

图的树分解算法及其应用



雷莹 许道云

贵州大学计算机科学与技术学院 贵阳 550025

(1152586591@qq.com)

摘要 一个图 $G=(V,E)$ 的树分解是将结点集 V 的子集作为树 T 的节点,使得在 T 上任意一条路径上的两个端节点的交集包含于该路径上的任意一个节点中。将 T 上最小(节点)对应子集的元素个数减 1 定义为分解树 T 的宽度,用宽度最小的分解树 T 的树宽度定义图 G 的树宽度。一个合取范式(Conjunctive Normal Form,CNF)公式 F 可以用一个二分图 $G=(V\cup C,E)$ 表示(公式的因子图),其中变元结点集 V 对应公式 F 中的变元集,子句结点集 C 对应公式 F 中的子句集,变元在子句中的正(负)出现用实(虚)边表示。忽略公式因子图中边上的符号,得到一个二分图。文中研究了图的树分解算法,并将树分解算法应用到 CNF 公式的因子图树分解。通过实验观察公式因子图的树宽度与求解难度之间的联系。

关键词 树分解;CNF 公式;树宽度;求解难度

中图法分类号 TP301

Tree Decomposition Algorithm of Graph and Its Application

LEI Ying and XU Dao-yun

College of Computer Science and Technology, Guizhou University, Guiyang 550025, China

Abstract A tree decomposition of the graph $G=(V,E)$ refers to taking a subset of the node set V as a node of the tree T , to make the intersection of the two endpoints on any path of T included in any node on the path. The number of elements of the minimum (node) corresponding subset on T minus 1 is defined as the width of the decomposition tree T , and the tree width of the graph G is defined by the tree width of the decomposition tree T with the smallest width. The CNF formula F can be represented by a bipartite graph $G=(V\cup C,E)$ (factor graph of the formula), where the variable node set V corresponds to the variable set, and the clause node set C corresponds to the clause set in the formula F . The positive (negative) occurrence of the element in the clause is represented by the real (virtual) side. In order to study the bipartite graph, the symbols on the edges of the formula factor graph are ignored. This paper studies the tree decomposition algorithm of graphs and applies the tree decomposition algorithm to the factor graph tree decomposition of CNF formula, aiming to explore the relationship between the tree width of the formula factor graph and the difficulty of the solution-finding through experiments.

Keywords Tree decomposition, CNF formula, Tree width, Difficulty of solution-finding

1 引言

设 $G=(V,E)$ 为一个简单无向图,对其进行树分解是将结点集 V 的子集作为树 T 的节点, T 中的节点 i 对应 G 的一个结点子集 X_i ,使得在 T 上任意一条路径 P 上的起点 s 和终点 t 满足条件:对于路径 P 上任意一个节点 i ,有 $X_s \cap X_t \subseteq X_i$ 。记 T 的节点指标集为 I_T ,正整数 $tw_T(G) = \min\{|X_i| : i \in I_T\} - 1$ 称为 G 关于分解树 T 的树宽度^[1-3]。对于 G 的所有分解树 T , $tw_T(G)$ 的最小值定义为图 G 的树宽度,记为 $tw(G)$ 。以图的树宽度 $tw(G)$ 为参数,可以对图进行分类。可以看出:一个非平凡图的树宽度至少为 1,树本身的宽度等于 1。以分解树的宽度为参数 k ,判定一个图 G 是否存在一个树

分解 T 使得 $tw_T(G) \leq k$,是一个 NP-难问题^[4]。因此,利用树分解的某些结构信息对 NP-难问题进行分类。

图的树分解是将给定图的结构关系用一种树形结构表示,通过研究树的结构信息来刻画原图的一些性质。树宽则刻画了给定的图形结构与分解树的相似程度。文献^[5]中的研究表明,基于图的树分解,并限制图的树宽时,很多复杂难解的问题可在多项式时间内进行求解。图的树分解和树宽在组合优化^[6]、人工智能^[7]和概率网络^[8]等方面有着广泛的应用及研究意义。

可满足性问题(The Satisfiability Problem, SAT)是指:对于任意给定的 CNF 公式 F ,判定是否存在一组对公式 F 中所有变元的真值指派,使得公式 F 的取值为真。在实际应用

到稿日期:2019-11-15 返修日期:2020-03-13 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。

基金项目:国家自然科学基金(61762019,61862051)

This work was supported by the National Natural Science Foundation of China(61762019, 61862051).

通信作者:许道云(dyxu@gzu.edu.cn)

中,大量带有协调性或者一致性的检验等求解问题,都可以转化为 SAT 问题来进行求解。因此, SAT 问题的理论和应用研究长期受到人工智能、数理逻辑及理论计算机科学等领域的普遍关注。在 SAT 问题的研究中,限制每个子句长度为 k 的 SAT 问题称为 k -SAT 问题,当 $k \geq 3$ 时,该问题是 NP-完全问题^[9]。尽管对整个 k -CNF 类来说,求解 SAT 问题是 NP-完全的,但统计现象表明:对大多数 k -CNF 公式而言,其判定问题并非“困难”。在随机 k -SAT 问题的研究中,子句约束密度 α 是一个非常重要的结构参数^[10-11],其中 α 表示在公式 F 中子句数(m)与变元数(n)的比值,即 $\alpha = n/m$ 。 α 的取值大小不仅会影响 k -CNF 公式的判定难度,还会影响 k -CNF 公式的可满足性。随着子句约束密度 α 的增大,存在一个与 k 有关的临界值点 α_c : 1) 当 $\alpha < \alpha_c$ 时,公式以高概率可满足; 2) 当 $\alpha > \alpha_c$ 时,公式以高概率不可满足。这种现象被称为随机 k -SAT 问题的相变现象, α_c 称为该类问题可满足的相变点, $\alpha_c(k)$ 表示 k 的在不同取值的相变阈值点。

目前,对 SAT 问题的研究主要集中在 CNF 公式实例的生成模型、求解算法、结构性质以及相变性质等方面。在 CNF 公式实例结构理论分析方面,文献[12-13]将公式实例表示为一个二分图,利用二分图的匹配理论对公式实例的结构进行分析。Xu 等^[14]利用极小不可满足公式作为构造工具,得到了一个保留 NP 完全性的正则 (3,4)-SAT 问题。这类问题有着规则的结构特性,公式中每个子句文字数恰好为 3,每个变元恰好出现 4 次。对于 (3,4)-正则二分图,研究者提出了一种利用此类型的二分图求解公式可满足性的方法,即如果正则 (3,4)-CNF 公式对应的因子图存在一条 P_7 -路径因子,则公式可满足,并给出了严格证明。同时利用图分解的方法来判定公式的可满足性,证明了对于任意的 (3,4)-CNF 公式,如果其对应的因子图能被划分为两个 (3,2)-双向正则二分图,那么公式是可满足的。

