命题编码中公理的组合与设计

江 鸿 刘大有 吕 帅 蔡敦波 史晶晶
(吉林大学计算机科学与技术学院 长春 130012)
(吉林大学符号计算与知识工程教育部重点实验室 长春 130012)

摘 要 近年来,基于可满足性的规划方法研究逐渐成为智能规划研究领域中的热点。提出3种基于Graphplan的编码方式中公理的改进:动作互斥的部分放松、动作互斥的完全放松方法、添加框架公理。基于SATPLAN2006规划系统分别实现上述3种改进的编码方式,并对国际规划竞赛中选用的标准后勤域与积木世界域的问题样例予以测试,分析不同编码方式的编码规模与求解效率,验证了基于Graphplan编码方式的改进在绝大多数情况下是有效的。最后,实现基于状态的编码方式,并对上述两个域进行测试,比较约简动作与约简状态这两种极端方式的求解效率和编码规模。实验结果表明,在后勤域的某些问题上基于状态的编码方式比基于动作的编码方式有效得多。上述的改进策略表明,可根据问题域的特性等来考虑该问题最适宜哪些公理组合的编码方式,而不固定使用某种特定的编码方式。

关键词 智能规划,基于可满足性的规划,Graphplan,公理

Combination and Designation of Axioms in Proposional Encodings

JIANG Hong LIU Da-you LU Shuai CAI Dun-bo SHI Jing-jing
(College of Computer Science and Technology, Jilin University, Changchun 130012, China)
(Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China)

Abstract In recent years, researches on planning as satisfiability have become a popular trend. We proposed partial relaxation and completed relaxation methods about mutex actions, appended the frame axioms finally. Taking SATPLAN2006 as the base, we implemented these improved encoding methods respectively, tested them in the logistics track and the block world track which are used in international planning competition, and analyzed the encoding scale and plan efficiency of different encoding methods, and then validated the improvements based on the graphplan encoding method in the overwhelming majority situation are effective. Finally, we implemented the state-based encoding method in SATPLAN2006 planner, tested on the above tracks and compared two extremely conditions, which eliminate actions and states respectively. The experimental results show that state-based encoding method is more effective in logistics track than action-based encoding. The above improving strategies about the SATPLAN2006 planner are proved effective, and we should decide which combination of axioms in one encoding method is most suitable by consideing the characteristics of different tracks, rather than using certain encoding method absolutely.

Keywords Intelligent planning, Planning as satisfiability, Graphplan, Axiom

智能规划是一个多学科交叉的领域,涉及知识表示、自动推理、非单调逻辑、情景演算、人机交互和知识挖掘等方面[1]。智能规划技术采取以适应性代替显示设计的方法,使得预期计算的结果得以自动地学习和演化。即使在设计者本人也无法预见的情况下,系统仍然可以自主地选择合适的动作来执行,因此智能规划技术被广泛应用于航空航天技术、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[2]。1957年Newell和Simon设计了问题求解程序GPS(General Problem Solving),Green设计了QA3系统[3]。而1971年Fike和Nilsson的STRIPS系统[4]使规划能非常容易地进行描述和操作,在智能规划领域中具有划时代的意义。近年来,基于可满足性(SAT)的规划方法逐渐成为智能规划研究领域中的热点。

SAT问题已被证明,也是第一个被证明为NPC(NP-complete)的问题[5]。基于SAT的规划系统的典型代表是Blackbox,是在1996年由Kautz和Selman等人根据将规划问

到稿日期:2008-11-05 返修日期:2009-02-25 本文受国家自然科学基金重大项目(60496321),国家自然科学基金项目(60573073,60503016,60603030,60773099,60703022,60873149),国家863高技术研究发展计划项目(2006AA10Z245,2006AA10A309),吉林省科技发展计划重点项目(20060213),欧盟项目TH/AsiaLink/010(111084)资助。

江 鸿(1985—),女,硕士研究生,主要研究领域为智能规划与自动推理、数据挖掘,E-mail:jh730@126.com;刘大有(1942—),男,教授,博士生导师,主要研究领域为知识工程与专家系统、数据挖掘等;吕 帅(1981—),男,博士研究生,主要研究领域为智能规划与自动推理;蔡敦波(1981—),男,博士研究生,主要研究领域为智能规划、约束满足问题;史晶晶(1985—),女,硕士研究生,主要研究领域为智能规划与数据挖掘。

题转化为命题可满足问题的求解思想设计并实现的规划系统，它是一种基于转换的公认的高效规划方法[5]。

在基于 SAT 的规划方法中，一个智能规划问题不再是需要被证明的一系列定理，而仅仅是一个公理集，此公理集需要保证它的任意一个有效模型对应着一个有效的规划解[5]。基于 SAT 的规划方法的核心是规划编码构造与 SAT 问题的判定过程。由于 SAT 问题求解技术已相对成熟，本文重点研究基于可满足性的规划方法中的编码方式的各种公理的组合。

1 SATPLAN2006 规划系统的编码方式

在第五届国际规划竞赛(IPC-5)中，由 Joerg Hoffmann，Henry Kautz，Shane Neph 和 Bart Selman 设计的 SATPLAN2006 规划系统[6]在命题最优域夺得了冠军。

1.1 SATPLAN2006 规划系统概述

SATPLAN2006 规划系统逐层扩展规划图，产生一个逻辑命题公式。如果该公式是可满足的，则表示规划解存在。其中运用符号表记录命题变量和规划实例之间的对应；运用简化技术(如单元子句法、纯文字消去法等)来降低 CNF 公式的规模；运用求解器来找到一个可满足的赋值集；运用解码器将赋值转换成一个规划解。如果求解器发现公式是不可满足的，那么编译器就会继续产生新的编码[11]。

