软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2019,30(12):3590−3604 [doi: 10.13328/j.cnki.jos.005598]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

一种量词约束满足问题的混合易解子类*

高 健, 陈 荣, 李 辉

(大连海事大学 信息科学技术学院,辽宁 大连 116026)
通讯作者: 陈荣, E-mail: rchen@dlmu.edu.cn

摘 要: 量词约束满足问题是人工智能和自动推理领域的一个重要问题.寻找多项式时间易解子类,是研究此类问题计算复杂性的关键.通过分析二元量词约束满足问题中的约束关系特征,以及量词前缀中的全称量词排列的顺序,提出了针对全称量词变量子结构的易解性质的分析方法.通过该方法,扩展了已知的基于 Broken-Triangle Property 的多项式时间易解子类,提出了一个更一般化的量词约束满足问题的混合易解子类.讨论了易解子类在问题结构分析中的一个应用,即通过易解子类确定量词约束满足问题的隐蔽变量集合,并通过实验分析不同易解子类所确定的集合大小.实验改造了基于回溯算法的求解器,在回溯过程中加入了易解子类的识别算法,并采用随机约束满足问题的生成模型作为测试基准.通过对比实验,验证了提出的多项式时间易解子类可以识别出更小的隐蔽变量集合,因此,新提出的易解子类在确定隐蔽变量集合方面更具优势.最后阐述了其他已有的混合易解子类也可以通过类似方法进行扩展,从而得到更多的一般化的理论结果.
关键词: 量词约束满足问题;易解子类;隐蔽集;回溯算法
中图法分类号: TP181

中文引用格式: 高健,陈荣,李辉.一种量词约束满足问题的混合易解子类.软件学报,2019,30(12):3590−3604. http://www.jos.org.cn/1000-9825/5598.htm
英文引用格式: Gao J, Chen R, Li H. Hybrid tractable class for quantified constraint satisfaction problems. Ruan Jian Xue Bao/Journal of Software, 2019,30(12):3590−3604 (in Chinese). http://www.jos.org.cn/1000-9825/5598.htm

Hybrid Tractable Class for Quantified Constraint Satisfaction Problems

GAO Jian, CHEN Rong, LI Hui

(College of Information Science and Technology, Dalian Maritime University, Dalian 116026, China)

Abstract: Quantified constraint satisfaction problem (QCSP) is a central problem in artificial intelligence and automated reasoning. The tractable class is an important method to analyze its computation complexity. This study proposed a new method to determine tractability of quantified variables by analyzing constraint structures and the ordering of universally quantified variables in the prefix on a binary QCSP. Based on this method, the existing tractable class was extended with the broken-triangle property, and then a more generalized hybrid tractable class was proposed. Furthermore, an application was presented that was identifying backdoor sets through the new tractable class, and the experimental results were analyzed to show the size of backdoor sets identified by those hybrid tractable classes. To perform the experiment, a state-of-the-art QCSP solver was modified based on a backtracking algorithm by integrating a backdoor set detection module, and the advantage of the new generalized tractable class is shown where the size of backdoor set identified by it is smaller than the existing one on randomly generated instances. Finally, it is indicated that the method proposed in this study can be employed to extend other hybrid tractable classes.
Key words: quantified constraint satisfaction problem (QCSP); tractable class; backdoor set; backtracking algorithm

* 基金项目: 国家自然科学基金(61402070, 61672122, 61602077)
Foundation item: National Natural Science Foundation of China (61402070, 61672122, 61602077)
收稿时间: 2017-12-28; 修改时间: 2018-03-17; 采用时间: 2018-05-01

约束是一个现实世界普遍存在的概念,它在工业领域和日常生活中都有体现,如生产过程中加工产品的先后顺序、字处理软件排版对齐的方式、棋类游戏中的限制规则等,都是常见的约束关系.约束满足问题(constraint satisfaction problem)[1]是人工智能领域中核心的研究方向之一,它用来刻画客观存在的约束关系,并被广泛地应用于建模和求解各类实际问题[2−4],如作业调度、智能规划、产品配置等.量词约束满足问题(quantified constraint satisfaction problem)[5]是经典约束满足问题的重要扩展,在该问题中引入了一个变量顺序的前缀,每个变量还被一个量词所限定,量词的种类有全称量词和存在量词.求解量词约束满足问题是十分复杂的,对于由全称量词限定的变量,求解过程需要为其每个可能的取值找到一组其后续变量的赋值,并满足它们之间的约束关系.从计算复杂性的角度来讲,量词约束满足问题是 PSPACE-完全的.而作为该问题的子问题,量词布尔范式(quantified Boolean formula)[6]是 PSPACE-完全类中的原型问题,即最早被证明是 PSPACE-完全的,因此,量词约束满足问题的求解难度通常难于传统的约束满足问题.近些年,量词约束满足问题在博弈论、机械产品设计、动态调度等领域被广泛应用[7−10].例如,在博弈游戏中,人们往往需要找到一个策略,使得无论对方如何选取决策都能保证是可以获胜的;在工业产品配置中,为了提高产品的竞争力,生产商需要寻求一种满足客户各种可能需求的配置方案.量词约束优化问题也得到关注,Lallouet 等人[11,12]提出了 Minimax 加权约束满足问题,该问题模型用于存在竞争对手的交互式决策问题中,目标是在竞争对手采取对自己最不利的决策时,最优化给定的目标函数.

由于量词约束满足问题可以用于描述诸多实际问题,因此对该问题的性质分析受到了学者的重视,而其中,对约束特性的分析是设计约束求解算法的基础[13,14].虽然目前求解算法主要采用基于回溯策略的系统搜索方法,如 Gent 等人提出的 QCSP-Solve[2,15]求解算法,但是算法中通常需要启发式和局部结构化简技术,如基于相容性技术的变量值域约减和预处理方法.另外,利用问题的结构特征设计启发式规则也是一种重要的方法.例如,利用子问题结构的相似性搜索结果的复用技术在求解器中也经常被使用.Bacchus 等人[16]提出了基于解重用的回跳技术,该技术在搜索算法找到满足约束的一个分支后,判断当前存在量词变量的赋值是否同时也是其他分支(即其他的全称量词变量赋值组合)的解,与当前赋值相容的其他分支则被标记,在后续的搜索中不再展开这些被标记的分支,从而减少了重复的计算;而求解算法 Block-Solve[17]先将问题分解,求解若干子问题后,将计算结果整合成整体问题的解.另一方面,隐蔽集(backdoor set)[18]是分析问题结构特征的一项重要理论工具.它是一个变量的集合,当该集合中所有变量被赋予某一组取值后,剩余的子问题可以很容易地(通常在多项式时间内)被求解.隐蔽集理论可以用来识别问题中隐含的易解结构,通过它,可以找出哪些问题是多项式时间内可解的,或者识别出多项式时间内可解的部分,并从问题中将容易求解的部分分离出来.该项理论在分析约束满足问题和量词约束满足问题中都起到了重要的作用[19−21].然而在寻找一个隐蔽集的同时,需要确定剩余的变量组成的子问题是否是多项式时间可解的,所以隐蔽集的研究又是以多项式时间易解子类(以下简称易解子类)为基础的[22].

约束满足问题的易解子类[23]是指该问题集合中的特例子集,这个子集中的问题实例可以在多项式时间内求解.寻找易解子类是约束满足问题的一项重要的基础研究.而量词约束满足问题是 PSPACE-完全的,所以找出该问题的可解子类,可以更大幅度地化简问题结构并提高求解效率.因此,易解子类的研究也在量词约束满足问题中展开,并为隐蔽集的分析提供了理论基础.寻找更一般化的易解子类是获得更小隐蔽集的关键所在,本文研究量词约束满足问题的易解子类,首先提出了针对全称量词变量的相容性概念并分析了其相关算法的时间复杂度.基于此概念,提出了一种新的混合易解子类,该易解子类一般化了基于 Broken-Tringle Property(BTP)的易解子类[24].本文还详细分析了所提出的混合易解子类的相关性质,其中包括子类识别问题所需的时间复杂度,并证明部分变量赋值后的子问题保持了量词相容性和易解属性.本文还在约束满足问题的隐蔽集定义基础上定义了量词约束满足问题的隐蔽集,通过改造经典的回溯求解算法,实验分析了所提出的易解子类对隐蔽集大小的影响.通过与原有基于 BTP 的易解子类的实验结果比较,可以看出该易解子类可以获得更小的隐蔽集,从而有效地将更多的变量分离到易解部分.

1 量词约束满足问题的定义

约束满足问题包括有限变量集合和有限约束集合,每个变量对应一个有限论域,每条约束限制了一些变量允许赋值的组合.约束满足问题的解是为每个变量赋一个对应论域上的值,使之满足所有约束.求解约束满足问题的目标是找到一个解或是全部解.事实上,量词约束满足问题是通过经典约束满足问题对每一个变量都限定一个存在(∃)或者全称(∀)量词扩展而来的.同时,量词约束满足问题也是量词布尔范式的扩展,量词布尔范式中,每个变量都在布尔域上取值,即取“真”或者“假”,而量词约束满足问题的变量值域可以是任意的有限集合.因为量词布尔范式是 PSPACE-完全类中的原型问题,所以量词约束满足问题也是 PSPACE-完全的.

