

最大公共子图的约束符号求解方法^{*}

Symbolic Algorithm of Constraint Problem for Maximum Common Sub Graph Problem

刘桂珍, 徐周波, 唐浩

LIU Guizhen, XU Zhoubo, TANG Hao

(桂林电子科技大学, 广西可信软件重点实验室, 广西桂林 541004)

(Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin, Guangxi, 541004, China)

摘要:【目的】探索求解两个图最大公共子图的方法。【方法】建立最大公共导出子图的软约束满足问题(Soft CSP)模型, 提出代数决策图(ADD)的符号求解算法。首先, 分别对两个图中的变量和值域进行编码, 完成两个图的 ADD 表示; 其次, 基于深度优先分支定界算法的思想, 利用符号 ADD 的相关操作, 实现对最大公共导出子图的求解。【结果】算例结果表明, 该方法准确可行。【结论】该方法能有效缩减搜索空间, 从而提高问题的求解效率。

关键词:最大公共子图 软约束满足问题 全局约束 ADD

中图分类号: TP301.6 文献标识码: A 文章编号: 1002-7378(2017)01-0025-07

Abstract:【Objective】To explore the methods to solving the maximum common sub graph of two graphs.【Methods】A soft CSP model for the maximum common induced sub graph is constructed, and the symbolic ADD algorithm is proposed. Firstly, the variables and domains of the two graphs are encoded, respectively, and then the two graphs are expressed by ADD. Secondly, based on the depth first branch and bound algorithm, the related operations of symbolic ADD technology are used, and then the maximum common induced sub graph is solved.【Results】The result shows that the method is accurate and feasible.【Conclusion】This approach can reduce the search space effectively, thus improving the efficiency of solving the problem.

Key words: maximum common induced sub graph, soft constraint satisfaction problem, global constraint, ADD

0 引言

【研究意义】图是一种用来描述事物之间特定关系的重要数学模型和数据结构, 广泛应用于生活中的各个领域, 例如万维网、社交网络、蛋白质交互网络、化学分子结构等^[1]。随着图数据规模的不断增大, 对其进行有效和快速的分析和处理仍然面临着严峻的挑战, 而图模式匹配技术(Graph Pattern Matching)作为实现图数据上高效查询的重要方法, 已成为国内外学者的研究热点。【前人研究进展】从

收稿日期: 2017-01-10

作者简介: 刘桂珍(1991—), 女, 硕士研究生, 主要从事符号计算、约束求解和图模式匹配研究, E-mail: 18877338387@126.com。

* 广西自然科学基金项目(2015GXNSFAA139285, 2014GXNSFAA118354)和桂林电子科技大学研究生创新项目(2016YJCX62)资助。

匹配结果与模式图是否完全一致的角度看,图匹配问题可以分为精确匹配和非精确匹配,其中,精确匹配一般通过定义图同构和子图同构来分析数据图和模式图之间的关系;非精确图匹配一般通过定义编辑距离、最小公共超图和最大公共子图来衡量两个图之间的相似程度^[1]。研究者对最大公共子图问题进行了大量的研究,1973年,Levi^[2]描述了一种求解最大公共子图的方法;1982年,Mcgregor^[3]提出最大公共子图的回溯搜索算法;2004年,Krissinel等^[4]针对公共子图同构,提出一种有更好搜索策略的基于回溯的树搜索算法;2005年,Suters等^[5]提出一种基于最大团问题的“团分支”算法;2007年,Abu-khazam等^[6]提出带有回溯策略的顶点覆盖和最大独立集相结合的技术;2016年,Nirmala等^[7]在顶点覆盖的基础上提出二元搜索树的算法,能够更快地求解基于点的最大公共子图;2011年,Ndiaye等^[8]首次提出结合约束满足问题(CSP)的求解思想,建立最大公共子图的软约束模型,利用约束传播等方法对模型进行求解,后来Minot等^[9]在这种软约束模型的基础上,利用相容图和独立子图的思想完成最大公共子图的求解。Ndiaye等^[8]和Minot等^[9]证明引入约束满足问题模型可以清晰的刻画两个图的信息以及对应关系,改善了最大公共子图的求解效率。约束满足问题是人工智能领域的一个重要研究课题,它在复杂调度、资源分配、优化排序、模式识别与规划等领域都有广泛的应用^[10]。其目标是寻找能满足所有约束条件的解,然而,在求解经典约束满足问题时,会因为约束限制过于严格而使得整个问题无解,所以,为了满足实际需要,研究者对经典CSP进行了拓展,提出了软约束满足问题(Soft SCP),将CSP的求解目标转化为对Soft CSP求解。Soft CSP是在经典CSP的基础上,给CSP中的每个约束附加一个权值,该权值表示约束的优先级、满足程度等,通过放松一些约束条件,就可能找到问题的最优解,弥补了经典CSP的不足^[11]。

