图的树分解及其算法应用研究进展

高文宇 李绍华

(广东商学院信息学院 广州 510320)

摘 要 图的树宽和树分解是图子式理论中发展起来的两个重要概念。图的树分解由于其本身的特性使得它在算法设计中有着极其重要的意义。从图的树宽特性、图的树分解算法、图的树分解在复杂算法问题求解中的应用等方面对近年来的相关研究进展做了深入的分析和介绍,结合一些简洁的实例分析了一些重要的原理和方法,讨论了其中的一些问题,并给出了今后的一些研究方向。

关键词 图子式,树宽,树分解,参数算法,近似算法

Tree Decomposition and its Applications in Algorithms: Survey

GAO Wen-yu LI Shao-hua

(School of Computer Science, Guangdong University of Business Studies, Guangzhou 510320, China)

Abstract Tree width and tree decomposition are two important concepts developed by graph minor theory. Because of its own characteristics, tree decomposition plays an important role in algorithm design. The tree width of graph, tree decomposition algorithm, applications of tree decomposition algorithm for problem solving in a complex problems were deeply analysed. Three aspects of the related research in recent years were given a thorough analysis and presentation, and a number of important principles and methods were presented by some simple examples. Furthermore, a few future research issues were outlined.

Keywords Graph minor, Tree width, Tree decomposition, Parameterized algorithm, Approximation algorithm

1 引言

无论是在网络还是人工智能应用中,抑或社会经济领域,都存在着大量的组合优化问题,其中很多都是NP完全问题[1]。求解这些NP完全问题,目前还没有有效的确定性算法。过去在应用领域多采用近似算法求解,某些近似算法在实际应用中也确实能取得非常好的效果。近年来发展起来的参数计算理论[2,3]对一些NP问题的求解提供了一些新的思路。伴随着参数计算理论的发展,也产生了一些新的算法设计技巧和方法,如核心化、有限搜索树等。而在20世纪80年代发展起来的图子式理论(Graph Minor)[4-10]也为参数算法的研究提供了有力的支持。图子式理论从一开始就与算法理论密切相关,并对算法研究产生了深远的影响。一方面借助图子式理论可以用一种非构造的方式从理论上证明某些问题的参数算法的存在性;另一方面图子式理论也发展出一些有效的算法设计技术,其中图的树宽(Tree width)和图的树分解(Tree decomposition)是图子式理论中引入的两个重要概念。图的树宽和树分解的引入给算法的分析和设计带来了一些新的思路。某些NP问题,若将其限定在有限树宽的图中,则其求解是可行的。也就是说,通过对图进行树分解,若图的树宽是有限的,则存在有效的多项式求解算法,这对于很多场合的实际应用问题有着重要的意义。

2 图的树宽特性

2.1 图的树宽定义

图子式理论是20世纪80年代发展起来的一个重要的图理论分支,其理论体系散见于从1983年开始直至现在的20余篇论文中[10]。从一开始,图子式理论就受到了算法研究者的密切关注。参数计算研究的先驱Fellows等人在20世纪80年代末就研究了图子式理论在算法设计和分析中的应用,借助于该理论对众多组合优化问题的判定问题给出了其多项式判定算法的存在性证明[11],并研究了如何将判定算法转化为求最优解的构造性算法的方法[12,13]。

例如,对于最多叶子生成树问题[14],给定一个图集F,若F中的图没有哪一个具有一棵生成树,使得该生成树有大于等于k个叶子节点,则根据子式运算(Minor)的规则,图集F中任意图的子式也不会有叶子节点数大于等于k的生成树,这种特性称为图集F关于子式运算封闭。那么根据图子式定理,图集F的障碍集(Obstructions)是有限的,而且可以在多项式时间内识别出这些障碍集,因此最多叶子生成树问题可以在多项式时间内判定。

但遗憾的是,利用图子式理论进行的研究主要是证明了解决问题的时间复杂度上限,其方法是非构造性的理论证明,即可以证明这个多项式时间判定算法是存在的,但并不知道这个算法的任何细节。也就是说,我们需要去寻找一个可行

