

基于 Agent 能力自信度的任务分配问题研究

曾广周 杨公平 王晓琳

(山东大学计算机科学与技术学院 济南 250061)

摘 要 借助结盟的任务分配机制是多 Agent 系统研究中的一个热点问题. 文中通过任务需求和 Agent 能力的 ECA 规则表示, 定义了一种 Agent 能力自信度, 并给出了基于合同网协议和 Agent 能力自信度的任务分配算法和应用示例, 其中, 任务需求 ECA 规则中的 C 项表示 Agent 执行该任务时必须做出的付出, 能力 ECA 规则中的 C 项表示 Agent 可以为该任务做出的承诺, 付出/承诺关系揭示了 Agent 的能力自信度, ECA 规则使得 Agent 能力自信度具有良好的可计算性.

关键词 多 Agent 系统; 联盟; 任务分配; Agent 能力自信度; ECA 规则; 合同网协议

中图法分类号 TP18

Study of Task Allocation Problem Based on Agent Ability Confidence

ZENG Guang-Zhou YANG Gong-Ping WANG Xiao-Lin

(School of Computer Science and Technology, Shandong University, Jinan 250061)

Abstract Task allocation mechanism via coalition formation is one of key problems in the multi-Agent systems. A kind of Agent ability confidence is defined using ECA rules for both task requirements and Agent abilities, and the task allocation algorithm based on the Agent ability confidence and contract net protocol is given in this paper. Owing to the condition items within the task requirement ECA rule indicate the payoff required by that task while an Agent going to execute it, the condition items within the Agent ability ECA rule represent the Agent commitment for that task when the Agent intends to compete it, not only the ratio between payoff and commitment can reveal Agent confidence state while he faced the task to be competing, but also the Agent ability confidence defined on the ECA rules is easy to calculate.

Keywords multi-Agent system; coalition formation; task allocation; Agent ability confidence; ECA rule; contract net protocol

1 引 言

在多 Agent 竞争问题求解系统中, 当某个 Agent 发现自己不能独立地对一项任务做出承诺, 并且该任务本身又可以分解为多个子任务时, 它可以与其它 Agent 结成利益联盟, 并在结盟过程中完成各子任务在联盟成员之间的分配. 因此, 借助结盟的任务

分配机制一直是多 Agent 问题求解系统研究中的一个热点问题^[1-2].

针对联盟利益的多 Agent 结盟研究大多采用效用模型(utility-based model)^[3], 并且通常作为一个协同博弈(cooperative game)问题求解^[4-5]. 效用模型较好地解决了联盟利益最大化以及联盟利益在成员个体之间的均衡分配问题, 前提是假定每个成员都具备独立完成某项任务的能力, 并且每个成员都

相信其它成员也能如同自己一样,一致合作地追求最大联盟利益,而没有评价成员信任对结盟的影响。

为了研究成员信任在结盟中的作用,文献[6]把信任(trust)定义为一个 Agent 如何看待另一个 Agent 履行承诺的程度. 文献[7]则引入置信(confidence)和信用(reputation)两个概念,其中,置信用于度量任务求解 Agent 履行承诺的程度,信用用于评价任务求解 Agent 与社会中其它 Agent 之间的交互历史. 文献[8]将 Agent 的行为自信度定义为一个 Agent 能够被其它 Agent 所信任并将动作 k 委托给它执行的程度,实质上仍然是一个信任问题. 由于信任的评价数据来自于协作结果,因此,信任度可以容易地随着 Agent 之间合作经历的增加而有所改变. 信任评价有助于规避新一轮合作伙伴选择时的风险,但不能具体指导任务分配。

文献[9]认为:如果 Agent 相信自己在执行动作时将获得成功,那么它是自信的,但没有说明如何评价自信度. 文献[10]进一步指出,一个自信的 Agent,必定是:如果知道自己能够做什么,那么它相信自己一定能够做;如果知道自己能够完成什么,那么它将尽力去完成,但没有说明 Agent 如何知道自己“能够做什么”和“能够完成什么”. 文献[11]把置信定义为 Agent 相信事件 e 将在未来的某个时间出现,并且区分为:如果事件 e 的出现是 Agent 自身动作的结果,则称 Agent 关于事件 e 自信(self confidence);如果事件 e 的出现是其它 Agent 导致的结果,比如说 j ,则 Agent 相信 j 最终会使 e 为真. 文献研究了基于时态逻辑的置信演算,但没有说明如何计量 Agent 的置信水平。

本文给出一种基于任务需求 ECA 规则和 Agent 能力 ECA 规则的自信度计算方法,并讨论采用合同网协议^[12]和 Agent 能力自信度的任务分配机制. 沿用文献[10]和文献[11]的观点,本文把 Agent 自信理解为评价自己“能够做什么”和“能够完成什么”的程度,把任务管理 Agent 置信理解为它相信问题承担 Agent 会完成指派的任务,假设所有 Agent 都是诚实的。

本文第 2 节依据合同网协议,给出任务分配问题的定义;第 3 节讨论 Agent 的能力自信度计算;第 4 节给出基于 Agent 能力自信度的任务分配算法,并讨论 Agent 联盟自信度计算问题;第 5 节简述一个应用的例子;第 6 节为相关工作的比较,最后为小结,进一步阐述本文的工作。

2 任务分配问题的定义

定义 1. 多 Agent 任务分配系统是一个合同网结构,记作 (TS, AS, SM, Ψ) ,其中 TS 是系统中独立任务的集合, AS 是系统中问题求解 Agent 的集合, $SM \notin AS$ 是系统管理 Agent,它负责 $T \in TS$ 在多个 Agent 联盟 $AC \subseteq AS$ 之间的分配, Ψ 是 SM 采用的任务指派标准。

