

基于可能回答集程序的多 Agent 信念协调

吴甜甜 王 洁

北京工业大学信息学部 北京 100124

(W-tt@emails.bjut.edu.cn)



摘要 多 Agent 系统(Multi-Agent System, MAS)是人工智能领域的一个非常活跃的研究方向。在多 Agent 系统中,由于 Agent 之间信念的差异,会不可避免地造成行动冲突。Sakama 等提出的严格协调方法只适用于各 Agent 之间有共同信念的情境,当不存在共同信念时,此协调方法无解。针对该问题,文中提出了一种基于可能回答集程序(Possibilistic Answer Set Programming, PASP)的信念协调方法。首先,针对各 Agent 的不同信念集,基于加权定量的方法计算 PASP 的回答集相对 Agent 信念的满足度,以此来弱化某些信念,并且引入缺省决策理论推理得到 Agent 信念协调的一致解。然后,根据一致解建立一致的协调程序,将其作为 Agent 共同认同的背景知识库。最后,以 dl_v 求解器为基础实现了多 Agent 信念协调算法,使 Agent 之间可以自主完成信念协调。文中以旅游推荐系统为例,说明该算法能够打破严格协调方法的局限,有效解决各 Agent 之间无共同信念时的协调问题。

关键词: 可能回答集程序;加权定量;缺省决策理论;协调程序;多 Agent 系统

中图法分类号 TP301

Belief Coordination for Multi-agent System Based on Possibilistic Answer Set Programming

WU Tian-tian and WANG Jie

Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China

Abstract Multi-agent system MAS is a very active research direction in the field of artificial intelligence. In multi-agent systems, action conflicts will inevitably occur due to the difference in beliefs between agents. The rigorous coordination method proposed by Sakama et al. is only applicable to situations where there is a common belief among agents. When there is no common belief, this coordination method has no solution. In order to solve this problem, this paper proposed a belief coordination method based on possibilistic answer set programming (PASP). Firstly, according to different belief sets of agents, the weighted quantitative method is used to calculate the satisfaction degree of PASP's answer set relative to Agent's belief, so as to weaken some beliefs, and the default decision theory is introduced to deduce the consistent solution of Agent's belief coordination. Then, a consistent coordination program is constructed according to the consistent solution, which serves as the background knowledge base commonly recognized by agents. Finally, the multi-agent belief coordination algorithm is implemented to enable the belief coordination among agents to be completed auto-nomously based on the DLV solver. The example of tourism recommendation system shows that this algorithm can break the limitations of rigorous coordination method and effectively solve the coordination problem when there is no common belief among all agents.

Keywords Possibilistic answer set programming, Weighted quantitative, Default decision theory, Coordination program, Multi-agent system

1 引言

Agent 是一种可以在变化的环境中自主采取行动的软件或硬件系统^[1]。多 Agent 系统是由多个 Agent 组成的集合。多 Agent 系统作为人工智能领域中一个非常重要和活跃的研究方向,至今已有几十年的历史。国内外学者都对该领域做了深入的研究,并将 Agent 技术成功应用到故障诊段、微电网能源管理、运输资源调度^[2-4]等领域。

在多 Agent 系统中,不同的 Agent 可能有不同的信念,由于信念冲突^[5]会不可避免地造成行动冲突,因此 Agent 之间需要形成一个可以接受的协议,形成协议的过程叫做 Agent 间的协调。协调是解决信念冲突的一种有效策略。Stavropoulos 等提出了一种基于击败规则的多 Agent 协调框架^[6],该框架定义了规则的优先级,通过废除优先级较低的规则解决了 Agent 之间的冲突。Manzoor 等提出了基于极限环的多 Agent 协调方法^[7],通过协调 Agent 的移动方向来躲避动态障碍物,

到稿日期:2019-01-13 返修日期:2019-04-23 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。

基金项目:国家自然科学基金(61876010)

This work was supported by the National Natural Science Foundation of China (61876010).

