

模糊环境中工作流任务分配的多级模型

肖郑进 何钦铭 陈 奇

(浙江大学计算机科学与技术学院 杭州 310027)

(xaozj1976@163.com)

A Multilevel Model of Task Assignment in Fuzzy Situations of Workflow

Xiao Zhengjin, He Qinming, and Chen Qi

(College of Computer Science and Technology, Zhejiang University, Hangzhou 310027)

Abstract The automatic assignment of user task is a key technology for increasing the running efficiency of workflow management system. Workflow engines are always obliged to assess a complex and confusing situation, identify the complex relationship of candidates and tasks, decide to assign a special task to the most appropriate candidates, and ensure workflow system executes efficiently. The effectiveness depends largely on the evaluation of the influencing factors. After systematically analyzing the influencing factors and their linguistic fuzziness of user task assignment, a sectional multilevel model of task assignment and a task assignment method corresponding to the multilevel model are put forward. Furthermore, the design method of all adaptive weighting factors in the task assignment model is also discussed briefly. Finally, the performance of the multilevel model for task assignment is given by comparing the simulation result of the multilevel model and other two task assignment methods.

Key words workflow; task assignment; fuzzy theory

摘要 对工作流实例中各用户型任务进行自动优化分配是提高工作流管理系统运行效率的关键技术之一。在详细分析了影响用户型任务分配的多种因素及其语意模糊性后,提出了一个可组合的任务分配多级模型以及相应的任务分配方法,详细讨论了具有自适应能力的影响因素权重设计方法。最后,通过与其他任务分配方法对比,分析了该任务分配方法的性能。

关键词 工作流;任务分配;模糊理论

中图法分类号 TP311

实施工作流管理的主要目的是通过合理地调用和分配与活动有关的资源以实现业务过程的高效、自动化执行。各活动的参与者定义是活动与其执行者间的一种抽象,工作流引擎计算这些抽象定义,并根据计算结果把任务分配给具体的人或程序执行^[1]。基于规则的参与者定义减小了因组织模型变化对过程模型产生的影响,提高过程模型的重用度^[2]。一旦组织模型中满足参与者抽象定义要求的资源不惟一时,必须有合适的分配策略来把任务分配其中的一个或一组候选人来完成,从而实现工作流管理系统“把任务分配给合适的人执行”的功能。

根据参与者类型不同,任务可以分为两种:由人完成的用户型任务和由应用程序或机器自动完成的自动型任务。与自动型任务分配研究成果相比^[3-4],目前对用户型任务分配策略的研究还比较少。Aalst提出了团队参与者工作流模型^[5],对需要多人协作完成的任务进行分析和建模;张晓光等人提出了一种面向角色与团队的混合任务分配方法^[6],以支持任务动态分配;为解决参与者使能问题,唐达等人提出了一种基于Petri网的参与者建模方法^[7]。但这些方法都没对候选人多于1人时,如何把任务分配给最适合的候选人这个问题进行研究。

传统的基于角色的任务分配只考虑权限的匹配^[2],没有考虑如任务重要性、员工经验等因素的影响。另一方面,传统的人力资源配置研究都采用精确值来评价任务及候选人的属性与状态,而实际应用中对于任务或候选人属性与状态的描述多具有模糊性,比如,很少采用类似于0.8这样的精确数值来评估一个任务的重要性,而多采用“非常重要”或“不重要”等具有模糊语意的语言变量来形容任务的重要性。所以用户型任务分配是一个模糊环境下多影响因素的决策问题。

Liang与Wang采用模糊理论来解决人力资源评估的不确定性问题,但他们只考虑候选人的能力是否匹配,没有考虑其他因素^[8]。Yaakob等人在Liang的基础上考虑了任务需要由团队完成时会受候选人私人关系好坏的影响^[9],但员工私人关系作为一种个人隐私,通常无法从组织模型中得到相关数据,所以这种扩展不具有可行性。Shen等人提出了一种多目标下的任务分配方法(multi-criteria task assignment, MCTA)^[10],对候选人的能力、负载和员工间社会关系进行简单加权,并没考虑任务重要性等的影响,也起不到相应的负载平衡作用。

本文在Shen等人提出的任务分配方法的基础上,对任务的重要性、类型以及候选人的负载、经验、能力等因素进行全面考虑,提出了一个任务分配多级模型(multilevel model of task assignment, MMTA),很好地解决了模糊语义环境中且受多因素影响的用户型任务分配问题。

1 影响任务分配的因素及其语意模糊性

实现用户型任务的自动分配需要考虑的因素比较多,包括待分配任务和候选人的属性与状态。对于待分配任务而言,主要考虑以下原则:

待分配任务的执行可能需要多种技能,但对于每种技能要求的权重不同,最合适的参与者应该是技能既能满足任务要求,以免影响任务的执行效果,又不超出任务的技能要求。不同任务具有不同的重要性,重要任务应该分配给完成高可靠性候选人,低重要性任务可分配给低可靠性候选人,以免低可靠性员工一直处于饥饿状态。分配任务时还要考虑任务的类型。经验型任务应该分配给有经验候选人,能耗型任务分配给谁就无所谓。任务的执行需要团队时,紧密配合型任务应该分配给配合密切的团队,团队成员间配合的好坏对执行效果影响不大,对团

队配合水平的影响可以少考虑一些。

影响任务分配的另一个重要因素就是各候选人的属性与当前状态:

为平衡候选人负载,减少任务的平均等待时间,候选人的负载越轻任务分配给他的可能性越大。候选人完成这类任务的平均成功率越高,任务分配给他的可能性就越大,而且任务重要性越高,对候选人的平均成功率就越重视。经验型任务应该分配给经验水平高的候选人。任务分配时可能要考虑的因素还有候选人的弹性程度,一个候选者往往不止担任一个角色,候选者所担任的角色越多其通用性越好,弹性程度越大。在任务分配时尽量让通用性好的候选人空闲着,为将来尽可能预留弹性资源。

如果任务执行需要一个团队,那么团队内各参与者间的关系对任务的执行效果就有影响,对于这种情况,任务分配最好还应该考虑各候选团队成员间的关系,包括各候选人间是否合作过、个人关系好坏等。上述这些影响任务分配的因素间关系如图1所示:

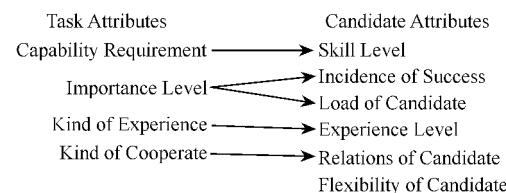


Fig. 1 The factors affect the task assignment and their relations.

图1 影响任务分配的各因素及各因素间关系

要将模糊情境用传统的量化方法加以合理地陈述是相当复杂且难以定义的,因此人造的语言变量在模糊情境中是相当必要的,Zadeh首先探讨了模糊环境中的决策问题^[11],使用语言变量如“不重要、非常重要”等来表示一些因素的要求或绩效。这种表示完全依靠建模者的直观感受来给出。例如,将任务重要性表示成“不重要、重要、很重要”等。采用语言变量来形容模糊情境中的因素不仅合理而且易于理解交流。

2 基于模糊理论的任务属性与参与者状态描述

2.1 模糊理论与三角模糊数

Dubois等人用模糊数来评估语言变量(linguistic variable)^[12]。三角模糊数 $\tilde{A}=(l, m, r)$ 是一模糊集合 \tilde{A} 的隶属函数 $\mu_{\tilde{A}}(x): R \rightarrow (0, 1)$ 表示 x 属于集合 \tilde{A} 的程度。 $\mu_{\tilde{A}}(x)$ 定义如下:

$$\mu_{\tilde{A}}(x) = \begin{cases} (x-1)(m-1), & 1 \leq x \leq m, \\ (r-x)(r-m), & m \leq x \leq r, \\ 0, & \text{其他.} \end{cases} \quad (1)$$

设 $\tilde{A}_1 = (l_1, m_1, r_1)$, $\tilde{A}_2 = (l_2, m_2, r_2)$, k 是任意实数. 三角模糊数的代数运算定义如下:

加法: $\tilde{A}_1 \oplus \tilde{A}_2 = (l_1 + l_2, m_1 + m_2, r_1 + r_2)$.

减法: $\tilde{A}_1 \ominus \tilde{A}_2 = (l_1 - l_2, m_1 - m_2, r_1 - r_2)$.

乘法: $\tilde{A}_1 \otimes \tilde{A}_2 = (l_1 l_2, m_1 m_2, r_1 r_2)$, $k \otimes \tilde{A}_1 = (k l_1, k m_1, k r_1)$.

除法: $\tilde{A}_1 \oslash \tilde{A}_2 = (l_1 / r_2, m_1 / m_2, r_1 / l_2)$, 其中, $l_2 \neq 0, m_2 \neq 0, r_2 \neq 0$.

本文用三角模糊数来表示描述任务属性与员工状态的语言变量, 则各员工或团队对于某一任务的适配度, 可以通过该任务的属性和员工状态的模糊数运算得到, 得到的每个适配度仍是三角模糊数, 要比较三角模糊数则必须对三角模糊数进行解模糊. 解模糊的过程即是求最佳非模糊效用值(best non-fuzzy performance value, BNP). 使用中央区域(center of area, COA)法来找 BNP 是最简易可行的方法, 且不包含任意衡量值的偏好. 下面给出三角模糊数 $\tilde{A} = (l, m, r)$ 的 BNP 值:

$$BNP = l + (m - l + r - l)/3. \quad (2)$$

根据各候选人或候选团队适配度的 BNP 值, 可以得到各候选人或候选团队对于任务的适配度, 从而根据适配度排序把任务分配给最适合执行任务的候选人或团队.

