

信息传递法求不确定系统中的状态可达关系

劳佳琪 文中华 伍小辉 李 洋

(湘潭大学信息工程学院 湘潭 411105)

摘 要 在不确定规划领域中,在求规划问题的解时,由于缺少引导信息,会导致许多无用状态和动作被搜索,造成冗余计算。所以在求规划解之前,找到不确定状态转移系统中状态之间的可达关系是很有意义的。以往的算法是通过矩阵相乘来模拟状态转移,但该类算法对于规模较大的系统开销较大。因此,提出了用信息传递法来求解可达关系,用矩阵来模拟不确定状态转移系统。其中每个状态记录了其他状态到达该状态的可达信息,通过状态之间的可达信息的传递,求得不确定系统的状态可达关系,以避免大量的矩阵运算。通过实验对比表明,当不确定系统规模较大时,所设计的算法优于矩阵相乘的算法。

关键词 不确定规划,可达关系,信息传递

中图分类号 TP18

文献标识码 A

DOI 10.11896/j.issn.1002-137X.2014.10.056

Method of Information Delivery for State Accessibility in Non-determinate System

LAO Jia-qi WEN Zhong-hua WU Xiao-hui LI Yang

(College of Information Engineering, Xiangtan University, Xiangtan 411105, China)

Abstract In the field of non-determinate planning, when solving planning problems, due to lack of guidance information, the search will lead to many unwanted states and actions, resulting in the redundant calculations. So before seeking planning solution, it is significant to find up to the reachability relations between states in an indeterminate state transfer system. Previous algorithm is simulated by the state transition matrix multiplication, but such algorithm for large-scale system overhead is large. Therefore we proposed the method of information transmission to solve state accessibility with the matrix analog uncertain system. Each state records reachability information that the other states reach the state, and by passing reachability information between states, seek relationship between status in uncertain system to avoid a large number of matrix operations. Contrastive experiments show that the algorithm is superior to the design matrix multiplication algorithm.

Keywords Non-determinate planning, State accessibility, Information delivery

1 引言

智能规划是人工智能近年来的热点领域,不确定规划^[1]是其中的一个重要分支,即一个状态执行同一个动作可能会得到不同的结果状态。模型检测^[2,3]是不确定规划中处理不确定问题的重要方法。在该领域内,有很多专家和学者取得了重要的成果,如:对不确定状态系统分层^[4],求解强、弱、强循环规划解^[5],循环或非循环可达关系^[6,7]的求解,观察信息约简^[8]等。

在求解不确定规划问题时,由于缺少状态之间的可达信息,会产生大量重复和无用的搜索,从而降低效率。因此有学者提出了在不确定规划中状态非循环或循环可达的求解方法:利用矩阵相乘模拟不确定系统每一步的状态转移^[9],从而得出不确定系统中状态之间的可达关系。该算法随着问题规模的扩大急剧加大计算量,因此只适用于规模较小的问题。

本文参考了计算机网络中的 RIP 路由协议^[10],采用矩阵

来刻画问题模型,根据矩阵所拥有的不确定系统中状态之间的可达信息,设计了一种基于信息传递求解状态间可达关系的算法。每个状态都记录了其他状态到达该状态的可达信息,若该状态能够确定到达另外一个状态,则将此信息传递给它。通过对矩阵的遍历完成不确定系统中所有状态之间的信息传递,从而求解状态之间的可达关系。

2 相关定义

定义 1 一个规划领域是一个不确定的状态转移系统 $\Sigma = (S, A, \gamma)$, 其中 S 是有限状态集, A 是有限动作集, $\gamma: S \times A \rightarrow 2^S$ 是状态转移函数^[1]。

γ 用来刻画不确定性:在状态 s 下执行动作 a 所得到的状态集合就是 $\gamma(s, a)$ 。若 $\gamma(s, a)$ 非空,则称动作 a 在状态 s 下是可执行的。在状态 s 下可执行动作的集合记为 $A(s) = \{a: \exists s \in \gamma(s, a)\}$, 并称 (s, a) 为状态动作序偶。