量词约束满足问题可以定义成一个四元组的形式(X,D,C,Q)[2],其中,

- X 是一个有序的变量集合$\{x_1,x_2,\dots,x_n\}$;
- $D=\{D_1,D_2,\dots,D_n\}$代表了对应变量的取值范围,函数 $D(x_i)$表示变量 x_i 的值域$(1\leqslant i\leqslant n)$;
- $C=\{c_1,c_2,\dots,c_e\}$是一个约束集合,每个约束限制了若干变量允许取值的关系元组;
- $Q=\{\lambda_1x_1,\lambda_2x_2,\dots,\lambda_nx_n\}$是一个有序的量词集合,$\lambda_i\in\{\exists,\forall\}(1\leqslant i\leqslant n)$.

函数 $Var(c_i)(1\leqslant i\leqslant e)$定义为参与约束 c_i 的变量集合.

量词约束满足问题 $P=(X,Q,D,C)$表示逻辑公式$\lambda_1x_1\lambda_2x_2\dots\lambda_nx_n(C)$,所以变量在 P 中是有序排列的,变量 x_i 在 x_j 之前是指 x_i 在前缀中出现在 x_j 之前,记作 $x_i\prec x_j$ 或 $i<j$.不同于约束满足问题的解,量词约束满足问题的解是树状结构,称为策略树(strategy).给定一个量词约束满足问题,其策略树 T 是包含了根节点和 n 层带标签节点的树,树的第 0 层为根节点,第 i 层的节点对应前缀中第 i 个变量.如果第 $i+1$ 层是全称量词变量,则第 i 层每个节点有$|D(x_i)|+1$ 个子节点;否则,每个节点有 1 个子节点.除根节点外,每个节点都包含一个标签,表示变量的赋值,含有全称量词变量的一组赋值称为全称量词分支.而方案(scenario)是一组包含全部变量的赋值,它对应策略树一条从根节点到叶节点的路径,如果策略树中全部的方案都是相容的,则该策略树是量词约束满足问题的一个解.部分策略树是只包含某一策略树 T 的第 0 层~第 i 层全部节点的树,记作 T_i.

给定一个量词约束满足问题,如果对于所有的约束 $c_i(1\leqslant i\leqslant e)$都有$|Var(c_i)|=2$,则该问题是二元的;如果存在约束 c_i 使得$|Var(c_i)|>2$,则说该问题是非二元的.本文主要探讨二元量词约束满足问题,因此,c_{ij} 被用来表示变量 x_i 与 x_j 之间的约束.这里包含了平凡约束,即在此约束里,所有的关系元组都是允许的.函数 $R(x_i,x_j)$表示约束 c_{ij} 中允许取值的二元组集合.$X^{\forall}$表示全部全称量词变量集合,$X^{\exists}$表示全部存在量词变量集合.z 为变量,$X^{\forall}(z)$表示在 z 之前的全称量词变量集合.

值 a 被赋给变量 y 时,称 a 为 y 的赋值,记作 $y=a$;一组赋值$\alpha=a_1a_2...a_k$ 被赋给变量集合 $Y\subseteq X$ 时,称α为 Y 的一组赋值,记作 $Y=\alpha$.若α和β分别是一组赋值且α和β所包含的变量交集为空,则$\alpha\beta$表示两组赋值的并集.在本文中,a,b 和 c 通常用于表示单个变量的赋值,而α和β表示对一组变量的赋值.若 $y=a$ 是一个赋值且 z 是变量,则 z 中与 a 相容的所有值组成了 a 关于 z 的支持集,记作 $I_z(a)$;若α是 Y 的一组赋值,则变量 z 中所有与α相容的值的集合记作 $I_z(\alpha)$.支持集集合$\Pi_y(\forall)=\{I_y(\alpha)$,其中,$\alpha$为任意一组 $X^{\forall}(y)$的赋值$\}$.若β为一组对存在量词变量的赋值,$\Pi_y(\forall,\beta)=\{I_y(\alpha\beta)|$其中,$\alpha$为任意一组 $X^{\forall}(z)$的赋值$\}$.

若α为一组赋值,则 $P(\alpha)$表示通过α赋值对 P 化简后的问题,即从 P 中删除被赋值的变量和这些变量参与的约束,同时删除未赋值变量中与α不相容的取值并在未赋值变量的约束中删除不相容的取值参与的元组关系后,得到的 P 的化简问题.

例 1:二元量词约束满足问题 P 的约束关系图如图 1 所示,P 包含了 4 个变量,其前缀顺序为$\forall x\exists y\exists z\exists t$,$x$ 和 y 的值域为$\{1,2,3\}$,而 z 和 t 的值域为$\{1,2,3,4\}$,图中变量用节点表示,两个节点间的边表示两个变量间存在约束关系.

例 1 的一个解(相容策略树)可用树状形式表示,如图 2 所示.由于 x 是全称量词变量,其上一层节点具有多个子节点,而变量 y,z 和 t 为存在量词变量,所以上一层的节点只具有一个子节点.

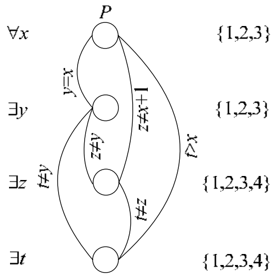


Fig.1 Constraint graph of the QCSP in Example 1

图 1 例 1 中量词约束满足的问题的约束图结构

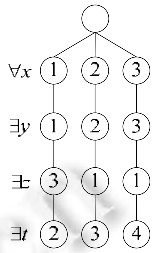


Fig.2 A solution tree of Example 1

图 2 例 1 的一个解树

2 全称量词相容性

首先介绍基本的量词相容性概念,而后提出一个新的相容性概念——全称量词相容性.并讨论了全称量词相容性与量词相容性的关系以及计算复杂度.

2.1 量词相容性

量词相容性是从约束满足问题的相容性扩展而来的,它的提出主要是为了化简问题的值域从而加速回溯算法的求解[25],这些量词相容性也被用来识别混合易解子类.给定一个量词约束满足问题,其解树的构造顺序与其量词前缀的变量顺序相关,因此,这里主要关注有向的量词相容性.

定义 1(有向的量词弧相容). 给定一个二元量词约束满足问题 $P=(X,Q,D,C)$,P 是有向量词弧相容的,如果对于每个有序的变量对(x_i,x_j)使得 $x_i\prec x_j$,满足以下条件之一.

- $\exists x_i\exists x_j$:对于每个值 $a\in D(x_i)$,存在一个值 $b\in D(x_j)$,使得$(a,b)\in R(x_i,x_j)$;
- $\forall x_i\forall x_j$:对于每个值 $a\in D(x_i)$,任意的值 $b\in D(x_j)$满足$(a,b)\in R(x_i,x_j)$;
- $\forall x_i\exists x_j$:对于每个值 $a\in D(x_i)$,存在一个值 $b\in D(x_j)$,使得$(a,b)\in R(x_i,x_j)$;
- $\exists x_i\forall x_j$:对于每个值 $a\in D(x_i)$,任意的值 $b\in D(x_j)$满足$(a,b)\in R(x_i,x_j)$.

给定一个有序的变量对(x_i,x_j)使得 $x_i\prec x_j$,如果(x_i,x_j)满足上述 4 个条件之一,则(x_i,x_j)是有向量词弧相容的.下面定义有向量词 k 相容:

定义 2(有向量词 k 相容). 给定一个二元量词约束满足问题 $P=(X,Q,D,C)$和一个自然数 k,使得$(k\leqslant n)$.如果对于每个有序的变量集合$(y_1,\ldots,y_{k-1},x_m)$,使得 $y_i\prec y_{i+1}$ 且 $y_i\prec x_m(1\leqslant i\leqslant k-1)$满足以下条件之一,则 P 是有向量词 k 相容的.

- $\lambda_1y_1,\lambda_2y_2,\ldots,\lambda y_{k-1}\exists x_m$对于任意$y_1y_2\ldots y_{k-1}$的相容赋值$(a_1,a_2,\ldots,a_{k-1})$,存在一个值$b\in D(x_m)$,使得$(a_i,b)\in R(y_i,x_m)(1\leqslant i\leqslant k-1)$;
- $\lambda_1y_1,\lambda_2y_2,\ldots,\lambda y_{k-1}\forall x_m$ 对于任意 $y_1y_2\ldots y_{k-1}$ 的相容赋值$(a_1,a_2,\ldots,a_{k-1})$,任意一个值 $b\in D(x_m)$,使得$(a_i,b)\in R(y_i,x_m)(1\leqslant i\leqslant k-1)$.

当 k 为 1 时,相容性为量词节点相容;当 k 为 2 时,相容性为有向量词弧相容.给定一个二元量词约束满足问题 P,如果对于每个自然数 $i\leqslant k$,P 是有向量词 i 相容的,则称 P 是强有向量词 k 相容的;如果 P 是强有向量词 n 相容的,则 P 是有向全局相容的.其中,n 为变量的数量.如果 P 是有向全局相容的,则 P 是可满足的,P 的一个解可以按量词前缀的变量顺序通过无回溯地扩展策略树而得到.

