般情况下,将 *T*3 交给 *R*4 是较优的选择.由于 SCT 算法在本文中只关注任务处理时间,所以并不能发现 *R*3 身兼多职、成功率低的问题,而奖励函数 *r*2 考虑的是执行者对整个案例的完成时间的影响.

- 实验 3

在 Q 学习任务分派算法中加入社会关系,相当于将二元组(任务,执行者)分为若干个三元组(任务,前置执行者,执行者),实验结果如图 6、图 7 所示.

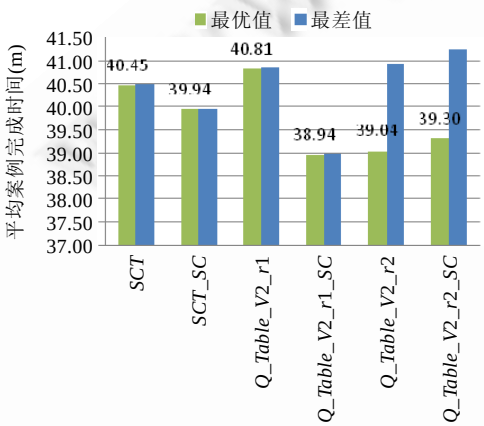


Fig.6 Simulation result of the same repetition probability with social relationships

图 6 有社会关系影响,相同重复概率设置下的仿真结果

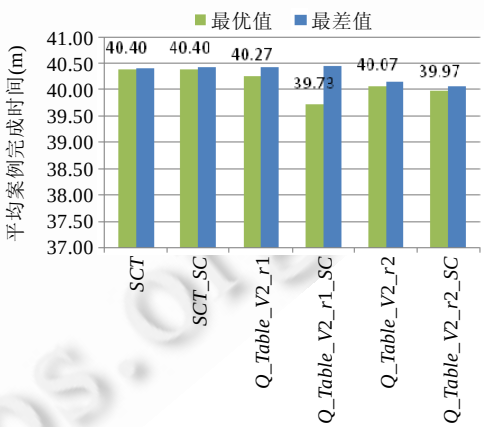


Fig.7 Simulation result of the same repetition probability without social relationships

图 7 无社会关系影响,相同重复概率设置下的仿真结果

实验 3 的仿真结果表明:有社会关系影响的前提下,在任务分派过程中考虑社会关系使得性能得到提升,平均案例完成时间降低了约 2 分钟(5%);在没有社会关系影响的前提下,即使在任务分派中考虑社会关系,也不会对性能带来负面影响.

5 结 论

本文主要研究了基于社会关系的工作流任务分派问题及其相关算法.通过引入 Q 学习方法,对资源之间的社会关系对于工作能力的影响进行建模,并利用这种社会关系的影响改进工作流任务分派策略,从而提高了工作流系统的整体性能.通过仿真实验可知,基于社会关系的 Q 学习任务分派策略是解决工作流系统中任务分派

© 中国科学院软件研究所 http://www.jos.org.cn

问题的一种可行方案.

关于社会关系的任务分派问题已有一定成果,但由于该问题在实际情况下影响的因素多且复杂,本文并未能全部考虑,今后可在以下两个方向进行完善:(1) 工作负载对工作效率的影响;(2) 针对真实工作流环境下的数据评测.

References:

[1] Xu JX, Huang ZG, Yu Y, Pan ML. A performance analysis on task allocation using social context. In: Proc. of the 2nd Int'l Conf. on Cloud and Green Computing (CGC). IEEE, 2012. 637−644. [doi: 10.1109/ CGC.2012.88]

[2] Yang HD, Wang CK, Liu YB, Wang JM. An optimal approach for workflow staff assignment based on hidden Markov models. In: Proc. of the Move toMeaningful Internet Systems: OTM 2008 Workshops. Berlin, Heidelberg: Springer-Verlag, 2008. 24−26. [doi: 10.1007/ 978-3-540-88875-8_12]