不同的任务分配模型可以采用不同的任务指派标准,例如效用模型中的最大联盟值标准^[1-5]. 本文采用 Agent 能力自信度标准, Agent 能力自信度将在第 3 节定义,任务分配算法将在第 4 节给出。

记系统管理者 SM 关于事件 ϕ 置信为 $Bel_{SM}\phi$. 根据合同网模型和定义 1 中独立任务的假设,采用文献[11]的观点,不加证明地有下面引理。

引理 1. $\forall T \in TS, Bel_{SM}T \Leftrightarrow Bel_{SM}exe(T, AC) \mid AC \subseteq AS$, 其中 $exe(T, AC)$ 表示联盟 AC 执行 T 。

引理 1 的解释是 SM 相信 T 会被完成等价于它相信联盟 AC 能够执行 T 成功。

定义 2. Agent 联盟 AC 是一个合同网结构 (T, AG, CM, φ) , 其中, $T \in TS$ 是 AC 的任务, $AG \subseteq AS$ 是 AC 中任务求解 Agent 的集合, $CM \in AS \wedge CM \notin AG$ 是联盟 AC 中 T 的管理者,它借助结盟过程完成 T 的各子任务在 AG 成员之间的分配, φ 是 CM 采用的基于 Agent 能力自信度的任务指派标准。

设任一任务 $T \in TS$ 的结构都可以归约为一棵“与-或”树(见图 1),记作 $and-or-tree(T)$,满足:树的根节点表示任务 T ,树的叶子节点表示可以被单个 Agent 独立完成任务(称作原子任务),并且约定:如果树上某个节点有儿子任务,那么其儿子任务之间的执行逻辑要么全“与”,要么全“或”^①,并分别记作 $and-F(t_1, t_2, \dots, t_p)$ 或 $or-F(t_1, t_2, \dots, t_q)$,其中 F 是节点标识, t_i 是 F 的儿子节点,例如图 1 中的 $and-T(1, 2, 3)$, $and-5(8, 9)$, $or-1(4, 5)$, $or-7(13, 14)$ 等. 根据上述约定,容易推得性质 1。

性质 1. $and-or-tree(T)$ 可以等价地用一个 ECA 规则集合表示。

例如图 1 所示 $and-or-tree(T)$ 的部分 ECA 规则可以表示为

On done(1) \wedge done(2) \wedge done(3) if union(1, 2, 3) = TRUE then done(T)

① 当某节点的儿子任务之间不满足全“全”或全“执行逻辑”时,可以通过引入新的子任务节点使其满足。

On $\text{done}(4) \vee \text{done}(5)$ if $\text{choice}(4,5) = \text{TRUE}$
 then $\text{done}(1)$
 On $\text{done}(8) \wedge \text{done}(9)$ if $\text{union}(8,9) = \text{TRUE}$
 then $\text{done}(5)$

其中, done (完成), union (同步合并)和 choice (选择)都是一阶谓词。

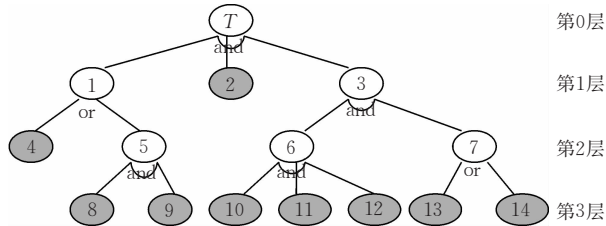


图 1 任务 T 的“与-或”树表示, 树的高度=3, 层数=4

任务可以按照不同的粒度分配. 本文约定: 联盟管理者 CM 负责规划 $\text{and-or-tree}(T)$ 结构并描述相应的 ECA 规则, 并且仅对 $\text{and-or-tree}(T)$ 上所有原子任务进行分配, 而树上其它非原子任务的执行与协调, 例如同步“与”任务、选择“或”任务等, 均由 CM 完成。

一般地说, 对于形如图 1 所示的 $\text{and-or-tree}(T)$, 只有“与”分组中的所有原子任务都得到分配, 其父任务才算得到分配; 只要“或”分组中的任何一个原子任务得到分配, 其父任务就算得到分配. 不过, 如果只分配“或”分组中的某一个原子任务, 那么, 在任务执行阶段将失去可选择性. 因此, 本研究对“或”分组的原子任务, 也采用全部分配的策略。

基于上述说明, 下面给出联盟 AC 的任务分配问题定义。

定义 3. 记 $\text{and-or-tree}(T)$ 上所有原子任务的集合为 AT . 设 AG 初始化为空集并且每个问题求解 Agent 只能指派一项任务, 则联盟 AC 的任务分配问题可以定义为:

$\forall t \in AT$ 如果 $\exists a \in AS$ 使得 $\text{alloc}(t, a)$ 满足 φ 则 $AG \leftarrow AG \cup \{a\}$, 其中, 一阶谓词 $\text{alloc}(t, a)$ 表示将 t 指派给 a 为真, φ 是基于 Agent 能力自信度的任务指派标准。

记联盟管理者 CM 关于事件 ϕ 置信为 $\text{Bel}_{CM}\phi$.

