

电子科技大学
UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

博士学位论文
DOCTORAL DISSERTATION



论文题目 反馈集相关问题的
参数与精确算法研究

学科专业 计算机科学与技术
学 号 201611260104
作者姓名 白 天
指导教师 肖鸣宇 教 授
学 院 计算机科学与工程学院
(网络空间安全学院)

分类号 TP301.6 密级 公开
UDC^{注1} 519.1

学 位 论 文

反馈集相关问题的参数与精确算法研究

(题名和副题名)

白 天

(作者姓名)

指导教师 肖鸣宇 教 授
电子科技大学 成 都

(姓名、职称、单位名称)

申请学位级别 博士 学科专业 计算机科学与技术
提交论文日期 2024 年 4 月 10 日 论文答辩日期 2024 年 5 月 25 日
学位授予单位和日期 电子科技大学 2024 年 6 月
答辩委员会主席
评阅人

注 1：注明《国际十进分类法 UDC》的类号。

Investigations Concerning Parameterized and Exact Algorithms for the Feedback Set and Related Problems

A Doctoral Dissertation Submitted to
University of Electronic Science and Technology of China

Discipline: **Computer Science and Technology**
Student ID: **201611260104**
Author: **Bai Tian**
Supervisor: **Prof. Xiao Mingyu**
School: **School of Computer Science and Engineering**
(School of Cyber Security)

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

作者签名：_____ 日期： 年 月 日

论文使用授权

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，同意学校有权保留并向国家有关部门或机构送交论文的复印件和数字文档，允许论文被查阅。本人授权电子科技大学可以将学位论文的全部或部分内容编入有关数据库进行检索及下载，可以采用影印、扫描等复制手段保存、汇编学位论文。

（涉密的学位论文须按照国家及学校相关规定管理，在解密后适用于本授权。）

作者签名：_____ 导师签名：_____

日期： 年 月 日

摘要

反馈集问题和子集反馈集问题是计算机科学领域中基础的 NP-完全图论问题。在生物信息学、编码理论、选举理论等多个领域，它们均是处理复杂系统中循环依赖和冲突的关键数学模型。反馈集问题旨在 n 阶图中找出大小不超过 k 的点集或边集，使得将其删除后的图中没有环路。子集反馈集问题是反馈集问题的推广形式，额外给定一个点子集或边子集（称作关键集），该问题旨在找出大小不超过 k 的点集或边集，使得将其删除后的图中不存在经过关键元素的环路。随着社交网络和人工智能技术的发展，如何在可接受的指数时间复杂度内实现精确求解成为了大数据时代面临的核心挑战。

本文系统性地开展了反馈集和子集反馈集问题的计算复杂性及参数与精确算法研究。一方面，本文解决了遗留数十年的计算复杂性分类难题。另一方面，本文对三个重要图类上子集反馈点集问题开展了深入的参数与精确算法研究，它们均为著名的隐式 3-碰撞集问题。3-碰撞集问题能否在 $\mathcal{O}^*(2^k)$ 时间内求解是一个著名的公开性难题，而本文为这三类隐式 3-碰撞集问题所设计的参数算法均突破了 2^k 这一关键时间复杂度瓶颈。整体上，本研究共取得了以下四方面的创新性理论成果。

（1）反馈集相关问题的计算复杂性分类

有向图以及平面有向图上反馈集问题的计算复杂性分类工作长期进展缓慢。当最大度为 3、4 和 5 时，有向图和平面有向图上反馈集问题的计算复杂性是一个长期未完全解决的公开性难题。本文引入非正规平面嵌入的概念，完备了关于最大度的计算复杂性分类。此外，本文全面整理了各种不同版本的子集反馈集问题及其归约关系。在此基础上，完成了子集反馈点集问题关于关键集大小的计算复杂性分类。

（2）竞赛图上的子集反馈点集问题的参数与精确算法

针对竞赛图上子集反馈点集问题，本文结合亚指数时间的解集分离技术、迭代压缩技术、基于分割块的分支技术以及基于平衡割的分治技术，成功将参数算法时间复杂度从 $\mathcal{O}^*(2.0755^k)$ 改进至 $\mathcal{O}^*(1.6181^k)$ ，将精确算法的时间复杂度从 $\mathcal{O}(1.5182^n)$ 改进至 $\mathcal{O}(1.3820^n)$ 。这一结果不仅首次将参数算法的时间复杂度突破了 2^k ，还与竞赛图上反馈点集问题的参数与精确算法的运行时间上界持平。此外，本文也改进了竞赛图上限制版子集反馈点集问题的参数与精确算法。

（3）分裂图上的子集反馈点集问题的参数与精确算法

针对分裂图上子集反馈点集问题，本文通过 Dulmage-Mendelsohn 分解，设计了基于分支搜索技术的参数算法。采用度量治之分析方法，本文证明了该算法时间复杂度为 $\mathcal{O}^*(1.8192^k)$ ，首次突破 2^k 。基于此，本文给出了运行时间为 $\mathcal{O}(1.3488^n)$ 的精确算法。此外，本文研究了超图上带惩罚的独立集问题，通过将其归约至分裂图上子集反馈点集问题，给出了运行时间为 $\mathcal{O}^*(1.8192^n)$ 的精确算法，成为了首个非平凡精确算法。同时，本文采用动态规划和分支搜索等技术也改进了分裂图上限制版子集反馈点集问题的参数与精确算法。

（4）弦图上的子集反馈点集问题的参数与精确算法

针对弦图上子集反馈点集问题，本文创新性地提出树分解上的分治算法，建立了分裂图和弦图上子集反馈点集问题的联系，将参数算法时间复杂度从 $\mathcal{O}^*(2^k)$ 改进至 $\mathcal{O}^*(1.8192^k)$ ；将精确算法时间复杂度从 $1.5^n 2^{o(n)}$ 改进至 $\mathcal{O}(1.3788^n)$ 。特别地，本文提出的参数算法首次突破了 2^k 的时间复杂度。此外，采用类似的分治思想，本文也改进了弦图上限制版子集反馈点集问题的参数与精确算法。

关键词：参数算法，精确算法，反馈集问题，子集反馈集问题，隐式碰撞集问题

ABSTRACT

Feedback set and subset feedback set problems are two fundamental NP-complete graph problems in computer science. In various research fields such as bioinformatics, coding theory, and election theory, these problems play pivotal roles in essential mathematical models for addressing circular dependencies and conflicts in complex systems. The feedback set problem aims to find a set of vertices or edges of size at most k in an n -order graph, whose removal makes the graph without cycles. The subset feedback set problem, a generalization of the feedback set problem, further gives a subset of vertices or edges/arcs (called the terminal set) and aims to remove at most k vertices or edges/arcs such that no terminal is contained in a cycle in the remaining graph. The development of social networks and artificial intelligence technologies highlights the importance of computing exact solutions with an acceptable exponential time complexity in the era of big data.

This dissertation systematically investigates the computational complexity and explores parameterized and exact algorithms for feedback set and subset feedback set problems. It addresses the longstanding challenge of computational complexity classification and provides in-depth research on parameterized and exact algorithms for subset feedback vertex set problems (SFVS) in three important graph classes, all well-known implicit 3-hitting set problems. Whether 3-hitting set problems can be solved in $\mathcal{O}^*(2^k)$ time is a famous open problem. This dissertation presents parameterized algorithms for these three implicit 3-hitting set problems breaking the crucial barrier 2^k . Overall, this research achieves innovative theoretical results in the following four aspects.

(1) Computational complexity classification of feedback set and related problems.

This work addresses the long-standing challenge of classifying the computational complexity of feedback set problems in (planar) digraph graphs, particularly for graphs with maximum degrees of 3, 4, and 5. Through the proposed concept of irregular planar embedding, we settle the classification of computational complexity regarding the maximum degree. Based on systematically organizing the reduction among various versions of subset feedback set problems, we also settle the classification of computational complexity regarding the size of the given terminal set.

(2) Parameterized and exact algorithms for SFVS in tournaments.

In addressing SFVS in tournaments, combining the sub-exponential time solution-separation technique, iterative compression, branching techniques based on blocks, and divide-and-conquer based on balanced cuts, this study improves the time complexity of parameterized algorithms from $\mathcal{O}^*(2.0755^k)$ to $\mathcal{O}^*(1.6181^k)$, and for exact algorithms from $\mathcal{O}(1.5182^n)$ to $\mathcal{O}(1.3820^n)$, respectively. These achievements not only provide the first parameterized algorithm breaking 2^k barrier but also match the running time bound of both parameterized and exact algorithms for the feedback vertex set problem in tournaments. Furthermore, this work significantly improves the parameterized and exact algorithms for the restricted version of SFVS in tournaments.

(3) Parameterized and exact algorithms for SFVS in split graphs.

This study designs a parameterized algorithm for SFVS in split graphs utilizing the Dulmage-Mendelsohn decomposition and branch-and-search technique. By employing the measure-and-conquer method, we prove that the parameterized algorithm runs in time $\mathcal{O}^*(1.8192^k)$, firstly breaking the 2^k barrier. Based on this algorithm, an exact algorithm running in time $\mathcal{O}(1.3488^n)$ is additionally presented. Further expanding the scope of this work, an exact algorithm is established for the prize-collecting maximum independent set problem on hypergraphs, achieved by reducing it to SFVS in split graphs. This algorithm runs in $\mathcal{O}^*(1.8192^n)$, which is the first non-trivial exact algorithm that outperforms the brute-force approaches. Furthermore, based on dynamic programming and branch-and-search techniques, this work improves the parameterized and exact algorithms for the restricted version of SFVS in split graphs.

(4) Parameterized and exact algorithms for SFVS in chordal graphs.

This study innovatively proposes a novel divide-and-conquer approach based on tree decomposition, effectively linking SFVS in split and chordal graphs. This approach implies the faster parameterized and exact algorithms for SFVS in chordal graphs, improving the running time bound from $\mathcal{O}^*(2^k)$ to $\mathcal{O}^*(1.6181^k)$, and from $1.5^n 2^{o(n)}$ to $\mathcal{O}(1.3788^n)$, respectively. Notably, the time complexity of the parameterized algorithm first breaks the 2^k barrier. Furthermore, by employing a similar divide-and-conquer approach, this work improves parameterized and exact algorithms for the restricted version of SFVS in chordal graphs.