随着研究的不断深入,文献[15]通过引入一种特殊的树形结构,将正则公式的因子图转换到树形结构上进行研究,证明了在随机正则 (3, r)-SAT 问题中的簇集相变点可能发生在 $r=8$ 附近。在参数化 CSP 问题^[16-17]的研究中,研究者们利用参数计算理论研究一些 SAT 问题是否为固定参数可解的,并提出了更高效的参数化算法求解固定参数可解问题,其中参数计算技术是设计有效固定参数算法的关键。为了进一步研究参数化 SAT 问题,文献[18]引入了树分解技术,将图尽可能地分解为一棵树。Bodlaender 等^[3]提出的树分解算法可在线性时间内判定一个图的树宽,为求解难解问题的图结构提供了理论基础。基于该理论基础,文献[19-21]利用树分解算法来求解受约束的可满足性问题。

如果一个 CNF 公式 F 所对应的因子图是一棵树,那么公式 F 也称为树公式。这类公式具有某些良好的性质:存在多项式时间算法判定公式的可满足性;在信息传播算法中警告传播(Warning Propagation, WP)算法以树公式^[22]为输入时,算法收敛。

一个 CNF 公式 F 称为线性公式,如果任意两个不同子句至多含有一个公共变元。文献[23-25]证明了限制 CNF 公式为线性公式,其可满足性判定问题仍然是 NP-完全的。一

个 CNF 公式 F 可以用一个二分图 $G=(V \cup C, E)$ 表示(公式的因子图),其中变元结点集 V 对应公式 F 中的变元集,子句结点集 C 对应公式 F 中的子句集,变元在子句中的正(负)出现用实(虚)边表示。如果忽略公式因子图中边上的符号,则得到一个线性二分图。一个二分图 $G=(V \cup C, E)$ 称为线性的,如果对于任意的两个不同结点 $c, c' \in C$,其至多有一个公共邻接结点,即 $|N(c) \cap N(c')| \leq 1$ 。

本文研究图的树分解问题及树分解算法,目的是通过树分解方法探究公式的结构性质,特别是一些结构(如线性公式)限制下 CNF 公式因子图的树分解问题。通过研究图的树分解算法,并将树分解算法应用到 CNF 公式的因子图上,来探究 NP-难问题的难解特征;同时通过实验观察公式因子图的树宽度与求解难度之间的联系。研究表明,很多复杂问题经过树分解后,可在多项式时间甚至线性时间内进行求解,而树分解后所得的树宽在很大程度上影响着问题的求解效率。因此,本文基于经典的启发式树分解方法提出了一种二分图的树分解算法。实验对比结果表明,本文方法可以求解出较优的树宽,为下一步探究难解实例的难解特征提供了实验基础。

2 基础知识

合取范式是命题逻辑公式的一种规范表示。命题变元 x 及其否定 $\neg x$ 统称为文字,一个子句 C 是若干个文字的析取,用 $C = \{L_1 \vee L_2 \vee \dots \vee L_k\}$ 表示,其中 k 表示子句 C 的长度。一个合取范式公式 F 由有限个子句的合取构成, $F = (C_1 \wedge C_2 \wedge \dots \wedge C_m)$ 。一个 CNF 公式称为 k -CNF 公式,若该公式中每个子句的长度恰好都为 k 。有时,子句 C 可用文字集合 $\{L_1 \vee L_2 \vee \dots \vee L_k\}$ 表示,公式 F 用子句集合 $\{C_1 \wedge C_2 \wedge \dots \wedge C_m\}$ 表示。

2.1 CNF 公式的因子图

一个 CNF 公式可以表示为一个因子图,这类图的一侧是以布尔变元集为布尔变元顶点,而另一侧则是以子句集为子句顶点。当某个布尔变元出现在某个子句中时,表示有一条边连接该变元顶点和子句顶点,其中用实线表示变元在子句中正出现,用虚线表示变元在子句中负出现。将 CNF 公式转换为因子图时,用矩形表示子句顶点,用圆形表示变元顶点。图 1 展示了 3-CNF 公式的一个实例 $F = (C_1 \wedge C_2 \wedge C_3 \wedge C_4 \wedge C_5 \wedge C_6)$ 的因子图,其中 $C_1 = (x_3 \vee x_4 \vee \neg x_5)$, $C_2 = (\neg x_1 \vee \neg x_4 \vee x_5)$, $C_3 = (x_1 \vee x_2 \vee \neg x_5)$, $C_4 = (\neg x_1 \vee \neg x_2 \vee x_3)$, $C_5 = (\neg x_1 \vee \neg x_2 \vee \neg x_4)$, $C_6 = (x_1 \vee x_2 \vee x_3)$ 。

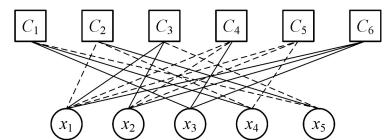


图 1 公式 F 对应的因子图

Fig. 1 Factor graph corresponding to formula F

在 CNF 公式的因子图中,如果忽略边上的符号,则得到一个二分图。对于二分图 $G=(V \cup C, E)$,如果对于任意的两个不同结点 $c, c' \in C$,其至多有一个公共邻接结点,即 $|N(c) \cap N(c')| \leq 1$,则称此二分图是线性二分图。其中, $N(c)$ 表示结

点 c 在图中的邻接结点集。线性二分图概念的引入主要针对线性 CNF 公式。一个 CNF 公式 F 称为线性公式,如果任意两个不同子句至多含有一个公共变元。文献[23-25]已经证明:限制 CNF 公式为线性公式,其可满足性判定问题仍然是 NP-完全的。

因此,考虑线性二分图的树分解有助于研究可满足性判定问题有效求解算法的设计和分析。

2.2 图的树分解和树宽

本文中的图都指的是简单无向图。图的树分解是对图 G 的结点集进行划分,生成与之对应的一棵分解树 T , T 中的节点 i 对应 G 的一个结点子集 X_i 。将分解树的宽度作为参数对 NP-难问题的复杂度进行分解,从而利用树分解的某些结构信息对 NP-难问题进行分类。