[3] Ho CJ, Vaughan JW. Online task assignment in crowsourcing markets. In: Proc. of the AAAI. Toronto, 2012. http://www.cs.ucla.edu/~cjho/pub/AAAI12_TaskAssignment.pdf

[4] Ho CJ, Jabbari SH, Vaughan JW. Adaptive task assignment for crowdsourced classification. In: Proc. of the 30th Int'l Conf. on Machine Learning (ICML 2013). 2013. 534−542. http://jmlr.csail.mit.edu/proceedings/papers/v28/ho13.pdf

[5] Bjornson E, Jorswirck E. Optimal Resource Allocation in Coordinated Multi-Cell Systems. 2013. [doi: 10.1561/0100000069]

[6] Koschmider A, Liu YB, Schuster T. Role Assignment in business process models. In: Proc. of the Business Process Management Workshops. Berlin, Heidelberg: Springer-Verlag, 2012. 37−49. [doi: 10.1007/978-3-642-28108-2_4]

[7] Meyer A, Pufahl L, Fahland D, Weske M. Modeling and enacting complex datadependencies in business processes. In: Proc. of the BPM. Springer-Verlag, 2013. [doi: 10.1007/978-3-642-40176-3_14]

[8] Liu TY, Cheng YL, Ni ZH. Mining event logs to support workflow resource allocation. In: Proc. of the Knowledge-Based Systems. 2012. [doi: 10.1016/j.knosys.2012.05.010]

[9] Russell N, Ter Hofstede AHM, Edmond D, Van der Aalst WMP. Workflow resource patterns. BETA Working Paper Series, WP 127. Eindhoven: Eindhoven University of Technology, 2004. [doi: 10.1007/11431855_16]

[10] Shen MX, Tzeng GH, Liu DR. Multi-Criteria task assignment in workflow management systems. In: Proc. of the 36th Annual Hawaii Int'l Conf. on System Sciences. IEEE, 2003. 9. [doi: 10.1109/HICSS.2003.1174458]

[11] Xiao ZJ, He QM, Chen Q. Multi-Level model of workflow task assignment in the fuzzy environment. Computer Research and Development, 2007,44(2):302−309 (in Chinese with English abstract). http://www.cnki.com.cn/article/cjfdtotal-jfyz200702016.htm

[12] Hu CH, Wu M, Liu GP. QoS scheduling based on trust relationships in the Web service workflow. Chinese Journal of Computers, 2009,32(1):42−53 (in Chinese with English abstract). [doi: 10.3724/SP.J.1016.2009.00042]

[13] Xiong F, Yuan YP, Wang YY, Wang GW. Task scheduling in multi-process with resource constraints under MG workflow. Advanced Materials Research, 2008,33:1425−1430. [doi: 10.4028/www.scientific.net/AMR.33-37.1425]

[14] Huang ZG, Lu XD, Duan HL. A task operation model for resource allocation optimization in business process management. IEEE Trans. on Systems, Man and Cybernetics, Part A: Systems and Humans, 2012,42(5):1256−1270. [doi: 10.1109/TSMCA.2012.2187889]

[15] Menon AK, Tamuz O, Gulwani S, Lampson B, Kalai AT. A machine learning framework for programming by example. In: Proc. of the 30th Int'l Conf. on Machine Learning (ICML 2013). 2013. 187−195. http://machinelearning.wustl.edu/mlpapers/paper_files/ICML2013_menon13.pdf

[16] Huang ZX, van der Aalst WMP, Lu XD, Duan HL. Reinforcement learning based resource allocation in business process management. Data & Knowledge Engineering, 2011,70(1):127−145. [doi: 10.1016/j.datak.2010.09.002]

[17] Bajaj A, Russell R. AWSM: Allocation of workflows utilizing social network metrics. Decision Support Systems, 2010,50(1):191−202. [doi: 10.1016/j.dss.2010.07.014]

