

规划问题编码为约束可满足问题的研究^{*})

伍丽华^{1,2} 陈蔼洋¹ 姜云飞¹

(中山大学软件研究所 广州 510275)¹ (中山大学数学与计算科学学院 广州 510275)²

摘要 基于约束可满足问题的规划求解是研究智能规划的重要技术方法。把规划问题编码为约束可满足(CSP)问题,是这种规划求解方法的关键技术之一。本文介绍把规划问题编码为约束可满足问题的方法,及一些已有的并且已经用于规划的可满足过程,并对这些编码方法做进一步的研究,主要讨论领域知识在编码方法中的应用,提出在编码求解中加入领域知识的观点。

关键词 规划问题, CSP, SAT, 领域知识

The Research of Constraint Satisfaction Problems Encoding of Planning Problems

WU Li-Hua^{1,2} CHEN Ai-Yang¹ JIANG Yun-Fei¹

(Software Research Institute of Sun Yat-Sen University, Guangzhou 510275)¹

(School of Mathematics and Computational Science, Sun Yat-Sen University, Guangzhou 510275)²

Abstract The planning solution based on constraint satisfaction problem(CSP) is one of the most important methods of researching intelligent planning. Encoding planning problems into CSPs play a major role in planning solving. In this paper, we focus on the encoding of planning problems into CSP and describe some existing satisfiability procedures that have been used extensively for planning, and the discussion of the application problem about domain knowledge in encoding approach is presented.

Keywords Planning problem, CSP, SAT, Domain knowledge

1 引言

智能规划最早的求解方法是通过归结定理证明的方法进行规划求解。近 10 年来,智能规划的研究在问题描述和问题求解两方面都得到了新的进展,如 Kautz 等学者把规划问题求解转化为可满足(SAT)问题^[4~6],利用在约束可满足问题算法上的突破,有效地解决了部分规划问题,与定理证明式求解方法有根本的区别。将规划问题转化为约束可满足问题来求解的规划方法,是现代智能规划系统 5 大类方法之一,是与领域无关的智能规划算法,代表规划系统有 Blackbox^[3]。基于此理论的 Blackbox 规划系统在第一届智能规划系统的比赛中表现了非凡的解决问题能力,开辟了解决规划问题的又一新途径。

规划可满足问题是把一个规划问题映射到一个已有有效算法或过程的著名问题,并从新问题的解中提取一个规划。其中一种经典技术是把规划问题编码为约束可满足问题(Constraint Satisfaction Problem)。而最常见的是编码为命题可满足(SAT)问题,其一般的思想方法是把一个规划问题公式化为一个命题可满足问题或其它的约束可满足问题,再利用 SAT 或 CSP 算法求解,在解中提取规划解。

SAT 与 CSP 算法的研究已很广泛深入,且有较成熟的有效算法,目前不少算法技术已应用到解决规划问题中。这些算法技术的研究与不断改进,提高了规划对于解决相关大问题和复杂问题的能力。但编码中的一些问题,如编码的存储空间、编码的复杂度等等,仍值得深入探讨。

本文简述把规划问题编码为 SAT 或 CSP 问题的方法,及一些已有的并且已经用于规划求解的可满足过程,并对这些编码方法做进一步的研究,主要讨论领域知识对编码方法

的影响。

2 规划问题与约束可满足问题

本文讨论的规划均为经典规划问题,即满足限定状态系统的 8 个限定^[1]。

2.1 规划问题

经典规划问题的形式化描述为:给定规划的一般性模型 $\Sigma = (S, A, \gamma)$, 其中: $S = \{s_1, s_2, \dots\}$ 为状态有限集或递归可数集; $A = \{a_1, a_2, \dots\}$ 为动作有限集或递归可数集; $\gamma: S \times A \rightarrow 2^S$ 为状态转换函数,以及给定初始状态 s_0 , 目标状态 S_g 的子集,求动作序列 $\langle a_1, a_2, \dots, a_k \rangle$, 对应于状态转换序列 (s_1, s_2, \dots, s_k) , 使得 $s_1 \in \gamma(s_0, a_1), s_2 \in \gamma(s_1, a_2), \dots, s_0 \in \gamma(s_k, a_k)$, 且 $s_k \in S_g$, 则 $\langle a_1, a_2, \dots, a_k \rangle$ 为 Σ 的一个规划解。