到稿日期:2011-04-13 返修日期:2011-07-18 本文受广东省自然科学基金(8151032001000013)资助。

高文宇(1972-),男,博士,教授,CCF高级会员,主要研究方向为算法理论及应用、计算机网络,E-mail:gwyy@163.com。

的构造性算法，而且利用子式运算的判定算法的复杂度中包含着极大的常数因子，这使得这类算法在实际应用中存在问题。

而伴随着图子式理论的建立也发展出很多有效的算法设计技术，众多研究者对图子式理论的算法应用做了深入的研究和介绍[15-17]。其中图的树宽和图的树分解是图子式理论中对算法研究有着深远影响的两个概念。

定义 1(图的树分解及树宽)　设 $G(V,E)$ 是一个无向简单图(无自环、无重边)，则图 G 的树分解由树 T 和 T 的每一个节点 t 关联的子集 $X_t \subseteq V$ 构成(称这些子集 X_t 是树分解的片段)。树 T 和片段集$\{X_t: t\in T\}$满足以下 3 个条件：

(1)$\bigcup(X_t: t\in T)=V$，即全部片段集 X_t 中包含的节点涵盖了图 G 的所有节点，或者说图 G 的每一个节点至少属于某一个片段 X_t。

(2)对图 G 的每一条边 $e\in E$，至少存在一个片段 X_t 包含 e 的两个端点。

(3)若 $t1,t2,t3$ 是树 T 的 3 个节点，其中 $t2$ 在 $t1$ 到 $t3$ 的路径上，那么，若 G 的节点 v 属于 X_{t1} 和 X_{t3}，则 v 也一定属于 X_{t2}。

树分解的宽度等于 $\max(|X_t|-1: t\in T)$。图 G 的树宽就是 G 的最小树分解的宽度，即在图 G 的所有树分解中，具有最小宽度的树分解的宽度称为图 G 的树宽。

图的树宽和弦图有着密切的关系。事实上，借助于弦图可以给出树宽的一个等价定义，即给定图 G，则图 G 的树宽为图 G 的所有超弦图 H 中最小的一个的最大团的大小减 1。

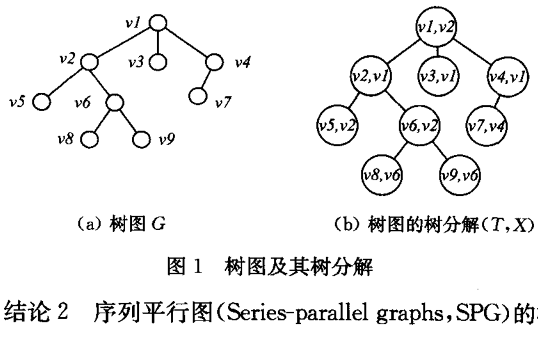
任意给定一个图 G，都存在着一个平凡的树分解，该树分解仅包含一个片段，该片段包含图 G 的所有节点。很容易验证这个平凡的树分解是满足树分解定义中的 3 个条件的。但是这个平凡的树分解对于算法问题的求解没有任何作用，因此我们需要找到非平凡的树分解，而且是宽度尽量小的树分解。

2.2　一些特殊图的树宽

结论 1　树的树宽为 1。

证明：对于一个满足树的定义的图 G(简称树图 G)，可以按如下方式来构造一个树分解(T,X)，树分解中的树 T 与树图 G 的结构完全一样，每个 X_i 包含 T 和树图 G 对应位置的节点及其父节点，T 的根节点对应的 X_i 包含树图 G 的根节点及其任意一个孩子节点。容易验证这样的树分解满足图的树分解定义中的 3 个条件，而树分解中每个 X_i 都只包含 2 个节点，因此这个树分解的宽度为 1。

图 1 给出了一个示例，图 1(a)是一个树图 G，图 1(b)是该树图的树分解(T,X)。



(a) 树图 G　　(b) 树图的树分解(T,X)