SATPLAN2006 以模块化的形式构造，因此可不加修改地调用任意 SAT 求解器，SAT 求解器在单独的过程中运行。详细内容见文献[7]。

SATPLAN2006 能够处理 STRIPS 型的规划域和问题，并在此基础上搜索最短并行长度的规划解。若干不互斥的动作可以在某个时间步并行地执行，保证时间步尽可能小。

基于可满足性的规划系统的框架如图 1 所示。

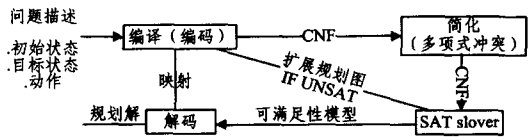

图 1 基于可满足性的规划系统框架

SATPLAN2006 的编码方式是基于 Graphplan 的编码方式和基于动作的编码方式。研究这两种编码方式中的公理及其具体实现，考虑它们内在的联系性和矛盾性，然后在此基础上进行改进，增加其它公理来重新实现 SATPLAN2006，并比较各种公理组合的编码规模和求解效率。

1.2 基于可满足性规划方法中的编码方式

线性编码[8]给出充要条件集，确保所有模型的域公理、初始状态和目标状态对应有效的规划解。其公理集为：1)动作蕴涵它的前提和效果；2)在每一个瞬时的时间步都只能发生一个动作；3)初始状态完全刻画；4)经典框架公理在所有动作上都成立。

基于 Graphplan 的编码[8,9]转换的公理集为：1)初始状态在第 0 层成立，目标状态则在最高层；2)操作蕴涵它的前提；3)每一个第 t 层的命题蕴涵第 $t-1$ 层所有将其作为添加效果的操作的析取；4)相冲突的动作是互斥的。

基于状态的编码[9]由于其强大的一致性公理，许多关于动作与动作的前提或效果的公理可被安全删除。约简命题和公理的过程可被执行到极端：完全消除动作的命题。

2 基于 Graphplan 的编码方式的改进

2.1 结合 Graphplan 和 SATPLAN 的规划方法

SATPLAN2006 结合了 Graphplan 和 SATPLAN 的优点[8]，在此选择基于 Graphplan 的编码方式来进行改进。SATPLAN2006 现在包含了局部搜索 SAT 求解器 Walksat 两种系统 SAT 求解器 satz 和 rel_sat 以及 Graphplan 引擎(搜索规划图来替代 CNF 形式)。satz 和 rel_sat 两种求解器尽管方法不同，但在能力上是相当的：satz 以前向检察(forward-checking)为基础，而 rel_sat 以相关直接的回溯(backtracking)为基础。在下面的测试工作中，选择 rel_sat 求解器作为 SAT 求解器，当然这种选择是任意的。尽管不同的求解器与 CNF 范式的形式有着不同的契合度，但本文着重考虑编码技术中公理的不同组合对编码规模与求解效率的影响。

在本文中，选用如下两个规划问题域：

1)后勤域，高度并行的规划域。建模一类使用运输工具运送货物到指定地点的物流问题。运输工具为卡车和飞机，卡车只能在一个城市内的不同地点移动，飞机只能在城市之间的飞机场运输货物。每个具体的问题涉及一些城市，每个城市包括一些地点：卡车停靠站和飞机场。问题的目标是将处于不同地点的一些包裹运到指定地点。

2)积木世界域，一个完全的串行域。积木世界是机器人世界的微观表示，是人工智能中最主要的问题域。在积木上可执行的动作包括抓住或放下物体，将物体放在其他积木或桌子上。每个具体问题涉及一些积木的堆放状态。问题的目标是通过不同的移动方式，将积木更改为指定的堆放状态。

2.2 互斥公理的改进

在基于 Graphplan 的编码方式中，动作的互斥通过函数 interfere(int op1，int op2)和 competing_needs(int time，int op1，int op2)计算(op1，op2 表示规划图上同一层的两个动作)，并存于 vector 类的 bit_A_exclusives 中。注意到竞争需求和命题间的互斥并不是操作的逻辑特性，更确切地说，它们是依赖于操作和初始条件的共同影响。

2.2.1 互斥公理的放松

Graphplan 运用了一些简单的规则来记录命题间和动作间的互斥关系。这些规则并不能保证找到所有的互斥关系，但其绝大多数[10]还是可被找到。因此可以考虑继续通过削弱互斥的规则，实现 Graphplan 系统运行效率的提高。通过对互斥关系的不同考虑，提出如下 3 种放松方法：

1)动作互斥的部分放松，只考虑冲突(而不考虑竞争需求)，即将函数中判断两个操作互斥的条件(互斥关系以 vector 类形式存储，调用 interfere(int op1，int op2)和 competing_needs(int time，int op1，int op2)两个函数在逐层扩展规划图时分配它们的值)替换为 interfere(int op1，int op2)。

2)动作互斥对完全放松，不考虑动作之间的互斥关系。

3)完全不考虑命题互斥的条件。

2.2.2 互斥公理转换的数据结构

首先明确互斥公理在程序中转化的存储结构。以动作互斥为例，互斥的子句由两个负文字组成，形式如下：

$$\neg load(A,R,L,2) \vee \neg move(R,L,P,2)$$

在此处负文字是将已编码的动作或命题变量乘以 -1 实现的。在实际的编码中，将表示为：

$-12,-23$

在此，假定 *load*$(A,R,L,2)$在 SATPLAN2006 中编码为 12，*move*$(R,L,P,2)$编码为 23，每个变量编码初值为-1。

2.2.3 实验结果

实验硬件条件如下：

CPU：P4—2.8GHz；内存：1G；操作系统：linux ubuntu；编程环境：linux GCC

分别实现在 2.2.1 节中提出的 3 种互斥公理的放松策略，并与源系统在后勤域和积木世界域的不同测试样例上进行对比分析。实验结果如下：

(1)动作互斥的部分放松(PMA：Partial Mutex Actions)