简单地讲,给出一个任务的初始状态、目标状态和所有可行的操作,规划问题就是如何自动找到完成从初始状态到目标状态这一任务的动作序列。

例 1 考虑一个简化的码头货运例子:1 个载货车(robot) r 和 2 个地点 l_1, l_2 , 载货车可在 2 个地点之间移动,且能装载(load)和卸载(unload)集装箱。初始状态是载货车在 l_1 , 且为 unload, 集装箱 c 在地点 l_1 ; 目标状态是把集装箱 c 移到地点 l_2 。一个规划解为: load(l_1, c, r), move(r, l_1, l_2), unload(l_2, c, r)。

2.2 约束可满足问题

约束可满足问题(CSP)是问题解决的一个普遍而有效的方法,其应用很广泛,在规划与时间表、计算机视觉、模式识别、CAD、建模和决策支持系统等领域都有其有效的应用。

根据文[1],在有限域上的 CSP 定义为一个三元组 $P = (X, D, C)$, 其中: $X = \{x_1, \dots, x_n\}$ 是 n 个变量的有限集, $D =$

^{*}) 国家自然科学基金项目,编号:60173039。伍丽华 高级工程师,博士研究生,研究方向为基于模型的诊断和智能规划;姜云飞 教授,博士生导师,主要研究领域为自动推理、基于模型的诊断及智能规划。

$\{D_1, \dots, D_n\}$ 是变量的有限域的集合, $x_i \in D_i, C = \{c_1, \dots, c_m\}$ 是约束的有限集, k 个变量的约束 c_j 限定了 k 个变量 $\{x_{j1}, \dots, x_{jk}\} \subseteq X$ 的子集所能取的值。 c_j 定义为一个笛卡尔乘积的子集: $c_j \subseteq D_{j1} \times \dots \times D_{jk}$, 即对这些变量, 约束允许可取值的元组的集合为: $\{(v_{j1}, \dots, v_{jk}) \in D_{j1} \times \dots \times D_{jk} \mid (v_{j1}, \dots, v_{jk}) \in c_j\}$ 。

一个约束可满足问题的一般公式是这样的: 给定变量集合及其相应的域, 以及建立在相容的变量值上的约束集, 问题的求解是对每一个变量在其域中找到一个值, 使其满足约束。

2.3 命题可满足问题

命题可满足问题(SAT)是约束可满足问题(CSP)的一个特例。

SAT问题是最早的 NP-完全问题, 已经用于规划求解的著名算法有 Davis-Putnam 过程和基于随机过程的 Local-Search-SAT 算法、Basic-GSAT 算法和 Iterative-Repair 算法等等, 其中 Davis-Putnam 过程可靠且完备, 而基于随机过程的算法可靠, 但不完备。

规划问题编码为 SAT 问题, 实际上是把一个规划问题编码为一个命题公式(CNF), 然后判定该命题公式是否可满足的问题, 而判定该命题公式是否可满足, 可利用可满足决定过程看公式在对命题变元赋值后是否可满足, 若可满足, 从赋值决定中提取一个规划。

本文不讨论 SAT 或 CSP 的算法, 主要讨论规划问题在编码为 SAT 或 CSP 过程中的问题。

3 编码问题

这里讨论的规划问题均为有界规划问题, 即把规划问题限定为对于某个给定正整数 n , 找到长度为 n 步的规划问题, 每个 $i(0 \leq i \leq n)$ 称为规划问题的每一步。文[1]中介绍了一些编码方法, 这里仅做简述。