Keywords: Parameterized Algorithm, Exact Algorithm, Feedback Set Problem, Subset Feedback Set Problem, Implicit Hitting Set Problem

目 录

第一章 绪 论	1
1.1 研究背景	1
1.1.1 反馈集问题与子集反馈集问题	1
1.1.2 NP-完全问题的算法研究	2
1.2 研究动机与挑战	7
1.3 研究内容与贡献	11
1.4 章节结构安排	15
第二章 理论基础	17
2.1 图论基础	17
2.2 精确与参数算法	20
2.2.1 精确算法	20
2.2.2 参数算法与核心化	22
2.2.3 精确算法与参数算法的联系	25
2.3 精确与参数算法设计技术	27
2.3.1 皇冠分解技术	27
2.3.2 树分解技术	28
2.3.3 分支搜索技术与度量治之	29
2.3.4 迭代压缩技术	30
2.3.5 平衡割技术	32
2.4 本章小结	32
第三章 理论基础	33
3.1 图论基础	33
3.2 精确与参数算法	36
3.2.1 精确算法	36
3.2.2 参数算法与核心化	38
3.2.3 精确算法与参数算法的联系	41
3.3 精确与参数算法设计技术	43
3.3.1 皇冠分解技术	43
3.3.2 树分解技术	44
3.3.3 分支搜索技术与度量治之	45

3.3.4 迭代压缩技术	46
3.3.5 平衡割技术	48
3.4 本章小结	48
第四章 反馈集相关问题梳理与研究进展分析	49
4.1 反馈集与子集反馈集问题	49
4.1.1 反馈集问题	49
4.1.2 反馈集问题的求解算法	50
4.1.3 子集反馈集问题	52
4.1.4 子集反馈集问题的求解算法	55
4.2 特殊图类上的反馈集与子集反馈集问题	57
4.2.1 平面图与度有界的图类	57
4.2.2 竞赛图图类	59
4.2.3 相交图图类	60
4.3 本章小结	62
第五章 反馈集与子集反馈集问题的计算复杂性分类	63
5.1 研究背景与主要贡献	63
5.2 一般图与平面图上反馈集问题	67
5.3 常数关键点的子集反馈集问题	75
5.4 本章小结	80
第六章 竞赛图上子集反馈点集问题的参数与精确算法	81
6.1 研究背景与主要贡献	81
6.2 竞赛图的基本结构与性质	83
6.3 一种多项式时间可解情形	85
6.3.1 基本的约简规则	85
6.3.2 典型序的相容性	86
6.3.3 基于典型序的动态规划算法	92
6.4 竞赛图上非限制版子集反馈点集问题	95
6.4.1 亚指数时间的解集分离技术	95
6.4.2 基于简单分支搜索技术的参数算法	99
6.4.3 基于平衡割技术的参数算法	101
6.5 竞赛图上限制版子集反馈点集问题	108
6.5.1 与顶点覆盖问题的归约关系	108
6.5.2 基于归约的参数与精确算法	111

6.6 本章小结	114
第七章 分裂图上子集反馈点集问题的参数与精确算法	115
7.1 研究背景与主要贡献	115
7.2 分裂图的基本结构与性质	118
7.3 分裂图上非限制版子集反馈点集问题	120
7.3.1 基于 Dulmage-Mendelsohn 分解的约简规则	120
7.3.2 一个非传统度量函数及其性质	124
7.3.3 基于分支搜索技术的参数算法	131
7.3.4 基于均摊思想的精确算法	135
7.4 分裂图上限制版子集反馈点集问题	136
7.4.1 与顶点覆盖问题的归约关系	136
7.4.2 基于归约的参数算法	139
7.4.3 基于动态规划的参数算法	140
7.4.4 基于分支搜索技术的精确算法	142
7.5 超图上带惩罚的独立集问题	148
7.6 本章小结	153
第八章 弦图上子集反馈点集问题的参数与精确算法	155
8.1 研究背景与主要贡献	155
8.2 弦图的基本结构与性质	157
8.3 弦图上非限制版子集反馈点集问题	158
8.3.1 基于点割的替换约简规则	159
8.3.2 基本的约简与分支规则	162
8.3.3 基于团树上分治技术的参数算法	168
8.3.4 基于均摊思想的精确算法	177
8.4 弦图上限制版子集反馈点集问题	179
8.4.1 基本的约简规则	179
8.4.2 基于团树上分治技术的参数与精确算法	184
8.5 本章小结	188
第九章 总结与展望	189
9.1 研究工作总结	189
9.2 后续工作展望	191
致 谢	193
参考文献	194

目 录

附录 A 全文涉及的计算问题定义 218

图目录

图 1-1 Karp 的 21 个基本 NP-完全问题	3
图 1-2 碰撞集问题算法研究的技术路线与本文研究动机	8
图 1-3 本文各章关系图	15
图 2-1 迭代压缩技术框架示意图	31
图 3-1 迭代压缩技术框架示意图	47
图 4-1 十二种子集反馈集问题的归约关系	55
图 5-1 Speckenmeyer 的替换组件	65
图 5-2 多重裂点操作示意图	68
图 5-3 基于线性森林分解的非正规平面嵌入示意图	72
图 5-4 证明 4 度平面图上 DFVS 问题 NP-难的替换组件	73
图 5-5 论证 2 个关键点的 R-DSFVS 问题 NP-难性的归约示意图	78
图 6-1 子图 $\mathbf{T}[U]$ 与 $\mathbf{T} - U$ 中关键点的关系示意图	87
图 6-2 子图 $\mathbf{T}[U]$ 与 $\mathbf{T} - U$ 中非关键点的关系示意图	88
图 7-1 二部图 \mathbf{F} 的 Dulmage-Mendelsohn 分解示意图	121
图 7-2 DM 约简规则在删除顶点后的执行情况示意图	126
图 7-3 DM 约简规则在隐藏关键点后的执行情况示意图	127
图 7-4 DM 约简规则在隐藏非关键点后的执行情况示意图	129
图 7-5 非关键点分支规则示意图	133
图 8-1 实例 \mathcal{I} 替换为实例 $\bar{\mathcal{I}}_i$ ($i \in [6]$) 的示意图	160
图 8-2 弦图中的内部关键点、分治割和单纯块示意图。	169
图 8-3 由分治割和单纯块决定的子实例示意图。	170
图 8-4 U_1 -分支规则和 Q_1 -分支规则均不可执行时的实例示意图	174

表目录

表 1-1	(隐式) 3-碰撞集问题参数算法研究历程与技术路线	10
表 1-2	反馈集与子集反馈集问题的计算复杂性分类结果	12
表 1-3	三个重要图类上子集反馈集问题的参数与精确算法主要结果	13
表 2-1	d -碰撞集的参数与精确算法复杂度	26
表 3-1	d -碰撞集的参数与精确算法复杂度	42
表 4-1	反馈集问题类别	50
表 4-2	反馈集问题参数算法研究历程	51
表 4-3	反馈集问题精确算法研究历程	51
表 4-4	反馈集问题算法研究现状	52
表 4-5	反馈集问题类别	54
表 4-6	子集反馈点集问题参数算法研究历程	56
表 4-7	子集反馈点集问题参数算法研究历程	56
表 4-8	子集反馈集问题算法研究现状	57
表 4-9	平面图与度有界的图类上的反馈集相关问题的计算复杂性	58
表 4-10	竞赛图和二部竞赛图上的反馈集与子集反馈集问题的研究现状	59
表 4-11	相交图类上的反馈集和子集反馈集问题的研究现状	61
表 5-1	度有界图上的反馈集问题的计算复杂性分类结果	66
表 5-2	关键集为常数的子集反馈集问题的计算复杂性分类结果	66
表 6-1	竞赛图上子集反馈点集问题的主要算法结果	83
表 7-1	分裂图上子集反馈点集问题的主要算法结果	118
表 7-2	多公共邻居高度关键点分支规则中的分支向量及其分支因子	145
表 7-3	高度关键点分支规则中的分支向量及其分支因子	147
表 8-1	弦图上子集反馈点集问题的主要算法结果	157

缩略词表

英文缩写	英文全称	中文全称
CSP	Constraint Satisfaction Problem	约束可满足性问题
DASFAS	Directed Arc Subset Feedback Arc Set	有向边子集反馈边集
DASFVS	Directed Arc Subset Feedback Vertex Set	有向边子集反馈点集
DFVS	Directed Feedback Vertex Set	有向反馈点集
DFAS	Directed Feedback Arc Set	有向反馈边集
DSFAS	Directed Subset Feedback Arc Set	有向点子集反馈边集
DSFVS	Directed Subset Feedback Vertex Set	有向点子集反馈点集
ESFES	Edge Subset Feedback Edge Set	无向边子集反馈边集
ESFVS	Edge Subset Feedback Vertex Set	无向边子集反馈点集
ETH	Exponential Time Hypothesis	指数时间假设
FES	Feedback Edge Set	无向反馈边集
FPT	Fixed-Parameter Tractable	固定参数可解
FVS	Feedback Vertex Set	无向反馈点集
MIS	Maximum Independent Set	最大独立集
R-SFVS	Restricted Subset Feedback Vertex Set	限制无向点子集反馈点集
R-	Restricted Directed Arc Subset Feedback	限制有向边子集反馈边集
DASFAS	Arc Set	
R-DSFVS	Restricted Directed Subset Feedback Vertex Set	限制有向点子集反馈点集
R-ESFES	Restricted Edge Subset Feedback Edge Set	限制无向边子集反馈边集
SAT	Boolean Satisfiability Problem	布尔可满足性问题
SCC	Set Cover Conjecture	集合覆盖猜想
SETH	Strongly Exponential Time Hypothesis	强指数时间假设
SFES	Subset Feedback Edge Set	无向点子集反馈边集
SFVS	Subset Feedback Vertex Set	无向点子集反馈点集
TSP	Travelling Salesman Problem	旅行商问题

第一章 绪论

1.1 研究背景

NP-完全问题是理论计算机科学中的一个主要研究对象，NP-完全问题的算法研究已成为理论计算机科学领域长期关注的核心课题之一^[1,2]。反馈集问题不仅是基本的NP-完全问题^[3]，也在现代计算机科学领域占据了举足轻重的地位^[4]。

1.1.1 反馈集问题与子集反馈集问题

反馈集问题（Feedback Set Problem）是计算机科学中研究最为广泛和深入的问题之一。反馈集问题旨在求解给定图中最小的顶点集或边集，使得移除该集合可消除图中的所有环路。早在20世纪中叶，反馈集问题就被引入并用于解决时序逻辑电路（Sequential Logic Circuit）的设计问题。20世纪末，反馈集问题的一般化形式——子集反馈集问题（Subset Feedback Set Problem）被提出^[5-9]。该问题关注图中给定的一些“关键”元素，旨在找到一个最小顶点集或边集，使得移除该集合能够消除所有经过“关键”元素的环路。子集反馈集问题的引入进一步拓展了在实际场景中的适用性，可以根据不同场景定义不同的“关键”元素。