引理 2. $\text{Bel}_{CM}T \Leftrightarrow \forall t \in AT \text{Bel}_{CM}exe(t, a) | a \in AG \wedge \forall (t \in T \wedge t \notin AT) \text{Bel}_{CM}coop(t, CM)$, 其中, $exe(t, a)$ 表示 a 是原子任务 t 的执行者, $coop(t, CM)$ 表示 CM 是所有非原子任务 t 的协调者与执行者。

证明. 根据合同网模型, $\forall t \in AT \text{Bel}_{CM}exe(t, a) | a \in AG$ 为真意味着 CM 把任务 t 指派给 a 并且

相信 a 会把 t 执行成功. 又根据性质 1, 通过设定 ECA 规则推理机就可以使 CM 具备协调与执行所有非原子任务的能力, 即 $\forall (t \in T \wedge t \notin AT) \text{Bel}_{CM}coop(t, CM)$ 为真. 既然 CM 相信所有问题求解 Agent 都会把各自分配的原子任务执行成功, 又相信自己具备协调和执行各非原子任务的能力, 由此可以推断 $\text{Bel}_{CM}T$ 成立, 即 CM 相信 T 将被执行成功. 证毕.

3 Agent 的能力自信度

$\forall t \in AT$, 设其流程结构可以表示为一个动作网络 $\text{Act_Net} = (A_node, C_node, E_arc)$, 其中, $A_node = \{act_1, act_2, \dots, act_n\}$ 是动作节点的集合, 动作是用一阶谓词表示的操作, 例如 $\text{Print}(o, s)$, 其中 o 是打印对象, s 是打印对象的属性; $C_node = \{junc_1, junc_2, \dots, junc_m\}$ 是连接节点的集合, 连接节点用于标识相继动作之间的逻辑关系, 例如顺序、与分叉、与合并、或分叉、或合并等; $E_arc \in A_node \times C_node \cup C_node \times A_node$ 是有向弧关系. 图 2 是一个动作网络的例子, 其中, 顺序型连接节点已被省去。

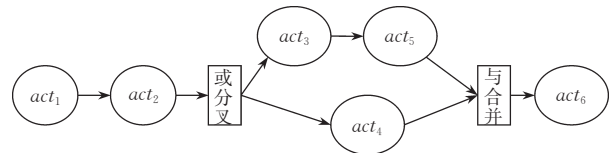


图 2 一个动作网络的例子

设任一动作 $act \in A_node$ 的执行都可以通过一条 ECA 规则表示, 例如 On 文档 DOC 编辑完成 if 打印机空闲 then $\text{Print}(\text{DOC}, 2 \text{ 份})$. 记动作需求 ECA 规则为

$\text{Required}(act): \text{on } E(e_1, e_2, \dots, e_p) \text{ if}$
 $C(c_1, c_2, \dots, c_q) \text{ then } act(o, s),$

其中, o 是 act 的操作对象, s 是 o 的属性; $E(e_1, e_2, \dots, e_p)$ 是用一阶谓词表示的触发事件, e_i 表示事件参数, $i=1, 2, \dots, p$; $C(c_1, c_2, \dots, c_q)$ 是条件表达式, c_j 给出 act 的执行条件和许可值, 例如时间、数据、货币或其它资源, $j=1, 2, \dots, q$, 不失一般地, 可以记作

$C(c_1, c_2, \dots, c_q) = \text{time} \leq \tau \wedge rs_1 \leq \lambda_1 \wedge \dots \wedge rs_{q-1} \leq \lambda_{q-1}$ (1)

这里, $\text{time} \leq \tau$ 表示 act 必须在时间段 τ 内完成, $rs_j \leq \lambda_j$ 表示执行 act 的第 j 类资源不能超过许可值 λ_j , $j=1, 2, \dots, q-1$.

因为当用 ECA 规则描述动作时, 动作之间的

连接逻辑蕴涵在 ECA 规则的执行顺序中,所以可以不加证明地给出引理 3.

引理 3. $Act_Net \Leftrightarrow \{Required(act)\} \mid \forall act \in A_node.$

记 Agent a 关于事件 ϕ 自信为 $Bel_a \phi$. 根据引理 3, 有下面引理 4.

引理 4. 给定任一原子任务 $t \in AT$ 的动作网络 Act_Net 及其动作 ECA 规则集合,

$Bel_a t \Leftrightarrow \forall act \in A_node Bel_a satisfy(Required(act))$, 其中, $satisfy$ 为可满足算子.

Agent a 关于 act 的可满足性可以通过动作 ECA 规则判定. 记 $a \in AS$ 执行 $act \in A_node$ 的能力 ECA 规则为

$CanDo(act): on E(e_1, e_2, \dots, e_p) \text{ if } C(c_1, c_2, \dots, c_q)$
then $act(o, s)$,

其中, $E(e_1, e_2, \dots, e_p)$ 和 $C(c_1, c_2, \dots, c_q)$ 的含义与 $Required(act)$ 中的解释相同, 不过, 这里 c_j 表示 a 对 act 执行条件的承诺, $j=1, 2, \dots, q$.

定义 4. 设 $act_a(o_a, s_a)$ 和 $act_t(o_t, s_t)$ 分别是 $CanDo(act_a)$ 和 $Required(act_t)$ 中的两个动作, $act_a, act_t \in A_node$. 如果 $o_a = o_t \wedge s_a = s_t$, 则称 $act_a(o_a, s_a)$ 和 $act_t(o_t, s_t)$ 功能等价, 简记为 act .