3.1 编码为 SAT

把有界规划问题编码为可满足问题中, 有界规划问题的状态和动作描述对应于每一步中描述状态和动作的命题(编码为 SAT), 从第 0 步(对应于初始状态)到第 n 步(对应于目标状态)。把一个规划问题编码为一个命题公式 Φ 。

编码有界规划问题的公式 Φ , 可构建为若干公式的合取范式(CNF)。这些公式描述初始状态和目标状态(分别在第 0 步和第 n 步用基原子列出), 以及描述 0 至 n 步的动作行为(如前件和后果)。

Φ 由下列 5 种公式集合组成, 其中第 i 步的一个状态的原原子公式记为 f_i 。例如, 表示第 i 步 robot r_1 在 location l_1 的状态基原子公式 $at(r_1, l_1, i)$ 记为 f_i 。表示在第 i 步执行动作的命题公式记为 a_i , 例如, $move(r_1, l_1, l_2, i)$ 记为 a_i 。

(1) 初始状态

初始状态编码为状态基原子公式的合取命题, 分别是在初始状态为真的状态基原子公式和不为真的状态基原子公式的否定, 全部在第 0 步罗列出来。

$$\bigwedge_{f \in s_0} f_0 \wedge \bigwedge_{f \notin s_0} \neg f_0 \quad (1)$$

(2) 目标状态集

目标状态由在所有目标状态中为真的状态基原子公式的合取部分描述, 目标状态集编码为必须在第 n 步为真的状态基原子公式的合取命题。

$$\bigwedge_{f \in K} f_n \wedge \bigwedge_{f \notin K} \neg f_n \quad (2)$$

(3) 动作

设 A 为所有可能的动作集合, 对每个 $a \in A$ 和每个 $0 \leq i$

$\leq n-1$, 有:

$$a_i \Rightarrow \left(\bigwedge_{p \in \text{precond}(a)} p_i \wedge \bigwedge_{e \in \text{effects}(a)} e_{i+1} \right) \quad (3)$$

a_i 是一个可应用的有效果的动作, 被编码为一个公式, 说明若有动作在某一步发生, 则其前件一定在此步为真, 且其效果将在下一步为真。

(4) 框架公理

这里采用解释框架公理。

有些动作是仅改变其效果的状态基原子公式, 例如移动一个集装箱, 不改变其它集装箱的位置, 对这一事实进行公式化和推理称为框架(frame)问题。更准确地说, 如果一个在第 i 步不为真的状态基原子公式 f , 在第 $i+1$ 步为真, 则正效果中有状态基原子公式 f 的动作之一已经在第 i 步执行。类似地, 若 f 在第 i 步为真, 而第 $i+1$ 步不为真, 则负效果中有状态基原子公式 f 的动作已经执行。于是我们用命题集列举已发生的动作集, 以考察状态的变化。这些公式称为解释框架公理(explanatory frame axioms), 对每个状态基原子公式 f 和每一 $0 \leq i \leq n-1$, 有:

$$\begin{aligned} (\neg f_i \wedge f_{i+1} \Rightarrow (\bigvee_{a \in A | f_i \in \text{effects}^+(a)} a_i)) \wedge (f_i \wedge \neg f_{i+1} \Rightarrow \\ (\bigvee_{a \in A | f_i \in \text{effects}^-(a)} a_i)) \end{aligned} \quad (4)$$

(5) 完全排除公理

完全排除公理是为了保证每一步只有一个动作发生。对每个 $0 \leq i \leq n-1$ 和每个不同的 $a_i, b_i \in A$, 有:

$$\neg a_i \vee \neg b_i \quad (5)$$

以上式(1)~(5)的合取式, 构成把有界规划问题编码为可满足问题的命题公式 Φ 。

从 Φ 的模型中提取一个规划解: 给定一个长度为 n 的有界规划问题, 提取出命题变量序列 $\langle a_1(0), \dots, a_n(n-1) \rangle$, 其中 $a_i(j)$ 为命题变量, 表示第 j 步的动作 a_i , 使得模型中所有 $a_{i+1}(i), 0 \leq i \leq n-1$ 都为 true, 则提取的规划解为 $\langle a_1, \dots, a_n \rangle$ 。

