

对不确定规划中观测约简的进一步研究*

饶东宁¹⁺, 蒋志华^{1,2}, 姜云飞¹, 朱慧泉³

¹(中山大学 信息科学与技术学院 软件研究所, 广东 广州 510275)

²(暨南大学 计算机系, 广东 广州 510632)

³(School of Computing, National University of Singapore, Singapore)

Further Research on Observation Reduction in Non-Deterministic Planning

RAO Dong-Ning¹⁺, JIANG Zhi-Hua^{1,2}, JIANG Yun-Fei¹, ZHU Hui-Quan³

¹(Software Research Institute, School of Information Science and Technology, Sun Yat-Sen University, Guangzhou 510275, China)

²(Department of Computer Science, Jinan University, Guangzhou 510632, China)

³(School of Computing, National University of Singapore, Singapore)

+ Corresponding author: E-mail: rdn2006@163.com

Rao DN, Jiang ZH, Jiang YF, Zhu HQ. Further research on observation reduction in non-deterministic planning. Journal of Software, 2009,20(5):1254–1268. <http://www.jos.org.cn/1000-9825/3453.htm>

Abstract: This paper improves the methods of observation reduction in non-deterministic planning (NDP) in three aspects: in finding MOS (minimal observation set); in finding out the optimal observation set (OOS) when observations have different costs; and in finding fault-tolerant OOS. A MOS problem is similar to a minimal set cover (MSC) problem, so it can be proved that finding MOS is NP-hard. Inspired by MSC methods, an $O(2^m m^2)$ but $\Omega(2^{m-1})$ algorithm for MOS is presented, where m is the number of observations. By using integer programming (IP) technologies, OOS or fault tolerant OOS can be found out. Proofs are provided to show that these algorithms can guarantee finding optimal solutions.

Key words: AI (artificial intelligent) planning; non-deterministic planning; observation reduction; minimal observation set; optimal observation set; fault-tolerant

摘要: 从3个方面改进了不确定规划(non-deterministic planning,简称NDP)中的观测约简:一是如何找最小观测集合(minimal observation set,简称MOS),二是如何在观测代价不均等时找最优观测集合(optimal observation set,简称OOS),三是如何找到容错的OOS.通过MOS问题和图论中的最小覆盖集问题(minimal set cover,简称MSC)的类似性,可证MOS是NP难的问题,还可参考MSC算法得出时间复杂性不超过 $O(2^m m^2)$ 且不低于 $\Omega(2^{m-1})$ 的算法,其中 m 是观测的个数.通过使用整数规划(integer programming,简称IP)技术,可找到OOS以及容错的OOS.可以证明,上述算法能够保证找到解,并且能够保证解的最优性.

关键词: 智能规划;不确定规划;观测约简;最小观测集;最优观测集;容错

中图法分类号: TP18 文献标识码: A

* Supported by the National Natural Science Foundation of China under Grant No.60773201 (国家自然科学基金)

Received 2008-04-28; Accepted 2008-08-28

智能规划是人工智能的一个分支,近年来成为一个研究热点^[1-5].而规划中研究不确定的动作效果的分支——不确定规划(non-deterministic planning,简称 NDP)更是热点中的热点^[6-11],近几年 IJCAI(International Joint Conferences on Artificial Intelligence)及 AAAI(Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence)均设立了专门的 NDP session.在 NDP 中,由于动作效果是不确定的,往往需要进行观测,以区分多种可能的状态.但是,在现实中观测信息的获取往往需要各种各样的代价,比如传感器的能耗以及通信代价等等^[12,13].同时,很多情况下,观测变量一般又都是可选的,甚至在规划执行的过程中用不到.因此要找到一个对规划的执行而言最小的观测集,或者一种最好的观测组合方案^[12,13].进一步来说,考虑到观测(例如传感器)也有可能出错,有时需要一个能容错的观测组合方案.

由于观测约简是 NDP 领域中比较前沿的课题,目前为止,还很少有文献讨论观测约简的方法^[10,11].IJCAI-05 上的文章^[10]描述了如何在不损失信息的前提下基于当前策略动态划分观测空间.文中提出,如果两个观测总是被用来区分同一个动作过程,那么它们被认为是不必区分的.进而,这些不必区分的观测可以进行合并,从而实现了观测约简.IJCAI-07 上的文章^[11]进一步扩展了这个问题的范围,希望在已知规划解的前提下找到一个最小观测集合(minimal observation set,简称 MOS).该文提出了一种贪心算法,并将这个问题正式命名为观测约简.为了进一步简化问题,该文首先抽取规划过程中所有需要被区分的状态对,并将它们映射到观测变量集合上,然后优先保留能够区分最多状态对的观测.上述两篇文章都假设所有的观测具有相同的代价.显然,一方面贪心算法无法保证得到最优解,另一方面,在现实问题中可能会出现观测代价不相同的情况.进一步来说,在文献[11]的未来工作中也指出,由于观测可能出错,需要对观测进行容错.

鉴于此,本文试图从3个方面进一步解决观测约简问题:一是找到一种高效的求 MOS 算法,二是提出能够在观测代价不相等的情况下找到最优观测组合方案的方法,三是提出能够找出容错的最优观测集的方法.观察发现,MOS 问题可以映射成图论中的最小覆盖集问题(minimal set cover,简称 MSC),据此不但可以证明 MOS 是一个 NP 难题,而且可以基于 MSC 算法设计出求 MOS 的算法.另一方面,求解带权值的最优化问题时常用的一种手段是整数规划(integer programming,简称 IP)^[14,15],所以可以使用 IP 来求解观测代价有权值情况下的最优观测集(optimal observation set,简称 OOS).对于容错的观测集,也可以采用 IP 方法进行求解.本文提出将 MOS 映射为 MSC,提出能够求得 MOS 问题的最优解的算法,同时,将 MOS 问题扩展到观测代价有权值的情况并使用 IP 技术进行求解,首次考虑求容错的观测集问题.

本文第 1 节介绍研究背景,包括不确定规划、观测约简问题以及 MSC 问题.第 2 节将 MOS 映射到 MSC 并提出求解算法.第 3 节介绍使用 IP 技术求解 OOS.第 4 节介绍将 IP 技术扩展到容错 OOS 的求解上.第 5 节对全文进行总结.

1 研究背景

1.1 不确定规划

不确定规划是人工智能规划的一个分支,在 NDP 中动作的效果是不确定的.根据状态是否完全可以被观测,NDP 可分为完全可观测 NDP^[8]以及部分可观测 NDP(partially observable Markov decision process,简称 POMDP)^[6].一般定义不确定规划领域、观测函数以及带有观测的不确定规划领域如下:

定义 1(不确定规划领域). 一个不确定规划领域是一个四元组 $\Sigma=(P,S,A,T)$:

- P 是命题集合;
- $S \subseteq 2^P$ 是状态集合;
- $A=\{a|a=\{pre(a),add(a),del(a)\}\}$ 是动作集合,其中, $pre(a)$ 是前提集合, $add(a)$ 和 $del(a)$ 分别是增加效果和删除效果集合;
- $T:S \times A \rightarrow 2^S$ 是状态转换函数.

一个动作 a 在状态 s 下是可用的,当且仅当 $pre(a) \subseteq s$.在不确定规划领域中,在状态 s 下应用动作 a ,由于动作效果是不确定的,将导致多个后继状态.

定义 2(观测函数). S 是一个不确定规划领域的状态集, V 是一个有限的观测变量集合. 观测函数 X 是映射: $S \times V \rightarrow \{T, \perp\}$.

不失一般性, 这里只考虑布尔类型的观测变量. 在定义 2 中, T 表示观测变量的值为真, \perp 表示观测变量的值为假. 观测函数表明在某个状态 s 下观测变量 v 的值. 观测变量也称为观测.

定义 3(带有观测的不确定规划领域). 带有观测的不确定规划领域是一个三元组 $D=(\Sigma, V, X)$, 其中:

- $\Sigma=(P, S, A, T)$ 是一个不确定规划领域;
- V 是一个有限的观测变量集;
- X 是观测函数: $S \times V \rightarrow \{T, \perp\}$.