2.2 各语言变量的三角模糊数

在任务分配中需要直接定义的语言变量分两类: 一是任务的属性, 包括任务技能要求权重、任务重要性、任务类型和任务配合类型; 二是员工的状态与属性, 主要指员工的技能水平. 语言变量具体定义结合应用领域进行设计. 下面, 分别根据各语言变量的“可能性”及“可理解性”, 给出这几个语言变量及其对应的三角模糊数定义, 对于每个因素, 本文都采用五级语言变量来表示其可能的值.

语言变量任务的技能要求及任务的重要性都具有 5 个可能的值, 分别为“不重要、些许重要、中等重要、比较重要、非常重要”, 对应的三角模糊数分别为 $(0, 0, 0.1)$, $(0, 0.3, 0.5)$, $(0.3, 0.5, 0.7)$, $(0.5, 0.7, 0.9)$, $(0.7, 0.9, 1)$. 相应的隶属函数如图 2 所示. 任务类型也有 5 个可能值, 分别为“耗时型、侧重耗时型、混合型、侧重经验型、经验型”, 对应的三角模糊数与技能要求对应的档位相同. 任务的配合

类型也有 5 个可能值, 分别为“简单合作、松散配合、一般配合、侧重配合、密切配合”, 对应的三角模糊数也与技能要求对应的档位相同.

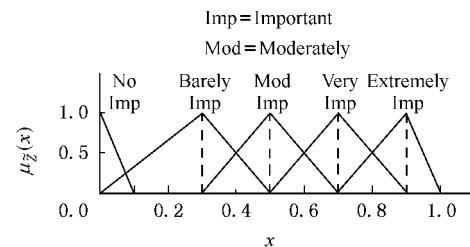


Fig. 2 The membership function of linguistic values capability.

图 2 技能要求语言变量的隶属函数

而语言变量员工的技能水平也有 5 个可能值, 分别为“空白、低水平、中等、高水平、专家级”, 对应的三角模糊数分别为 $(0, 0, 0)$, $(0, 0.2, 0.4)$, $(0.2, 0.4, 0.6)$, $(0.4, 0.6, 0.8)$, $(0.6, 0.8, 1)$. 其他属性或状态可由上述语言变量和相关数据计算得到, 具体计算公式如文章第 3.2 节所示. 由于员工的负载情况、成功率、弹性程度等是由相关数据计算得到, 其结果都表示成一个具有不确定性的数值, 为了评价这种带有不确定性的数值, 需要将这个数值转换成一个三角模糊数. 转换方法如下: 设属性 B 可以表示一个带有不确定性的数值 a , 则 B 所对应的三角模糊数可记为 $\tilde{B} = (a, a, a)$.

3 任务分配模型

3.1 任务分配多级模型

由上节的各语言变量定义可以得到模糊环境下影响任务分配各因素的三角模糊数表示. 下面根据各语言变量的三角模糊数给出任务分配问题的形式化定义.

定义 1. 记待分配的任务为 J_{new} , 完成 J_{new} 需要具有 k 种技能($C_t, t=1, \dots, k$), 且对技能 C_t 的要求为 $\tilde{w}(J_{\text{new}}, C_t)$, 任务重要性记为 \tilde{w}_{imp} , 任务类型记为 \tilde{w}_{kind} , 任务共需要由 $m (\geq 1)$ 个参与者组成的团队完成, 若 $m > 1$, 记任务的配合类型为 \tilde{w}_{coo} . 任务分配问题就是根据这些任务属性及候选人状态把任务分配给最适合的候选人.

记任务 J_{new} 的候选集合为 $U = \{U_1, \dots, U_n\}$, 根据第 1 节所分析的任务分配原则及任务分配问题定义, 可以得到图 3 所示的任务分配多级模型.

图 3 所示的多级模型可以用式(3)表示:

$$\tilde{S}(U_i, J_{\text{new}}) = \widetilde{WD}_{5 \times n}. \quad (3)$$

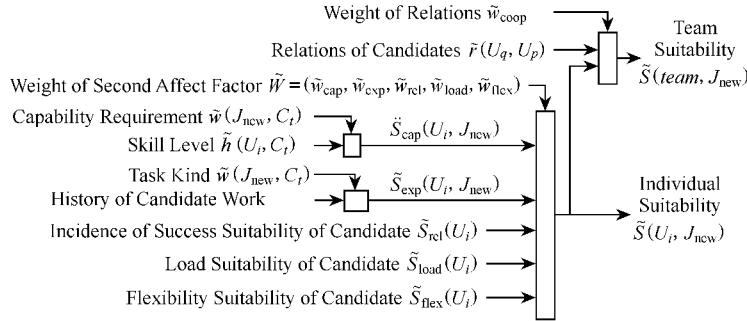


Fig. 3 Multilevel model of task assignment.