例 2 假设在例 1 中, 限定规划长度为 $n=3$, 操作为:

$move(r, l, l')$;; robot r at location l moves to a connected location l'

precond: $at(r, l)$

effects: $at(r, l'), \neg at(r, l)$

$load(c, r, l)$;; robot r loads container c at location l

precond: $at(r, l), in(c, l), unloaded(r)$

effects: $loaded(r, c), \neg in(c, l), \neg unloaded(r)$

$unload(c, r, l)$;; robot r unloads container c at location l

precond: $at(r, l), loaded(r, c)$

effects: $unloaded(r), in(c, l), \neg loaded(r, c)$

初始状态和目标状态分别编码为公式(init)和(goal):

(init) $at(r, l_1, 0) \wedge \neg at(r, l_2, 0)$

(goal) $at(r, l_2, 1) \wedge \neg at(r, l_1, 1)$

为节省篇幅, 以下讨论仅以动作 $move$ 为例, 列出相关的命题。

动作编码为:

$$(move1) move(r, l_1, l_2, 0) \Rightarrow at(r, l_1, 0) \wedge at(r, l_2, 1) \wedge \neg at(r, l_1, 1)$$

$$(move2) move(r, l_2, l_1, 0) \Rightarrow at(r, l_2, 0) \wedge at(r, l_1, 1) \wedge \neg at(r, l_2, 1)$$

解释框架公理为:

$(at_1) \neg at(r, l_1, 0) \wedge at(r, l_1, 1) \Rightarrow move(r, l_2, l_1, 0)$

$(at_2) \neg at(r, l_2, 0) \wedge at(r, l_2, 1) \Rightarrow move(r, l_1, l_2, 0)$

$(at_3) at(r, l_1, 0) \wedge \neg at(r, l_1, 1) \Rightarrow move(r, l_1, l_2, 0)$

$(at_4) at(r, l_2, 0) \wedge \neg at(r, l_2, 1) \Rightarrow move(r, l_2, l_1, 0)$

完全排除公理为:

$$\neg \text{move}(r, l_1, l_2, 0) \vee \neg \text{move}(r, l_2, l_1, 0)$$

把规划问题编码为命题公式后有一模型,赋予下列各式为 true:

$$\text{move}(r, l_1, l_2, 1), \text{load}(l_1, c, r, 0), \text{unload}(l_1, c, r, 2)$$

按照规划步数 i , 提取的规划为序列:

$$\langle \text{load}(l_1, c, r), \text{move}(r, l_1, l_2), \text{unload}(l_2, c, r) \rangle$$

值得注意的是,如果规划无解,只能说明不存在长度为 n 的规划解,并不表示原问题无解。

3.2 编码为 CSP

CSP 为约束可满足问题,考虑到 CSP 中约束的表示,在规划问题的几种不同表示方式中,采用状态变量的形式表示更符合约束的表示形式,因此把规划问题编码为 CSP。首先要把有界规划问题 P 转化为约束可满足问题 P' ,也即使规划问题用状态变量的形式表示出来,然后才通过 CSP 求解,其主要步骤如下。

Step 1: 定义 P' 的 CSP 变量;

Step 2: 初始状态 s_0 和目标状态 g 的约束编码;

Step 3: 动作约束编码;

Step 4: 框架公理约束编码。

CSP 的主要算法有 Backtrack 搜索算法和运用过滤技术的 Forward-checking 算法。

SAT 编码与 CSP 编码有很多类似,如都需要对初始状态、目标状态、动作及框架公理进行编码。但也有所区别,如在规划问题的表示形式上,SAT 编码中规划问题采用经典表示方法,而 CSP 编码中规划问题采用状态变量表示法。此外,采用的框架公理也有所不同,SAT 中采用解释框架公理,是一种完全排除公理,而 CSP 采用的是经典框架公理。