反馈集问题的研究可追溯至1957年。麻省理工学院（Massachusetts Institute of Technology）电子学研究实验室的一份技术报告^[10]指出，时序逻辑电路设计中需要在每一个逻辑信号环路上放置特定的反馈元件，以确保电路的正常运作。因此，一个电路所对应的有向图上最小反馈点集的大小表征了电路所需反馈元件的数量。在软件工程领域，通过建立程序的依赖关系图并求解其反馈点集，可以有效识别导致程序出错或崩溃的关键路径和代码模块^[11]。常见的应用之一是利用最小反馈点集来解决操作系统死锁（Deadlock）问题。

随着21世纪社交网络和人工智能技术的快速发展，反馈集与子集反馈集问题在理论研究和实际应用中引起了更广泛的关注，它们在竞赛排名^[12-15]、大规模集成电路设计^[16]、编码设计^[17-19]等多个领域展现出了更显著的应用价值。

在社交网络中，一般用节点表示个体，边表示个体间的关系。反馈集和子集反馈集问题可用于识别社交网络中有影响力 的用户或者信息传播的重要节点，这有助于判断信息传播路径，对理解和预测信息传播模式具有重要意义^[20,21]。此外，对于循环赛制（Round-Robin）竞赛排名^[14,15,22]和选举（Voting）^[23,24]等组合博弈论（Combinatorial Game Theory）模型而言，反馈集和子集反馈集问题也是一个十分有力的数学工具。例如，Kemeny-Young方法是社会选举理论（Social Choice

Theory) 中的一种选举机制, 通过有效地整合与分析选民的偏好得到最能代表选民总体意愿的候选人排名, 这一方法可以被建模为有向反馈边集问题进行求解。

人工智能领域的许多问题能建模为约束可满足性问题 (Constraint Satisfaction Problem, CSP) 并可通过反馈集问题进行求解^[25]。具体地, 将约束可满足性问题中的每一个变量表示为无向图的一个顶点, 若两个变量之间存在约束关系, 则连接一条边。通过计算此无向图上的最小反馈点集, 可优先对反馈点集上的变量进行赋值, 便可高效地求解余下变量的最优赋值。此外, 在贝叶斯推理 (Bayesian Inference) 中, 回路割问题 (Loop Cutset Problem) 也可以通过反馈集问题来进行计算^[22, 26]。这一技术路线在多个领域内展现了重要的应用价值。特别地, 在生物信息学 (Bioinformatics) 中, 回路割问题能用来估计基因重组 (Genetic Recombination) 概率^[27]; 在编码理论中, 它有助于优化译码过程, 提高信息传输的准确性^[17-19]; 同时在机器学习领域, 该问题能够帮助改进算法的学习效率和预测准确率^[28]。

在理论计算机领域, 许多著名问题的计算复杂性与反馈集问题密切相关。例如, 当最小反馈点集大小有界时, 图同构问题 (Graph Isomorphism Problem) 可高效求解^[29]。因此, 这类问题都可预先计算最小反馈点集, 进而能较快地完成求解, 这推动了反馈集相关问题在复杂性理论和算法设计方面的深入研究, 特别是在一些重要图类上的算法研究。

1.1.2 NP-完全问题的算法研究

在现实生活和工业应用中, 大量优化问题被证明是 NP-完全问题, 如布尔可满足性问题 (Boolean Satisfiability Problem, SAT)、旅行商问题 (Travelling Salesman Problem, TSP)、碰撞集问题 (Hitting Set Problem)、独立集问题 (Independent Set Problem)、反馈集问题^①。图1-1展示了 Karp 的 21 个基本 NP-完全问题^[3]。

NP-完全理论表明, 在 $P \neq NP$ 假设下, NP-完全问题不存在多项式时间的精确求解算法。许多 NP-完全问题都存在平凡的穷举搜索算法, 但这些方法因计算复杂度过高而难以适用于实践实例的规模。因此, 面对实际应用中的复杂优化问题, 启发式算法成为了一种常用的求解方法, 虽然无法保证解的最优性或近似最优性, 但往往能获得比随机算法更好的结果。另一方面, 近似算法和随机算法作为有理论保障的求解算法, 是长期热门的研究方向。这两类算法在一定条件下能提供近似或概率意义上的次优解, 但是无法保证一定得到最优精确解。因此, 如何高效并精确地解决 NP-完全问题一直是算法研究领域的核心挑战。

^① 根据目标函数的不同, 优化问题通常分为最大化和最小化两类问题。在本文涉及的优化问题中, 最大化和最小化问题的其中之一是平凡的。因此, 在不引起歧义的情况下, 本文的命名方式将统一省略前缀“最小”或“最大”。

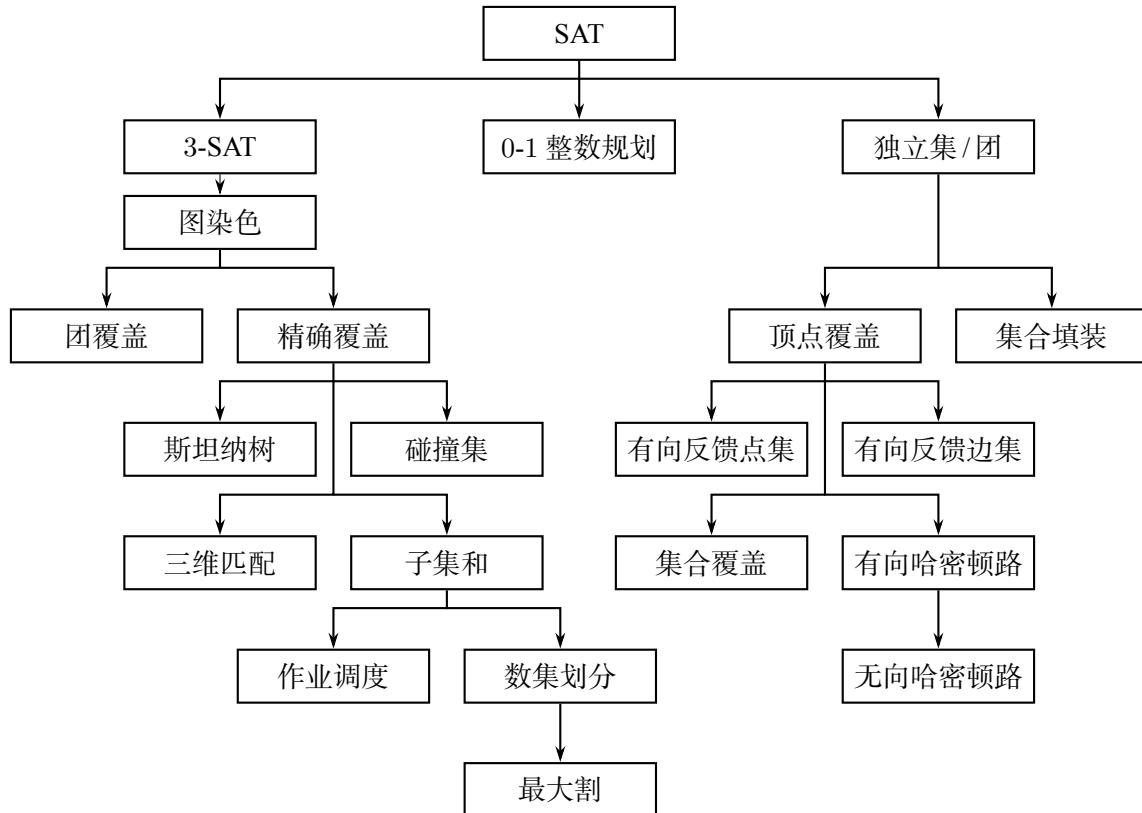


图 1-1 Karp 的 21 个基本 NP-完全问题。箭头表示多项式时间归约方向。

精确算法 精确算法 (Exact Algorithm) 是指总能为优化问题计算得到最优解的 (超多项式时间) 算法, 在实际应用和理论研究中均占有重要地位。

实践方面, 对于需要高度精确或涉及大规模数据的领域, 近似算法或随机算法往往具有一定局限性而无法胜任。例如, 一个 2 倍近似算法可能被视为理论上足够优秀的算法, 甚至可能已经触及了近似率下界。然而, 在一些生物学领域, 这种近似级别往往不足以满足需求。在一些场景下, 如涉及到基因或蛋白质结构的优化问题, 相关研究成果的可靠性和实用性对求解结果的精确性高度敏感, 即便是微小的误差也可能导致结论完全失效。同样, 在大规模集成电路设计等工业应用中, 算法得出的优化结果将直接影响生产成本和效率。因此, 只要算法能够提供精确解, 即使是需要花费数日计算时间, 也是完全可接受的。

理论方面, 学者们对 NP-完全问题计算时间的理解和认识仍十分有限, 甚至还无法证明 SAT 问题一定需要超线性时间进行求解。这突显了理论界对 NP-完全或 NP-难问题更精细化描述的迫切需求。近期, 一些新兴的研究方向成功地将指数时间算法与多项式时间算法联系起来, 使得人们更加专注于研究精确算法的指数运行时间。研究者们发现, 针对 NP-完全问题时间复杂度下界的研究揭示了一个重要的事实: 如果一些时间复杂度下界的猜想成立, 那么 P 类中的许多问题不存在比

当前最快算法更高效的算法^[30]。在这一研究领域，最著名的猜想包括强指数时间假设（Strong Exponential Time Hypothesis, SETH）^[31]和集合覆盖猜想（Set Cover Conjecture, SSC）^[32,33]。基于这些猜想，数百个问题已在近年得到了系统性研究并获得了相应的计算复杂性下界结果。这些研究进展表明，精确算法在 NP-完全问题研究中的重要性不仅仅在于提供高效的解决方案，更在于通过深化各类问题计算复杂性的理解，建立或反驳当前流行的计算复杂性猜想。