定义 5. 设 $E_a(e_1, e_2, \dots, e_p)$ 和 $E_t(e_1, e_2, \dots, e_p)$ 分别是 $CanDo(act)$ 和 $Required(act)$ 中的两个事件谓词, $act \in A_node$. 称 $E_a(e_1, e_2, \dots, e_p)$ 与 $E_t(e_1, e_2, \dots, e_p)$ 等价, 记作 $E_a \Leftrightarrow E_t$, 当且仅当 $E_a(e_1, e_2, \dots, e_p)$ 和 $E_t(e_1, e_2, \dots, e_p)$ 之间满足下述条件之一:

(1) $E_a(e_1, e_2, \dots, e_p)$ 和 $E_t(e_1, e_2, \dots, e_p)$ 是同一个谓词公式;

(2) $E_a(e_1, e_2, \dots, e_p)$ 和 $E_t(e_1, e_2, \dots, e_p)$ 是通过替代-置换达到合一的谓词公式.

定义 6. 设 $C_a(c_1, c_2, \dots, c_q)$ 和 $C_t(c_1, c_2, \dots, c_q)$ 分别是 $CanDo(act)$ 和 $Required(act)$ 中的两个条件表达式, $act \in A_node$. 称 $C_a(c_1, c_2, \dots, c_q)$ 蕴含 $C_t(c_1, c_2, \dots, c_q)$, 记作 $C_a \Rightarrow C_t$, 当且仅当 $\tau_a \leq \tau_t \wedge \forall k(\lambda_a^k \leq \lambda_t^k)$, 其中, τ_t 表示动作 act 给定的时间许可, τ_a 表示 a 做出的时间承诺, λ_t^k 表示动作 act 的第 k 类资源许可, λ_a^k 表示 a 对第 k 类资源做出的承诺, $k=1, 2, \dots, q-1$.

定义 7. 给定 $CanDo(act)$ 和 $Required(act)$, $act \in A_node$. 称 $CanDo(act)$ 可以按照 $Required(act)$ 实例化, 当且仅当 $E_a \Leftrightarrow E_t \wedge C_a \Rightarrow C_t$.

实例化后的能力 ECA 规则记作 $CanDone(act)$.

推论 1. $CanDone(act) \Rightarrow Bel_a act$.

证明. 根据定义 7, $CanDone(act) \Rightarrow a$ 不仅知道自己能够完成 $Required(act)$, 而且知道自己能

够以低于许可的代价完成 $Required(act)$. 沿用文献[9-10]中的观点: 如果 Agent 知道自己能够做什么, 那么它相信自己一定能够做, 于是推论 1 成立.

证毕.

定义 8. 若 $Bel_a act$, 则称 $\sigma_a^{act} = \omega_1 \tau + \omega_2 \lambda$ 为 a 关于动作 act 的能力自信度, 式中

$$\tau = \tau_t / \tau_a \quad (2)$$

$$\lambda = \sum_{k=1}^{q-1} \mu_k (\lambda_t^k / \lambda_a^k) \quad (3)$$

这里, τ_t 是关于 act 的时间许可, τ_a 是 a 对 τ_t 的承诺, λ_t^k 是关于 act 的第 k 类资源的许可, λ_a^k 是 a 对 λ_t^k 的承诺, μ_k 是相应资源的权重, $\sum_{k=1}^{q-1} \mu_k = 1$; ω_1 和 ω_2 按照时间和资源在 act 中的重要性确定分配比例, $\omega_1 + \omega_2 = 1$.

性质 2. $Bel_a act \Rightarrow \sigma_a^{act} \geq 1$.

证明. 根据定义 7 和定义 6, $CanDone(act)$ 当且仅当 $\tau_a \leq \tau_t \wedge \forall k \lambda_a^k \leq \lambda_t^k, k=1, 2, \dots, q-1$, 于是有 $CanDone(act) \Rightarrow \sigma_a^{act} \geq 1$. 又根据推论 1, 性质 2 成立.

证毕.

当 a 面对 A_node 中的多个动作时, 每个动作的执行难度都会影响它关于 t 的自信度. 式(1)给出了一种计算 a 关于 t 的能力自信度的悲观策略, “悲观”的含义可以这样解释: a 相信自己能够完成任务 t 的程度, 取决于 A_node 中执行难度最大的那个动作.

$$\sigma_a^t = \min_{\forall j \in A_node} (\sigma_a^j) \quad (4)$$

式中, j 表示 A_node 中的第 j 个动作, σ_a^j 按照定义 8 计算.

性质 3. $\sigma_a^t < 1 \Rightarrow \neg Bel_a t$.

证明. 根据式(4)和性质 2, $\sigma_a^t < 1 \Rightarrow \exists j \in A_node \sigma_a^j < 1 \Rightarrow \neg Bel_a j$. 又根据引理 4, 有 $\neg Bel_a j \Rightarrow \neg Bel_a t$.

证毕.

推论 2. $\forall act \in A_node CanDone(act) \Rightarrow Bel_a t \wedge \sigma_a^t \geq 1$.

证明. 根据推论 1 和引理 4, $\forall act \in A_node CanDone(act) \Rightarrow \forall act \in A_node Bel_a act \Rightarrow Bel_a t$. 又根据性质 2 和式(1), $\forall act \in A_node Bel_a act \Rightarrow \forall act \in A_node \sigma_a^{act} \geq 1 \Rightarrow \sigma_a^t \geq 1$.

证毕.

推论 3. 给定 $t \in AT$ 和任意两个 Agent a_1 和 a_2 , 设 $Bel_{a_1} t \wedge Bel_{a_2} t$.

$$\sigma_{a_1}^t > \sigma_{a_2}^t \Rightarrow Bel_{a_1} t \xrightarrow{\text{高于}} Bel_{a_2} t.$$

证明. 根据引理 4, $Bel_{a_1} t \wedge Bel_{a_2} t$ 表明 a_1 和 a_2 都相信自己有能力完成 t . 又根据定义 8, $\sigma_{a_1}^t > \sigma_{a_2}^t$

表明关于 t, a_1 可以比 a_2 承诺更少的付出. 对于理性 Agent 来说, 付出越少, 自信度越高, 符合其心智状态.

