·先进计算· 文章编号:1000-3428(2016)01-0007-04 文献标识码:A 中图分类号:TP301

Holant 问题中的 Gadget 计算

杨 阳

(复旦大学上海市智能信息处理重点实验室,上海 200433)

摘 要: 在 Holant 二分理论的证明过程中,F-gate 和内插法是常用的归约技术,构件 Gadget 的计算则是其中的重要步骤。为提高 Gadget 计算效率,在边枚举和矩阵计算的基础上设计通用算法。针对该算法只能在指数时间内计算的缺点,引入特殊函数给出加速算法,依据自由度优先调用特殊函数,使 Holant 问题在多项式计算时间内得到解决。此外,研究发现 Gadget 计算能够推广为问题 $Holant(F\cup[1,1])$,加速算法也同时能描述其相应的易解函数类。

关键词: 计算复杂性;计数复杂性;Holant 二分理论;Gadget 计算;内插法;加速算法

中文引用格式: 杨 阳. Holant 问题中的 Gadget 计算[J]. 计算机工程,2016,42(1):7-10.

英文引用格式: Yang Yang. Gadget Computation in Holant Problems[J]. Computer Engineering,2016,42(1):7-10.

Gadget Computation in Holant Problems

YANG Yang

(Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, Shanghai 200433, China)

【Abstract】 F-gate and interpolation method are common reduction techniques in process of Holant dichotomy theorem's proof, and Gadget computation plays an important role. In order to increase the computational efficiency of Garget, this paper proposes a general algorithm based on edge enumeration and matrix computation. But the general algorithm only computes in exponential time, so aiming at this problem, this paper proposes an accelerative algorithm by introducing the special function. It calls special functions according to the degrees of freedom, which makes Holant problems be solved in polynomial computation time. Moreover, this paper finds that Gadget computation can be extended to Holant($F\cup[1,1]$) problem, and the accelerated algorithm can also show the tractable families of the problem.

【Key words】 computational complexity; counting complexity; Holant dichotomy theorem; Gadget computation; interpolation method; accelerated algorithm

DOI: 10.3969/j.issn.1000-3428.2016.01.002

1 概述

计数复杂性是计算复杂性理论的一个重要分支,其目标是根据计数问题的复杂度对其进行分类。积和式[1]计算是一类基本的计数问题,另一类重要的计数问题框架是#CSP[2],CSP 存在广泛的应用[3]。近年来,受到全息算法[4]的启发,一种新的框架 Holant 问题[5-6]被提出,#CSP 可以认为是一类特殊的 Holant 问题。顶点覆盖与完美匹配相应的计数问题都能够通过 Holant 框架表示。由于 Holant 框架十分具体,因此可以在上面证明二分理论[5]:计数问题属于 P 或者#P-hard。

在证明二分理论的过程中,研究者会使用相应的归约技术,其中包括 F-gate 和内插法,构件 Gadget 计算则是其中的关键步骤,如果能够高效地计算 Gadget,那么就更容易得到 Holant 二分理论。因此,本文给出计算 Gadget 的通用算法。然而,该算法对于特殊函数,指数计算将得到约简。为此,本文又对通用算法加以改进,提出加速算法。此外,研究发现 Gadget 计算可以视为一类使用函数[1,1]的 Holant 问题,而加速算法也同时能描述该类问题的易解函数。

2 Holant 问题与归约技术

在引入归约技术之前,给出 Holant 问题的形式化描述,记 $\Omega=(G,F,\pi)$,其中,$G=(V,E)$为图;π 表示每

基金项目: 国家自然科学基金资助项目(61170208);上海市基础研究基金资助重点项目(12JC1401400)。

作者简介: 杨 阳(1989-),男,硕士研究生,主研方向为计算理论。

收稿日期: 2014-12-16 **修回日期:** 2015-02-12 **E-mail:** yangyangsteven@aliyun.com

一个顶点 $v \in V$,都对应一个函数 $f_v \in F$,而且 f_v 将 $\{0,1\}^{\deg(v)}$ 映射到 R 中。考虑所有 0-1 分配,对每一条边 $e \in E$,分配 σ 给出估计式 $\prod_{v \in V} f_v(\sigma|_{E(v)})$,$E(v)$ 表示顶点 v 的关联边,$\sigma|_{E(v)}$ 表示 σ 对 $E(v)$ 的约束,于是,在 Ω 上的计数问题就是计算:

$$Holant_{\Omega} = \sum_{\sigma: E \to \{0,1\}} \prod_{v \in V} f_v(\sigma|_{E(v)}) \tag{1}$$

