基于量化布尔公式的条件规划编码方式

高冰冰 张长海 吕 帅
(吉林大学计算机科学与技术学院 长春 130012)

摘 要 介绍条件规划问题及其相关的求解系统,着重分析以逻辑为基础的编码方式。针对基于量化布尔公式的转换方法进行详细分析,给出3种不同形式的量化布尔公式编码。最后,对这3种编码进行比较,分析基于命题逻辑公式与量化布尔公式这两种不同转换方式的优劣,讨论基于量化布尔公式的规划方法未来的研究方向和发展趋势。

关键词 条件规划,不确定性规划,量化布尔公式,编码

中图法分类号 TP181 **文献标识码** A

Conditional Planning Encodings Based on Quantified Boolean Formulas

GAO Bing-bing ZHANG Chang-hai LU Shuai
(College of Computer Science and Technology, Jilin University, Changchun 130012, China)

Abstract This paper introduced the conditional planning problems and their associated planners, and analyzed logic based encoding methods. By analyzing translation based planning methods, whose targets are quantified boolean formulas, it introduced three different forms of quantified boolean formula encodings. Finally, it compared the above encodings, analyzed their respective advantages and disadvantages of two different translation mode based on propositional logic formulas and quantified boolean formulas, and then discussed the future research directions and trends on planning methods based on quantified boolean formulas.

Keywords Conditional planning, Nondeterministic planning, Quantified boolean formulas, Encoding

智能规划是人工智能领域的一个重要分支,近年来被众多国内外研究者所关注,发展迅速。在现实世界中,环境、资源、时间和操作等不同的约束条件刻画了不同种类的智能规划问题。早期的智能规划研究主要针对带有许多假设的经典规划,这种对于时间、世界乃至动作效果的严格界定往往是不符合实际需求的。随着研究的逐步深入,研究者把目光投向了具有更高难度的不确定性规划问题,这也标志着智能规划研究逐渐趋近于模型的实用性。

不确定性规划主要分为conformant规划、条件规划、概率规划和条件概率规划等。不确定性规划允许存在动作、世界以及状态的不确定性,更加符合人们对资源或环境往往不够了解的现实情况。本文主要针对条件规划的相关内容进行介绍。在条件规划问题中,不确定性主要体现在其动作和状态都可能被某种条件所约束,由此增加了规划求解的难度。

近年来,涌现出许多解决条件规划问题的方法。1992年Peot等人提出的CNLP算法[1]和1996年Pryor提出的Cassandra算法[2]是以最小承诺或偏序规划为基础的。它们的主导思想是:首先将条件规划问题转化为一系列确定性子问题,然后分别求解这些子问题。这种转化方式往往是繁琐的,通常无法在线性时间内完成。1998年,Cimatti等人提出了一个枚举条件规划问题状态空间的算法[3],该算法导致了大规模的规划解,并且对于复杂的规划问题,算法的性能并不清晰。随后的一些能够解决包含观测(部分可观察性)的条件规划问题的规划方法,大都将其看作在信念状态空间上的启发式搜索过程[4]。

1992至1996年,Kautz等人将命题逻辑的可满足性(proposition satisfiability, SAT)问题用于求解经典规划[5-7],引发了人们以逻辑为基础解决条件规划问题的设想。然而,事实上将条件规划转化为SAT问题求解是不可行的,甚至判定是否存在多项式大小的规划解也超出了NP复杂度。因此,研究者将基于SAT的经典规划方式进行推广。1999年,Rintanen首次提出将条件规划问题转化成比命题逻辑公式更为复杂的量化布尔公式(Quantified Boolean Formulas, QBF)[8],将规划问题转换成前缀为∃∀∃形式的QBF公式。这种用QBF公式解决规划问题的系统称为QBFPlan。该思想为随后产生的多个解决不确定性问题的方法奠定了基础。

2000年,Giunchiglia等人设计了一种针对带有不确定性信息的*C*域[9]上的*C*-Plan系统[10],该系统的运行机理与QBFPlan类似。*C*-Plan将规划过程分离为寻找规划解及证明解的正确性两个部分,后者被归约为一个可满足性判定。Giunchiglia等人认为,利用QBF求解器的搜索同样可以反映出通过*C*-Plan进行的搜索过程。*C*-Plan系统并没有明确地

到稿日期:2009-12-15 返修日期:2010-03-07 本文受国家自然科学基金(60603031,60773097,60873044,60873148),高等学校博士学科点专项科研基金(20060183044,20070183057)资助。

高冰冰(1986-),女,硕士生,主要研究方向为智能规划与自动推理,E-mail:gaobingbing1031@126.com;张长海(1949-),男,教授,博士生导师,主要研究方向为程序设计语言、程序分析技术、智能规划;吕 帅(1981-),男,博士,主要研究方向为智能规划与自动推理。

应用 QBF 公式形式，而是隐含地利用前缀为 $\forall\exists$ 和 $\exists\forall$ 形式的 QBF 语义。这种前缀数量由 3 缩减为 2 的 QBF 被 Rintanen 明确地应用到条件规划问题的求解过程中。

2004 年，Brafman 等人在 *C*-Plan 基础上提出了一个基于启发式搜索的规划方法[11]。2005 年，Palacios 等人隐含地运用了 Rintanen 前缀为 $\exists\forall\exists$ 的 QBF 形式，主要针对 QBF 的求解问题进行研究，设计了一种基于 DNNF/SAT 的 QBF 求解器[12,13]。其主导思想是：通过消除 $\exists\forall\exists$ 的两个内层量词 $\forall\exists$，将 QBF 约简为 SAT 问题来进行求解，而其规划解是通过寻找使结果公式可满足的真值指派来完成的。2006 和 2007 年，Rintanen 等人对 QBF 规划编码进行了进一步研究，先后提出了解决冲突问题的最优线性编码[14]和前缀数量较少的 QBF 形式[15]，从而大大降低了编码的复杂度。