NP-完全问题具有表达任意 NP 问题的能力，这意味着它们在计算上具有高度的难求解性。然而，精确算法领域的相关研究揭示了求解 NP-完全问题存在超越平凡的穷举搜索的可能性。长期研究表明，通过探索 NP-完全问题内在的结构特性和利用丰富的算法设计技巧，可以发掘更高效的方法。例如，分支搜索（Branch and Search）和动态规划（Dynamic Programming）技术，作为获得非平凡算法的经典技术手段，显著提高了许多问题的求解效率。具体而言，分支搜索技术递归地分割解空间，试图在计算过程中推断出无需进一步遍历的区域，从而有效地缩短了搜索时间；动态规划技术通过自底向上地存储较小实例的解来避免重复计算，并通过递归关系逐步构建出更大实例的解。这两种技术极大地推动并深化了对 NP-完全问题的精确算法研究，不仅拓展了对相关问题的理解，也为实际应用提供了理论依据。作为一个典型案例，独立集问题是否能在 $\mathcal{O}(1.2^n)$ 时间内求解曾是一个备受关注的公开性难题。经过大量研究人员 30 余年的持续努力和对分支搜索等技术的深度挖掘，该问题运行时间上界目前已被改进到 $\mathcal{O}(1.1996^n)$ ^[34]。此成就不仅在算法设计领域具有里程碑式的意义，也使得处理大规模实例成为可能。这一案例充分表明，即便是 NP-完全问题，通过深入挖掘和巧妙设计，依然有机会找到相当高效的算法。

遗憾的是，当前一些核心的 NP-完全问题的最佳算法依然是平凡的穷举搜索算法，即使尝试各种精巧的算法设计技术，它们的运行时间仍未能得到改进。例如，SAT 问题的最快算法运行时间仍然是 $\mathcal{O}^*(2^n)$ 穷举搜索算法^①（ n 表示变量个数）。若 SETH 成立，那么包括 SAT 问题和碰撞集问题在内的许多基础性 NP-完全问题将不存在运行时间远快于 2^n 的精确算法。另一个例子是旅行商问题，早在 20 世纪 60 年代就存在运行时间为 $\mathcal{O}(2^n n^2)$ 的动态规划算法^[35]（ n 表示顶点数），但过去 50 多年内一直未能打破 2^n 的时间复杂度瓶颈。因此，突破此类问题的 2^n 时间复杂度瓶颈对算法理论具有十分重大的意义。

^① 演近符号 \mathcal{O}^* 省略了多项式部分，例如 $\mathcal{O}^*(f(n)) = f(n)(n + m)^{\mathcal{O}(1)}$ 。

参数算法 参数算法 (Parameterized Algorithm), 亦称为参数化算法, 是精确算法领域里一个相对年轻的研究分支, 着重于探索 NP-难问题是否能够在某些参数较小的情况下被更快速地精确解决。

参数算法的概念源于对特定计算问题在特定输入条件下的早期算法研究。例如, 1985 年, Coppersmith 和 Vishkin^[36]在研究顶点覆盖问题 (Vertex Cover Problem) 时, 通过图的特定参数来设计高效的算法; 1991 年, Bodlaender^[37,38]针对反馈点集问题, 在解集大小受限的条件下进行了研究, 旨在探索给定参数限制下的求解算法效率。此后, 参数复杂性理论在 Downey 和 Fellows 的一系列奠基性工作^[39-42]推动下逐步建立, 该理论提供了一个对困难问题更精细化的分类框架^[43]。

相较于经典的计算复杂性理论, 参数复杂性理论不仅考虑输入实例的大小, 而且需要根据特定的参数来衡量计算复杂性。通过关注给定参数所提供的结构信息, 参数复杂性理论可以更准确地刻画计算问题的复杂性, 从而定义参数较小时的可解性。这里的参数允许以任意方式依赖输入实例。事实上, 各个应用领域都存在一些能够表征复杂性但不同于输入长度的常见参数。例如, 在数据库查询场景中, 问题的输入包含两部分, 即数据库以及查询请求, 其中查询请求次数是参数复杂性分析中一个常用的参数。注意到, 数据库大小通常远大于查询请求次数, 而在查询请求较少时, 查询任务一般能够被高效地执行。在组合优化领域, 集合系统 (Set System) 是一个普适的研究对象。集合系统包括一个集合及其上的子集族, 其中集合大小是参数复杂性分析中的一个常用参数。集合大小一般远小于子集族的大小, 但当集合较小时, 通常存在高效的求解算法。在计算几何 (Computational Geometry) 领域, 随着维度的增加, 问题的求解难度往往呈现指数级增大。使用维度大小作为计算几何问题的参数能够有效刻画维度爆炸 (Curse of Dimensionality)。另外, 优化问题的近似算法是一个更有理论动机的例子。近似算法的输入一般包括输入实例和可接受的误差上界 ε , 以 $1/\varepsilon$ 作为参数能够自然地刻画近似算法的效率。可接受的误差越小, 对应的参数值则越大, 近似算法的计算复杂度也越高。此外, 在图算法领域, 最常用的参数包括解集大小和一些图结构参数, 它们的大小通常也远小于实例规模。

为了描述在参数较小的情况下计算问题的可解性, 参数复杂性理论引入了核心概念: 固定参数可解 (Fixed-Parameter Tractable, FPT)。这一概念对经典的可解性概念 (即多项式时间可解性) 进行了松弛, 允许算法时间复杂度受参数控制的部分是超多项式的。具体地, 若某 NP-难问题关于参数 k 是 FPT 的, 意味着此问题可在 $\mathcal{O}^*(f(k))$ 的时间内被解决, 其中 $f(\cdot)$ 是某个仅与参数 k 相关的可计算函数 (Computable Function)。这表明, 尽管 NP-难问题需要在输入大小的超多项式时间

内求解，但如果此问题的求解时间与输入大小 n 的多项式以及一个仅与参数 k 有关的超多项式 $f(k)$ 成正比，那么可以认为在 k 足够小时，此问题依然能够被高效解决。例如，某算法的时间复杂度为 $\mathcal{O}(2^k n)$ ，当 $k = \mathcal{O}(\log n)$ 时，该问题是多项式时间可解的；而当参数 k 较小而输入大小 n 较大时，这类算法甚至可能展现出比时间复杂度为 $\mathcal{O}(n^3)$ 的算法更好的效率。以苏黎世联邦理工学院（Eidgenössische Technische Hochschule Zürich）的达尔文项目组^[44,45]的研究成果为例。该项目组遇到的一项基因冲突课题可被转化为顶点覆盖问题进行求解。针对该问题的生物学特性，可以保证解集大小上限为 60。基于该特点，项目组采用了顶点覆盖问题当时最快的参数算法^[46,47]，其时间复杂度为 $\mathcal{O}(1.2852^k + kn)$ ，从而在极短的时间内精确地解决了该问题。

自 21 世纪初以来，参数复杂性理论得到了长足发展，核心化（Kernelization）作为参数算法研究领域的一项关键技术而被提出。核心化不仅是设计 FPT 算法的重要手段，也为预处理（Preprocessing）技术提供了坚实的理论基础^[43,48]。通过核心化，可以将问题实例高效地简化为一个大小受参数 k 控制的等价实例，称作该问题的核（Kernel）。特别地，寻求困难问题的多项式核（Polynomial Kernel），即核的大小能被 k 的多项式控制，成为了参数算法领域的一个新兴研究分支。

近年，许多研究人员开始聚焦于参数算法的理论与实验的差异性。相关研究表明，参数算法的理论研究已经能够逐渐与实际算法相结合并得到应用。理论算法的常见做法包括采用多种分支和约简规则来控制其运行时间上界。相较之下，实验算法会更多地结合特定问题的下界，采用分支定界（Branching and Bound）技术并借助各类求解器实现，而不过多地依赖复杂的分支规则。然而，针对顶点覆盖问题和反馈点集问题的相关研究表明^[49–52]，某些为理论目的而设计的分支规则，在实践中展现出了高度的实用性。基于这些分支规则而设计的实验算法，在与其他前沿实验算法的对比中表现出强大的竞争力。顶点覆盖问题和反馈集问题作为基准性问题，在理论和实践层面均引起了广泛的探讨。这进一步证实了理论参数算法正逐步满足实验算法对指导性和可解释性需求。

参数算法研究不仅在理论上为 NP-难问题提供了更精细的复杂性分类，还为实验算法的可解释性研究开辟了新的道路。遗憾的是，当前一些核心 NP-完全问题的最佳参数算法和实验结果相比仍存在显著的差距。例如， d -碰撞集问题的最快算法运行时间略高于 $\mathcal{O}^*((d-1)^k)$ ，但实验结果表明^[53]，求解 d -碰撞集问题的运行时间似乎能够突破 $(d-1)^k$ 的时间复杂度界限。特别是 3-碰撞集问题，目前该问题已被证明能在 $\mathcal{O}^*(2.0755^k)$ 时间内求解^[54]，但难以触及 2^k 界限。此外， 2^k 是众多算法技术自然的计算复杂度下界，进一步突破 2^k 面临着巨大的技术挑战。

1.2 研究动机与挑战

反馈集相关问题是计算机科学中基础而核心的 NP-完全问题。在 $P \neq NP$ 的假设下，此类问题没有多项式时间的求解算法；在指数时间假设下（Exponential Time Hypothesis, ETH），此类问题甚至没有亚指数时间的求解算法^[55]。因此，为了在可接受的时间复杂度内得到精确解，降低精确与参数算法指数时间的复杂度成为了研究相关问题最基本的研究动机。

事实上，反馈集和子集反馈集问题均属于隐式碰撞集问题（Implicit Hitting Set Problem），改进指数运行时间的精确与参数算法同样面临着极大的挑战。碰撞集问题（Hitting Set Problem）是一个高度泛化的组合优化（Combinatorial Optimization）问题。给定大小为 n 的集合 U 以及大小为 m 的集族 $\mathcal{F} \subseteq 2^U$ ，碰撞集问题的目标是找到大小不超过 k 的子集 $S \subseteq U$ （称为碰撞集），使得 S 与 \mathcal{F} 中的每一个集合均有公共元素。换言之，碰撞集问题旨在找到足够小的集合“碰撞”集族 \mathcal{F} 内所有子集。大多数具体问题往往能够使用图结构对集族 \mathcal{F} 进行编码，而不需要显式地给出 \mathcal{F} ，这类问题均属于隐式碰撞集问题。

本节将进一步以碰撞集问题作为宏观背景详述本文的研究动机与挑战，本文整体研究思路参见图1-2。

隐式碰撞集问题的精确算法瓶颈 隐式碰撞集问题包括顶点覆盖问题、反馈集问题、子集反馈集问题、支配集问题（Dominating Set Problem）、奇圈交叉问题（Odd Cycle Transversal Problem）以及各类割问题等诸多图论问题。反馈集相关问题是隐式碰撞集问题的典型代表，图中所有的无向环或有向环与碰撞集问题中集族包含的子集一一对应。然而，从算法角度考虑，反馈集问题所对应集族的结构特征难以被算法有效利用，主要原因在于以下两个方面。首先，集族中的子集具有全局性，因为无向环或有向环的大小均无界。其次，集族中的各子集具有较弱的关联性，因为与路径或树等其他图结构不同，环结构不具有遗传性（即一个大环的子结构不一定是环）。