定义 2 在一个不确定状态转移系统中,从状态 s_1 开始,

到稿日期:2013-12-24 返修日期:2014-03-07 本文受国家自然科学基金资助项目(61070232, 61272295)资助。

劳佳琪(1990—),男,硕士生,主要研究方向为智能规划、图论, E-mail: lywhlao@qq.com; 文中华(1966—),男,博士,教授,主要研究方向为智能规划; 伍小辉(1988—),男,硕士生,主要研究方向为智能规划; 李 洋(1991—),男,硕士生,主要研究方向为智能规划。

执行 $\gamma(s_1, a_1), \gamma(s_2, a_2), \dots, \gamma(s_n, a_n)$, 可能到达状态集合 $S_L = \{s | s \in \gamma(s_x, a_x), s \notin \{s_1, s_2, \dots, s_n\}, 1 \leq x \leq n\}$ 中的一个状态, 其中任意 $s_i \in \gamma(s_j, a_j) (\forall i, \exists j, 1 \leq i \leq n, 1 \leq j \leq i)$, 称 $L = \gamma(s_1, a_1), \gamma(s_2, a_2), \dots, \gamma(s_n, a_n)$, 为状态 s_1 的一条路径, 状态集合 S_L 为 s_1 经过路径 L 的可达信念状态。若 $s_i \in \gamma(s_j, a_j) (\exists i, i < j)$, 则称路径 L 是带循环的路径, 若 $s \in S_L$ 且 $|S_L| = 1$, 则称路径 L 是确定到达 s 的。

定义 3 在一个不确定状态转移系统中, 当状态 s_1 存在一条确定到达状态 s_2 的路径 L 时, 则称 s_1 到 s_2 确定可达; 若路径 L 中只存在确定动作, 则称为 1 型确定可达, 否则称为 2 型确定可达。当状态 s_1 存在一条路径 L 并且 $s_2 \in S_L$ 时, 则称 s_1 到 s_2 不确定可达。当状态 s_1 的任意一条路径 L 都有 $s_2 \notin S_L$ 时, 则称 s_1 到 s_2 不可达。

定义 4 设 $\Sigma = (S, A, \gamma)$ 是一个不确定状态转移系统, 用函数 $d(s_1, s_2)$ 来表示状态 s_1 到状态 s_2 之间的可达关系, 由可达关系组成的矩阵称为可达矩阵。可达关系的定义如下:

$$d(s_1, s_2) = \begin{cases} T, & s_1 \text{ 到 } s_2 \text{ 不确定可达} \\ 1, & s_1 \text{ 到 } s_2 \text{ 确定可达} \\ 0, & s_1 \text{ 到 } s_2 \text{ 不可达} \end{cases}$$

特别地 $d(s_i, s_i) = 1, (1 \leq i \leq n)$ 。

定义 5 设 $\Sigma = (S, A, \gamma)$ 是一个不确定状态转移系统, 对每个不确定动作给定一个唯一的标识, 记为 T_x ^[7]。状态 s_i 执行 T_x 后可能到达的状态集合 T_x , $mark = \{s_m, s_n, s_o, \dots, s_z\}$ 称为不确定动作 T_x 的信念状态。 $T_x.num$ 是不确定动作 T_x 的信念状态个数。

定义 6 设 $\Sigma = (S, A, \gamma)$ 是一个不确定状态转移系统, 用 $\tilde{s}_i$ 表示状态 $s_j (1 \leq j \leq n)$ 到状态 s_i 的可达信息, 即状态 s_i 所拥有的可达信息。 $\tilde{s}_i = \{d(s_1, s_i), d(s_2, s_i), \dots, d(s_n, s_i)\}$ 。若 $d(s_k, s_i) = T_x$, 则用 $s_{i-k}[T_x].value$ 表示状态 s_i 已知的 T_x 信念状态个数; 若 $s_{i-k}[T_x].value = T_x.num$, 则 $d(s_k, s_i) = 1$, 即状态 s_k 可以确定到达 s_i 。集合 $s_{i-k}[T_x].mark$ 记录了状态 s_i 已知 T_x 的信念状态的来源, 在未传递信息前 $s_{i-k}[T_x].mark = \{s_i\}$ 。

