

通用规划综述*

蒋志华¹, 饶东宁², 姜云飞³

(1. 暨南大学 计算机科学系, 广州 510632; 2. 广东工业大学 计算机学院, 广州 510090; 3. 中山大学 信息科学与技术学院 软件研究所, 广州 510275)

摘要: 对智能规划中的通用规划思想及方法进行了全面综述。通用规划是对特定的规划领域制定通用的解形式,从而可以不通过搜索而直接得到具体规划问题的解。当搜索空间随着问题规模指数级增长时,人们希望通过归纳已有规划解的规律,以较小的代价来获取大规模问题的解。介绍了通用规划的基本思想和产生方法,讨论了其与相关工作的联系以及局限性,并给出了对其未来研究方向的展望。

关键词: 智能规划; 通用规划; 参数化方法; 循环展开

中图分类号: TP181 **文献标志码:** A **文章编号:** 1001-3695(2010)12-4414-05

doi:10.3969/j.issn.1001-3695.2010.12.004

Review on generalized plans

JIANG Zhi-hua¹, RAO Dong-ning², JIANG Yun-fei³

(1. Dept. of Computer Science & Technology, Jinan University, Guangzhou 510632, China; 2. Faculty of Computer, Guangdong University of Technology, Guangzhou 510090, China; 3. Software Research Institute, School of Information Science & Technology, Sun Yat-sen University, Guangzhou 510275, China)

Abstract: This paper provided a comprehensive review on ideas and methods of generalized plans in automated planning. By generalizing plans for a field, it could access directly plan solutions for a specific problem in that field without the favor of searching. This issue received widespread attention in recent years. When the search space exponentially grows with the problem size, people naturally hope to solve large-size problems at the low cost, by inducing laws from existing plan solutions. This paper introduced basic ideas, technologies, related works and future research directions of generalized plans.

Key words: automated planning; generalized plans; parameterized methods; loop unrolling

0 引言

智能规划^[1]是人工智能(artificial intelligence)的重要分支,其研究目标是制定算法或者系统来自动获取从问题的初始状态到达目标状态的动作序列。近年来,通用规划^[2~9]问题成为规划领域的一个广为关注的热点。它是指对特定的规划领域制定通用的解形式,从而可以不通过搜索而直接得到具体规划问题的解。研究通用规划的主要原因有两个:a)避开代价高的搜索过程;b)扩展规划解的形式。但是就目前已有的方法而言,这种一劳永逸的想法适用范围还比较小,只能用于某些简单的或者特殊的规划领域。

简单来说,通用规划是像算法一样的规划解,由动作加上顺序、分支和循环等控制结构构成^[6,8]。其中,分支结构是对执行环境的抽象,根据感知动作的结果或者观测变量的值来选择动作;循环结构是对领域对象数量的抽象,在具体对象数量未知的情况下描述实现目标的规律性行为。如果把动作看成语句,那么经典规划解是只包含顺序结构的程序,允许分支结构则变成了不确定环境下的条件规划,再加上循环结构就扩展成为针对领域的通用规划解。所以,通用规划解是现有规划解

形式的一种自然扩展。另一方面,搜索空间的复杂性也促使了对通用规划解的关注。对于规模较小的问题实例,规划系统能够快速找到规划解;但是当规模较大时,规划系统的求解速度很快就慢下来了。人们可以不断地通过改善搜索技术来加快搜索速度,然而不失时机地避开代价高昂的搜索过程也是一种可取代的方法。

1 通用规划的概述

1.1 研究现状

通用规划的早期研究^[10~12]主要是把已有规划解的子序列参数化为宏动作(macro actions)来求解子问题。当宏动作的数量非常多的时候,可以通过定义经常使用的或者求解难题的宏动作来进行优先选取^[11]。但是这样生成的宏动作只能求解非常有限的子问题,且数量非常庞大,因此可应用性差。早期研究中的参数化方法对后期的研究工作^[5,6,8]有着非常大的影响。

随着规划领域研究的发展,自 2003 年开始,通用规划的研

收稿日期: 2010-06-30; **修回日期:** 2010-07-26 **基金项目:** 国家自然科学基金资助项目(60773201,61003179,60903178);广东工业大学博士启动基金资助项目(093032);中央高校基本科研业务费专项资金资助项目(11610305)

