基于学习的规划技术研究

陈蔼祥[1,2] 姜云飞[2] 胡桂武[1] 柴啸龙[1,2] 边 芮[1,2]
(广东商学院数学与计算科学学院 广州 510320)[1] (中山大学软件研究所 广州 510275)[2]

摘 要 经过近十多年的努力,现代智能规划器无论是效率还是处理能力均得到了极大提高。鉴于现有规划理论的局限性,进一步提高现有规划技术效率已愈显困难。现有的大多数规划器均不具备学习能力,无法从先前求解经验中学习有用知识。综述了基于学习的规划技术的发展现状,然后重点介绍了规划大赛中最佳学习器所使用的学习技术,最后指出当前基于学习的规划技术研究领域中存在的主要问题。

关键词 智能规划,基于学习的规划技术,规划器

Research of Learning-based Planning Techniques

CHEN Ai-xiang[1,2] JIANG Yun-fei[2] HU Gui-wu[1] CHAI Xiao-long[1,2] BIAN Rui[1,2]
(School of Mathematics and Computer Science, Guangdong University of Business Studies, Guangzhou 510320, China)[1]
(Software Research Institute, Sun Yat-Sen University, Guangzhou 510275, China)[2]

Abstract After nearly 10 years of effort, the mordern smart planner, whether its efficiency or processing capacity, all have been greatly enhanced. Subject to the limitations of current planning theory, to further enhance the efficiency of existing planning techniques under current framework has become more difficult. Existing planners have not to learn ability, most of them, can not learn from previous experience, useful knowledge to solve. In this paper, we first review the development of planning techniques to learn, and then focused on learning techniques used on the best learning-based planner among international planning competition, concluded the main problems and challenges in current learning technology research of planning.

Keywords Intelligent planning, Learning-based planning techniques, Smart planner

1 引言

规划问题是人工智能研究领域中的一个经典问题,其核心是在给定的已知条件(初始状态)下,如何通过产生式规则(领域动作)获得所需要的结论(目标状态)。对智能规划的研究自人工智能学科诞生之际就已存在,经过近半个多世纪的发展,至今仍然是人工智能研究中的一个极其活跃的研究方向。历届国际人工智能联合会(ijcai)均将智能规划的研究作为会议的一大主题。人工智能领域中的国际性权威杂志"Artificial Intelligence","Journal of Artificial Intelligence Research"均以不同程度的篇幅介绍规划研究相关的论文。"Artificial Intelligence"还多次为智能规划研究发行专刊,最新一期的专刊于2009年4月发行(官方网站上有相应的电子版)[1]。此外,为提供对规划器性能评价的客观标准,国际上专门成立了国际智能规划大赛IPC[2]。迄今为止,该大赛已经举行了6届,形成了统一的规划描述语言PDDL及其各种扩展版本,并有相应的基准测试例子。这极大地推动了智能规划研究的发展,涌现出了一大批性能卓越的规划器[3]。

经过近半个世纪,尤其是最近十多年来的发展,智能规划器无论在求解性能还是在处理能力方面,均得到了极大的提高。然而,由于智能规划问题是NP-complete[4]问题,这一理论局限性使得在传统技术路线下要想进一步提高现有规划器的求解效率愈显困难。如果将智能规划器的求解过程看作是人类思维分析能力的模拟,那么人类智能另一方面的能力——学习能力,则是现代智能规划器所不具备的。智能规划器每次求解时,均像是第一次碰到该问题一样,而不具备从以前的问题求解经验中学习的能力,这极大地影响了规划器求解效率。因此,无论是出于对人类智能模拟的研究的目的(人工智能终极目标),还是出于打破现有规划技术存在的效率瓶颈以进一步提高现有规划器的求解效率,均有必要对规划求解过程中的学习问题进行研究。为此,IPC6上开辟了专门针对基于学习的规划系统的比赛轨道。

规划器的学习能力对提高规划系统的效率是非常重要的,这一点从实验结果中可以明显看到。当我们对TLPlan[5]和SHOP[6]系统提供手写的领域控制知识时,其求解效率均得到了显著提升。

智能规划领域中存在大量学习搜索控制知识的研究,然而这方面很少出现能够让人信服的、在大范围规划领域有效

到稿日期:2010-02-05 返修日期:2010-05-07 本文受国家自然科学基金(60773201)资助。
陈蔼祥(1978—),男,博士,讲师,主要研究方向为智能规划与诊断,E-mail:cax413@163.com;姜云飞(1945—),男,教授,博士生导师,主要研究方向为自动推理、智能规划和基于模型的诊断等;胡桂武(1970—),男,博士,教授,主要研究方向为智能计算;柴啸龙(1980—),男,博士,讲师,主要研究方向为智能规划;边 芮(1982—),女,博士,讲师,主要研究方向为智能规划。

