

基于启发式搜索的灵活规划的算法研究与系统实现^{*}

李 杨^{1,2} 陈佳豫^{1,2} 韩诚山¹ 孙 焱³ 谷文祥³

(中国科学院长春光学精密机械与物理研究所 长春 130033)¹ (中国科学院研究生院 北京 100049)²
(东北师范大学计算机学院 长春 130117)³

摘 要 随着智能规划研究的深入,经典规划已不能满足实际应用的需要。本文分析了经典规划无法满足实际应用要求及产生灵活规划的原因。在对启发式搜索和灵活规划深入研究的基础上,提出了利用启发式搜索的方法来处理灵活规划问题的思想,并给出了基于启发式搜索的灵活规划算法和求解模型。采用智能规划中的基准问题对该算法进行测试,实验表明该方法在处理很多领域问题上都可以得到非常好的效果。

关键词 人工智能,智能规划,灵活规划,启发式搜索

Research of Flexible Planning Algorithm and System Implement Based on Heuristic State Search

LI Yang^{1,2} CHEH Jia-Yu^{1,2} HAN Cheng-Shan¹ SUN Yan³ GU Wen-Xiang³

(Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033)¹
(Graduate School of the Chinese Academy of Sciences, Beijing 100049)²
(School of Computer Science, Northeast Normal University, Changchun 130117)³

Abstract Traditionally, planning problems are cast in terms of imperative constraints that are either wholly satisfied or wholly violated. In this paper, why classical planning can not capture the full subtlety of many real problems is argued. A new flexible planning problem is defined which supports the soft constraints often found in reality. A new concept using heuristic state search theory to solve flexible planning problem is described. This paper also introduces a novel solving model of flexible planning using heuristic state search theory and exploits the algorithm framework of flexible planning based on heuristic state search.

Keywords AI, Intelligent planning, Flexible planning, State heuristic search

1 引言

智能规划是一门涉及知识表达、知识推理、非单调逻辑、情景演算、人机交互和知识挖掘等各个方面知识的多领域交叉性学科^[1],被广泛运用于航空航天技术、机器人控制、后勤调度、游戏角色设计、系统建模等方面,带来的成果有目共睹。近几年来,随着客观条件的改善,世界上特别是一些发达国家在此领域获得了长足的发展,在国防和空间技术领域中得到成功应用,取得了巨大的经济和社会效益。

90年代,规划系统的研究达到高潮。目前解决规划问题主要有三种方法:第一种方法是 Blum 和 Furst 提出的 Graphplan 方法^[2];第二种方法是 Kautz 和 Selman 提出的约束可满足方法^[3];第三种方法是 Bonet 和 Geffer 提出的启发式搜索规划方法^[4]。用启发式函数产生规划实例的说明,用来指导规划在状态空间中的搜索。

随着规划的不断深入研究,经典规划存在的不足越来越明显。它对现实世界的许多问题,例如一些复杂的航空航天问题,都无法有效地刻画,因此产生了一种新的规划即灵活规划。用于处理这种灵活规划的规划器称为灵活规划器。规划器实际上是一个软件,在这里规划算法无疑成为规划器的核心,目前用于解决灵活规划的规划器所使用的算法多数是基

于 Graphplan 和 CSP 的。

本文提出了基于启发式搜索的灵活规划的算法。启发式搜索是在搜索中加入了与问题有关的启发式信息,用以指导搜索朝着最有希望的方向前进,加速问题的求解过程并最终找到最优解。在多次国际规划器比赛中发现,采用启发式知识的规划器比没有采用启发式知识的规划器得到的效果要好,表现出了很强的问题求解能力。基于以上启发式搜索和灵活规划的优点,把启发式思想运用到了灵活规划中,产生了新的有效算法,为灵活规划的求解开辟了新的途径。

本文的其它部分安排如下:第2节是本文的研究基础,主要介绍灵活规划的提出背景及灵活规划的相关知识和启发式搜索的知识;第3节提出基于启发式搜索的灵活规划的算法和求解模型;第4节是实验结果分析;最后是总结。

2 研究基础

2.1 灵活规划(Flexible Planning)概述

现实世界非常复杂,很多实际问题的约束条件灵活。经典规划是一种强约束问题,即对于约束条件或者完全满足或者完全不满足,操作的可用性在特定情况下是布尔的。这种框架对于刻画现实世界的很多问题太过于严格。为了解决上述情况,产生了灵活规划。灵活规划描述支持软约束,是对经