以 QBF 为基础解决某些不确定性规划问题明显表现出较好的性能，比如条件规划问题，当条件规划问题被限定而无可观察性时，也就是著名的 conformant 规划问题。显然以 QBF 为基础的规划系统也可以应用于求解 conformant 规划问题。事实上，虽然理论上将条件规划问题转换成 QBF 公式是可行的，但由于条件规划问题往往具有可观察性，增加了转化难度，因此现有的几种解决不确定性规划的 QBF 编码形式主要针对不带观测的、相对结构较简单的 conformant 问题。但不可忽视的是，这几种处理 conformant 规划问题的 QBF 编码，都能够直接推广成带有部分可观察性的更通用的条件规划问题。

本文第 1 节先简单介绍 QBF 相关知识；第 2 节介绍 3 种前缀分别为 $\exists\forall\exists$，$\exists\forall$ 和 $\forall\exists$ 的 QBF 编码，并分析这几种编码的构成和含义；第 3 节对这 3 种编码进行比较与分析；最后对以 QBF 为基础的规划编码方式进行总结与展望。

1 QBF 相关知识

若干文字的析取式称为子句。

定义 1(量化布尔公式) 一个量化布尔公式(QBF)定义为如下形式：

$q_1x_1\cdots q_nx_n\Phi$

式中，$q_1x_1\cdots q_nx_n$ 为 QBF 的前缀，Φ 为主体。前缀由两部分组成：可取全称量词 $\forall$ 或存在量词 $\exists$ 的量词 $q_i(i=1,\cdots,n)$，以及 Φ 中出现的所有命题变量 $x_i(i=1,\cdots,n)$。主体部分 Φ 是合取范式，形式为 $\alpha_1\wedge\alpha_2\wedge\cdots\wedge\alpha_n$，其中 $\alpha_i(i=1,\cdots,n)$ 是子句。

QBF 前缀中 $q_ix_i(i=1,\cdots,n)$ 的语义是：变量 x_i 受量词 q_i 约束，同一个变量只能被 $\forall$ 量词或 $\exists$ 量词中的一个约束。当 q_i 为 $\forall$ 量词，称 x_i 为全称变量；当 q_i 为 $\exists$ 量词，称 x_i 为存在变量。

为了简便，当相邻变量被同一量词约束，可省略中间量词。例如，$\forall x_1\forall x_2\exists x_3\exists x_4\exists x_5\Phi$ 等价于 $\forall x_1x_2\exists x_3x_4x_5\Phi$。

如果存在一个真值指派使 QBF 为真，则 QBF 可满足；否则，不可满足。有时，也将判定 QBF 的可满足性问题简称为 QSAT 问题。

例 1 $\forall x_1\exists x_2(x_1\vee x_2)\wedge(\neg x_1\vee\neg x_2)$ 是可满足的，因为当 x_1 取真，存在 x_2 取假的指派，使 QBF 公式的主体为真；当 x_1 取假，存在 x_2 取真的指派，使主体为真。$\exists x_2\forall x_1(x_1\vee x_2)\wedge(\neg x_1\vee\neg x_2)$ 则是不可满足的，因为当 x_2 取真，x_1 为假时，主体为真，但对于 x_1 为真，主体为假；当 x_2 取假，x_1 为真时，主体为真，但对于 x_1 为假，主体为假。因此改变前缀后的 QBF 公式是不可满足的。由此也可以看出：QBF 前缀中量词的顺序决定着它的可满足性，不可以任意改变量词的顺序。

记 $\sum\limits_0^p=P,\prod\limits_0^p=P,\sum\limits_{i+1}^p=NP^{\Sigma_i^p},\prod\limits_{i+1}^p=co\text{-}\sum\limits_{i+1}^p$

前缀为 $\exists x_1x_2\cdots x_n$ 的 QBF 可满足性是 $NP=\sum\limits_1^p$-*complete* 问题；前缀为 $\forall x_1x_2\cdots x_n$ 是 *co*-NP$=\prod\limits_1^p$-*complete* 问题。如果采用上例 $\forall x_1x_2\exists x_3x_4x_5\Phi$ 这种简化形式，那么前缀数量为 i，且形式为 $\forall\exists\forall\cdots$ 的 QBF 是 $\prod\limits_i^p$-*complete*；而为 $\exists\forall\exists\cdots$ 则是 $\sum\limits_i^p$-*complete*[16]。

2 基于 QBF 的条件规划编码

定义 2(操作效果) 设 A 是状态变量集，$\langle c,e\rangle$ 代表一个操作 o，其中 c 是关于 A 的前提，e 是关于 A 的效果。e 被递归定义如下[14]：

1) 对于变量 $a,a\in A$，则 a 和 $\neg a$ 是关于 A 的效果。

2) 若 $e_1,\cdots,e_n$ 是关于 A 的效果，则 $e_1\wedge\cdots\wedge e_n$ 是关于 A 的效果(特殊情况，当 $n=0$ 时，是空效果)。

3) 若 c 是关于 A 的公式，且 e 是关于 A 的效果，则 $c\rhd e$ 是关于 A 的效果。

4) 若 $e_1,\cdots,e_n$ 是关于 A 的效果，则 $e_1|\cdots|e_n$ 是关于 A 的效果。

在定义 2 中，混合效果 $e_1\wedge\cdots\wedge e_n$ 是同时执行所有效果 $e_1,\cdots,e_n$ 的表示符号；在条件效果 $c\rhd e$ 中，如果 c 在当前状态为真，则 e 被执行，这也是条件规划问题的表现形式，某些效果需要满足一定条件才能成立；不确定性效果 $e_1|\cdots|e_n$，表示在 $e_1,\cdots,e_n$ 中随机选取一个效果。