的成果。导致学习和非学习规划器这种性能上的差异主要是因为定义既要对大多数规划领域足够丰富，又必须足够紧凑，能为有效和可靠的学习提供支持的表示控制知识的假设空间是非常困难的。事实上，大多数先前研究的一个共同缺点正是其相应的假设空间仅对特定的小范围内的规划领域有效，而对许多其他领域则不够丰富，导致基于学习的规划系统的研究远落后于领域无关的非学习规划系统的研究。

基于学习的规划技术研究是智能规划研究领域中新出现的一个趋势，这方面的研究刚刚起步。本文对基于学习的规划技术研究领域的发展进行了系统分析，重点介绍了规划大赛中最佳学习器所使用的学习技术，最后对规划大赛学习轨道的比赛结果进行了分析和总结，指出了目前基于学习的规划技术研究领域中存在的困难和不足。

2 基于学习的规划技术发展现状

对于规划中的学习问题的研究，最早可以追溯到STRIPS规划器[7]，该系统具备学习宏的能力。许多基于学习的规划系统均是基于解释学习(explanation-based learning，EBL)模式的，比如Minton等的工作[8]。EBL属于演绎学习方法，即学习到的控制知识是可证明的。尽管这方面存在大量的工作，但有效成果甚少，甚至很多情况下学习损害了系统整体性能。这主要是因为EBL学习了太多过于具体的控制规则，结果导致规划器在控制规则的评估方面时间开销过多，超过了因控制规则使用对搜索空间的压缩的效果。

部分地受EBL方法局限性影响，许多规划系统则采用归纳学习方法。归纳学习方法主要是采用统计学习原理寻找能区分好的搜索方向和坏的搜索方向的共同模式。与EBL不同，归纳学习出来的控制知识不能保证其正确性。然而，这种知识往往具有普遍性，因而在实用中更有效。这类有代表性的系统有基于部分序规划的学习[9]、基于可满足规划的学习[10]以及基于Prodigy means-ends框架[11]的学习。这些系统一般比对应的EBL系统具有较好的扩展性，其一般用于较小范围内规划领域和/或较少量的测试问题。尚未有足够的试验结果表明这类系统具有足够鲁棒性，能与大范围规划领域中对非学习的规划系统相比较。

最近，Khardon等人提出从规划领域中学习反应式策略(reactive policies)的方法[12-14]，他们通过使用统计学习技术学习将给定规划领域中的状态-目标影射到合适动作的策略或函数。给定好的领域反应式策略，通过迭代使用这些策略就可快速解决规划问题，而不需要进行搜索。这些方法比较简单，也取得了一定程度上的成功，但仍然无法在大范围领域内超越那些非学习的规划系统。

再励学习的思想也被应用于AI规划领域学习控制策略中。关系再励学习(RPL)[15]使用带关系函数逼近的Q-learning方法，在Blocksworld领域中有较好的实验结果。从传统的再励学习的角度来看，他们使用的Blocksworld问题比较复杂，具有大规模的状态动作空间。而从AI planning角度来看，这些问题则是相对比较简单的。这一方法并没有能够呈现出对现有规划器能处理的大规模问题的良好扩展性。另一相关方法采用更强大的再励学习形式——近似策略迭代[16]，在许多规划领域显示出良好的结果。同样，该方法在很多领域表现并不佳，总体上尚未能在所有的比赛基准测试例子中与最先进的规划器相媲美。

Macro-FF[17]和Marvin[18]则是在前向状态空间搜索中学习宏动作序列，然后将学习到的宏动作应用于新问题的求解。不同的是，Marvin是在线学习的，即它在具体问题求解时获取宏动作，然后在后续阶段的搜索中使用宏动作。然而，从最近规划的大赛结果来看，没有一个学习的规划系统能比最好的非学习规划器占上风。

Beniamino Galvani等人注意到规划器的性能受搜索空间的结构影响较大，规划领域决定了其搜索空间的基本结构。这导致某些规划器在某些领域表现出良好性能，而在另一些领域则表现不如其他规划器。基于此，他们提出了多种规划器自动配置的基于公文包的求解策略：首先，为公文包中的规划器计算宏动作集合，然后从中选择有竞争力的几个规划器以及相应的有用宏动作，最后按照轮盘赌的方式为这些选中的规划器配置运行时间片。他们的规划器PbP[19]共使用了Fast Downward，Metric-FF，LPG-td，MacroFF，Marvin，SGPLAN5，YAHSP等7个规划器。

最后，值得一提的是Gerevini和Schubert[20]以及Fox和Long[21]等，对领域分析技术进行研究，试图通过对领域定义进行分析，以揭示领域中的结构。但这些方法尚未呈现出在一定范围的规划领域中对规划器性能有所提升。