基于上述定义可以定义规划问题和规划解.

定义 4(规划问题). 一个规划问题是一个三元组 $\Pi=(D, I, G)$:

- $D=(\Sigma, V, X)$ 是规划领域;
- I 是初始状态;
- G 是目标状态.

定义 5(规划解). 不确定规划问题的规划解是策略 π , π 是由状态-动作序对构成的集合:

$$\pi = \{ \langle s, a \rangle | s \in S, a \in A, a \text{ 在 } s \text{ 中是可应用的} \}.$$

通过策略 π , 可以使规划问题从初始状态转换到目标状态.

NDP 有很多比较成熟的求解技术^[6-9], 本文中假设规划解是已知的.

1.2 观测约简

观测约简就是在规划解确定之后约简观测变量, 以节约规划执行的开销. 在未求出规划解的规划空间中, 需要区分状态对往往非常多, 所以本文及以前对观测约简的讨论^[10,11] 都只考虑规划解确定之后的问题. 不过, 只要能够枚举需要区分的状态对, 那么这些技术都可以不加修改地应用到未求出规划解的规划空间中. 此时需要先根据初始状态和动作的可能效果进行状态的可达性分析, 并找出可能需要区分的状态对. 这样的状态对仍然很多, 但比所有状态的组合要少得多. 因此在可用 NDP 描述的领域中, 本文的技术比一般传感器优化技术更有优势, 这是因为本文的技术利用了规划的特点, 只区分需要区分的状态对, 而一般的传感器优化技术需要识别具体的状态.

在求得规划解之后, 规划器或者系统控制器会按照规划解来执行动作. 这样才能改变系统状态, 实现从初始状态到目标状态的转变, 达到规划的目的. 尽管初始状态往往是已知的, 但动作的效果是不确定的, 即有多个可能的后继状态. 规划解是一个基于状态的策略, 所以在执行过程中需要观察系统当前状态.

例 1: 有一个迷宫游戏, 如图 1 所示.

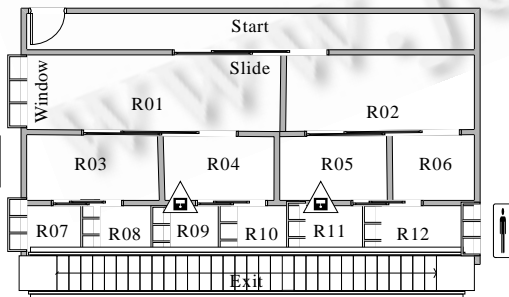


Fig.1 A maze game

图 1 一个迷宫游戏

迷宫游戏的说明如下: 不包括游戏起点, 共有 12 个游戏房间, 每个游戏房间都提供不同的游戏. 迷宫中的门称为滑动门, 它会随机地将游戏者带到两个可能的游戏房间中的一个. 例如, 从游戏起点, 游戏者通过滑动门可能进入房间 R01 或者房间 R02, 但在游戏过程中到底是进入了哪个房间, 游戏者自己并不能控制. 为了给游戏者以指引, 工作人员需要通过房间内的无线传感器或者工作人员的观察了解游戏者的位置. 其游戏规则为: 在某个房间中做对应的游戏并通过滑动门进入下一个房间.

迷宫中监控情况如下: 可以选择开启监控器监控两个相邻的房间, 监控器分别装在房间 R04 和房间 R09 之间以及房间 R05 和房间 R11 之间. 房间 R01 和房间 R07 的窗在同侧, 可以安排工作人员观察. 房间 R07 到房间 R12 中间都是落地窗, 可以安排工作人员观察. 但是,

安排工作人员需要成本,开启传感器也需要成本,需要尽可能地在保证能够指引游戏者的同时降低成本.在游戏中,假设每次只允许 1 个游戏者进入游戏,并且进入时监控者总是知道的.

这个游戏就是一个已知规划解而让游戏者去执行这个规划解的过程,节约成本则是一个观测约简的问题.在进行观测约简时,往往只需在动作的不确定效果中进行区分,即仅需判断当前状态是少数几个状态中的哪一个.例如,游戏者从起点第 1 次推开滑动门,要么到了房间 01,要么去了房间 02,因此只需安排工作人员观察是否在房间 01 就可以了.在以前的工作^[11]中,将这些需要通过观测进行区分的状态组合称为状态对,本文也遵从这种约定.

例 2:在例 1 中需要通过观测进行区分的状态对包括:

$state-pair_0 = \{ \text{游戏者在房间 R01 中}, \text{游戏者在房间 R02 中} \};$

$state-pair_1 = \{ \text{游戏者在房间 R03 中}, \text{游戏者在房间 R04 中} \};$

$state-pair_2 = \{ \text{游戏者在房间 R05 中}, \text{游戏者在房间 R06 中} \};$

$state-pair_3 = \{ \text{游戏者在房间 R07 中}, \text{游戏者在房间 R08 中} \};$

$state-pair_4 = \{ \text{游戏者在房间 R09 中}, \text{游戏者在房间 R10 中} \};$

$state-pair_5 = \{ \text{游戏者在房间 R11 中}, \text{游戏者在房间 R12 中} \}.$

例 1 中可选的观测包括:

$v_0 =$ 观测房间 R07,R08,R09,R10,R11,R12 的工作人员;

$v_1 =$ 观测房间 R01 和房间 R07 的工作人员;

$v_2 =$ 观测房间 R04 和房间 R09 的无线传感器;

$v_3 =$ 观测房间 R05 和房间 R11 的无线传感器.

假设所有的观测变量代价相同,可以定义最小观测集(MOS)如下:

定义 6(区分状态对). 设 V 是观测变量集合,称两个需要被区分的状态 s_0 和 s_1 为一个状态对 $state-pair = \langle s_0, s_1 \rangle$,其中 $s_0 \in S, s_1 \in S$. 称该状态对是可以被 $v \in V$ 区分的,当且仅当 $X(s_0, v) \neq X(s_1, v)$. 其中, X 是观测函数.

图 2 是例 2 中各观测变量及其能够区分的状态对之间的对应关系.比如, v_2 是装在房间 R04 和房间 R09 之间的传感器,当它开启时,能够同时看到房间 R04 和房间 R09 中的情况,因此就能判断当游戏者从房间 R01 往下走的时候是在房间 R03 还是在房间 R04,或者当游戏者从房间 R04 往下走的时候是在房间 R09 还是在房间 R10.

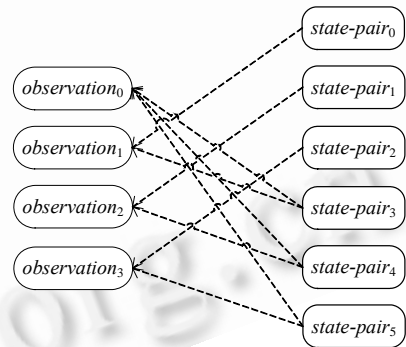


Fig.2 Observation variables in Example 1

图 2 例一中的观测变元

定义 7(观测集(observation set)). 设 $\Delta = \{ state-pair_0, state-pair_1, \dots, state-pair_{n-1} \}$ 是 n 个状态对的集合, $V = \{ v_0, v_1, \dots, v_{m-1} \}$ 是 m 个观测的集合. 对 V 的一个子集 A , 如果 Δ 中每个状态对都可区分, 则称 A 是 Δ 的一个观测集.

定义 8(最小观测集(MOS)). 设 $\Delta = \{ state-pair_0, state-pair_1, \dots, state-pair_{n-1} \}$ 是 n 个状态对的集合, $V = \{ v_0, v_1, \dots, v_{m-1} \}$ 是 m 个观测的集合, A 是 Δ 的一个观测集. 若对 Δ 的任意观测集 A' , 总有 $|A| \leq |A'|$, 则 A 就是 Δ 的最小观测集, 简记为 MOS.

在无二义性的前提下,求解最小观测集的问题也简写为 MOS.