因此，当前许多隐式碰撞集问题仅存在平凡的穷举搜索算法，为这类问题设计非平凡的精确算法是一个长期的挑战性任务。

研究动机一：为隐式碰撞集问题设计突破 2^n 的精确算法。

碰撞集问题可以使用穷举搜索或动态规划算法在 $\mathcal{O}(m2^n)$ 或 $\mathcal{O}(nm2^m)$ 时间内求解^[56]，它们均为当前最高效的精确算法。尤其是以 n 为参数时，穷举搜索依然是解决碰撞集问题目前理论上最有效手段。在此背景下，理论界提出了以下两个影响深远的计算复杂性猜想。

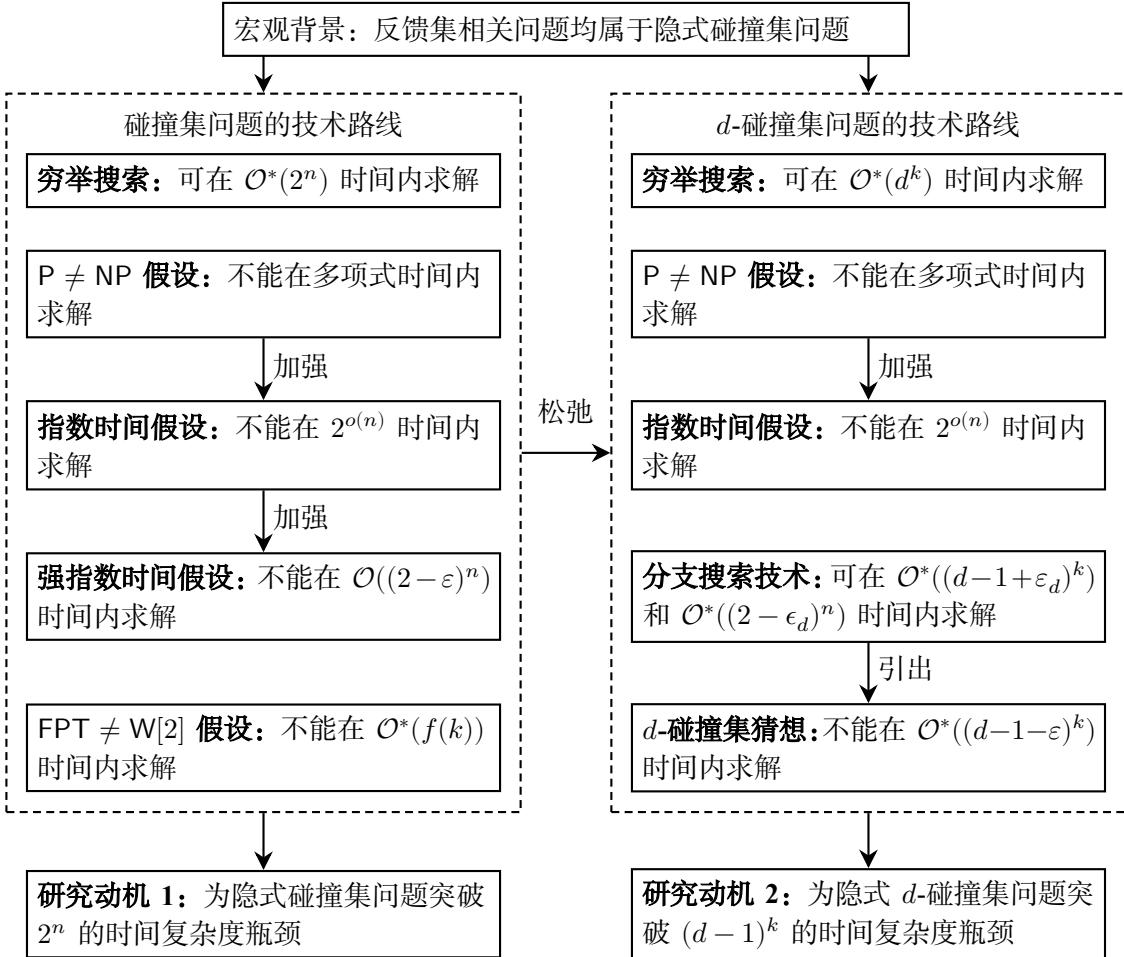


图 1-2 碰撞集问题算法研究的技术路线与本文研究动机

猜想 1 (强指数时间假设, SETH) 对任意正实数 $\varepsilon > 0$, 存在足够大的正整数 $d \in \mathbb{N}^+$, 使得 d -SAT 问题不能在 $\mathcal{O}((2 - \varepsilon)^n)$ 时间内求解, n 表示变量个数。

猜想 2 (集合覆盖猜想, SCC) 对任意正实数 $\varepsilon > 0$, 存在足够大的正整数 $d \in \mathbb{N}^+$, 使得 d -集合覆盖问题不能在 $\mathcal{O}((2 - \varepsilon)^n)$ 时间内求解, n 表示集合大小。

若 SETH 成立, 碰撞集问题无法在 $\mathcal{O}((2 - \varepsilon)^n)$ 时间内求解 ($\varepsilon > 0$)^[32,33]; 类似地, 若 SCC 成立, 碰撞集问题无法在 $\mathcal{O}((2 - \varepsilon)^m)$ 时间内求解 ($\varepsilon > 0$)^[32,33]。这两个猜想揭示了无论是以 n 还是 m 为参数, 碰撞集问题都具有高度难求解的特性, 这推动了对隐式碰撞集问题更深入的研究。

目前, 相当一部分隐式碰撞集问题的精确算法计算复杂度未能突破 2^n , 如子集奇圈交叉问题 (Subset Odd Cycle Transversal Problem), 有向奇圈交叉问题 (Directed Odd Cycle Transversal Problem)、有向子集奇圈交叉问题 (Directed Subset Odd Cycle Transversal Problem)、有向多路割问题 (Directed Multiway Cut Problem)、有向多割问题 (Directed Multicut Problem)、超图上带惩罚独立集问题 (Prize-Collecting Independent Set Problem on Hypergraphs)^[57,58]。其中, 超图上带

惩罚的独立集问题可以视为一种子集反馈集问题的特殊变体，该问题的精确算法是本文的研究内容之一。

针对上述问题，突破 2^n 的计算复杂度瓶面临着巨大的挑战，主要原因在于这些问题都具有算法设计中难以捕捉和利用的全局结构。本文研究的子集反馈点集问题的算法将为超图上带惩罚独立集问题提供首个非平凡精确算法。

隐式 d -碰撞集问题的参数算法瓶颈 d -碰撞集问题是指集族中集合大小均不超过 d 的碰撞集问题。基于 $\text{FPT} \neq \text{W}[2]$ 的参数复杂性假设，碰撞集问题不存在以解集大小 k 为参数的 FPT 算法^[39]，在参数复杂性框架下具有高度难解的特点。因此，研究者们采用与研究 SAT 问题相似的技术路线，考虑 d -碰撞集问题（ d -Hitting Set Problem）的参数算法。长期的研究显示， d -碰撞集问题能够在非常接近 $(d-1)^k$ 的时间内求解。然而，即使是一些隐式 d -碰撞集问题，目前也未能突破这一时间复杂度界限。

因此，对于隐式 d -碰撞集问题，乃至 d -碰撞集问题，是否能够在 $\mathcal{O}^*((d-1)^k)$ 时间内求解成为了一个备受关注的课题。

研究动机二：为隐式 d -碰撞集问题设计突破 $(d-1)^k$ 的参数算法。

假设 d -碰撞集问题最快参数算法的时间复杂度为 $\mathcal{O}^*(\xi_d^k)$ 。由于此问题的穷举搜索算法运行时间为 $\mathcal{O}^*(d^k)$ ^[59]，故有 $\xi_d \leq d$ 。随着研究的深入， d -碰撞集问题是否存在比穷举搜索更高效的参数算法成为一个备受关注的问题。经过一系列理论研究^[53, 60–63]，目前仅将时间复杂度上界推进到

$$\xi_d \leq d - 1 + \mathcal{O}(d^{-2}).$$

然而，大量实验算法^[53]的表现似乎显示了 $\xi_d \leq d - 2 + o(1)$ ($d \geq 3$) 成立的可能性。因此，(隐式) d -碰撞集问题是否能在 $\mathcal{O}^*((d-1)^k)$ 时间内求解成为一项重要的挑战和公开性难题^[53, 63, 64]。

猜想 3 (d -碰撞集猜想) 对任意正实数 $\varepsilon > 0$ ， d -碰撞集问题不能在 $\mathcal{O}^*((d-1-\varepsilon)^k)$ 时间内求解。

在 $d = 2$ 时， d -碰撞集猜想是 ETH 的特例^[55]；换言之，若 ETH 成立，那么顶点覆盖问题（即 2-碰撞集问题）不存在运行时间为 $\mathcal{O}^*(2^{o(k)})$ 的参数算法。但当 $d \geq 3$ 时， d -碰撞集猜想是否成立尚不明确。事实上，针对 3-碰撞集问题，参数算法领域内已经展开了一系列深入的研究并取到了显著的进展^[54, 60, 65, 70]。其中，Wahlström^[54]的贡献尤为突出，在其 2007 年发表的博士论文中提出了一个复杂的分支搜索算法，并通过相当精细的度量治之（Measure and Conquer）分析手段，证