3 ObtuseWedge规划学习技术[14]

ObtuseWedge是IPC-6规划大赛学习轨道上获得最佳学习奖的规划器，因此本文重点对该技术进行介绍。ObtuseWedge规划器通过学习领域相关控制知识，并将学习到的控制知识应用于前向启发式规划搜索框架中，来提高规划器求解效率。ObtuseWedge的主要贡献是提出了一种新的用以表示控制知识的特征空间以及在该特征空间下的控制知识：反应式策略(决策列表规则)和线性启发式函数-放宽式规划特征的线性组合，并给出了这些控制知识的学习和使用方法。

3.1 关系数据库、类别表达式、特征空间

在介绍上述3种控制知识及其学习方法之前，需要对表示这些控制知识的特征空间进行介绍。ObtuseWedge中使用的特征空间是相对于某一个搜索节点而言的(即某一搜索节点的特征空间)，特征空间的表示要用到关系数据库D以及概念表达式两个概念。

关系数据库D：给定搜索节点(s,A,g)，关系数据库$D(s,A,g)$包含以下事实：

1)s中所有事实。

2)放宽规划中每一动作ai的名字。每一动作名字实质是领域定义中动作类型y。

3)对动作ai添加表中的状态事实，在其前面加字母'a'，然后添加到关系数据库中。例如事实holding(B)在动作pickup(B)的添加表中，则将aholding(B)添加到数据库中。

4)类似地，动作ai删除表中的事实，在其谓词前面添加'd'，然后添加到数据库中。

5)目标状态中的事实，在其前添加'g'，然后添加到数据库中。

6)在当前状态成立且在目标状态中亦要求成立的事实，在其前面添加'c'，然后添加到数据库中。这样，字母'c'开头

的谓词提供了一种表示当前状态与目标状态关系的机制。

类别表达式则是用来表示在搜索节点(s,A,g)和相应关系数据库$D(s,A,g)$中满足某一特定性质的工具。下面给出类别表达式的语法形式,并解释其语义。

类别表达式的语法形式:类别表达式可从谓词集合P构造得到,$n(P)$表示谓词P参数个数。定义于P上的所有可能的类别表达式按下列语法给出:

$C:=$a-thing$|P_1|C\cap C|\neg C|(P\,C_1\cdots C_{i-1}?\ C_{i+1}\cdots C_{n(P)})$

式中,C和C_j是类别表达式,P_1是任意一元谓词,P是参数为2个或以上的任意谓词。由上述语法表达式可以看到,原子类别表达式为符号a-thing,用以表示所有对象集合。而一元谓词P_1表示该谓词为真的所有对象集合。然后,通过交、求补、关系复合(最后一条规则)等操作构造复合类别表达式。在定义类别表达式语义前,先给出类别表达式C的深度$depth(c)$的定义:a-thing和单目谓词深度为0,$depth(C_1\cap C_2)=1+\max(depth(C_1),depth(C_2))$,$depth(\neg C)=1+depth(C)$,$depth(P\,C_1\cdots C_{i-1}?\ C_{i+1}\cdots C_{n(P)})=1+\max(depth(C_1),\cdots,depth(C_{n(P)}))$。请注意,类别表达式数目可以是无穷的,但我们在学习过程中通常对类别表达式的深度作适当限定,以限制类别表达式数目。

类别表达式的语义:下面给出类别表达式的语义描述,该语义是基于有限数据库D和有限常量符号或对象集。可将D简单地看作是一有限一阶模型,或者Herbrand解释。给定一类别表达式C和D,用$C[D]$表示C关于D的对象集合。也用$P[D]$表示在D中的对应谓词P的对象元组(tuples of objects)(即使P为true的元组)集合。

如果$C=$a-thing,则$C[D]$表示D中所有对象集合。例如,如果D是从Blocksworld中的一个状态构造的,则a-thing表示所有积木块集合。如果C是单目谓词符号,则$C[D]$为D中所有使P为true的对象集合。例如clear$[D]$和ontable$[D]$分别表示在D中处于clear和在桌面上的积木块集合。如果$C=C_1\cap C_2$,则$C[D]=C_1[D]\cap C_2[D]$,例如(clear$\cap$ontable)$[D]$表示在桌面上且clear的积木块集合。如果$C=\neg C$,则$C[D]=$a-thing$-C'[D]$。最后,如果$C=P\,C_1\cdots C_{i-1}?\ C_{i+1}\cdots C_{n(P)}$,则$C[D]$表示存在$c_j\in C_j[D]$的常量$c$,使得元组$(C_1\cdots C_{i-1}\ c\ C_{i+1}\cdots C_{n(P)})$在$P[D]$中的所有$c$的集合。例如(on clear ?)[D]表示直接处于clear状态的积木块下面的积木块集合。