使用贪心算法^[11]在例 1 中找到的 MOS 需要使用所有 4 个观测变量(因为 v_0 能够区分最多的状态对,会被优先选取),而实际上采用 $\{ v_1, v_2, v_3 \}$ 作为 MOS 就足够了.所以,贪心算法不能保证找到最小观测集.但每个观测的代价可能是不相同的,因此寻找最小观测集对于节约规划执行的成本是有利的.

1.3 最小覆盖集(MSC)和最小控制集(MDS)

定义 9(最小覆盖集(MSC)). 设集合 $C = \{ C_0, \dots, C_n \}$ 是集合 S 的幂集,即 $\forall C_i \subseteq S (0 \leq i \leq n)$. 若存在 $C' \subseteq C$, 且对

$\forall u \in S, \exists c \in C'$ 使得 $u \in c$, 则称 C' 是 S 的一个覆盖. 如果存在 S 的一个覆盖 C' , 对任意 S 的覆盖 C'' 满足 $|C'| \leq |C''|$, 则称 C' 是 S 的最小覆盖集.

定义 10(图). 图 G 是有序三元组 $(V(G), E(G), \Psi_G)$, 其中, $V(G)$ 是非空顶点集合, $E(G)$ 是与 $V(G)$ 不相交的边集合, Ψ_G 则是联系 G 中的边和一对无序顶点之间的关联函数. 若 e 是 G 中的一条边, 而 u 和 v 是 G 中的顶点, 满足 $\Psi_G(e) = uv$, 则称 e 是连接 u 和 v 的边, u 和 v 与 e 相关联, 顶点 u 和 v 称为 e 的端点.

定义 11(完全图). 任意两个顶点之间都有弧相连的图称为完全图.

定义 12(最小控制集(MDS)). 图 $G = (V(G), E(G), \Psi_G)$, 集合 $S \subseteq V(G)$ 称为 G 的控制集, 当且仅当 G 中任意一边至少有 1 个端点在 S 中. 若 G 中不存在控制集 S' 使得 $|S'| < |S|$, 则称 S 为最小控制集.

最小覆盖集问题和图论中的最小控制集(MDS)问题是等价的. 可以用如下办法将 MSC 映射为 MDS: 对 MSC 的 C , 构造一个完全图 C_k , 其中 C 的每个元素对应一个点, 然后在 C_k 中对应每个 S 中的元素添加一个点, 并且在该点和包含它的 C 的元素对应的点之间增加边, 就完成了映射. 反过来, 对应每个 MDS 图中的点构造一个集合, 它的元素由自身以及所有与这个点邻接的点所对应的元素构成, 就形成了一个 MSC.

MSC 和 MDS 都已被证明是 NP 难题^[16]. 求解 MDS/MSC 的方法包括穷举、贪心算法以及基于搜索树的方法^[17]等, 同时也有一些近似算法^[18]. 目前最优的完全算法仍然是指数级的^[17]. 控制集目前也是传感器网络优化中的重点^[19]. 但是, 传感器网络优化与 MOS 相比至少有如下 3 点不同: (1) 传感器网络优化要求监控的对象是所有可能的状态, 而 MOS 仅仅要区分不确定规划解(策略 π) 执行过程中产生的状态; (2) 传感器网络优化要求所有的监控点之间是连通的^[19]; (3) 传感器网络优化一般不要求最优解, 而是采用近似解^[19].

2 求MOS的算法

对于 MOS 可以使用贪心算法得到近似解, 也可以采用穷举的方法得到最优解. 下面根据目前最优的 MSC 算法^[17]提出一种复杂性与其等价且能保证最优解的方法.

2.1 引理

为了便于研究, 定义如下一个观测的区分集:

定义 13(区分集(division set)). 一个观测 $v \in V$ 的区分集是一个状态对的集合 $\Delta_v = \{state\text{-pair} | state\text{-pair} \text{ 可以被 } v \text{ 区分的状态对}\}$.

将一个观测 $v \in V$ 的区分集 Δ_v 看成在状态对的全集 Δ 上的一个子集, MOS 问题就是一个特殊的 MSC 问题, 可见, MOS 问题也是 NP 难题. 一般 MCS/MDS 问题仅需求最小覆盖集、最小控制集的秩, 但 MOS 还要给出拥有最小秩的具体的观测变量集合, 因此求 MOS 时需要增加对变量集合的调整, 而复杂性也随之增加.

基于 MCS/MDS 的一些研究成果, 可以发现 MOS 问题有如下性质:

引理 1. 对 MOS 问题,

- 1) 若存在两个不同的观测 v' 和 v'' , 其区分集分别为 $\Delta' \subseteq \Delta$ 和 $\Delta'' \subseteq \Delta$, 且 $\Delta' \subseteq \Delta''$, 则 MOS 中一定可以不包含 v' ;
- 2) 若 $\exists u \in \Delta$, 有且仅有唯一的 $v \in V$, 使得 u 可以被区分, 则 MOS 中一定包含 v .

证明: 1) 设包含 v' 的 MOS 为 $M, v' \in M$. 如果 $v'' \in M$, 由于 $\Delta' \subseteq \Delta''$ 则显然 $M - \{v'\}$ 是一个更小的观测集, 这与定义 8 矛盾. 因此 $v'' \notin M$, 用 v'' 代替 v' , 同样, 由于 $\Delta' \subseteq \Delta''$, 则 $M - \{v'\} \cup \{v''\}$ 也是观测集, 且秩与 M 相等, 即 MOS. 因此, MOS 中一定可以不包含 v' .

2) 若存在不包含 v 的 MOS, 不妨记为 M , 则 M 中不存在观测能够区分 u , 于是与定义 8 矛盾. □

由于一般会用尽可能少的传感器来监控所有可能的状态^[19], 所以一般状态对 u 最多被两个观测 v' 和 v'' 区分. 若 $v' = v''$, 则可用引理 1 中的结论 2 将 v' 加入 MOS; 否则, 设 v' 和 v'' 的区分集有交集 Δ , Δ 中的状态对只能由 v' 或 v'' 来区分. 若 v' 和 v'' 均尚未被选中到 MOS, 则在选中其他观测后, v' 和 v'' 对应的在待区分状态对中的区分集会逐渐变小, 最终总有一个会首先退化为 Δ , 则此时可用引理 1 中的结论 1 对 MOS 进行断言. 由此可见, 引理 1 在实际的观测约简中有广泛的应用.

2.2 算 法

算法假设待区分的状态对、观测变量以及观测函数已知.算法先根据观测函数找到观测对应的区分集,这可以通过调用 *constructDivisionSet* 函数实现.然后使用递归程序 *dpsMOS* 来寻找 MOS.其中当规模较小(状态对少于 3 个)时调用穷举子函数 *enumMos*,并在递归过程中用函数 *fixSet* 调整所有变量集合.流程如图 3 所示.

在完全可观测的不确定规划问题中,在规划执行过程中产生的一个需要区分的状态对至少可以由 1 个观测来进行区分.因此,首先要找出每个观测 v_i 对应的区分集 A_i ,如图 4 所示(算法 1).

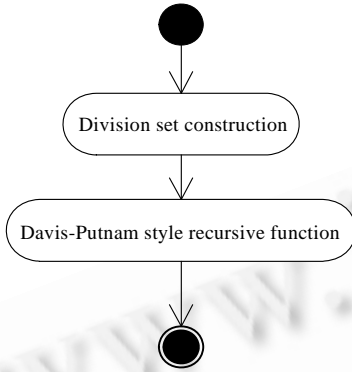


Fig.3 Overall MOS procedure
图 3 求 MOS 的整体流程

Algorithm 1. Division set construction (*constructDivisionSet*)
Input: state pair set $\Delta=\{state\text{-}pair_0, state\text{-}pair_1, \dots, state\text{-}pair_n\}$,
 and observation set $V=\{v_0, v_1, \dots, v_m\}$
Output: division set $\Phi=\{A_0, A_1, \dots, A_m\}$
Begin
 1. Let $A_j=\emptyset$ ($0\leq j<m$);
 2. For each $state\text{-}pair_i$ ($0\leq i<n$) in Δ , suppose $state\text{-}pair_i=\langle s_{i0}, s_{i1} \rangle$
 3. For each v_j ($0\leq j<m$) in V
 4. if $(X(s_{i0}, v_j) \neq X(s_{i1}, v_j))$
 then add $state\text{-}pair_i$ into A_j ;
 5. return Φ ;
End

Fig.4 Observe set construction algorithm
图 4 观测集构造算法

显然,算法 1 的复杂性是 nm .然后,根据引理 1 可以使用一个递归的程序来求 MOS,如图 5 所示.