2.2 全称量词相容性

为了更好地揭示问题内部隐藏的易解属性,本节深入研究量词约束满足问题的结构性质.不同于经典的约束满足问题,量词约束满足问题的结构特征除了变量间的约束,还决定于量词变量在前缀中的顺序.因此,这里将提出一个新的相容性概念,即全称量词变量相容性,它用于刻画一组存在量词变量和所有的全称量词变量的

约束关系.类似于量词相容性,全称量词变量相容性也由一系列的定义组成.

定义 3($\lambda\forall$相容). 给定一个二元量词约束满足问题 $P=(X,Q,D,C)$,如果对于每个有序的变量对$(\lambda x_i,\forall x_j)$使得$x_i \prec x_j$,且对于每个值 $a\in D(x_i)$,任意的值 $b\in D(x_j)$满足$(a,b)\in R(x_i,x_j)$,其中,λ是$\exists$或者$\forall$,则称 P 是$\lambda\forall$相容的.

$\lambda\forall$相容保证了 P 是非平凡的,若 P 不满足该相容性,则全称量词变量存在不可被满足的赋值,所以该问题平凡无解.

定义 4($\forall$量词节点相容). 给定一个二元量词约束满足问题 $P=(X,Q,D,C)$,如果 P 是$\lambda\forall$相容的,且对于任意存在量词变量 x_i,对于任意的一组 $X^{\forall}(x_i)$的赋值α,存在一个 x_i 的取值 $a\in D(x_i)$,使得α与 a 相容,则称 P 是$\forall$量词节点相容的.

从定义 4 可以看出,如果 P 是$\forall$量词节点相容的,则 P 中每一变量对$(\forall x_i,\forall x_j)(x_i \prec x_j)$是有向量词弧相容的.类似地,$(\forall x_i,\exists x_j)$和$(\exists x_i,\forall x_j)$也是有向量词弧相容的.但是这并不意味着 P 是有向量词弧相容,原因在于不能保证 P 中的$(\exists x_i,\exists x_j)$也是有向量词弧相容的.事实上,将 P 化简成$\forall$量词节点相容的形式可以删除不支持任何一个量词分支的值.

接下来,定义有向$\forall$量词 k 相容.

定义 5(有向$\forall$量词 k 相容). 给定一个二元量词约束满足问题 $P=(X,Q,D,C)$,如果 P 是$\lambda\forall$相容的,且对于每个有序的变量集合$\{y_1,y_2,\dots,y_{k-1},x_m\}$,对于任意一个 $X^{\forall}(x_m)\cup\{y_1,y_2,\dots,y_{k-1}\}$的相容赋值$\alpha a_1a_2\dots a_{k-1}$,存在一个 $a_k\in D(x_m)$,使得$\alpha a_1a_2\dots a_{k-1}a_k$是相容的,则称 P 是有向$\forall$量词 k 相容的.

特殊地,有向$\forall$量词 1 相容是$\forall$量词节点相容;有向$\forall$量词 2 相容称为有向$\forall$量词弧相容.类似于量词相容性,强有向$\forall$量词 k 相容是指对于每个 $i\leqslant k$,P 是有向$\forall$量词 i 相容的;如果 P 是强有向$\forall$量词 r 相容的,则它是有向全局$\forall$量词相容的.其中,r 是存在量词变量的个数,即 $r=|X^{\exists}|$.定义 5 保证了 P 中的$(\exists x_i,\exists x_j)$是有向量词弧相容的,所以强有向$\forall$量词弧相容蕴含了有向量词弧相容.

值得注意的是,如果 P 是有向全局量词相容的,那么 P 一定是有向全局$\forall$量词相容的,但是反之不成立.这是因为在满足有向$\forall$量词 k 相容时,可能存在一组存在量词变量的赋值不与任何全称量词分支相容,而对于该赋值,其后续的存在量词变量并不需要存在一个与之相容的值.上述讨论说明,有向全局$\forall$量词相容的限制条件比有向全局量词相容松弛,但这并不影响其保证量词约束满足问题的可满足性.

引理 1. 给定一个二元量词约束满足问题 P,如果 P 是有向全局$\forall$量词相容的,则 P 是可满足的.

证明:首先,由于 P 是$\forall$量词节点相容的,那么第 1 个变量的值域不为空,因此存在一个相容的 1 层策略树(包含根节点).不失一般性,这里证明:对于任意的、相容的 $k-1$ 层相容策略,都可以扩展成一个 k 层策略.如果第 k 层变量 x_k 是一个全称量词变量,由于全称量词节点相容蕴含了包含该变量的变量对是量词弧相容的,则 x_k 的任何值都是相容的,所以扩展出的 k 层策略树是相容的.如果第 k 层变量 x_k 是一个存在量词变量,若前 $k-1$ 层包含 m 个存在量词变量,则 $m+1\leqslant r$(r 为全部存在量词变量的数量).又因为 P 是有向$\forall$量词 $m+1$ 相容的,所以对于部分策略树中的每一个全称量词分支α,都存在一个 x_k 的值 a,使得αa 是相容的,因此扩展出的 k 层策略是相容的.综上,存在一个相容的 n 层策略是 P 的一个解,所以 P 是可满足的. □

在引理 1 中,有向全局$\forall$量词相容是 P 的可满足性的一个充分条件.事实上,如果对于任意的 $k\leqslant n$,前 k 个变量是有向量词相容的,即可保证 P 的可满足性.显然,前 k 个变量的相容性松弛了有向$\forall$量词 k 相容和量词 k 相容的约束限制.

2.3 全称量词相容性算法

本节讨论与有向$\forall$量词相容性相关的算法和计算复杂性,并详细阐述判断其节点相容和弧相容的验证方法.这里首先定义支持集 I 上的交集算子.设$\Pi=\{I_1,I_2,...,I_m\}$为存在量词变量 z 的一个支持集集合,而 I_z 是 z 的一个支持集,交集算子被定义为$\Pi\cap I_z=\{I|I=I_i\cap I_z$,其中,$I_i\in\Pi\}$.该算子在产生的新集合中重复的元素被去除,而新集合中可以存在空集元素,即$\varnothing\in\Pi\cap I_z$.给定一个存在量词变量 z,则全称量词变量集合 $X^{\forall}(z)$关于 z 的支持集集合记为$\Pi_z(\forall)=\{I|$存在一组赋值 $X^{\forall}(z)=\alpha$,使得 I 是α关于 z 的支持集$\}$.

基于上述交集算子,我们提出一个计算给定存在量词变量 x 的全部支持集的算法.该算法采用增量计算的方法,具体过程如算法 1 所示.

算法 1. 全称量词变量支持集集合算法.

输入:$P=(X,Q,D,C)$,z 是一个存在量词变量;

输出:支持集集合$\Pi_z(\forall)$.

1. 令 $X'=X^{\forall}(z)$
2. **if** 存在一个变量 $x_i\in X'$且 $D(x_i)$是空集 **then**
3.   **return** $\Pi_z(\forall)=\varnothing$
4. 令 x_i 为 X'中按前缀顺序的第 1 个变量
5. **foreach** $a\in D(x_i)$ **do**
6. $\Pi[x,a]=\{I_z(a)\}$
7. $X'=X'-\{x_i\}$
8. **while** $X'\neq\varnothing$ **do**
9. $x_j=x_i$
10.   令 x_i 为 X'中按前缀顺序的第 1 个变量
11.   **foreach** $a\in D(x_i)$
12. $\Pi[x_i,a]=\bigcup(I_z(b)\cap I_z(a)), \forall b\in D(x_j)$
13. $X'=X'-\{x_i\}$
14. $\Pi_z(\forall)=\bigcup(\Pi[x_j,b]), \forall b\in D(x_j)$
15. **return** $\Pi_z(\forall)$

算法 1 输入的问题 P 需要满足$\lambda\forall$相容性,而算法过程中不再判断 2 个全称量词变量之间的约束关系.算法 1 进行迭代求解,最后返回 z 的全称量词变量支持集集合.如果该集合中包含$\varnothing$元素,则表明存在至少一个全称量词分支,使得 z 中不存在一个赋值与该路径相容.因此,输入问题 P 不是有向$\forall$量词节点相容的.

在算法 1 计算出的集合中,支持集个数的一个上界是 $M^{n_\forall}$,但因为重复的元素被去除,支持集数量的另一个上界是 2^M,因此上界是 $\min\{M^{n_\forall},2^M\}$,其中,$M=|\max(D(x_i))|$,$n_\forall=|X^\forall|$.所以,算法 1 在通常情况下的时间复杂度是指数级的.但是该算法的时间只随着全称量词变量的个数 $n_\forall$呈指数级增长,这不同于问题的求解,其算法的时间复杂度与所有变量的个数 n 相关.因此,如果 $n_\forall$不与 n 相关,则算法 1 的时间并不随问题的规模指数级增长.如果 $n_\forall$是常数,则可以认为算法 1 具有多项式时间的复杂度.