通信作者:王洁(wj@bjut.edu.cn)

实验证明此方法是有效的。Eiter 等提出了一种基于遗忘理论的方法^[8],此方法通过遗忘特定的引起冲突的知识,来削弱每个 Agent 的部分信念,从而实现信念协调。Sakama 等提出了严格协调方法^[9]。该方法只承认所有 Agent 共有的信念,强制每个 Agent 舍弃与其他 Agent 不相交的信念,能够解决各 Agent 存在相交信念情境下的冲突问题。但是实际中,当 Agent 之间不存在共有信念时,严格协调方法将难以解决冲突。

为了解决 Agent 无共同信念时的冲突,找到一个相对满意的解,本文提出了一种基于可能回答集程序的协调方法。首先,针对不同 Agent 的信念集,基于加权定量的方法计算回答集相对 Agent 信念的满足度来弱化某些信念,获取相对满意的一致解。然后,根据一致解构建一致的协调程序,并将其作为 Agent 共同认同的背景知识库。最后,以 dlv 求解器为基础,实现多 Agent 信念协调算法,使 Agent 之间可以自主完成信念协调。

2 PASP 相关知识

可能回答集程序 PASP 是一种有效的不确定知识的描述方法^[10],其规则形式为:

$$c:L_0;\dots;L_m \leftarrow L_{m+1},\dots,L_k,notL_{k+1},\dots,notL_n$$

其中, $n \geq k \geq m \geq 0$, $L_i(i=0,\dots,n)$ 称为文字,规则左边称为规则头部,记作 $head(r)$,右边称为规则体部,记作 $body(r)$,";"表示析取, c 是规则的确度。规则的含义是当体部文字 L_{m+1},\dots,L_k 同时为真,并且没有理由相信 L_{k+1},\dots,L_n 为真时, L_0,\dots,L_m 中有且只有一个为真,且为真的确度为 c 。

2014年,Confalonieri 等将可能回答集程序应用于定性决策领域^[11]。2016年,Maia 等使用可能回答集程序完成信念推理^[12]。2017年,Son 使用完成多 Agent 的行动推理,实现行动规划^[13]。

本文使用可能回答集程序 PASP 这种形式化语言来描述多 Agent 系统中不确定的知识,进而完成协调。

3 基于 PASP 的多 Agent 信念协调

在多 Agent 系统中,不同功能的 Agent 具有不同的角色类型。本文主要分为两种角色类型:中介 Agent (mediator Agent) 和用户 Agent (user Agent)^[14]。中介 Agent 作为中介器,为用户 Agent 建立一组固定的推荐方案,综合各个用户 Agent 的信念,使用户 Agent 对推荐方案的选择达成一致。中介 Agent 的知识用可能回答集程序 P_0 表示;用户 Agent 的偏好知识,用扩展回答集程序 P_1,\dots,P_n 表示。

本文以旅游景点推荐的场景为例说明多 Agent 信念协调问题。旅行推荐系统(中介 Agent)推荐了 3 个旅游景点并列出了有关这 3 个景点的描述:

(1) 3 个景点分别为日本(*Japan*)、香港(*HongKong*)、西安(*Xi'an*)。

(2) 如果景点天气不好,会组织室内展览(*indoor_exhibition*)。

(3) 如果景点天气良好,会组织室外活动(*outdoor_activities*)。

(4) 同时参加室内展览和室外活动项目,花销较高(*high_cost*)。

(5) 日本近期天气较差($\neg good_weather$)。

(6) 西安近期天气良好(*good_weather*)。

(7) 香港近期天气良好。

(8) 香港会举办室内展览活动。

用可能回答集程序 P_0 描述以上场景:

$$r_1:1:Japan;HongKong;Xi'an \leftarrow.$$

$$r_2:1:indoor_exhibition \leftarrow \neg good_weather.$$

$$r_3:1:outdoor_activities \leftarrow good_weather.$$

$$r_4:0.9:high_cost \leftarrow indoor_exhibition, outdoor_activities.$$

$$r_5:0.7:\neg good_weather \leftarrow Japan.$$

$$r_6:0.8:good_weather \leftarrow Xi'an.$$

$$r_7:0.6:good_weather \leftarrow HongKong.$$

$$r_8:1:indoor_exhibition \leftarrow HongKong.$$

$$r_9:1:\neg high_cost \leftarrow not\ high_cost.$$

在程序 P_0 中,使用 $[0,1]$ 区间的数值来评估确定度,数值越大,确定程度越高。通过计算 P_0 的回答集集合得到 3 种旅行方案:

$$S_1 = \{(Japan,1),(\neg good_weather,0.7),(indoor_exhibition,0.7),(\neg high_cost,1)\}$$

$$S_2 = \{(HongKong,1),(good_weather,0.6),(outdoor_activities,0.6),(high_cost,0.6)(indoor_exhibition,1)\}$$

$$S_3 = \{(Xi'an,1),(good_weather,0.8),(outdoor_activities,0.8),(\neg high_cost,1)\}$$

现有 3 组顾客(用户 Agent)组成旅行团一起去旅行,因此需要共同决策一种方案。3 组顾客的偏好知识如下:

(1) 更喜欢室内展览,如果有室内展览项目,就选择一个花销低的景点,表示为回答集程序 P_1 :

$$r_1:indoor_exhibition \leftarrow$$

$$r_2:\neg high_cost \leftarrow indoor_exhibition$$

计算程序 P_1 的回答集,得到第 1 组顾客的信念集 B_1 为:
 $\{indoor_exhibition,\neg high_cost\}$

(2) 更倾向在天气好的条件下参加室外项目,表示为回答集程序 P_2 :

$$r_1:good_weather \leftarrow$$

$$r_2:outdoor_activities \leftarrow good_weather$$

计算程序 P_2 的回答集,得到第 2 组顾客的信念集 B_2 为:
 $\{good_weather,outdoor_activities\}$

(3) 更倾向同时参加室内展览和室外活动,表示为回答集程序 P_3 :

$$r_1:indoor_exhibition \leftarrow$$

$$r_2:outdoor_activities \leftarrow$$

计算程序 P_3 的回答集,得到第 3 组顾客的信念集 B_3 为:
 $\{indoor_exhibition,outdoor_activities\}$

可以看到,3 组顾客的信念集不相交,根据信念,顾客 1 会选择方案 S_1 ,顾客 2 会选择方案 S_2 或方案 S_3 ,顾客 3 会选择方案 S_2 ,不存在同时满足所有 Agent 信念集的方案。为了解决此类问题,本文考虑通过弱化某些 Agent 的信念找到一个相对满意的方案,实现协调一致。

3.1 加权定量

加权定量的方法^[15-16]可以用于处理不一致或冲突问题。

本文基于加权定量的思想计算方案相对 Agent 信念的满足度(权重)来弱化某些信念,使 Agent 信念达成一致。首先定义集合的满足关系,根据满足关系计算满足度。

定义 1 已知两个集合 X 和 Y ,对于集合 X 中的任意文字 l ,如果 $l \in Y$,那么 X 在 l 上满足 Y 。

根据满足关系定义(定义 1),计算 X 中与 Y 具有满足关系的文字集合 $X \cap Y$ 。然后,计算集合 X 关于集合 Y 的满足度(satisfaction degree) ω_{XY} :

$$\omega_{XY} = \frac{|X \cap Y|}{|Y|} \quad (1)$$

其中, $|X|$ 为集合 X 中的文字数量。

由此计算候选方案 S 关于 $Agent_i$ 信念的满足度,定义如下。

定义 2 确定环境下,候选方案 S 关于 $Agent_i$ 信念($belief_i$)的满足度为:

$$satDegree_i(S, belief_i) = \frac{|L \cap belief_i|}{|belief_i|} \quad (2)$$

其中, $S \in AS(P_0)$, $belief_i \in AS(P_i)$, $L = \{L_i \mid S = \{(L_1, c_1), \dots, (L_n, c_n)\}\}$, $1 \leq i \leq n$, $AS(P_i)$ 为程序 P_i 的回答集集合。