【本研究切入点】在Soft CSP中引入代数决策图(ADD)技术,可以用于求解最大公共子图。**【拟解决的关键问题】**在Ndiaye等^[8]和Minot等^[9]提出的软约束模型基础上,引入了符号ADD求解方法,并结合深度优先分支定界算法,实现对最大公共子图问题的求解。

1 预备知识

1.1 软约束满足问题

软约束满足问题一般可以定义为一个四元组 $\langle X, D, C, \lambda \rangle$,其中:

- (1) $X = \{x_1, x_2, \dots, x_n\}$ 为 n 个变量的有限集;
- (2) $D = \{D(x_1), D(x_2), \dots, D(x_n)\}$ 为 n 个变量的值域集合; D_i 为变量 X_i 的值域, $i = \{1, 2, \dots, n\}$;
- (3) $C = \{c_1, c_2, \dots, c_m\}$ 为约束关系集, $c_i(x_{i_1}, x_{i_2}, \dots, x_{i_j}) \subseteq D(x_{i_1}) \times D(x_{i_2}) \times \dots \times D(x_{i_j}), i = 1, 2, \dots, m, 1 \leq j \leq n, v_{ij} \in X (l = 1, 2, \dots, j), D_{ij}$ 为变量 x_{ij} 的域, 称 x_i 为定义在变量集 $\{x_{i_1}, x_{i_2}, \dots, x_{i_j}\} \subseteq X$ 上的 j 元约束。若 C 所有的约束均为一元或者二元约束, 则称该 Soft CSP 为二元 Soft CSP。本文仅讨论二元 Soft CSP, 对于二元 CSP 中的一组变量 $\{x_1, x_2, \dots, x_n\}$ 的一个完全赋值。

λ 表示在匹配过程中不匹配节点所产生的代价值, 这里指不匹配的节点个数。若二元 Soft CSP 中的一组变量 $\{x_1, x_2, \dots, x_i\} \subseteq X, t = \langle v_1, v_2, \dots, v_i \rangle \in D(x_1) \times D(x_2) \times \dots \times D(x_i), 1 \leq i \leq n$, 其中 v_i 为变量 x_i 的取值。当 $t = n$ 时, 称赋值 t 是 Soft CSP 的一个完全实例化, 否则为部分实例化, 将总约束代价值表示为 $\text{cost}(t)$, 则有约束的约束值表示 $\text{cost}(t) = \lambda$ 。Soft CSP 求解的目标是找到变量集 X 的一种完全实例化 t , 从而使得 $\text{cost}(t)$ 最小。

1.2 代数决策图(ADD)

有序二叉决策图(OBDD)是布尔函数的一种有效图形和数学描述技术^[12-13]。在OBDD的基础上, Bahar等^[14]通过引入多个非布尔终节点, 将OBDD的布尔域扩展到有限域, 给出了ADD的相关定义和基本操作, 极大地改善了伪布尔函数和有限域取值函数的描述能力。

定义 1.1 任意一个自变量取值只能是 0 或 1, 函数值可为实数的 n 元函数, 称为 n 元伪布尔函数。如果实数集为 $R, B = \{0, 1\}$, 那么 n 元伪布尔函数可以表示为 $f: B_n \rightarrow R$ 。

定义 1.2 对于从 $\{0, 1\}^n$ 到 S 的伪布尔函数 $f(x_1, x_2, \dots, x_n)$, 它的伪布尔函数族是指在布尔变量 x_1, x_2, \dots, x_n 的不同输入模式下的伪布尔函数构成的一个伪布尔函数的集合, 记为 $\#f(x_1, x_2, \dots, x_n)$ 。

定义 1.3 对于从 $\{0, 1\}^n$ 到 S 的伪布尔函数 $f(x_1, x_2, \dots, x_n)$, ADD 是一个有向无环图, 它用来表示伪布尔函数族 $\#f(x_1, x_2, \dots, x_n)$, 满足:

(1) S 为 ADD 代数结构的有限值域,且 $S \subseteq Z$ (整数集合)。

(2) 节点分为根节点、终止点和内部节点 3 类。终止点集合为 T 。对于 $\forall t \in T, s(t)$ 均被标识为值域 S 中的一个元素。

(3) 每个非终止点都满足四元组属性 $(f^u, \text{var}, \text{left}, \text{right})$, f^u 表示非终止点 u 所对应的伪布尔函数, var 表示非终止点 u 标记变量; 标识节点 u 的 $\text{var}=0$ 时, 对应 0-分支子节点和连接弧 0-边; high 标识节点 u 的 $\text{var}=1$ 时, 多对应 1-分支子节点和连接弧 1-边。