图 3 任务分配多级模型

式(3)中, $\tilde{W} = (\tilde{w}_{cap}, \tilde{w}_{exp}, \tilde{w}_{rel}, \tilde{w}_{load}, \tilde{w}_{flex})$ 是图 3 中 5 种二级影响因素的权重向量。矩阵 $\tilde{D}_{5 \times n}$ 表示各候选人的 5 项二级影响因素值。注意, 权重向量与矩阵 $\tilde{D}_{5 \times n}$ 中的每个元素都为一个三角模糊数。

如果任务只需要一个参与者, 只要对由式(3)得到的各候选人 U_i 的适配度 $\tilde{S}(U_i, J_{new})$ 按式(2)计算其 BNP, 根据各候选人适配度的 BNP, 把任务 J_{new} 分配给 BNP 最大的候选人。如果任务需要团队来完成, 即定义 1 中的 $m > 1$, 则必须对团队的适配度进行计算。根据参考文献[10], 可以通过对团队内成员两两间的关系进行评价从而得到整个团队的关系评价, 由此可以求得由 m 个人组成的团队对任务 J_{new} 的适配度为

$$\tilde{S}(team, J_{new}) = \tilde{w}_{coop} \otimes \frac{\sum_{q,p} \tilde{r}(U_q, U_p)}{m(m-1)/2} + \sum_{i=1}^m \tilde{S}(U_i, J_{new}). \quad (4)$$

n 个候选人总共可以组成 C_n^m 个候选团队, 按式(4)分别计算得到 C_n^m 个候选团队适配度 $\tilde{S}(team, J_{new})$ 的 BNP 值并进行比较, 把任务 J_{new} 分配给 BNP 值最大的候选团队就完成了任务分配。其中团队关系权重 $\tilde{w}_{coop} = \tilde{w}_{coo}$ 。

图 3 中的任务分配多级模型并不是铁板一块, 可以根据使用环境需要对模型进行简化, 只考虑 5 项二级因素中的某几项, 不影响模型在所选择影响因素上的综合匹配效果。

3.2 各影响因素的计算

由图 3 可以知道影响任务分配的主要因素包括技能、经验、成功率、负载和弹性。下面, 分别给出各适配度的计算方法。

候选人 U_i 的技能 C_t 水平记为 $\tilde{h}(U_i, C_t)$, 对于候选人 U_i 来说, 其对任务 J_{new} 的适配度 $\tilde{S}_{cap}(U_i, J_{new})$ 可由式(5)计算得到技能适配度 $\tilde{S}_{cap}(U_i, J_{new})$:

$$\tilde{S}_{cap}(U_i, J_{new}) = \frac{1}{k} \sum_{t=1}^k [\tilde{h}(U_i, C_t) \otimes \tilde{w}(J_{new}, C_t)]. \quad (5)$$

为了计算 U_i 相对任务 J_{new} 而言的经验值, 必须对 U_i 执行过的历数据和当前正在等待执行的数据进行统计, 考虑到很早前执行过的任务对 U_i 可能影响不大, 所以只对 U_i 最近一个时间段内的数据进行统计。设 J_j 是 U_i 在这时间段内执行的第 j 个任务(包括已分配给 U_i 但还没执行的任务), 任务 J_j 对技能 C_t 的要求记为 $\tilde{w}(J_j, C_t)$ 。当 J_j 对技能 C_t 的要求高于 J_{new} 对技能 C_t 的要求时, 可以认为就技能 C_t 而言, U_i 执行 J_j 的经验相对 J_{new} 而言是完全适用的, 所以经验值为 1。而 J_j 对技能 C_t 的要求低于 J_{new} 对技能 C_t 的要求时, 就技能 C_t 而言, U_i 执行 J_j 的经验相对 J_{new} 而言是由 $\tilde{w}(J_j, C_t)$ 和 $\tilde{w}(J_{new}, C_t)$ 对应的两个三角形交集部分的面积与并集部分的面积之比值决定的, 当两者交集为空时经验值为 0。综上所述, 记 $\tilde{w}(J_{new}, C_t) = (l_1, m_1, r_1)$, $\tilde{w}(J_j, C_t) = (l_2, m_2, r_2)$, 就技能 C_t 而言, U_i 执行 J_j 相对 J_{new} 的经验 $s(J_j, J_{new}, C_t)$ 可表示为

$$s(J_j, J_{new}, C_t) = \begin{cases} 1, & l_2 \leqslant l_1, \\ \frac{\int \min(\tilde{w}(J_j, C_t), \tilde{w}(J_{new}, C_t))}{\int \max(\tilde{w}(J_j, C_t), \tilde{w}(J_{new}, C_t))}, & \text{otherwise} \\ 0, & r_2 \leqslant l_1. \end{cases} \quad (6)$$

由此可得 U_i 执行 J_j 相对 J_{new} 的经验值 $\tilde{s}(J_j, J_{new})$ 为

$$\tilde{s}(J_j, J_{new}) = \frac{1}{k} \sum_{t=1}^k s(J_j, J_{new}, C_t). \quad (7)$$