为了描述方便，称在基于 Graphplan 的编码中考虑动作互斥的部分放松的编码方式为基于 PMA 的编码方式。表 1 对比了两种编码方式对于不同测试问题的运行结果。

其中 prob001 — prob013 为后勤域问题，prob014 — prob024 为积木域问题，*step* 表示最优解的步数，*realtime* 为程序运行的总时间(时间单位为秒)，*var* 为编码的变量的总数目，*clauses* 为转化后的 SAT 问题对应的子句集的规模，'—'表示在 900s 内不能得出规划解。

表 1 动作互斥的部分放松的实验结果

Problem	step	Graphplan			PMA		
		realtime	var	clauses	realtime	var	clauses
Prob001	9	0.14	1297	15916	0.36	1297	6070
Prob002	7	1.27	1623	47668	0.39	1623	10120
Prob003	11	1.65	2822	49634	0.38	2822	14252
Prob004	17	57.24	9523	351271	—	—	—
Prob005	10	3.46	6086	259788	0.97	6086	47701
Prob006	—	—	—	—	—	—	—
Prob007	11	8.74	6999	328743	2.57	6999	59420
Prob008	10	5.06	6759	331536	1.4	6759	57711
Prob009	12	68.79	8216	375120	90.75	8216	64743
Prob010	11	9.43	5951	240129	4.89	5951	45757
Prob011	11	243.3	8056	409807	315.33	8056	70017
Prob012	12	8.25	6718	247739	14.09	6718	49121
Prob013	13	78.75	8049	310863	35.43	8049	59795
Prob014	12	1.12	1581	45594	0.56	1581	25896
Prob015	12	7.82	2829	147904	3.32	2829	84085
Prob016	8	0.03	319	2779	0.05	319	1708
Prob017	2	0.01	31	76	0.01	31	62
Prob018	6	0.06	231	1896	0.03	231	1156
Prob019	6	0.13	421	6184	0.09	421	3548
Prob020	6	0.07	371	4881	0.05	371	2822
Prob021	10	0.39	909	17334	0.22	909	9916
Prob022	12	1.13	1445	38780	0.56	1445	21904
Prob023	20	7.31	3347	114314	3.71	3347	66633
Prob024	30	50.12	7475	382016	24.27	7475	228714

实验结果表明，在基于 Graphplan 的编码方式中，部分考虑动作互斥是有效的。对于大部分问题实例，其求解效率优于源 SATPLAN2006 规划系统。

基于 PMA 的编码与基于 Graphplan 的编码有以下几点说明：

首先，基于 PMA 的编码中的变量个数没有变化。因为互斥公理必须作用在每层已经生成并编码的动作或命题变量上，它本身不会增减编码中的变量个数。

其次，基于 PMA 的编码中的子句个数明显减少。一般情况下都比基于 Graphplan 的编码中的子句个数降低了 5 至 7 倍，因而可减少 SAT 求解器判定子句集是否可满足的工作量，缩减系统运行的总时间。

最后，程序运行的求解效率有显著提高，一般在 2 倍左右，甚至有的达到了 4 倍。

另外，对于 prob001，prob004 和 prob011 问题，基于 PMA 的编码方式的求解效率有所降低，prob004 甚至超时，这可能与求解问题本身的性质有关。

因此在基于 Graphplan 的编码方式中考虑动作部分互斥是有意义的。在绝大部分问题实例上的求解效率优于源 SATPLAN2006 规划系统。

(2)动作互斥的完全放松(CMA：Complete Mutex Actions)

为了叙述方便，称在基于 Graphplan 的编码中考虑动作互斥的完全放松的编码方式为基于 CMA 的编码方式。表 2 对比了两种编码方式对于不同测试问题的运行结果。

该编码方式是不完备的，因为有的问题在少于最优解步数时调用 SAT 求解器判定，就已经找到了"最优解"，说明实际上只能串行的动作由于互斥公理的放松而可能并行地执行。

舍弃步数不满足最优解的问题后的实验结果如表 2 所列。

表 2 动作互斥的完全放松的实验结果

Problem	step	Graphplan			CMA		
		realtime	Var	Clauses	realtime	var	Clauses
Prob001	9	0.14	1297	15916	0.12	1297	2599
Prob005	10	3.46	6086	259788	0.74	6086	18432
Prob008	10	5.06	3195	88238	0.98	6759	19180
Prob012	12	8.25	6718	247739	0.92	6718	21428
Prob019	6	0.13	421	6184	0.06	421	1466
Prob020	6	0.07	371	4881	0.05	371	1258
Prob021	10	0.39	909	17334	0.16	906	3340
Prob022	12	1.13	1445	38780	0.28	1445	6009
Prob023	20	7.31	3347	114314	1.86	3347	14130
Prob024	30	50.12	7475	382016	3.62	7475	20597

实验结果表明，当考虑动作互斥的完全放松而能求出最优解时，在求解效率与编码规模上都有很大程度的改进。

在积木世界域问题上，基于 CMA 的编码方式对于大部分问题可以得出正确的规划解。最根本的原因是由于积木世界域问题是完全串行的，在同一个时间步内不存在两个或两个以上的动作同时发生。其中，prob024 的求解效率提高了将近 16 倍。但在后勤域上则只有少数几个问题可以求得正确的规划解。

将求解步数小于最优解步数的问题与图中的问题对比，发现对于前者，基于 CMA 的编码的变量个数小于源基于 Graphplan 的编码的变量个数，而后者的变量个数不变。这说明动作互斥的完全放松导致某些原来能编码的动作变量不参与 SAT 求解过程。这可能是先于最优解步数得出最终"失效"规划解的主要原因。