图的树分解是为了将给定的图尽可能地分解为树形结构,通过研究树上的结构信息来刻画原图的一些性质。引入树宽的概念是为了反映给定的图形结构与分解后的树形结构的相似程度。在实际应用中,很多问题涉及的图的规模都很大,直接求解的难度也很高。若将这类问题的图转化为树,则可在多项式时间甚至线性时间内求解问题。因此,越来越多的研究者基于图的树分解思想,将规模大且难解的问题转化为规模较小且易解的子问题,从而进一步有效求解 NP-难问题。

设 $G=(V,E)$ 为一个简单无向图, $T=(I,F)$ 为一棵树。直观上,图 G 的一个结点子集构成一个包(bag),给一个包赋予一个标记,标记集 I 作为分解树结点。从而,图 G 的树分解由包(bag)对应构成。

图 G 的一个树分解可以表示为一个序对 $T_G=(X,T)$,其中 $X=\{X_i:i \in I\}$ 为图 G 的结点集 V 的一族非空子集 (I 是有限指标集,其中指标代表 V 的一个子集)。 $T_G=(X,T)$ 满足如下条件:

(1) $\bigcup_{i \in I} X_i = V$, 即指标集 I 对应的集族 $X=\{X_i:i \in I\}$ 是图 G 的结点集的一个覆盖。

(2) 对于 $\forall (u,v) \in E, \exists i \in I$, 使得 $u,v \in X_i$, 即图 G 中每一条边的两个结点同时包含在树分解中的某个节点中。

(3) 对于 $i,j,k \in I$, 如果在树 T 中 k 出现在从 i 到 j 的一条路径上, 则 $X_i \cap X_j = X_k$ 。

对于树分解 $T_G=(X,T)=(\{X_i:i \in I\}, T)$, 其宽度定义为 $\max(|X_i|-1)$ 。

一个图 G 的树分解不唯一。图 G 的树宽(Tree Width)是指图 G 所有可能的树分解中宽度的最小值,记为 $tw(G)$, 即 $tw(G)=\min_{(X,T)} \max_{i \in I} |X_i|-1$ 。

由图的树分解定义可知,当图 G 本身是树时,其树宽等于 1;当图 G 是 n 结点完全图时,其树宽等于 $n-1$ 。进一步,容易得到如下基本性质。

性质 1^[1] 设 G 为一个图, $T_G=(\{X_i:i \in I\}, T)$ 是 G 的一个树分解,则对于任意子集 $W \subseteq V$, 如果 W 在 G 中是一个团, 则一定 $\exists i \in I$, 使得 $W \subseteq V_i$ 。

性质 1 表明:图 G 的树宽度不低于 G 中最大团的结点数减 1, 从而容易得出如下的树宽度下界估计。

性质 2^[2] 设 G 为一个图, 则 $tw(G) \geq \gamma(G) \geq \delta(G)$ 。其中, $\delta(G)$ 为图 G 中的最小结点度数, $\gamma(G)$ 则由如下定义给出:

$$\gamma(G) = \begin{cases} |V|-1, & \text{if } G \text{ is a clique} \\ \min_{u,v \in V, (u,v) \in E} \max\{d(u), d(v)\}, & \text{otherwise} \end{cases} \quad (1)$$

为了设计近似最优树分解算法,并估计图的树宽上界,需要用到如下概念之间的关系及方法。文中所谓的近似分解,是指分解的树宽度接近图本身的树宽度。

设 G 为一个图,连接一个回路中不相连的两个结点之间的一条边称为一条弦(Chordal)。称一个图 G 为弦图,如果图中任意一个至少有 4 条边的回路至少有一条弦。任何一条长度超过 4 的回路都被称作“三角剖分”。

给定一个图 $G=(V,E)$, 对于指定的结点 $v \in V$, 引入记号 $N_G(v) = \{w:(v,w) \in E, v \neq w\}$, 表示 v 且不含 v 的邻接结点集;记号 $N_G[v] = N_G[v] \cup \{v\}$, 表示 v 与 v 的邻接结点集的集合。通常,从图 G 中删去结点 v 是指:从图 G 中删去结点 v 及其关联的边,得到的图记为 G_v 。

在图的树分解中,利用消点运算,由图 G 得到一个新的图是指:从图 G 中删去结点 v , 并在其邻接结点集 $N_G(v)$ 中不相邻的结点对之间加上边,使之在 G_v 中构成一个团。从图 G 中消去结点 v 后得到的图记为 $G_{e(v)}$, 并将因消去结点 v 而在邻接结点集 $N_G(v)$ 中加入的边集合记为 F_v^e 。

定义 1(消点序) 设 $G=(V,E)$ 为一个图, $n=|V|$, 双射 $f:V \rightarrow [n]$ 称为一个消点序(Elimination Ordering), 其中 $[n]=\{1,2,\dots,n\}$ 。

直观上, f 规定了图中结点的顺序。称一个消点序 f 是完美的,如果对于每一个结点 $v \in V$, 其邻接结点集中序值高于 v 的结点集构成一个团。即, 结点集 $\{w: [(v,w) \in E] \wedge [f(w) > f(v)]\}$ 在 G 中构成一个团, 如图 $G=(V,E)$ 中, $V=\{1,2,3,4\}$, $E=\{(1,2), (1,4), (2,3), (3,4), (2,4)\}$ 。若图 G 有一个完美的消点序 f , 定义 $f(i)=i (i=1,2,3,4)$ 即可。

设 $G=(V,E)$ 为一个图, 任给一个消点序 $V=\{v_1, \dots, v_n\}$, 依次消去结点 $\pi^{-1}(1), \dots, \pi^{-1}(n)$, 最后得到一个空图。在消点过程中,伴随着一个图序列和一个边集合序列:

$$\begin{aligned} G^0 &= G, \\ G^1 &= C_{e(\pi^{-1}(1))}^0, & F^1 &= F_{G_0}^{\pi^{-1}(1)}, \\ G^2 &= C_{e(\pi^{-1}(2))}^1, & F^2 &= F_{G_1}^{\pi^{-1}(2)}, \\ &\dots & &\dots \\ G^k &= C_{e(\pi^{-1}(k))}^{k-1}, & F^k &= F_{G_{k-1}}^{\pi^{-1}(k)}, \\ &\dots & &\dots \\ G^n &= C_{e(\pi^{-1}(n))}^{n-1}, & F^n &= F_{G_{n-1}}^{\pi^{-1}(n)} \end{aligned}$$

注意:边集合序列中有些是空集合。

根据图序列和边集合序列得到图 $G_\pi=(V, E \cup F)$, 其中, $F=F^1 \cup \dots \cup F^n$ 表示边集合序列。结合消点序的定义,可知消点序 π 是图 G_π 的一个完美消点序。