作者简介: 蒋志华(1978-),女,广西桂林人,讲师,博士,主要研究方向为智能规划;饶东宁(1977-),男,广东兴宁人,讲师,博士,主要研究方向为智能规划、图论、机器学习(raodn@gdut.edu.cn);姜云飞(1945-),男,吉林长春人,教授,博导,主要研究方向为定理机器证明、智能诊断、智能规划。

人^[2-8]分别提出了生成—测试法、程序合成法和抽象状态计数法,形成了目前从规划解中提取通用规划的三种主要方法。简单来说,生成—测试法通过逻辑程序来描述规划问题并生成循环规划解;程序合成法从规划解中直接识别循环结构;抽象状态计数法在抽象状态空间搜索满足条件的目标状态。此外,2009 年的智能规划和调度国际会议(International Conference on Automated Planning and Scheduling, ICAPS)专门举办了通用规划的 workshop^[13],研究议题包括通用规划的评估标准^[4,14]、在随机环境^[15]或者多 Agent 环境^[16,17]下的通用规划技术以及通用规划在网络服务合成(Web service composition, WSC)^[18]、工作流(workflows)^[19]、多 robot 路径规划^[20]等领域的应用。总的来说,通用规划的提取技术依赖于特定的领域知识和基本的推理或机器学习方法,若干规划领域知识的自动提取技术可参考相关的研究工作^[21-23]。

1.2 描述形式

尽管通用规划从智能规划领域研究的早期就开始了探索,但是目前还没有统一的描述形式。各个研究团队都是根据自己的研究方法和需求制定出相应的通用规划的表示。即使是用来描述规划领域和问题的标准语言 PDDL(planning domain description language),至今也没有扩展能够表示循环结构或者多分支结构等程序设计语言的特性。因此,下面通过例子来说明几种常见的通用规划表示方式。

1) 砍树问题

砍树问题描述的是不断做砍树动作直到把树砍倒为止,要求记录总的砍树次数,其包含动作 chop 和 look。其中 chop 描述砍树的操作;look 是感知动作,观察树是否已经被砍倒。Levesque 等人^[3,7]给出了该问题的通用规划解。其中,用 loop-endl 表示循环结构,用 case-encd 表示多分支结构。

```
loop
case look of
-down: exit
-up: chop;
next
endc
endl;
store
```

2) Gripper 问题

Gripper 问题描述的是一个机械手要把一个房间里的所有球移动到另一个房间,它包含动作 move、pick 和 drop,分别描述 gripper 的移动、拾球和放球动作。Winner 等人^[6,8]给出了该问题的通用规划解,如下所示:

```
while inCurState (at(v?1;ball v?2;room)) and
inGoal-State (at(v?1;ball v?3;room)) do
if inCurState (at-robby(?5;room)) then
move(?5 ?2)
end if
if inCurState (at-robby(?3;room)) then
move(?3 ?2)
end if
pick(?1 ?2)
move(?2 ?3)
drop(?1 ?3)
end while
```

其中,while-do 表示循环结构,谓词 inCurState 表示已经在当前状态下成立,谓词 inGoalState 表示需要在目标状态下成立,以?开头的数字表示变量。

3) 运输问题

运输问题描述的是按要求把货物从出发地用车辆搬运到各自的目的地,它包含 go、load 和 unload 等基本领域动作,分别描述车辆在不同城市间行驶、装货和卸货动作。Srivastava^[2,4,5]给出了该问题的通用规划解,如下所示:

```
1 setDest(dock)
2 go()
3 choose(item)
4 findDest()
5 load()
6 setDest(itemDest)
7 go()
8 unload()
9 setDest(dock)
10 go()
11 choose(item)
12 findDest()
13 load()
14 setDest(itemDest)
15 go()
16 Repeat:
17 unload()
18 setDest(dock)
19 go()
20 choose(item)
21 if (#item = 1)
21a exit loop
22 findDest()
23 load()
24 setDest(itemDest)
25 go()
26 findDest()
27 ...
```

其中:repeat 表示循环结构,setDest、choose、findDest 都是辅助动作,JHJitem 表示货物的数量。

以上三种表示形式既有相似点也有不同点:a)三种表示形式均采用了常见的程序设计语言中的关键字(即 loop / while-do/repeat)来表示循环结构,可读性强;b)如果把动作看成是语句,则砍树问题和 gripper 问题的规划解更像完整的程序,而运输问题的规划解是没有显式结束条件的语句序列;c)砍树问题的通用规划解包含感知动作,适合于不确定规划领域;gripper 问题的规划解包含 inCurState/inGoalState 等谓词,继承了动作策略^[24]的风格;运输问题的通用规划解是从少数几个用例中提取的,其外观特征非常依赖于所采用的具体用例。