这里使用对称函数,对称函数 f 表示为 $[f_0, f_1, \cdots, f_k]$,其中,f_w 是 f 在海明权重 w 上的值。本文主要考虑 2-3 正则图,它是二部图,其中左边顶点的度为 2,右边顶点的度为 3。

下文将介绍证明 Holant 二分理论中常用的归约技术。

2.1 F-gate

一个 F-gate 可以认为是 (H, F, π),$H = (V, E, D)$ 是包含常规边 E 和悬边 D 的图,可以将 E 中常规边记为 $1, 2, \cdots, m$,D 中悬边记为 $m+1, m+2, \cdots, m+n$,从而可以定义函数:

$$\Gamma(y_0, y_1, \cdots, y_n) = \sum_{x_0, x_1, \cdots, x_m} H(x_0, x_1, \cdots, x_m, y_0, y_1, \cdots, y_n) \tag{2}$$

其中,$x_0, x_1, \cdots, x_m \in \{0,1\}$,$(y_0, y_1, \cdots, y_n) \in \{0,1\}^n$ 表示悬边上的分配;H 表示所有边分配的 Holant 值。这里,给出 F-gate 的一个例子[7],假设有函数 [1,0,0,1] 和 [1,1],可以使用图 1 的 F-gate 来实现函数 [1,0,1]。

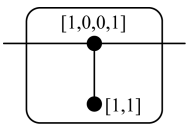

图 1 F-gate

使用 F-gate 的思想,可以完成 2 个 Holant 问题之间的归约。假设 g 为某 F-gate 构成的函数,那么 $Holant(F \cup \{g\}) \leqslant_T Holant(F)$。可以使用 Gadget 来表示 F-gate,因此,如果能够计算 Gadget,那么就可以完成归约。

2.2 内插法

内插法[8]是在研究计数问题中一个强有力的工具,文献[9]对其作了进一步推广。本文考虑内插二元函数,内插一元函数也有类似的结论[5,10]。

对于某些函数集合,希望对所有二元函数 $f = [x, y, z]$,有 $Holant(F \cup \{[x, y, z]\}) \leqslant_T Holant(F)$,更多细节可参考文献[11]。为了完成内插,需要递归构造合适的 Gadget,这里使用 2 个 Gadget,一个是二元初始函数 $g = [x_0, y_0, z_0]$,另一个是四元 Gadget,可以使用 3×3 的矩阵 $\boldsymbol{A}$ 表示。递归构造如图 2 所示。

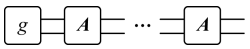

图 2 二元递归构造

从而得到递归关系:

$$\begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x_{s-1} \\ y_{s-1} \\ z_{s-1} \end{bmatrix}$$

在文献[11]中,有如下定理来保证内插成功:

定理 令 α, β, γ 为 $\boldsymbol{A}$ 的 3 个特征值,如果以下的 3 个条件都满足:(1) $\det(\boldsymbol{A}) \neq 0$;(2) 初始函数 $[x_0, y_0, z_0]$ 不与 $\boldsymbol{A}$ 的任何特征向量正交;(3) 对于所有 $(i, j, k) \in \mathbb{Z}_3 - \{(0,0,0)\}$,且 $i + j + k \neq 0$,有 $\alpha^i \beta^j \gamma^k \neq 1$,那么递归构造 $(\boldsymbol{A}, g)$ 可以内插所有二元函数。

由上述定理可知,g 的选择完全取决于矩阵 $\boldsymbol{A}$。在条件(1)和条件(3)下,构造合适的 Gadget 并不容易,因此,需要通用的算法来计算 Gadget。