表 1-1 (隐式) 3-碰撞集问题参数算法研究历程与技术路线

问题名称	作者 主要技术路线	时间复杂度	发表年份
3-碰撞集问题	Cai ^[59] 基于导出子图的通用搜索框架	$\mathcal{O}^*(3^k)$	1992
3-碰撞集问题	Downey ^[65] 归约至顶点覆盖问题	$\mathcal{O}^*(2.5616^k)$	1997
3-碰撞集问题	Niedermeier 和 Rossmanith ^[60] 标准分支搜索技术	$\mathcal{O}^*(2.270^k)$	2003
聚类图顶点删除问题	Gramm 等 ^[66] 计算机辅助的搜索树自动生成技术	$\mathcal{O}^*(2.260^k)$	2004
竞赛图上反馈点集问题	Dom 等 ^[62,67] 迭代压缩技术	$\mathcal{O}^*(2^k)$	2006
3-碰撞集问题	Wahlström ^[54] 基于度量治之的分支搜索技术	$\mathcal{O}^*(2.0755^k)$	2007
聚类图顶点删除问题	Hüffner 等 ^[68,69] 迭代压缩技术	$\mathcal{O}^*(2^k)$	2008
3-碰撞集问题	Fernau ^[70] 标准分支搜索技术	$\mathcal{O}^*(2.1788^k)$	2010
聚类图顶点删除问题	Boral 等 ^[71,72] 基于局部极小顶点覆盖的分支搜索技术	$\mathcal{O}^*(1.9102^k)$	2014
3 长路顶点覆盖问题	Tu ^[73] 迭代压缩技术	$\mathcal{O}^*(2^k)$	2015
3 长路顶点覆盖问题	Wu ^[74] 基于度量治之的分支搜索技术	$\mathcal{O}^*(1.8818^k)$	2015
竞赛图上反馈点集问题	Kumar 和 Lokshtanov ^[75] 解集分离技术和平衡割技术	$\mathcal{O}^*(1.6181^k)$	2016
3 长路顶点覆盖问题	Katrenic ^[76] 标准分支搜索技术	$\mathcal{O}^*(1.8172^k)$	2016
3 长路顶点覆盖问题	Chang 等 ^[77] 基于路分解的标准均摊技术, 需指数空间复杂度	$\mathcal{O}^*(1.7485^k)$	2016
3 长路顶点覆盖问题	Xiao 和 Kou ^[78,79] 基于低度图结构的分支搜索技术	$\mathcal{O}^*(1.7485^k)$	2017
弦图上子集反馈点集问题	Philip 等 ^[80] 基于单纯点的分支搜索技术	$\mathcal{O}^*(2^k)$	2019
3 长路顶点覆盖问题	Tsur ^[81] 基于弱支配点的分支搜索技术	$\mathcal{O}^*(1.7130^k)$	2019
聚类图顶点删除问题	Tsur ^[82] 基于局部极小顶点覆盖与孪生点对的分支搜索技术	$\mathcal{O}^*(1.8106^k)$	2021
3 长路顶点覆盖问题	Cervený 和 Suchý ^[83] 计算机辅助的分支规则生成技术	$\mathcal{O}^*(1.7080^k)$	2023
聚类图顶点删除问题	Tian 等 ^[84] 基于低度图结构的分支搜索技术	$\mathcal{O}^*(1.7549^k)$	2023

明了该算法的时间复杂度为 $\mathcal{O}^*(2.0755^k)$ ，这几乎触及了 2^k 的时间复杂度瓶颈。然而，此上界结果至今未能进一步得到改进。

近十年，许多著名的隐式 3-碰撞集问题的参数算法成果相继问世，不少隐式 3-碰撞集问题成功突破了 2^k 的计算复杂度瓶颈，表1-1展示了相关问题的研究历程与技术路线。著名的隐式 3-碰撞集问题包括三角形顶点删除问题（Triangle-free Vertex Deletion Problem）、聚类图顶点删除问题（Cluster Vertex Deletion Problem）、竞赛图上有向反馈点集问题（Directed Feedback Vertex Set Problem in Tournaments）、竞赛图上有向子集反馈点集问题（Directed Subset Feedback Vertex Set Problem in Tournaments）、分裂图上子集反馈点集问题（Subset Feedback Vertex Set Problem in Split Graphs）、弦图上子集反馈点集问题（Subset Feedback Vertex Set Problem in Chordal Graphs）、3-长路顶点覆盖问题（3-Path Vertex Cover Problem）。其中有多类问题与反馈集和子集反馈集问题直接相关，这进一步凸显了反馈集和子集反馈集问题在相关领域研究中的核心地位。然而，三角形顶点删除问题和竞赛图上有向子集反馈点集问题长时间无法得到快于 $\mathcal{O}^*(2.0755^k)$ 的参数算法；分裂图和弦图上子集反馈点集问题也未突破 2^k 的时间复杂度瓶颈。

值得强调的是，在参数算法领域， 2^k 是一个难以逾越的壁垒。这是因为 2^k 是众多算法技术的自然时间复杂度下界，如简单搜索算法、动态规划算法、容斥原理（Inclusion-Exclusion）算法、子集卷积（Subset Convolution）算法以及迭代压缩算法等等。

综上所述，为隐式 3-碰撞集问题寻找能够突破 2^k 的参数算法，既是一项技术挑战，也是一种理论创新。这要求深刻理解问题的本质结构，创新算法设计理念，并细致地分析算法性能。因此，各类隐式 3-碰撞集问题能否打破 2^k 的时间复杂度瓶颈成为了本文聚焦于竞赛图、分裂图和弦图上子集反馈点集问题算法研究的主要动机。针对这些问题，本文提出的参数算法成功地突破了 2^k 的时间复杂度瓶颈，为推动解决 d -碰撞集猜想提供了更多证据。

1.3 研究内容与贡献

本文主要研究了反馈集与子集反馈集问题的计算复杂性分类，以及三类特殊图上子集反馈点集问题的参数与精确算法。本文的研究成果已形成四篇学术论文，另有两篇论文正处于同行评审阶段，且已经以预印版形式发布。

首先，本研究全面地调研了反馈集相关问题的经典计算复杂性，对各图类上反馈集与子集反馈集问题的计算复杂性进行了综述和分析。在此工作基础上，本文解决了遗留近五十年的反馈集计算复杂性分类问题。相关研究工作已在国内外顶

级期刊《计算机研究与发展》上发表一篇论文，另有一篇论文处于同行评审阶段。

其次，本文研究了竞赛图（Tournament）、分裂图（Split Graph）和弦图（Chordal Graph）上的子集反馈点集问题。通过结合和扩展多种前沿算法设计技术，全面改进上述三个问题的现有参数和精确算法的计算复杂度上界，其中的参数算法均成功突破 2^k 。相关研究工作已在国际高水平会议《Theory and Applications of Models of Computation》和《International Symposium on Mathematical Foundations of Computer Science》上各发表一篇论文，在理论计算机国际顶级期刊《Theoretical Computer Science》发表论文两篇，另有一篇论文处于同行评审阶段。

计算复杂性分类 本文按照最大度和关键集大小两个基本参数，对反馈集相关问题的计算复杂性开展了分类工作（详见表1-2）。

表 1-2 反馈集与子集反馈集问题的计算复杂性分类结果

图类	问题名	多项式可解	NP-完全
有向图	反馈点集问题	最大度 $\Delta \leq 2$	最大度 $\Delta \geq 3$
有向图	反馈边集问题	最大度 $\Delta \leq 2$	最大度 $\Delta \geq 3$
平面有向图	反馈点集问题	最大度 $\Delta \leq 3$	最大度 $\Delta \geq 4$
有向图	限制版有向子集反馈点集问题	关键点数 $\tau \leq 1$	关键点数 $\tau \geq 2$
有向图	限制版有向边子集反馈边集问题	关键点数 $\tau \leq 1$	关键点数 $\tau \geq 2$
有向图	有向反馈边集问题	关键点数 $\tau \leq 1$	关键点数 $\tau \geq 2$

第一个工作研究了有向图上反馈集问题关于最大度的计算复杂性。作为图结构的一个基本参数，按照最大度对图问题的计算复杂性进行分类，是一个重要且常见的研究课题。然而，对重要图类上反馈集相关问题计算复杂性整理分析后，发现有向图和平面有向图上的反馈集问题关于最大度的计算复杂性分类工作进展缓慢，长期停留在平凡的上下界结论上。对于最大度为 3、4 和 5 的情况，有向图及平面有向图上反馈集问题的计算复杂性长期未被解决，成为了公开性的难题。本文明确了当且仅当最大度至少为 3 时，一般有向图上的反馈点集和反馈边集问题是 NP-难的；进一步地，当且仅当最大度至少为 4 时，平面有向图上的反馈点集问题是 NP-难的。这一成果完备了一般图和平面图上反馈集问题关于最大度的计算复杂性分类。

第二个工作探讨了关键集大小为常数时，子集反馈点集问题的计算复杂性。在图连通性问题中，以关键点数量进行计算复杂性分类也是一个重要且热门的课题^[85-89]。根据输入图是否为有向图、关键集和解集是否为点集，以及是否为限制版本，子集反馈集问题共可细分为十二种不同版本。本文全面地整理归纳了各版本问题间的归约关系，并基于此证明了三种子集反馈集问题至少存在 2 个关键元

素时是 NP-难的。这一结果完备了子集反馈集问题关于关键集大小的计算复杂性分类。

三类隐式 3-碰撞集问题的算法研究 竞赛图和弦图分别是图论和算法领域内十分重要的有向图和无向图，分裂图为弦图的子图。本文全面改进了竞赛图、分裂图和弦图上子集反馈点集问题的参数与精确算法（详见表1-3），它们均为隐式 3-碰撞集问题。长期以来，针对这些问题寻找比 3-碰撞集问题更快速的参数和精确算法面临诸多挑战。本文为这三个问题及其限制版^①设计了精确算法和以解集大小 k 为参数的参数算法。特别地，本文提出的参数算法实现了突破性进展，运行时间首次快于 $\mathcal{O}^*(2^k)$ 。作为推论，本文也为超图上带惩罚的独立集问题设计了比穷举搜索算法更快的精确算法，运行时间首次突破 2^n 。

表 1-3 三个重要图类上子集反馈集问题的参数与精确算法主要结果

图类	问题名	精确算法	参数算法
竞赛图	(非限制版) 子集反馈点集问题	$\mathcal{O}(1.3820^n)$	$\mathcal{O}^*(1.6181^k)$
	限制版子集反馈点集问题	$\mathcal{O}(1.1996^n)$	$\mathcal{O}^*(1.2529^k)$
分裂图	(非限制版) 子集反馈点集问题	$\mathcal{O}(1.3488^n)$	$\mathcal{O}^*(1.8192^k)$
	限制版子集反馈点集问题	$\mathcal{O}(1.1520^n)$	$\mathcal{O}^*(1.2529^k)$
弦图	(非限制版) 子集反馈点集问题	$\mathcal{O}(1.3788^n)$	$\mathcal{O}^*(1.8192^k)$
	限制版子集反馈点集问题	$\mathcal{O}(1.1520^n)$	$\mathcal{O}^*(1.2529^k)$

(1) 竞赛图上的子集反馈点集问题

本文首先研究了竞赛图上子集反馈点集问题。作为竞赛图上反馈点集问题的一般形式，现有的相关技术均无法直接应用于该问题上。

本文引入解集分离技术，通过结合迭代压缩技术进行了改良和简化，克服了原技术对可近似性的依赖性，优化了时间复杂度。此外，本文创新性地提出了基于分割块的分支技术，以保证图结构具有足够弱的连通性。在此基础上，结合基于平衡割的分治思想，成功地将竞赛图上子集反馈点集问题的参数算法时间复杂度从 $\mathcal{O}^*(2.0755^k)$ 降至 $\mathcal{O}^*(1.6181^k)$ ，首次突破了 2^k 的时间复杂度瓶颈。基于这一参数算法结果，本文也将该问题的精确算法时间复杂度从 $\mathcal{O}(1.5182^n)$ 降至 $\mathcal{O}(1.3820^n)$ 。值得强调的是，本文给出的算法复杂度已与当前最快的竞赛图上反馈点集问题时间复杂度持平。