图 1　树图及其树分解

结论 2　序列平行图(Series-parallel graphs，SPG)的树宽为 2。

所谓序列平行图可以定义如下：仅包含一条边的图是一个序列平行图。任意给定一个序列平行图，可以通过如下方式得到一个新的序列平行图，即在原图中增加某一条边的平行边，或将原有的某一条边从中间分开，在中间添加一个新的节点。因此所有的序列平行图都可以从最初的一条边通过不断地添加平行边和分割边来产生。

证明：(1)因为任何 SPG 的生成都可以由一条边通过增加平行边和分割一条边这两个操作来得到，所以增加平行边对树分解无影响。因为平行边所依附的两个节点是一样的，所以包含平行边和不包含平行边的图的树分解是一样的。

因此下面考虑边的分割操作。

(2)采用数学归纳法。假设某个 SPG 的树宽为 2，则只需证明对该图任意一条边进行分割操作后，新的 SPG 图的树宽还是 2。

(3)假设(T,X)是图 G 的一个树分解，树宽为 2，$u-v$ 为图 G 的一条边。现在对边 $u-v$ 进行分割操作，即 $u-v$ 变成两条相邻的边 $u-w$ 和 $w-v$，以及一个新的节点 w，得到新图 G'。

(4)设 r 是原图 G 的树分解(T,X)中的树 T 上的一个节点，且 $u,v\in X_r$。现在构造一棵新的树 T'，T'包含 T，并且增加一条边 $r-q$，q 是一个新节点。在树 T'中，令 $Y_t=X_t(\forall t\in V(T))$，且 $Y_q=\{u,v,w\}$，容易验证(T',Y)是图 G'的一个树分解，且该树分解的宽度也是 2。因为树 T'的其它节点的基数没变，而新增节点 q 的基数为 3(因为 $Y_q=\{u,v,w\}$)，所以树分解(T',Y)的宽度仍为 2。

结论 3　m-网格图(m-grid)的树宽为 m。

证明略。

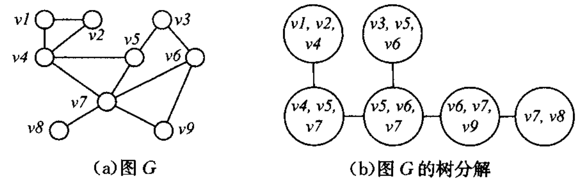
网格图在图子式理论及算法应用中都有着重要的地位，因为任意 n 个节点的平面图一定是 m-grid 的子式($m=n*n$)。

从前面对几种特殊图的分析中可以看出，图的某些约束和特性与图的树宽有着密切的联系。类似的结论还包括确定弦图(chordal graph)、排列图(permutation graph)等的树宽。

3　图的树宽及树分解在算法设计中的应用

树分解在众多领域有着重要的意义，在人工智能和概率推理领域称为团树(Clique tree)，在关系数据库领域称为连接树(Join tree)。

图 2 展示了一个图 G(见图 2(a))和它的一个树分解(见图 2(b))，该树分解的宽度为 2。



(a)图 G　　(b)图 G 的树分解

图 2　图及图的树分解

结合图 2 来分析树分解的特性。树分解的意义在于将一个图转换成一个类似于树的结构。我们知道，在一棵树中删去一条边，则原来的树被分成两个连通分量；若在树中删去一个节点，则原来的树被分成若干个连通分量(连通分量的个数等于该节点的度)。那么对于图 G 的树分解，若在图 G 中删去树分解中某个片段包含的节点(如图 2(b)中片段$\{v4,v5,$

v7}包含的 3 个节点)，则图 G 被分成若干个不相交的连通分量；若在图 G 中删去树分解的树 T 中某条边关联的两个片段的交集(如图 2(b)中片段$\{v6,v7,v9\}$和片段$\{v7,v8\}$的交集)，则图 G 被分成至少两个不相交的连通分量。