若 U_i 在这时间段内共执行过 z 个任务, 则 U_i 相对于 J_{new} 的经验值 $\tilde{S}'_{exp}(U_i)$ 为

$$\tilde{S}_{\text{exp}}'(U_i) = \frac{1}{z} \otimes \sum_{j=1}^z s(J_j J_{\text{new}}). \quad (8)$$

对 n 个候选人的经验值进行归一化处理后, 得到候选人 U_i 的经验适配度 $\tilde{S}_{\text{exp}}(U_i J_{\text{new}})$ 为

$$\tilde{S}_{\text{exp}}(U_i J_{\text{new}}) = \tilde{S}_{\text{exp}}'(U_i) / \sum_{i=1}^n \tilde{S}_{\text{exp}}'(U_i). \quad (9)$$

分配任务给候选人时需要考虑候选人成功率, 记候选人 U_i 总共执行过 tc_i 次这类任务, 而成功次数为

sc_i 次, 各候选人平均成功率为 $\bar{s} = \frac{1}{n} \times \sum_{i=1}^n s_i$, 平均执

行次数为 $\bar{tc} = \frac{1}{n} \times \sum_{i=1}^n tc_i$. 则成功率适配度 $\tilde{S}_{\text{re}}(U_i)$ 计算采用式(10):

$$\tilde{S}_{\text{re}}(U_i) = \begin{cases} s_i, & tc_i > \bar{tc}, \\ (\min(\bar{s}, s_i) s_i \max(s_i, \bar{s})), & tc_i < \bar{tc}. \end{cases} \quad (10)$$

同样任务对于不同候选人而言所需要时间不同, 以任务数量来衡量候选人负载会导致伪负载平衡, 必须以候选人的执行时间为负载平衡指标。对于待执行任务 J_j 而言, 其所需要的执行时间由其执行过的这类任务的平均执行时间来计算。若 U_i 共有 l 个待执行任务, 则 U_i 当前绝对负载为

$$\tilde{L}_{\text{load}}(U_i) = \sum_{j=1}^l \tilde{T}_{\text{aver}}(U_i J_j), \quad (11)$$

其中 $\tilde{T}_{\text{aver}}(U_i J_j)$ 表示 U_i 执行与 J_j 同类任务的平均处理时间, 该时间由历史数据得到。 $\tilde{L}_{\text{load}}(U_i)$ 是一个绝对负载, 对绝对负载进行归一化处理得到 U_i 的相对负载 $\tilde{S}_{\text{load}}(U_i)$ 。 $\tilde{S}_{\text{load}}(U_i)$ 越大表示 U_i 的负载越轻:

$$\begin{aligned} \tilde{S}_{\text{load}}(U_i) &= \\ &\left\{ \sum_{j=1}^n \tilde{L}(U_j) - \tilde{L}_{\text{load}}(U_i) \right\} / \sum_{j=1}^n \tilde{L}_{\text{load}}(U_j). \end{aligned} \quad (12)$$

若候选人 U_i 所担任的角色数目为 $r(U_i)$, 由于 U_i 的弹性适配度越大任务分配给他的可能性越小, 类似求 $\tilde{S}_{\text{load}}(U_i)$, U_i 的弹性适配度 $\tilde{S}_{\text{flex}}(U_i)$ 可由式(13)计算得到, $\tilde{S}_{\text{flex}}(U_i)$ 越大表示 U_i 的资源弹性越差:

$$\tilde{S}_{\text{flex}}(U_i) = \left\{ \sum_{j=1}^n r(U_j) - r(U_i) \right\} / \sum_{j=1}^n r(U_j). \quad (13)$$

3.3 各影响因素的权重计算

由式(5)~(13)可以分别求得各候选人的能力、经验、成功率、负载指标和弹性适配度等, 仅仅由这些适配度指标还无法完成任务分配, 还要考虑这些

适配度的权重。

技能、经验和成功率适配度权重比较简单, 由于越重要的任务越注重技能、成功率, 所以技能及成功率指标的权重可以由任务的重要性来决定, 而由任务类型定义可知, 经验指标的权重就由任务类型决定。即 $\tilde{w}_{\text{capability}} = \tilde{w}_{\text{imp}}$, $\tilde{w}_{\text{reliability}} = \tilde{w}_{\text{imp}}$, $\tilde{w}_{\text{experience}} = \tilde{w}_{\text{kind}}$ 。