3 通用算法

本文将给出计算 Gadget 的通用算法。首先,给出一个具体的 Gadget:Gadget1,如图 3 所示,它已经被用来证明二分理论[11-12]。

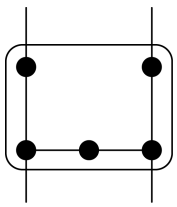

图 3 Gadget1

3.1 基本概念

在 Gadget1 中,悬边可作为布尔输入变量,常规边则在 $\{0,1\}$ 上完全分配。进一步,可以将悬边分为首边和尾边,一个 Gadget 只有首边(尾边)可以作为列(行)向量。

对于 Gadget1,度为 2 的顶点被分配为 [0,1,0],度为 3 的顶点分配为 [1,1,0,1],Gadget 可以视为 4×4 矩阵 $\boldsymbol{B}$,由于只考虑对称函数变换,存在等价 3×3 矩阵 $\boldsymbol{A}$,将列向量写成对称函数形式,于是得到:

$$\boldsymbol{B} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 2 & 1 & 1 & 0 \end{bmatrix}, \boldsymbol{A} = \begin{bmatrix} 0 & 2 & 0 \\ 1 & 1 & 1 \\ 2 & 2 & 0 \end{bmatrix}$$

在 Gadget1 中,增加平行边和顶点,可以推广到 2-k 正则图,如图 4 所示,其中平行边和顶点的数量为 $k-2$,它已经被用于文献[12]的内插技术中。

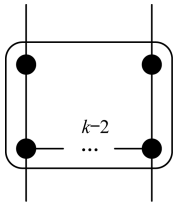

图 4　2-k 正则图上的 Gadget

3.2　算法步骤

在本文设计的通用算法中,考虑 Ω,将无向图中某些度为 1 的顶点作为外部顶点,剩下的为内部顶点。将外部顶点记为 V_1,内部顶点记为 V_2,$G=(V_1,V_2,E)$,V_1 中的顶点上的函数为[1,1],V_2 中顶点的函数根据 π 在 F 中分配。算法步骤如下:

输入　$\Omega=(G,F,\pi)$

输出　矩阵 $\boldsymbol{A}$

步骤 1　初始化矩阵 $\boldsymbol{B}$。

步骤 2　对矩阵 $\boldsymbol{B}$ 中每一项进行计算:

(1)悬边赋值。

(2)对余下的常规边的值进行枚举,每一次赋值都计算 $\prod_{v\in V}f_v(\sigma|_{E(v)})$,然后对这些值求和,就得到式(2)中的 Γ。

(3)将 Γ 的值写入矩阵 $\boldsymbol{B}$ 的相应项。

步骤 3　变换矩阵 $\boldsymbol{B}$ 到 $\boldsymbol{A}$。

可以根据矩阵 $\boldsymbol{B}$ 的下标索引对悬边赋值,例如,矩阵第 1 项对应于首边和尾边全都赋值为 0。显然步骤 2 可以认为是积和式的计算,它是整个算法的核心。步骤 3 可以通过矩阵运算完成。

下面分析算法的时间复杂度。在最坏的情况下,需要 2^n 步才能完成计算,如果 Gadget 包含100 条边,总共需要计算 2^{100} 步,远超过 DES 密码系统的计算量。幸运的是,在二分理论证明中使用的 Gadget 不包含如此多的边。

4　加速算法

通用算法显然不是一个多项式时间算法。当边的数量快速增加时,可能无法直接得到矩阵。然而笔者发现对于某些函数可以高效地进行计算。

4.1　退化函数

任何退化函数能够表示成一元函数的张量积。在 2-3 正则图中,退化的二元函数和三元函数有如下形式:[0,0,0],[0,0,1],[1,0,0],[1,1,1];[0,0,0,0],[0,0,0,1],[1,0,0,0],[1,1,1,1]。先不考虑[1,1,1],[1,1,1,1],对于其余情况,当悬边确定之后,最多只有一种情形,使得整个值非零。因此,在积和式中,最多只需计算 1 次。

下面分析[1,1,1]和[1,1,1,1],根据定义,[1,1,1,1]可以表示成 $[1,1]^{\otimes 3}$,[1,1,1]可以表示成 $[1,1]^{\otimes 2}$,因此,函数[1,1,1,1]的顶点被分解成 3 个[1,1]的顶点。于是,整个 Gadget 将被分解,分解后的 Gadget1 如图 5 所示。