(3)完全不考虑命题的互斥

改进(1)中测试的例子都无法求解，出现死循环，直至规划图扩展的层数超出程序中设置的最大层数(105)。故说明在基于 Graphplan 的编码中，命题互斥的公理是必需的。

2.3 框架公理的添加

一般来说，每增加一个新的命题都需要增加和域中动作数大致一样多的新的框架公理。每增加一个新的动作，都需要增加与域中变量的数目大致相等的框架公理。也就是说，

在一个域中，如果有 n 个命题和 m 个动作，则总共需要 $n*m$ 个框架公理。框架问题研究的是如何以一种形式化的、逻辑的方法表示动作的影响，而不用写出所有的框架公理则是刻画哪些行动在怎样的条件下，不会引起命题或状态的改变。要全面刻画动态系统的行为特征，就必须考虑每种行为对每个命题的影响情况[9]。

在 SATPLAN2006 规划系统中，并未生成框架公理，因此考虑添加框架公理来改进基于 Graphplan 的编码方式。

2.3.1 框架公理的生成算法

框架公理分为经典框架公理和解释性框架公理，首先需要弄清框架公理在 SATPLAN2006 中的表示形式。

假定存在某命题 P，在规划图的第 i 和 $i+1$ 命题层分别记为 P_i 和 P_{i+1}，在第 i 层上能将其 P_{i+1} 作为添加效果的所有动作记为 $A_1, A_2, \cdots, A_n$，在第 i 动作层上将 P_{i+1} 作为删除效果的所有动作记为 $D_1, D_2, \cdots, D_m$，则

1)经典框架公理

$P_{i+1} \Leftrightarrow P_i \vee A_1 \vee A_2 \vee \cdots \vee A_n$

$\neg P_{i+1} \Leftrightarrow \neg P_i \vee D_1 \vee D_2 \vee \cdots D_m$

将蕴涵公式转换成子句形式，即为：

$\neg P_{i+1} \vee P_1 \vee A_1 \vee A_2 \vee \cdots \vee A_n$

$P_{i+1} \vee \neg P_i \vee D_1 \vee D_2 \vee \cdots \vee D_m$

2)解释性框架公理

$\neg P_i \wedge \neg A_1 \wedge \neg A_2 \wedge \cdots \wedge \neg A_n \Leftrightarrow \neg P_{i+1}$

$P_1 \wedge \neg D_1 \wedge \neg D_2 \wedge \cdots \wedge \neg D_m \Leftrightarrow P_{i+1}$

将蕴涵公式转换成子句形式，即为：

$\neg P_{i+1} \vee P_1 \vee A_1 \vee A_2 \vee \cdots \vee A_n$

$P_{i+1} \vee \neg P_i \vee D_1 \vee D_2 \vee \cdots \vee D_m$

可知框架公理在 SATPLAN2006 中的表示形式是相同的，故添加两种框架公理的效果是一样的。添加框架公理的算法 1 如下所示。

算法 1 frame_axiom($t, i, lnum\_clauses$)

1)if 第 t 层上的第 i 个命题 ft 未编码
  then continue
2)将负的 ft 编码的文字存入已分配空间的子句里
  for $k = 0, \cdots gft\_conn[i].num\_A$ do
    if ft 在第 $t-1$ 层上的第 k 个支持动作存在
    then 将支持动作已编码的文字存入子句里
  end for
  将 $t-1$ 层上第 i 个已编码的命题文字存入子句里
3)lnum_clauses++
4)$flag \leftarrow 0$
  for $k = 0, \cdots gft\_conn[i].num\_D$ do
    if ft 在第 $i-1$ 层上的第 k 个删除动作存在
    then $flag \leftarrow 1$, break
  end for
  if $flag == 0$ then continue
  将 ft 编码的文字存入已分配空间的子句里
  for $j = 0, \cdots gft\_conn[i].num\_D$ do
    if ft 在第 $i-1$ 层上的第 k 个删除动作不存在
    then continue
    if 第 $t-1$ 层上的第 j 个删除动作未编码
    then 将其编码并记录下来
    将 $t-1$ 层上第 i 个已编码的动作文字存入子句里
  end for
  将 $t-1$ 层上第 i 个已编码的负的命题文字存入子句里
5)$lnum\_clauses$++

算法 1 生成每个命题对应的框架公理之前，先确定该命题在该层上是否存在，再对该命题在规划图上层存在的支持动作(支持动作的个数为 $gft\_conn[i].num\_A$)生成正框架公理(对应于算法中的第 2 步)。若规划图的上层存在该命题的删除动作(在规划图的第 $i-1$ 层上若一个动作的删除效果包含第 i 层上的命题，则称此动作为该命题的删除动作，其个数为 $gft\_conn[i].num\_D$)，则对其生成负框架公理(对应于算法中的第 4 步)，每个公理都以子句的形式存储，生成完毕，子句个数($lum\_clauses$)加 1。该算法必须与效应公理在每层上交替构造，因为如果单独生成框架公理，对于规划图上的每一层的命题该公理不起任何作用。

需要注意的是在算法 1 中，当规划图上第 t 层上第 i 个命题存在时，则必须在第 $t-1$ 层上存在至少一个该命题的支持动作(包含 $noop$ 动作)，所以必须生成正框架公理；而第 $t-1$ 层上该命题的删除动作则可能不存在，故负框架公理不一定用到。

2.3.2 实验结果

在 2.2.3 节中，对动作互斥的部分放松的实验结果表明，基于 PMA 的编码方式在后勤域和积木世界域的绝大多问题上是有效的，故可以在此基础上添加框架公理，再实现一种新的编码方式。