特征空间:给定搜索节点(s,A,g)及数据库$D(s,A,g)$、类别表达式C,该类别表达式关于数据库D的对象集合用$C[D(s,A,G)]$表示。这样,每一C可被看作定义了搜索节点的一个特征,该特征取值为$C[D(s,A,G)]$。因此,给定一类别表达式C,其关于某一搜索节点的关系数据库对应的特征值为f_C,有$f_C=C[D(s,A,G)]$。这样,所有可能的类别表达式就构成了搜索节点对应的特征空间。

3.2 学习启发式函数

给定放宽规划特征空间,下面描述如何用该空间表示和学习启发式函数。该启发式函数在前向状态空间搜索中用作控制知识。

3.2.1 启发式函数表示

启发式函数$H(s,A,g)$为状态s、动作集合A和目标g的函数,作为从状态s用A中动作到达目标g估计代价。这里,我们考虑学习的启发式函数表示为函数f_i的线性组合,即$H(s,A,g)=\sum_i w_i f_i(s,A,g)$。特别地,对每一规划领域,我们希望学习出不同的函数f_i及其对应的权重的集合,以便能在得到的线性启发式函数的引导下,具有良好的规划求解性能。这里,每一函数对应前述定义的放宽规划数据库下的类别表达式C_i,记为f_G。给定搜索节点(s,A,g),f_{C_i}取C_i关于$D(s,A,g)$的基数,即$f_G(s,A,g)=|C_i[D(s,A,g)]|$。

3.2.2 启发式函数的学习

学习算法的输入为规划问题集合及每一规划问题对应的解。这些解不一定是最优的,但应是好的、合理的。学习目标是学习启发式函数,使得该启发式函数所代表的距离能逼近训练问题的解产生的状态离目标的距离。为此,首先产生训练集J,J中的训练例子包含解路径中的所有状态。特别地,对每一训练问题(s_0,A,g),其对应的解路径为$(s_0,s_1,\cdots,s_n)$。在J中添加n个形如$\{((s_i,A,g),n-i)i=0,\cdots,n-1\}$的例子,每一例子均是规划问题和该问题在解路径中离目标的距离的二元组。给定如此派生的训练集J,我们试图通过学习能充分逼近J中记录的距离与FF放宽式规划长度$RPL(s,A,g)$启发式信息之间的差异的实值函数$\Delta(s,A,g)$。最终启发式函数取$H(s,A,g)=RPL(s,A,g)+\Delta(s,A,g)$。

将$\Delta(s,A,g)$表示成f_G的有限线性组合,其中C_i是选取自放宽式规划特征空间,即$\Delta(s,A,g)=\sum_i w_i f_i(s,A,g)$。注意$H(s,A,g)$的总体表示是特征的线性组合,此时$RPL(s,A,g)$的特征权重置为1。另一种可行的设计方案是让$RPL(s,A,g)$的权重也通过学习得到。但初步研究表明,将$RPL(s,A,g)$的权重取值固定为1,只学习$\Delta(s,A,g)$,在某些领域已取得好的效果。

学习上述$\Delta(s,A,g)$涉及到从无穷的类别表达式空间中选择一类别表达式集合并分别为之设置权重的问题。为此,可给出表达式深度界限,然后学习表达式深度在给定界限内的所有特征的权重(例如用最小均方误差)。然而,存在的一个问题是特征数随表达式深度界限指数增长,这导致该方法仅在深度界限较小时才有效。该方法的另一问题是无法在给定深度界限内识别那些重要的特征。此外,我们通常希望使用尽可能小的特征,因为评价学习到的启发式函数的复杂度随所选特征线性增长。

图1是从派生的训练集J学习$\Delta(s,A,g)$的算法。主程序Learn-Delta首先创建一修改过的、与J对应的训练集J'。J'中每一训练例子中与目标的距离取J中相应训练例子与目标的距离和FF放宽式规划长度启发式距离之间的值。Learn-Delta每次迭代维护一类表达式Φ集合,表示当前考虑的特征集合。最初Φ被置为深度为0和1的表达式集合。每次迭代有两个主要步骤:首先用例程Learn-Approximation选择Φ中类表达式子集,并计算各自特征权重;然后创建新的深度加深的表达式集合所代表的特征集Φ,将当前被选择的特征作为种子调用例程Expand-Features。这将导致包含原种子特征在内的更大的候选特征集,该特征集合反过来又被Learn-Approximation使用,以进一步提高逼近效果。如此交替进行,直到精度不能再进一步提高。这里逼近精度采用R-square值表示。

```
Learn-DeIta(J,D)
    //J is pairs of problem states and plan length from them
    //D is domain definition to enumerate class expressions
J'←{((s,A,g),d−RPL(s,A,g))|((s,A,g),d)∈J}
    d is the plan length, the remaining states in the solution trajectories
Φ←{C|C is a class expression of depth 0 or 1}
repeat until no R-square value improvement observed
    (Φ',W)←Learn-Approximation(J',Φ)
      //Φ'is newly selected features, W is the set of weights for Φ'
    Φ←Expand-Features(D,Φ')
Return Φ',W

Learn-Approximation(J,Φ)
Φ'←{}//return features
repeat until no improvement in R-square value
    C←arg max_{C∈Φ} R-square(J,Φ'∪{C})
      //R-square is computed after linear approximation with the features
    Φ'←Φ'∪{C}
W←lm(J,Φ')
    //lm, least square approximation, returns weights
Return Φ',W

Expand-Features(D,Φ')
Φ←Φ'//return features
for-each C∈Φ'
    Φ←Φ∪Relational-Extension(D,C)∪Specialize(D,C)∪(¬C)
Return Φ
```