Algorithm 2. Davis-Putnam style MOS (*dpsMOS*).
Input: state pair set $\Delta=\{state\text{-}pair_0, state\text{-}pair_1, \dots, state\text{-}pair_n\}$, and observation set $V=\{v_0, v_1, \dots, v_m\}$, and
 division set $\Phi=\{A_0, \dots, A_m\}$, $MOS=\{v_i, \dots, v_j\}$ ($0\leq i\leq j\leq m$).
Output: $MOS=\{v_i, \dots, v_j\}$ ($0\leq i\leq j\leq m$).
Begin
 1. if $(|\Delta|=0)$ then return MOS ;
 2. if $(|V|=1)$ then return $MOS=MOS\cup V$;
 3. if $(\exists A_j, A_k \in \Phi: A_j \subseteq A_k)$ return *dpsMOS*($\Delta, V-\{v_j\}, \Phi-A_j, MOS$);
 4. if $(\exists state\text{-}pair_i \in \Delta(0\leq i\leq n), \exists$ a unique $v_j \in V$ and $state\text{-}pair_i \in A_j)$ return *dpsMOS*(*fixSet*(Δ, V, Φ, MOS, j));
 5. if $(|\Delta|\leq 2)$ then *enumMos*(Δ, V, Φ, MOS);
 6. Find $A_j \in \Phi$, and $|A_j|\geq |A_k|$ for any $A_k \in \Phi$ ($k\neq j$);
 7. $MOS'=\textit{dpsMOS}(\Delta, V-\{v_j\}, \Phi-A_j, MOS)$;
 8. $MOS''=\textit{dpsMOS}(\textit{fixSet}(\Delta, V, \Phi, MOS, j))$;
 9. If $(|MOS'|<|MOS''|)$ return MOS' ;
 10. else return MOS'' ;
End

Fig.5 Davis-Putnam style MOS algorithm
图 5 Davis-Putnam 风格的 MOS 算法

在图 5 中,*dpsMOS* 算法首先排除那些被其他区分集包含的区分集以及它们对应的观测(Step 3),因为根据引理 1 它们肯定可以不被包含在 MOS 中.然后会优先选择那些能够唯一区分某一状态对的观测(Step 4),根据引理 1,这些观测肯定在 MOS 中.接着,找一个拥有最大秩的区分集的观测 v_j ,分两种情况来递归调用 *dpsMOS*:一种是不考虑 v_j 求原问题的 MOS(Step 7),一种是求 $v_j \in MOS$ 后剩余问题的 MOS(Step 8).选择两种情况中较小的 MOS

作为原问题的解.与一般的动态规划算法类似,算法 2 对问题规模较小的情况使用直接的求解方法(Step 5),这里采用枚举的方法(算法 4).图 6 中 *fixSet* 函数把 v_j 加入 *MOS* 后对剩余问题进行精简,可以由 v_j 区分的状态不再需要区分,如果需要可用对区分集有序的存储来合并具有相同区分集的观测.

Algorithm 3. Fix sets (*fixSet*).

Input: state pair set $\Delta = \{\text{state-pair}_0, \text{state-pair}_1, \dots, \text{state-pair}_n\}$, and observation set $V = \{v_0, v_1, \dots, v_m\}$,
and division set $\Phi = \{\Delta_0, \dots, \Delta_m\}$, $MOS = \{v_i, \dots, v_j\}$ ($0 \leq i \leq j \leq m$), index k of observation v_k .

Output: state pair set $\Delta = \{\text{state-pair}_0, \text{state-pair}_1, \dots, \text{state-pair}_{n'}\}$, and observation set $V = \{v_0, v_1, \dots, v_{m'}\}$,
and all division set $\Phi = \{\Delta_0, \dots, \Delta_{m'}\}$, $MOS = \{v_i, \dots, v_j\}$ ($0 \leq i \leq j \leq m'$).

Begin

1. $\Delta = \Delta - \Delta_k$;
2. $V = V - \{v_k\}$;
3. $\Phi = \Phi - \Delta_k$;
4. $MOS = MOS \cup \{v_k\}$;
5. for every state-pair $s \in \Delta_k$
6. for every division set Δ_i
7. {if ($s \in \Delta_i$) then $\Delta_i = \Delta_i - \{s\}$;
8. if ($|\Delta_i| = 0$) then { $\Phi = \Phi - \Delta_i$; $V = V - \{v_i\}$;}}
9. return Δ , V , Φ , MOS ;

End

Fig.6 Fix set sub-function algorithm

图 6 调整集合的子函数算法

Algorithm 4. Enumerate MOS (*enumMos*).

Input: state pair set $\Delta = \{\text{state-pair}_0, \text{state-pair}_1\}$, and observation set $V = \{v_0, v_1, \dots, v_m\}$,
and division set $\Phi = \{\Delta_0, \dots, \Delta_m\}$, $MOS = \{v_i, \dots, v_j\}$ ($0 \leq i \leq j \leq m$).

Output: $MOS = \{v_i, \dots, v_j\}$ ($0 \leq i \leq j \leq m$).

Begin

1. if ($\exists \Delta_j, \Delta_j = \Delta$) then return $MOS = MOS \cup \{v_j\}$;
2. for every division set Δ_j
3. if $\text{state-pair}_0 \in \Delta_j$
4. then { $MOS = MOS \cup \{v_j\}$;
5. break;}
6. for every division set Δ_j
7. if $\text{state-pair}_1 \in \Delta_j$
8. then { $MOS = MOS \cup \{v_j\}$;
9. break;}
10. return MOS ;

End

Fig.7 Enumerate MOS sub-function algorithm

图 7 基于穷举的 MOS 子函数算法

在图 7 中, *enumMos* 函数对于待区分状态对的数目小于 3 个的情况使用穷举的方法.这里不作赘述,直接认为 *enumMos* 在多项式时间内返回正确的解.

2.3 例子

例 3:在例 2 中,使用 *dpsMOS* 算法的过程如下:

1) 根据算法 1 得到 $\Phi = \{A_0, A_1, \dots, A_3\}$, 其中:

$$A_0 = \{\text{state-pair}_3, \text{state-pair}_4, \text{state-pair}_5\},$$

$$A_1 = \{\text{state-pair}_0, \text{state-pair}_3\},$$

$$A_2 = \{\text{state-pair}_1, \text{state-pair}_4\},$$

$$A_3 = \{\text{state-pair}_2, \text{state-pair}_5\};$$

2) 第 1 次调用算法 2,发现 *state-pair*₀ 仅能被 v_1 区分,于是将 v_1 放入 *MOS*;

3) 第 2 次调用算法 2,发现 *state-pair*₁ 仅能被 v_2 区分,于是将 v_2 放入 *MOS*;

4) 第 3 次调用算法 2,发现 *state-pair*₂ 仅能被 v_3 区分,于是将 v_3 放入 *MOS*;

5) 第 4 次调用算法 2,发现 $|A|=0$;

6) 返回 $MOS = \{v_1, v_2, v_3\}$.

可见,算法 2 能够给出正确的 *MOS*.

2.4 算法分析

定理 1. 算法 2 是正确的.

证明:1) 算法 2 得到的解能够区分所有状态对.这是因为,算法的终止有 3 种情况:要么因为待区分状态对集合为空,要么由 *enumMos* 返回了一个解,要么在包含或者不包含特定观测变量两种选择下返回较小的观测集.如果待区分状态对集合为空,那么找到的观测变量集合能够区分所有状态对.当待区分状态对的数目小于 3 时,*enumMos* 函数通过枚举能够返回一个观测集.对于第 3 种终止情况,两种选择下,较小的观测集即为原问题的观测集.因此,算法 2 得到的解是一个观测集,能够区分所有的状态对.