接下来讨论判断一个二元量词约束满足问题有向$\forall$量词节点相容性和有向$\forall$量词弧相容性的方法.

a) 对于$\forall$量词节点相容性而言,我们首先需要判断全称量词变量间的相容性,这个过程可以直接判断两个全称量词变量间是否存在约束关系:如果存在且这个约束关系是非平凡的,则可以判定 P 不是$\forall$量词节点相容的;否则,根据$\forall$量词节点相容性的定义,对于每个存在量词变量 x,我们都需要判断每一个全称量词分支是否存在一个 x 的赋值与之相容,而全称量词的取值组合是随全称量词变量个数指数级增长的,逐个分支判断会导致计算时间是指数级的.但是还可以通过计算 x 的支持集集合完成这一判断过程.如果支持集集合中包含了$\varnothing$,则存在一个全称量词分支不能扩展到变量 x,因此 P 不是$\forall$量词节点相容的;否则,P 是$\forall$量词节点相容的.

b) 对于有向$\forall$量词弧相容性而言,我们可以使用与$\forall$量词节点相容性同样的方式进行判断.对于每一个存在量词变量对(y,z),我们可以计算 $I_z(a)$的值,计算过程可以通过改造算法 1 得到.其中,在第 11 行的对于每个 x_j 的值,首选判断该值是否与 a 相容:如果相容,则进行交集的计算;否则,尝试下一个值.在计算集合 $I_z(a)$后,我们只需考察集合中是否包含$\varnothing$元素:如果包含,则 P 不是有向$\forall$量词弧相容的.值得注意的是,支持集的集合 $I_z(a)$本身也可能是空集,这是由于没有与 a 相容的全称量词分支造成的,所以它不是判断不满足有向$\forall$量词弧相容性的依据.若 $n_\forall$是常数,则有向$\forall$量词弧相容性的判断过程也是多项式时间的.

显然,判断$\forall$量词节点相容性和有向$\forall$量词弧相容性的时间复杂度主要决定于支持集的数量.设 L 为全部存

在量词变量支持集的数量,则不难推出判断有向∀量词节点相容性的时间复杂度为 $O(LMn)$,而判断有向∀量词弧相容性的时间复杂度为 $O(LM^2n^2)$.

但在一些特定情况下,支持集的计算过程可以是多项式时间的.这里首先简要介绍 BTP(broken-triangle property),它最早是由 Cooper 等人提出的[24,26],并用来确定约束满足问题中的易解子类.如果一个约束满足问题中的 3 个变量组成一个有序三元组(x,y,z),如果对于任意一个值对$(a,b)\in R(x,y)$使得 a 和 b 关于 z 的支持集满足 $I_z(a)\subseteq I_z(b)$或者 $I_z(a)\supseteq I_z(b)$,则称(x,y,z)满足 BTP.在一个二元量词约束满足问题中,若给定一个存在量词变量 x_k,在 x_k 之前的任意两个全称量词变量 x 与 y 有三元组(x,y,z)满足 BTP,算法 1 中的支持集交集运算并不会产生新的支持集,因此计算过程中,$|\Pi[x_i,a]|$的元素个数的一个上界是 $d|X^{\forall}|$,所以$\Pi_z(\forall)$可以在多项式时间内计算.

给定一个二元量词约束满足问题 P,如果对于每个存在量词变量 z,计算其全称量词关于 z 的支持集集合都是多项式时间的,则称 P 满足全称量词支持集易解属性.基于这一属性,在下一节讨论量词约束满足问题的易解子类.

3 易解子类

近年来,量词约束满足问题的混合易解子类得到了深入的研究.现有易解子类是基于经典约束满足问题的 BTP 属性扩展得到的.本节将讨论如何通过全称量词相容性扩展已有的混合易解子类.这里基于 BTP,说明全称量词相容性在判断易解子类中的作用.

首先简要说明已知的 BTP 易解子类.给定一个二元约束满足问题,如果存在一个变量的全序,使得任意 3 个变量在该序下满足 BTP,则可以在多项式时间内判断约束满足问题是否是可满足的.这里只需要将给定的约束满足问题化简为弧相容的形式,如果化简结果不出现值域为空集的变量,则该问题是可满足的.BTP 还被扩展到软约束问题中[27],而后被用来识别量词约束满足问题中的易解子类[28].给定一个量词约束满足问题 P,P 在序$\prec$下满足 BTP 是指对于 P 中任意有序三元组(x,y,z)使得 $x\prec y\prec z$,(x,y,z)满足 BTP.由于量词约束满足问题的变量需要按其前缀中的顺序赋值,所以只需判断在该变量序下,量词约束满足问题是否满足 BTP 即可.定理 1 确定了基于 BTP 的量词约束满足问题易解子类.

定理 1[28]. 给定一个二元量词约束满足问题 $P=(X,Q,D,C)$,如果 P 是有向量词弧相容的,且 P 在其前缀中的变量序下满足 BTP,则 P 是可满足的.

同时,确定一个易解子类不仅需要证明其中包含的量词约束满足问题可在多项式时间内求解,还需证明给定的一个二元量词约束满足问题判别其是否属于该易解子类也是多项式时间的.不难看出,判断 P 是否为有向量词弧相容的并且是否满足 BTP 是多项式时间的.

接下来,基于全称量词变量的支持集集合本文提出一个新的混合易解子类,这里首先定义基于支持集集合的 BTP 性质.

给定一个二元量词约束满足问题 $P=(X,Q,D,C)$,x,y 和 z 是 3 个存在量词变量,$\Pi_z(\forall)$是全称量词变量关于 z 的支持集集合,如果对于任意值对$(a,b)\in R(x,y)$和任意的$I\in\Pi_z(\forall)$,满足$I_z(a)\cap I\subseteq I_z(b)\cap I$或者$I_z(a)\cap I\supseteq I_z(b)\cap I$,则称三元组$(x,y,z)$满足 uBTP.

定义 6(uBTP). 给定一个二元量词约束满足问题 $P=(X,Q,D,C)$,P 中任意存在量词变量组成的三元组(x,y,z)使得 $x\prec y\prec z$,满足 uBTP,则称 P 满足 uBTP.

例 2(uBTP 的实例):若 P 是一个二元量词约束满足问题,x,y,z 是其中的 3 个存在量词变量.若 z 的支持集集合 $I=\{\{1,2\},\{2,3\},\{1,2,4\}\}$并且 $I_z(x=1)=\{1,2,3\}$,$I_z(y=1)=\{1,2,4\}$,则(x,y,z)满足 uBTP 属性;若 $I_z(x=2)=\{1,2\}$并且 $I_z(y=2)=\{2,4\}$,则由于 $I_z(x=2)$和 $I_z(y=2)$分别与 I 中第 1 个元素取交集后的两个集合不存在包含关系,所以$(x=2,y=2,z)$不满足 uBTP 属性.

定理 2. 给定一个二元量词约束满足问题 $P=(X,Q,D,C)$,如果 P 是强有向∀量词弧相容的,并且 P 满足 uBTP,则 P 是可满足的.

证明:这里只需证明 P 是有向∀量词全局相容的,因此采用归纳法证明:对于任意的 $k>2$,如果 P 是有向∀量词

k–1 相容的,那么它是有向∀量词 k 相容的.不妨设 P 是有向∀量词 k–1 相容的,β是一个含有 k–1 个存在量词变量的相容赋值,z 是这 k–1 个变量后的一个存在变量.考虑任意一个包含 $X^{\forall}(z)$的赋值α使得$\alpha\beta$是相容的,因为 P 满足 uBTP,则对于β中任意的两个赋值 $x=a,y=b$,有 $I_z(\alpha a)\subseteq I_z(\alpha b)$或者 $I_z(\alpha a)\supseteq I_z(\alpha b)$.因此在所有的支持集 $I_z(f)$中,f 为β中所包含的任意一个赋值,存在一个最小的支持集被所有其他支持集包含,即 $I_z(c)\subseteq I_z(f)$,其中,c 是β中的一个赋值.所以存在一个 z 的取值 c,使得$\alpha\beta c$ 是相容的.因此可以得到结论,P 是有向∀量词 k 相容的.又因为 P 是有向∀量词节点相容的和有向∀量词弧相容的,所以 P 是有向∀量词全局相容的.根据引理 1 知,P 是可满足的. □

定理 3. 给定一个二元量词约束满足问题 $P=(X,Q,D,C)$,如果 P 满足∀量词支持集易解属性,判断 P 是否满足 uBTP 可以在多项式时间内完成.

证明:对于任意的由存在量词变量组成的三元组(x,y,z)使得 $x\prec y\prec z$,如果对于任意的值对$(a,b)\in R(x,y)$和每个在非空集合$\Pi_z(\forall,ab)$中的支持集 $I,I\cap I_z(a)$和 $I\cap I_z(b)$存在包含关系,则对于任意的 $X^{\forall}(z)$的赋值α使得 $I_z(\alpha)=I$,有 $I_z(\alpha a)\subseteq I_z(\alpha b)$或者 $I_z(\alpha a)\supseteq I_z(\alpha b)$,因此 P 满足 uBTP.又因为任意的$\Pi_z(\forall,ab)$是多项式时间可以获得的,整个判断过程是多项式时间的.因此,判断 P 是否满足 uBTP 是多项式时间的. □

这里需要解释的是:若集合$\Pi_z(\forall,ab)$为空,则表示不存在一个α使得α与 a,b 相容,则此种情况不需要进行集合包含的判断.

uBTP 是对 BTP 概念的扩展,因此使用 uBTP 识别的易解子类包含了文献[28]中用 BTP 识别的易解子类,这里将举例说明满足 uBTP 的量词约束满足问题.