从树分解的特性可以看出，通过树分解可以将原图 G 的节点划分为有限相关的节点集。这样在面对某些图问题时，可以将图分解成若干部分，然后在每部分上独立求解，最后再结合起来。因此利用树分解求解一些复杂的图问题(甚至是 NP 难问题)的步骤就是首先对图进行树分解(若树分解得到的宽度较小)，然后采用动态规划求解原问题。

在文献[18-24]中研究了如何利用树分解来求解一些组合优化问题，在文献[25,26]中探讨了利用树分解来解决一些计算生物学中的问题，在文献[27,28]中利用树分解来求解受约束的可满足性问题。在人工智能和概率推理领域，树分解又称为团树。文献[29-32]讨论了利用树分解来提高推理效率。

利用树分解来求解一些复杂组合优化问题，通常的做法是首先选定树分解(T,X)中树 T 的一个节点 r 作为根节点，对任意 $t\in T$，用 T_t 表示 T 中以 t 为根的子树，而节点集$\bigcup\{X_v: v\in V(T_t)\}$中的节点只能通过 X_t 与图 G 中的其余节点相连。若树分解(T,X)的宽度为 w，则 X_t 中最多包含$(w+1)$个节点。设 $Y_t=\bigcup\{X_v: v\in V(T_t)\}$，用 G_t 表示由 Y_t 诱导的 G 的子图。例如对于点覆盖问题[33]，若已知一个图 G 的树分解(T,X)，则该树分解的宽度为 w(w 为一较小的常数)。对于 X_t 的每一个子集 Z，用 $f(Z,t)$表示 G_t 包含 Z 的最小点覆盖集的势，则一个有着 2^{w+1} 项的表可以用来记录每一个 t 的这些信息。不难看出，沿着树 T 自底向上可以很方便地更新这些表。而根据树根 r 对应的表求 $\min\{f(Z,r): Z\subseteq V_r\}$，就能得到点覆盖问题的解。

4 图的树分解算法

从前面的分析中可知，利用树分解能在某些限定条件下(有限树宽)有效地解决一些复杂的问题。而且现实世界中的很多问题所涉及的图的规模虽然很大，但常常具有有限的、很小的树宽。例如一个 1000 个节点的网络，其树宽可能仅为 4[34]。在这种情况下，我们利用树分解可以很容易地解决以该网络为背景的各种复杂问题。

然而，对于一个给定的图如何得知它的树宽以及对应的树分解呢？对某些特殊图，很容易得到它们的树宽。如树的树宽为 1，序列平行图的树宽为 2，m-网格图的树宽为 m。但对于一般的图，计算图的树宽是 NP 难的。不过根据图子式理论，树宽不超过 k 的图集是子式运算封闭的，因此是多项式时间可检验的。文献[35]中给出了一个树分解算法，对于给定的常数 k，该算法可以在线性时间下判定一个图的树宽要么大于 k，要么给出该图的一个树分解，使得树分解的宽度小于等于 k。然而在实际应用中，该算法中隐藏在线性时间复杂度中的巨大的常数因子使得其本身并不可行。文献[36]通过实验也验证了这一点。

那么在实际应用中又要如何计算树宽呢？通常可以采用启发式算法构造树分解，从而得到近似最优的树宽。事实上，求解图的树宽的近似算法也是多年来的一个热点研究领域。一方面树分解在众多研究领域都有着重要的用途，另一方面求解近似树宽时涉及到的一些问题本身也是 NP 难的。因此，如何构造一个更有效的树分解算法，也是一个具有挑战性的开放问题。在过去的研究中，构造树分解的启发式方法通常有如下两类。