2 产生通用规划的方法

2.1 生成—测试法

生成—测试法的具体过程包括生成和测试两个阶段,两个阶段交替进行。在生成阶段,先给一个小常数 N_1 ,该常数用来限定问题实例的对象数目。在规划参数 N_1 下搜索循环规划解,具体做法是先找一个条件规划,然后变成循环规划。使用 Prolog 语言来描述规划问题和提取通用规划解的过程,关键问题是如何知道一个已有的规划解匹配某个循环结构的展开。定义特殊谓词 unwind(P, Q) 在逻辑程序中进行监测,该谓词成立当且仅当循环 P 展开成 Q 。在测试阶段,需要给出一个大常数 N_2 ,用生成的通用规划解来测试对象数目不超过 N_2 的问题实例。如果测试失败,则再进入生成阶段,直到测试通过为止。

生成—测试法使用逻辑程序来描述规划问题和提取过程,符合规划语言说明式的风格。不过由于编程的难度,它适合求解小但是难的问题,而不适合求解大但是简单的问题。例如在打蛋问题(打鸡蛋到碗里,坏的鸡蛋倒掉,好的鸡蛋留在碗里)^[3]中,如果只需要一个好蛋,该方法很快能找到通用规划解,但当需要多个好蛋时,由于缺乏判断好蛋足够多的感知动作,找解时间会很长。

2.2 程序合成法

程序合成法找到的通用规划解称为 dsPlanner 程序,通过不断地将更多的规划例合并到 dsPlanner 中来增强通用性。合

并过程分为三个步骤:a)规划例参数化。将对象映射为变量,相同的对象用同一变量;b)转换为 if 语句,找相关的初始条件与相关的目标作为 if 语句的条件,规划例的动作作为 if 语句的结论;c)判断新的 if 语句是否与已有的匹配(子规划或者重叠),如果匹配则组合,否则追加到末尾。其中,子规划要满足以下若干条件:非重叠、动作名相同、因果链的方式相同、条件和效果相同、合一等。

接下来从生成的 if 语句中识别循环。循环体由共同的子规划组成,其中唯一不同的变量变成循环变量。一个循环展开就是匹配的子规划集合。dsPlanner 能够识别的循环有并行和串行两种。前者迭代间没有因果联系,后者迭代间有因果联系,迭代按顺序进行。图 1(a)给出了并行循环展开的示例,其中 op1、op2 和 op3 组成多步的并行循环。并行循环的识别方法是:找出所有的并行匹配步,从中抽取最大公共部分形成循环。并行匹配步只有一个变量不同,用循环变量替换掉此唯一不同的变量,匹配步作为循环体,没有被匹配步内部满足的动作前提和依赖匹配步的目标作为循环条件。图 1(b)给出了识别之后的循环结构(虚线框环绕,lv 是循环变量,循环条件列于虚线框上方)。串行循环的识别是在以上方法的基础上再根据因果链关系来识别的。

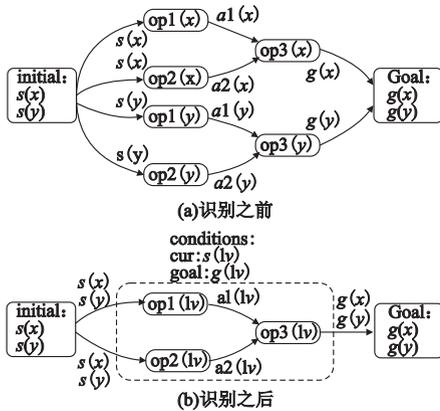


图1 识别并行循环

该方法借鉴了程序自动生成技术来提取通用规划解。其优点非常明显,即用例少、学习时间短,且由通用解求得实例问题解的速度也很快;但缺点是所能识别的循环种类非常有限,适用于求解大但是简单的领域。

2.3 抽象状态计数法

抽象状态计数的主要思想是:使用抽象来描述多个状态空间,动作在抽象状态上的结果包括动作对其所有成员状态的结果,在抽象状态空间搜索满足条件的目标状态,持续搜索直到所有路径搜索完,或者覆盖所有初始状态。抽象状态是具体状态集合,其间的差别在于状态中所包含的对象数不同。对象根据角色来划分,角色是由抽象谓词或其否定组成的文字合取式。抽象谓词是与领域相关的特殊一元谓词,如在 Delivery 领域,不同的箱子扮演着同一角色 Crate,不同的城市扮演同一角色 location。把相同角色的对象分成等价类,以获取状态的抽象表示。实际对象由抽象对象来替代,从而获得包含循环的通用规划解。