图 1　学习启发式信息伪码

Learn-Approximation 使用简单贪心搜索策略。从空特征集合出发，每次迭代把 Φ 中最能提高当前特征集合的 R-square 值的特征包含进来，直到 R-square 值不能再提高为止。给定当前特征集合，新考虑的特征的质量通过调用统计工具 R 的 lm 函数进行评价，该函数能输出 R-quare 值以及包含该新特征时的线性逼近的权重。若发现 R-square 不再有所改善，Learn-Approximation 将返回最终选择的特征集及其各自权重，从而得到 $\Delta(s,A,g)$ 的线性逼近。

例程 Expand-Fearures 创建包含种子集合和由种子产生的新表达式所构成的新的表达式集合。给定种子 C，有许多方法可产生新的扩展特征集。这里考虑 3 种实用中表现良好的方法。第一个函数 Relational-Extension 以一种子表达式 C 为输入，返回形如 $(P\ c_0\cdots c_{j-1}\ C\ c_{j+1}\cdots c_{i-1}?\ c_{i+1}\cdots c_{n(P)})$，其中 P 为参数大于 1 的谓词符号，ci 为所有 a-thing，$i,j\leqslant n(P)$。其结果是谓词的一个参数被 C 中对象所约束，其他参数不变的所有可能结果。例如，Blocksworld 中，对 holding 这一类表达式的关系扩展，可表示成(gon holding ?)，该扩展表达式表示的是所有在目标状态中处于当前状态中被抓取的积木块下面的积木块集合。

第二个扩展类表达式的例程是 Specialize。该例程简单地生成所有类表达式，它是通过用深度为 0 或者 C 的一个子表达式 c'，代替该子表达式与其他深度为 0 或一个类表达式的交创建得到的。因此，由 Specialize 产生的所有表达式 be subsumed by C。也就是说，对于任意这样产生的表达式 C'，任意 D 有 $C'[D]\subseteq C[D]$。例如，给定 Blocksworld 表达式(on ? a-thing)，表示所有在积木块上的积木块，Specialize 产生的表达式可能是(on ? (a-thing∩gclear))，表示在当前状态下处于积木块上，而在目标状态下要求为 clear 的积木块。最后，将种子类别的补添加到扩展特征集中。例如，在 Logistic-sworld 中，输入类别表达式为(cin ? a-thing)，表示已经处于目标位置的 packages，则其输出的补为¬(cin ? a-thing)，表示尚未处于目标位置的那些 packages。

3.3　学习反应式策略

我们首先将要学习的策略表示成分类决策列表的形式。下面首先考虑分类决策列表的表示形式，然后考虑如何学习该分类决策列表。

3.3.1　分类决策列表的表示

所谓分类决策列表是列表形式的动作-选择规则(action-selection rules)，每一规则形式如下：

$$a(x_1,\cdots x_k):L_1,L_2,\cdots L_m$$

式中，a 为带 k 个参数的动作，L_i 为文字，x_i 为动作的参数变量。每一文字有类似 $x\in C$ 的形式，其中 C 是分类语法类别表达式，x 为动作的参数变量。

给定搜索节点 (s,A,g) 以及动作参数对象列表 $o=(o_1,\cdots,o_k)$，称文字 $x_i\in C$ 被满足，如果 $o_i\in C[D(s,A,g)]$，也就是说，对象 o_i 满足类别表达式 C 所表示的约束。称规则 $R=a(x_1,\cdots x_k):L_1,L_2,\cdots L_m$ 在 (s,A,g) 中建议使用动作 $a(o_1,\cdots o_k)$，如果规则中每一文字在给定的 (s,A,g)，O 中为真，并且动作的前提条件在 s 中被满足。注意，如果某一包含动作 a 的规则中没有文字存在，则 a 的所有合法动作都将被该规则建议。规则可看作是对动作可能作用的对象集合施加的互斥约束。一条规则可能不建议任何动作，也可能建议同一类型的多个动作。给定这种规则的决策列表，一动作被该列表所建议，如果该动作被该列表中的某些规则所建议，并且先前没有规则建议任何动作。同样，决策列表可能不建议任何动作，也可能建议同一类型多个动作。