回顾例 1 中的量词约束满足问题,显然,该问题不满足 BTP.原因是三元组(y,z,t)之间的不等式约束不满足 BTP,但是不难验证 P 满足 uBTP,因此可以直接构造 P 的一个解树而不需要回溯搜索.而 uBTP 所确定的易解子类是一个更一般化的易解子类形式,它可以囊括更多的易解问题实例.

命题 1(uBTP 的值约减属性). 给定一个二元量词约束满足问题 $P=(X,Q,D,C)$使得 P 是强有向∀量词弧相容的,并且 P 满足 uBTP,如果α是前 k 个变量的一组相容赋值,则 $P(\alpha)$是强有向∀量词弧相容的且 $P(\alpha)$满足 uBTP.

证明:先证强有向∀量词弧相容,令 z 为 $P(\alpha)=(X',Q',D',C')$中的一个存在量词变量,β为 $P(\alpha)$中 $X'^{\forall}(z)$的一组相容赋值,因为 P 满足 uBTP,则 $P(\alpha)$是可满足的,所以 $P(\alpha)$中 $I_z(\beta)$不为空,所以 $P(\alpha)$是∀量词节点相容的.若 y 为 z 后的一个存在量词变量,a 是 z 在 $P(\alpha)$中一个取值且βa 相容,则由于 P 是有向∀量词弧相容的,所以在 P 中存在一个 y 的取值 b,使得$\alpha\beta ab$ 相容.由于 b 与α是相容的,则在 $P(\alpha)$中,b 仍是 y 的一个取值,因此 $P(\alpha)$是有向∀量词弧相容的.再证 $P(\alpha)$满足 uBTP,由于 $P(\alpha)$中的变量的值域被约减后并不影响支持集之间的包含关系,所以 $P(\alpha)$仍满足 uBTP. □

uBTP 的值约减属性说明了变量的赋值过程保持了 uBTP 及易解性质,为隐蔽集的识别方法的设计提供了理论依据.

4 隐蔽集的识别

易解子类可以将一部分带有特殊约束结构的问题囊括其中,从而证明这些特殊问题是多项式时间可解的.但由于量词约束满足问题本身的计算复杂性,能够被直接证明属于某个已知易解子类的特殊问题是有限的,大量的量词约束满足问题仍属于 PSPACE 完全的范畴.然而通常情况下,可以在这些问题中识别出一些子结构或子问题属于某个已知的易解子类.为了分析问题中所隐藏的易解结构并分离出易解部分,同时衡量不同易解子类在识别隐藏结构的效果,隐蔽集的概念被提出.它是由 Williams 等人提出的[18],用于研究 SAT 的易解子结构,并用于分析 SAT 求解算法的性能.而后,隐蔽集的概念被扩展到量词布尔范式中[19].隐蔽集是指一个变量的集合,存在一组该变量集合的赋值使得剩余的子问题是可满足的且在多项式时间内可以判定.通常情况下,隐蔽集的研究需基于某一种已知的易解子类.对于 SAT 而言,易解子类可以是 2-SAT(子句长度不超过 2 的合取范式)或者 Horn 子句等.隐蔽集的长度影响着求解算法的效率,小隐蔽集意味着求解算法尝试较少的变量赋值即可找到问题的解.

根据 SAT 和量词布尔范式中的隐蔽集的定义,本文给出量词约束满足问题的隐蔽集定义.

定义 7(隐蔽集). 给定一个量词约束满足问题 P,若存在一个部分策略树 T_i,使得对于该树中任何分支对应的赋值α,存在一个多项式时间的算法返回 $P(\alpha)$的一个解,则称 P 的前 i 个变量是 P 的一个隐蔽集.

若已知一个量词约束满足问题的隐蔽集,求解该问题则简化为寻找一个满足隐蔽集定义的部分策略树 T_i,其计算复杂度与 i 值相关,而问题的变量数量只影响计算复杂度的多项式部分.因此,隐蔽集的存在降低了求解算法的计算时间.

显然,本文提出的 uBTP 易解子类和已知的 BTP 易解子类可以用于识别隐蔽集.为了研究本文提出的混合易解子类对隐蔽集的影响,我们将分析使用不同易解子类时隐蔽集的大小,即隐蔽集中变量的个数.

本文基于量词约束满足问题求解器 QCSP-Solve[5]分析隐蔽集的大小.QCSP-Solve 是一个基于回溯算法的完备的量词约束满足问题求解器,它可以求解二元问题.同时,QCSP-Solve 中还加入了诸多在 CSP 求解中使用较为成熟的技术.例如,基于冲突的回跳技术[29]在产生冲突赋值时,会分析失败的原因,并回跳至产生冲突变量对应的层.这种技术不同于一次只回溯 1 层的普通回溯算法,它可以大幅度减少不必要的搜索空间.在预处理过程中,QCSP-Solve 使用了量词弧相容算法 QAC-2001[30]和对称消除策略等技术,用于加速求解器的效率.

在实验分析中,本文改造了 QCSP-Solve 中的算法,在回溯搜索的基础上加入了易解子问题的识别过程.在展开每个节点并对变量赋值后,算法调用识别算法,判断后续未赋值变量间是否满足 BTP(uBTP)属性,从而识别出易解子问题.若未赋值变量属于给定的易解子类且是可满足的,则表明当前的量词分支可以构成隐蔽集对应的部分策略树的一个分支,因此算法回溯并尝试下一个量词分支.否则继续尝试下一个未赋值的变量.当算法找到问题一个解树时,根据命题 1 的结论,uBTP 易解子类中的一个问题实例 P 在对前 i 个变量赋值为α后,其剩余的子问题 $P(\alpha)$仍然满足有向$\forall$量词弧相容和 uBTP.所以在回溯搜索过程中,具有最多变量赋值的量词分支的长度即为所找到的隐蔽集的大小.但是找出某个量词约束满足问题的全部隐蔽集需要遍历整个解空间,这对于 PSPACE 完全类中的问题是十分困难的,而计算最小的隐蔽集也是困难的.因此,本文只分析找到第 1 个解时隐蔽集的大小.

5 实验分析

本节通过实验分析隐蔽集大小.分析隐蔽集大小的实验采用一系列随机生成的量词约束满足问题作为测试实例.所有实验采用一台工作站进行,其中,CPU 为 E5-1650v4,内存为 16GB,Windows Server 2016 作为操作系统.实验算法采用 C++编写,并将检测隐蔽集的代码与 QCSP-Solve 算法结合,实验系统采用 Visual Studio 2010 编译.本节首先介绍量词约束满足问题的生成模型,然后分析实验结果,比较不同约束密度的情况下,两个混合易解子类的隐蔽集大小,以及 QCSP-Solve 在回溯搜索过程中找到易解部分时未赋值变量个数平均值.

5.1 生成模型

在衡量量词约束满足问题的算法效率时,随机生成的问题实例通常被作为测试基准.使用最多的随机生成模型是 Model B[31].该模型最早用于生成约束满足问题,其改进的版本具备了精确的可满足性相变点,称其为 Model RB[32].而后,量词约束满足问题也使用此类模型作为测试基准.

在约束满足问题的 Model B 中,4 个参数被用来控制生成问题的规模和难度,所以通常用四元组(n,d,p,q)表示,其中,n 为变量的个数,d 为每个变量值域的大小,p 为约束的密度参数,它控制问题中约束的数量.

模型从 $n(n-1)/2$ 个可能的二元约束中均匀地随机地选取$|pn(n-1)/2|$个约束,q 为约束的松紧度参数.对于每个二元约束模型,将均匀地随机地选取 qd^2 个值对设为允许的关系元组.对于量词约束满足问题,还需要额外的3 个控制参数,$n_\forall$指定了全称量词变量的个数,$q_{\forall\exists}$和 $q_{\exists\exists}$被用来取代 q,$q_{\forall\exists}$表示一个全称量词变量和一个存在量词变量之间的约束松紧度,$q_{\exists\exists}$表示两个存在量词变量之间的约束松紧度.

另外,全称量词变量在前缀中的排列位置也会影响量词约束满足问题的约束结构.在本文的实验中,两种排列方式将被采用.第 1 组中,$n_\forall$个全称量词变量的位置是随机选择的;第 2 组中,全称量词变量和存在量词变量被划分到变量块中,两种变量块交替排列.

但是研究指出,量词约束满足问题的生成模型存在平凡不可满足的缺陷(flaw issue)[31].如果 $n_\forall$足够大且 $q_{\forall\exists}$

过小,则生成的问题中会存在某个全称量词分支,使某个存在量词变量中没有相容的赋值,这样导致了所生成的问题是平凡无解的.为了解决这一缺陷,Gent 等人提出了基于双映射的结构化约束关系元组生成方案,代替完全随机的生成模型.但是结构化的生成方案破坏了随机均匀地值对选取规则,而且这种方案使得 90%以上的值对都是允许的关系元组,即 $q_{\forall\exists}$大于 0.9.事实上,如果控制 $q_{\forall\exists}$在 0.9 以上,平凡不可满足的缺陷发生概率是极低的.因此在本节的实验中,$q_{\forall\exists}$的取值为 0.95.

5.2 实验结果

本节实验对比 BTP 易解子类和 uBTP 易解子类确定量词约束满足问题中的隐蔽集大小的效果.