^{*}国家自然科学基金项目(编号为:60573067 和 60473042)。李 杨 博士研究生,主要研究方向为智能规划与规划识别、软件可靠性、软件工程;陈佳豫 博士研究生,主要研究方向为软件工程;韩诚山 研究员,硕士生导师,主要研究方向为空间相机控制方法研究;孙 焱 硕士研究生,主要研究方向为图像处理;谷文祥 教授,博士生导师,主要研究方向为智能规划与规划识别、形式语言与自动机理论、模糊数学及其应用。

典规划约束的一种放宽。利用灵活规划可以刻画很多经典规划所无法刻画的实际世界中的实际问题。但对于这种灵活规划,传统的布尔规划器无能为力,只能借助于灵活规划器。通过灵活规划器可以找到一个在布尔规划器中完全得不到的折衷规划。现在有很多研究人员都在致力于寻找通用的算法来处理这种灵活规划问题。Ian Miguel, Peter Jarvis 和 Shen Qiang 做了很深入的研究^[5,6]。目前解决灵活规划问题的规划器主要是应用 Graphplan 框架和结合 CSP 的,而利用启发式搜索理论来指导灵活规划的规划器还有待于进一步开发和完善。

2.2 启发式搜索理论概述

启发式搜索的针对性强,原则上只需要搜索问题的部分状态空间,因而效率较高。在搜索过程中,关键的一步是如何确定下一个要考察的节点,确定的方法不同就形成了不同的搜索策略。规划求解问题可以看成是状态空间搜索问题。把一个规划问题用状态空间表示法来表示,然后利用启发函数来指导搜索,找到规划解。

一个规划问题用状态空间表示法可以表示为一个初始状态集合 S_0 、目标集合 G 和确定的动作集 Ω 。每个动作 $a \in \Omega$ 都有一个前提列、添加效果列和删除效果列,分别用 $Prec(a)$, $Add(a)$ 和 $Del(a)$ 表示。一个规划问题的解是要找到一个动作的序列,通过执行这个动作序列可以从初始状态到达目标状态。

在正向状态空间搜索中,如果一个动作 a 的所有前提都出现在状态 S 中,即 $prec(a) \subseteq S$,则称动作 a 适用于状态 S 。状态 S 通过动作 a 向前扩展记为:

$$Progress(S, a) = S + Add(a) - Del(a)$$

状态 S 的启发函数是从状态 S 到达目标 G 的成本估计。

在逆向状态空间搜索中,如果 $Add(a) \cap S \neq \emptyset$ 并且 $Del(a) \cap S = \emptyset$,则称动作 a 适用于状态 S ,状态 S 通过动作 a 后退,记为

$$Regress(S, a) = S + Prec(a) - Add(a)$$

状态 S 的启发函数是从初始状态 S_0 到达状态 S 的成本估计。

常用的启发函数有“和代价法”、“最大代价法”、“取层号法”以及“和互斥代价法”等。

3 基于启发式搜索的灵活规划算法及求解模型

3.1 问题定义

灵活规划问题可以表示成与经典规划问题相似的四元组,但对操作和目标做了相应的修改,即对于相应的操作加入了满意度,对于命题加入了真值度。这样在灵活规划中,一个灵活问题 Ψ 就可以用一个四元组 $\langle \Phi; O; I; \Gamma \rangle$ 来描述^[6],其中的 Φ, O, I, Γ 分别用来描述规划对象、灵活操作,以及包含灵活命题的初始条件和灵活目标条件。

原来在经典规划里的布尔命题在这里被灵活命题 ρ 所代替。 ρ 的形式为 $(\rho \varphi_1, \varphi_2, \dots, \varphi_n k_i)$,其中 $\varphi_i \in \Phi, k_i$ 是一个完全有序集合 K 中的一个元素,其中 K 表示的是对于灵活操作的命题的主观真值度, K 由 $k_{\perp}, k_1, \dots, k_{\top}$ 组成。布尔命题的主观真值度用 K 的两个端点来刻画,即分别用 k_{\perp} 和 k_{\top} 来表示完全为假或者完全为真。