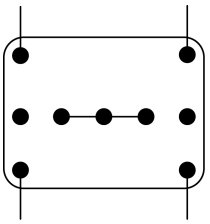

图 5　分解后的 Gadget1

此时尾边已经不影响计算,矩阵 $\boldsymbol{B}$ 可以表示为一个列向量,对于函数 $[x_0,x_1,x_2]$ 和[1,1,1,1],得到:

$$\boldsymbol{B}=(x_0+2x_1+x_2)\begin{bmatrix}(x_0+x_1)^2\\(x_0+x_1)(x_1+x_2)\\(x_0+x_1)(x_1+x_2)\\(x_1+x_2)^2\end{bmatrix}$$

从而可以在多项式时间内对其进行计算。

4.2　Equality 函数

在 2-3 正则图中,Equality 函数包含[1,0,1]和[1,0,0,1]。在 2-3 正则图中同时使用[1,0,1],[1,0,0,1],当悬边确定之后,Gadget 中的常规边也随即确定,矩阵 $\boldsymbol{B}$ 的首项和末项为 1,其他为 0,得到:

$$\boldsymbol{B}=\begin{bmatrix}1&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&1\end{bmatrix},\boldsymbol{A}=\begin{bmatrix}1&0&0\\0&0&0\\0&0&1\end{bmatrix}$$

对于这种情形,问题可以在 $O(n)$ 步内解决。

如果顶点上的函数为[1,0,0,1],那么对于它的关联边只有 2 种可能,同样可以加速计算,对于 Gadget1,二元函数为[a,1,b],有:

$$\boldsymbol{B}=\begin{bmatrix}a^3&a&a&b\\a^2&ab&1&b^2\\a^2&1&ab&b^2\\a&b&b&b^3\end{bmatrix},\boldsymbol{A}=\begin{bmatrix}a^3&2a&b\\a^2&ab+1&b^2\\a&2b&b^3\end{bmatrix}$$

对于 Disequality 函数[0,1,0],同样可以在多项式时间内完成计算。

4.3　函数特征

定义　对于 0-1 输入,一个函数 f 拥有非零可能性的数量称为它的自由度,记为 $M(f)$。

由定义可知,如果 f 是[0,0,0],[0,0,0,0],那么 $M(f)=0$;如果 f 是[0,0,1],[1,0,0],[0,0,0,1],[1,0,0,0],那么 $M(f)=1$;如果 f 是[1,0,1],[0,1,0],[1,0,0,1],那么 $M(f)=2$;其余情况,$M(f)>2$。

当 $M(f)\leqslant 2$ 时可以高效地进行计算,因此,自由度可以认为是 Gadget 计算函数的一个重要特征。

4.4 算法描述

在给出算法之前,笔者将 $M(f)\leqslant 2$ 的函数集合定义为 P。由于函数[1,1,1],[1,1,1,1]具有独立的加速算法,这里只考虑 P 中的函数。实际上,只需重新设计通用算法中步骤 2 的计算部分。

步骤 1 判断 $P\cap F$ 是否为空集,如果为空集,使用通用算法的步骤,否则进入步骤 2。

步骤 2 选择 P 中的函数进行计算。

(1)选择函数属于 P 的顶点 v。

(2)对顶点 v 的关联边进行赋值,使得函数值非零。

步骤 3 对剩下的边的值进行枚举,并且计算 Γ 的值。

改进算法的主要思想就是在边分配中减少枚举的次数。考虑这样的情况,在 Gadget1 中使用函数[a,1,b]和[1,0,0,1],使用通用算法,则需要枚举所有的边,共有 $2^8=256$ 种可能,而使用改进算法时,首先在尾边顶点的函数[1,0,0,1],有 4 种可能的情况,对于首边也有 4 种可能,共 16 种情形,从而减少了整个计算次数。