直观上, $L \cap belief_i$ 反映了方案 S 满足 $Agent_i$ 的相关信念。例如,在旅游景点推荐的实例中,方案 S_1 满足第 3 组顾客的部分信念,即 $\{indoor_exhibition\}$, 满足度为:

$$\frac{|\{indoor_exhibition\}|}{|\{indoor_exhibition, outdoor_activities\}|} = 0.5$$

但是,定义 2 在计算满足度时,没有考虑方案 S 成立的确定性度量。例如,在方案 S_1 中, $\neg good_weather$ 成立的确定性是 0.7,当用户考虑天气因素时,不仅需要考虑 $\neg good_weather$ 是否满足,还需要考虑满足的确定度。因此,根据方案 S 关于 $belief_i$ 的满足度(定义 2),并结合确定度,计算不确定环境下方案 S 相对 $Agent_i$ 的信念 $belief_i$ 的满足度。

定义 3 在不确定环境下,结合方案成立的确定度,候选方案 S 关于 $Agent_i$ 信念的满足度为:

$$cerSatDegree_i(S, belief_i) = \frac{\sum_{m=1}^n c_m \times e_{L_m}}{|belief_i|} \quad (3)$$

其中, $S = \{(L_1, c_1), \dots, (L_n, c_n)\}$ 。 e_L 是方案 S 关于信念 $belief_i$ 在文字 L 上是否存在满足关系的度量,如果存在,则 $e_L = 1$;否则, $e_L = 0$ 。 c 是方案 S 所满足的信念 $belief_i$ 中对应文字 L 的确定度。

定义 4(方案(回答集)) S 关于 n 个 Agent 信念的整体满足度为:

$$satDegree(S) = \sum_{i=1}^n cerSatDegree_i(S, belief_i) \quad (4)$$

直观上,某方案的满足度越大,选择该方案时对 Agent 信念的弱化程度越小。为了使多 Agent 之间能根据方案的满足度自主推理出信念一致解,引入缺省决策理论^[17-18]。缺省决策理论的形式为:

$$e \rightarrow A^s \text{ if } \epsilon(S, e) = \max_{S_i \in A} \epsilon(S_i, e) \quad (5)$$

$$\epsilon(S_i, e) = \sum_{j=1}^n cerSatDegree_j(S_i, belief_j) \quad (6)$$

其中, e 表示当前多 Agent 系统中的已知信息, $A = \{S_1, \dots, S_n\}$, $\epsilon(S_i, e)$ 表示可执行方案 S_i 对于所有 Agent 信念的整体满足度。缺省决策理论的含义是在已知信息 e 的条件下, A

中的所有可执行方案中,执行整体满足度最大的方案是最优的决策结果。

在实际推理过程中,使用谓词 $to_execute(S)$ 表示可执行方案 S , $satDegree(to_execute(S), X)$ 表示执行 S 方案的满足度 X , $consistentSolution(S)$ 表示信念协调一致解为 S 。为了推理信念一致解,把上面的缺省决策规则改写为回答集程序的形式:

$$consistentSolution(S_i) \leftarrow to_execute(S_i), \dots, to_execute(S_n), satDegree(to_execute(S_i), X_1), \dots, satDegree(to_execute(S_n), X_n), X_i > X_{i-1} > \dots > X_1, X_i > X_{i+1} > \dots > X_n \quad (7)$$

通过规则求解器可以自动推理出信念协调一致解。本文使用 $optimal$ 表示信念协调一致解(也称为最优解)的集合(一致解集)。由此说明,所有用户 Agent 在方案 $optimal$ 上达成一致。直观上, $optimal$ 可以使得 Agent 信念整体实现最大程度协调,但是可能某些 Agent 需要舍弃较多的信念,有些则需要舍弃较少的信念,反映了 Agent 必须进行一定程度的妥协,才能实现一致。