为了描述方便，称在基于 PMA 的编码中添加框架公理的编码方式为基于 FA 的编码方式。表 3 对比了两种编码方式对于不同测试问题的运行结果。

表 3 添加框架公理的实验结果

Problem	Step	PMA			FA		
		realtime	var	clauses	realtime	var	clauses
Prob001	9	0.36	1297	6070	0.13	1297	6614
Prob002	7	0.39	1623	10120	0.32	1623	10775
Prob003	11	0.38	2822	14252	0.4	2822	15542
Prob004	17	—	—	—	—	—	—
Prob005	10	0.97	6086	47701	1.11	6086	50124
Prob006	13	—	—	—	4.33	3314	20288
Prob007	11	2.57	6999	59420	6.57	6999	61944
Prob008	10	1.4	6759	57711	1.57	6759	60118
Prob009	12	90.75	8216	64473	52.23	8216	68097
Prob010	11	4.89	5951	45757	1.2	5951	48131
Prob011	11	315.33	8056	70017	28.17	8056	72955
Prob012	12	14.09	6718	49121	25	6718	51892
Prob013	13	35.43	8049	59795	10.52	8049	63165
Prob014	12	0.56	1581	25896	0.57	1581	25896
Prob015	12	3.32	2829	84085	3.29	2829	84085
Prob016	8	0.05	319	1708	0.04	319	1708
Prob017	2	0.01	31	62	0.01	31	62
Prob018	6	0.03	231	1156	0.03	231	1156
Prob019	6	0.09	421	3548	0.14	421	3792
Prob020	6	0.05	371	2822	0.07	371	3038
Prob021	10	0.22	909	9916	0.27	909	10440
Prob022	12	0.56	1445	21904	0.71	1445	22720
Prob023	20	3.71	3347	66633	5.92	3347	68511
Prob024	30	24.27	7475	228714	46.14	7475	232826

实验结果表明，在基于 PMA 的编码方式中添加框架公理是有意义的。对于大部分问题实例，其求解效率优于基于 PMA 的编码的求解效率。

基于 FA 的编码与基于 PMA 的编码有以下几点说明：

首先，基于 FA 的编码中的变量个数没有变化。因为框

架公理必须运用在每层上已经生成并编码的命题变量上，它本身不会增减编码中的变量个数。

其次，子句数有所增加。对于积木世界域的 11 个问题(prob014－prob018)的编码规模不变，而运行总时间也几乎没有变化，这说明框架公理在这 5 个问题上并没有生成；prob019－prob024 的子句个数和运行总时间均有所增加，这说明框架公理不适宜用在积木域问题上。在后勤域的 13 个问题上的子句个数增加不多。

最后，程序运行的求解效率有显著提高。对于后勤域问题，有 4 个问题的求解效率是近乎相等的。prob007 和 prob012 问题的运行时间较基于 PMA 的编码方式的求解效率慢了两倍，但对于其它问题，求解效率则提高了至少 2 倍。特别对于 prob011，基于 FA 的编码方式的求解效率提高了一个数量级以上。

值得注意的是，prob006 在基于 Graphplan 的编码方式和基于 PMA 的编码方式下均显示超时。但对于添加了框架公理的基于 FA 的编码方式下，却能在 4.33s 内得到最优解。这也说明在后勤域的问题上运用框架公理在大多数情况下是高效的。

3 基于状态的编码方式的实现

3.1 基于状态的编码方式

在基于状态的编码方式(State-based encoding)中，只需逐层对每个命题进行编码。在第 0 层和最高层上的处理方式都与基于 Graphplan 的编码方式相同；再根据命题蕴涵上一层的支持动作，而动作又蕴涵产生它的前提，从而将该层命题与上层命题直接联系起来，逐层对每个命题编码；然后对已编码的命题搜索互斥对，产生互斥公理；最后需要统一转换为 CNF 范式。

基于 Graphplan 的编码方式在表达上有较多的缺陷，而基于状态的编码方式则能进一步合并表达上的求精[9]。运用“state-based”这个术语是因为它首先利用每个单独的状态是否有效的公理断言，并利用状态公理转移描述操作。例如，在积木世界中，状态公理断言只能有一个积木在另一个积木上面，每一个积木必须在某个物体上面，一个积木不能既在其上存在某个物体又不存在任何物体。在后勤域中，状态公理断言每个可运输的物体只能在一辆卡车上，而且一辆卡车只能在一个位置。

状态编码在纯表示立场上是值得注意的：没有明确的框架公理、前提公理与效应公理、动作间的互斥公理，所有这些都包含在命题/状态之间简单的一致关系。

基于状态的编码的规划问题，其规划解对应了一个状态序列。因为每一对相邻状态都对应于一个长度为 1 的无序规划的简单问题，“缺失”的动作可以从该序列中得到(一般情况下，找到长度为 1 的无序规划解是 NP-hard 的)。然而在测试的域中，包括后勤域和积木世界问题，存在一个找到这类规划解的线性算法。本文只考虑不同的编码方式的编码规模与求解效率，故不考虑从由规划问题转换成 SAT 问题的解中如何提取动作序列(即在 main()主函数中屏蔽掉函数 create_solution())。

3.2 基于状态的编码方式的实现

本文只考虑两种极端情况：完全“删除”动作或命题变量。前者称为基于状态的编码方式，后者称为基于动作的编码方式。

3.2.1 基于动作的编码与基于状态的编码

分别举例说明 SATPLAN2006 中“删除”命题变量的基于动作的编码方式和“删除”动作的基于状态的编码方式的思想[9]。

例 1　假设第 t 层上有命题 P_1 和 P_2，均为第 $t-1$ 层上的动作 A_1，A_2 的前提，则有 $A_1 \supset P_1$，$A_2 \supset P_1$，$A_1 \supset P_2$，$A_2 \supset P_2$，第 $t-1$ 层上的动作 B_1，B_2 的添加效果均包含 P_1 和 P_2，则 $P_1 \supset (B_1 \vee B_2)$，$P_2 \supset (B_1 \vee B_2)$，那么由蕴涵的传递性可将作为桥梁的命题 P_1 和 P_2 删除，即为 $A_1 \supset (B_1 \vee B_2)$，$A_2 \Leftrightarrow (B_1 \vee B_2)$。此编码方式产生的 CNF 子句在最坏的情况下为每层上已编码的动作变量的总数。