(4) 每个节点都对应唯一的函数 f^u 。

(5) 在给定变量序 x_1, x_2, \dots, x_n 下, 任何一条有向路径上, 每个变量都必须依照变量序所限定的次序至少出现一次。

例如, 伪布尔函数 $f(x_1, x_2, x_3, x_4) = x_1x_2 + 2x_3x_4$, 在变量序 $\pi: x_1 < x_2 < x_3 < x_4$ 的 ADD 如图 1 所示。

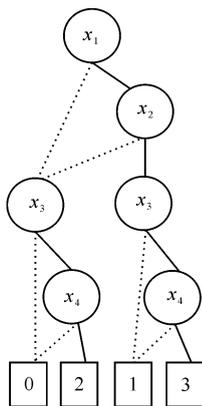


图 1 伪布尔函数 $f(x_1, x_2, x_3, x_4) = x_1x_2 + 2x_3x_4$ 的 ADD 表示

Fig. 1 ADD for pseudo-Boolean functions $f(x_1, x_2, x_3, x_4) = x_1x_2 + 2x_3x_4$

2 最大公共导出子图 (MCIS) 的符号 ADD 算法

2.1 MCIS 描述

图是由顶点集和顶点间的二元关系集合(边的集合或弧的集合)组成的数据结构, 通常可以用 $G = \langle V, E \rangle$ 来表示, V 和 E 分别表示图的顶点集与边集。

两个图的公共子图通常用来定义两个图之间的距离, 而最大公共子图是通过删除节点个数或者边数得到, 最大公共子图分为基于边的最大部分子图和基于点的最大公共导出子图 (MCIS)。本研究主要研究基于点的最大公共导出子图, 在图中寻找两

个图之间所包含的最大节点数的公共子图, 即为最大公共导出子图 (MCIS)。

定义 2.1 已知两个图结构 G 和 G' , 如果存在另外一个图结构 g , 满足 $g \subseteq G, g \subseteq G'$, 并且不存在图结构 $g' (g' \subseteq G, g' \subseteq G')$, 使得 $|g'| \geq |g|$, 则称 g 是图 G 和 G' 的最大公共子图, 记为 $\text{MCIS}(G, G')$ 。

如图 2 所示定义两个图 G 和 G' , 它们的最大公共子图^[15] 为 $g = \text{MCIS}(G, G')$ 。

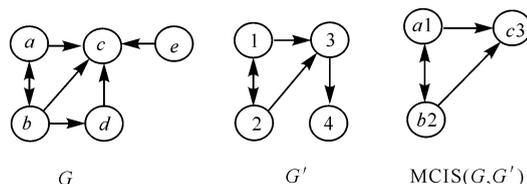


图 2 图 G, G' 和 $\text{MCIS}(G, G')$

Fig. 2 The diagrams of G, G' and $\text{MCIS}(G, G')$

2.2 全局约束

在最大公共子图求解过程中, 为提高求解效率, 引入了全局约束 all Different, 目的是对不同顶点的所有可能的赋值进行约束。由于每条边的二进制约束只包含一对不同的顶点, 不存在两个不同的顶点有相同的赋值, 所以在全局约束的限制条件下, 能更加有效地缩减搜索空间。

定义 2.2 对变量 x_1, x_2, \dots, x_n , 有

$$\text{all Different}(x_1, \dots, x_n) = \{(d_1, \dots, d_n) \mid \forall d_i \in D(x_i), \forall i \neq j, d_i \neq d_j\}.$$

从语法看, 定义一个全局约束 all Diff $\langle V, E, V', E', L \rangle$, 则有

(1) 两个变量集: V, V' 。

(2) 两个边集: $E \subseteq V \times V$ 和 $E' \subseteq V' \times V'$ 。

(3) L 是一组值对 (x_u, u) , 即任意一个顶点在值域中, 变量 x' 所对应的取值, 同时, L 中任意两个不同的值对 (x_u, u) 和 (x_v, v) , 满足 $x_u \neq x_v, u \neq v$ 。

那么从语义上分析, 该约束是全局一致性的, 比如: 对于任意一个值对 $(x_u, u) \in L$, 如果存在 $u' \in D(x_u)$, 那么 $u' = f(u)$ 。