定义 7 在一个不确定状态转移系统中, 若状态 s_i 到 s_j 是确定可达的, 则状态 s_i 可将它所拥有的可达信息 $\tilde{s}_i$ 传递给 s_j , 用式子 $d(s_k, s_i) \rightarrow d(s_k, s_j); d'(s_k, s_j)$ 表示将状态 s_k 到状态 s_i 的可达信息传递给 s_j 。 $d'(s_k, s_j)$ 表示接收信息后状态 s_k 到状态 s_j 的可达关系。信息传递规则如下:

1. 若 $d(s_x, s_i) = 1 (x \neq i, j), d(s_x, s_j) = 1 \parallel T_n \parallel 0$, 则: $d(s_x, s_i) \rightarrow d(s_x, s_j); d'(s_x, s_j) = 1$ 。
2. 若 $d(s_x, s_i) = 0 (x \neq i, j), d(s_x, s_j) = 1 \parallel T_n \parallel 0$, 则: $d(s_x, s_i) \rightarrow d(s_x, s_j); d'(s_x, s_j) = d(s_x, s_j)$ 。
3. 若 $d(s_x, s_i) = T_n, d(s_x, s_j) = 0 \parallel T_y, y \neq n$ 则: $d(s_x, s_i) \rightarrow d(s_x, s_j); d'(s_x, s_j) = T_n + d(s_x, s_j), s_{j-n}[T_n].value = s_{i-n}[T_n].value, s_{j-n}[T_n].mark = \{s_i\}$ 。
4. 若 $d(s_x, s_i) = T_n, d(s_x, s_j) = 1$, 则: $d(s_x, s_i) \rightarrow d(s_x, s_j); d'(s_x, s_j) = 1$ 。
5. 若 $d(s_x, s_i) = T_n, d(s_x, s_j) = T_n$, 则: $d(s_x, s_i) \rightarrow d(s_x, s_j); d'(s_x, s_j) = T_n, s_{j-x}[T_n].value = s_{j-x}[T_n].value + card(s_{i-x}[T_n].mark - (s_{i-x}[T_n].mark \cap s_{j-x}[T_n].mark)), s_{j-x}[T_n].mark = s_{i-x}[T_n].mark \cup s_{j-x}[T_n].mark$ 。

定义 8 用矩阵 Adj 表示不确定转移系统 $\Sigma = (S, A, \gamma)$ 的状态邻接关系^[9], 邻接矩阵的表示规则如下:

状态 s_j 到 s_i 状态没有动作可以到达表示为:

$$Adj[i][j] = 0$$

状态 s_j 到状态 s_i 有动作可以到达表示为:

$$Adj[i][j] = 1$$

状态 s_j 到状态 s_i 可以通过不确定动作 T_x 到达, 则表示为:

$$Adj[i][j] = T_x$$

3 求状态可达性方法

本文通过矩阵模拟不确定转移系统 $\Sigma = (S, A, \gamma)$, 再通过状态之间的信息传递对不确定动作进行处理, 最终求得状态可达关系。状态之间的信息传递可分为 3 部分:

1) 信息的收集。若状态 s_i 需要进行信息传递, 则首先需要获得可达信息 $\tilde{s}_i$, 即记录矩阵 $Adj[i][x] (1 \leq x \leq n, x \neq i)$ 的信息。

2) 信息的传递。对第 i 列进行遍历, 若 $Adj[x][i] (1 \leq x \leq n, x \neq i)$, 则将状态 s_i 的信息 $\tilde{s}_i$ 传递给状态 s_x 。