证毕.

4 基于 Agent 能力自信度的任务分配

定义 9. 给定 $t \in AT$ 和任意两个诚实 Agent a_1 和 a_2 . 联盟管理者 CM 基于 Agent 能力自信度的任务指派标准 φ 定义为

$$\varphi: Bel_{a_1} t \wedge Bel_{a_2} t \wedge \sigma_{a_1}^t \geq \sigma_{a_2}^t \Rightarrow alloc(t, a_1) \xrightarrow{\text{优先于}} alloc(t, a_2).$$

算法 1. 基于 Agent 能力自信度的任务分配.

Begin

$AG = \{\}; //$ 初始化联盟成员集合

For all $t \in AT$ Do

{ 记 t 的投标者集合为 Ω_t ;

IF Ω_t 不为空集 THEN $AG \leftarrow AG \cup \{a\}$ 满足

$$\sigma_a^t = \max_{\forall i \in \Omega_t} (\sigma_i^t)$$

ELSE CM 邀标^①, 并为其指定一个 $\sigma_a^t \geq 1$ 的值;
在 and-or-tree(T) 的 t 节点上标记 (a, σ_a^t) ;

};

Output(AG , and-or-tree(T));

End.

算法 1 在所有叶子节点上标记承担者及其能力自信度, 设标记操作为原子操作. 如果系统中问题求解 Agent 的总个数是 k , 则对任一原子任务 t , 其投标者的个数 $|\Omega_t| \leq k$, 意味着 max 运算最多需要比较 k 次. 考虑到节点标记操作, 则一次循环的最大运算次数为 $(k+1)$. 若 AT 中原子任务的总个数是 n , 则 n 次循环的总运算次数为 $(k+1) \times n$. 由此, 算法 1 的时间复杂度可以记作 $O(n)$, 其中, n 是原子任务的个数.

算法 2 将在上述的 and-or-tree(T) 上, 进一步计算联盟 AC 关于任务 T 的能力自信度 σ_{CM}^T . 设 and-or-tree(T) 的高度为 h , 其中, 树根 T 在第 0 层 (参见图 1).

算法 2. 计算联盟 AC 关于任务 T 的能力自信度 σ_{CM}^T .

Begin

For $i = h-1$ to 0 Do

{ 自左至右遍历 and-or-tree(T) 的第 i 层, 如果存在尚未标记能力自信度的子任务 t , 则

{ IF and- $t(t_1, t_2, \dots, t_p)$ THEN

$$\sigma_t \leftarrow \min_{j=1,2,\dots,p} (\sigma_{t_j}); // p \geq 1 \text{ 是 } t \text{ 的“与”子节点数}$$

IF or- $t(t_1, t_2, \dots, t_q)$ THEN

$$\sigma_t \leftarrow \max_{j=1,2,\dots,q} (\sigma_{t_j}); // q \geq 1 \text{ 是 } t \text{ 的“或”子节点数}$$

在 t 节点上标记 σ_t ;

}

};

$\sigma_{CM}^T \leftarrow \sigma_i //$ 第 0 层节点标记的能力自信度 σ_i 即为 σ_{CM}^T
Output(and-or-tree(T));

End.

算法 2 在所有非叶子节点上标记能力自信度, 设标记操作为原子操作. 由于“与”节点需要比较 p 次, 标记 1 次, “或”节点需要比较 q 次, 标记 1 次, 因此, 如果第 i 层上“与”节点的个数是 k_{i_1} , “或”节点的个数是 k_{i_2} , 则第 i 层的运算次数为 $(p+1) \times k_{i_1} + (q+1) \times k_{i_2}$. 树上的总运算次数为 $(p+1) \sum_{i=0}^{h-1} k_{i_1} + (q+1) \sum_{i=0}^{h-1} k_{i_2}$, 其中, $\sum_{i=0}^{h-1} k_{i_1} = n_1$ 是树上“与”节点的个数, $\sum_{i=0}^{h-1} k_{i_2} = n_2$ 是树上“或”节点的个数. 由此, 算法 2 的时间复杂度可记作 $O(n)$, 其中, $n = n_1 + n_2$ 为树上所有非叶子节点的个数.

推论 4. $\forall t \in AT, \forall a \in AG (Bel_a^t \wedge \sigma_a^t \geq 1) \Rightarrow Bel_{CM}^T \wedge \sigma_{CM}^T \geq 1$.

证明. 由 $\forall t \in AT, \forall a \in AG \exists Bel_a^t$ 和引理 2, 容易推得 Bel_{CM}^T . 不失一般地, 记 and-or-tree(T) 上第 i 层的某个任务 t 的能力自信度为 σ_i^t . 根据算法 1, 有 $\forall t \in$ 第 n 层 $\sigma_n^t \geq 1$. 令 $i = n-1$, 由算法 2, 不论第 $n-1$ 层的 t 是“与”分组还是“或”分组, 都有 $\forall t \in$ 第 n 层 $\sigma_n^t \geq 1 \Rightarrow \forall t \in$ 第 $n-1$ 层 $\sigma_{n-1}^t \geq 1$. 依此类推, 直至 $i=0$, 有 $\sigma_{i=0}^t = \sigma_{CM}^T \geq 1$. 证毕.