2) 算法 2 得到的观测集是最小的.使用数学归纳法证明如下:

设 N_v 表示 V 中观测变量的个数,即 $N_v = |V|$.

当 $N_v = 1$ 时, V 中仅包含 1 个观测变量,记为 v_0 .对于一个 *MOS* 问题 (A, V, Φ, MOS) ,根据完全可观测的前提, A 中任何一个状态对都至少可以由 1 个观测来区分,因此 v_0 可以区分 A 中所有状态对.根据算法 2 的 Step 2, $MOS = \{v_0\}$.显然此观测集是最小的.

设 $N_v \leq m$ 时,算法 2 得到的观测集是最小的.当 $N_v = m + 1$ 时,对于一个 *MOS* 问题 (A, V, Φ, MOS) ,首先可以根据引理 1 来进行化简.如果 *MOS* 问题中包含能够唯一区分一个状态对的观测,或者存在两个不同的观测 v' 和 v'' ,其区分集分别为 $A' \subseteq A$ 和 $A'' \subseteq A$,且 $A' \subseteq A''$,则可以通过算法 2 的 Step 3~Step 4 将原问题简化为规模更小的 *MOS* 问题 ($N_v \leq m$),根据假定,算法 2 得到的观测集是最小的.假设经过此化简步骤后, N_v 仍然为 $m + 1$.在算法 2 的 Step 9~Step 10,算法返回的 *MOS* 是 MOS' 和 MOS'' 中秩较小的一个,其中, MOS' 是不考虑观测变量 v_j 时原问题的解, MOS'' 是将 v_j 加入到 *MOS* 后剩余问题的解.无论是 MOS' 还是 MOS'' ,都是在问题规模更小 ($N_v \leq m$) 的情况下,算法 2 找到的观测集,因此根据假定, MOS' 和 MOS'' 是最小观测集,所以 *MOS* 也是原问题的最小观测集.

综上所述,算法 2 得到的解是最小观测集,因而算法 2 是正确的. □

从构造观测集的过程以及定义 5 与定义 6 可以看出,对一个具体问题来说,观测数量是固定的,而待区分状态对的个数是与具体规划解相关的.尽管算法 2 的复杂性本应由待区分状态对个数和观测的个数共同决定,但是因为通常规划解是未知的而观测是已知的,所以这里需要一个与具体的规划解无关而仅使用观测的个数来进行的复杂性分析.

定理 2. 算法 2 的复杂性不超过 $O(2^m m^2)$ 且不低于 $\Omega(2^{m-1})$, 其中 m 为观测的个数.

证明:设算法运行时间为 $T(n, m)$, 其中 n 为待区分的状态对个数而 m 为观测的个数.先证算法运行时间为 $T(n, m) = O(2^m m^2)$.

首先,易见算法 3 的复杂性是 $O(nm)$, 其中 n 为待区分的状态对数, m 为观测变量数.算法 4 的复杂性不超过

2n.在最坏情况下,MOS 问题中不包含能够唯一区分一个状态对的观测,也不存在两个不同的观测 v' 和 v'' ,其区分集分别为 $A' \subseteq A$ 和 $A'' \subseteq A$,且 $A' \subseteq A''$.否则,原 MOS 问题可以在线性时间内化简为规模更小的 MOS 问题.算法 2 简化为针对是否包含一个特定观测的两个递归调用,即 Step 6~Step 8.在每次迭代中都能找到这样的 v_j ,使得对于任意的 $v_k, A_k \in \Phi(k \neq j)$,有 $|A_j| \geq |A_k|$.

设 v_j 的秩为 i ,则算法运行时间 $T(n, m)$ 有如下计算公式(其中 C_1, C_2 均为常数):

$$T(n, m) \leq \begin{cases} C_1, & n < 3 \text{ 或 } m < 2 \\ T(n, m-1) + T(n-i, m-1) + C_2 \cdot n \cdot m, & n \geq 3 \text{ 且 } m \geq 2 \end{cases} \quad (1)$$

公式(1)中复杂性为常数的情况分别对应算法 2 中 Step 2 和 Step 5 的情况.公式(1)可以进一步放大:

$$T(n, m) < 2 \cdot T(n, m-1) + C_2 \cdot n \cdot m \quad (2)$$

考虑 n 的情况,由于观测变量是布尔类型的,即对一个给定的待区分状态对,每个观测要么能够区分要么不能区分.若 $2^m < n$,则总有两个以上待区分状态对有相同的观测集,此时可以从这些状态对中任选一个(可在输入的同时做判断).这样,初始待区分状态对总数总满足 $n \leq 2^m$,且由于算法 3 中合并了具有相同区分集的观测使得 $n \leq 2^m$ 总被满足.同时由于 $T(n, m)$ 是 n 的单调增函数,公式(2)可以化简为与 n 无关的公式(3),令 $S(m) = T(n, m)$.

$$S(m) < 2 \cdot S(m-1) + C_2 \cdot 2^{m-1} \cdot m \quad (3)$$

公式(3)中, $S(m)$ 即 $T(n, m)$ 被递归调用的次数不超过 $m-1$ 次,扩展此递推式可得:

$$S(m) < 2^{m-1} \cdot C_1 + \sum_{i=0}^{m-2} 2^i \cdot C_2 \cdot 2^{m-i} \cdot (m-i) = 2^{m-1} \cdot C_1 + 2^m \cdot C_2 \sum_{i=0}^{m-2} (m-i) < 2^{m-1} \cdot C_1 + 2^{m-1} \cdot C_2 \cdot m^2 \quad (4)$$

即算法 2 的复杂性是 $O(2^{m-1} m^2)$,一般地,在算法分析中记为 $O(2^m m^2)$.

再证明 $T(n, m) = \Omega(2^{m-1})$.

根据 MOS 与 MDS 的相似性,可以如下构造例子,说明算法在某些情况下会达到 $\Omega(2^{m-1})$ 的复杂性.

对一个完全图 $G=(V(G), E(G), \Psi_G)$ 构造一个 MOS 问题 (A, V, Φ, MOS) . 对 $\forall v \in V(G)$ 构造一个且仅有一个观测 $v_i \in V$, 其中 $0 \leq i < m$; 对 $\forall e \in E(G)$ 构造一个状态对 $sp, \Psi_G(e) = uv$ 表示 sp 可以由顶点 u 和 v 分别对应的观测来进行区分, Φ 表示每个观测对应的区分集的集合.于是获取 MOS 的过程就是寻找该图上 MDS 的过程,选取一个观测进入 MOS 的动作对应为在图中删除一个顶点以及与之相邻的边,当 $|E(G)|=0$ 时就得到了 MOS.完全图可以保证上述的最坏情况,即不存在引理 1 中的两种情况.完全图中有 m 个顶点,需要删除 $m-1$ 个顶点才能得到 MOS.于是, $T(n, m) = \Omega(2^{m-1})$. \square

3 求 OOS 的算法

在现实中进行观测的代价不可能总是相同的,因此需要找到一个代价最小的观测组合方案.

3.1 定义

定义 14(代价(COST)). 设 $A = \{state-pair_0, state-pair_1, \dots, state-pair_{n-1}\}$ 是状态对集合, $V = \{v_0, v_1, \dots, v_{m-1}\}$ 是观测集合.对 $v_i \in V (0 \leq i < m)$, 设其代价为 c_i, c_i 是一个非负实数,则得到一个与观测对应的代价集合 $C_V = \{c_0, c_1, \dots, c_{m-1}\}$. 对任意观测集 $A \subseteq V$, 其代价(cost of observation set, 简称 COST)是 $C_A = \sum_{v_j \in A} c_j$.

定义 15(最优观测集(OOS)). 设有 $A = \{state-pair_0, state-pair_1, \dots, state-pair_{n-1}\}$ 是状态对集合, $V = \{v_0, v_1, \dots, v_{m-1}\}$ 是观测集合, $C_V = \{c_0, c_1, \dots, c_{m-1}\}$ 是代价集合.对 A 的观测集 $A \subseteq V$, 若对任意 A 的观测集 $A' \subseteq V$ 有 $C_A \leq C_{A'}$, 则称 A 是最优观测集.