一个灵活操作, $o \in O$,解释了它的前提条件被满足的程度。灵活操作可以用从前提条件空间映射到全序满意度 L 的模糊关系和灵活效果命题集合来描述。 L 由有限的隶属度

$l_{\perp}, l_1, \dots, l_{\top}$ 组成,其中端点 $l_{\perp} \in L, l_{\top} \in L$ 分别表示不完全满足和完全满足。 L 与 K 描述的不同在于, L 表述的是在当前状态下要执行该操作所必须满足的折衷度,而 K 是指命题的主观真值度。

灵活规划目标 $\gamma \in \Gamma$ 是从灵活命题空间到 L 的映射。某种程度上满足规划目标的相互一致的命题集合可能不止一个,因此规划的满意度必须同时将灵活操作的满意度和目标的满意度都考虑进去。灵活规划的满意度应该通过考虑整个规划中所用到的每个操作的满意度和每个目标的满意度来定义。一个规划的质量是它的满意度和规划长度的合取,当两个规划具有相同的满意度时,规划短的更好。

3.2 互斥定义

将灵活规划中的互斥分为动作互斥和命题互斥。

1. 动作互斥

当同一层中两个动作实例满足以下三者中的任意一个时,称之为具有互斥关系。

1) 效果不一致:一个动作产生的效果的真值度与另一个动作产生的具有相同核心命题的效果的真值度不同。

2) 冲突:一个动作的一个效果命题的真值度和另一个动作需要的具有相同核心命题的前提表达不同的真值度。

3) 竞争需要:两个动作的前提在上一层相互冲突。

2. 命题互斥

当同一层的两个命题满足以下三者中的任意一个时,称之为具有互斥关系。

1) 一个命题是另一个命题的否定。

2) 不一致支持:获得此灵活命题的所有动作与获得另一灵活命题的所有动作都具有互斥关系。

3) 两个灵活命题包含的核心命题表达不一致的真值度。

3.3 启发函数定义

和互斥代价法是 HSP-r 的默认启发函数,HSP-r 是反向搜索规划器,因此它的定义适合于反向搜索的使用。而本算法采用的是正向搜索,所以对其做相应的改进,以适合于本算法的使用。改进后的和互斥代价法如下所示:

$$Heuristic H(S) \leftarrow \infty \text{ if } p, q \in S \text{ mutex}(p, q) \\ \sum_{p \in S} h(p) \text{ otherwise}$$

其中 $h(p)$ 可通过一个迭代的过程来完成:(1)初始时,若命题 p 在初始状态中,则 $h(p)$ 为 0,否则为 ∞ ;(2)对于每个动作 a ,若命题 $p \in Add(a)$,则修改 $h(p)$;如果动作 a 的前提条件为 $C = r_1, r_2, \dots, r_n$,则 $h(p)$ 更新为

$$h(p) \leftarrow \min\{h(p), 1 + \sum_{i=1, n} h(r_i)\}$$

3.4 算法基本思想

在灵活规划问题中,对于满意度要求不同,所得到的折衷规划也不同。本文在进行规划搜索时,用户可以先给出所要求达到的目标的满意度值及规定的最大规划长度,然后进行搜索,从而尽可能找到符合该满意度的长度的最短的规划。

对启发式代价做如下规定:动作的代价与它的前提条件有关,一个动作的代价等于支持它的前提条件的命题代价之和;命题的代价用修改后的和互斥代价法计算。

将提出的基于启发式搜索的灵活规划算法称为 FP-H (Flexible Planning based on Heuristic) 算法。本算法采用有序搜索的策略,进行规划的正向搜索,在搜索过程中不断计算各状态的满意度,使用改进的和互斥代价法计算各状态的启发函数值。边搜索边检查目标状态是否出现,直到找到规划解或搜索失败为止。

算法的基本思想:将初始命题集合作为初始状态,使用合适的动作并利用启发式信息生成状态空间搜索树,不断检查所要求满意度的目标状态是否出现,同时进行剪枝。若目标状态出现,并且该状态中的所有命题都不互斥,则搜索成功。若未完全出现,则再寻找一个状态作为当前状态,继续进行探索。重复上述过程,直到目标状态出现或不再有可供使用的状态及操作时为止,规划完毕。其流程图如图 1 所示。