3) 信息的更新。根据信息传递规则对状态 s_x 已知的可达信息 $\tilde{s}_x$ 进行更新。

通过以上 3 个步骤, 便完成了状态 s_i 对不确定系统中 $\Sigma = (S, A, \gamma)$ 其他状态的信息传递。然后依次对其他状态进行相同的处理, 全部处理完毕后, 整个系统便完成了初步的信息传递。由于不确定系统中, 状态之间可能存在 2 型确定可达, 因此需要再次遍历每个状态, 对每个状态中的可达信息 $\tilde{s}_x$ 进行归纳, 例如, 对于状态 s_k , 若 $Adj[k][i] = T_x, s_{k-i}[T_x].value = T_x.num$, 则修改 $Adj[k][i] = 1$, 即状态 s_i 可以确定可达状态 s_k , 然后再将状态 s_k 的信息 $\tilde{s}_k$ 传递给状态 s_i 。依次修改, 直到整个不确定系统中的不确定动作都已被检查。此时不确定系统中所有确定可达的关系都已被找出, 接下来需要处理不确定可达的状态对。首先对矩阵按列进行遍历, 找出值为 0 的元素, 假设该点为 $Adj[i][j] = 0$, 若 $Adj[i][k] = T_x (k \neq i)$, 同时存在 $Adj[k][j] = T_y$ 或 $Adj[k][j] = 1$, 那么就修改 $Adj[i][j] = T$, 若没有符合条件的点, 则 $Adj[i][j] = 0$ 。对整个不确定系统遍历完毕后, 就找出了所有不确定可达的状态对。此时, 矩阵中元素值为 0 的点则表示不可达。

证明:

信息的发送: 状态 s_i 所拥有的信息 $\tilde{s}_i$ 记录了其他状态 $s_k (k \neq i, 1 \leq k \leq n)$ 到达 s_i 的情况, 若 $d(s_i, s_j) = 1$, 即状态 s_i 可以确定到达 s_j , 那么状态 s_i 就可以向状态 s_j 发送信息。若 $d(s_k, s_i) = 1$, 那么状态 s_k 可以通过状态 s_i 确定到达状态 s_j , 即 $d(s_k, s_j) = 1$; 若 $d(s_k, s_i) = T_x$, 同理可得 $d(s_k, s_j) = T_x$ 。若 $d(s_k, s_i) = 0$, 则 $d(s_k, s_j) = 0$ 。由此可得, 通过状态 s_i , 状态 s_j 获得了其他状态是否可以到达它的信息。

由于不确定状态转移系统中可能存在环, 造成不确定动作信息会存在重复发送的情况, 因此需要对每个状态的不确定动作进行标记。集合 $s_{i-k}[T_x].mark$ 记录了状态 s_i 所知 T_x 的信念状态的来源, 在未进行信息传递前, $s_{i-k}[T_x].mark = \{s_i\}$ 。当状态 s_j 向状态 s_i 发送 T_x 的信息时, 首先判断该信息的 $s_{j-k}[T_x].mark$ 集合, 若 $s_{j-k}[T_x].mark \subseteq s_{i-k}[T_x].mark$, 则表明 s_i 状态已经拥有了状态 s_j 所拥有的 T_x 的信息, 不需要再进行信息发送。若 $s_{j-k}[T_x].mark \not\subseteq s_{i-k}[T_x].mark$, 说明状态 s_j 拥有状态 s_i 没有的关于 T_x 的信息, 则进行信息发送。

信息的更新:当状态 s_j 获得信息后,需要对自己本身所持有的信息进行比较,按照 $0 < T_x < 1$ 的优先规则进行修改。例如 $d(s_i, s_j) = 1$, 而状态 s_j 获得的信息为 $d(s_k, s_j) = 0$, 则不进行修改。若 $d(s_k, s_j) = T_x$, 且 $T_x.mark = \{s_{i1}, s_{i2}, \dots, s_{im}\}$, $s_{j-i}[T_x].value = T_x.num = n$, 则表明 $\forall s_m \in T_x.mark, (d_m, s_j) = 1$, 那么状态 s_i 在执行 T_x 后, 无论到达哪个状态, 它都可以确定到达状态 s_j , 所以 $d(s_i, s_j) = 1$ 。

以上叙述证明了本文提出的信息传递方法能够成功求解不确定系统中的状态之间的可达关系。

4 状态可达性算法及分析

设 $\Sigma = (S, A, \gamma)$ 是一个不确定状态转移系统, S 中有 n 个状态, Adj 是 $\Sigma = (S, A, \gamma)$ 的邻接矩阵。算法如下:

```
1. function STATEACCESS( $\Sigma, Adj$ )
2. if  $\exists i, \exists j, Adj[i][j] = T_x$  then
3.  $s_{i-j}[T_x].value = 1$ 
4.  $s_{i-j}[T_x].mark = \{s_i\}$ 
5. fi
6. FINDCERTAIN( $\Sigma, Adj$ );
7. FINDUNCERTAIN( $Adj$ );
8. return  $Adj$ ;
```