定义 3(条件规划问题) 动作效果或初始状态被某些特定条件约束的不确定性规划问题称为条件规划问题。

条件规划问题可伴有可观察性约束(部分可观察性)。

定理 1(条件规划问题的复杂度) 对于一个条件规划问题，其规划解的存在性是 $\prod\limits_2^p$-hard[8]。

根据不同的规划解定义、可观察性假设以及不确定性构成，可以得到不同类型的条件规划问题。当条件规划不带有可观察性时，它也就是通常的 conformant 规划。下面分别介绍这种无可观察性条件规划问题的 QBF 编码，有关内容详见文献[8-15]。由于着重研究动作效果的不确定性，假定初始状态是确定的。

2.1 规划框架

与 SAT 经典问题的解决方法相似，以 QBF 为基础的规划系统由 3 个相互独立的部分组成：

1) 指定规划长度为 t 的规划问题的高效 QBF 编码 $\Phi(t)$。

2)判定 $\Phi(t)$ 的高效算法，即 QBF 求解器。

3)寻找最优规划解步骤 t 的高效算法。

本文着重讲解第 1)部分，也就是规划问题的逻辑编码方式。本节介绍前缀分别为 $\exists\forall\exists$，$\exists\forall$ 以及 $\forall\exists$ 的 3 种不同

的 QBF 编码，这些编码能够直接推广成带有部分可观察性的、更通用的条件规划问题，并且可以直接利用相关的 QBF 求解器予以求解。目前，已经有很多不同种类的 QBF 求解器，比如 Giunchiglia 等人提出的基于 DPLL 算法的 QuBE-BT[17]，QuBE-BJ[17]，QuBE-Rel[18]，Biere 等人提出的基于消除量词的算法[18]，以及 Palacios 等人提出的基于 DNNF 的算法[12]等，本文不对求解器做详细的介绍。对于第 3)部分，最简单的方法是从 $t=0,1,2\cdots$顺序进行测试，直到得到满足条件的最小 t 值。类似于 SAT 方法中的求解技术，也可以允许通过测试最优规划长度来提高求解效率[14]。

2.2 编码

本节分别介绍条件规划编码的相关内容。

2.2.1 原子效果 l 的执行条件

基于并发规划解的定义：只要操作不冲突，则允许多个操作同时执行，且不论执行顺序如何，效果都相同。

一个操作 $o=\langle c,e\rangle$，由于本文重在针对操作效果 e 的不确定性进行分析，因此简化对前提 c 的判断，不考虑 $s\not\models c$ 时操作 o 的执行情况，而是采用以下更严格的假定。

假定 1(操作执行)　操作 o 是可执行的，当且仅当它的前提 c 成立。

假定 2(条件效果)　条件效果 $c\rhd e$，效果 e 不包含不确定性"|"。

将不确定性效果 $e_1|\cdots|e_n$ 表示成双重选择形式，例如 $a|b|c|d$ 表示成$(a|b)|(c|d)$。这样便于引入辅助变参进行编码。每一个选择$(a|b)$，$(c|d)$，a,b,c,d 都可以用不同的辅助变参来独立编码。

定义 4(原子效果)　不包含$\wedge$，$\rhd$，|的单一效果 l 为原子效果。

定义 5(原子效果的条件)　当操作 $o=\langle c,e\rangle$的效果 e 被执行时，那么原子效果 l 的效果前提表示为$EPC_l^{nd}(e,\sigma)$(EPC 为 Effect PreCondition 的缩写)。其中，整数序列 σ 用来生成控制不确定性的辅助变参 x_σ。$EPC_l^{nd}(e,\sigma)$定义如下：

1) $EPC_l(T)=\bot$：空效果 T 不包含任何原子效果，所以原子效果 l 的执行前提为假；

2) $EPC_l(l)=T$：效果 l 被执行，而其自身的执行条件为 T；

3) $\forall l\neq l'.EPC_l(l')=\bot$：$l'$本身就是原子效果，且 $l\neq l'$，对于此操作 o，l 不被执行，故 l 的执行条件为假$\bot$；

4) $EPC_l(e_1\wedge\cdots\wedge e_n)=EPC_l(e_1)\vee\cdots\vee EPC_l(e_n)$：$e_i$ 包含 l $(i=1,\cdots,n)$，那么 l 在 $e_1\wedge\cdots\wedge e_n$ 下的执行条件要在 $EPC_l(e_i)$中选取；

5) $EPC_l(c\rhd e)=c\wedge EPC_l(e)$：$l$ 在 $c\rhd e$ 下的执行条件是c 成立且 l 在 e 下的条件成立；

6) 若 e 是确定性的，$EPC_l^{nd}(e,\sigma)=EPC_l(e)$：$\sigma$ 是为了生成不确定性辅助变参 x_σ 的整数序列，当 e 是确定性时，σ 没有继续使用的意义；

7) $EPC_l(c\rhd e,\sigma)=c\wedge EPC_l(e,\sigma)=c\wedge EPC_l(e)$：根据假定 2，$e$ 为确定性效果，故结果与 5)相同；

8) $EPC_l^{nd}(e_1|e_2,\sigma)=(x_\sigma\wedge EPC_l^{nd}(e_1,\sigma1))\vee(\neg x_\sigma\wedge EPC_l^{nd}(e_2,\sigma1))$：用 x_σ 和$\neg x_\sigma$ 来编码选择性，x_σ 代表选择 e_1 分支；$\neg x_\sigma$ 代表选择 e_2 分支；