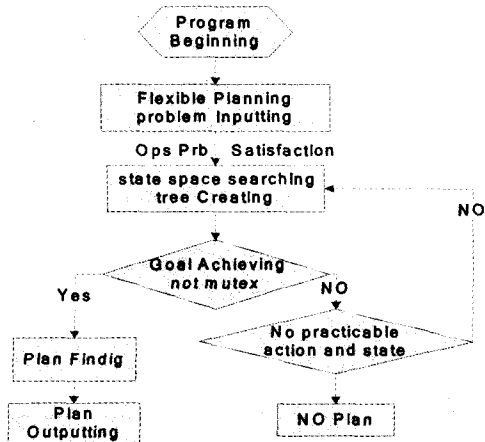


图 1 流程图

3.5 算法描述

(1)将灵活规划问题用状态空间表示法进行表示,将初始条件集合作为初始状态 S_0 , 目标集合作为目标状态 G , 动作集合作为状态搜索的操作集合 Ω , 初始状态 S_0 为当前状态。

(2)检查目标集是否出现在初始状态中。如果出现,则不用搜索,否则生成状态空间搜索树,进行搜索。

(3)把初始状态 S_0 作为当前状态,检查操作集合 Ω 中的所有动作。如果一个动作的所有前提都出现在初始状态 S_0 中,则将状态 S_0 通过该动作向前扩展。过程如下:将 S_0 状态通过所有适用的动作向前扩展,得到很多个子状态(后继状态),用 $S_{10}, S_{11}, \dots, S_{1n}$ 来表示;检查目标集是否出现。若出现则搜索成功,转到 5;否则继续向下执行,进行搜索。

(4)将扩展所得到的各个子状态所使用的动作及其满意度进行标记,同时标记互斥。

1) 计算各状态的满意度。各状态的满意度用 M_i 表示, M_i 为当前状态的上一状态的满意度与扩展到当前状态所使用的操作的满意度的合取。而上一状态的满意度可以通过此方法继续计算得到。当某一状态的满意度 M_i 小于用户所要求的规划满意度 l_i 时,该状态可以舍弃。

2) 计算所有未舍弃的状态的启发函数,其计算过程如下:

$$H(S) \leftarrow \infty \text{ if } p, q \in S_{mutex}(p, q)$$

$$\sum_{p \in C} h(p) \text{ otherwise}$$

其中 $h(p)$ 可通过一个迭代的过程来完成:①初始时,若命题 p 在初始状态中,则 $h(p)$ 为 0, 否则为 ∞ ;②对于每个动作 a , 若命题 $p \in Add(a)$, 则修改 $h(p)$, 如果动作 a 的前提条件为 $C = r_1, r_2, \dots, r_n$, 则 $h(p)$ 更新为 $h(p) \leftarrow \min\{h(p), 1 + \sum_{i=1, n} h(r_i)\}$ 。

3) 根据用户所要求的规划的满意度选择当前状态。若有多于一个 M_i 符合时,选择启发函数值小的状态;若仍然有多个,则任意选择一个,并标记其它状态,留待回溯时使用。

4) 选择一个状态作为当前状态进行搜索,方法同 3。将

生成的各子状态用 $S_{10}, S_{11}, \dots, S_{1n}$ 表示,检查目标集是否完全不互斥的出现。若出现,则搜索成功,转到 5;若不出现,则重复过程 4。

5) 如果将某一状态进行扩展,所得的所有子状态都不能满足要求时则回溯,返回到上一个可供选择的已标记的状态,将其作为当前状态,返回到 4)。

(5)目标集出现,将由该目标状态到初始状态所使用的所有动作反向输出,即为规划解。

(6)若在规定的最大规划长度内仍找不到规划解,则搜索失败。

3.6 基于启发式搜索的灵活规划问题的求解模型

该系统集成了灵活规划和启发式搜索两者的优点,其大致的工作流程如下:

1)将所要解决的灵活规划问题用 PDDL 语言进行描述:包括初始状态描述、目标状态以及动作描述。

2)读入规划问题,将其编译后用规划器调用所要解决的规划问题,找到互斥关系并标记。

3)根据找到的互斥关系,利用改进后的启发函数计算所需的各状态的启发式的值。

4)利用启发性信息进行搜索,生成规划解,然后输出。

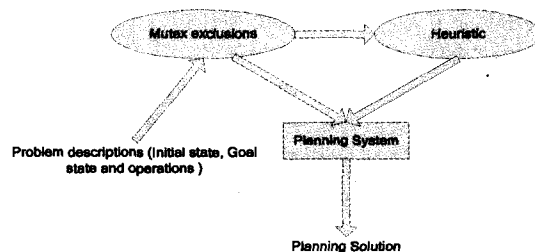


图 2 规划系统求解模型

4 实验结果分析

本算法为了与 FGP 对比也用 Java 实现。运用智能规划中的基准问题和灵活规划问题对本算法与比较有代表性的 FGP(基于规划图和 CSP 的)和 GP 进行实验对比。实验机器为 Pentium(R) 4; CUP2. 80GHz; 内存为 512MB; 操作系统为 Windows XP。通过多次实验,将 Logistic, Rescue 和 Rocket 等领域的对比实验结果列表如表 1(长度对比)。

表 1 FP-H, FGP 与 GP 的对比

| Problem | FP-H | FGP | GP |
|-------------|------|-----|------|
| Logistics-a | 10 | 11 | 1955 |
| Rescue -a | 22 | 25 | -- |
| Rescue-b | 20 | 20 | -- |
| Rocket-a | 6 | 14 | 75 |
| Rocket-b | 10 | 21 | 154 |

实验表明,在很多灵活问题的处理上,本算法要优于 FGP, 有些时候与 FGP 基本相同;而在处理布尔问题时,灵活算法明显优于布尔规划器。

总结 越来越多的研究者开始关注灵活规划的研究,很多学者提出了相应的求解算法。由于启发式搜索在规划领域的特殊地位和突出的效率,经过对启发式搜索和灵活规划的深入研究,本文提出了利用启发式搜索的方法来处理灵活规划问题的思想,给出了利用启发式知识解决灵活规划问题的

求解模型,并提出了基于启发式搜索的灵活规划算法框架。在算法提出的过程中,考虑到本算法的实际情况,对启发式代价做了相应的规定,同时对于启发式函数也做了相应改进。经过测试,该方法在处理很多灵活规划问题上都可以得到较好的效果,具有较高的效率。

本文的后续工作如下:进一步完善基于启发式搜索的灵活规划的相关算法,将该算法运用到实际的智能系统中。

参考文献

- 1 丁德路,姜云飞. 智能规划及其应用的研究. 计算机科学, 2002, 29(2): 100~103
- 2 Blum A L, Furst M L. Fast Planning Through Planning Graph Analysis [C]. In: Proceeding of the International Joint Conference on Artificial Intelligence, 1995. 1636~1642
- 3 Kautz H, Selman B. Planning as Satisfiability [C]. In: Proceedings of the European Conference on Artificial Intelligence (ECA92). Vienna, Austria, August 1992
- 4 Bonet B, Geffer H. Planning as heuristic search: New results [C]. In: Biundo S, Fox M, eds. Recent Advances in AI Planning.

(上接第 204 页)

是 s 的子状态节点。如果 A 和 O 均存在 $(s_1, s_2 \dots s_n)$ 的组合, 则 $s_1, s_2 \dots s_n$ 的与或关系是不一致的。

假设 $(s_i \dots s_j)$ 是 A 和 O 中均存在的组合, 按照前面描述的逻辑表示, 在析取范式集合 G 中必然存在 $\{\neg s \vee s_i, \dots, \neg s \vee s_j, \neg s \vee \neg s_i \vee \dots \vee \neg s_j\}$ 析取式子集, 依据逻辑推理方法,

$$s, \{\neg s \vee s_i, \dots, \neg s \vee s_j, \neg s \vee \neg s_i \vee \dots \vee \neg s_j\} \vdash s, \{\neg s \vee (s_i \wedge \dots \wedge s_j), \neg s \vee \neg s_i \vee \dots \vee \neg s_j\}$$

$$\vdash s, \{\neg s \vee (s_i \wedge \dots \wedge s_j \wedge (\neg s_i \vee \dots \vee \neg s_j))\}$$

$$\vdash s, \{\neg s \vee \text{false}\}$$

$$\vdash s \wedge \neg s$$

$$\vdash \text{false}$$

由此, 可以证明 s_i, \dots, s_j 的与或关系是不一致的。

4.3 语义检验

G 是模型 R 逻辑表示的析取范式集合。假设 X 是用户界面需求相关的领域公理集合, E 是用户需求规定的允许的界面跳转流程示例逻辑表示集合, \bar{E} 是用户需求规定的不允许的界面跳转流程示例逻辑表示集合, 采用人工智能诊断的方法, 如果