第一类是利用消元顺序(Elimination ordering)来构造树分解的启发式算法。所谓消元(Elimination)就是在无向图 G 中删除一个节点 v 以及与节点 v 关联的边，同时在剩下的图中补充一些边，使得 v 原来的邻节点能构成一个团(Clique)。通过在初始的图 G 中逐个消元图 G 的所有节点，就可以得到一个消元顺序，根据这个消元顺序便能构造出图 G 的一个树分解。问题的关键在于不同的消元顺序可以得到不同的树分解，而求解最优消元顺序也是 NP 难的，所以也需要采用启发式方法，如最大势搜索(Maximum Cardinality Search)[37]及其改进算法[38]、词典广度优先搜索(Lexicographic Breadth First Search)[39,40]、最小缺边搜索(Minimum Deficiency Search, MDS)[41,42]，以及其它一些利用消元顺序的启发式算法[43-46]。

第二类是利用割集(Separator)来构造树分解的启发式算法，即通过在初始图中寻找一些割集来构造树分解。所谓割集就是连通图 G 的节点集的一个子集，记为 S，若在原图 G 中删除 S 所包含的节点和关联的边，则剩余图变为至少两个连通分量。这样的节点集 S 称为图 G 的一个割集。

由于图的割集可以将图分割为两个以上的连通分量，因此可以利用割集递归地分割图，并在分割后的连通分量里构造树分解，最终把不同连通分量中的树分解合并在一起。事实上，寻找割集涉及到图的一些全局特性，因此这类启发式方法通常较利用消元顺序的方法更为有效，但同时也更复杂。文献[47-54]中的算法都是这一类型。

结束语 图的树宽和图的树分解是图子式理论中发展起来的重要概念，在其他研究领域有着类似的概念，如概率推理中的团树、数据库中的联合树等。图的树分解由于其本身的特性使得它在算法设计中有着极其重要的意义。本文对近年来图的树分解算法、图的树分解在复杂算法问题求解中的应用研究进展做了一个简要的介绍。在今后的研究中，我们可以从以下几个方面找到一些有价值的研究问题。

(1)图的树宽与图的其它特性之间的联系。虽然计算图的树宽是 NP 难的问题，然而一些特殊图的树宽却是确定的，如前面提到的树、m-网格图等。因此可以研究一些特殊图的树宽，例如节点的度受限的图的树宽、包含或者不包含某些特定子式的图的树宽，通过分析这些特殊图的树宽特性来发现一些影响图的树宽的因素。然后逐步建立起图的树宽与图的其它特性的联系，例如图的树宽与图中节点的度，与图中的割点、割集，与图中的团，与图的生成树的叶子数等之间的联系，因为这能帮助我们在某些特定的应用场合快速地判断图的树宽。另外，研究图的树宽特性有助于从“广义上”对图的连通性形成更为深刻的理解。图的树分解将图变成了一棵“树”，因此给涉及连通性的问题带来了更多的求解方法。通过对图的树宽特性的分析和研究也可以帮助我们发展出一些新的参数算法设计和分析的技巧。

(2)图的树分解算法。求图的树宽是一个 NP 问题，但现在已有一些图的树分解算法，能对给定的图构造出一个树分解，虽然不一定是最优的，但在实际应用中有重要的用途。分析和比较现有树分解算法有重要的应用意义。

在充分研究图的树宽特性的基础上，应设计更好的树分解算法，以更好地逼近图的树宽。同时从理论上分析树分解算法的性能，即算法构造出的树分解所得的“宽度”的上、下界以及与图的树宽的近似比。这些分析有助于确定实际问题中后继求解算法的可行性。另外，还可以利用图的一些特性设计一些图的预处理方法，从而提高树分解的效率和性能。

(3)图的树分解在复杂算法问题中的应用。前面提到，在实际应用背景中很多图的树宽都是较小的，因此利用树分解来求解一些复杂算法问题是非常有意义的。对图的覆盖、支配问题的连通性变形问题如连通点覆盖、连通支配集等，由于受到连通性这样的全局特性的约束，使得一些常规的算法优化技术难以实施。然而，可以通过树分解将原问题分割为有限相关的若干子问题，然后采用动态规划求解。

参考文献

[1] Garey M R, Johnson D S. Computers and intractability: a guide to the theory of NP-completeness[M]. New York: W. H. Freeman & Co., 1979

[2] Downey R G, Fellows M R. Parameterized Complexity[M]. New York: Springer, 1999