3.2 构建协调程序

根据加权定量的方法可以得到 Agent 协调的一致解。但是,原有的程序 P_0 对于用户 Agent 协调之后的信念来说是不一致的,因此需要及时更新程序,形成一个一致的逻辑程序——协调程序。协调程序作为共同知识库服务于所有的 Agent, Agent 将会继续基于协调程序行动、决策。协调程序的定义如下。

定义 5 P_0 是一个可能回答集程序, $optimal$ 是一致解集,那么协调程序为:

$$Coordination(P_0) = \bigcup_{S \in optimal} R(P_0, S) \quad (8)$$

其中,

$$\begin{aligned} R(P_0, S) &= \{c: head(r^+) \leftarrow body(r), not_body^-(r^+) \mid r \in P_0\} \\ head(r^+) &= head(r) \cap L \\ not_body^-(r^+) &= \{not L' \mid L' \in head(r) \setminus L\} \\ L &= \{L_i \mid S = \{(L_1, c_1), \dots, (L_n, c_n)\}\} \end{aligned}$$

引理 1 P_0 是一个可能回答集程序, S 是 P_0 的一个回答集,那么 S 是 $R(P_0, S)$ 的一个回答集。

证明:1) 因为 S 是 P_0 的一个回答集,根据回答集定义^[10], S 必然是 P_0^S 的一个回答集(P_0^S 是经过 Gelfond-Lifschitz 约简之后的程序),即是满足 P_0^S 中所有形如 $c: head(r) \leftarrow body^+(r)$ 规则的最小文字集合($body^+(r) \subseteq L$)。2) 通过定义 5 中 $R(P_0, S)$ 的定义可知, P_0^S 中每一条规则 $c: head(r) \leftarrow body^+(r)$ 都对应一条 $R(P_0, S)^S$ 的规则 $c: head(r) \cap L \leftarrow body^+(r)$ (因为 $body^-(r) \cap L = \emptyset$ 且 $(head(r) \setminus L) \cap L = \emptyset$)。因为 1) 成立,所以 S 也是满足 $R(P_0, S)^S$ 中的所有规则的最小文字集合,根据回答集定义, S 是 $R(P_0, S)$ 的一个回答集(注:证明过程中 $head(r) \setminus L$ 表示集合 $head(r)$ 减集合 L)。

定理 1 P_0 是一个可能回答集程序, $optimal$ 是一致解集, $Coordination(P_0)$ 是协调程序,则:

$$AS(Coordination(P_0)) = optimal$$

证明:1) 证明 $optimal \Rightarrow AS(Coordination(P_0))$ 。

令 $S \in optimal$, 因为 $optimal \subseteq AS(P_0)$, 所以 S 满足 P_0 中每条规则 $c: head(r) \leftarrow body(r)$ 。因此,通过引理 1 可知, S

满足 $R(P_0, S)$ 中的每条规则。同时, S 满足与 P_0 对应的 $R(P_0, T)$ 中的每条规则 $c: head(r) \cap T \leftarrow body(r), not(head(r) \setminus T)$, 其中 $T \in optimal$ 。故, S 满足 $Coordination(P_0)$ 。假设 S 不是 $Coordination(P_0)$ 的回答集, 存在集合 $U \subset S$ 满足 $Coordination(P_0)^S$, 此时, U 满足 $R(P_0, S)^S$ 。通过引理 1 可以得知, 如果 $S \in AS(P_0)$, 那么 S 是满足 $R(P_0, S)^S$ 的最小集合, 与假设矛盾。因此, S 是 $Coordination(P_0)$ 的一个回答集。