决策列表 L 通过下列方式定义确定性策略 $\pi(L)$：如果在节点 (s,A,g) 中，L 没有建议任何动作，则 $\pi(L)(s,A,g)$ 为那些前提条件在 s 中满足的字典序最小动作，否则，$\pi(L)(s,A,g)$ 为 L 建议的最小动作。值得说明的是，由于 $\pi(L)$ 仅考虑合法动作，即前提条件满足的动作，因此规则中无须对前提条件进行显式编码，使规则比较简单，且方便学习。换句话说，我们考虑规则时相当于隐式包含了该类型动作的前提条件。

作为分类决策列表的例子，考虑 Blocksworld 领域中一个简单的问题，要求将所有积木块放置在桌上(on the table)。下列策略适用于该领域的任何问题：

putdown($x1$)：$x1\in$ holding，

pickup($x1$)：$x1\in$(on ? (on ? a-thing))

第一条规则建议 agent 放下被抓着的积木块。否则，如果 agent 没有抓任何积木块，则第二条规则建议 agent 抓起某一积木块(该积木块又在另一其他对象，或者是桌子，或者是另一积木块上面)上面的积木块(注意，如果第二条规则被改成 pickup($x1$)：$x1\in$(on ? a-thing)的形式，则会导致循环，因为有可能会出现将刚刚放在桌上的积木块抓起的情况)。

3.3.2　学习决策列表

图 2 的算法是决策列表学习算法。Learn-Decision-List 中的训练集 J 为包含从解路径中观察到的搜索节点和相应动作的二元组的多集(multi-set)。学习算法的目标是寻找决策列表，即训练集中每一搜索节点所建议的相应动作。

```
Learn-Decision-List(J,d,b)
    //J:set of training instances where each instance is a search node
        labeled by an action
    //d:the depth limit of class expressions
    //b:beam width,used in search for the rules
L←()
while (J≠{})
    R←Find-Best-Rule(J,d,b)
    J←J−{j∈J|R suggests an action for j}
    L←L:R; //append rule to end of current list
Return L

Find-Best-Rult(J,d,b)
Hvalue-best-rule←−∞;R←()
for-each action type a
    Ra←Beam-Search(a,J,d,b)
    if H(J,Ra)>Hvalue-best-rule
        //H(J,Ra) is learning heuristic function in Equation 2
        R←Ra
        Hvalue-bset-rule←H(J,Ra)
Return R

Beam-Search(a,J,d,b)
Lset←{(xk∈C)|k≤n(a),depth(c)≤d}
    // the set of all possible literals involving class expressions of
      depth d or less
bearm←{a(x1,…,ak)} //initial beam contains rule with empty rule
                     body
Hvalue-best←−∞;Hvalue-best-new←0
while (Hvalue-best<Hvalue-best-new)
    Hvalue-best←Hvalue-best-new
    candidates←{R,l|l∈Lset,R∈beam}
        //the set of all possible rules resulting from adding one liter-
          al to a rule in the beam
    beam←set of b best rules in candidates according to heuristic H
          from Equation 2
    Hvalue-best-new←H value of best rule in beam
Return best rule in beam
```

图 2　学习策略的伪码

算法采用类似 Rivest-style[22] 的决策列表学习算法。每次学习一条规则，从高优先级到低优先级，直到结果规则集覆盖了所有训练数据。这里一规则覆盖一训练例子如果该规则对例子中的状态建议了一动作。理想规则是指只建议训练集中的动作。

主程序 Learn-Decision-List 最初将规则列表初始化为空表，然后调用 Find-Best-Rule 子例程，选择那些能覆盖最多训练例子，并且这些被覆盖的训练例子中正确率最高的规则——即具有高覆盖面和高精确性的规则。将选择的结果添加到当前决策列表的表尾，同时将那些被覆盖的训练例子从训练集中移除。接着，算法继续从约简后的训练集中搜索高覆盖率和高精度的规则。重复上述过程，直到不再有训练例子未被覆盖。注意，通过将被前述规则覆盖的训练例子从训练集中移除，可使算法能够集中在那些尚未被当前规则集建议任何动作的训练例子。

整个算法的核心是 Find-Best-Rule 例程，该例程每次迭代要在指数规模的规则空间搜索好的规则。我们知道，每条规则有下列形式：$a(x_1,\cdots,x_k):L_1,L_2,\cdots,L_m$。由于规则空间是指数大小的，因此使用贪心 beam-search 算法。特别地，Find-Best-Rule 主循环遍历动作类型 a，对每一动作类型使用 beam search 为该动作 a 构造文字集合。这些最佳规则(用一启发式函数进行评价)将被返回。接下来介绍在文字集合上执行 beam search 以及启发式评价函数。