定义 2(树交叉图) 设 $G=(V,E)$ 为一个图, 称 G 是某棵树的子树交叉图(Intersection Graph), 如果对于一棵树 $T=(I,F)$ 中的每一个结点 $v \in V$, 都有 T 的一棵子树 $T_v=(I_v,$

F_v), 使得对应的两个不同结点 $v, w \in V$:

$$(v, w) \in E \text{ 当且仅当 } I_v \cap I_w \neq \emptyset$$

即不同结点对应的子树指标集至少有一个公共指标结点。

弦图、完美消点序、子树交叉图以及树分解之间存在如下关系。

定理 1^[1-2] 设 $G=(V, E)$ 为一个图, 则下列关系等价:

- (1) G 是一个弦图;
- (2) G 有一个完美消点序;
- (3) G 是一棵树的子树交叉图;
- (4) G 有一个树分解 $(\{X_i; i \in I\}, T=(I, F))$, 使得对于每个 $i \in I, X_i$ 在 G 中是一个团。

由定理 1, 可得出如下简单关系:

- (1) 图的树分解与弦图密切相关;
- (2) 图的消点序与分解树的构造顺序关联;
- (3) 图的最大团与树宽度关联;
- (4) 图的极大团在树分解的贪心算法设计中起作用。

在图论中, 可以通过三角剖分 (Triangulation) 过程得到一个弦图。一个图的三角剖分过程的描述见定义 3。

定义 3(三角剖分) 称一个图 $H=(V_H, E_H)$ 是图 $G=(V, E)$ 的三角剖分, 如果 H 是在 G 的基础上适当增加某些边后得到的弦图; 即在图 G 中长度超过 4 的回路中增加某些弦边, 使之成为弦图。

极小化三角剖分 $H=(V_H, E_H)$ 是指:

- (1) 本身是 G 的三角剖分;
- (2) 不存在 E_H 的真子集 $F: E \subseteq F \subseteq E_H$, 使得 $H'=(V_H, F)$ 是 G 的三角剖分。

注意: 一个图 G 可能存在两个不同的极小三角剖分, 但任意两个不同极小三角剖分所含的边数相等。

根据上述分析, 进一步构造贪心算法来求解图 G 的树分解。

给定一个图 G 以及一个双射 $\pi: V \rightarrow [n]$ (作为一个消点序), 依 π 规定的顺序适当加边后得到一个图 G_π^+ , 使 π 成为 G_π^+ 的一个完美消点序。对 G_π^+ 消去新边, 还原出原始图 G 。下面给出迭代加边算法 $Fill(G, \pi)$ 的大致描述。

- (1) 初始化: $E_H = E$ 。
- (2) 从 $i=1$ 到 n 做如下迭代:
 - 1) 根据消点序 π 求出第 i 个结点 $v: v = \pi^{-1}(i)$;
 - 2) 对 v 的所有邻接结点对 w, w' :

如果 $\pi(w) > i, \pi(w') > i$, 并且 $(w, w') \notin E_H, E_H \leftarrow \{(w, w')\} \cup E_H$ 。
- (3) 返回 $H=(V, E_H)$ 。

定义图函数 $G_\pi^+ = Fill(G, \pi)$, 由上述分析可知 π 是 G_π^+ 的一个完美消点序。

根据定理 1, G_π^+ 是一个弦图, 并且 G_π^+ 有一个树分解 $(\{X_i; i \in I\}, T=(I, F))$, 每个 $i \in I, X_i$ 在 G_π^+ 中是一个团。一个自然的问题是: 如何找到一个消点序 π , 使得从 G 到 G_π^+ 增加的边数最少。

定理 2 进一步说明了算法 $Fill(G, \pi)$ 在计算一个图的树分解及树宽过程中存在的关系。

定理 2^[2] 设 $G=(V, E)$ 为一个图, $n=|V|$, 设 k 为不超

过 n 的一个非负整数, 则下列结果等价:

- (1) G 的树宽度至多为 k ;
- (2) G 有一个三角剖分 H , 使得 H 中最大团的大小至多为 $k+1$;
- (3) 存在一个消点序 π , 使得 G_π^+ 中最大团的大小至多为 $k+1$;
- (4) 存在一个消点序 π , 使得 G_π^+ 中不存在结点 v, v 有超过 k 个邻接结点, 其在 π 下的序值比 v 大。

3 图的树分解算法

给定一个图 $G=(V, E)$, 假定 $n=|V|$, 对于任意给定的消点序 $\pi: V \rightarrow [n]$, 由第 2 节提出的算法 $Fill(G, \pi)$ 皆可生成一个图 G_π^+ , 使得 π 是 G_π^+ 的一个完美消点序。此外, 图 G_π^+ 还是一个弦图。

从而, 依据 π 可以构造 G_π^+ 的一个树分解, 并能精确得到 G_π^+ 的树宽度。以 G_π^+ 的一个树分解为原始图 G 的树分解, 并由此得到图 G 的树宽度的上界。

本节给出了一个最小度、最小填边数优先排序的贪心算法, 如算法 1 所示, 以产生图 G 的一个消点序, 进一步构造图 G 的一个树分解。

算法 1 *GreedyMinDegree(G)*

输入: 无向连通图 $G=(V, E), V=\{v_1, \dots, v_n\}$

输出: 消点序 $\pi: V \rightarrow [n]$ (双射)

1. 初始化: $G_0 \leftarrow G, V_0 \leftarrow V, E_0 \leftarrow E$;
2. $i=0$;
3. While $i < n$ do
4. 求出 $G_i=(V_i, E_i)$ 中度数最小的结点集 MV_i (可能不止一个最小度结点);
5. 在 MV_i 中找一个结点 v , 使得 (在图 G_i 中) 邻接结点集 $N(v)$ 中的填边数最少 (即 $N(v)$ 两点之间无边相连的“结点对”数量最少);
6. 消点运算
 - 6.1. 定义: $\pi(v) = i+1$;
 - 6.2. 删点: $V_{i+1} \leftarrow V_i \setminus \{v\}, E_i \leftarrow E_i \setminus \{(v, w): w \in N(v)\}$;
 - 6.3. 填边: $E_{i+1} \leftarrow E_i \cup F_i, F_i = \{(w, w'): w, w' \in N(v), (w, w') \notin E_i\}$;
 - 6.4. $G_{i+1} = (V_{i+1}, E_{i+1})$;
7. $i \leftarrow i+1$;
8. Enddo;
9. 输出 π 。