第 2—5 行对该不确定系统进行初始化。第 6 行即子函数 $FINDCERTAIN(\Sigma, Adj)$ 找出不确定系统中可以确定可达的状态对。第 7 行即子函数 $FINDUNCERTAIN(Adj)$ 找出不确定系统中所有不确定可达的状态对, 其定义如下:

$$FINDUNCERTAIN(\Sigma, Adj) = \{Adj[i][j] = T \mid Adj[i][j] = 0 \ \&\& \ (\exists k, Adj[i][k] \neq 0 \ \&\& \ Adj[k][j] \neq 0)\}$$

下面阐述 $FINDCERTAIN(\Sigma, Adj)$ 函数, 其具体过程如下:

```
1. function FINDCERTAIN( $\Sigma, Adj$ )
2. for  $i = 1$  to  $n$ 
3. for  $j = 1$  to  $n$ 
4. if  $Adj[i][j] \neq 0 \ \&\& \ i \neq j$  then
5.  $\tilde{s}_i = \tilde{s}_i \cup Adj[i][j]$ 
6. next  $j$ 
7. if  $\exists k, Adj[k][i] = 1 \ \&\& \ k \neq i$  then
8. TRANSMIT( $s_k, \tilde{s}_i$ )
9. fi
10. next  $i$ 
11. UPDATE( $\Sigma, Adj$ )
```

第 4、5 行收集状态 s_i 所拥有的信息 $\tilde{s}_i$ 。第 7—9 行寻找状态 s_i 可以确定到达的状态, 通过 $TRANSMIT(s_k, \tilde{s}_i)$ 函数将 $\tilde{s}_i$ 传递给状态 s_k 。第 11 行通过 $UPDATE(\Sigma, Adj)$ 函数, 对整个不确定系统中所有状态的不确定动作进行更新。两个子函数具体过程如下:

$TRANSMIT(s_k, \tilde{s}_i)$ 具体过程如下:

```
1. function TRANSMIT( $s_k, \tilde{s}_i$ )
2. for  $j = 1$  to  $n$ 
3. if  $Adj[i][j] = 1 \ \&\& \ j \neq i$  then
4.  $Adj[k][j] = 1$ 
5. fi
6. if  $Adj[i][j] = T_x \ \&\& \ Adj[k][j] = T_x \ \&\&$ 
7.  $s_{i-j}[T_x].mark \not\subset s_{k-j}[T_x].mark$  then
8.  $s_{k-j}[T_x].mark = s_{i-j}[T_x].mark \cup s_{k-j}[T_x].mark$ 
```

```
 $s_{k-j}[T_x].value += card(s_{i-j}[T_x].mark - (s_{k-j}[T_x].mark \cap s_{i-j}[T_x].mark))$ 
```

```
9. fi
10. if  $Adj[i][j] = T_x \ \&\& \ Adj[k][j] = T_x \ \&\& \ 0$  then
11.  $Adj[k][j] = Adj[k][j] \cup T_x$ 
12.  $s_{k-j}[T_x].value = s_{i-j}[T_x].value$ 
13.  $s_{k-j}[T_x].mark = s_{i-j}[T_x].mark$ 
14. fi
15. UPDATE( $\Sigma, Adj$ )
16. next  $j$ 
```

第 2—14 行根据信息传递规则, 修改状态 s_k 所拥有的可达信息。第 15 行调用 $UPDATE(\Sigma, Adj)$, 对状态 s_k 中的不确定信息进行修正, 其具体过程如下:

```
1. function UPDATE( $\Sigma, Adj$ )
2. if  $\exists k, \exists j, s_{k-j}[T_x].value = T_x.num$  then
3.  $Adj[k][j] = 1$ 
4. TRANSMIT( $s_k, \tilde{s}_j$ )
5. fi
```