- 1) $x \in X, x \wedge G$ 是逻辑为真的
- 2) $\forall e \in E, e \wedge G$ 是逻辑为真的
- 3) $\forall \bar{e} \in \bar{E}, \neg \bar{e} \wedge G$ 是逻辑为真的

则 G 在 X, E, \bar{E} 规定的语义环境下是正确的。

更进一步, 将模型 R 的析取范式集合 G 改写为命题函数字, 即谓词形式的公式集合 H , 求解是否存在解释 I , 使得 H 是可满足的。如果 I 满足 H , 则 I 是界面调转流程的例。反过来说, 如果 I 是用户需求规定允许的界面跳转流程的例, I 不满足 H , 则 H 在 I 定义的需求下是不正确的。由于当解释的域是有限集合时, 一阶逻辑公式可满足问题等价于 SAT 问

5th European Conference on Planning (ECP'99) Durham, UK. Springer-Verlag, 1999

- 5 Miguel I, Shen Q, Jarvis P. Efficient Flexible Planning via Dynamic Flexible Constraint Satisfaction. Engineering Applications of Artificial Intelligence, 2001, 14(3): 301~327
- 6 Miguel I, Jarvis P, Shen Q. Flexible Graphplan. In: Proceedings of the Fourteenth European Conference on Artificial Intelligence, 2000. 506~510
- 7 Li Yang, Gu Wen-xiang, Yin Ming-hao, et al. Planning system based on heuristic. In: Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, 2005
- 8 谷文祥, 李杨, 殷明浩. 带标记的反向搜索的规划识别算法. 计算机科学, 2004, 31(10): 243~244
- 9 彭祥云, 姜云飞. 基于资源约束和局部启发搜索的规划系统. 控制工程, 2006, 13(2): 185~189
- 10 杨宁, 田蔚风, 金志华. 一种求解 DGPS 动态整周模糊度问题的交叉禁忌搜索. 光学精密工程, 2006, 14(5): 891~895
- 11 杨照华, 祁振强, 房建成. 空中目标的免疫识别算法. 光学精密工程, 2006, 14(5): 922~927

题, 因此可以利用 SAT 求解器对模型进行检验。

主要工作总结 通过引入与或逻辑, 丰富了界面关系模型的表达能力, 可以表达传统有限状态机所不能表示的复杂界面关系。我们的主要工作是, 与用户需求的业务流程相结合, 建立体现业务流程的界面关系模型; 其次, 在模型中引入与或逻辑关系, 通过与或逻辑的组合表达实际应用中存在的复杂界面关系; 第三, 讨论了模型的性质, 给出求解效率最优业务流程的算法; 另外, 给出模型的逻辑表示, 利用命题逻辑的推理和演绎, 探讨了模型的语法和语义检查问题。

参考文献

- 1 Schlunbaum E. Model-based User Interface Software Tools: Current state of declarative models[R]. GIT-GVU-96-30, Graphics, Visualization & Usability Center, Georgia Institute of Technology, Atlanta, November 1996
- 2 邵维忠, 杨芙清. 面向对象的系统分析[M]. 清华大学出版社、广西科学技术出版社, 1998
- 3 梁伟晨, 李磊. 基于表单的业务系统界面逻辑模型获取的研究[J]. 计算机工程, 2007, 33(5): 56~68
- 4 张涌, 钱乐秋, 王渊峰. 基于扩展有限状态机测试中测试输入数据自动选取的研究[J]. 计算机学报, 2003, 26(10): 1295~1303
- 5 Memon A M. A Comprehensive Framework for Testing Graphical User Interfaces[D]. [doctoral thesis]. the Graduate Faculty of Arts and Sciences, University of Pittsburgh, 2001
- 6 Nilsson N J. Principles of Artificial Intelligence[M]. North Holland: Tioga Publishing, 1980
- 7 Lee D, Yannakakis M. Principles and Methods of Testing Finite State Machines A survey[J]. In: Proc. of the IEEE, 1996, 84(8): 1090~1123
- 8 李磊. 机器学习在需求工程中的应用[C]. 见: 第八届中国机器学习学术会议报告, 2002