负载适配度权重比较复杂, 如果负载指标的权重太大则可能导致能力、经验和成功率指标被淹没, 而如果负载指标的权重太小, 当在某段时间内到达的任务对能力要求较高, 任务也比较重要时, 就会导导致能力强的候选人负载远远高于能力弱的候选人, 结果导致每个任务的等待时间过长, 使得任务的平均完成时间急剧上升。负载权重需要考虑两方面因素, 一方面, 确定负载权重实际上就是给出一个精确值或一个模糊数来衡量各候选人负载的离散程度, 当负载不平衡比较大时, 负载权重应该大一些, 以平衡各候选人负载; 另一方面, 当任务比较重要时, 允许负载不平衡程度大一些, 即负载权重应该小一些。基于此, 本文提出了一种动态负载权重。动态负载权重随着候选人负载的波动加大而增加, 随着任务的重要性升高而减小。如果候选人集合为 $U = \{U_1, \dots, U_n\}$, 候选人 U_i 的绝对负载 $\tilde{L}_{\text{load}}(U_i)$ 可由式(11)得到。因此, 负载的权重 \tilde{w}_{load} 可以由式(14)计算得到:

$$\tilde{w}_{\text{load}} = \frac{\tilde{w}_{\text{imp}}}{n} \times \sum_{i=1}^n |\tilde{L}_{\text{load}}(U_i) - \frac{1}{n} \times \sum_{i=1}^n \tilde{L}_{\text{load}}(U_i)|. \quad (14)$$

此时, 负载权重 \tilde{w}_{load} 将随着负载不均衡的增加而加大, 从而增加式(12)计算得到的相对负载在最终适配度中的比重, 从而自动平衡各候选人负载。如果负载的计量单位比较大, 比如“小时”, 由式(12)得到的相对负载 $\tilde{S}_{\text{load}}(U_i)$ 本身就比较小, 当候选人间负载相差几十个单位, 影响也比较大, 此时, 可以在式(14)计算得到的权重 \tilde{w}_{load} 基础上适当加大负载权重, 反之亦然。

对于弹性适配度而言, 如果负载不平衡程度比较大或任务比较重要时, 则首先应该考虑尽量让负载平衡, 以减小各任务的平均等待时间, 当负载比较平衡时或任务不怎么重要时, 可以适当加大弹性适配度的权重 \tilde{w}_{flex} , 以保持资源的弹性。基于此, 得到弹性适配度的权重计算公式:

$$\tilde{w}_{\text{flex}} = \min(1/\tilde{w}_{\text{load}}, 1). \quad (15)$$

4 算法结果分析

为了评价本文提出了任务分配模型性能,我们在一个工作流原型系统上实现了 MMTA,MCTA 以及 Round-Robin 任务分配算法,并对这 3 种算法的性能进行比较分析。

仿真环境设置如下:工作流模式如图 4 所示,此测试工作流模型在或分支 C_1 处选择活动 T_2 的概率为 0.3,选择活动 T_3 的概率为 0.7。流程实例到达速率为“1 个/时间单位”,到达任务选择 T_2 或 T_3 是完全随机的,各参与者对分配到任务的处理顺序是按先进先出(FIFO)原则进行,不进行任何优化调度。

Table 1 Capability Requirements of Tasks and Skill Level of Candidates
表 1 各活动的技能要求与各候选人的技能水平

Capability Requirements	Task T_2	Task T_3	Level of Skill	Candidate A	Candidate B	Candidate C
write	extremely	no	write	normal	good	expert
compute	barely	extremely	compute	expert	good	low
project	moderately	extremely	project	expert	normal	normal
design	very	barely	design	good	normal	normal

Table 2 The Incidence of Success and Time-Consuming of Candidates for Tasks
表 2 各候选人处理任务的成功率与时间

Candidates	Time-Consuming		Incidence of Success	
	Task T_2	Task T_3	Task T_2	Task T_3
Candidate A	5	3	0.93	0.99
Candidate B	6	7	0.92	0.90
Candidate C	7	8	0.99	0.60

根据上述测试环境设置,MMTA,MCTA 以及 Round-Robin 运行负载结果如图 5 所示。图中所示负载均表示各候选人当前绝对负载。而各算法的成功率、平均处理时间等如表 3 所示:

Table 3 The Performance Index of Three Algorithms

表 3 各算法的性能指标

Algorithms	Overall Incidence of Success	Incidence of Success for Importance Task	Average Complete Time of Tasks
MMTA	0.977	0.980	315.00
MCTA	0.977	0.989	390.04
Round-Robin	0.879	0.851	501.87

Round-Robin 算法的负载非常不平衡。相对 T_2,T_3 而言,C 的能力最差,A 的能力最高,B 的能力中等。虽然每个参与者都分配相同任务数,但能力弱的候选人处理速度慢,其负载一直高于能力强

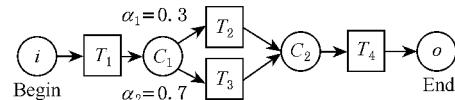


Fig. 4 The workflow model of test.