可以看出: 在算法 1 中, 依次删去的点为 $\pi^{-1}(1), \dots, \pi^{-1}(n)$, 依次填补的边集合为 F_0, \dots, F_{n-1} (可能有空集出现)。它们关联的结点子集分别为 $\pi^{-1}(1), \dots, \pi^{-1}(n)$ 的邻接结点集中的某个子集。

最后, 令 $G'=(V, E \cup F), F=F_0 \cup F_1 \cup \dots \cup F_{n-1}$, 从而得到图 $G=(V, E)$ 的三角剖分。

定理 3 设 $G=(V, E)$ 为一个图, $n=|V|$, 则有:

- (1) 收集算法 1 中新增边集合产生的图 $G'=(V, E \cup F)$ 形成图 G 的一个三角剖分图, 其中 $F=F_0 \cup F_1 \cup \dots \cup F_{n-1}$ 。
- (2) 利用算法 1 输出的消点序 π , 填充算法得到图: $G_\pi^+ = Fill(G, \pi) = G'$ 。

推论 1 设 $G=(V, E)$ 为一个图, $\Delta(G)$ 是图 G 的最大结

点度数,则 G 的树分解宽度 $tw(G) \leq \Delta(G)$ 。

算法 1 给出了消点序的一个构造方法,定义 $\pi = GreedyMinDegree(G)$,进一步依据 π 设计出一般图的树分解算法,如算法 2 所示。

算法 2 *MinDegreeTreeDecomposition(G)*

输入:无向连通图 $G=(V,E)$

输出: G 的一个树分解 $(\{X_i; i \in I\}, T=(I,F))$

第 1 阶段:基础树分解 $(\{X_i; i \in I_0\}, T_0=(I_0, F_0))$

1. 调用生成消点序: $\pi \leftarrow GreedyMinDegree(G)$, 设 $V = \{\pi^{-1}(1), \dots, \pi^{-1}(n)\} = \{v_1, \dots, v_n\}$;

2. 产生一个结点集合序列(包序列) X_1, \dots, X_n :

$$X_1 = N_G[v_1] = N_G(v_1) \cup \{v_1\};$$

$$X_2 = N_{G_{e(v_1)}}(v_2) \cup \{v_2\};$$

...

$$X_k = N_{G_{e(v_1, \dots, v_{k-1})}}(v_k) \cup \{v_k\};$$

...

$$X_n = \{v_n\};$$

其中, $G_{e(v_1, \dots, v_{k-1})}$ 表示依次从 G 中消去 v_1, \dots, v_{k-1} 产生的图。

3. 分解树的结点标号集: $I_0 = \{1, \dots, n\}$;

4. 分解树的边集合: $F_0 = \{(1, j_1), \dots, (n-1, j_{n-1})\}; j_k = \min\{k' : (v_k, v_{k'}) \in E(G_{e(v_1, \dots, v_{k-1})})\}, k=1, 2, \dots, n-1; E(G_{e(v_1, \dots, v_{k-1})})$ 表示图 $G_{e(v_1, \dots, v_{k-1})}$ 的边集合。

第 2 阶段:相邻包合并

1. 令 $I \leftarrow I_0; F \leftarrow F_0$;

2. 重复如下操作,直到不出现一个包是另一个包的真子集:

如果有一对 $i \neq j, X_i \subset X_j$, 并且不存在 k , 使得 $k \neq i, j, X_i \subset X_k \subset X_j$, 则:

2.1. 删去 X_i 包;

2.2. 删去标号 i 及关联 i 的边,并将原来连 i 的点改为连 j ,即:

$$I \leftarrow I \setminus \{i\}; F \leftarrow (F \setminus \{(i, k) : (i, k) \in F\}) \cup \{(j, k) : (i, k) \in F\};$$

3. 输出: $(\{X_i; i \in I\}, T=(I, F))$ 。

由于真包含的包的上升序列存在于 T 的一条路径上,实际合并时可以采用如下操作:

(1) 将极大严格上升的包序列上的包合并到极大包 X , 单点下标合并成一个下标子集, 并取得其中最大标记作为新标记, 对应标识该极大包 X ;

(2) 两个新标记之间有边相连, 当且仅当对应的两个原始标记子集中有一对标记在 T_0 中有边。

图 2 给出了无向图 $G=(V,E)$ 的一个示例。

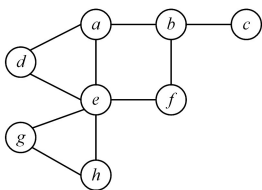


图 2 无向图 $G=(V,E)$

Fig. 2 Undirected graph $G=(V,E)$

根据上述算法对图 2 所示无向图进行树分解, 具体分解过程如下。

(1) 利用算法 1 求消点序 π , 消点过程如图 3 所示。

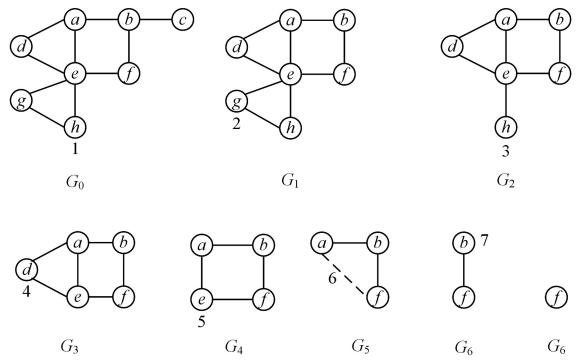


图 3 求解图 $G=(V,E)$ 的消点序的过程

Fig. 3 Process of solving elimination order of graph $G=(V,E)$

(2) 反向递归生成结点包序列以及分解树, 其中各结点包的构成如下:

$$X_8 = \{f\}, X_7 = \{b, f\}, X_6 = \{a, b, f\}, X_5 = \{e, a, f\}, X_4 = \{d, a, e\}, X_3 = \{h, e\}, X_2 = \{g, e, h\}, X_1 = \{c, b\}$$