9) $EPC_e^{nd}(e_1\wedge\cdots\wedge e_n,\sigma)=EPC_l^{nd}(e_1,\sigma1)\vee\cdots\vee EPC_l^{nd}(e_n,\sigma n)$：$\sigma$ 后面排列的 $1,\cdots,n$ 分别代表效果 $e_1,\cdots,e_n$ 的编号。

总的来看，编码操作效果的不确定性主要依赖于 8)和 9)，用 x_σ 和$\neg x_\sigma$ 分表代表选取 $e_1|e_2$；用σi 代表选取 $e_1\wedge\cdots\wedge e_n$ 中的 $e_i(i=1,\cdots,n)$

例 2　$EPC_e^{nd}(c\wedge(b|d\rhd e)),\sigma)=EPC_e^{nd}(c,\sigma1)\vee EPC_e^{nd}((b|d\rhd e),\sigma2)$

$=EPC_e(c)\vee EPC_e^{nd}((b|d\rhd e),\sigma2)$

$=\bot\vee(x_{\sigma2}\wedge EPC_e^{nd}(b,\sigma21))\vee(\neg x_{\sigma2}\wedge EPC_e^{nd}(d\rhd e,\sigma21))$

$=(x_{\sigma2}\wedge EPC_e(b))\vee(\neg x_{\sigma2}\wedge d\wedge EPC_e(e))$

$=(x_{\sigma2}\wedge\bot)\vee(\neg x_{\sigma2}\wedge d\wedge T)$

$=\neg x_{\sigma2}\wedge d$

$EPC_l^{nd}(e,\sigma)$的定义对于解决条件规划是至关重要的。如上例的最终结果为：效果 $c\wedge(b|d\rhd e)$中含有的原子效果 e，它对应的编码是$\neg x_{\sigma2}\wedge d$。

通过设置辅助变参 x_σ 来编码不确定性，不同的辅助变参对应不同的效果。

2.2.2 框架公理

设 $e_1,\cdots,e_n$ 分别是操作 $o_1,\cdots,o_n$ 的效果，且每个操作 o_i $(i=1,\cdots,n)$都有一个唯一的整数标示 $\Omega(o_i)$。$\Omega(o_i)$也就是上面 σ 的初始取值。对于状态变量 $a\in A$，用命题变量 a 来表示当前状态中 a 的值，a'来表示后继状态中 a 的值。

解释性框架公理表示在何种条件下状态变量 a 可能由真到假，或者由假到真。

$$(a\wedge\neg a')\rightarrow\bigvee_{i=1}^{n}(o_i\wedge EPC_{\neg a}^{nd}(e_i,\Omega(o_i)))\tag{1}$$

$$(\neg a\wedge a')\rightarrow\bigvee_{i=1}^{n}(o_i\wedge EPC_a^{nd}(e_i,\Omega(o_i)))\tag{2}$$

也就是说，如果某个状态变量的值发生变化，那么一定执行了某个或某些操作，并且这些操作的效果中关于此状态变量的执行条件被满足。

2.2.3 操作的相关公式

对于任何被执行的操作 $o=\langle c,e\rangle$，都有如下公式成立，此公式描述了操作执行时各状态变量的值：

$$(o\rightarrow c)\wedge\bigwedge_{a\in A}(o\wedge EPC_a^{nd}(e,\Omega(o))\rightarrow a')\wedge\bigwedge_{a\in A}(o\wedge EPC_{\neg a}^{nd}(e,\Omega(o))\rightarrow\neg a')\tag{3}$$

此公式表示：如果操作 o 被执行，那么它的前提 c 一定成立(根据假设 1)，并且原子效果 a 在 o 下的效果条件成立，那么当操作执行后，效果 a 成立(这里用 a'表示)。

例 3　有两个操作 o_1 和 o_2，$\Omega(o_i)$取操作的下标：$\Omega(o_1)=1$，$\Omega(o_2)=2$。$o_1=\langle\neg a,(b|(c\rhd d))\wedge(a|c)\rangle$，$o_2=\langle\neg b,(((d\rhd b)|c)|a)\rangle$这两个操作的执行被描述成如下形式：

1) 应用式(1)和式(2)，得到 8 个公式：

①$\neg(a\wedge\neg a')$

②$(\neg a\wedge a')\rightarrow((o_1\wedge x_{12})\vee(o_2\wedge\neg x_2))$

③$\neg(c\wedge\neg c')$

④$(\neg c\wedge c')\rightarrow((o_1\wedge\neg x_{12})\vee(o_2\wedge x_2\wedge\neg x_{21}))$

⑤$\neg(b\wedge\neg b')$

⑥$(\neg b\wedge b')\rightarrow((o_1\wedge x_{11})\vee(o_2\wedge x_2\wedge x_{21}\wedge d))$

⑦$\neg(d\wedge\neg d')$

⑧$(\neg d\wedge d')\rightarrow(o_1\wedge\neg x_{11}\wedge c)$

2) 操作 o_1 应用式(3)，得到 5 个公式：

①$o_1 \to \neg a$ ②$(o_1 \wedge x_{12}) \to a'$ ③$(o_1 \wedge x_{11}) \to b'$

④$(o_1 \wedge \neg x_{12}) \to c'$ ⑤$(o_1 \wedge \neg x_{11} \wedge c) \to d'$

3) 操作 o_2 应用式(3),得到 4 个公式:

①$o_2 \to \neg b$ ②$(o_2 \wedge \neg x_2) \to a'$

③$(o_2 \wedge x_2 \wedge x_{21} \wedge d) \to b'$ ④$(o_2 \wedge x_2 \wedge \neg x_{21}) \to c'$