算法以问题实例的规划解和状态演化序列为输入,输出通用规划解。状态演化序列是应用实例规划解于初始状态而得到的。主要步骤如下:首先将初始状态转换成抽象状态;然后将规划解应用于初始抽象状态得到抽象状态演化序列;接着从

中形成循环结构;如果最终的抽象状态包含目标条件,那么提取通用规划解保证正确性的适用条件。由于对象被抽象成角色,不管有多少对象,只要角色相同都属于同一等价类。使用计数器来跟踪角色所对应的对象数目,在从通用规划解得到问题实例的解时可以自动进行增减。抽象状态计数法最大的优点是能够提供通用规划解的正确性保证,即确保在适用范围内所得到的问题实例解是正确的。因此,实际上是提供了针对一个问题实例集合的通用规划解。

2.4 方法小结

还有其他的方法来产生通用规划。Guestrin 等人^[25]基于 RMDPs (relational Markov decision processes)对环境进行建模,通过近似值函数的计算来制定在相似游戏场景中的通用行动策略。Langley 等人^[26]从规划解中学习带有循环的逻辑程序并实现其解释器。Petrick 等人^[17]把 Golog 语言描述的复杂动作看成子程序进行直接规划,复杂动作可以包含循环效果和分支效果。Baier 等人^[27]使用过程式的控制知识和大量用户偏好来指导搜索过程,这些控制知识和偏好是针对领域的,具有通用意义。Bonet 等人^[15]从一类随机规划问题中学习将观察映射为动作的通用函数。

下面给出上述各种产生通用规划解的方法中所使用的主要技术统计表,如表 1 所示。其中,相同或者相关的研究工作列在同一行。大部分参数方法是使用变量来替代规划解中的常量,而文献[2,4,5]是使用角色来替代规划解中的对象常量。在其他技术列中,文献[12]使用 EBL 技术,文献[15,25]使用 MDP 技术。另外,值得一提的是,表 1 所列举的主要技术也频繁地出现在其他的领域知识学习方法中,如文献[21]使用循环识别技术来构造递归的派生谓词规则,文献[22]使用参数化方法和抽象状态来构造不确定的动作模型。

表 1 产生通用规划解的方法中所使用的技术

文献编号	技术				
	参数化	识别循环	抽象状态	逻辑程序	其他(MDP/EBL)
[3,7]		√		√	
[6,8]	√	√			
[2,4,5]	√	√	√		
[10,11]	√				
[26]				√	
[12,15,25]					√

3 通用规划的评估标准

当产生或者使用一个通用规划解的时候,还需要一些评估标准来衡量其质量和代价。Srivastava 等人^[4]给出了五个评估标准:

a)规划实例化的复杂性。规划实例化指的是给定问题实例,从通用规划解中构造求解该问题实例的具体规划解的过程;如果包含感知动作,此过程还需与规划执行过程交替进行。规划实例化的复杂性表示从通用规划解中获取实例规划解的时间开销。如果该开销是多项式级别的(如积木块世界中,通用策略的求解时间是积木块数量的 2 次方),采用经典规划技术求解是指数级别的,那么通用规划的优越性能就体现出来了。

b)实例规划的质量。实例规划指的是从通用规划中得到的具体规划解,其质量根据某种度量(动作数或者时间)来衡量。理想状态下,实例规划应该是最优的。但是,如果不依赖

于最优规划器来求解,仅仅判断一个解是否最优是一件相当困难的事情,因此大部分已有的方法只找可行的通用规划解。此外,要求一个通用规划解是最优的,即对该领域的所有问题实例都能产生最优解,也是不现实的。

c) 领域覆盖范围。它指的是能够求解的不同问题实例的集合。当构造一个能求解所有问题的通用规划解很难时,可以转而寻找求解某些问题的通用解。这是通用规划解最重要的评估标准,研究它的初衷也在于此。具体规划解的所有评估标准几乎都可以与通用规划解相同,但唯独不包含这个标准,因为其只能求解一个问题实例。假设问题实例是分布均匀的,通用规划解的覆盖范围越广,表明其通用性越强。