首先分析第 1 组实验.在生成实例时,变量的顺序随机排列.这里取 n=20,d=10,而 $n_\forall$的取值分为两种情况,$n_\forall$=6 和 $n_\forall$=8.约束密度参数 p 的值从 0.2 递增到 0.5,增量为 0.025.当 p=0.2 时,约束关系图的密度较为疏松,所生成的问题实例较为简单;而 p=0.5 时,约束图密度增大,约束增多,问题实例的求解难度也随之增加.对于每个 p 值,$q_{\exists\exists}$的值从 0.7 递增到 0.9,增量为 0.05,共 5 个取值,以生成不同难度的约束关系.$q_{\exists\exists}$值越小,相容的赋值就越小.而当 $q_{\exists\exists}$过小时,生成的实例可能是不可满足的,因此 $q_{\exists\exists}$的值最小为 0.7,以保证部分生成的实例是可满足的.对于每组 p 和 $q_{\exists\exists}$的取值,生成 20 个实例,并计算它们的隐蔽集大小.

表 1 列出了 $n_\forall$=6 时的实验结果,表 2 列出了 $n_\forall$=8 的结果.

Table 1 Comparison of the backdoor size detected by BTP and uBTP ($n_\forall$=6)

表 1 BTP 与 uBTP 的隐蔽集大小的对比结果($n_\forall$=6)

p 值	可满足实例数	隐蔽集大小		平均搜索深度	
		BTP	uBTP	BTP	uBTP
0.200	100	15.60	13.20	15.40	12.68
0.225	100	16.21	13.91	15.91	13.37
0.250	100	16.41	14.49	16.02	13.72
0.275	100	16.79	15.51	16.33	14.56
0.300	99	16.78	15.83	16.02	14.81
0.325	100	16.50	15.50	15.80	14.53
0.350	98	16.71	15.89	15.83	14.73
0.375	95	16.66	15.87	15.79	14.79
0.400	90	16.52	16.06	15.44	14.63
0.425	86	16.65	16.01	15.59	14.85
0.450	82	16.80	16.26	15.73	14.90
0.475	73	17.03	16.70	15.74	15.00
0.500	65	17.18	16.83	16.13	15.37

Table 2 Comparison of the backdoor size detected by BTP and uBTP ($n_\forall$=8)

表 2 BTP 与 uBTP 的隐蔽集大小的对比结果($n_\forall$=8)

p 值	可满足实例数	隐蔽集大小		平均搜索深度	
		BTP	uBTP	BTP	uBTP
0.200	99	16.59	14.02	16.35	13.16
0.225	99	16.85	14.30	16.33	13.51
0.250	93	16.83	15.03	16.27	14.20
0.275	94	17.00	15.67	16.28	14.51
0.300	83	17.02	16.04	16.42	14.83
0.325	75	17.07	16.33	16.42	14.95
0.350	69	17.16	16.55	16.29	15.12
0.375	54	17.39	17.02	16.38	15.13
0.400	37	17.54	17.00	16.39	15.03
0.425	43	17.60	17.23	16.71	15.48
0.450	37	17.49	17.14	16.65	15.38
0.475	26	17.58	17.19	16.56	15.42
0.500	19	17.53	17.11	16.68	15.77

对于每个 p 值,计算 $q_{\exists\exists}$的 5 个取值的 100 个实例求解到的隐蔽集大小的平均值,同时还统计了 QCSP-Solve 在回溯搜索过程中找到易解部分时未赋值变量个数平均值.从表 1 和表 2 中可以看出,随着 p 值的增大,问题实例的难度也随之增加.由于约束数量增多,在回溯过程中赋值后,剩余的变量构成一个易解问题的可能性变小,

所以求解到的隐蔽集的规模随 p 值增加而增大.而对于所有 p 值,使用 uBTP 作为易解子类的隐蔽集规模比 BTP 小,这一结果与 uBTP 易解子类包含 BTP 易解子类的事实相符,所以 uBTP 应识别到更小的隐蔽集.而对于简单问题(p 值较小时),规模差别更大一些,uBTP 的隐蔽集更具有优势.而对于每个 p 值平均搜索深度略小于隐蔽集大小,uBTP 的平均搜索深度也小于使用 BTP 时的平均值.这意味着 uBTP 减小了回溯搜索过程所展开的节点数量.随着 $n_{\forall}$的增加问题变难,所以隐蔽集规模略有增加.但是对于约束密度较大时,两种易解子类的效果不明显.

接下来分析第 2 组实验.同样取 n=20,d=10.而全称量词变量的排列不再是随机的,而是结构化的.每个全称(存在)量词变量块包含 1 个(2 个)全称(存在)量词变量,全称量词变量块和存在量词变量块交替出现,并设定 $n_{\forall}$=6,前缀中变量的顺序是:2 个存在量词变量,然后 1 个全称量词变量,以此类推.p 的取值范围为 0.2~0.5,$q_{\exists\exists}$的值为 0.7~0.9.对于每个 p,100 个实例的平均值被计算.表 3 列出了该组实验的结果.与前一组结果类似,uBTP 可以识别到更小的隐蔽集,而这种优势在低密度约束问题中体现得更加明显.

Table 3 Comparison of the backdoor size detected by BTP and uBTP (alternative blocks)

表 3 BTP 与 uBTP 的隐蔽集大小的对比结果(变量块交替)

p 值	可满足实例数	隐蔽集大小		平均搜索深度	
		BTP	uBTP	BTP	uBTP
0.200	100	15.78	14.06	15.53	13.54
0.225	100	16.08	14.67	15.71	13.92
0.250	100	16.31	15.21	15.89	14.42
0.275	100	16.92	16.01	16.30	14.98
0.300	100	16.91	16.33	16.16	15.13
0.325	100	16.80	16.33	15.99	14.97
0.350	100	16.79	16.46	15.71	14.74
0.375	93	17.29	17.11	16.19	15.26
0.400	87	17.10	16.82	15.86	15.09
0.425	80	17.26	17.08	15.90	15.28
0.450	73	17.30	17.19	15.89	15.23
0.475	63	17.63	17.59	16.26	15.65
0.500	58	17.72	17.71	16.37	15.75

6 相关工作

约束满足问题可解子类的研究可以追溯到 20 世纪 80 年代.Freuder 首先指出:如果一个约束满足问题的约束图是树形结构,则存在一个不需要回溯的算法在多项式时间内求解该问题[33].而后,易解子类的研究逐步兴起.一方面,利用约束图结构的特性识别易解子类和分析计算复杂度[34−36];另一方面,研究约束关系的特征也是一个重要的确定可解子类的方法.这方面的研究主要集中在约束语言和约束关系的代数学方法上.例如,Bulatov 等人研究了有限域上的复杂性分类[37].混合易解子类是最近几年提出的一种全新的识别易解子类的技术.它结合了结构子类和关系子类的特点,同时限制约束图的结构和每个约束中的允许的元组关系,从而得到更一般化的易解子类.

一个代表性的混合易解子类是 Cooper 等人提出的 BTP 概念,他们通过一般化树形结构并限制 3 个变量之间的约束关系,得到了约束满足问题上的混合易解子类[24].基于 BTP 的混合易解子类是基于树状约束结构子类更一般化的性质,通过加入约束关系的限制,使得树状结构的特征这一限制条件可以适当地放宽,因此,该易解子类在增加元组关系限制的同时放松了对结构的限制条件,从而得到比树状约束结构子类更一般化的结论.因此,BTP 子类可以囊括更多的易解情况.BTP 定义了 3 个变量间的约束关系,而更多的变量间的关系也被讨论.例如,Cyril 等人提出了 Extendable-Triple Property(ETP)[38],定义了 4 个变量间的性质.BTP 的概念还被扩展到约束优化问题中,与之对应的是 Joint-winner 属性[27].该属性同样定义了 3 个变量之间的约束元组关系,符合该属性的软约束问题是多项式时间内可解的.一个更一般化的结果是 Naanaa 提出的 Rank CSPs[39].该子类利用集合论中秩的概念,计算支持集的集合中,若干支持集取交集的秩.基于 Rank CSPs 的易解子类被证明包含了诸多已知的混合易解子类,例如 BTP、ETP、行凸属性 Row convexity[40]和 m-tight[41]约束属性等.但是 BTP 具有比其他易解子类更多的性质,例如 BTP 在弧相容算法约减后仍满足 BTP.而类似的性质,其他混合易解子类并不具

备,ETP 在路径相容约减运算时(3 个变量间的相容性)下并不是封闭的.Cohen 等人还深入研究了更多的混合可解子类的情况,并提出了二元约束满足问题上的复杂性分类定理[42].

BTP 易解子类以及其他的混合易解子类还用于变量的消元中[43].若存在一个变量全序使得最后的一些变量满足 BTP,则这些变量是可以被消元的.但是给定一个约束满足问题,寻找含有变量数最多的子集使得子集中的变量在某个变量顺序下满足 BTP 属性却是 NP 难的.基于 BTP,更多的用于变量消元的微结构被识别出来,但是这些结构具有十分复杂的元组关系[44].另一方面,基于 BTP 的值合并技术也被提出[45].该技术可以对约束满足问题进行化简,在保证问题可满足性不变的情况下,合并具有相似结构的值,从而化简问题的规模.基于 BTP 值合并技术一个混合易解子类也被提出.