对 1)中的①②进行分析,其余同理。

①$\neg(a \wedge \neg a')$:由于操作 o_1 的前提是$\neg a$ 成立,故 $a \wedge \neg a'$是不成立的。

②对操作 o_1 应用式(3),得:

$(\neg a \wedge a') \to (o_1 \wedge EPC_a^{nd}((b|(c \rhd d)) \wedge (a|c),1)) \vee (o_2 \wedge EPC_a^{nd}(((d \rhd b)|c)|a,2))$

其中,$EPC_a^{nd}((b|(c \rhd d)) \wedge (a|c),1)$

$= EPC_a^{nd}(b|(c \rhd d),11) \vee EPC_a^{nd}(a|c,12)$

$= (x_{11} \wedge EPC_a^{nd}(b,111)) \vee (\neg x_{11} \wedge EPC_a^{nd}(c \rhd d,111)) \vee (x_{12} \wedge EPC_a^{nd}(a,121)) \vee (\neg x_{12} \wedge EPC_a^{nd}(c,121))$

$= (x_{11} \wedge \bot) \vee (\neg x_{11} \wedge c \wedge \bot) \vee (x_{12} \wedge T) \vee (\neg x_{12} \wedge \bot)$

$= x_{12}$

$EPC_a^{nd}(((d \rhd b)|c)|a,2)$

$= (x_2 \wedge EPC_a^{nd}((d \rhd b)|c,21)) \vee (\neg x_2 \wedge EPC_a^{nd}(a,21))$

$= (x_2 \wedge ((x_{21} \wedge EPC_a^{nd}(d \rhd b,211)) \vee (\neg x_{21} \wedge EPC_a^{nd}(c,211)))) \vee (\neg x_2 \wedge T)$

$= (x_2 \wedge ((x_{21} \wedge d \wedge \bot) \vee (\neg x_{21} \wedge \bot))) \vee (\neg x_2)$

$= \neg x_2$

故,②$= (\neg a \wedge a') \to ((o_1 \wedge x_{12}) \vee (o_2 \wedge \neg x_2))$。

当两个操作 o_1 和 o_2 不发生冲突时,它们才可以并行执行。用子句$\neg(o_1 \wedge o_2)$来限制互相冲突的操作。

用$R(A,A',O,X)$表示以上所有公式的合取,其中 X 是公式中出现的所有 x_σ 的集合。用命题变量 a^i 表示 i 时刻状态变量$a(a \in A)$的值,A^i 表示所有命题变量 a^i 的集合。例如,用以下形式表示描述 0 时刻到 2 时刻长度为 2 的操作序列:$R(A^0,A^1,O^0,X^0) \wedge R(A^1,A^2,O^1,X^1)$。为了简化,将 $R(A^i,A^{i+1},O^i,X^i)$缩写为 $T(i,i+1)$。

2.3 前缀为∃∀∃形式的编码

前缀为∃∀∃形式,解决条件规划问题的标准 QBF 编码:

$$\exists P \forall C \exists E(I^0 \to (\bigwedge_{i=0}^{t-1} T(i,i+1) \wedge G^t)) \tag{4}$$

其中,

1) $P=\bigcup_{i=0}^{t-1} O^i$表示所有规划解操作的集合;

2) $C= A^0 \cup \bigcup_{i=0}^{t-1} X^i$ 表示初始状态以及所有不确定性事件的编码集合;

3) $E=\bigcup_{i=1}^{t} A^i$ 表示其余所有状态变量的集合;

4) I 和 G 分别表示初始状态和目标状态;

5) $T(i,i+1)$表示操作的执行过程。

该公式表示:存在一个规划解 P,对于所有的随机事件 C,都存在一个执行 E,使得所有目标状态得以实现。确定规划解的存在性也就是判断是否存在集合 P。对于任何随机事件 C,都有一个执行 E 使得规划可以从初始状态依次进行各规划步(每步操作都含有不确定性效果),直至达到目标状态。

在预备知识中已经提及,求解前缀数量为 i,且形式为∃∀∃…的 QBF 公式是复杂度为$\sum_i^p$-*complete* 问题,则求解条件规划编码$\exists P \forall C \exists E(I^0 \to (\bigwedge_{i=0}^{t-1} T(i,i+1) \wedge G^t))$是$\sum_3^p$-*complete* 的。根据定理 1,对于一个条件规划问题,证明其规划解的存在性是$\prod_2^p$-*hard*。

那么很自然将产生一个疑问:用 3 级多项式复杂的编码来处理 2 级复杂的问题,是否会将原本简单的问题复杂化。虽然 Rintanen 以实验证实了大多数情况下,前缀为∃∀∃的 QBF 编码并不会影响对条件规划的求解,相反利用此编码方式会使很多问题实例得到理想的运行时间。但设想一下,如果前缀数量为 2,即将复杂度降至 2 级的编码是否会更加理想。

2.4 前缀为∃∀形式的编码

考虑将前缀为$\exists P \forall C \exists E$ 的 QBF 编码转化成∃∀的形式,有两种可行的办法:删除最内层的存在变量 E;或者将 E 转化为全称变量。实验结果表明,第一种方法最坏的情况会以指数大小增加 QBF 公式。因此选择第二种方法,编码如下:

$$\exists P \forall E(\iota \wedge [(I^0 \wedge \bigwedge_{i=0}^{t-1} T(i,i+1)) \to G^t] \wedge \nu) \tag{5}$$

式中,

1) $P=\bigcup_{i=0}^{t-1} O^i$,表示所有规划解操作的集合,与式(4)相同;

2) $E=\bigcup_{i=0}^{t} A^i \cup \bigcup_{i=0}^{t-1} X^i$,表示将式(4)中的 C 和 E 合并在一起的集合;