图 4 测试工作流模型

活动 T_2,T_3 的参与者候选人都为 A,B,C。 T_2 和 T_3 的技能要求与 3 个候选的技能水平如表 1 所示。 T_2 重要级别为中等重要, T_3 的重要级别为非常重要。因缺少经验值数值,本测试例子没有考虑经验值和弹性程度的影响。3 个候选人处理 T_2,T_3 的时间与成功率如表 2 所示。

Table 1 Capability Requirements of Tasks and Skill Level of Candidates

表 1 各活动的技能要求与各候选人的技能水平

的候选人,这种差别随着到达任务的增加而不断加大。MCTA 算法的负载更加不平衡,虽然该算法是综合能力与负载的情况来分配任务,但从实际效果来看并没起到负载平衡作用。 T_3 由 C 来执行效果最好,因此 T_3 几乎 100% 被分配给 C 执行,由于 T_3 在总任务中大约占 70%,所以 C 的负载远远超出 A 和 B,这种不平衡还将随着 T_3 所占比例增加而增加。MMTA 算法在绝大多数时间内,各候选人的负载都相差不大,基本交叉在一起。其中能力较高的候选人 A 的负载会略高于其他两个候选人的负载。这是因为适合 A 执行的任务 T_3 所占比例比较大,所以分配给他的任务比其他两个会略多一些。

再比较 3 种算法的其他性能,从表 3 中可以看出,Round-Robin 效果最差,不仅平均完成时间长,而且任务总成功与重要任务 T_3 的成功率都远低于其他两种方法。而 MCTA 算法的成功率略高于 MMTA 方法,但其平均完成时间大于 MMTA 方法。MCTA 和 Round-Robin 算法平均完成时间长的原因是因为负载不平衡而引起任务的等待时间增加。MCTA 算法因为总把任务分配给技能最适合的参与者来完成,当技能水平与成功率成正比时,MCTA 算法的成功率要高于其他两种方法。而 MMTA 算法综合考虑了技能、负载与成功率等因素,具有良好的综合效果。

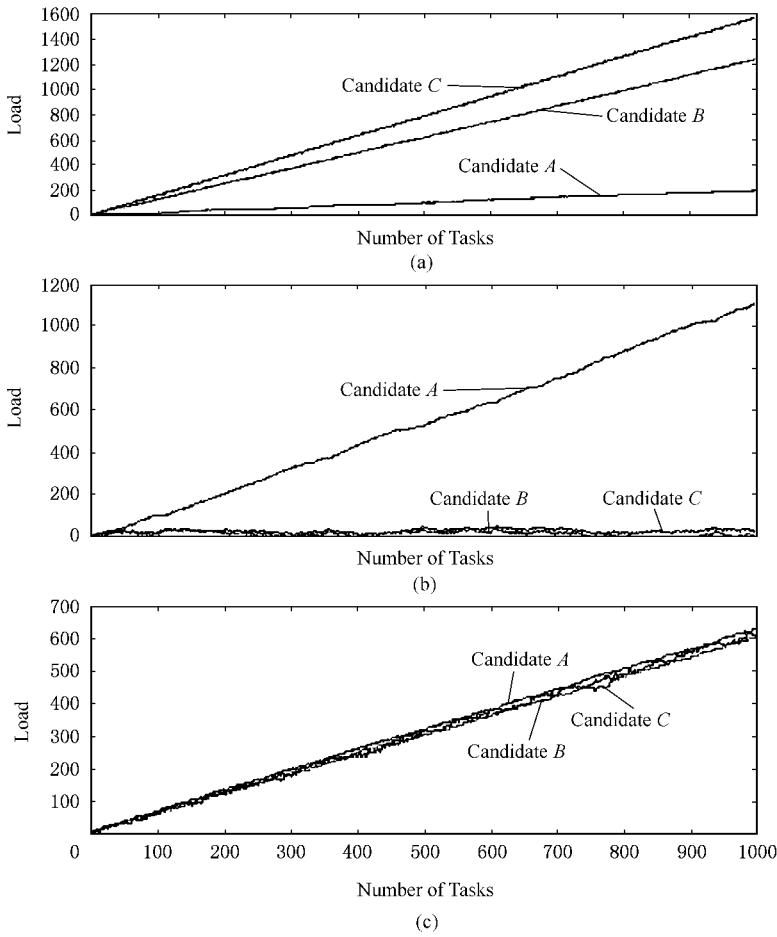


Fig. 5 The candidates load of the three task assignment algorithms. (a) The candidates load of the task assignment algorithm Round-Robin ;(b) The candidates load of the task assignment algorithm MCTA ; and (c) The candidates load of the task assignment algorithm MMTA .

图5 三种算法的候选人负载曲线。(a) Round-Robin 算法的参与者负载曲线;(b) MCTA 算法的参与者负载曲线;(c) MMTA 算法的参与者负载曲线

我们重复修改测试环境配置,测试3种算法效果,发现Round-Robin与MCTA的算法性能完全取决于测试环境中的资源配置。所以这两种方法只适用于资源配置非常合理的情况,而MMTA无论在什么样的资源配置环境下都具有非常好的综合性能。