4 领域知识在基于约束可满足规划中的应用

智能规划的求解是困难的,规划问题的时间、空间复杂度都是 NP 完全的。目前,虽然与领域无关的规划系统取得了较大的突破,但是如果不利用到相关的领域知识,则很难提高规划系统的效率。

什么是领域知识?领域知识通常是指各应用领域中的专家知识或经验知识,与领域有较强的依赖关系。

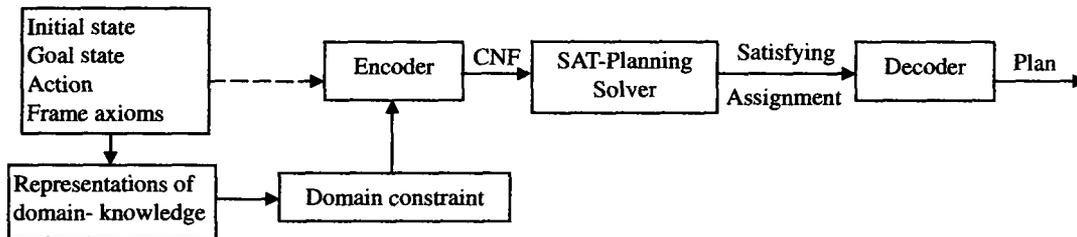


图1 领域约束的 CSP 规划系统体系结构

在此体系结构中,纵向看,我们要从问题的描述、领域知识的表述以及领域知识如何表示为领域约束上进行研究,也即研究领域知识的表达方式和表达能力。

在规划中如何利用相关的领域知识,首先是领域知识的表达与应用。文[2]中提出利用领域知识,介绍了一种将领域知识表示为领域约束,并添加到与领域无关的通常规划系统中。考虑了领域知识在规划中的应用,将领域知识表示为领域约束添加到规划系统中,将领域约束分为3种:对象约束过程约束和时序约束。采用对象约束来表达状态中对象之间的关系,采用过程约束来表达动作之间的关系,采用时序约束表

4.1 领域无关与领域依赖

领域无关与领域依赖的问题,实际上是系统通用性与专家系统的问题。领域无关,则系统通用性较强;领域知识过多,则成了专家系统。这两方面存在着一个平衡点。但适当和有效的领域知识对提高规划系统的效率其作用是显著的。

通用规划系统由于其通用性,一般不考虑在应用领域上任何隐含的有用领域信息,系统要考虑许多在实际应用领域里根本不可能产生的状态,或在某些状态下根本不可能采取的动作。其解决问题的效率非常低,因为在实际应用中的规划问题必然与问题领域有很强的依赖性。事实上,在实际应用中,由于缺乏领域知识,这些规划系统的效率都是很差的。对稍微复杂一点的问题,就会因为空间被耗尽或时间太长而无法得出结果。

在上述规划问题的编码中,因为不考虑相关的领域知识,完全按照编码步骤进行编码,所以命题数量非常大,在实际应用中将产生很多冗余。如果动作的编码采用规则编码法,且框架公理采用解释框架公理,则规划的复杂度为 $O(n|F||A| + n|A|^2)$,其中 n 为规划步数, $|F|$ 为状态谓词数, $|A|$ 是基动作数。

目前很多智能规划系统为了提高求解效率,几乎都采用了启发信息,或增加领域知识,但启发式知识用在与领域无关的规划系统上比较困难。因此,将跟领域密切相关的知识添加到通用规划系统上,使得它成为领域约束规划,将能够更好地解决实际应用的问题^[2]。领域依赖规划的基本思想是专家们通常比较了解该领域的各种知识,有时对领域专家来说甚至是十分简单的知识,都会对实用规划系统的效率产生重要影响,这些领域知识可以用来有效地帮助软件设计人员提高应用系统的效率。