在改进算法的步骤 2 中,可以使用贪心选择。本文的选择使用了 2 个标准:最大的函数元数及最小的函数自由度,这样做的目的是为了尽可能快地传递边的值。

5 相关讨论

5.1 Gadget 构造

Gadget 计算是 Gadget 构造中的核心问题。当 Gadget 构造完成后,并不知道它是否能够实现某个函数或者保证内插成功,Gadget 计算可以作为验证 Gadget 构造正确性的工具。此外,如果能够高效地计算 Gadget 矩阵,则可构造尽可能多的 Gadget,从而更容易找到需要的 Gadget。

5.2 Holant 问题间的关系

Gadget 计算与 Holant 求和都可以认为是积和式计算,然而,它们主要的区别就是 Gadget 计算使用悬边作为输入,这里给出如下例子。

在 Holant 二分理论[11]中,存在一类易解函数,它使得 Holant 的值总是为 0。其中包含[0,1,0]|[0,1,0,0],在 Gadget1 中使用这些函数,有:

$$\boldsymbol{B}=\begin{bmatrix}0&0&0&0\\1&0&0&0\\1&0&0&0\\0&1&1&0\end{bmatrix},\boldsymbol{A}=\begin{bmatrix}0&0&0\\1&0&0\\0&2&0\end{bmatrix}$$

可见,矩阵 $\boldsymbol{A}$ 并不是零矩阵,Holant 值不为 0。

6 结束语

计算复杂性中最基本的二分理论是对多项式时间和指数计算问题进行分类,有许多方法可以对特定问题加速计算,例如动态规划、线性规划、量子计算等。本文给出一个通用算法来计算 Holant 问题中的 Gadget,它需要指数时间计算。同时,笔者发现存在与加速算法相关的特殊函数,它们可使问题在多项式时间内计算,因此,在加速算法中,依据自由度优先使用这类特殊函数。如果存在其他函数可以加速计算,也可把它们加入到函数集合 P 中来进一步优化本文提出的算法。受到 P 中函数的启发,下一步将把 Gadget 计算的研究重点放在函数理论上,对于可以加速计算的函数,构建完整的函数理论。此外,研究发现 Gadget 计算可以表示为更一般的 Holant 问题 $Holant(F\cup[1,1])$。因此,今后将通过加速算法中的易解函数,给出这类问题完整的二分理论。

参考文献

[1] 王 磊. 积和式近似算法的分析及应用[D]. 北京:清华大学,2013.

[2] 黄 平. #CSP 的相变及近似算法研究[D]. 长春:东北师范大学,2012.

[3] 董存祥,王文俊,杨 鹏. 基于约束满足问题的应急决策[J]. 计算机工程,2010,36(7):276-278.

[4] Valiant L G. Holographic Algorithms[J]. SIAM Journal on Computing,2008,37(5):1565-1594.

[5] Cai Jinyi, Liu Pinyan, Xia Mingji. Computational Complexity of Holant Problems[J]. SIAM Journal on Computing,2011,40(4):1101-1132.

[6] Cai Jinyi,Kowalczyk M. Partition Functions on k-regular Graphs with {0,1}-vertex Assignments and Real Edge Functions[J]. Theoretical Computer Science,2013,494:63-74.

[7] Cai Jinyi,Liu Pinyan,Xia Mingji. A Computational Proof of Complexity of Some Restricted Counting Problems[J]. Theoretical Computer Science,2011,412:2468-2485.

[8] Valiant L G. The Complexity of Enumeration and Reliability Problems[J]. SIAM Journal on Computing,1979,8(3):410-421.

[9] Vadhan S P. The Complexity of Counting in Sparse, Regular, and Planar Graphs[J]. SIAM Journal on Computing,2001,30(2):398-427.

[10] Cai Jinyi,Liu Pinyan,Xia Mingji. Holant Problems and Counting CSP[C]//Proceedings of the 41st Annual ACM Symposium on Theory of Computing. Bethesda, USA:ACM Press,2009:715-724.

[11] Cai Jinyi, Liu Pinyan, Xia Mingji. Holographic Algorithms by Fibonacci Gates and Holographic Reductions for Hardness[C]//Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science. Philadelphia,USA:IEEE Press,2008:644-653.

[12] Cai Jinyi, Kowalczyk M. Spin Systems on k-regular Graphs with Complex Edge Functions[J]. Theoretical Computer Science,2012,461:2-16.

编辑 金胡考