2) 证明 $AS(Coordination(P_0)) \Rightarrow optimal$ 。

令 $S \in AS(Coordination(P_0))$, 由定义 5 可知, 对于任意 $T \in optimal$, S 是满足 $R(P_0, T)$ 中每条规则 $c: head(r) \cap T \leftarrow body(r), not(head(r) \setminus T)$ 的最小集合。通过引理 1 可以得知, T 也是满足 $R(P_0, T)$ 的最小集合, 假设存在任意文字 $K \in S \setminus T, M \in T \setminus S$, 由于 $R(P_0, T)$ 中每条规则头部为 $head(r) \cap T$, 不包含文字 $K \in S \setminus T$, 因此 K 不包含在回答集 S 中, 进而得到 $S \setminus T = \emptyset$ 。同理, $T \setminus S = \emptyset$, 因为 T 和 S 都是最小的, 所以 $T = S, S \in optimal$ 。

由定理 1 可知, 协调程序的回答集集合等于一致解集, 保证了协调程序的正确性与一致性。

以下性质证明, 构建协调程序的时间复杂度相对严格协调方法的复杂度并未增加。

性质 1 在已知一致解集 $optimal$ 的条件下, 构建协调程序的时间复杂度是 $O(|P_0| \times |optimal|)$ 。其中, $|P_0|$ 表示 P_0 中规则的数量, $|optimal|$ 表示一致解的数量。

证明: 由定义 5 可知, 构建协调程序的过程是基于 P_0 中每条规则操作的, 因此时间复杂度为 $O(|P_0|)$, 如果存在多个一致解, 则时间复杂度为 $O(|P_0| \times |optimal|)$ 。

协调程序不仅可以作为共同认同的知识库, 而且当 Agent 接受一种方案时, 程序中规则的信息能够帮助 Agent 理解其他方案的知识, 解释了最终选择 $optimal$ 作为一致解集的原因。

3.3 多 Agent 信念协调算法的实现

为了实现多 Agent 之间的自主协调信念, 本文实现了多 Agent 信念协调算法。该算法的输入包括可能回答集程序。但是, dlV 求解器目前不能直接处理 PASP 的规则形式, 因此, 本文首先设计了 dlV 前端程序——front 程序, 用于将规则转化为 dlV 可以处理的形式, 供 dlV 调用求解; 然后计算程序的回答集权重, 求出信念协调一致解; 最后实现协调程序的自动构建算法——coordination 算法。具体的信念协调算法如算法 1 所示。

算法 1 多 Agent 信念协调算法

输入: P_0, \dots, P_n

输出: 信念协调一致解, 协调程序

1. $P_0' = front(P_0)$; //调用 front 程序将 P_0 转化为扩展析取逻辑程序
2. $AS(P_0'), \dots, AS(P_n) = dlV(P_0', \dots, P_n)$; //调用 dlV 系统求解
3. $AS(P_0) = calPossiAs(AS(P_0'))$; //求 P_0' 的回答集中文字的确定度
4. for(S in $AS(P_0)$)
 - for($belief_i$ in $AS(P_i)$)
 - 计算 $cerSatDegree_i(S, belief_i)$;
 - 计算 $satDegree(S)$;
5. $asp = rules(satDegree(S))$; //将方案满足度信息转化为规则形式
6. $ConsistentSolution = dlV(asp)$; //调用 dlV 求解信念协调一致解
7. $Coordination(P_0, consistentSolution)$; //调用 coordination 算法, 计算协调程序

4 实例

本文以第 3 节的旅行景点推荐场景为例, 说明上述协调方法在群体决策领域的应用。

首先通过 front 程序将可能回答集程序 P_0 转化为 dlV 求解器可处理的程序, 使用 dlV 求解器计算程序的回答集集合 $AS(P_0)$, 对应 3 种候选方案。然后, 通过 dlV 求解器计算 P_1, P_2, P_3 的回答集集合, 对应 3 组顾客的信念。3 组顾客的信念不相交, 并且 3 种候选方案均不能同时满足顾客的信念。为了使顾客实现一起旅行的目标, 通过加权定量的方法找到一个相对满意的旅行方案。具体来说, 根据定义 3 计算 3 种旅行方案相对每组顾客信念的满足度, 如表 1 所列。