[3] Chen J. Parameterized computation and complexity: A new approach dealing with NP-hardness[J]. Journal of Computer Science and Technology, 2005, 20: 18-37

[4] Robertson N, Seymour P D. Graph Minors. I. Excluding a Forest [J]. Journal of Combinatorial Theory, Series B, 1983, 35: 39-61

[5] Robertson N, Seymour P D. Graph Minors. II. Algorithmic Aspects of Tree-Width[J]. Journal of Algorithms, 1986, 7(3): 309-322

[6] Robertson N, Seymour P D. Graph Minors. III. Planar Tree-Width[J]. Journal of Combinatorial Theory, Series B, 1984, 36: 49-64

[7] Robertson N, Seymour P D. Graph minors. IV. Tree-width and well-quasi-ordering[J]. Journal of Combinatorial Theory, Series B, 1990, 48: 227-254

[8] Robertson N, Seymour P D. Graph minors. X. Obstructions to tree-decomposition[J]. Journal of Combinatorial Theory, Series B, 1991, 52: 153-190

[9] Robertson N, Seymour P D. Graph minors. XXI. Graphs with unique linkages[J]. Journal of Combinatorial Theory, Series B, 2009, 99: 583-616

[10] Robertson N, Seymour P D. Graph minors. XXIII. Nash-Williams's immersion conjecture[J]. Journal of Combinatorial Theory, Series B, 2010, 100: 181-205

[11] Fellows M R. Non-constructive Tools for Proving Polynomial-time Decidability[J]. Journal of the ACM, 1988, 35(3): 727-739

[12] Brown D J, Fellows M R, Langston M A. Polynomial-time self-reducibility: theoretical motivations and practical results[J]. International Journal of Computer Math, 1989, 31: 1-9

[13] Bodlaender H L. Improved self-reduction algorithms for graphs with bounded treewidth [J]. Discrete Applied Mathematics, 1994, 54: 101-115

[14] Fellows M R, Langston M A. On well-partial-order theory and its application to combinatorial problems of VLSI design[J]. SIAM J. DIscrete Math, 1992, 5(1): 117-126

[15] Bienstock D, Langston M A. Algorithmic Implications of the Graph Minor Theorem

[16] Demaine E D, Hajiaghayi M T, Kawarabayashi K. Algorithmic Graph Minor Theory: Decomposition, Approximation, and Coloring[C]// Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science. 2005

[17] Lovasz L. Graph minor theory[J]. Bulletin (new series) of the American Mathematical Society, 2005, 43(1): 75-86

[18] Arnborg S, Proskurowski A. Linear time algorithms for NP-hard problems restricted to partial k-trees[J]. Disc. Appl. Math., 1989, 23: 11-24

[19] Bern M W, Lawler E L, Wong A L. Linear time computation of optimal subgraphs of decomposable graphs[J]. J. Algorithms, 1987, 8: 216-235

[20] Koster A, Hoesel S, Kolen A. Solving partial constraint satisfaction problems with tree decomposition[J]. Networks, 2002, 40 (3): 170-180

[21] Telle J A, Proskurowski A. Algorithms for vertex partitioning problems on partial k-trees[J]. SIAM J. Disc. Math., 1997, 10: 529-550

[22] Wimer T V, Hedetniemi S T, Laskar R. A methodology for constructing linear graph algorithms[J]. Congressus Numerantium, 1985, 50: 43-60

[23] Guo J, Niedermeier R. Exact algorithms and applications for Tree-like Weighted Set Cover [J]. Journal of Discrete Algorithms, 2008, 4(4): 608-622

[24] Guo J, Hüffner F, Kenar E, et al. Complexity and exact algorithms for vertex multicut in interval and bounded treewidth graphs[J]. European Journal of Operational Research, 2008, 186: 543-553

[25] Zhao J, Che D, Cai L. Comparative pathway annotation with protein-DNA interaction and operon information via graph tree decomposition[C]// Proceedings of Pacific Symposium on Biocomputing. volume 12, 2007: 496-507