例 2　假设一个命题 S_2，可作为其添加效果的若干个动作记为 A_1，A_2 和 A_3，则有 $S_2 \Leftrightarrow (A_1 \vee A_2 \vee A_3)$。每一个这样的动作都有一对不同的前提，$A_1$ 需要前提 P_1 和 P_2，A_2 需要前提 P_3 和 P_4，A_3 需要前提 P_5 和 P_6，则有 $S_2 \Leftrightarrow (P_1 \wedge P_2) \vee (P_3 \wedge P_4) \vee (P_5 \wedge P_6)$。

要将其转换成 CNF 范式，则需从每对合取的命题中任选一个，再组成析取的范式。例如 $S_2 \Leftrightarrow (P_1 \vee P_3 \vee P_5)$，$S_2 \Leftrightarrow (P_2 \vee P_4 \vee P_6)$ 等。在最坏情况下，基于状态的编码产生的 CNF 范式是指数级的。

3.2.2 基于状态的编码生成算法

本节在 SATPLAN2006 中实现基于状态的编码方式。

算法 2　state_based_encoding(*layer*)

```
for t=0,…,layer do
    for i= 0,…,gnum_ft_conn do
      if 第 i 个命题在第 t 层上不存在
  then continue
      if t== 0 then 生成初始状态公理
      if 该命题在第 t-1 层上的支持动作都不存在
      then continue
      将负的 ft 编码的文字存入已分配空间的子句里
      for j=0,…,gft_conn[i].num_A do
        if 第 j 个动作在第 t-1 层上存在
        then 将该动作的第一个已编码前提存入子句
      iterate_num ←1
Iterate_generate_clauses(t,iterate_num)
```

算法 2 实现基于状态的编码方式中命题间对应的框架，其中对于命题蕴涵上一层的支持动作，而动作又蕴涵产生它的前提的公理，由算法 3 实现。

算法 3　Iterate_generate_clauses(*t*,*iterate_num*)

```
for j =0,…,gft_conn[i].num_A do
  if 在第 t-1 层上的第 j 个动作 op 不存在
  then continue
  q ← iterate_num
  for k =1,…,gop_conn[op].num_P do
    for n =0,…,q do
      将负的 ft 编码的文字存入已分配空间的子句
      for w =0,…,gft_conn[i].num_A do
        if w == j
        then 将第 j 个动作的第 k 个命题替换原子句中该动作的命题文字
        continue
        其它动作的命题文字仍旧不变存入子句
        iterate_num++
```

算法 2 产生一个基准子句(选择第 t 层上的给定命题的每个支持动作的第一个已编码的前提)，需要迭代的子句个数(*iterate_num*)初值为 1，在外层 for 循环中选定命题的第 j 个

支持动作，在内层 for 循环中选择该支持动作的第 k 个命题，替换原来所有的基准子句中该位置的文字，其它文字不变，生成新子句，*iterate_num* 加 1，并将它们添加到基准子句集中。

在算法 2 中已产生了一个基准子句（选择第 t 层上的该命题上每个支持动作的第一个已编码的前提），需要迭代的子句个数（*iterate_num*）初值为 1，在外层的 for 循环中选定命题的第 j 个支持动作，在内层的 for 循环中选择该支持动作的第 k 个命题，替换原来所有的基准子句中该位置的文字，其它文字不变，替换生成一个子句，*iterate_num* 加 1，并将新子句添加到基准子句集中。现举例说明算法 3 的执行过程。

例 3　假设在第 t 层上有命题 S，在第 $t-1$ 层上将其作为添加效果的动作有 A_1 和 A_2，A_1 的前提为 P_1，P_2 和 P_3，A_2 的前提为 P_4 和 P_5，则首先产生一个基准的 CNF 范式：$\neg S \vee P_1 \vee P_4$（每个动作都选取第一个前题），此时 *iterate_num* 为 1，则当选定 A_1 时，其它文字不变，只替代 P_1，产生两个 CNF 范式：$\neg S \vee P_2 \vee P_4$，$\neg S \vee P_3 \vee P_4$，*iterate_num* 变为 3。继续选定 A_2，其它文字不变，只替代 P_4，产生 3 个 CNF 范式：$\neg S \vee P_1 \vee P_5$，$\neg S \vee P_2 \vee P_5$，$\neg S \vee P_3 \vee P_5$，*iterate_num* 变为 6。迭代结束。

3.3　实验结果

将基于状态的编码方式与源系统中基于动作的编码方式在后勤域和积木世界域的不同测试样例上进行对比分析，实验结果如表 4 所列。

表 4　基于状态的编码与基于动作的编码的实验结果

Problem	step	Action_based			State_based		
		realtime	var	clauses	realtime	var	clauses
Prob001	9	0.18	968	14588	0.12	329	1895
Prob005	10	4.17	4649	248501	1.92	1437	21247
Prob008	10	5.84	5291	320466	3.98	1468	24761
Prob012	12	7.09	5107	234027	1.75	1611	20933
Prob019	6	0.3	274	5422	0.26	147	17342
Prob020	6	0.14	237	4232	0.24	134	16260