第 2 行, 若状态 s_k 拥有 T_x 的所有信念状态, 则表明状态 s_j 可以 2 型确定可达状态 s_k , 修改 $Adj[k][j] = 1$ 。第 4 行, 由于 s_j 可以确定到达状态 s_k , 因此将状态 s_k 的信息传递给状态 s_j 。

5 算法示例及实验分析

设 $\Sigma = (S, A, \gamma)$ 是一个确定状态转移系统, 如图 1 所示。运用信息传递的方法来解决该系统的状态可达关系。

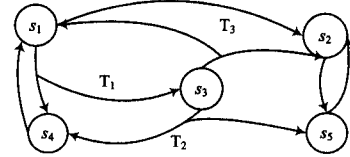


图 1 不确定状态转移系统超图

设 Adj 为邻接矩阵, 初始化为:

$$Adj = \begin{bmatrix} 1 & 0 & T_3 & 1 & 0 \\ 1 & 1 & T_3 & 0 & 1 \\ T_1 & 0 & 1 & 0 & 0 \\ T_1 & 0 & T_2 & 1 & 0 \\ 0 & 1 & T_2 & 0 & 1 \end{bmatrix}$$

首先从状态 s_1 开始进行信息转发, 由于 $Adj[2][1] = 1$, 因此 s_1 把信息传递给状态 s_2 。 s_2 接受信息, 由于 $s_{2-3}[T_3].value = T_3.num = 2$, 因此 $Adj[2][3] = 1$, 同时由于 $Adj[1][4] = 1, Adj[2][4] = 0$, 根据信息传递规则 1 修改 $Adj[2][4] = 1$, 因此 $Adj[2] = \{1, 1, 1, 1, 1\}$ 。

$$Adj = \begin{bmatrix} 1 & 0 & T_3 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ T_1 & 0 & 1 & 0 & 0 \\ T_1 & 0 & T_2 & 1 & 0 \\ 0 & 1 & T_2 & 0 & 1 \end{bmatrix}$$

再由 s_2 进行信息转发, 由于 $Adj[5][2] = 1$, 因此 s_2 将信息传递给状态 s_5 , 根据信息传递规则 1 得到 $Adj[5] = \{1, 1, 1, 1, 1\}$ 。

$$Adj = \begin{bmatrix} 1 & 0 & T_3 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ T_1 & 0 & 1 & 0 & 0 \\ T_1 & 0 & T_2 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

接下来 s_3 进行转发, 因为 $Adj[2][3]=1, Adj[5][3]=1$, 所以 s_3 向状态 s_2 和状态 s_5 进行转发, 根据信息传递规则 2 和 4, 不做修改。

同理, 接下来由 s_4, s_5 进行信息转发, 得到的 Adj 为:

$$Adj = \begin{bmatrix} 1 & 0 & T_3+T_2 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ T_1 & 0 & 1 & 0 & 0 \\ T_1 & 0 & T_2 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

这样找出了不确定转移系统中所有确定可达的状态关系, 接下来需要对 Adj 中元素值为 0 的状态对进行判定。由于 $Adj[3][4]=0, Adj[3][1]=T_1, Adj[1][4]=1$, 因此 $Adj[3][4]=T_1$ 。

最后得到的 Adj 矩阵为:

$$Adj = \begin{bmatrix} 1 & 0 & T_3+T_2 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ T_1 & 0 & 1 & T_1 & 0 \\ T_1 & 0 & T_2 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

以下为采用矩阵相乘方法和信息传递方法求状态可达关系的实验结果比较。实验环境均为 Windows7+Core(TM)i3-3220 3.3GHz+4.0GB 内存+VC6。两个算法使用的数据输入输出过程相同, 故其运行时间没有包括在内。

表 1 运行时间比较

状态数/ms	矩阵相乘/ms	信息传递/ms
5	0.0018	0.0021
10	0.0069	0.0070
20	0.0366	0.0269
30	0.1403	0.0765
50	1.0374	0.2049
70	4.8399	0.4231
100	30.2210	0.8885
130	90.3567	1.5140
150	283.5321	1.8891
200	917.8564	3.0163
300	9061.065	6.1892