定义 10. 给定 $T \in TS$ 和任意两个 Agent 联盟 AC_1 和 AC_2 . 基于联盟能力自信度的任务指派标准 Ψ 定义为

$$\psi: Bel_{CM_1}^T \wedge Bel_{CM_2}^T \wedge \sigma_{CM_1}^T \geq \sigma_{CM_2}^T \Rightarrow$$

$$alloc(T, AC_1) \xrightarrow{\text{优先于}} alloc(T, AC_2).$$

基于合同网模型和联盟能力自信度的任务分配过程类似算法 1, 本文不再赘述.

5 一个应用的例

本文方法作为迁移 workflow 研究的一部分^[14], 已在本实验室研制的移动协同产品商务实验平台上进行了验证. 与传统的工作流模型不同, 迁移工作流是一个或多个迁移实例在不同的工作位置之间连续迁移并就地利用服务完成任务的过程, 其中, 迁移实例

① 邀标的目的是为了以防因某个(些)任务无投标者而导致联盟竞标失败, 邀标方法见文献[13].

通过自身携带的工作流说明和旅行图去实现 workflow 组织者的意图, 工作位置代表参与业务过程的企业为迁移实例提供 workflow 服务. 因此, 迁移 workflow 模型需要解决的核心问题之一是如何为迁移实例规划一个可编制旅行图的工作位置集合. 本研究把满足某个特定业务流程服务需求的所有企业(工作位置)的集合称作 workflow 域, 并且把 workflow 域的组织抽象为一个基于企业能力自信度结盟的任务分配问题, 即: 如果一个企业相信自己有能力为业务流程中的某个任务提供服务, 那么它就可以去竞争该任务; 如果流程管理者相信一个企业在竞争者中具有完成某个任务的最好能力, 那么它就会把任务委托给该企业去做, 假定所有企业都是理性的, 竞争是合作型的. 简化的实验系统结构如图 3 所示, 其中, 所有 Agent 均被设计为社会人的助手. 下面, 以订单式生产中的制造联盟为例, 说明基于制造能力自信度实现任务分配的过程, 其中, 制造能力自信度以产品造价和供货时间为评价因素, 假定制造商 B 只负责产品装配而不进行部件制造, 并且相信自己的装配能力.

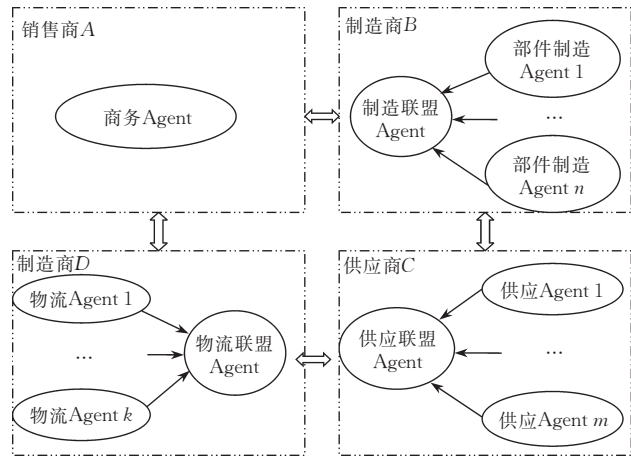


图 3 简化的协同产品商务系统架构

(1) 销售商 A 通过商务 Agent 向制造商 B 发布订单式商品制造任务(描述项包括: 名称、规格、数量、单价、供货时间、到货地点等).

(2) 制造商 B 依据订单说明和产品结构, 确定产品价格和供货时间在产品部件之间的分配值, 并构造一棵形如图 4 所示的制造需求树. 产品制造需求可以通过 ECA 规则描述, 例如

```
on 产品 if (( $\lambda_1 + \lambda_2 + \lambda_A$ ) <  $\lambda$ )  $\wedge$  ( $\max(\tau_1, \tau_2, \tau_A)$ ) <  $\tau$ ) then Done;
on 子体 1 if (( $\lambda_B + \lambda_C + \lambda_D$ ) <  $\lambda_1$ )  $\wedge$  ( $\max(\tau_B, \tau_C, \tau_D)$ ) <  $\tau_1$ ) then Done;
on 部件 B if ( $\lambda$  <  $\lambda_B$   $\wedge$   $\tau$  <  $\tau_B$ ) then Done;
on 部件 C if ( $\lambda$  <  $\lambda_C$   $\wedge$   $\tau$  <  $\tau_C$ ) then Done;
```

on 部件 D if (λ < λ_D \wedge τ < τ_D) then Done;

...

其中, λ 和 τ 分别表示单价和到货时间.

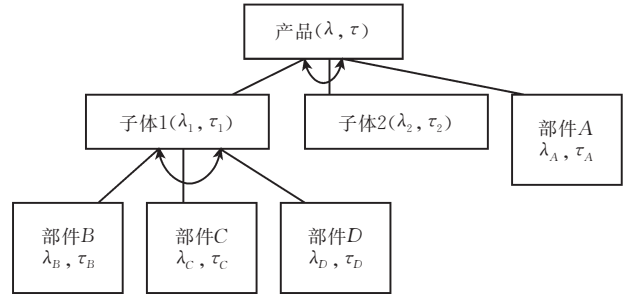


图 4 产品制造需求树的例

(3) 制造商 B 依据产品说明和产品制造需求规则, 借助制造联盟 Agent 向各部件制造 Agent 发布部件制造任务. 简化的招标规则形如

On 部件 X (规格、数量) if (供货时间 τ < τ_X) then bid, 其中, τ_X 是时间阈值.

(4) 各部件制造 Agent 投标. 简化的投标规则形如

On 部件 X bid(规格、数量、单价 λ 、供货时间 τ).