2.3 MCIS 的 Soft CSP 符号模型

2.3.1 MCIS 的 Soft CSP 模型

给定两个图, 分别为数据图 $G = (V, E)$ 和模式图 $G' = (V', E')$, 其中 V 和 V' 分别表示两个图中节点的集合, E, E' 分别表示两个图中边的集合。最大公共子图要求图 G 中的变量在 G' 中都有相应的赋值, 同时, 相匹配的节点都保持节点之间的邻接关系(边关系)。所以给定图 G 中任意两个节点 u 和 v , 所

对应的变量分别 x_u 和 x_v , 它们的值域分别为 $D(x_u)$ 和 $D(x_v)$ 。此外定义一个值 φ , 表示两图中不匹配的节点的值, 则对于 MCIS 进行 Soft CSP 建模如下:

定义一个四元组 Soft CSP $\langle X, D, C, \lambda \rangle$, 则有

(1) 任意顶点 u 所对应的变量 x_u 满足: $u \in V$, 即 $X = \{x_u \mid u \in V\} \cup \{x_\varphi\}$;

(2) 变量 x_u 的值域满足:

$D(x_\varphi) = \{\varphi\}, \forall u \in V, D(x_u) = \{u' \in V'\} \cup \{\varphi\}$ 。

(3) 给定任意两个变量 x_u 和 x_v , 利用分层约束将约束划分为

硬约束: $\forall \{u, v\} \subseteq V$,

$C_{\text{edge}}(x_u, x_v) \equiv (x_u = \varphi) \vee (x_v = \varphi) \vee ((u, v) \in E \Leftrightarrow (x_u, x_v) \in E')$ 。

软约束: 全局约束 all Diff $\langle V, E, V', E', L \rangle$ 。

根据 Soft CSP 的定义, 对不满足约束条件的变量和赋值, 定义一个代价值为 $\text{cost}(t)$, 表示不匹配的节点所产生的代价, 最终目标是在满足约束的条件下使求得的代价值最小(节点数最多), 得到 Soft CSP 的最优解。

2.3.2 Soft CSP 的符号 ADD 描述

给定一个二元 Soft CSP $P = \langle X, D, C, \lambda \rangle$, 为后续表述方便, 变量集 X 中有 n 个变量, 分别用变量下标 $1, 2, \dots, n$ 来表示变量, 即 $|X| = n$ 在变量的值域集 D 中有 $d = \max\{|D(x_i)| \mid 1 \leq i \leq n\}$ 。

在 Soft CSP 的符号 ADD 表示中, 可以用二进制向量 $X = (x_0, \dots, x_{l-1})$ 表示变量集 X 中的 n 个变量; 用二进制向量 $V = (v_0, \dots, v_{m-1})$ 来表示每个变量域 D_i 中的任意值, 其中: $l = \lceil \log^n \rceil, m = \lceil \log^d \rceil, x_i, v_j \in \{0, 1\}, i = 0, 1, \dots, l-1, j = 0, 1, \dots, m-1$ 。用二进制向量 $(X, V) = (x_0, \dots, x_{l-1}, v_0, \dots, v_{m-1})$ 来表示 Soft CSP 中的任意一元约束 c_i , 其中 $(X, V) = (x_0, \dots, x_{l-1})$ 和 $V = (v_0, \dots, v_{m-1})$ 分别是变量 x_i 和变量 x_i 在值域 $D(x_i)$ 中赋值的二进制向量表示。用二进制向量 $(X, Y, V, W) = (x_0, \dots, x_{l-1}, y_0, \dots, y_{l-1}, v_0, \dots, v_{k-1}, w_0, \dots, w_{k-1})$ 表示 Soft CSP 中的任意二元约束 c_{ij} , 其中 $Y = (y_0, \dots, y_{l-1}), W = (w_0, \dots, w_{k-1})$ 分别是变量 x_j 和变量 x_j 在值域 $D(x_j)$ 中赋值的二进制向量表示。因此一元和二元约束可以表示为如下的特征函数:

$$C(X, Y, V, W) =$$

$$\begin{cases} \lambda(x_0 \wedge \dots \wedge x_{l-1} \wedge y_0, \dots \wedge y_{l-1} \wedge v_0 \wedge \dots \wedge v_{l-1} \\ \wedge w_0, \dots \wedge w_{m-1}), C_{ij} \in C, d_i \in D_i, d_j \in D_j, \\ d_i, d_j \in D \\ \lambda(x_0 \wedge \dots \wedge x_{l-1} \wedge v_0, \dots \wedge v_{m-1}), C_{ij} \in C, \\ d_i \in D_i, d_j \in D_j, \\ \perp, C_{ij} \in C, d_i \notin D_i \text{ 或 } d_j \notin D_j \text{ 或 } d_i, d_j \notin D, \\ T, C_{ij} \in C, d_i \notin D_i \text{ 或 } d_j \notin D_j, \\ 0, \text{ 否则,} \end{cases}$$