BTP 描述了二元约束之间的关系,为了描述二元以上的约束满足问题,对偶的 BTP 被提出[46],一个多元约束满足问题可以转化成对偶的二元形式,即将约束映射成变量,值域为约束中的元组关系,从而在其对偶的问题上定义 BTP 以确定多元问题的易解结构.另外,BTP 还应用于识别隐含的易解结构,Mouelhi 等人研究了在不同相容性等级下,BTP 易解子类涵盖各类约束满足问题的能力[47],并指出,高阶相容性可以将更多的约束满足问题确定为是多项式时间可解的.

对于量词约束满足问题易解子类的研究,目前主要的思路是扩展经典约束满足问题的易解子类.例如,在 20 世纪 70 年代就证明了量词布尔范式实例中,若子句的长度都是 2 则是多项式时间内可解的[48],基于 Horn 子句的量词布尔范式实例也是多项式时间内可解的[49].但是这种方法却面临很大的挑战,原因在于:全称量词很大程度上改变了约束结构,原有的易解子类在量词约束满足问题中并不一定成立,因此还需要考虑更多的限制条件.通过这样的思路,首先,基于结构特征的易解子类被确定出来,Gottlob 等人研究了全称量词受限情况下基于结构的易解子类[50];其次,Chen 等人深入分析了结构易解子类的特征,还分析了基于约束关系的易解子类的相关性质[51];更进一步地,Börner 等人分析了更多的基于约束关系的可解情况和博弈游戏中多项式时间可解的问题[52];另外,混合方式的结构分析技术也被应用在量词约束满足问题中[28].

易解子类的另一个重要的应用是识别隐蔽集[18].所谓的隐蔽集是约束满足问题中的一个变量子集,并且存在一个多项式时间的算法,使得该算法可以求解剩余的问题.换言之,在对隐蔽集中的变量赋值后,剩下的子问题是多项式时间可解的.例如,环割集事实上就是一个隐蔽集,在去除环割集中的变量后,余下的约束图结构是无环的,这使得余下的子问题可在多项式时间内求解.利用这一性质,李占山等人研究了基于环割集的约束满足问题结构分解技术和相关的约束传播算法[53−55].Kilby 等人[20]通过实验分析得出,问题求解难度和隐蔽集的大小间存在着密切的关系.而 Ruan 等人[21]也研究了隐蔽集对问题结构和求解难度的影响,并指出:当隐蔽集足够小时,存在多项式时间算法可以求解该问题.研究者还根据隐蔽集的概念和相关性质,提高了约束满足问题求解算法的效率.因此,基于何种易解子类,决定了隐蔽集的识别效率和求解算法的效率.诸多易解子类都被用于识别隐蔽集中,例如在 SAT 问题中,主要使用了 2-SAT 和 Horn 子类;Misra 等人讨论了隐蔽集的上界和下界性质[56];Samer 等人讨论了量词布尔范式中的隐蔽集的复杂性[19].基于约束关系的易解子类也被应用于隐蔽集的识别中,而利用混合易解子类的变量消元技术本质上也可以被看作是一个寻找隐蔽集的过程.因此,易解子类的研究为隐蔽集的识别和搜索算法的设计提供了理论基础.

7 总 结

本文深入研究了量词约束满足问题中的约束关系及混合易解子类所需要的条件,并提出一个新的量词约束满足问题的混合易解子类.正如我们所知:由于全称量词变量的引入,量词约束满足问题结构性质不同于其对应的约束满足问题.因此,本文提出的方法将量词约束满足问题易解子类的识别分为两部分:在第 1 部分中,我们分析一个存在变量与全称量词变量的约束关系和相容性关系,并提出全称量词相容的概念,同时讨论了多项式时间的全称量词变量相容性判别的情况;在第 2 部分中,在全称量词相容的情况下,分析存在量词变量之间的约束关系,并在原有的 BTP 混合易解子类基础上一般化其限制条件,提出 uBTP 的概念,从而达到扩展易解子类的目的.同时,本文的易解子类识别方法还可以应用在扩展其他混合易解子类之中,由于 ETP、Row convexity 属性

和Rank CSP与BTP有类似的性质,所以还可以利用本文的方法通过扩展这些性质得到诸多一般化的理论结果,其中的Rank CSP定义在非二元约束上,而该方法对非二元问题中易解子类的识别也是有效的.本文还将混合易解子类应用到识别隐蔽集中,并分析不同易解子类所确定的隐蔽集的大小.通过实验验证了所提出的易解子类在识别隐蔽集方面比原有子类更具优势,同时还可以减少回溯算法中搜索树的平均深度,以达到减小搜索空间的目的.

References:

[1] Dechter R. Constraint Processing. San Francisco: Morgan Kaufmann Publishers, 2003.

[2] Amilhastre J, Fargier H, Marquis P. Consistency restoration and explanationsin dynamic CSPs—Application to configuration. Artificial Intelligence, 2002,135(1):199−234.

[3] Barták R, Salido MA. Constraint satisfaction for planning and scheduling problems. Constraints, 2011,16(3):223−227.

[4] Van Beek P, Chen X. CPlan: A constraint programming approach to planning. In: Hendler J, ed. Proc. of the 16th National Conf. on Artificial Intelligence and 11th Conf. on Innovative Applications of Artificial Intelligence. Orlando: AAAI Press, The MIT Press, 1999. 585−590.

[5] Gent IP, Nightingale P, Rowley A, Stergiou K. Solving quantified constraint satisfaction problems. Artificial Intelligence, 2008, 72(6-7):738−771.

[6] Büning HK, Karpinski M, Flögel A. Resolution for quantified Boolean formulas. Information and Computation, 1995,117(1): 12−18.

[7] Nightingale P. Non-binary quantified CSP: Algorithms and modelling. Constraints, 2009,14(4):539−581.

[8] Palmieri A, Lallouet A. Constraint games revisited. In: Sierra C, ed. Proc. of the 26th Int'l Joint Conf. on Artificial Intelligence. 2017. 729−735.

[9] Lin XH, Feng YX, Tan JR, Feng Y, Gao YC. Product quality characteristics robust optimization design based on quantified constraint satisfaction problem. Chinese Journal of Mechanical Engineering, 2013,49(15):169−179 (in Chinese with English abstract).

[10] Benedetti M, Lallouet A, Vautard J. Modeling adversary scheduling with QCSP+. In: Wainwright RL, ed. Proc. of the 2008 ACM Symp. on Applied Computing (SAC). Fortaleza: ACM Press, 2008. 151−155.

[11] Lallouet A, Lee J, Mak T. Consistencies for Ultra-weak solutions in minimax weighted CSPs using the duality principle. In: Milano M, ed. Proc. of the 18th Int'l Conf. on Principles and Practice of Constraint Programming. LNCS 7514, Québec: Springer-Verlag, 2012. 373−389.

[12] Lallouet A, Lee J, Mak T, Yip J. Ultra-weak solutions and consistency enforcement in minimax weighted constraint satisfaction. Constraints, 2015,20(2):109−154.

[13] Jin JW, Ma FF, Zhang J. Integrating standard dependency schemes in QCSP solvers. Journal of Computer Science and Technology, 2012,27(1):37−41.

[14] Yin MH, Zhou JP, Sun JG, Gu WX. Heuristic survey propagation algorithm for solving QBF proble. Ruan Jian Xue Bao/Journal of Software, 2011,22(7):1538−1550 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3859.htm [doi: 10.3724/SP.J. 1001.2011.03859]

[15] Gent IP, Nightingale P, Stergiou K. QCSP-solve: A solver for quantified constraint satisfaction problems. In: Kaelbling LP, ed. Proc. of the 19th Int'l Joint Conf. on Artificial Intelligence. Edinburgh: Professional Book Center, 2005. 138−143.

[16] Bacchus F, Stergiou K. Solution directed backjumping for QCSP. In: Bessiere C, ed. Proc. of the 13th Int'l Conf. on Principles and Practice of Constraint Programming (CP 2007). LNCS 4741, Providence: Springer-Verlag, 2007. 148−163.

[17] Verger G, Bessière C. A bottom-up approach for solving quantified CSPs. In: Benhamou F, ed. Proc. of the 12th Int'l Conf. on Principles and Practice of Constraint Programming (CP 2006). LNCS 4204, Nantes: Springer-Verlag, 2006. 635−649.

[18] Williams R, Gomes C, Selman B. Backdoors to typical case complexity. In: Gottlob G, eds. Proc. of the 18th Int'l Joint Conf. on Artificial Intelligence. Acapulco: Morgan Kaufmann Publishers, 2003. 1173−1178.

[19] Samer M, Szeider S. Backdoor sets of quantified Boolean formulas. Journal of Automated Reasoning, 2009,42(1):77−97.

[20] Kilby P, Slaney JK, Thiébaux S, Walsh T. Backbones and backdoors in satisfiability. In: Veloso MM, ed. Proc. of the 20th National Conf. on Artificial Intelligence and the 17th Innovative Applications of Artificial Intelligence Conf. Pittsburgh: AAAI Press, The MIT Press, 2005. 1368−1373.

[21] Ruan Y, Kautz HA, Horvitz E. The backdoor key: A path to understanding problem hardness. In: McGuinness DL, ed. Proc. of the 19th National Conf. on Artificial Intelligence, 16th Conf. on Innovative Applications of Artificial Intelligence. San Jose: AAAI Press, The MIT Press, 2004. 124−130.