分配过程、平衡资源负载,提高任务的成功率。虽然该任务分配模型比较复杂,但并不是铁板一块,可以根据使用环境需要对模型进行简化,简化后的模型依然具有良好的综合效果。

参 考 文 献

- [1] Workflow Management Coalition. Process definition interface—XML process definition language [R]. Tech Rep : TC1025 , 2002
- [2] Y-N Huang , M-C Shan. Policies in a resource manager of workflow systems : Modelling , enforcement and management [R]. Hewlett Packard , Tech Rep : HPL-98-156 , 1998
- [3] Prasant Mohapatra. Dynamic real-time task scheduling on hypercubes [J]. Journal Algorithms , 1997 , 46(1):91–100
- [4] Li-jie Jin , Fabio Casati , Mehmet Sayal , et al. Load balancing in distributed workflow management system [C]. The ACM Symp on Applied computing , Las Vegas , Nevada , USA , 2001

5 总 结

实施工作流管理的重要目标之一是把组成业务过程的各任务分配给合适的人或设备来完成。本文提出了一个基于模糊环境和多因素的任务分配模型,该模型对影响任务分配的各种具有不确定性的因素进行综合考虑,从而计算得到各候选人执行待分配任务的适配程度并根据适配度排序,完成任务的自动分配。该任务分配方法在各种资源配置环境下都具有非常好的综合性能,可以有效地优化任务

- [5] W M P van der Aalst , A Kumar. A reference model for team-enabled workflow management systems [J]. Data and Knowledge Engineering , 2001 , 38(3): 335–363
- [6] Zhang Xiaoguang , Cao Jian , Zhang Shensheng. Team-and-role-enabled task distribution under policy constraints in adaptive WFMS [J]. Journal of Computer Research and Development , 2002 , 39(12): 1156–1163 (in Chinese)
(张晓光 , 曹健 , 张申生 . 策略约束面向角色和团队关系的工作流任务分配管理 [J]. 计算机研究与发展 , 2002 , 39(12): 1156–1163)
- [7] Tang Da , Yang Yuan-sheng. Research on workflow participant mechanism and dynamic characteristic based on hierarchically refined petri nets [J]. Journal of Computer Research and Development , 2004 , 41(9): 1545–1553 (in Chinese)
(唐达 , 杨元生 . 基于层次细化 Petri 网的工作流参与者机制与动态特性研究 [J]. 计算机研究与发展 , 2004 , 41(19): 1545–1553)
- [8] Gin-Shuh Liang , Mao-Jiun J Wang. Personnel placement in a fuzzy environment [J]. Computers and Operations Research , 1992 , 19(2): 107–121
- [9] SB Yaakob , S Kawata. Worker ' s placement in an industrial environment [J]. Fuzzy Set and Systems , 1999 , 106(3): 289–297
- [10] Minxin Shen , Gwo-Hshiung Tzeng , Duen-Ren Liu. Multi-Criteria task assignment in workflow management systems [C]. Hawaii Int'l Conf on System Sciences , Hawaii , USA , 2003
- [11] L Zadeh. The concept of a linguistic variable and its application to approximate reasoning [J]. Information Sciences , 1975 , 8(3): 43–80

Research Background

The primary purpose of workflow application involves realizing the business process executed automatically and efficiently by scheduling and assigning the resource and information reasonably. For that reason , the automatic optimize assignment of user task is one of the important functions of workflow management system (WfMS) , and the key of improving the efficiency of WfMS as well. Traditional task assignment only considers with respect to the matching of role rather than the effect of other factors , such as importance of task , skill level of worker , etc. Once the number of the qualified workers is more than one , it can ' t evaluate the suitability of the candidates. Furthermore , traditional study on human resources makes use of precise value to evaluate the attributes and the state of tasks and candidates which ignore fuzziness of the parameters. Therefore , based on the task assignment method presented by Minxin , this paper analyzes in detail all the influencing factors and their linguistic fuzziness of user task assignment and puts forward a multilevel model of the task assignment that can be combined and the corresponding task assignment measure. After then , the design method of the weight with self-adapt ability of the affect factors also are discussed in the paper. The comparison in the experiment proves that the multilevel model has very good comprehensive performance under the circumstances of different resource configurations.

- [12] D Dubois , H Prade. Operating on fuzzy numbers [J]. International Journal of Systems Science , 1978 , 9(6): 613–629



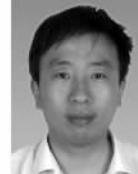
Xiao Zhengjin , born in 1976. Ph. D. His main research interests include Workflow , artificial intelligence , etc.

肖郑进 , 1976 年生 , 博士 , 主要研究方向为工作流、人工智能等。



He Qinming , born in 1965 , Professor and Ph. D. supervisor in Zhejiang University , Hangzhou , China. His main research interests include machine learning , artificial intelligence , Web information retrieval , etc .

何钦铭 , 1965 年生 , 教授 , 博士生导师 , 主要研究方向为机器学习、人工智能、网络信息搜索等。



Chen Qi , born in 1963. Associate professor in Zhejiang University , Hangzhou , China. His main research interests include intelligent decision support system , GIS , framework for software architecture , etc .

陈奇 , 1963 年生 , 副教授 , 主要研究方向为智能决策支持系统、GIS、软件架构技术等。