其中 λ 是一元和二元约束的代价值。在特征函数中, 若变量 i 的值不在域 D_i 或者 D 中, 则设 $\lambda = T$, 特征函数可由 ADD 来表示。

给定数据图 G 和模式图 G' , 分别用伪布尔函数表示如下图 3。图 2 中的图 G , 使用如下节点编码:

$a:000, b:001, c:010, d:100, e:110$,

其对应的伪布尔函数为

$$G(x, y) = x_0'x_1'x_2'y_0'y_1'y_2 + x_0'x_1'x_2y_0'y_1'y_2' + x_0'x_1'x_2'y_0'y_1'y_2' + x_0'x_1x_2y_0'y_1'y_2' + x_0x_1x_2'y_0'y_1'y_2'$$

在变量序 $\pi: x_0 < x_1 < x_2 < y_0 < y_1 < y_2$ 的前提下, 根据伪布尔函数的表示, 可以将 G 图中约束集的 ADD 表示为图 3。

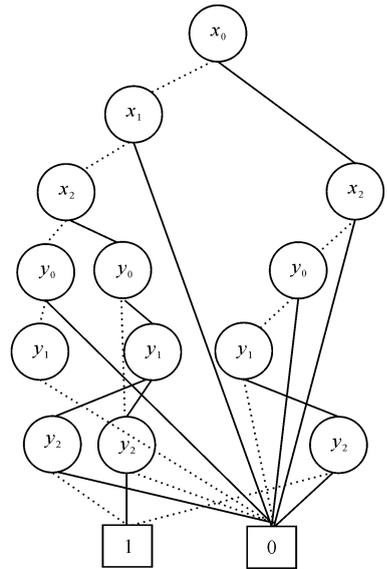


图 3 G 图中约束集的 ADD 表示

Fig. 3 ADD formulation of constraint sets in figure G

同样对图 3 的模式图 G' 使用如下节点编码

1:00 2:01 3:10 4:11

其对应的伪布尔函数为

$$G'(v, w) = v_0'v_1'w_0'w_1 + v_0'v_1'w_0w_1' + v_0'v_1w_0w_1' + v_0v_1'w_0w_1 + v_0'v_1w_0'w_1'$$

在变量序 $\pi: v_0 < v_1 < w_0 < w_1$ 的前提下, 根据上述的伪布尔函数, 可以将 G' 图中约束集的

ADD 表示为图 4。

2.4 MCIS 的约束求解符号算法

运用深度优先分支定界法 (DFBB) 结合 ADD 符号算法对 Soft CSP 进行求解。DFBB 算法的基本思想是按照变量实例化顺序, 依次对问题中变量进行实例化, 在搜索过程中保存最优完全实例化 t , 此时最优代价用 $\text{cost}(t)$ 表示。同时, 在每次实例化一个变量时, 比较当前赋值的总代价和最优代价, 如果当前赋值代价小于最优代价 $\text{cost}(t)$, 则继续下一个变量, 否则选择该变量值域中的其他值来进行实例化。算法会有回溯的过程, 即对于当前变量值域为空时, 表示由部分实例化 t 扩展不能得到最优解, 执行回溯, 对上一层进行实例化, 并更新下界值, 直至找到最优解。

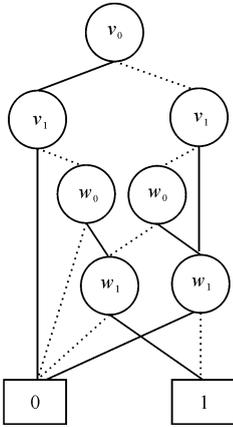


图 4 图 G' 中约束集的 ADD 表示

Fig. 4 ADD formulation of constraint sets in figure G'

根据 MCIS 的 Soft CSP 模型, 最大公共导出子图求得的是节点数最多的公共子图, 所以, 根据模型定义一个节点不匹配数 $\lambda(\varphi)$ 。在 DFBB 搜索过程中, 对每个变量进行依次赋值时, 如果出现不满足约束 (不匹配的节点), 则记为 $\lambda(\varphi) = \text{cost}(t) = 1$, 否则记为 $\lambda(\varphi) = \text{cost}(t) = 0$ 。同时, 求解过程中, 利用深度优先搜索对变量进行赋值时, 只要出现不匹配的节点, 用 φ 表示, 即表示该变量的取值为 φ 。然后继续对下一个变量进行实例化, 直至对所有空间搜索完毕, 算法执行结束, 得到一组或者多组解, 找到匹配节点数对最多的解, 即 $\lambda(\varphi)$ 值最小的解。基于 DFBB 的符号 ADD 算法的流程如图 5 所示。