显然, MOS 是 OOS 的一个特例, 其中, $c_0 = c_1 = \dots = c_{m-1}$.

3.2 求 OOS 的算法

为了更好地描述算法, 定义关于某个状态对的观测集如下:

定义 16(状态对观测集(state-pair observation set)). 设 $A = \{state-pair_0, state-pair_1, \dots, state-pair_{n-1}\}$ 是状态对

集合, $V=\{v_0,v_1,\dots,v_{m-1}\}$ 是观测集合.对 $\forall sp=\langle s,s'\rangle\in\Delta$,关于 sp 的观测集为 $A_{sp}=\{v|v\in V \text{ 且 } X(s,v)\neq X(s',v)\}$.

整个求解算法是首先找出每个状态对的观测集(算法如图 8 所示),然后根据每个状态对都被区分的要求,构造整数规划(IP)约束不等式组,最后调用 IP 求解器(比如 MatLab)进行求解.整个流程如图 9 所示.

Algorithm 5. State-Pair observation set construction (*constructSPOS*).

Input: state-pair set $\Delta=\{state-pair_0, state-pair_1, \dots, state-pair_n\}$, and observation set $V=\{v_0, v_1, \dots, v_m\}$, and observe function X .

Output: $statepairObservationSet_j (0\leq j\leq n)$.

Begin

1. For each $state-pair_i(s_{i0}, s_{i1})$ in Δ
2. For each $v_j (0\leq j\leq m)$ in V
3. if $(X(s_{i0}, v_j)\neq X(s_{i1}, v_j))$ then add v_j into $statepairObservationSet_j$;
4. return $statepairObservationSet_0, \dots, statepairObservationSet_n$;

End

Fig.8 State-Pair observation set construction algorithm

图 8 状态对观测集构造算法

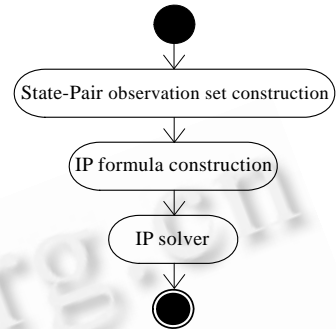


Fig.9 Overall OOS procedure

图 9 OOS 整体流程

IP 技术是求解最优化问题常用的手段^[14,15],其主要思想是通过一组整数变量所构成的约束来求某个目标函数的最大值或最小值.变量只能取 0 或者 1 的整数规划称为 0-1 规划.在完全可观测的前提下,OOS 问题中的每个状态对至少由 1 个观测来进行区分,因此在图 10 的算法中通过 Step 3 来建立这样的约束.由图 10 所构造的 $formula_j(0\leq j\leq n)$ 再加上目标函数 $\sum_{0\leq i\leq m} c_i x_i$,可以将 OOS 问题转化为整数规划问题.将这样的整数规划问题用 IP 求解器来求解,比如 Matlab 程序,可以得到原 OOS 问题的解.在得到的解中,如果 x_i 为 1 则表示对应观测 v_i 在 OOS 中,否则(为 0)表示不在.由于 OOS 问题中的变量均为布尔类型,所以它是一个 0-1 规划问题.由于 0-1 规划的可行域是有限的,所以不存在无界解的情况.进一步地,由整数规划的性质可知,只要有解就可以得到最优解.这里对整数规划的具体方法不作介绍.

Algorithm 6. IP formula construction (*constructIP*).

Input: observation set $V=\{v_0, v_1, \dots, v_m\}$, $statepairObservationSet_j (0\leq j\leq n)$.

Output: $formula_j (0\leq j\leq n)$.

Begin

1. For each observation $v_i (0\leq i\leq m)$ in V , assign a variable for it, namely $x_i (x_i=0 \text{ or } 1)$;
2. For each $statepairObservationSet_j=\{v_{j1}, v_{j2}, \dots, v_{jk}\}$, construct a formula for it, namely
3. $formula_j="x_{j1}+x_{j2}+\dots+x_{jk}>0"$;
4. return $formula_0, formula_1, \dots, formula_n$;

End

Fig.10 IP inequation construction algorithm

图 10 IP 不等式构造算法

3.3 例子

例 4:在例 1 中,假设增加了一个在房间 R01,R04,R05 中间的传感器,如图 11 所示.也就是说,增加了 v_4 =观测房间 R01,R04 和 R05 的无线传感器.其对应的区分集为 $\Delta_4=\{state-pair_0, state-pair_1, state-pair_2\}$,观测变量及其区分集如图 12 所示.假设使用任意一个传感器的代价相同,不妨假设为 1,使用任意一个工作人员的代价也相同,不妨假设为 10.于是,代价集合 $C_V=\{10,10,1,1,1\}$.

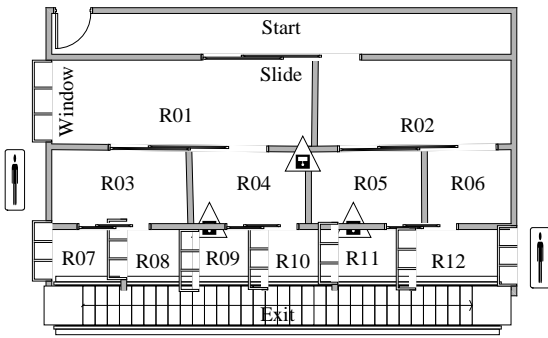


Fig.11 An extension of Example 1

图 11 例 1 的一个扩展

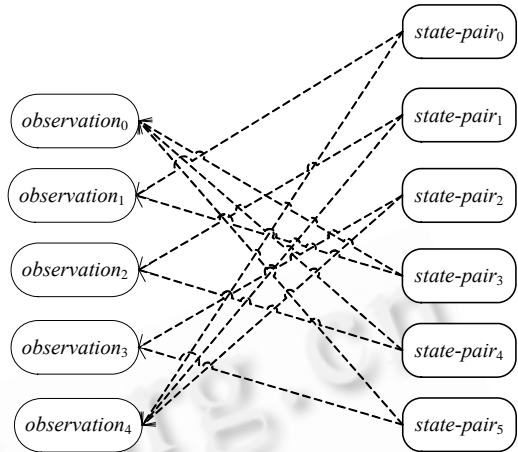


Fig.12 Observation variables

图 12 例 4 中的观测变量

根据每个状态对至少被 1 个观测区分的要求,基于状态对的观测集和观测总代价最小的目标可以建立如下的约束:

$$x_1+x_4>0 \tag{5}$$

$$x_2+x_4>0 \tag{6}$$

$$x_3+x_4>0 \tag{7}$$

$$x_1+x_0>0 \tag{8}$$

$$x_2+x_0>0 \tag{9}$$

$$x_3+x_0>0 \tag{10}$$

$$\text{Minimize } \sum_{0 \leq i < m} c_i x_i, m=5 \tag{11}$$

其中, x_0, \dots, x_4 只能取 0 或 1, 分别对应于观测 v_0, \dots, v_4 , 表示该观测是否属于 OOS. 不等式(5)~(10)分别说明 $state-pair_0, \dots, state-pair_5$, 需要至少被 1 个观测区分. 比如对 $state-pair_0$, 它可以被 v_1 或 v_4 区分, 那么 v_1 和 v_4 之间至少要有 1 个被选入 OOS. 所以 v_1 和 v_4 对应的变量之和必须大于 0. 不等式(11)就是要求 COST 最小. 求解这个 IP 不等式组可得:

$$x_0=x_4=1; x_1=x_2=x_3=0;$$

$$\sum_{0 \leq i < m} c_i x_i = 11.$$

其意义是, 在最佳观测组合方案中, 采用观测房间 R07, R08, R09, R10, R11, R12 的工作人员(v_0)和观测房间 R01, R04 和 R05 的无线传感器(v_4).

4 处理观测中的容错

实际应用中的容错要求驱动了学术界对容错系统和容错算法的研究^[20,21], 规划领域也出现了专门的容错规划器^[22]. 在观测约简问题中, 随着传感器数目的增加, 传感器出现故障的可能性也相应增大. 为了避免传感器故障造成的严重后果, 需要为系统提供一定的容错能力, 保证在部分传感器出错时仍然可以判定当前的系统状态, 以提高整个系统的可靠性.