针对限制版问题，本文证明了该问题与顶点覆盖问题的参数和精确算法有相同的指数时间复杂度。因此，竞赛图上限制版子集反馈集点集问题能在 $\mathcal{O}(1.1996^n)$ 和 $\mathcal{O}^*(1.2529^k)$ 时间内求解。

^① 限制版本特指解集不允许包含关键元素的变体问题。区分限制版与非限制版是针对图连通性问题的常见研究途径。

(2) 分裂图上的子集反馈点集问题

本文其次研究了分裂图上子集反馈点集问题。该工作将为后续弦图上子集反馈点集问题的算法研究奠定基础。

本文借助 Dulmage-Mendelsohn 分解设计了基于分支搜索技术的参数算法。同时, 本文创新性地引入了一种非传统度量函数, 采用度量治之分析手段, 证明了该算法的运行时间为 $\mathcal{O}^*(1.8192^k)$ 。在此基础上, 也证明了分裂图上子集反馈点集问题能在 $\mathcal{O}(1.3788^n)$ 时间内求解。另外, 本研究指出, 分裂图上子集反馈点集问题可以视为超图上带惩罚的独立集问题的推广形式。本文进一步建立了这两个问题的联系: 对任意实数 $\alpha > 1$, 若前者存在运行时间为 $\mathcal{O}^*(\alpha^k)$ 的参数算法, 那么后者则存在 $\mathcal{O}^*(\alpha^n)$ 的精确算法。因此, 超图上带惩罚的独立集问题能在 $\mathcal{O}^*(1.8192^n)$ 时间内求解, 首次得到了比穷举搜索更高效精确算法。

针对限制版问题, 本文证明了该问题与顶点覆盖问题的参数算法有相同的指教时间复杂度, 从而得到了运行时间为 $\mathcal{O}^*(1.2529^k)$ 的参数算法。进一步, 本文使用基于度量治之的分支搜索技术, 为限制版问题设计了比顶点覆盖问题更快的精确算法, 其时间复杂度为 $\mathcal{O}(1.1520^n)$ 。作为推论, 本文证明了独立集问题以团覆盖数 (Edge-Clique Cover Number) 为参数是 FPT 的。

(3) 弦图上的子集反馈点集问题

本文最后研究了弦图上子集反馈点集问题。直至 2019 年, 弦图上子集反馈点集问题才被证明存在比 3-碰撞集问题更高效的参数算法, 其时间复杂度为 $\mathcal{O}^*(2^k)$ 。该算法使用了复杂的分支搜索技术而难以取得进一步突破。

本文深入研究了分裂图和弦图的结构特征, 采用树分解上的分治技术, 建立了两类图上子集反馈点集问题的联系。本文摒弃了树分解上传统的动态规划技术路线, 而是通过寻找具有关键性质的极小点割, 多次调用分裂图上子集反馈点集问题的求解算法, 将部分子结构完全删除进而有效地缩小实例规模。因此, 在可接受的时间复杂度内, 能够逐步简化实例直至图结构退化为分裂图。

基于上述技术路线, 本文证明了若分裂图上子集反馈点集问题能在 $\mathcal{O}^*(\alpha^k)$ 时间内求解 ($\alpha > 1$), 那么弦图上子集反馈点集问题则能在 $\mathcal{O}^*(\alpha^k + 1.6181^k)$ 时间内求解, 从而将弦图上子集反馈点集问题的参数算法时间复杂度从 $\mathcal{O}^*(2^k)$ 降至 $\mathcal{O}^*(1.8192^k)$, 首次打破了 2^k 的时间复杂度界限。由于弦图上反馈点集问题是多项式时间可解的, 本文进一步采用均摊思想, 将弦图上子集反馈点集问题的精确算法时间复杂度从 $1.5^n 2^{o(n)}$ 降至 $\mathcal{O}(1.3788^n)$ 。基于类似的技术路线, 本文也证明了弦图上限制版子集反馈点集问题能在 $\mathcal{O}(1.1520^n)$ 以及 $\mathcal{O}^*(1.2529^k)$ 时间内求解, 与分裂图上子集反馈点集问题的算法时间复杂度持平。

1.4 章节结构安排

全文共分为八个章节，其中第三章介绍了本研究的理论基础；第四章梳理了问题定义并分析了研究现状；第五章聚焦于计算复杂性分类工作；第六章至第八章分别对三种不同图类上的子集反馈点集问题的精确与参数算法进行了研究；最后，第九章对整篇论文进行归纳总结。各章关系见图1-3。

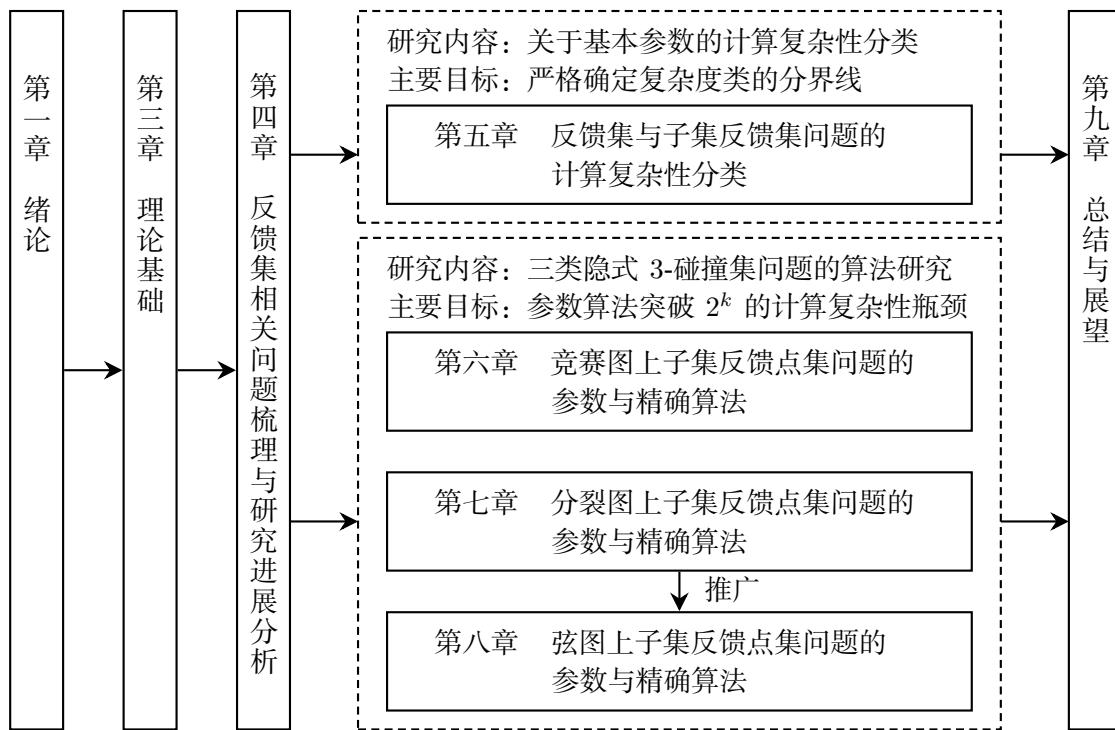


图 1-3 本文各章关系图

第三章为理论基础介绍。3.1节介绍了图论的基本概念和符号系统。3.2节介绍了计算理论以及本文涉及的参数复杂性理论的基础知识。3.3节介绍了本研究涉及的精确和参数算法的设计技术。

第四章为问题定义梳理与研究现状分析。4.1节给出了反馈集和子集反馈集问题的各版本定义，分析并补全了完整的归约关系。4.2节对全文涉及的特殊图上的反馈集和子集反馈集问题的计算复杂性研究现状进行了综述和分析。

第五章为计算复杂性分类工作。5.1节概述了本研究中关于计算复杂性分类的研究背景、研究现状和主要贡献。5.2节研究了反馈集问题关于最大度的计算复杂性分类。5.3节研究了子集反馈集问题关于关键集大小的计算复杂性分类。

第六章为竞赛图上子集反馈点集问题的算法研究工作。6.1节概述了竞赛图上子集反馈点集问题的研究背景、研究现状和主要贡献。6.2节整理归纳了竞赛图上顶点的序结构，这些结构性质将应用于相关算法研究。6.3节提出了竞赛图上子集反馈点集问题的一个多项式时间可解情形，给出了基于动态规划的多项式时间算

法, 该算法将应用于后续参数与精确算法研究。6.4节研究了竞赛图上非限制版子集反馈点集问题, 给出了基于解集分离技术、分割块分支技术和平衡割技术的参数算法, 突破了 2^k 的时间复杂度瓶颈; 最后提出了基于单调局部搜索技术的精确算法。6.5节研究了竞赛图上限制版子集反馈点集问题的参数与精确算法, 通过建立与顶点覆盖问题的归约关系, 给出了该问题的精确算法、参数算法和线性核。

第七章为分裂图上子集反馈点集问题的算法研究工作。7.1节概述了分裂图上子集反馈点集问题的研究背景、研究现状和主要贡献。7.2节整理归纳了分裂图上独立集和团的结构特点, 这些结构性质将应用于相关算法研究。7.3节研究了分裂图上非限制版子集反馈点集问题的参数与精确算法, 提出了基于 Dulmage-Mendelsohn 分解的约简技术, 利用该技术给出了突破 2^k 的参数算法; 然后利用分裂图结构, 给出了基于均摊思想的精确算法。7.4节研究了裂图上限制版子集反馈点集问题的参数与精确算法, 通过建立与顶点覆盖问题的归约关系, 给出了该问题的参数算法和平方核; 然后基于分支搜索技术, 提出了比求解顶点覆盖问题的算法更快的精确算法。7.5节研究了超图上带惩罚的独立集问题, 建立此问题与分裂图上子集反馈点集问题的关系, 首次提出了此问题非平凡的精确算法。