Beam-Search 的输入为动作类型 a、当前训练集、宽度为 b 的 beam、考虑的类表达式的深度界限 d。beam 的宽度和表达式的深度界限是用户给定的参数，这两个参数确定了搜索的总量。一般取 $d=2,b=10$。搜索初始化后，当前 beam 只含空规则，即只有头 $a(x_1,\cdots,x_k)$，没有文字的规则。每次迭代构造一候选规则集。每一种添加新文字(深度小于或等于 d)到 beam 中某一规则的方法，均可得到一条规则。如果存在 n 个可能文字，将会产生 nb 条规则。接下来，将使用规则启发式评价函数，以选择这些规则中最好的 b 条规则，保留在 beam 中，供下一次迭代，其他候选规则将被丢弃。上述过程重复进行，直到搜索再也不能发现未被改进的规则(根据启发式评价函数)。

最后，规则启发式评价函数根据训练集 J 对规则 R 进行评价。对规则的启发式评价方法有很多，这里使用的是实验中显式效果好的方法。直观地，该启发式函数倾向那些能对训练集中许多搜索节点建议正确动作，而同时这些建议的动作中不在训练集中的数目尽可能少的规则。用 $R(s,A,g)$ 表示由规则 R 在 (s,A,g) 所建议的动作集合。据此，根据下列公式，对规则 R 在训练实例 $((s,A,g),a)$ 下的优劣评价方法如下：如果训练集动作 a 不被规则 R 所建议，则 R 得分为 0，否则该规则得分随 $R(s,A,g)$ 的大小下降。总的启发式值是所有训练实例的启发式值的总和。这种方式，对于那些只能覆盖少量例子的规则以及能覆盖大量例子但所建议的动作大多是在训练集外的动作的规则，均被赋以较小的启发式值。

$$\text{benefit}(((s,A,g),a),R)=\begin{cases}0, & a\notin R(s,A,g)\\ \dfrac{1}{|R(s,A,g)|}, & a\in R(s,A,g)\end{cases}$$

$$H(J,R)=\sum_{j\in J}\text{benefit}(j,R)$$

结束语　经过近半个世纪，尤其是最近十多年来的发展，智能规划器无论在求解性能还是在处理能力方面，均得到了极大的提高。然而，如果将智能规划器的求解过程看作是人类思维分析能力的模拟，那么人类智能另一方面的能力——学习能力，则是现代智能规划器所不具备的。智能规划器每次求解时，均像是第一次碰到该问题一样，而不具备从以前的问题求解经验中学习的能力。这极大地影响了规划器求解效率。因此，无论是出于对人类智能模拟的研究的目的(人工智能终极目标)，还是出于打破现有规划技术存在的效率瓶颈以进一步提高现有规划器的求解效率，均有必要对规划求解过程中的学习问题进行研究。

目前基于学习的规划技术基本上是在完全实例化搜索空间中提取各种经验知识(包括各种策略、控制规则、启发式函数、宏动作)。而一般来说，规划问题的实例化搜索空间异常庞大，导致经验知识的提取困难且费时。并且，通过这种方式所学习出来的经验知识，往往过于具体(与具体的实例化对象高度相关)，这导致经验知识的评价和使用高度依赖上下文搜索，极大地降低了经验知识在指导规划求解的有效性。这是目前基于学习的规划系统在效率上甚至不如非学习的规划系

(下转第 61 页)

ACM MobiCom. Rome, Italy, July 2001

[5] Savarese C, Rabaey J, Langendoen K. Robust positioning algorithms for distributed ad-hoc wireless sensor networks[C]// Proceedings of the USENIX technical annual conference. Monterey, CA, USA, June 2002

[6] He T, Huang C D, Blum B M, et al. Range-free localization schemes for large scale sensor networks[C]// Proceedings of ACM MobiCom. San Diego, CA, USA, Sep. 2003

[7] Hu L, Evans D. Localization for mobile sensor networks[C]// Proceedings of ACM MobiCom. Philadelphia, PA, USA, Sep 26-Oct 1, 2004

[8] Rudafshani M, Datta S. Localization in wireless sensor networks [C]// Proceedings of ACM/IEEE IPSN. Cambridge, MA, USA, Apr. 2007

[9] Lederer S, Wang Y, Gao J. Connectivity-based localization of large scale sensor networks with complex shape[C]// Proceedings of IEEE INFOCOM. Phoenix, AZ, USA, Apr. 2008

[10] Kung H T, Lin C, Lin T, et al. Localization with snap-inducing shaped residuals(SISR): coping with errors in measurement[C]// Proceedings of ACM MobiCom. Beijing, China, Sep. 2009

[11] Zhong Z, He T. Achieving range-free localization beyond connectivity[C] // Procee-dings of ACM SenSys. Berkeley CA, USA, Nov. 2009

[12] Niculescu D, Nath B. Ad-hoc positioning system[C]// Proceedings of IEEE Globecom. San Antonio, TX, USA, Nov. 2001