(5) 制造联盟 Agent 以单价 λ 和供货时间 τ 为制造能力约束, 应用算法 1 评价各部件制造 Agent 的能力自信度, 并实现产品部件在各部件制造 Agent 之间的分配.

(6) 制造联盟 Agent 应用算法 2, 计算联盟能力自信度, 并依据计算结果向商务 Agent 确认是否接受订单生产任务.

6 相关工作的比较

在多 Agent 协作研究中, 信任被视作一种基于规避协作风险、选择合作伙伴的手段, 信度被定义为一个 Agent 评价另一个 Agent 履行承诺的程度. 履行承诺的程度只能通过评价合作结果计算, 包括成功案例和失败案例两种情况, 因此, 初次合作时, 信度必须通过推断建立.

文献[6]给出了一种基于动机和信度建立同趣组(clan)的方法, 同趣组是一个互相信任并且具有相似目标的多 Agent 集合, 其中, 动机是 Agent 的一种高级协作愿望, 用于指导和控制 Agent 的协作目标与行为; 信度是一个 Agent 评价另一个 Agent 履行承诺的程度, 在 $[0, 1]$ 区间上取值, 用于选择合作伙伴时的比较, 没有具体意义. 文献[6]指出, 信度可以依照 Agent 的心态(乐观或悲观)推断初值, 并且按照当前信度、合作结果和 Agent 的心态进行更

新,但没有给出具体的推断方法和更新规则.文献[7]将信任看成是置信和信用的加权效果,其中,置信定义为管理 Agent 对某个 Agent 履行承诺的评价,信用定义为其它 Agent 向管理 Agent 提供的关于该 Agent 的评价信息.置信水平和信用水平用模糊集合表示,例如{差、一般、好} $\rightarrow[0,1]$,加权后映射到区间 $[-1,1]$,管理 Agent 可以据此选择合作伙伴.文献[8]把 Agent j 能够被 Agent i 所信任并将动作 k 委托给它执行的程度,称为 j 相对于动作 k 的行为自信度.显然,该定义仍然属于信任评价,而非 Agent 自信评价.文献[8]给出了 Agent 行为信度的修正规则和动作委托算法,但基于动作定义自信度不仅缺乏应用级的语义,而且难于把握动作的粒度,因而不适于任务级的信度评价及任务分配.另外,文献[8]定义的行为信度仅针对单个 Agent,没有涉及联盟信度计算问题.

上述基于合作结果评价信度的方法,是一种“亡羊补牢”的策略.虽然评价合作历史可以降低下一轮选择协作伙伴的风险,但不能根据能力指导当前成员之间的任务分配,特别不利于初次合作.一般地说,多 Agent 任务协作需要经历 3 个阶段:任务分配、问题求解和结果评价.从社会心理学的观点看,问题求解 Agent 只有相信自己有能力完成某个任务 t ,它才会勇于竞争 t 并履行承诺;问题管理 Agent 只有相信某个问题求解 Agent 有能力完成 t ,它才会将 t 委托给该 Agent 去做.

文献[9]认为,如果 Agent 相信自己在执行动作时将获得成功,那么它是自信的,但没有说明如何计量自信.文献[10]指出,如果 Agent 知道自己能够做什么,那么它相信自己一定能够做;如果 Agent 知道自己能够完成什么,那么它将尽力去完成,但没有说明 Agent 如何去度量“知道”.如果事件 e 的出现是 Agent i 自身动作的结果,文献[11]认为 i 关于事件 e 自信;如果事件 e 的出现是 Agent j 导致的结果,文献[11]认为 i 相信 j 最终会使 e 为真.文献[11]基于时态逻辑研究了上述置信的演算,但没有给出具体的置信计量方法.

本文沿用了文献[10-11]的观点,不同的是:把自信定义为 Agent 相信自己能够完成任务的程度,把信任定义为管理 Agent 相信问题求解 Agent 能够完成任务的程度,自信水平通过任务需求 ECA 规则和 Agent 能力 ECA 规则计量,信任水平通过评价 Agent 能力自信度获取.因此,本文方法不仅为自信计算提供了一种量化手段,而且为信任评价提供了一种基于 Agent 能力而非合作结果的途径.

如果把基于合作历史的信度评价与合作伙伴选择视作后馈控制,那么,本文给出的方法则可以视作伙伴选择和任务分配阶段的前馈控制,前馈控制可以强化问题求解 Agent 和问题管理 Agent 合作成功的信念,减少任务分担失败的风险.当然,前馈控制和后馈控制也可以结合应用,例如在一个互信的群体中邀标并评价成员的能力.

7 结束语

本文基于任务需求 ECA 规则集合和 Agent 能力 ECA 规则集合,定义了一种 Agent 的能力自信度,并给出了采用合同网模式和 Agent 能力自信度的任务分配算法.因为任务需求 ECA 规则集合中的条件项表达了 Agent 执行该任务时必须做出的付出,Agent 能力 ECA 规则集合中的条件项表达了 Agent 可以做出的承诺,所以,能力 ECA 规则的实例化过程可以使 Agent 清楚地知道自己能够做什么以及完成的难度.任务给定的条件越宽松,需要 Agent 做出的付出越少,Agent 竞标的胆量就会越大,自信度就会越高.因此,基于付出与承诺之间的关系定义 Agent 的能力自信度,不仅可以定量地揭示 Agent 抉择任务的心态,而且,基于 Agent 能力自信度进行分配任务,可以强化问题管理 Agent 成功的信念,减少任务分担失败的风险.

因为能力 ECA 规则实例化和投标是一个 Agent 对付出做出承诺的自主过程,所以,为了保证能力自信度的可信性,本文假定系统中的所有 Agent 都是诚实的,即不会故意制造虚假信息.对于那些非诚信 Agent,任务分配系统可以采用黑名单规则进行管理.