表 1 各方案满足度信息

Table 1 Satisfaction information of each solution

	B_1	B_2	B_3
S_1	0.85	0	0.35
S_2	0.50	0.60	0.80
S_3	0.50	0.80	0.40

为了使多 Agent 自主推理得到信念协调的一致解, 这里将缺省决策理论应用于该场景中。首先, 将已知的方案满足度信息 e 转化为以下事实规则:

```

to_execute( $S_1$ ) ←
to_execute( $S_2$ ) ←
to_execute( $S_3$ ) ←
satDegree(to_execute( $S_1$ ), 1.2) ←
satDegree(to_execute( $S_2$ ), 1.9) ←
satDegree(to_execute( $S_3$ ), 1.7) ←

```

结合 3.1 节中的缺省规则(7)得到一个新的回答集程序 P , 求解程序 P 的回答集 A , 得到 $consistentSolution(S_2) \in A$, 表明一致解是方案 S_2 。显然, 第一组顾客舍弃了花销少的偏好, 才使得 3 组顾客在方案 S_2 上达成一致。直观上, 方案 S_2 可以最大程度地满足所有顾客的信念, 是一个相对满意的一致解。但是此时, 原来的程序 P_0 并不能实时反映当前 Agent 的一致解 S_2 , 需要形成一个新的逻辑程序 $Coordination(P_0)$, 其将作为共同的知识库继续服务于所有的 Agent。因此, 根据 coordination 算法自动构建协调程序 $Coordination(P_0)$ 为:

```

 $r_1: 1: HongKong \leftarrow not Xi'an, not Japan$ 
 $r_2: 1: indoor\_exhibition \leftarrow \neg good\_weather$ 
 $r_3: 1: outdoor\_activities \leftarrow good\_weather$ 
 $r_4: 0.9: high\_cost \leftarrow indoor\_exhibition, outdoor\_activities$ 
 $r_5: 0.7: \leftarrow Japan, not \neg good\_weather$ 
 $r_6: 0.8: good\_weather \leftarrow Xi'an$ 
 $r_7: 0.6: good\_weather \leftarrow HongKong$ 
 $r_8: 1: indoor\_exhibition \leftarrow HongKong$ 
 $r_9: 1: \leftarrow not high\_cost, not \neg high\_cost$ 

```

可以验证 $AS(Coordination(P_0)) = \{S_2\}$, 反映了 $Coordination(P_0)$ 的回答集集合正好对应 3 组顾客信念协调的一致解。并且 $Coordination(P_0)$ 中的规则可以解释选择方案 S_2

而不选择其他方案的原因。例如, r_5 的信息表明日本天气不好,解释了不选择日本的原因。

该实例说明了本文提出的多 Agent 信念协调方法打破了严格协调方法的局限性,可以解决在各 Agent 信念不相交时,求解相对满意的一致信念的问题。并且,协调程序的建立可以更好地反映和解释一致信念,其建立过程具备正确性和一致性。

结束语 多 Agent 技术是目前人工智能的研究热点,Agent 之间通过协调、合作,能够处理动态复杂的领域问题^[19]。然而,由于 Agent 信念的不一致,会不可避免地造成行动冲突。本文提出的基于可能回答集程序的信念协调方法能够更好地处理当多个 Agent 没有共同信念时的冲突问题。该方法通过引入加权定量以及缺省决策理论,来推理得到相对满意解(一致解),使其可以最大程度地满足所有 Agent 的信念,并且通过构建一致的协调程序,反映 Agent 共同认同的知识库。下一步的工作将考虑用回答集优化(ASO)^[20]的方法处理多 Agent 系统中的协调问题。