结点包的连接方式如图 4 所示。

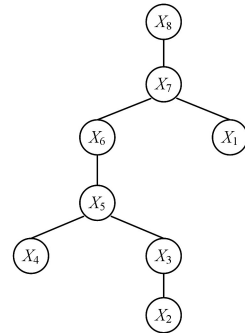


图 4 图 $G=(V,E)$ 的结点包序列

Fig. 4 Node package sequence of graph $G=(V,E)$

(3) 对部分结点包进行合并操作:

$$X_8 \subset X_7 \subset X_6 = \{a, b, f\}, 8 = \max\{6, 7, 8\}$$

$$X_3 \subset X_2 = \{g, e, h\}, 3 = \max\{2, 3\}$$

$$X_5 = \{e, a, f\}, X_4 = \{d, a, e\}, X_1 = \{c, b\}$$

$$I = \{1, 3, 4, 5, 8\}$$

合并相应包序列后, 对应的分解树如图 5 所示。

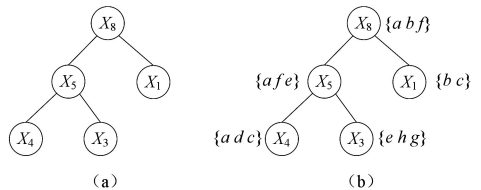


图 5 图 $G=(V,E)$ 的一棵分解树

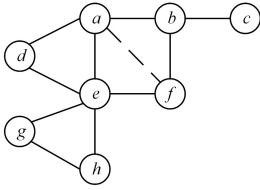
Fig. 5 Decomposition tree of graph $G=(V,E)$

(4) 消点序及弦图

$$\text{消点序: } \pi = \begin{pmatrix} a & b & c & d & e & f & g & h \\ 6 & 7 & 1 & 4 & 5 & 8 & 2 & 3 \end{pmatrix};$$

$$\text{弦图: } G_\pi^+ = (V, E \cup F), F = \{(a, f)\}.$$

消点序 π 是图 G_π^+ 的一个完美消点序, 图 6 给出了图 G 的弦图 G_π^+ 。

图6 图 $G=(V,E)$ 的弦图Fig. 6 Chordalgraph of graph $G=(V,E)$

4 二分图的树分解算法

设 $G=(X \cup C, E)$ 为一个二分图, 对于 $c \in C$, 记 $N(c)$ 为结点 c 在图中的邻接结点集。对于 C 的一个非空子集 C' , 假定 $\bar{C}' = C - C'$ 非空, 并记 $N(C') = \bigcup_{c \in C'} N(c) = \{x \in X: (\exists c \in C')[(x, c) \in E]\}$ 。非空子集 C' 及非空补集 $\bar{C}' = C - C'$, 形成 C 的一个划分。

考虑两个二分图 $G_{C'} = (X_{C'} \cup C', E_{C'})$, $G_{\bar{C}'} = (X_{\bar{C}'} \cup \bar{C}', E_{\bar{C}'})$, 其中 $X_{C'} = N(C')$, $X_{\bar{C}'} = N(\bar{C}')$ 。

在结点集 X 中, 定义公共邻接结点集为 $X_{C' \cap \bar{C}'} = N(C') \cap N(\bar{C}')$ 。

在算法 3 所示的递归分解算法中, 每次对 C 进行划分, 即表示在当前二分图 $G=(X \cup C, E)$ 中找一个 C' , 使得 $|N(C') \cap N(\bar{C}')|$ 最小。递归算法的边界条件为: $|N(C)| \leq 1$ or $|N(X)| \leq 1$ 。

算法 3 TreeDecompositionBigraph(G)

输入: 一个二分图 $G=(X \cup C, E)$

输出: 分解树 T_G , 以及一个正整数 d_G (即树宽)

- 如果 $|N(C)| \leq 1$ or $|N(X)| \leq 1$, 则 G 是一棵树, 对其进行平凡分解得到分解树 T_G , 并输出 $d_G = 1$; 然后终止 (记 T_G 的根节点为 $root_G$)。
- 否则, 在二分图 $G=(X \cup C, E)$ 中找一个 $C' = C^*$, 使得 $|N(C') \cap N(\bar{C}')|$ 最小。

2.1. 定义根节点: $root := N(C') \cap N(\bar{C}')$;

2.2. 调用 TreeDecompositionBigraph($X_{C'} \cup C', E_{C'}$), 返回 T_{G_1} , d_{G_1} , $root_{G_1}$;

2.3. 调用 TreeDecompositionBigraph($X_{\bar{C}'} \cup \bar{C}', E_{\bar{C}'}$), 返回 T_{G_2} , d_{G_2} , $root_{G_2}$;

3. 输出:

T_G ; $root_{G_1}$ 和 $root_{G_2}$ 作为 $root_G$ 的两个儿子节点;

$d_G = \max\{d_{G_1}, d_{G_2}, d-1\}$, 其中 $d = |N(C') \cap N(\bar{C}')|$ 。

上述算法可以得到二分图的一个树分解, 并且由树宽度的定义可知, 算法输出的正整数 d_G 是图的树宽度的一个上界。

5 CNF 公式因子图的树宽与公式来求解难度的分析

如果一个 CNF 公式的因子图是一棵树, 则称该公式为树公式。树公式有许多良好性质: 树公式的可满足性问题可以在多项式时间内求解; 树公式作为警示传播算法的输入公式, 算法是收敛的。由于树公式的因子图是树, 因此树宽度为 1,

我们试图寻找因子图的树宽度与 CNF 公式来求解难度之间的关系。

本节基于前文所提出的二分图的树分解算法, 通过实验来探究 CNF 公式因子图的树宽度与其求解难度的关系。模型 $G(n, 3, m)$ 按如下方式生成随机 3-SAT 实例: 1) 随机均匀地从所有可能的子句 (这样的子句共有 $2^3 \cdot \binom{n}{3}$ 个) 中选择 m ($m = an$) 个子句; 2) 选出的 m 个子句以合取方式连接。其中, n 表示变元数, m 表示子句数, 3 为子句长度。本文取两组子句约束密度、变元规模不同的数据进行数值实验分析, 分别取子句约束密度 $\alpha = 2, 2.5, 3, 3.5$, 变元分别为 $n = 100, 300, 500$ 。图中每个数据点由模型 $G(n, 3, m)$ 生成的 100 个随机实例构成。

在第 1 组生成的随机实例中, 子句约束密度 α 分别取 2, 2.5, 3, 3.5, 变元 $n \in [20, 500]$, 实例公式因子图的树宽上界和公式求解时间随公式子句约束密度 α 取值的变化趋势如图 7 所示。