3) $\iota=\bigwedge_{i=0}^{t-1} \bigwedge_{o_1,o_2 \in O \text{且互斥}} \neg(o_1^i \wedge o_2^i)$,表示限制冲突操作同时执行;

4) $\nu=I^0 \to \bigwedge_{i=0}^{t-1}[(\bigwedge_{j=0}^{i-1} T(j,j+1)) \to \bigwedge_{o \in O}(o^i \to prec(o)^i)]$

式中,$prec(o)$是操作 o 的前提,ν 声明规划解的可执行性,当前操作步的执行结果必须使下一步操作的前提为真,以确保规划由初始状态一直到目标状态的执行。

与前缀为$\exists P \forall C \exists E$ 的编码方式不同,由于规划解的有效性分别由 ι 和 ν 进行测试,故$\exists P \forall E$ 编码中的 $T(i,i+1)$不需要包括冲突和前提约束。

对于公式$\exists P \forall E \Phi$,给出一个规划解 P,所有变量 E 的值必须满足 Φ。这些值既代表此规划解需要达到目标的所有执行,也代表那些不对应执行、不用满足目标的状态序列。

式(5)的规模为 $O(n^2)$,因为 ι 的规模为 $O(n^2)$,且 $T(i,i+1)$在 ν 中的出现次数也为 $O(n^2)$。首先通过递归定义元-变量 $\phi_i, i \in \{0,\cdots,t\}$,削减 $T(i,i+1)$的数量级到 t:

$$\phi_i=\begin{cases} G^t, & i=t \\ \bigwedge_{o \in O}(o^i \to prec(o)^i) \wedge (T(i,i+1) \to \phi_{i+1}), & i<t \end{cases}$$

ϕ_i 以递归的形式描述了相邻操作步的关联以及操作的执行必须是在前提为真的情况下。将 ϕ_i 应用于式(5)(由 G^t 向前递归推导),得:

$$\exists P \forall E(\iota \wedge (I^0 \to \phi_0)) \tag{6}$$

而对于冲突公式 ι,Rintanen 还提出了一个渐进最优线性编码 ι'。设 X 为公式中出现的新的辅助变参集,在式(5)的基础上得到线性大小的编码$\exists P \exists X \forall E(\iota' \wedge (I^0 \to \phi_0))$,简记为:

$$\exists P X \forall E(\iota' \wedge (I^0 \to \phi_0)) \tag{7}$$

2.5 前缀为∀∃形式的编码

大多数 QBF 求解器都是将 QBF 公式的主体部分转化为 CNF 范式，这种转化过程往往会增加指数大小的公式数。Rintanen 提出一种利用辅助变量的较好的线性 CNF 转换器，然而对于式(7)这种转换器不能使用，因为辅助变量必须为∃变量，且出现在 E 的后面。

前缀为∀∃的 QBF，是通过对式(7)取反所得：

$$\forall PX \exists E(\neg\iota' \vee (I^0 \wedge \neg\phi_0)) \quad (8)$$

该 QBF 编码表示：对于所有的候选规划解，或者规划解是非法的(冲突或无法执行)；或者存在执行，但无法达到目标，因为最内层为∃变量 E，可以使用线性 CNF 转换器。与∃∀形式的 QBF 不同，式(8)对应不可满足性。也就是说，规划解从 P 值中获得，而针对这些 P 值存在 E 使公式 $\neg\iota' \vee (I^0 \wedge \neg\phi_0)$ 为假。

对于 QBF 公式，最外层的约束变量越少(即式(8)中的 $\forall PX$)，求解效率会越高[16]。Rintanen 用一个新的公式 ι_2 对式(7)中的 $\neg\iota'$ 进行替换，并引入了新的辅助变量 X'，得到新的编码：

$$\forall P \exists X'E(\iota_2 \vee (I^0 \wedge \neg\phi_0)) \quad (9)$$

公式 ι_2 与 ι' 的含义相近：a 是一个状态变量，如果一个使 a 为真的操作没有被执行，那么一定有一个后续(顺序是随机选择的)的使 a 为真的操作被执行。并且对于一个使 a 为真的已经执行的操作，存在一个后续的执行操作，由于 a 出现在该后续操作的前提中或者为某个效果的前提，故此操作或者使 a 为假，或者不要求 a 为真。

相比式(8)，式(9)的最外层被∀约束的变量数显著减少。

3 编码比较

通过标准测试问题，对这 3 种 QBF 编码进行比较，并与 SAT 技术进行对比分析。

3.1 3 种编码的比较

单从复杂度的角度而言，求解 $\exists P \forall C \exists E$ 形式的复杂度是 $\sum_3^p$，而∃∀和∀∃仅为 $\sum_2^p$，应该比前者更为理想。但让人意外的是，对于某些问题，$\exists P \forall C \exists E$ 编码却能得到较短的运行时间。换句话说虽然条件规划问题的难度是 $\prod_2^p$，但∃∀∃编码的求解复杂度 $\sum_3^p$ 并没有给规划带来很大的负担。

Rintanen 通过标准测试问题 sorting networks[8]、BTC、积木世界(BW)、ring 和 square 问题，对前缀数量为 3 的式(4)和前缀数量为 2 的式(9)两种 QBF 编码[18]以及 Yu 等人的 yQuaffle[20]进行实验。

实验结果表明，∃∀∃和∀∃的 QBF 公式在求解过程中都是线性增长的。对于 QuBE-Rel 求解器，∃∀∃编码总是优于∀∃；而对于 yQuaffle 求解器，大多数情况∀∃较为理想。∃∀∃/QuBE-Rel 的优势主要展现在 BTC，BW 以及 ring 问题；而∀∃/yQuaffle 的优势主要展现在 sorting networks 问题。总的来说，对于结构较为简单的问题，∃∀∃占有优势；而对于像 sorting networks 这样结构复杂的问题，∀∃会有更快的执行时间。对于大多数标准测试样例，其最难可解的 QBF 公式大小在 10kB 至 30kB 之间。