参 考 文 献

- [1] WANG J, LIU C. Agent-Oriented Probabilistic Logic Programming with Fuzzy Constraints [C] // Pacific Rim International Workshop on Multi-agents. Springer Berlin Heidelberg, 2006:4088-664-671.
- [2] CARRERA, ÁLVARO, IGLESIAS C A, et al. A real-life application of multi-agent systems for fault diagnosis in the provision of an Internet business service[J]. Journal of Network & Computer Applications, 2014, 37(1): 146-154.
- [3] JIA X, DOU C, BO Z, et al. Application of multi-agent technology in micro-grid system [C] // International Conference on Advanced Power System Automation & Protection. 2011: 16-20.
- [4] ZOUHAIER H, SAID L B. An Application Oriented Multi-Agent Based Approach to Dynamic Truck Scheduling at Cross-Dock [C] // International Conference on Parallel & Distributed Computing. IEEE, 2016, 1: 233-239.
- [5] ABE J M, REIS N F D, CRISTINA C D O, et al. A Logical Framework for Imprecise and Conflicting Knowledge Representation for Multi-agent Systems [C] // Ifip International Conference on Advances in Production Management Systems. Springer International Publishing, 2015, 459: 202-210.
- [6] STAVROPOULOS T G, RIGAS E S, KONTOPOULOS E, et al. A Multi-agent Coordination Framework for Smart Building Energy Management [C] // International Workshop on Database and Expert Systems Applications. IEEE, 2014: 126-130.
- [7] MANZOOR S, CHOI Y. Multi-agent coordination using limit cycles in dynamic environment [C] // International Conference on Control. IEEE, 2017.
- [8] EITER T, WANG K. Semantic forgetting in answer set programming [J]. Artificial Intelligence, 2008, 172(14): 1644-1672.
- [9] SAKAMA C, INOUE K. Coordination in Answer Set Programming [J]. Acm Transactions on Computational Logic, 2008, 9(2): 1-30.
- [10] NIEVES J C, OSORIO M, CORTÉ S U. Semantics for Possibilistic Disjunctive Programs [J]. Theory and Practice of Logic Programming, 2013, 13(1): 33-70.
- [11] CONFALONIERI R, PRADE H. Using possibilistic logic for modeling qualitative decision: Answer Set Programming algorithms [J]. Elsevier Science Inc., 2014, 55(2): 711-738.
- [12] MAIA G, ALCÂNTARA J. Reasoning about Trust and Belief in Possibilistic Answer Set programming [C] // Brazilian Conference on Intelligent Systems. IEEE Computer Society, 2016: 217-222.
- [13] SON T C. Answer set programming and its applications in planning and multi-agent systems [C] // International Conference on Logic Programming and Nonmonotonic Reasoning. Springer, Cham, 2017, 10377: 23-35.
- [14] ROSSI S, NAPOLI C D, BARILE F, et al. A Multi-Agent System for Group Decision Support Based on Conflict Resolution-Styles [M] // Conflict Resolution in Decision Making. 2017: 134-148.
- [15] LU F F, WANG J. An improved method for processing inconsistent answer set program [J]. Computer Science, 2015, 42(S1): 542-545.
- [16] AN X M, ZHANG Y Y, WANG J, et al. Optimization of the product configuration solution based on the weighted quantitative method [J]. Computer Engineering & Science, 2010, 32(8): 145-148.
- [17] WANG J, ZHANG T T. Research on context-aware in smart space based on ASP [J]. Computer Applications and Software, 2017, 34(2): 20-26.
- [18] XU B S. Research on conflict and deadlock in smart space [D]. Beijing, Beijing University of Technology, 2015.
- [19] FACCIN J, NUNES I. BDI-Agent Plan Selection Based on Prediction of Plan Outcomes [C] // 2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT). ACM, 2015.
- [20] ZHU Y, TRUSZCZYNSKI M. On Optimal Solutions of Answer Set Optimization Problems [C] // International Conference on Logic Programming and Nonmonotonic Reasoning. Springer Berlin Heidelberg, 2013: 556-568.



WU Tian-tian, born in 1992, postgraduate. Her main research interests include answer set programming and multi-agent system.



WANG Jie, born in 1972, Ph.D, associate professor. Her main research interests include logic programming and multi-agent system.