[26] Zhao J, Malmberg R L, Cai L. Rapid ab initio prediction of RNA pseudo-knots via graph tree decomposition[J]. Journal of Mathematical Biology, 2008, 56(1/2): 145-159

[27] Chen H. Quantified constraint satisfaction and bounded treewidth[C]// de Mantaras R L, Saitta L, eds. Proceedings of the 16th Eureopean Conference on Artificial Intelligence. 2004: 161-165

[28] Gottlob G, Leone N, Scarcello F. A comparison of structural csp decomposition methods[J]. Acta Informatica, 2000, 124: 243-282

[29] Shafer G, Shenoy P. Probability propagation [J]. Annals of Mathematics and Artificial Intelligence, 1990, 2: 327-352

[30] Madsen A L, Jensen F V. Lazy propagation in junction trees[C]// Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence. 1998: 362-369

[31] Zhang N L. Computational properties of two exact algorithms for Bayesian networks[J]. Applied Intelligence, 1998, 9(2): 173-183

[32] Lauritzen S J, Spiegelhalter D J. Local computations with probabilities on graphical structures and their application to expert systems[J]. The Journal of the Royal Statistical Society, Series B (Methodological), 1988, 50: 157-224

[33] Bodlaender H L, Koster A. Treewidth computations I. Upper bounds[J]. Information and Computation, 2010, 208: 259-275

[34] Kleinberg J, Tardos E. Algorithm Design[M]. Boston: Addison-Wesley, 2005

[35] Bodlaender H L. A linear time algorithm for finding tree-decompositions of small treewidth[J]. SIAM J. Computer, 1996, 25: 1305-1317

[36] Rohrig H. Tree decomposition: A feasibility study [J]. Saarbrucken. Germany: Max-Planck-Institut fur Informatik, 1998

[37] Tarjan R E, Yannakakis M. Simple linear time algorithms to test chordiality of graphs, test acyclicity of graphs, and selectively reduce acyclic hypergraphs[J]. SIAM J. Comput., 1984, 13: 566-579

[38] Berry A, Blair J, Heggernes P, et al. Maximum cardinality search for computing minimal triangulations of graphs[J]. Algorithmica, 2004, 39: 287-298

[39] Rose D J, Tarjan R E, Lueker G S. Algorithmic aspects of vertex elimination on graphs[J]. SIAM J. Comput., 1976, 5: 266-283

[40] Villanger Y. Lex M versus MCS-M[J]. Disc. Math., 2006, 306: 393-400

[41] Bertele U, Brioschi F. Nonserial dynamic programming [M]. New York: Academic Press, 1972

[42] Hell P, Kirkpatrick D G. Algorithms for degree constrained graph factors of minimum deficiency[J]. Journal of Algorithms, 1993, 14(1): 115-138

[43] Becker A, Geiger D. A sufficiently fast algorithm for finding close to optimal clique trees[J]. Artificial Intelligence, 2001, 125 (1): 3-17

[44] Lauritzen S L, Jensen F V. Local computation with valuations from a commutative semi group[J]. Annals of Mathematics and Artificial Intelligence, 1997, 21: 51-69

[45] Meila M, Jordan M I. Triangulation by continuous embedding [M]. Advances in Neural Information Processing System. Cambridge: MIT Press, 1997

[46] Meila M, Jordan M I. An objective function for belief net triangulation[C]// Proceedings of the 1997 Conference on Artificial Intelligence and Statistics. Ft. Lauderdale, FL, 1997

[47] Reed B. Finding approximate separators and computing treewidth quickly[C]// Proceedings of the 24th Annual Symposium on Theory of Computing. New York: ACM Press, 1992: 221-228

[48] Koster A. Frequency Assignment-Models and Algorithms[D]. Maastricht, The Netherlands: Univ. Maastricht, 1999

[49] Amir E. Efficient approximation for triangulation of minimum treewidth[C]// Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence. 2001: 7-15