从表 1 中可以发现, 当状态数较少时, 由于计算量较少, 矩阵相乘法速度较快; 当状态数较多时, 矩阵相乘法的计算量

也急剧增加, 信息传递法的优势就体现出来了。

结束语 本文设计了用信息传递方法求解不确定系统中可达关系的算法。充分利用不确定系统中每个状态所拥有的可达信息, 通过信息收集、信息传递、信息更新, 可以快速求出确定系统中状态之间的可达关系。本文的进一步工作有:

1. 优化信息传递策略, 加快求解可达关系的速度。
2. 运用信息传递的思想求解不确定规划的强、弱、强循环规划解。

参考文献

- [1] Weld D S. Recent advances in AI planning [J]. AI Magazine, 1999, 20(2): 93-123
- [2] Cimatti A, Roved M, Traverso P. Strong planning in nondeterministic domains via model checking [C] // Proceedings of the 4th International Conference on AI Planning Systems. 1998: 36-43
- [3] Cimatti A, Roveri M. Conformant planning via symbolic model checking [J]. Journal of Artificial Intelligence Research, 2000, 13(3): 305-338
- [4] Wen Zhong-hua, Huang Wei, Liu Ren-ren. Method of hierarchical states in planning based on model checking [J]. Journal of Software, 2009, 20(4): 858-869
- [5] Cimatti A, Pistore M, Roveri M, et al. Weak, strong, and strong cyclic planning via symbolic model checking [J]. Artificial Intelligence, 2003, 47(1): 35-84
- [6] 黄丽芳, 文中华, 胡雨隆, 等. 不确定规划中状态循环可达关系的求解方法 [J]. 计算机应用研究, 2013, 30(9): 2689-2693
- [7] 胡雨隆, 文中华, 常青, 等. 不确定规划中状态非循环可达关系的求解方法 [J]. 计算机仿真, 2012, 29(5): 114-117
- [8] Huang Wei, Wen Zhong-hua, Jiang Yun-fei, et al. Observation reduction for strong plans [C] // Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07). 2007: 1930-1935
- [9] 文中华, 黄巍, 刘任任, 等. 模型检测规划中的状态之间的可达关系研究 [J]. 计算机学报, 2012, 35(8): 1634-1643
- [10] 谢希仁. 计算机网络 (第五版) [M]. 北京: 电子工业出版社, 2007: 147-152

(上接第 260 页)

- [5] 邓志鸿, 唐世渭, 张铭, 等. Ontology 研究综述 [J]. 北京大学学报: 自然科学版, 2002, 38(5): 730-738
- [6] 崔巍. 用本体实现地理信息系统语义集成和互操作 [D]. 武汉: 武汉大学, 2004
- [7] 孙瑜. 本体修正 [D]. 北京: 中国科学院计算技术研究所, 2006
- [8] Severance C, Lan Horrocks. Standardizing OWL [J]. Computer, 2013, 46(11): 8-9
- [9] Repplinger, Chowdhury J G G. Introduction to Modern Information Retrieval (3rd ed) [M]. College and Research Libraries, 2011
- [10] Mitra P, Wiederhold G, Kersten M L. A Graph-oriented model for articulation of ontology interdependencies [C] // Proceedings of the 7th International Conference on Extending Database Technology, Advances in Database Technology. Konstanz, Ger-

many, 2000: 86-100

- [11] Noy N F, Musen M A. Smart: Automated support for ontology merging and alignment [C] // 21th Workshop on Knowledge Acquisition, Modeling, and Management. Banff, Canada, 1999
- [12] McGuinness D L, Fikes R, Rice J, et al. An environment for merging and testing large ontologies [C] // Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning. Colorado, USA, 2000: 483-493
- [13] Chalupsky H. Ontomorph: A translation system of symbolic logic [C] // Proceeding of the 7th International Conference (KR2000). San Francisco, CA, 2000: 471-482
- [14] Noy N F, Musen M A. Prompt: algorithm and tool for automated ontology merging and alignment [C] // Proceedings of the 2000 National Conference on Artificial Intelligence. Texas, 2000: 450-455