[13] Shang Y, Shi H, Ahmed A. Performance study of localization methods for ad-hoc sensor networks[C]// Proceedings of IEEE MASS. Fort Lauderdale, FL, USA, Oct. 2004

[14] Wang C, Xiao L. Locating sensors in concave areas[C]// Proceedings of IEEE INFOCOM. Barcelona, Catalunya, Spain, Apr. 2006

[15] Li M, Liu Y H. Rendered path: range-free localization in anisotropic sensor networks with holes[C]// Proceedings of ACM MobiCom. Montreal, Quebec, Canada, Sep. 2007

(上接第 19 页)

统的主要原因。

虽然基于学习的规划系统的研究已经取得了一定的成果，但如何在降低学习代价的同时，提高学习的质量，尽可能减少学习对规划系统整体性能的负面影响，使得基于学习的规划系统能通过有效学习在性能表现上能可靠地、令人信服地超越非学习的规划系统，改变目前最佳的非学习的规划系统的表现优于所有基于学习的规划系统的尴尬局面，仍然是尚待解决的难题。

参 考 文 献

[1] Fox M, Thiébaux S. Advances in Automated Plan Generation [J]. Artificial Intelligence, 2009, 173(5/6): 501-788

[2] 国际智能规划大赛网址[EB/OL]. http://ipc. informatik. uni-freiburg. de/, 2008

[3] 智能规划器的部分列表[EB/OL]. http://www. aiai. ed. ac. uk/links/planning. html, http://planning. cis. strath. ac. uk/plansig/index. php? page = planners, http://www. csc. ncsu. edu/faculty/stamant/planning-resources. html

[4] Bylander T. The computational complexity of propositional STRIPS planning[J]. Artificial Intelligence, 1994, 69(1/2): 165-204

[5] Bacchus F, Kabanza F. Using temporal logics to express search control knowledge for planning[J]. Artificial Intelligence Journal, 2000, 16: 123-191

[6] Nau D, Cao U, Lotem A, et al. Shop: Simple hierarchical ordered planner[C]// Proceedings of the International Joint Conference on Artificial Intelligence. 1999: 968-973

[7] Fikes R E, Nilsson N J. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving[J]. Artif. Intell., 1971, 2(3/4): 189-208

[8] Minton S, Carbonell J, Knoblock C A, et al. Explanation-based learning: A problem solving perspective[J]. Artificial Intelligence Journal, 1989, 40: 63-118

[9] Estlin T A, Mooney R. Multi-strategy learning of search control for partial-order planning[C] // Proceedings of 13th National Conference on Artificial Intelligence. 1996

[10] Huang Yi-Cheng, Selman B, Kautz H. Learning declarative control rules for constraint-based planning[C] // Proceedings of Seventeenth International Conference on Machine Learning. 2000: 415-422

[11] Aler R, Borrajo D, Isasi P. Using genetic programming to learn and improve control knowledge[J]. Artificial Intelligence Journal, 2002, 141(1/2): 29-56

[12] Khardon R. Learning action strategies for planning domains[J]. Artificial Intelligence Journal, 1999, 113(1/2): 125-148

[13] Martin M, Geffner H. Learning generalized policies in planning domains using concept languages[C]// Proceedings of Seventh International Conference on Principles of Knowledge Representation and Reasoning. 2000

[14] Yoon Sungwook, Fern A, Givan R. Learning Control Knowledge for Forward Search Planning[C]// JMLR. 2008, 9: 683-718

[15] Dzeroski S, Raedt L D, Driessens K. Relational reinforcement learning[J]. Machine Learning Journal, 2001, 43: 7-52

[16] Fern A, Yoon Sungwook, Givan R. Approximate policy iteration with a policy language bias: Solving relational markov decision processes[J]. Journal of Artificial Intelligence Research, 2006, 25: 85-118

[17] Botea A, Enzenberger M, Muller M, et al. Macro-FF: Improving AI planning with automatically learned macro-operators[J]. Journal of Artificial Intelligence Research, 2005, 24: 581-621

[18] Coles A I, Smith A J. Marvin: A heuristic search planner with online macro-action learning[J]. Journal of Artificial Intelligence Research, 2007, 28: 119-156

[19] Gerevini A, Saetti A, Vallati M. An Automatically Configurable Portfolio-based Planner with Macro-actions: PbP[C]// Proceedings of 19th International Conference on Automated Planning and Scheduling. 2009

[20] Gerevini A, Schubert L K. Discovering state constraints in DISCOPLAN: some new results[C]// Proceedings of National Conference on Artificial Intelligence. AAAI Press/The MIT Press, 2000: 761-767

[21] Fox M, Long D. The automatic inference of state invariants in TIM[J]. Journal of Artificial Intelligence Research, 1998, 9: 367-421

[22] Rivest R L. Learning decision lists[J]. Machine Learning, 1987, 2(3): 229-246