3.2 与 SAT 的比较

正如我们在 QBF 相关知识中提到的，确定前缀为 $\exists x_1 x_2 \cdots x_n$ 的 QBF 的可满足性是 $NP = \sum_1^p$-*complete* 问题；前缀为 $\forall x_1 x_2 \cdots x_n$ 是 co-NP$=\prod_1^p$-*complete* 问题。总的来说，如果使用 $\forall x_1 x_2 \exists x_3 x_4 x_5 \Phi$ 这种简化形式，那么前缀数量为 i，且形式为∀∃∀…的 QBF 是 $\prod_i^p$-*complete*；而为∃∀∃…则是 $\sum_i^p$-*complete*。

由此可以看出，命题逻辑的 SAT 问题实质上是 QSAT 问题的特殊形式，QSAT 将 SAT 泛化。当 QBF 公式的前缀为 $\exists x_1 x_2 \cdots x_n$ 时就相当于 SAT 问题。自然，QSAT 问题要比 SAT 问题复杂得多。理论上，QSAT 问题是 PSPACE-*complete*，而 SAT 问题是 NP-*complete*[21]。从求解方面考虑，QBF 公式前缀中量词所约束变量的位置是至关重要的，根据定义 2 及 QBF 的标准形式 $q_1 x_1 q_2 x_2 \cdots q_n x_n \Phi$，首先要分析最外层的 x_1，因为它约束着后面较低的层次，然后再考虑 x_2，以此类推，逐层进行求解；而命题公式如按 QBF 表示为 $\exists x_1 \exists x_2 \cdots \exists x_n \Phi$，那么 Φ 中出现的所有变量 $x_1 x_2 \cdots x_n$ 都只是受∃量词约束，简写成 $x_1 x_2 \cdots x_n$，那么对于 $x_1 x_2 \cdots x_n$ 的讨论顺序没有限制，较为灵活。

在逻辑推理角度，中山大学的熊明辉等人对目前 QBF 以及 SAT 的发展情况进行了总结。首先，理论方面，QBF 广泛应用于各种基于逻辑的问题解决及计算中；应用方面，现有的 SAT 系统已经可以解决上百万变元的公式验证问题。而 QBF 系统由于本身的高度计算复杂性，当前研究进展相对较慢，同时也具有比较大的发展空间。

目前，QBF 和 SAT 技术被广泛地应用到人工智能的多个领域。它们在解决问题的方式上有很多相似之处，也各有利弊。比如，在本文谈论的智能规划方面，利用 SAT 求解经典规划问题是具有优势的。多年来，研究者以 SAT 为主导思想提出了许多优秀的经典规划编码方式和规划系统。但当问题面向更复杂的不确定性规划时，单纯地基于 SAT 的转化便不能满足实际需要。相反，QBF 以其复杂的结构却能够成功地转换相应的问题。从文中也不难看出，如同动作效果的不确定性，只含有存在变量的命题公式是无法进行描述的。

结束语 目前，以逻辑方式解决不确定性规划问题的方法，大多是以 QBF 为基础的。它们或者直接使用 QBF 的标准形式，或者间接利用了 QBF 的语义思想。对于不确定性规划问题，基于 QBF 的表述方式和构造过程有着非常重要的地位。

本文着重介绍了 3 种标准形式的 QBF 编码，并对这 3 种不同前缀结构进行了分析和比较。对于不同类型的规划问题，前缀数为 3 和 2 的几种编码分别体现出各自的优势。综合这几种编码，其主要优势在于灵活性，它们可以应用于任何形式的 QBF 求解器。对于不同的求解器，这 3 种 QBF 编码会有不同的执行效果。如果能够创建与相应 QBF 编码足够切合的求解器，那么 QBF 规划系统的效率将会是非常可观的。

值得一提的是，对于 QBF 求解器，其中由 QBF 向 CNF 范式的转化过程占用了很大一部分时间。2008 年，Horian Lonsing 提出了一种以否定范式(NNF)的形式表示公式的 QBF 求解器[22]。与以往转换成 CNF 相比，在最坏的情况下，利用 CNF 范式删除存在变量，会带来平方级的公式增量，而这种 NNF 范式只是线性增加 QBF 公式的长度。实验结果表明，该求解器与基于 CNF 的求解器相比，空间需求显著减少。

(下转第 281 页)

formation and its application in the image information covering [J]. Science in China (Series E),2000,43(3):304-412
[3] 王泽辉.二维随机矩阵置乱变换的周期及在图像信息隐藏中的应用[J].计算机学报,2006,29(12):2218-2224
[4] 杨军,覃伯平,雷开彬,等.基于广义猫映射的组播密钥管理方案研究[J].计算机科学,2008,35(1):80-82
[5] 邵利平,覃征,刘波,等.二维双尺度矩形映射及其在图像置乱上的应用[J].计算机辅助设计与图形学学报,2009,21(7):1025-1033
[6] Arnold V I, Avez A. Ergodic Problems of Classical Mechanics [M]// Mathematical Physics Monograph Series. NewYork:W A Benjamin,INC,1968
[7] 李用江,李昌利,李司东,等. Fibonacci 数列模 p^r 的周期性研究[J].数学的实践与认识,2009,39(17):138-143
[8] 马在光,丘水生.基于广义猫映射的一种图像加密系统[J].通信学报,2003,24(2):51-57
[9] 朱桂斌,曹长修,胡中豫,等.基于仿射变换的数字图像置乱加密算法[J].计算机辅助设计与图形学学报,2003,15(6):711-715
[10] Garrett P. 密码学导论[M].吴世忠,等译.北京:机械工业出版社,2003:94-100,154