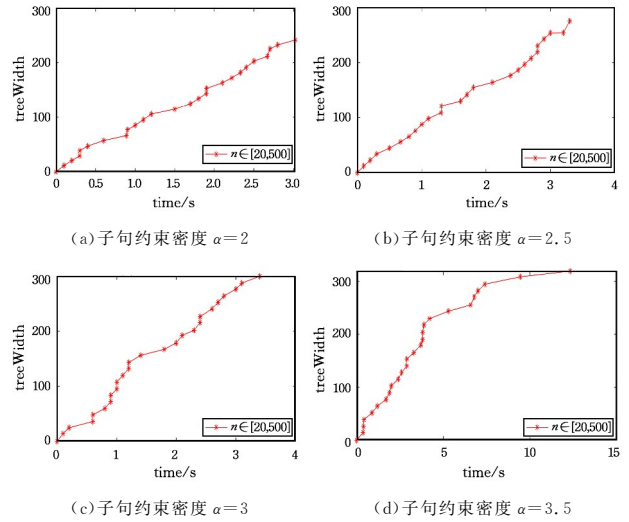
(a) 子句约束密度 $\alpha=2$ (b) 子句约束密度 $\alpha=2.5$ (c) 子句约束密度 $\alpha=3$ (d) 子句约束密度 $\alpha=3.5$

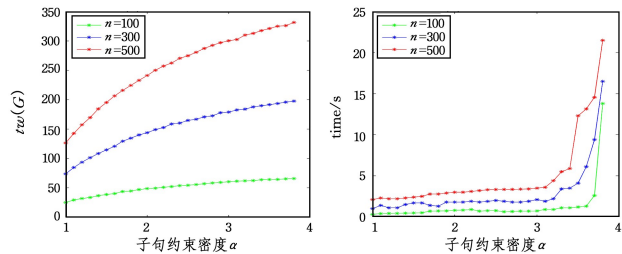
图 7 公式因子图的树宽与求解时间之间的变化关系

Fig. 7 Changing relationship between tree width of formula factor graph and solving time

图 7 中 4 个子图分别描述了当子句约束密度 $\alpha = 2, \alpha = 2.5, \alpha = 3$ 和 $\alpha = 3.5$ 时, 实例公式因子图的树宽上界与公式求解时间之间的关系变化趋势, 其中纵坐标为公式因子图的树宽 $tw(G)$, 横坐标为实例公式求解时间 $time$, 属性值即变元 n 的取值介于 20~500 之间, 并以 20 的幅度递增。从图中可以看出, 在子句约束密度一定的情况下, 公式因子图的树宽 $tw(G)$ 随公式求解时间的增加而增加; 在变元数量一定的情况下, 随着子句约束密度的增大, 公式的求解时间明显增加, 但公式因子图的树宽上界仍在一定范围内变化。

在第 2 组数据中, 变元 n 分别取 $n = 100, 300, 500$, 子句约束密度 $\alpha \in [1, 3.8]$, 随机 3-SAT 实例公式因子图的树宽上界公式求解时间随子句约束密度 α 的变化情形如图 8 所示。图 8(a) 描述了实例公式因子图的树宽上界随子句约束密度 α 变化的变化趋势, 其中纵坐标为树宽 $tw(G)$, 横坐标为子句约束密度 α , 变元 n 分别取 $n = 100, n = 300, n = 500$ 。图 8(b) 描述了实例公式的求解时间随子句约束密度 α 变化的变化趋势,

其中纵坐标为求解时间 $time$,横坐标为子句约束密度 α ,变元 n 分别取 $n=100, n=300, n=500$ 。从图 8 可以看出:当 $\alpha \leq 3.5$ 时,公式因子图的树宽上界随着子句约束密度 α 的增大而逐渐增加,且公式的求解时间较短;当 $3.5 < \alpha < 3.8$ 时,公式的求解时间突然增加,但公式因子图的树宽上界仍处于缓慢增长状态。



(a)公式因子图的树宽随子句约束密度的变化趋势 (b)公式求解时间随子句约束密度的变化趋势

图 8 树宽和求解时间随子句约束密度 α 的变化情形

Fig. 8 Trend of changes in tree width and solving time with clause constraint density α

表 1 不同树分解算法求解的树宽及执行时间的比较

Table 1 Comparison of tree widths solved with different tree decomposition algorithms and runtime

$G(n, 3, m)$	$tw(G)$				t/s			
	max-cardinality	min-degree	min-fill	本文	max-cardinality	min-degree	min-fill	本文
$G(50, 3, 100)$	42	28	26	25	0.32	0.37	0.42	0.34
$G(100, 3, 200)$	82	50	48	46	0.35	0.38	0.70	0.90
$G(150, 3, 300)$	115	74	72	71	0.44	0.74	0.87	1.73
$G(200, 3, 400)$	154	99	95	94	0.61	0.89	1.46	2.08
$G(50, 3, 125)$	48	29	28	26	0.36	0.45	0.46	0.41
$G(100, 3, 250)$	93	56	54	52	0.64	0.53	0.73	1.36
$G(150, 3, 375)$	135	84	82	79	0.71	0.70	1.18	2.26
$G(200, 3, 500)$	178	113	110	108	0.87	0.76	2.03	3.17
$G(50, 3, 150)$	52	31	31	30	0.99	0.40	0.54	0.54
$G(100, 3, 300)$	102	67	59	57	0.53	0.48	0.91	1.45
$G(150, 3, 450)$	148	92	90	89	0.79	0.83	1.23	2.76
$G(200, 3, 600)$	196	122	119	118	1.25	0.96	2.35	3.16
$G(50, 3, 175)$	55	37	33	31	1.07	0.61	1.14	0.48
$G(100, 3, 350)$	108	76	78	79	2.15	1.36	2.40	1.41
$G(150, 3, 450)$	159	107	97	93	2.31	1.30	2.87	3.43
$G(200, 3, 700)$	207	135	129	125	3.23	1.30	3.84	4.69

结束语 本文研究了无向图 $G=(V, E)$ 的树分解算法,给出了一种二分图树分解算法。基于此算法,可以求解出二分图的树宽度的上界。由于 CNF 公式的因子图对应一个二分图 $G=(V \cup C, E)$,树公式的因子图树宽为 1,并且树公式属于易解公式类,因此联想到当公式因子图的树宽受限并且不是很大时,公式的可满足性求解是容易的,进而将此算法应用于 CNF 公式所对应的因子图来探究难解实例与公式树宽的结构关系。实验结果表明,即使在随机 3-SAT 实例的难解区域内,实例公式对应的因子图的树宽上界也远小于公式的变元规模,这为我们研究难解实例的难解特征提供了一种视角。此外,因子图的分解树中度数较大的树节点和与之相邻的树节点中所包含的结点信息也反映了公式中存在的某些中心变元。对这些中心变元的消解也可为其他启发式求解算法判定公式的可满足性提供借鉴。