4.2 领域知识的应用

在基于 CSP 或 SAT 的规划求解中,利用领域知识无疑可以提高问题的描述及编码求解的效率。但如何使用领域知识却是一个新的课题,在这问题上做具体深入的研究是有益的。在这种求解技术中,首先需要挖掘领域知识,这些领域知识可做进一步处理,主要方法之一是由领域约束表示作用于各命题,然后再进行编码。增加了领域约束的 CSP 规划系统体系结构如图1所示。

达动作跟状态中对象之间的关系。通过领域约束对领域知识进行表示,对象约束、过程约束和时序约束三者相互作用来减少规划系统的搜索空间,利用领域知识来提高描述能力和求解效率。但是,对于解决统一的领域约束形式化表示和领域约束的自动抽取,仍没有一个好的方法。具体到各种规划方法,如何添加领域知识,需要深入研究。

在规划问题的编码过程中,如果考虑利用领域知识,则可减少命题变元数,从而减少后续编码的复杂度,提高规划的描述能力和求解效率。如在例2中,对货物运输领域抽取相

(下转第292页)

3.2 SOA 分析与设计

分析业务流程,按照 SOA 的设计原理对系统进行设计,将每个业务逻辑都打包成服务,并发布到电信企业总的服务注册中心,资源管理系统、计费帐务系统和客户服务系统也按照此流程打包成 Web 服务发布到注册中心,当前流行的服务注册中心有 UDDI 和 WSIL,SOA 运行原理如图 4 所示。

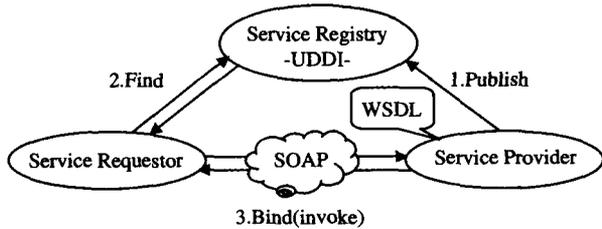


图 4 SOA 的运行原理

申请系统发出 SOAP 请求到 UDDI 中心,查询是否有存在可用的导线和端口资源的服务(UDDI 提供了这项服务),若有,UDDI 到资源管理系统中的调用查询服务。若可用,将服务运算结果返回给中心,再转回给申请系统^[5]。

通过服务的不断打包和发布,可用的资源越来越多,电信企业也可以根据服务的增多灵活地对业务进行编排,通过

(上接第 189 页)

关的领域知识。通过分析可以得到,前提 location 不能被动作所改变,at 和 in 可以在一起编码,因为一个集装箱 c 可以在一个 location,同时也可以 robot 上。另外,货物运输规划中的货物必须装上车后才能开车,货物也要等到车到站后才能卸车,说明对象之间、动作之间相互关联或不关联。将这些领域知识表示为领域约束,再将这些领域约束添加到 SAT 或 CSP 规划系统中,复杂度 $O(n|F||A|+n|A|^2)$ 中的 $|F|$ 与 $|A|$ 将大大减少。

但是,当对实际的应用领域做具体的实用规划时,这些实用规划系统的规模通常是较大的,包含大量的复杂的动作集合。根据以往学者们对领域规划问题的研究状况,我们要从领域知识的表达方式和表达能力上进行研究,使得领域知识的描述能力尽量满足现实规划问题的需要,并且使得规划问题的求解更加自然,以利于提高效率。

规划系统体系结构从横向看,需要对领域约束如何加入到编码器中,也即在规划求解的过程中添加领域知识进行研究,从而实现领域规划问题。也即研究如何将领域知识自然地添加到目前先进的 CSP 或 SAT 规划系统中。相当于对规划问题增加了若干有用的约束,使得问题的解空间变得较小,从而提高了规划问题解决的速度和求解结果的质量,使得系统达到或接近实际应用水平。

在领域依赖规划方面,有两种方法利用领域知识:一种是层次任务分解,另一种是搜索策略控制。基于 CSP 的规划求解是一种搜索策略,所以考虑的领域依赖规划也是基于搜索策略控制。