定义 N 容错的观测集如下:

定义 16(N 容错的观测集). 设 \mathcal{A} 是状态对集合, \mathcal{V} 是观测集合. 如果对 \mathcal{V} 的一个子集 \mathcal{A} , 使得在 \mathcal{A} 中不多于 N 个观测出错时, \mathcal{A} 仍然是 \mathcal{A} 的观测集, 则称 \mathcal{A} 是 \mathcal{A} 的一个 N 容错的观测集.

定义 17(N 容错的最优观测集). 设 A 是状态对集合, V 是观测集合, C_V 是代价集合. 对 N 容错的观测集 $A \subseteq V$, 若对任意 N 容错的观测集 $A' \subseteq V$ 有 $C_A \leq C_{A'}$, 则称 A 是 A 的 N 容错的最优观测集.

一个观测有两种出错的方式: 无响应或者返回错误的观测值. 判断一个观测返回的值是错误的只能通过比较更多的观测返回的值. 一般来说会对所有观测同一范围(本文中为同一状态对)的观测进行排序, 从中选出奇数个来进行表决. 表决采用少数服从多数的原则, 即如果超过半数的观测返回同一个观测值, 则认为该观测值是正确的. 于是, 如果需要对 N 个观测出错的情况进行容错, 就需要有 $N+1$ 个其他观测. 即观测同一范围(本文中为同一状态对)的观测总数应大于 $2N$. 可令 IP 约束不等式的右端为 $2N$, 此时得到的观测组合方案即为 N 容错的 OOS. 其中对无界解和最优解的情况分析与一般 OOS 的分析相同.

例 5: 在例 4 中, 假设增加了一个装在房间 R01 和 R03 之间的传感器, 增加了一个在房间 R02 和 R06 之间的传感器, 并且增加了一个巡视 R07, R08, R09, R10, R11, R12 的工作人员, 如图 13 所示. 即增加了:

- v_5 =观测房间 R01 和房间 R03 的无线传感器;
- v_6 =观测房间 R02 和房间 R06 的无线传感器;
- v_7 =观测房间 R07, R08, R09, R10, R11, R12 的工作人员.

其对应的区分集为

$$A_5 = \{state-pair_0, state-pair_1\};$$

$$A_6 = \{state-pair_0, state-pair_2\};$$

$$A_7 = A_0.$$

观测变量及对应的区分集如图 14 所示.

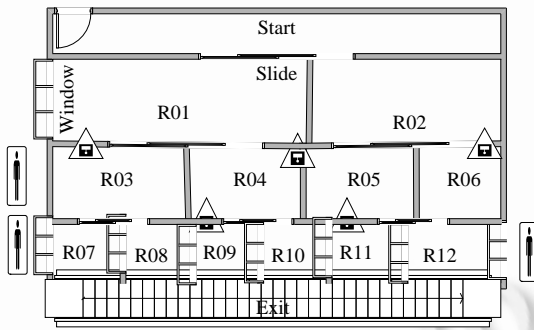


Fig.13 An extension of Example 4

图 13 例 4 的一个扩展

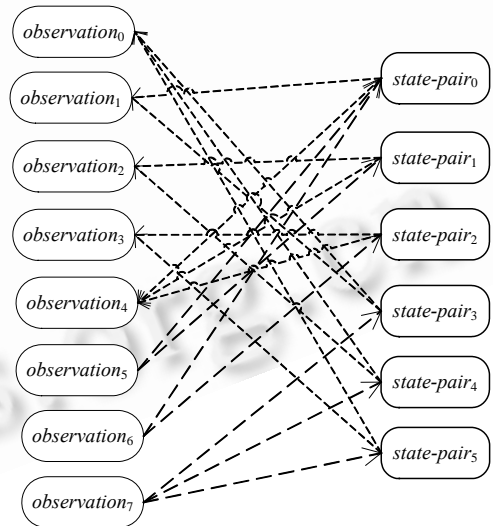


Fig.14 Observation variables in Example 5

图 14 例 5 中的观测变量

设使用任意一个传感器的代价相同, 假设为 1, 使用任意一个工作人员的代价也相同, 不妨假设为 10. 于是,

$$V_c = \{10, 10, 1, 1, 1, 1, 1, 10\}.$$

构造整数规划的约束不等式如下:

$$x_1 + x_4 + x_5 + x_6 > 2 \tag{12}$$

$$x_2 + x_4 + x_5 > 2 \tag{13}$$

$$x_3 + x_4 + x_6 > 2 \tag{14}$$

$$x_1 + x_0 + x_7 > 2 \tag{15}$$

$$x_2+x_0+x_7>2 \quad (16)$$

$$x_3+x_0+x_7>2 \quad (17)$$

$$\text{Minimize } \sum_{0 \leq i < m} c_i x_i \quad (m=7) \quad (18)$$

求解这个 IP 不等式可得:

$$x_0=x_1=x_2=x_3=x_4=x_5=x_6=x_7=1, \\ \sum_{0 \leq i < m} c_i x_i = 35.$$

其意义是采用所有 5 个传感器和 3 个工作人员来进行观测,可以得到一个 1 容错的最优观测集,即任意一个传感器或者工作人员出错都可以保证观测是正确的.根据只能取 0 或 1 的整数变量的定义,不可能满足“ $x_1+x_0+x_7>4$ ”约束,即例 5 只能 1 容错.

定义 18(观测的错误率). 设 $V=\{v_0, v_1, \dots, v_m\}$ 是观测的集合,对 $v \in V$,其错误率为

$$\delta = \frac{\text{观测失效或者出错的次数}}{\text{进行观测的总次数}}.$$

对于 N 容错的观测系统,如果不多于 N 个观测出错时,则称该系统是可靠的.

定理 3. 假定每个观测的错误率均为 δ ,且观测之间是独立的.那么 N 容错的观测系统的可靠性为

$$\sum_{k=0}^N C_{2N+1}^k \delta^k (1-\delta)^{2N+1-k}.$$

证明: N 容错的观测系统是可靠的,当且仅当不多于 N 个观测同时出错.根据概率论, $k(0 \leq k \leq N)$ 个观测出错的概率为 $C_{2N+1}^k \delta^k (1-\delta)^{2N+1-k}$,其中 C_{2N+1}^k 为 $2N+1$ 个观测中 k 个观测出错的组合数.因此,整个 N 容错的观测系统的可靠性 $\sum_{k=0}^N C_{2N+1}^k \delta^k (1-\delta)^{2N+1-k}$.

显然,在 δ 一定的情况下, N 越大,系统越可靠.类似地可以证明,如果这 $2N+1$ 个观测的错误率各不相同,设分别为 $\delta_1, \dots, \delta_{2N+1}$,令 $\delta_{\min} = \min\{\delta_1, \dots, \delta_{2N+1}\}$, $\delta_{\max} = \max\{\delta_1, \dots, \delta_{2N+1}\}$,那么, N 容错的观测系统的可靠性大于 $\sum_{k=0}^N C_{2N+1}^k \delta_{\min}^k (1-\delta_{\max})^{2N+1-k}$. □

5 总 结

本文可以看作是以前工作的进一步扩展^[11],更深入和全面地研究了观测约简的问题.一方面在观测的代价相同的情况下,本文提出了 MOS 问题与 MSC/MDS 问题的相似性,并据此提出了解决 MOS 问题的非枚举算法.同时,将简化的 MOS 问题映射为 MDS 问题,可以证明 MOS 问题也是 NP 难题.另一方面,如果观测的代价不相同,本文定义了最优观测集,并提出了用整数规划的方法来解决它.最后,本文还考虑了容错情况下最优观测集的概念和求法.从理论的角度来看,像以前的工作一样,本文也是用于规划执行阶段^[10,11],用以节省观测代价的.从应用的角度来看,本文研究的观测约简问题可以作为传感器网络优化的一种特殊情况^[19].