[18] Fleischmann A, Schmidt W, Stary C, Strecker F. Nondeterministic events in business processes. In: Proc. of the Business Process Management Workshops. Berlin, Heidelberg: Springer-Verlag, 2013. 364−377. [doi: 10.1007/978-3-642-36285-9_40]

[19] Huang ZX, van der Aalst WMP, Lu XD, Duan HL. An adaptive work distribution mechanism based on reinforcement learning. Expert Systems with Applications, 2010,37(12):7533−7541. [doi: 10.1016/j.eswa.2010.04.091]

© 中国科学院软件研究所 http://www.jos.org.cn

[20] Hollands JG, Wickens CD. Engineering Psychology and Human Performance. Prentice Hall, 1999. http://webfiles.ita. chalmers.se/~mys/HumanAspects/WickensHollands/0_Wickens_Index_Preface.pdf

[21] Nakatumba J, van der Aalst WMP. Analyzing resource behavior using process mining. In: Proc. of the Business Process Management Workshops. Berlin, Heidelberg: Springer-Verlag, 2010. 69–80. [doi: 10.1007/978-3-642-12186-9_8]

[22] Zhan WH, Gao JJ, Chen YW. The influence of organizational climate on job burnout: Mediating effect of job stress. Journal of Zhejiang University (Science Edition), 2013,40(1):112–118 (in Chinese with English abstract). [doi: 10.3785/j.issn.1008-9497.2013.01.023]

[23] Pei W. Research on the relationship between individual social networks and team performance in the in virtual teams [MS. Thesis]. Guangzhou: South China University of Technology, 2011 (in Chinese with English abstract). http://cdmd.cnki.com.cn/ Article/CDMD-10561-1011190262.htm

[24] Song J, Kim M, Kim H, Kim K. A framework: Workflow-based social network discovery and analysis. In: Proc. of the 2010 IEEE 13th Int'l Conf. on Computational Science and Engineering (CSE). IEEE, 2010. 421–426. [doi: 10.1109/ CSE.2010.74]

[25] Aalst W. Process Mining: Discovery, Conformance and Enhancement of Business Processes. Berlin: Springer-Verlag, 2011. [doi: 10.1007/978-3-642-19345-3]

[26] Mitchell TM, Wrote; Zeng HJ, Zhang YK, Trans. Machine Learning. Beijing: China Machine Press, 2003. (in Chinese)

附中文参考文献:

[11] 肖郑进,何钦铭,陈奇.模糊环境中工作流任务分派的多级模型.计算机研究与发展,2007,44(2):302–309. http://www.cnki.com.cn/article/cjfdtotal-jfyz200702016.htm

[12] 胡春华,吴敏,刘国平.Web 服务工作流中基于信任关系的 QoS 调度.计算机学报,2009,32(1):42–53. [doi: 10.3724/SP.J.1016.2009.00042]

[22] 詹文慧,高金金,陈毅文.组织气氛对工作倦怠的影响:工作压力的中介作用.浙江大学学报:理学版,2013,40(1):112–118. [doi: 10.3785/j.issn.1008-9497.2013.01.023]

[23] 裴伟.虚拟团队中个人社会网络与团队绩效的关系研究[硕士学位论文].广州:华南理工大学,2011. http://cdmd.cnki.com.cn/Article/CDMD-10561-1011190262.htm

[26] Mitchell TM,著;曾华军,张银奎,译.机器学习.北京:机械工业出版社,2003.

余阳(1966－),男,江西奉新人,博士,教授,CCF 高级会员,主要研究领域为工作流,服务计算,软件工程.

刘醒梅(1987－),女,硕士,主要研究领域为工作流,软件平台技术.

王颖(1990－),女,硕士,主要研究领域为虚拟组织,信任信誉模型.

陈健(1989－),男,硕士,主要研究领域为工作流,服务计算.

© 中国科学院软件研究所 http://www.jos.org.cn