d) 计算通用规划的复杂性。除了产生通用规划方法本身的复杂性之外,规划描述形式也是影响其计算复杂性的主要因素。增加循环结构明显增强了通用规划解的表达力。包含分支/循环结构的程序风格的通用规划解易于测试可用性条件和实现其解释器,但是决策树形式描述的通用规划解的规模随着未知谓词元组的数目指数级地增长。

e) 检查可应用性的复杂性。用通用解产生具体解的时候可以有两种选择:测试其可用性和直接尝试实例化。能够预先定义的可用性测试是比较好的选择,因为当问题不可解时,就不需要花费规划实例化过程的开销。但是这样的可用性测试一般都很难构造,而且最好是在线性时间内完成。最后的求解时间包括可用性测试的时间和规划实例化的时间。直接尝试实例化省去了测试时间,但是当问题不可解时,其计算代价是高昂的。

通过以上五个评估标准可以得到产生通用规划的三种主要方法的性能比较,结果如表 2 所示。a) 关于规划实例化代价和计算代价。由于三种方法主要是与基于搜索的经典规划方法相比,再加上各自适用的问题领域不同,相关文献都没有提供它们之间在两种代价上的互相比。然而,与经典规划方法相比,在所选用的实验领域中,三种方法的代价都是较低的。b) 关于规划质量,均不保证最优性。c) 关于领域覆盖范围,只有程序合成法能保证求解适用领域的所有问题实例,但是这取决于所选用领域的特殊性。d) 关于可用性测试,生成—测试法提供了在规划参数附近值的部分测试,抽象状态计数法虽然没有,但是提供了能保证正确性的通用规划解的适用条件。

表 2 产生通用规划方法的性能比较

评估标准	方法		
	生成—测试法	程序合成法	抽象状态计数法
规划实例化	低	低	低
规划质量	不保证最优	不保证最优	不保证最优
领域覆盖	部分实例	整个领域	部分实例
计算代价	低	低	低
可用性测试	部分测试	没有	没有,但在适用条件下保证正确

4 相关工作

由于通用规划的研究较早,且方法和目标总是在发展中,在智能规划领域与通用规划相关的概念和研究工作较多。本文选取了最相关的五个来进行讨论,包括:

a) 宏动作。最早提出的通用规划其实是宏动作^[10,11]。Fikes 等人^[10]把规划解的子序列参数化为宏动作, Minton^[11]通过定义一些特殊的宏动作来进行优先选取。宏动作与通用规划的主要区别在于,前者用来求解子问题,后者用来求解整个

问题。因此,一个规划领域往往具有数量众多的宏动作,却可能只包含一个或者少数几个通用规划解。

b) 控制知识。控制知识用来引导搜索过程,使得搜索过程能够更快或者更好地朝着目标方向进行^[27]。控制知识可以在求解之前预先设定,或者在求解过程中自发地获取。因此,控制知识与通用规划的最大区别在于,前者依附于搜索过程,后者不需要通过搜索来求解。控制知识的主要缺点在于效率低,信息难以存储和管理以及选择;通过人工预先设定会提高效率,但是耗时、难度大。

c) 动作策略。动作策略由 if-then 形式的决策规则构成^[24,28]。每次产生一个新状态,看是否有匹配的决策规则可用,如果有,则生成下一个状态;如果没有,则宣告求解失败。使用动作策略也不需要搜索过程,但是需要实现类似解释逻辑程序的解释器。与通用规划的主要区别在于,决策规则按照优先级进行排序,优先选取排在前面的规则进行匹配,而通用规划的语句之间是顺序执行的;此外,动作策略不包含循环结构,其具体循环次数由解释器根据问题实例来决定。其主要缺点是求解能力弱以及需要示例多等。

d) 条件规划。在很长一段时间里,条件规划都是通用规划的代名词。条件规划用来表示不确定环境下的规划解^[22,29],由感知动作的结果或者观测变量的值来作为分支条件。与经典规划解相比,条件规划能够求解更多的问题实例,但是由于缺乏明显的循环结构,其表达能力远不如通用规划解。不过两者的关系非常紧密,从已有规划例中产生通用规划解,往往是通过将其转换为条件规划然后识别循环结构来进行。

e) 子程序。如果把动作看成是子程序,通用规划解就是由序列、分支、循环等控制结构组成的完整程序。子程序和动作的主要区别在于:子程序实现某种功能,其调用条件由主程序来决定,而动作描述本身就包含了显式地决定可应用性的前提条件,不满足前提则不会执行动作;子程序改变运行环境中可见变量的值,动作改变的是世界的状态。