2.5 实例分析

基于 DFBB 的基本思路, 结合 ADD 符号算法, 对 MCIS 的 Soft CSP 模型进行求解, 同时为了验证该算法的正确性和可行性, 运用实例进行分析, 求解的主要步骤如下:

(1) 对问题进行建模, 给定 MCIS 的 Soft CSP

模型 $\text{Soft CSP}\langle X, D, C, \lambda \rangle$ 。设定变量选取顺序, 构建变量序 π , 并将约束集表示为符号 ADD 形式。

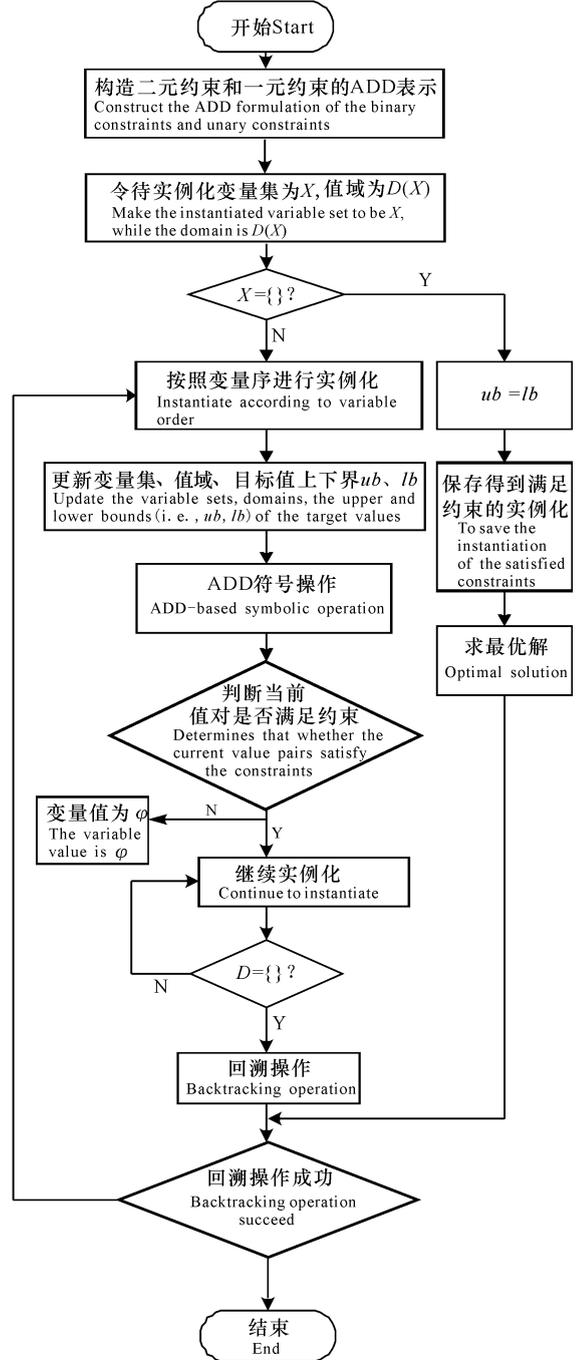


图 5 基于 DFBB 的符号 ADD 算法流程

Fig. 5 A flow diagram of symbolic ADD algorithm based on DFBB

(2) 采用基于 DFBB 算法的 ADD 符号技术进行求解, 对于上述两图的变量集、值域集和约束集分别表示为

① 变量集:

$$X = \{x_0'x_1'x_2', x_0'x_1'x_2, x_0'x_1x_2', x_0x_1'x_2', x_0x_1x_2'\} \cup \{\varphi\}.$$

② 值域集:

$$D = \{v_0'v_1', v_0'v_1, v_0v_1', v_0v_1\} \cup \{\varphi\}.$$

③ 约束集表示为 $G(x, y)$ 和 $G'(v, w)$ 。

在 ADD 算法中, Soft CSP 的初始目标值上界用 ub 表示, 且有 $ub = T$, 下界用 lb 表示, 且 $lb = \perp$ 。按照变量实例化顺序, 对数据图中的顶点变量 $\{a, b, c, d, e\}$ 依次进行实例化。