(上接第 256 页)
但针对不同的 QBF 公式所得结果并不稳定,对于某些问题仍然会花费较长时间,但这种非 CNF 的思想是值得深入研究的。

基于 QBF 的规划方法的求解性能依赖于 QBF 编码的构造和求解两个方面,不同 QBF 编码的选用很大程度上依赖于对应的 QBF 求解器的性能。可以考虑两种解决策略:设计与具体求解器完美匹配的有效编码或者设计求解具体编码的最佳性能的 QBF 求解器。显然,区别于 QBF 求解技术的研究,有效的 QBF 编码设计是基于转换的规划方法的核心问题。随着各种复杂规划问题的提出,并进一步参与规划系统性能的衡量,迫切要求为其设计新的对应编码方式,使得日趋成熟的 QBF 技术得以在智能规划领域中得到利用。

参考文献

[1] Peot M A,Smith D E. Conditional nonlinear planning[C]// Hendler J,ed. Proceedings of the First International Conference on Artificial Intelligence Planning Systems. 1992:189-197
[2] Pryor L,Collins G. Planning for contingencies: A decision-based approach[J]. Journal of Artificial Intelligence Research, 1996 (4):287-339
[3] Cimatti A, Roveri M, Traverso P. Automatic OBDD - based generation of universal plans in non-deterministic domains[C]// Proceedings of the Fifteenth National Conference on Artificial Intelligence(AAAI-98) and the Tenth Conference on Innovative Applications of Artificial Intelligence (IAAI-98). 1998:875-881
[4] Bonet B, Geffner H. Planning with incomplete information as heuristic search in belief space[C]// Chien S,Kambhampati S, Knoblock C A,eds. Proceedings of the Fifth International Conference on Artificial Intelligence Planning Systems. 2000:52-61
[5] Kautz H,Selman B. Planning as satisfiability[C]// Neumann B, ed. Proceedings of the 10th European Conference on Artificial Intelligence. 1992:359-363
[6] Kautz H,Selman B. Pushing the envelope: planning, propositional logic,and stochastic search[C]// Proceedings of the 13th National Conference on Artificial Intelligence and the 8th Innovative Applications of Artificial Intelligence Conference. 1996: 1194-1201
[7] Kautz H,McAllester D,Selman B. Encoding plans in prepositional logic[C]// Proc. 5th International Conference of Principles of Knowledge Representation and Reasoning. 1996
[8] Rintanen J. Constructing conditional plans by a theorem-prover [J]. Journal of Artificial Intelligence Research, 1999(10): 323-352
[9] Giunchiglia E,Lifschitz V. An action language based on causal explanation:Preliminary report[C]// Proceedings of The Fifteenth National Conference on Artificial Intelligence. 1998:623-630
[10] Giunchiglia E. Planning as satisfiability with expressiveaction languages:concurrency, constraints and nondeterminism[C]// Cohn A G,Giunchiglia F,Selman B,eds. Proceedings of the Seventh International Conference of Principles of Knowledge Representation and Reasoning. 2000:657-666
[11] Brafman R I, Hoffmann J. Conformant planning via heuristic forward search: A new approach[C]// Proceedings of the 14th International Conference on Automated Planning and Scheduling. 2004:355-364
[12] Palacios H,Geffner H. Reducción de la planificación conformante a SAT mediante compilación a d-DNNF[C]// 11th Conferencia de la Asociación Espanola para la Inteligencia Artificial; an ICAPS' 05 Workshop as Mapping Conformant Planning into SAT Through Compilation and Projection. 2005
[13] Palacios H, Bonet B, Darwiche A, et al. Pruning conformant plans by counting models on compiled d-DNNF representations [C]// Biundo S,Myers K,Rajan K,eds. Proceedings of the Fifteenth International Conference on Automated Planning and Scheduling(ICAPS 2005). 2005:141-150
[14] Rintanen J, Heljanko K, Niemelä I. Planning as satisfiability: parallel plans and algorithms for plan search[J]. Artificial Intelligence,2006,170(12/13):1031-1080
[15] Rintanen J. Asymptotically optimal encodings of conformant planning in QBF[C]// Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI-07). 2007:1045-1050
[16] Balcázar J L,Díaz I,Gabarró J. Structural Complexity I[M]. Berlin:Springer-Verlag,1995
[17] Giunchiglia E, Narizzano M, Tacchella A. Backjumping for Quantified Boolean Logic Satisfiability [J]. Artificial Intelligence, 2003,145(1/2):99-120
[18] Giunchiglia E, Narizzano M, Tacchella A. Learning for quantified Boolean logic satisfiability[C]// Proceedings of the 18th National Conference on Artificial Intelligence (AAAI-2002) and the 14th Conference on Innovative Applications of Artificial Intelligence (IAAI-2002). 2002:649-654
[19] Biere A. Resolve and expand: Theory and Applications of Satisfiability Testing[C]// 7th International Conference SAT. Vancouver,BC,Canada,May 2004
[20] Zhang L, Malik S. Conflict driven learning in a quantified Boolean satisfiability solver[C]// Proceedings of the 2002 IEEE/ ACM International Conference on Computer Aided Design (ICCAD2002). 2002:442-448
[21] Stockmeyer L J,Meyer A R. Word problems requiring exponential time[J]. Journal of the ACM,1973:1-9
[22] Florian L, Armin B. Expanding NNF for QBF Solving[C]// SAT 2008. Vol. 4996. Springer,2008:196-210