将一个问题转换为相应的数学模型,并利用该模型中已有的成果是解决现实问题的捷径.这是因为这些数学问题已经被深入研究了,即便不能直接套用到应用问题上,其中也会有很多可以借鉴的方法.另一方面,面对带有权值的优化问题,IP 技术始终是一种很好的选择.

致谢 感谢姜定俊教授和黄巍博士对本文的讨论.

References:

- [1] Jiang ZH, Jiang YF. An improved method for calculating activation sets of action derived preconditions. Chinese Journal of Computers, 2007,29(12):2061-2073 (in Chinese with English abstract).

- [2] Wu XJ, Jiang YF, Ling YB. Research on relations of effect of action for STRIPS domain. *Journal of Software*, 2007,18(6): 1328–1349 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/18/1328.htm>
- [3] Wu XJ, Jiang YF, Ling YB. Strategy of extracting domain knowledge for STRIPS world. *Journal of Software*, 2007,18(3): 490–504 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/18/490.htm>
- [4] Chen AX, Jiang YF, Zhang XN, Liu GY. GP: Genetic planning algorithm based on planning graph. *Chinese Journal of Computers*, 2007,29(1):153–160 (in Chinese with English abstract).
- [5] Lai ZF, Jiang YF. Learning action model in AI planning based on genetic algorithm. *Chinese Journal of Computers*, 2007,29(6): 945–953 (in Chinese with English abstract).
- [6] Weld D, Anderson C, Smith D. Extending graphplan to handle uncertainty and sensing actions. In: Edelkamp S, Korf RE, eds. *Proc. of the 13th National Conf. on Artificial Intelligence (AAAI'98)*. Madison: AAAI Press/MIT Press, 1998. 897–904.
- [7] Bonet B, Geffner H. Planning with incomplete information as heuristic search in belief space. In: Chien S, Kambhampati S, Knoblock CA, eds. *Proc. of the 5th Int'l Conf. on Artificial Intelligence Planning Systems (AIPS 2000)*. Breckenridge: AAAI Press, 2000. 52–61.
- [8] Pineau J, Roy N, Thrun S. A hierarchical approach to POMDP planning and execution. In: Brodley CE, Danyluk AP, eds. *Proc. of the 18th Int'l Conf. on Machine Learning (ICML 2001) Workshop on Hierarchy and Memory in Reinforcement Learning*. Williamstown: Morgan Kaufmann Publishers, 2001.
- [9] Cimatti A, Pistore M, Roveri M, Traverso P. Weak, strong, and strong cyclic planning via symbolic model checking. *Artificial Intelligence*, 2003,147(1-2):35–84.
- [10] Hoey J, Poupart P. Solving POMDPs with continuous or large discrete observation spaces. In: Kaelbling LP, Saffiotti A, eds. *Proc. of the 20th Int'l Joint Conf. on Artificial Intelligence (IJCAI 2005)*. Edinburgh: Morgan Kaufmann Publishers, 2005. 1332–1338.
- [11] Huang W, Wen ZH, Jiang YF, Wu LH. Observation reduction for strong plans. In: Veloso MM, ed. *Proc. of the 21st Int'l Joint Conf. on Artificial Intelligence (IJCAI 2007)*. Hyderabad, India: Morgan Kaufmann Publishers, 2007. 1930–1935.
- [12] Chen L, Tian KS. A multi-sensor management algorithm based on performance and cost. *Journal of Air Force Radar Academy*, 2004,18(2):18–20 (in Chinese with English abstract).
- [13] Liu GX. Minimizing cost redundance and optimizing algorithm for multisensor fusion system. *Journal of Southeast University (Natural Science Edition)*, 2000,30(1):69–72 (in Chinese with English abstract).
- [14] Cai ZP, Liu F, Zhao WT, Liu XH, Yin JP. Deploying models and optimization algorithms of network measurement. *Journal of Software*, 2008,19(2):419–431 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/19/419.htm>
- [15] Jiang H, Zhang XC, Chen GL. Directed black and white traveling salesman problem. *Chinese Journal of Computers*, 2007,29(3): 431–440 (in Chinese with English abstract).
- [16] Afif M, Hifi M, Paschos VT, Zissimopoulos V. A new efficient heuristic for the minimum set covering problem. *The Journal of the Operational Research Society*, 1995,46(10):1260–1268.
- [17] Fomin FV, Grandoni F, Kratsch D. Measure and conquer: Domination—A case study. In: Caires L, *et al.*, eds. *Proc. of the 32nd Int'l Colloquium on Automata, Languages and Programming (ICALP 2005)*. Berlin, Heidelberg: Springer-Verlag, 2005. 191–203.
- [18] Feige U. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 1998,45(4):634–652.
- [19] Jiang J, Fang L, Zhang HY, Dou WH. An algorithm for minimal connected cover set problem in wireless sensor networks. *Journal of Software*, 2006,17(2):175–184 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/17/175.htm>
- [20] Liu H, Fei SM. A fault-tolerant scheduling algorithm based on EDF for distributed control systems. *Journal of Software*, 2003, 14(8):1371–1378 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/14/1371.htm>
- [21] Li QH, Han JJ, Essa AA, Zhang W. Dynamic scheduling algorithms with software fault-tolerance in hard real-time systems. *Journal of Software*, 2005,16(1):101–107 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/16/101.htm>
- [22] Jensen RM, Veloso M, Bryant RE. Fault tolerant planning: Toward probabilistic uncertainty models in symbolic non-deterministic planning. In: Zilberstein S, Koehler J, Koenig S, eds. *Proc. of the 14th Int'l Conf. on Automated Planning and Scheduling (ICAPS 2004)*. Whistler: AAAI Press, 2004. 335–344.

附中文参考文献:

- [1] 蒋志华,姜云飞.一种计算动作派生前前提的激活集的改进方法.计算机学报,2007,29(12):2061-2073.
- [2] 吴向军,姜云飞,凌应标.STRIPS 规划领域中动作效果关系的研究.软件学报,2007,18(6):1328-1349. <http://www.jos.org.cn/1000-9825/18/1328.htm>
- [3] 吴向军,姜云飞,凌应标.基于 STRIPS 的领域知识提取策略.软件学报,2007,18(3):490-504. <http://www.jos.org.cn/1000-9825/18/490.htm>
- [4] 陈嵩祥,姜云飞,张学农,刘国英.GP——基于规划图的遗传规划算法.计算机学报,2007,29(1):153-160.
- [5] 赖志锋,姜云飞.智能规划中基于遗传算法的动作模型学习.计算机学报,2007,29(6):945-953.
- [12] 陈莉,田康生.一种基于性能和代价的多传感器管理算法.空军雷达学院学报,2004,18(2):18-20.
- [13] 刘贵喜.多传感器融合系统最小代价冗余及优化算法.东南大学学报(自然科学版),2000,30(1):69-72.
- [14] 蔡志平,刘芳,赵文涛,刘湘辉,殷建平.网络测量部署模型及其优化算法.软件学报,2008,19(2):419-431. <http://www.jos.org.cn/1000-9825/19/419.htm>
- [15] 江贺,张宪超,陈国良.有向黑白旅行商问题.计算机学报,2007,29(3):431-440.
- [19] 蒋杰,方力,张鹤颖,窦文华.无线传感器网络最小连通覆盖集问题求解算法.软件学报,2006,17(2):175-184. <http://www.jos.org.cn/1000-9825/17/175.htm>
- [20] 刘怀,费树岷.基于 EDF 的分布式控制系统容错调度算法.软件学报,2003,14(8):1371-1378. <http://www.jos.org.cn/1000-9825/14/1371.htm>
- [21] 李庆华,韩建军,Essa AA,张薇.硬实时系统中基于软件容错的动态调度算法.软件学报,2005,16(1):101-107. <http://www.jos.org.cn/1000-9825/16/101.htm>



饶东宁(1977-),男,广东兴宁人,博士生,工程师,主要研究领域为智能规划,图论.



姜云飞(1945-),男,教授,博士生导师,主要研究领域为定理机器证明,智能诊断,智能规划.



蒋志华(1978-),女,博士,讲师,主要研究领域为智能规划.



朱慧泉(1978-),男,博士生,主要研究领域为图论,游戏 AI,数据挖掘.