首先赋值 $t = \{\}$, 则 $lb = 0, ub = 5$, 从 X 中选择一个待实例化变量 $x_0'x_1'x_2'$ (变量 x_a), 并从 X 中删去变量 $x_0'x_1'x_2'$, 并进行赋值。当 $x_a = v_0'v_1'$ 时, 此时更新待实例化变量集和当前值域, 则有

$$\text{变量集: } X_1 = \{x_0'x_1'x_2, x_0'x_1x_2', x_0x_1'x_2', x_0x_1x_2'\} \cup \{x_\varphi\}; \text{值域: } D_1 = \{v_0'v_1, v_0v_1', v_0v_1\} \cup \{\varphi\};$$

继续对 x_b 进行实例化, $x_b = v_0'v_1$, 则此时

$$\text{变量集: } X_2 = \{x_0'x_1x_2', x_0x_1'x_2', x_0x_1x_2'\} \cup \{\varphi\}; \text{值域集 } D_2 = \{v_0v_1', v_0v_1\} \cup \{\varphi\};$$

此时判断以上取值是否满足约束条件, $C(x_a, x_b) = \{v_0'v_1'v_2'v_0'v_1'v_2\}$ 。根据符号 ADD 操作规则, 判断 (x_a, x_b) 是否满足边约束, 由如下所示的符号 ADD 操作来实现:

$$a \rightarrow b \text{ 边约束为 } C(x_a, x_b) = x_0'x_1'x_2'y_0'y_1'y_2.$$

$$b \rightarrow a \text{ 边约束为 } C(x_b, x_a) = x_0'x_1'x_2y_0'y_1'y_2'.$$

$$1 \rightarrow 2 \text{ 边约束为 } C(1, 2) = v_0'v_1'w_0'w_1.$$

判断是否满足约束条件, 将边 $a \rightarrow b$ 约束与图 G 中的约束集 $G(x, y)_{\text{ADD}}$ 进行 ADD 符号操作中的 \wedge 操作 (合取操作), 用 F 表示得到的约束图。判断得到的结果, 如果结果为相应的边约束的 ADD 表示, 那么, 该变量的赋值满足 Soft CSP 中的强约束 C_{edge} , 例如:

$$\text{边 } a \rightarrow b: F(a \rightarrow b) = x_0'x_1'x_2'y_0'y_1'y_2 \wedge G(x, y)_{\text{ADD}} = x_0'x_1'x_2'y_0'y_1'y_2;$$

$$\text{边 } b \rightarrow a: F(b \rightarrow a) = x_0'x_1'x_2y_0'y_1'y_2' \wedge G(x, y)_{\text{ADD}} = x_0'x_1'x_2'y_0'y_1'y_2';$$

$$\text{边 } 1 \rightarrow 2: F(1 \rightarrow 2) = v_0'v_1'v_2'v_0'v_1'v_2 \wedge G'_{\text{ADD}} = v_0'v_1'v_2'v_0'v_1'v_2.$$

通过操作得知, $\langle x_a, 1 \rangle$ 和 $\langle x_b, 2 \rangle$ 满足约束条件, 更新目标值的上界和下界, $lb = 0$ 小于 ub , 继续对变量 x_c, x_d 进行赋值, $x_c = v_0v_1'$, 则有变量集: $X_3 = \{x_0x_1'x_2', x_0x_1x_2'\} \cup \{\varphi\}$ 。值域集 $D_2 = \{v_0v_1\} \cup \{\varphi\}$ 。

同样对边 $a \rightarrow c, b \rightarrow c, 1 \rightarrow 3, 2 \rightarrow 3$ 进行约束

条件判断, 结果表明 $\langle x_c, 3 \rangle$ 满足约束条件。

当 $x_d = v_0v_1$ 时, 变量集: $X_4 = \{x_0x_1x_2'\} \cup \{\varphi\}$ 。值域集 $D_2 = \{\} \cup \{\varphi\}$ 。

同样对边 $a \rightarrow c, b \rightarrow c, 1 \rightarrow 3, 2 \rightarrow 3$ 进行约束条件判断, 结果表明 $\langle x_d, 4 \rangle$ 不满足约束条件, 则 $x_d = \varphi$ 。同理对 x_e 进行赋值, 可以得到 $x_e = \varphi$, 此时已经对变量全部实例化, 得到 d, e 即为不匹配的节点, 更新目标值的上界和下界, 分别为 $lb = ub = 2$ 。求解得到的一组解为 $\{x_a = 1, x_b = 2, x_c = 2, x_d = \varphi, x_e = \varphi\}$, 此时约束代价值为 $\lambda(\varphi) = 2$ 。根据 DFBB 算法, 可以得知当前值域为空, 算法进行回溯, 继续对变量重新进行赋值。在这个过程中, Soft CSP 的目标值不断更新, 最终求得的每组解, 都会产生一个代价值。通过比较, 取 $\lambda(\varphi)$ 值最小的解为问题的最优解, 即求得两图之间的最大公共子图。该算例中最小代价值为 $\lambda(\varphi) = 2$, 即最大匹配节点数为 3, 所以问题的最优解为 $\{x_a = 1, x_b = 2, x_c = 3\}$, 从而求得最大公导出子图。