实验结果表明，对于后勤域的某些问题，基于状态的编码方式明显优于基于动作的编码方式。

基于状态的编码与基于动作的编码有以下几点说明：

在积木世界域的问题中，基于状态的编码方式表现较差，在能正确求解的问题实例上的求解效率也明显低于基于动作的编码方式，且其对应子句数也高于后者。对于不能正确求解的问题实例，都是由于超出内存空间而异常终止程序的运行，因此基于状态的编码方式不适合积木世界等串行规划问题域。

在后勤域的问题中，基于状态的编码的变量个数与子句个数都远远小于基于动作的编码中的变量与子句个数，并且求解效率也优于后者。虽然基于状态的编码方式的公理目前还不足够完全，但至少表明，较基于动作的编码方式而言，基于状态的编码方式更适合后勤域等并行规划问题域。

上述两类规划域问题的最大差别就是规划解中动作的并行性和串行性，这也表明两种编码方式的侧重点是不同的。

3.4　小结

通过上述 4 种公理组合策略以及相关实验结果的对比分析，对智能规划问题的编码方式与求解效率有如下结论：

对动作互斥的放松策略，不仅显著缩小了规划问题的编码规模，并在两个域的绝大多数问题上都能提高求解效率，这也表明规划问题的编码规模与求解效率并不是一种折衷的关系。此外，这种改进有很好的通用性。

框架公理的添加在后勤域的绝大多数问题上能使求解效率有所提高，而在积木世界域的问题上则表现不佳。通过实验结果分析，可能与问题的串行和并行特性有关，因此可以在添加框架公理之前首先考察问题域的特性。

与基于动作的状态的编码方式相比，基于状态的编码方式更适宜并行规划域问题的趋势。

综上所述，可以将上述的几种组合策略应用于 SATPLAN2006 规划系统中，并根据问题域的特性等来考虑它最适宜于哪种编码方式，而不是绝对采用某种特定的编码方式。

基于可满足性的规划方法的核心是规划编码构造与 SAT 问题的判定过程。一种特定的编码方式构造的 SAT 问题对于不同的 SAT 求解器表现出的性能是截然不同的。

结束语　本文首先介绍了 SATPLAN2006 规划系统的思想和具体流程，以及将规划问题转换成 SAT 问题的不同的编码方式。

在基于 Graphplan 的编码方式的基础上，首先研究了 SATPLAN2006 规划系统中公理的存储形式和实现方法，设计了基于 PMA 的编码方式、基于 CMA 的编码方式和基于 FA 的编码方式等 3 种编码方式。分别实现了上述改进的编码方式，并对国际规划竞赛中选用的标准后勤域与积木世界域的问题样例予以测试，验证了基于 Graphplan 编码方式的改进在绝大多数情况下是有效的。

最后，本文实现了基于状态的编码方式，并对上述两个域进行测试，比较了约简动作与约简状态的两种极端方式的求解效率和编码规模。实验结果表明，在后勤域的某些问题上，基于状态的编码方式比基于动作的编码方式要有效得多。

下一步工作是基于上述构造的基于 PMA 的编码方式、基于 CMA 的编码方式、基于 FA 的编码方式、基于状态的编码方式以及基于动作的编码方式考察不同的 SAT 求解器的性能，建立针对某种特定编码方式的编码构造与 SAT 求解的最佳耦合工作方式，以实现编码方式与 SAT 求解的完美结合。

参 考 文 献

[1] McDermott D, Hendler J. Planning: what it is, what it could be, an introduction to the special issue on planning and scheduling [J]. Artificial Intelligence, 1995, 76: 1-16

[2] Green C. Application of theorem proving to planning[C]// International Joint Conference of Artificial Intelligence. 1969: 219-239

[3] Adorf H M, Johnston M D. A discrete stochastic neural network algorithm for constraint satisfaction problems[C]// Proceedings of the International Joint Conference on Neural Networks. San Diego, CA, 1990

[4] Fikes R E, Nilsson N J. STRIPS: a new approach to the application of theorem proving to problem solving[J]. Artificial Intelligence, 1971, 2(3): 189-208

[5] Kautz H, Selman B. Planning as Satisfiability[C]// Proceedings of the 10th European Conference on Artificial Intelligence. Vienna. Austria, 1992: 359-363

[6] Kautz H, Selman B, Hoffmann J. SatPlan: Planning as Satisfiability[C]// 5th International Planning Competion. 2006

[7] Kautz H, Selman B. Pushing the Envelope: Planning, Propositional Logic, and Stochastic Search[C]// Proceedings of the 13th National Conference on Artificial Intelligence. 1194-1201

[8] Kautz H, Selman B. Unifying SAT-based and Graph-based Planning[C]// Proceedings of the 16th International Joint Confe-

rence on Artificial Intelligence. 1999:318-325
[9] Kautz H, McAllester D, Selman B. Encoding Plans in Propositional Logic[C]// Proceedings of the 4th International Conference on Knowledge Representation and Reasoning. 374-385
[10] Blum A L, Furst M L. Fast planning through planning graph analysis[J]. Artificial Intelligence, 1997, 90:281-300
[11] 姜云飞,吴康恒. 智能规划的研究和应用[J]. 计算机科学,2002,2(29):100-103

(上接第 191 页)

实验中被给出。表 4 给出各数据集的描述以及各算法的实验结果,其中标黑的数据表示最优结果。其中,"Full"表示不考虑特征选择的算法。

由表 4 可以看出,大多数据集中 SFRSC 的分类精度都高于其它算法。特别地,在所有数据集中 SFRSC 的效果都要比 LS 好,证明了先验知识在特征选择中的作用。而 Full 算法却在 glass 和 ecoli 数据集中的效果更为显著,可能是由于特征个数较少的原因。