[22] Carbonnel C, Cooper MC. Emmanuel hebrard: On backdoors to tractable constraint languages. In: O'Sullivan B, ed. Proc. of the 20th Int'l Conf. on Principles and Practice of Constraint Programming. LNCS 8656, Lyon: Springer-Verlag, 2014. 224−239.

[23] Carbonnel C, Cooper MC. Tractability in constraint satisfaction problems: A survey. Constraints, 2016,21(2):115−144.

[24] Cooper MC, Jeavons PG, Salamon Z. Generalizing constraint satisfaction on trees: Hybrid tractability and variable elimination. Artificial Intelligence, 2010,174(9-10):570−584.

[25] Stergiou K. Preprocessing quantified constraint satisfaction problems with value reordering and directional arc and path consistency. Int'l Journal on Artificial Intelligence Tools, 2008,17(2):321−337.

[26] Cooper MC, Jeavons PG, Salamon Z. Hybrid tractable CSPs which generalize tree structure. In: Ghallab M, ed. Proc. of the 18th European Conf. on Artificial Intelligence (ECAI 2008). Patras: IOS Press, 2008. 530−534.

[27] Cooper MC, Zivny S. Hybrid tractability of valued constraint problems. Artificial Intelligence, 2011,175(9-10):1555−1569.

[28] Gao J, Yin M, Zhou J. Hybrid tractable classes of binary quantified constraint satisfaction problems. In: Burgard W, ed. Proc. of the 25th AAAI Conf. on Artificial Intelligence. San Francisco: AAAI Press, 2011.

[29] Bessière C, Régin J. MAC and combined heuristics: Two reasons to forsake FC (and CBJ?) on hard problems. In: Freuder EC, ed. Proc. of the 2nd Int'l Conf. on Principles and Practice of Constraint Programming. LNCS 1118, Cambridge: Springer-Verlag, 1996. 61−75.

[30] Bessière C, Régin J, Yap R, Zhang Y. An optimal coarse-grained arc consistency algorithm. Artificial Intelligence, 2005,165(2): 165−185.

[31] Achlioptas D, Kirousis LM, Kranakis E, Krizanc D, Molloy M, Stamatiou YC. Random constraint satisfaction: A more accurate picture. In: Smolka G, ed. Proc. of the 3rd Int'l Conf. on Principles and Practice of Constraint Programming (CP'97). LNCS 1330, Linz: Springer-Verlag, 1997. 107−120.

[32] Xu K, Li W. Exact phase transitions in random constraint satisfaction problems. Journal of Artificial Intelligence Research, 2000, 12(1):93−103.

[33] Freuder EC. A sufficient condition for backtrack-free search. Journal of the ACM, 1982,29(1):24−32.

[34] Marx D. Tractable hypergraph properties for constraint satisfaction and conjunctive queries. Journal of the ACM, 2013,60(6): Article No.42.

[35] Grohe M. The structure of tractable constraint satisfaction problems. In: Kralovic R, ed. Proc. of the 31st Int'l Symp. on Mathematical Foundations of Computer Science 2006. LNCS 4162, Stará Lesná: Springer-Verlag, 2006. 58−72.

[36] Marx D. Tractable structures for constraint satisfaction with truth tables. Theory of Computing Systems, 2011,48(3):444−464.

[37] Bulatov AA, Jeavons P, Krokhin AA. Classifying the complexity of constraints using finite algebras. SIAM Journal on Computing, 2005,34(3):720−742.

[38] Jegou P, Terrioux C. The extendable-triple property: A new CSP tractable class beyond BTP. In: Bonet B, ed. Proc. of the 29th AAAI Conf. on Artificial Intelligence. Austin: AAAI Press, 2015. 3746−3754.

[39] Naanaa W. Unifying and extending hybrid tractable classes of CSPs. Journal of Experimental and Theoretical Artificial Intelligence, 2013,25(4):407−424.

[40] van Beek P, Dechter R. On the minimality and global consistency of row-convex constraint networks. Journal of the ACM, 1995, 42(3):543−561.

[41] Zhang Y, Yap R. Set intersection and consistency in constraint networks. Journal of Artificial Intelligence Research, 2011,27(1): 441−464.

[42] Cohen DA, Cooper MC, Creed P, Marx D, Salamon AZ. The tractability of CSP classes defined by forbidden patterns. Journal of Artificial Intelligence Research, 2012,45:47−78.

[43] Cohen D, Cooper MC, Escamocher G, Zivny S. Variable and value elimination in binary constraint satisfaction via forbidden patterns. Journal of Computer and System Sciences, 2015,81(7):1127−1143.

[44] Mouelhi A. A BTP-based family of variable elimination rules for binary CSPs. In: Singh SP, ed. Proc. of the 21st AAAI Conf. on Artificial Intelligence. San Francisco: AAAI Press, 2107. 3871−3877.

[45] Cooper MC, Duchein A, Mouelhi A, Escamocher G, Terrioux C, Zanuttini B. Broken triangles: From value merging to a tractable class of general-arity constraint satisfaction problems. Artificial Intelligence, 2016,234(9-11):196−218.

[46] Mouelhi A, Jégou P, Terrioux C. A hybrid tractable class for non-binary CSPs. Constraints, 2015,20(4):383−413.

[47] Mouelhi A, Jégou P, Terrioux C. Hidden tractable classes: From theory to practice. In: Proc. of the 26th IEEE Int’l Conf. on Tools with Artificial Intelligence. Limassol: IEEE Computer Society, 2014. 437−445.

[48] Aspvall B, Plass MF, Tarjan RE. A linear-time algorithm for testing the truth of certain quantified Boolean formulas. Information Processing Letters, 1979,8(3):121−123.

[49] Karpinski M, Buning K, Schmitt PH. On the computational complexity of quantified horn clauses. In: Börger E, ed. Proc. of the 1st Workshop on Computer Science Logic (CSL’87). LNCS 329, Karlsruhe: Springer-Verlag, 1987. 129−137.

[50] Gottlob G, Greco G, Scarcello F. The complexity of quantified constraint satisfaction problems under structural restrictions. In: Kaelbling LP, ed. Proc. of the 19th Int’l Joint Conf. on Artificial Intelligence. Edinburgh: Professional Book Center, 2005. 150−155.

[51] Chen H. Meditations on quantified constraint satisfaction. In: Constable RL, ed. Proc. of the Logic and Program Semantics. LNCS 7230, Springer-Verlag, 2012. 35−49.

[52] Börner F, Bulatov AA, Chen H, Jeavons P, Krokhin AA. The complexity of constraint satisfaction games and QCSP. Information and Computation, 2009,207(9):923−944.

[53] Li ZS, Li HB, Zhang YG, Wang ZW. An approach of solving cosntraint satisfaction problem based on cycle-cut. Chinses Journal of Computers, 2011,34(8):1528−1536 (in Chinese with English abstract).

[54] Li HB, Liang YC, Li ZS. Simple tabular reduction for generalized arc consistency on negative table constraints. Ruan Jian Xue Bao/Journal of Software, 2016,27(11):2701−2711 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4874.htm [doi: 10.13328/j.cnki.jos.004874]

[55] Yang MQ, Li ZS, Li Z. Optimizing MDDc and STR3 for solving CSPs. Ruan Jian Xue Bao/Journal of Software, 2017,28(12): 3156−3166 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5242.htm [doi: 10.13328/j.cnki.jos.005242]

[56] Misra N, Ordyniak S, Raman V, Szeider S. Upper and lower bounds for weak backdoor set detection. In: Järvisalo M, ed. Proc. of the 16th Int’l Conf. on Theory and Applications of Satisfiability Testing (SAT 2013). LNCS 7962, Helsinki: Springer-Verlag, 2013. 394−402.

附中文参考文献:

[9] 林晓华,冯毅雄,谭建荣,冯勇,高一聪.基于量词约束满足的机械产品质量特性稳健优化设计方法.机械工程学报,2013,49(15): 169−179.

[14] 殷明浩,周俊萍,孙吉贵,谷文祥.求解 QBF 问题的启发式调查传播算法.软件学报,2011,22(7):1538−1550. http://www.jos.org.cn/1000-9825/3859.htm [doi: 10.3724/SP.J.1001.2011.03859]

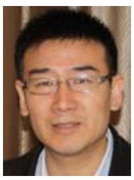
[53] 李占山,李宏博,张永刚,王孜文.一种基于环切割的约束满足问题求解算法.计算机学报,2011,34(8):1528−1536.

[54] 李宏博,梁艳春,李占山.负表约束的简单表缩减广泛弧相容算法.软件学报,2016,27(11):2701−2711. http://www.jos.org.cn/1000-9825/4874.htm [doi: 10.13328/j.cnki.jos.004874]

[55] 杨明奇,李占山,李哲.优化求解约束满足问题的 MDDc 和 STR3 算法.软件学报,2017,28(12):3156−3166. http://www.jos.org.cn/1000-9825/5242.htm [doi: 10.13328/j.cnki.jos.005242]

高健(1983－),男,辽宁沈阳人,博士,副教授,CCF 专业会员,主要研究领域为约束程序设计,自动推理.

李辉(1983－),男,博士,副教授,主要研究领域为众包软件,复杂网络.



陈荣(1969－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为软件诊断,群体智能,优化调度.