3 结论

从理论上分析, 一方面约束满足模型能够充分的对问题的结构信息、约束关系进行分析和刻画。另一方面 ADD 符号技术极大地改善了伪布尔函数和有限域取值的函数的描述能力, 具有高效紧凑和容易操作的优点, 减少问题的空间需求, 能够减缓状态爆炸问题。

基于此, 本研究对图结构进行的 ADD 刻画, 实现了以较小的空间存储一定规模的有向图结构。求解过程中, 在 Soft CSP 建模中引入边和全局约束等约束条件, 对变量和相应的取值进行约束, 同时采用 ADD 符号技术和深度优先分支定界法的思想, 使得在算法能有效缩减搜索空间, 从而提高问题的求解效率, 最终求得最大公导出子图。

参考文献:

- [1] 于静, 刘燕兵, 张宇, 等. 大规模图数据匹配技术综述 [J]. 计算机研究与发展, 2015, 52(2): 391-409.
YU J, LIU Y B, ZHANG Y, et al. Survey on large-scale graph pattern matching [J]. Journal of Computer Research and Development, 2015, 52(2): 391-409.
- [2] LEVI G. A note on the derivation of maximal common subgraphs of two directed or undirected graphs [J]. Calcolo, 1973, 9(4): 341-352.
- [3] MCGREGOR J J. Backtrack search algorithms and the maximal common subgraph problem [J]. Software

- Practice and Experience,1982,12(1):23-34.
- [4] KRISSEL E B, HENRICK K. Common subgraph isomorphism detection by backtracking search[J]. Software Practice and Experience,2004,34(6):591-607.
- [5] SUTERS W H, ABU-KHZAM F N, ZHANG Y, et al. A new approach and faster exact methods for the maximum common subgraph problem[M]//WANG L S. Computing and combinatorics. Berlin Heidelberg: Springer,2005.
- [6] ABU-KHZAM F N, SAMATOVA N F, RIZK M A, et al. The maximum common subgraph problem: Faster solutions via vertex cover[C]//Proceedings of 2007 IEEE/ACS international conference on computer systems and applications. Amman:IEEE,2007,367-373.
- [7] NIRMALA P, LEKSHMI R S, NADARAJAN R. Vertex cover-based binary tree algorithm to detect all maximum common induced subgraphs in large communication networks[J]. Knowledge and Information Systems,2016,48(1):229-252.
- [8] NDIAYE S N, SOLNON C. CP models for maximum common subgraph problems [M]//LEE J. Principles and practice of constraint programming-CP 2011. Berlin Heidelberg:Springer,2011:637-644.
- [9] MINOT M, NDIAYE S N. Searching for a maximum common induced subgraph by decomposing the compatibility graph[C]//Bridging the gap between theory and practice in constraint solvers, CP2014-workshop. Lyon, France,2016.
- [10] 徐周波,古天龙,常亮,等. 加权约束满足问题的符号 ADD 求解算法[J]. 模式识别与人工智能,2011,24(1):14-21.
- XU Z B, GU T L, CHANG L, et al. Symbolic ADD algorithms for weighted constraint satisfaction problem [J]. Pattern Recognition and Artificial Intelligence, 2011,24(1):14-21.
- [11] 徐周波. 约束满足问题的符号算法及其在装配规划中的应用研究[D]. 西安:西安电子科技大学,2011.
- XU Z B. Research on symbolic algorithm for constraint satisfaction problem and their application in assembly planning [D]. Xi'an: Xidian University, 2011.
- [12] 古天龙,徐周波. 有序二叉决策图及应用[M]. 北京:科学出版社,2009.
- GU T L, XU Z B. Ordered binary decision diagram and its application[M]. Beijing:Science Press,2009.
- [13] BRYANT R E. Symbolic boolean manipulation with ordered binary-decision diagrams[J]. ACM Computing Surveys,1992,24(3):293-318.
- [14] BAHAR R I, FROHM E A, GAONA C M, et al. Algebraic decision diagrams and their applications [J]. Formal Methods in System Design, 1997,10(2/3):171-206.
- [15] ZAMPELLI S. A constraint programming approach to subgraph isomorphism [D]. Louvain: Université Catholique de Louvain,2008.

(责任编辑:尹 闯)