表 4 各个数据集信息及各个算法的结果

Dataset	Instance	Dimesion	Class	Full	ReliefF	LS	SFRSC
Iono	351	34	2	73.17	74.53	67.82	**76.31**
glass	214	10	7	**65.44**	62.73	58.12	60.91
sonar	208	60	2	58.86	67.21	67.73	**71.04**
ecoli	336	8	8	**70.52**	**70.47**	67.16	66.58
wine	178	13	3	77.02	78.35	62.05	**89.15**
KDD	20995	42	5	62.49	71.04	71.39	**73.55**

为了进一步与其它方法进行比较,实验还考虑了随着已标记样本个数或特征子集大小的变化,基于各算法得到的特征子集,分类精度的变化情况。图 5 以数据集 wine,sonar 为代表说明特征子集的大小对分类精度的影响,可以看出 SFRSC 基本上都是优于 ReliefF 和 LS 的。由于选取的已标记样本较少,该对比结果体现了 SFRSC 能充分利用标记样本和未标记实例携带的信息。随着所选特征数目的增加,各算法的性能有所提高。当接近于原始特征集大小时,算法的性能会有所降低。该现象在 wine 数据集中更明显,这可能是由于后期加入的 feature 对精度的影响。

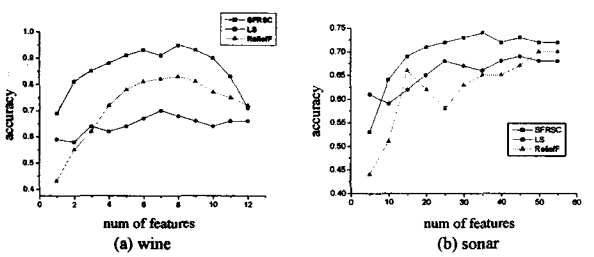

图 5 特征子集的大小对分类精度的影响

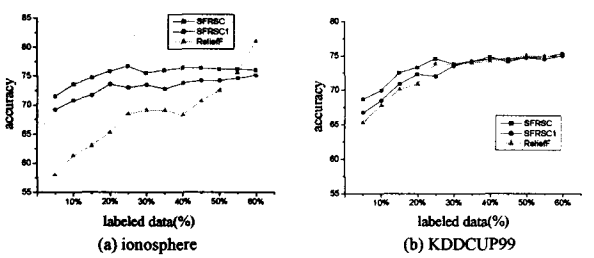

图 6 已标记样本集合的大小对分类精度的影响

图 6 在数据集 Ionosphere 和 KDDCUP99 上给出不同大小的标记样本集与分类精度的关系。特征数目固定取 11。由于算法 LS 的性能与标记样本的个数无关,这里考虑的是 SFRSC-1(与 SFRSC 的区别仅在于对重叠部分的处理不同)。如图 6 所示,总体来说增加标记样本的个数可以提高算法的精度。当标记样本较少时,精度的增长较快,并且 SFRSC 和 SFRSC-1 的优势也较为明显。但标记样本占整个数据集的比例超过一定阈值(大约是 25%)时,曲线变得比较平缓,精度增长减慢。所以过多的先验知识并不能过快地提高分类精度,SFRSC 可以基于小标记样本集得到较好的学习效果。另外,对于相同大小的标记样本集,SFRSC 总是优于 SFRSC-1,这也证明了式(5)优于式(4)。

结束语 在已标记样本有限的前提下,本文基于 RSC 模型提出了一种半监督特征选择算法 SFRSC。该算法是一个迭代的过程,在特征子集对应的空间中,将已标记样本作为核心点,将类标号扩展到未标记样本上,并以 S 的最终划分的复合结果作为衡量该特征子集的标准。同时还考虑了一个样本被标记为多个类的情况。实验证明,该算法是比较有效的。

本文的工作基于这样的假设:初始的已标记样本集中包含所有的类别信息。下一步将考虑在确定类标号的样本集中,某些类别没有对应的样本。另外,本文考虑的先验知识是类别信息,但往往样本之间的约束关系(must-link 和 cannot-link)更容易得到,在这种情况下考虑特征选择也是非常有意义的。

参 考 文 献

[1] Yang K, Yoon H, Shahabi C. A Supervised Feature Subset Selection Technique for Multivariate Time Series
[2] Liu H, Yu L. Toward Integrating Feature Selection Algorithms for Classification and Clustering[J]. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(4):491-502
[3] Zhao Zheng, Liu Huan. Searching for Interacting Features[C]// ijcai 2007
[4] Seeger M. Learning with labeled and unlabeled data[R]. 2000
[5] Houle M E. Clustering without data : the GreedyRSC heuristic [C]//Proc. International Workshop on Data-Mining and Statistical Science(DMSS 2006). Sapporo, Japan, September 2006:62-69
[6] Houle M E, Grira N. A Correlation-Based Model for Unsupervised Feature Selection[C]//CIKM'07
[7] Izutani A, Uehara K. A Modeling Approach Using Multiple Graphs for Semi-Supervised Learning[J]. Discovery Science, 2008:296-307
[8] Nakatani Y, Zhu K, Uehara K. Semisupervised learning using feature selection based on maximum density subgraphs[J]. Systems and Computers in Japan(SCJAPAN), 2007, 38(9):32-43
[9] Ren Jiangtao, Qiu Zhengyuan, Fan Wei, et al. Forward Semi-Supervised Feature Selection[C]//PAKDD08. 2008
[10] Zhao Z, Liu H. Semi-supervised Feature Selection via Spectral Analysis[C]//SIAM International Conference on Data Mining (SDM-07). 2007
[11] Houle M E. Clustering without data: the relevant set correlation model[C]//Proc. International Workshop on Data-Mining and Statistical Science (DMSS 2006). Sapporo, Japan, September 2006:54-61
[12] Handl J, Knowles J. Semi-supervised feature selection via multiobjective optimization[C]//IJCNN06. 2006