在编码方法中应用领域知识的基本思想是,针对每个具体测试问题来手工抽取问题知识,做些预处理,或通过挖掘和机器学习,建立一套机制来表达领域知识,使这些领域知识能方便地添加到基于 CSP 或 SAT 的规划系统中,使得规划系统的表达能力和求解效率有所提高。实现的关键技术是机器学习,即通过加入学习功能,使得系统能从以前规划的结果中自动抽取相关领域知识,并将其作用在规划求解上,增加了

SOA 的即插即用特性(Plug & Play),不必担心新开发业务对旧系统的影响。

结束语 从上述电信领域的应用实例不难看出,利用 SOA 来进行商务集成和企业应用集成,有着明显的优势^[6]:

(1)标准的互操作性,根据电信系统不同时期开发的系统,不同的操作环境,都可以发布成 Web service,并在 UDDI 中心注册;

(2)商务规则定义方便,并可根情况灵活地变化;

(3)松散的耦合,能够即插即用;

(4)服务的重复使用,其长期作用在于减少企业的冗余功能,简化基础架构,降低维护代码的成本。

参考文献

- Zimmermann O, Krogdahl P. Elements of Service-Oriented Analysis and Design. [M]. 北京:中信出版社,2003
- IBM redbook. Patterns: Implementing an SOA using an Enterprise Service Bus [M]. 2004
- 吴信才,刘少雄. 面向服务分析设计原理[M]. 北京:机械工业出版社,2004
- Han J, Kambr. Telecom in EAI [M]. Beijing: Higher Education Press. 2004
- 张议,李俊仁. 浅谈 SOAP [J]. 计算机应用,2002
- 李健. 软件设计新方法[M]. 北京:北京航空航天大学出版社,2005

领域知识约束,减少了搜索空间,也提高了求解效率。

在规划过程中规划知识被人们广泛关注,智能规划问题的操作的前提之间有很强的依赖与冲突关系,一个操作的使用常常使另一个操作无法执行,甚至导致最终目标无法实现。因此,在智能规划中领域知识的引入也有助于解决操作间的冲突。

结束语 本文提出了在规划问题编码为约束可满足问题过程中,考虑领域知识,并利用领域知识,以提高编码过程及求解效率的观点。在如何使用领域知识,从而提高规划系统的描述能力和求解效率,这个工作仍需做深入的研究。

虽然讨论的规划均为经典规划问题。实际上并不局限于经典规划问题,“可满足规划”方法近年来已扩展到各种规划问题,尤其是不确定领域的规划问题。

参考文献

- Ghallab M, Nau D, Traverso P. Automated Planning Theory and Practice. Morgan Kaufmann Publishers, 2004
- 吴康恒,姜云飞. 基于模型检测的领域约束规划. 软件学报,2004(11)
- 凌应标. 基于 SAT 的规划理论与算法研究.[学位论文]. 广州:中山大学,2005
- Kautz H, McAllester D, Selman B. Encoding plans in prepositional logic. In: Proc. 5th Int Conf. Principles of Knowledge Representation and Reasoning, 1996
- Kautz H, Selman B. Blackbox: A new approach to the application of theorem proving to problem solving. AIPS98 Workshop on Planning as Combinatorial Search, 1998
- Kautz H, Selman B. Pushing the envelope: Planning, prepositional logic, and stochastic search. In: Proc. 13th Nat Conf. AI, 1996
- Koehler J. Planning under Resource Constraints. 13th European Conference on Artificial Intelligence, ECAI98, 1998
- Bayardo R, Schrag R. Using CSP look-back techniques to solve real-world SAT instances. In: Proc. 14th Nat Conf. AI, Providence, R I, July 1997
- de Kleer J. A comparison of ATMS and CSP techniques. In: Proc. IJCAI-89, 1989
- Haas A. The case for domain-specific frame axioms. The Frame Problem in Artificial Intelligence. In: Proc. of the 1987 Workshop. Morgan Kaufmann, 1987