最后值得指出,如果协作被限定在一个互信的群体中进行,那么,本文给出的基于能力自信度的任务分配算法,仍然适用,尽管本文内容没有包括基于合作历史的信度计算.另外,本文也没有考虑联盟收益计算.当同时考虑自信、信任、联盟利益和成员利益时,借助结盟的任务分配是一个多因素决策问题.构建多因素决策的任务分配机制,是本研究正在进行的课题.

参 考 文 献

- [1] Shehory O, Kraus S. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 1998, 101(1-2): 165-200

- [2] Lau H C, Zhang L. Task allocation via multi-agent coalition formation: taxonomy, algorithms and complexity//Proceedings of the 15th IEEE Conference on Tools with Artificial Intelligence (ICTAI'03). Sacramento, California, USA, 2003; 346-350
- [3] Matthias K, Andreas G. Dynamic coalition formation among rational agents. IEEE Intelligent Systems, 2002, 17(3): 42-47
- [4] Kahan J P, Rapoport A. Theories of Coalition Formation. Hillsdale, New Jersey: Lawrence Erlbaum Associates Publishers, 1984
- [5] Osborne M J, Rubinstein A. A Course in Game Theory. Cambridge, Mass.: MIT Press, 1994
- [6] Griffiths N, Luck M. Coalition formation through motivation and trust//Proceedings of the AAMAS'03. Melbourne, Australia, 2003; 17-24
- [7] Ramchurn S D, Jennings N R, Sierra C, Godo L. Devising a trust model for multi-agent interacts using confidence and reputation. Applied Artificial Intelligence, 2004, 18(9-10): 833-852
- [8] Fan Xiao-Cong, Xu Dian-Xiang, Zheng Guo-Liang. Self-confidence degree of agent's behaviors and its revision rules. Chinese Journal of Computers, 1998, 21(1): 40-46(in Chinese)
- [9] Kraus S, Sycara K, Evenchik A. Reaching agreements through argumentation: A logical model and implementation. Artificial Intelligence, 1998, 104(1-2): 1-69
- [10] Shi Zhong-Zhi. Intelligent Agent and Application. Beijing: Science Press, 2000(in Chinese)
(史忠植. 智能主体及其应用. 北京: 科学出版社, 2000)
- [11] Fisher M, Ghidini C. The ABC of rational agent modeling//Proceedings of the AAMAS'02. Bologna, Italy, 2002; 849-856
- [12] Smith R G. The contract net protocol: High-level communication and control in a distributed problem solver. IEEE Transactions on Computers, 1980, 29(12): 1104-1113
- [13] Zeng Guang-Zhou, Wan Wu-Nan. Study on the application of contract net to bidding for engineering and equipment. Computer Simulation, 2003, 20(8): 103-105(in Chinese)
(曾广周, 万武南. 合同网模型在工程设备招标中的应用研究. 计算机仿真, 2003, 20(8): 103-105)
- [14] Zeng Guang-Zhou, Dang Yan. Study of migrating workflow based on the mobile computing paradigm. Chinese Journal of Computers, 2003, 26(10): 1343-1349(in Chinese)
(曾广周, 党妍. 基于移动计算范型的迁移 workflow 研究. 计算机学报, 2003, 26(10): 1343-1349)



ZENG Guang-Zhou, born in 1947, professor, Ph.D. supervisor. His research area includes CSCW, intelligent computing, mobile computing and workflow technology.

YANG Gong-Ping, born in 1970, Ph. D., associate professor. His research interests focus on intelligent computing and migrating workflow technology.

WANG Xiao-Lin, born in 1962, associate professor. Her research interests focus on CSCW and intelligent computing.

Background

The contract net protocol is a key mechanism in solving the cooperative work and has been widely used in the multi-agent systems, but there is no guarantee that a contracted agent will actually make the task successful and a mean to avoid partner selection risk for task manager is very important. At present, the most approaches for decreasing risk in selecting partners are evaluating the cooperative results and providing some trust revising rules to fit experience changing. As we known, agents cooperate usually go through three main phases: Task allocation, task execution and results evaluation. Therefore, acquiring trust degree of partners by evaluating the cooperative results was essentially a feedback control strategy, it only useful in next times and the cooperation opportunity for the agent who has no interaction with task manager may be missed.

Can the task manager not only decrease his risk but also allow the fresh agent to join cooperation in current task allocation phase? The answer is formative and providing a method to help the task manager is our pursuit aim. A kind of agent ability confidence and task allocation algorithm based on the agent ability confidence using contract net protocol are given in this paper. The basic idea is that agent ability confi-

dence can be revealed by the ratio of required payoff of the task and commitment of the agent to that task using ECA rules in bidding manner. The ECA rule is easy to represent so that ability confidences for an agent and the coalition formed by agents are easy to calculate. The main contribution of this paper is that the method puts the agent confidence concept into a computational model and the risk of partner selection can be avoid in current task allocation phase and allow the fresh agent to take part in the cooperative work.

This work is supported by the National Natural Science Foundation of China under grant No. 60573169 and the Shandong Province Project under grant No. 031110123 for migrating workflow research and cooperative product commerce application in mobile computing paradigm. Within a migrating workflow system, the traveling path is consisting of work places which represent enterprises or organizations, and the workflow manager must allocate tasks among participation in a safe way before process can be started.

On the project of migrating workflow research, the authors have proposed a migrating workflow framework and some methods for supporting workflow management such as navigation, communication, security, service awareness, etc.