[50] Bodlaender H L, Gilbert J R, Hafsteinsson H, et al. Approximating treewidth, pathwidth, frontsize, and minimum elimination tree height[J]. J. Algorithms, 1995, 18: 238-255

[51] Diestel R, Jensen T R, Gorbunov K Y, et al. Highly connected sets and the excluded grid theorem[J]. J. Comb. Theory, Series B, 1999, 75: 61-73

[52] Kloks T. Treewidth, Computations and Approximations [C]// Lecture Notes in Computer Science, Vol. 842. Berlin: Springer-Verlag, 1994

[53] Bouchitte V, Kratsch D, Muller H, et al. On treewidth approximations[J]. Disc. Appl. Math., 2004, 136: 183-196

[54] Amir E. Approximation algorithms for treewidth[J]. Algorithmica, 2010, 56(4): 448-479

(上接第 13 页)

[32] Wang S, Rundensteiner E A, Mani M. Optimization of nested XQuery expressions with orderby clauses[J]. Data & Knowledge Engineering, 2007, 60(2): 303-325

[33] Re C, Simeon J, Fernandez M. A complete and efficient algebraic compiler for XQuery[C]// 22nd International Conference on Data Engineering(ICDE '06). Atlanta, GA, United states: Institute of Electrical and Electronics Engineers Computer Society, April 2006

[34] Chun Z, Naughton J, Dewitt D, et al. On supporting containment queries in relational database management systems[C]// 2001 ACM SIGMOD International Conference on Management of Data. ACM, May 2001

[35] Al-Khalifa S, Jagadish H V, Koudas N, et al. Structural joins: a primitive for efficient XML query pattern matching[C]// Proceedings 18th International Conference on Data Engineering. Los Alamitos, CA, USA: IEEE Comput. Soc, Feb. 2002

[36] Bruno N, Koudas N, Srivastava D. Holistic twig joins: Optimal XML pattern matching[C]// ACM SIGMOD 2002 Proceedings of the ACM SIGMOD International Conference on Managment of Data, Madison, WI, United States: Association for Computing Machinery, June 2002

[37] Jiang H, Wang W, Lu H, et al. Holistic twig joins on indexed XML documents[C]// VLDB '2003. VLDB Endowment, 2003

[38] Lu J, Chen T, Ling T W. Efficient processing of XML twig patterns with parent child edges: A look-ahead approach[C]// Proceedings of the Thirteenth ACM Conference on Information and Knowledge Management(CIKM 2004). Association for Computing Machinery, November 2004

[39] Lu J, Chen T, Ling T W. TJFast: Effective processing of XML twig pattern matching[C]// 14th International World Wide Web Conference(WWW2005). Chiba, Japan: Association for Computing Machinery, May 2005

[40] Chen T, Lu J, Ling T W. On boosting holism in XML twig pattern matching using structural indexing techniques[C]// SIGMOD '05. New York, NY, USA: ACM, 2005

[41] Chen S, Li H, Tatemura J, et al. Twig2Stack: bottom-up processing of generalized-tree-pattern queries over XML documents [C]// Proceedings of the 32nd International Conference on Very Large Data Bases. Seoul, Korea: VLDB Endowment, 2006

[42] Qin L, Yu J X, Ding B. TwigList: Make twig pattern matching fast[C]// 12th International Conference on Database Systems for Advanced Applications (DASFAA 2007). Bangkok, Thailand: Springer Verlag, April 2007

[43] Wu X, Liu G Q. XML twig pattern matching using version tree [J]. Data & Knowledge Engineering, 2008, 64(3): 580-599

[44] Jiang H, Lu H, Wang W. Efficient processing of XML twig queries with OR-predicates[C]// SIGMOD '04. New York, NY, USA: ACM, 2004

[45] Lu J H, Ling T W, Bao Z F, et al. Extended XML Tree Pattern Matching: Theories and Algorithms[J]. IEEE Transactions on Knowledge and Data Engineering, 2011, 23(3): 402-416