5 结束语

本文对智能规划领域中的通用规划思想和方法进行了全面综述,包括研究现状、描述形式、主要产生方法、评估标准以及相关工作等。通过归纳已有规划解的规律,以较小的代价来获取大规模问题的解,这种理想的状态对于崇尚知识获取和应用的学者们来说无疑存在着很大的吸引力。但是,由于实际的规划领域非常复杂,规划解的表达能力有限,产生通用规划的主要方法都只适用于各自的特殊领域,而缺乏广泛的适用性;此外,通用规划的质量和效率也是一个严峻的问题。因此,通用规划的研究还有很长的路要走。未来的研究方向如下:

a) 为了更好地通信和建模,应该建立统一的描述形式。Golog 语言可以描述带有循环/分支结构的复杂动作,但是它不是规划领域描述的标准语言。因此,目前的标准规划描述语言 PDDL^[30]应考虑增加描述复杂动作的语言特性,方便规划系统在求解时能够得到更通用的规划解。

b) 通用规划解应该扩大领域覆盖范围,即力争能够求解更多的问题实例。这种正确性保证可以通过理论证明或者实验验证来获取。理论证明应给出通用规划解的适用条件,实验验证应给出可应用性测试。

c) 通用规划解应尽可能提供最优性保证。可以采用最优规划解来作为学习示例, 或者进行优化组合处理。

d) 应该把方法推广到更一般的规划领域上, 而不是仅仅停留在某些特型领域。此外, 提取时态规划、度量规划或者不确定规划环境下的通用规划解也是很有研究意义和实用价值的。

e) 目前的通用规划提取均只基于由动作实例所构成的规划解, 而派生谓词规则^[21]刻画了规划领域中独立于动作模型的因果联系, 因此学习派生规划领域中的通用规划可以蕴涵更丰富和深层的领域知识, 而这个研究方向也是笔者正在进行的研究工作之一。

参考文献:

- [1] GHALLAB M, NAU D, TRAVERSO P. Automated planning theory and practice [M]. San Francisco: Morgan Kaufmann Publishers, 2003.
- [2] SRIVASTAVA S, IMMERMANN N, ZILBERSTEIN S. Computing applicability conditions for plans with loops [C]//Proc of the 20th International Conference on Automated Planning and Scheduling. 2010: 161-168.
- [3] HU Yu-xiao, LEVESQUE H J. Planning with loops: some new results [C]//Proc of ICAPS Workshop on Generalized Planning: Macros, Loops, Domain Control. 2009.
- [4] SRIVASTAVA S, IMMERMANN N, ZILBERSTEIN S. Challenges in finding generalized plans [C]//Proc of ICAPS Workshop on Generalized Planning: Macros, Loops, Domain Control. 2009.
- [5] SRIVASTAVA S, IMMERMANN N, ZILBERSTEIN S. Learning generalized plans using abstract counting [C]//Proc of the 23rd National Conference on Artificial Intelligence. 2008: 991-997.
- [6] WINNER E, VELOSO M. LoopDISTILL: learning domain-specific planners from example plans [C]//Proc of ICAPS Workshop on Planning and Learning. 2007.
- [7] LEVESQUE H J. Planning with loops [C]//Proc of the 19th International Joint Conference on Artificial Intelligence. San Francisco: Morgan Kaufmann Publishers, 2005: 509-515.
- [8] WINNER E, VELOSO M. DISTILL: towards learning domain-specific planners by example [C]//Proc of the 20th International Conference on Machine Learning. 2003: 800-807.
- [9] PENNA G, MAGAZZENI D, MERCORIO F, *et al.* A tool for universal planning on PDDL + problems [C]//Proc of the 19th International Conference on Automated Planning and Scheduling. 2009.
- [10] FIKES R, HART P, NILSSON N. Learning and executing generalized robot plans [J]. *Artificial Intelligence*, 1972, 3(4): 251-288.
- [11] MINTON S. Selectively generalizing plans for problem solving [C]//Proc of the 9th International Joint Conference on Artificial Intelligence. San Francisco: Morgan Kaufmann Publishers, 1985: 596-599.
- [12] GERVASIO M T. Learning general completable reactive plans [C]//Proc of the 8th National Conference on Artificial Intelligence. 1990: 1016-1021.
- [13] ICAPS 2009 workshop on generalized planning: macros, loops, domain control [EB/OL]. (2009-09-20). <http://www.cs.umass.edu/~siddhart/genplan09/>.
- [14] FILIPPOVSKA Y, KADA M, FRITSCH D. Quality evaluation of ground plan generalization [C]//Proc of 12th AGILE International Conference on Geographic Information Science. 2009.
- [15] BONET B, PALACIOS H, GEFFNER H. Automatic derivation of memoryless policies and finite-state controllers using classical planners [C]//Proc of the 19th International Conference on Automated Planning and Scheduling. 2009: 34-41.
- [16] GIACOMO G D, PATRIZI F, SARDINA S. Solving high-level planning programs [C]//Proc of ICAPS Workshop on Generalized Planning: Macros, Loops, Domain Control. 2009.
- [17] PETRICK R P A. P²: a baseline approach to planning with control structures and programs [C]//Proc of ICAPS Workshop on Generalized Planning: Macros, Loops, Domain Control. 2009: 59-64.
- [18] SOHRABI S, PROKOSHYNA N, MCILRAITH S A. Web service composition via the customization of golog programs with user preferences [C]//Proc of ICAPS Workshop on Conceptual Modeling: Foundations and Applications. 2009: 319-334.
- [19] FORTH J, WALDINGER R. Deductive formation of recursive workflows [C]//Proc of ICAPS Workshop on Generalized Planning: Macros, Loops, Domain Control. 2009.
- [20] SURYNEK P. On pebble motion on graphs and abstract multi-robot path planning [C]//Proc of ICAPS Workshop on Generalized Planning: Macros, Loops, Domain Control. 2009.
- [21] 饶东宁, 蒋志华, 姜云飞, 等. 从规划解中学习一阶派生谓词规则 [J]. *计算机学报*, 2010, 33(2): 251-266.
- [22] 饶东宁, 蒋志华, 姜云飞, 等. 从 WSPEL 程序中学习 Web 服务的不确定动作模型 [J]. *计算机研究与发展*, 2010, 47(3): 445-454.
- [23] 饶东宁, 蒋志华, 姜云飞. 过度规划、部分可满足规划及简单偏好综述 [J]. *计算机应用研究*, 2010, 27(11): 4001-4005.
- [24] KHARDON R. Learning to take actions [J]. *Machine Learning*, 1999, 35(1): 57-90.
- [25] GUESTRIN C, KOLLER D, GEARHART C, *et al.* Generalizing plans to new environments in relational MDPs [C]//Proc of the 18th International Conference on Artificial Intelligence. 2003: 1003-1010.
- [26] LANGLEY P, CHOI D. Learning recursive control programs from problem solving [J]. *Journal of Machine Learning Research*, 2006, 7(11): 493-518.
- [27] BAIER J A, FRITZ C, BIENVENU M, *et al.* Beyond classical planning: procedural control knowledge and preferences in state-of-the-art planners [C]//Proc of the 23rd National Conference on Artificial Intelligence. [S. l.]: AAAI Press, 2008: 1509-1512.
- [28] MARTIN M, GEFFNER H. Learning generalized policies from planning examples using concept languages [J]. *Applied Intelligence*, 2004, 20(1): 9-19.
- [29] RINTANEN J. Conditional planning in the discrete belief space [C]//Proc of the 19th International Joint Conference on Artificial Intelligence. San Francisco: Morgan Kaufmann Publishers, 2005: 1260-1265.
- [30] 饶东宁, 蒋志华, 姜云飞. 规划领域定义语言的演进综述 [J]. *计算机工程与应用*, 2010, 46(22): 23-25, 46.
- [31] ROBERT P G. Partial observability, quantification, and iteration for planning [C]//Proc of ICAPS Workshop on Generalized Planning: Macros, Loops, Domain Control. 2009.
- [32] SRIVASTAVA S, IMMERMANN N, ZILBERSTEIN S. Finding plans with branches, loops and preconditions [C]//Proc of ICAPS Workshop on Verification and Validation of Planning and Scheduling Systems. 2009.
- [33] CLABEN J, HU Yu-xiao, LAKEMEYER G. A situation-calculus semantics for an expressive fragment of PDDL [C]//Proc of the 22nd Conference on Artificial Intelligence. [S. l.]: AAAI Press, 2007: 956-961.