进一步,利用不同的树分解算法求解由模型 $G(n, 3, m)$ 生成的实例的树宽。实验结果表明,与其他树分解算法相比,

通过以上两组实验结果可得出:在模型 $G(n, 3m)$ 生成的随机 3-SAT 实例中,当变元规模一定时,随着子句约束密度 $\alpha=m/n$ 的增加,实例公式的求解难度增加,但公式因子图的树宽上界仍在一定范围内缓慢增长,即使在随机 3-SAT 实例的难解区域,其树宽大小也远小于实例公式的变元规模。这为我们进一步探究难解实例的结构信息奠定了一定的实验基础。

进一步,在相变点前的有解区域,使用模型 $G(n, 3m)$ 生成不同类型、不同规模的实例,利用不同的树分解算法求解 3-CNF 公式的因子图的树宽,以更加充分地验证本文算法的优越性。表 1 列出了 4 种算法在不同 3-SAT 实例中的平均树宽和平均执行时间。表 1 中,每一组生成 100 个随机实例, max-cardinality 表示利用最大势搜索算法进行树分解, min-degree 表示最小度算法, min-fill 表示最小填边数算法。由表 1 的实验结果可知,基于 min-fill 算法求解的平均树宽均优于 max-cardinality 算法和 min-degree 算法求解的平均树宽,而本文提出的二分图树分解算法求解的平均树宽又优于 min-fill 算法求解的平均树宽。

本文提出的算法能得到较优的树宽。

下一步,将从理论上研究限定树宽的 CNF 公式与其求解难度的关系,并从理论上证明,当公式的树宽度很小(如不超过 2)时,公式的可满足性问题求解是容易的。

参考文献

- [1] BODLAENDER H L, KOSTER A M C A. Treewidth computations II. Lower bounds[J]. Information and Computation, 2011, 209(7):1103-1119.
- [2] HAMMERL T, MUSLIU N, SCHAFFHAUSER W. Metaheuristic algorithms and tree decomposition[M]//Springer Handbook of Computational Intelligence. Berlin: Springer, 2015: 1255-1270.
- [3] BODLAENDER H L. A linear-time algorithm for finding tree-decompositions of small treewidth[J]. SIAM Journal on Computing, 1996, 25(6):1305-1317.
- [4] COOK S A. The complexity of theorem-proving procedures

- [C] // Proceedings of the Third Annual ACM Symposium on Theory of Computing. ACM, 1971: 151-158.
- [5] ABSEHER M, DUSBERGER F. Improving the efficiency of dynamic programming on tree decompositions via machine learning[J]. Journal of Artificial Intelligence Research, 2017, 58: 829-858.
- [6] TELLE J A, ROSKUROWSKI A. Algorithms for vertex partitioning problems on partial k -trees[J]. SIAM Journal on Discrete Mathematics, 1997, 10(4): 529-550.
- [7] BLIEM B, MOLDOVAN M. The Impact of Treewidth on ASP Grounding and Solving[C] // Twenty-sixth International Joint Conference on Artificial Intelligence. Melbourne, Australia, 2017: 852-858.
- [8] ZHANG N L. Computational properties of two exact algorithms for Bayesian networks[J]. Applied Intelligence, 1998, 9(2): 173-183.
- [9] DUBOIS O, MONASSON R, SELMAN B, et al. Phase transitions in combinatorial problems[J]. Theoretical Computer Science, 2001(265): 1-2.
- [10] XU K, LI W. Exact phase transitions in random constraint satisfaction problems[J]. Journal of Artificial Intelligence Research, 2000, 12(1): 93-103.
- [11] SZEIDER S. Generalizations of matched CNF formulas[J]. Annals of Mathematics and Artificial Intelligence, 2005, 43(1/2/3/4): 223-238.
- [12] SZEIDER S. Matched formulas and backdoor sets [C] // International Conference on Theory and Applications of Satisfiability Testing. Berlin: Springer, 2007: 94-99.
- [13] SURYNEK P. Compact representations of cooperative path-finding as SAT based on matchings in bipartite graphs[C] // 2014 IEEE 26th International Conference on Tools with Artificial Intelligence. IEEE, 2014: 875-882.
- [14] XU D Y, WANG X F. A Regular NP-Complete problem and its inapproximability[J]. Journal of Frontiers of Computer Science and Technology, 2013, 7(8): 691-697.
- [15] KRISHNAMURTHY S, SAHOO S. Balanced k -satisfiability and biased random k -satisfiability on trees[J]. Physical Review E, 2013, 87(4): 042130.
- [16] CHEN J. Parameterized Computation and Complexity: A New Approach Dealing with NP-Hardness[J]. Journal of Computer Science and Technology, 2005, 20(1): 18-25.
- [17] RAIBLE D, FERNAU H. A new upper bound for Max-2-SAT: A graph-theoretic approach[C] // International Symposium on Mathematical Foundations of Computer Science. Berlin: Springer, 2008(8): 551-562.
- [18] ROBERTSON N, SEYMOUR P D. Graph minors. X. Obstructions to tree-decomposition[J]. Journal of Combinatorial Theory (Series B), 1991, 52(2): 153-190.
- [19] GOTTLÖB G, LEONE N, SCARCELLO F. A comparison of structural CSP decomposition methods[J]. Artificial Intelligence, 2000, 124(2): 243-282.
- [20] CHEN H B. Quantified Constraint Satisfaction and Bounded Treewidth. [C] // European Conference on Artificial Intelligence. DBLP, 2004: 161-165.
- [21] BORDEAUX L, CADOLI M, MANCINI T. CSP properties for quantified constraints: Definitions and complexity[C] // AAAI. 2005: 360-365.
- [22] WANG X F, XU D Y, WEI L. Convergence of Warning Propagation Algorithms for Random Satisfiable Instances[J]. Journal of Software, 2013, 24(1): 1-11.
- [23] XU D Y, DENG T Y, ZHANG Q S. k -LSAT is NP-Complete for $k \geq 3$ [J]. Journal of Software, 2008(3): 511-521.
- [24] PORSCHE S, SPECKENMEYER E, RANDE RATH B. On linear CNF formulas[M] // Proc. of the 19th Int'l Conf. on Theory and Applications of Satisfiability Testing-SAT 2006. New York: Springer-Verlag, 2006: 212-225.
- [25] PORSCHE S, SPECKENMEYER E. NP-Completeness of SAT for restricted linear formulas classes [C] // Proc. of the Guangzhou Symp. & On Satisfiability in Logic-Based Modeling. 2006: 111-123.



LEI Ying, born in 1992, postgraduate, is a member of China Computer Federation. Her main research interest is include algorithm design and analysis.



XU Dao-yun, born in 1959, professor, Ph.D supervisor, is a senior member of China Computer Federation. His main research interests include computability and computational complexity.