第八章为弦图上子集反馈点集问题的算法研究工作。8.1节概述了弦图上子集反馈点集问题的研究背景、研究现状和主要贡献。8.2节整理归纳了弦图的树状结构特征, 这些结构性质将应用于相关算法研究。8.3节研究了弦图上非限制子集反馈点集问题的参数与精确算法, 基于树结构的分治技术, 通过多次调用分裂图上子集反馈点集问题的参数算法, 为弦图上子集反馈点集问题设计了突破 2^k 的参数算法; 然后基于均摊思想, 改进了该问题的精确算法。8.4节研究了弦图上限制版子集反馈点集问题的参数与精确算法, 基于树结构上的类似地分治技术, 通过多次调用分裂图上限制版子集反馈点集问题的算法, 同时改进了此问题上的参数和精确算法; 最后为此问题给出了一个平方核。

第九章为全文研究工作的总结。9.1节对本文的主要研究内容与贡献进行了总结。9.2节进一步对未来的研究重心和难点进行了归纳。

第二章 理论基础

本章首先在3.1节对涉及全文的图论基础和主要术语进行了介绍，然后在3.2和3.3节分别概述了精确算法和参数算法的基本概念与关键技术。本文假定读者对图论和复杂度理论的基础概念已基本了解。全文所涉及的图论基本概念可参见专著 [90,91]，所涉及的精确与参数算法基本概念可参见专著 [1,2,92]。

2.1 图论基础

本文研究的反馈集相关问题属于经典图论问题，本节首先介绍全文统一使用的图论基本概念与符号系统。

在图论中，一个图 (Graph) 通常使用符号 \mathbf{G} 来表示，由其顶点集和边集唯一决定。图 \mathbf{G} 的顶点集表示为 $V(\mathbf{G})$ ，边集表示为 $E(\mathbf{G})$ 。在上下文不产生混淆的情况下，本文简记 $V = V(\mathbf{G})$ 和 $E = E(\mathbf{G})$ ，同时使用正整数 n 表示点集 $V(\mathbf{G})$ 的大小，用 m 表示边集 $E(\mathbf{G})$ 的大小。一个图中的顶点 (Vertex) 一般使用小写字母 u, v, w 等来表示。若 \mathbf{G} 是无向图 (Undirected Graph)，图中的 (无向) 边 (Edge) 表示为一些无序点对；若 \mathbf{G} 是有向图 (Directed Graph 或 Digraph)，图中的有向边 (Arc) 则表示为一些有序点对。图中的边表示为小写字母 e 或二元对 $(v, w) \in V(\mathbf{G})^2$ ，简记为 vw 。本文通常使用符号 \mathbf{D} 表示有向图。

若一条边的两个端点均为同一顶点 v ，则称此边为顶点 v 上的自环 (Loop)。无向图或有向图的边集是一个多重集合 (Multiset)，即集合可以包含多个相同的元素。若图中的边集包含两条或多条与同一对顶点相连接的边，则称这些边为重边或平行边 (Multiple Edges)。没有自环和平行边的图称为简单图 (Simple Graph)。

本文使用符号 $[n]$ 简记前 n 个正整数构成的集合 $\{1, 2, \dots, n\}$ 。图 \mathbf{G} 中的一条路或路径 (Path) 是至少包含一个顶点的子图 \mathbf{P} ，表示为

$$\mathbf{P} = (v_1, v_2, \dots, v_\ell),$$

其中连续的两个顶点相邻，即 $E(\mathbf{P}) := \{v_i v_{i+1} : i \in [\ell - 1]\} \subseteq E(\mathbf{G})$ 。路 \mathbf{P} 的长度是 \mathbf{P} 中顶点的个数 ℓ ，记作 $|\mathbf{P}|$ 。图 \mathbf{G} 中的一个环或圈 (Cycle) 是一个至少包含三个顶点的子图 \mathbf{C} ，表示为

$$\mathbf{C} = (v_1, v_2, \dots, v_\ell),$$

其中连续的两个顶点以及首尾顶点相邻，即 $E(\mathbf{C}) := \{v_i v_{i+1}, v_\ell v_1 : i \in [\ell - 1]\} \subseteq E(\mathbf{G})$ 。环 \mathbf{C} 的长度是 \mathbf{C} 中顶点的数量 ℓ ，记作 $|\mathbf{C}|$ 。图 \mathbf{G} 中连接了路 \mathbf{P} (或环 \mathbf{C})

中不相邻顶点的边称为路 **P** (或环 **C**) 的弦 (**Chord**)。长度为 ℓ 的路径和环分别记作 \mathbf{P}_ℓ 和 \mathbf{C}_ℓ 。特别地, 长度为 3 的环 \mathbf{C}_3 亦被称为三角形 (**Triangle**)。本文中的路和环要求其中的顶点互不同, 在一些文献中也被分别称为简单路和简单环。

一个图中的点子集或边子集一般使用 A, B, X, Y 等大写字母表示。给定图 **G**, 点子集 $X \subseteq V(\mathbf{G})$ 的导出子图 (**Induced Subgraph**), 记作 $\mathbf{G}[X]$, 是 **G** 的一个子图, 其点集为 X , 边集为两个端点都在 X 中的边构成的集合。从图 **G** 中删掉点子集 $X \subseteq V(\mathbf{G})$ 以及与 X 中顶点邻接的边得到的子图用 $\mathbf{G} - X := \mathbf{G}[V(\mathbf{G}) \setminus X]$ 表示。当某一点集为单元集时, 通常将省略集合括号, 如 $\mathbf{G} - \{v\}$ 可简写为 $\mathbf{G} - v$ 。

在图 **G** 中, 点子集 $X \subseteq V(\mathbf{G})$ 的 (开) 邻域 (**Neighborhood**) $N_{\mathbf{G}}(X)$ 是所有与 X 中的点相邻且不属于 X 中的点集, 即

$$N_{\mathbf{G}}(X) := \{w \in V(\mathbf{G}) \setminus X : v \in X, wv \in E(\mathbf{G}) \text{ 或 } vw \in E(\mathbf{G})\}$$

集合 $N_{\mathbf{G}}(v)$ 中的元素称为 v 的 (开) 邻居 (**Neighbor**)。点子集 X 的闭邻域 (**Closed Neighborhood**) 定义为

$$N_{\mathbf{G}}[X] := N_{\mathbf{G}}(X) \cup X.$$

集合 $N_{\mathbf{G}}[v]$ 中的元素称为 v 的闭邻居 (**Closed Neighbor**)。对于有向图 **D**, 点子集 $X \subseteq V(\mathbf{D})$ 的入邻域 (**In-Neighborhood**) 和出邻域 (**Out-Neighborhood**) 分别是不属于 X 的点中, 连入 X 的顶点构成的点集和 X 连出的顶点构成的点集, 分别记作

$$\begin{cases} N_{\mathbf{D}}^-(X) := \{w \in V(\mathbf{D}) \setminus X : v \in X, wv \in E(\mathbf{D})\}; \\ N_{\mathbf{D}}^+(X) := \{w \in V(\mathbf{D}) \setminus X : v \in X, vw \in E(\mathbf{D})\}. \end{cases}$$

入邻域和出邻域内的元素分别称为 v 的入邻居 (**In-Neighbor**) 与出邻居 (**Out-Neighbor**)。类似地, X 的闭入邻域 (**Closed In-Neighborhood**) 和闭出邻域 (**Closed Out-Neighborhood**) 分别定义为

$$\begin{cases} N_{\mathbf{D}}^-[X] := N_{\mathbf{D}}^-(X) \cup X; \\ N_{\mathbf{D}}^+[X] := N_{\mathbf{D}}^+(X) \cup X. \end{cases}$$

在图 **G** 中, 顶点 v 的度 $\deg_{\mathbf{G}}(v)$ 定义为以 v 为端点的边的数量 (自环计算两次)。那么当 **G** 为简单图时, 有 $\deg_{\mathbf{G}}(v) = |N_{\mathbf{G}}(v)|$ 。类似地, 在简单有向图 **D** 中, 顶点 v 的入度 (**In-Degree**) $\deg_{\mathbf{D}}^-(v)$ 和 v 的出度 (**Out-Degree**) $\deg_{\mathbf{D}}^+(v)$, 分别定义为其入邻域和出邻域的大小, 即 $\deg_{\mathbf{D}}^-(v) := |N_{\mathbf{D}}^-(v)|$, $\deg_{\mathbf{D}}^+(v) := |N_{\mathbf{D}}^+(v)|$ 。在上下文不产生混淆的情况下, 本文将省略邻域符号和度符号的下标。

图 **G** 的最大度 (**Maximum Degree**) 定义为图中所有顶点度的最大值, 即

$$\Delta(\mathbf{G}) := \max_{v \in V(\mathbf{G})} \deg_{\mathbf{G}}(v).$$

有向图 **D** 的最大入度 (**Maximum In-Degree**) 和最大出度 (**Maximum Out-Degree**) 分别定义为

$$\begin{cases} \Delta^-(\mathbf{D}) := \max_{v \in V(\mathbf{D})} \deg_{\mathbf{D}}^-(v); \\ \Delta^+(\mathbf{D}) := \max_{v \in V(\mathbf{D})} \deg_{\mathbf{D}}^+(v), \end{cases}$$

同时, 在部分文献中会使用到符号

$$\Delta^*(\mathbf{D}) := \max_{v \in V(\mathbf{D})} \min \{ \deg_{\mathbf{D}}^-(v), \deg_{\mathbf{D}}^+(v) \},$$

本文将其定义为有向图的定向最大度。

在无向图 **G** 中, 如果两个顶点 v 和 w 之间存在一条路连接它们, 则称它们是连通的 (**Connected**), 否则称它们是不连通的 (**Disconnected**)。如果无向图 **G** 的任意两个点均连通, 则称 **G** 为连通图 (**Connected Graphs**)。如果 $C \subseteq V(\mathbf{G})$ 是一个极大的点子集, 使得其导出子图是连通的, 则称其是 **G** 的连通分量 (**Connected Component**)。在本文中, 满足某种性质的极大/极小集合均是指其在包含意义下极大/极小 (Inclusion-Wise Maximal/Minimal), 换言之, 在所有满足该性质的集合中, 没有此集合的真超集/真子集 (Proper Superset/Proper Subset)。一个有向图 **D**, 如果去掉有向边的方向所得到的无向图是连通的, 则称有向图 **D** 是连通的或弱连通的 (**Connected, Weakly Connected**)。对于有向图 **D** 中的两个顶点 v 和 w , 如果存在从 v 到 w 的 (有向) 路, 则称 w 关于 v 可达 (**Reachable**) 或 v 可到达 w 。若有向图 **D** 中的顶点两两相互可达, 则称 **D** 是强连通的 (**Strongly Connected**)。如果 $C \subseteq V(\mathbf{D})$ 是一个极大的点子集, 使得其导出子图是强连通的, 则称 C 为 **D** 的强连通分量 (**Strongly Connected Component**)。

在图 **G** 中, 如果删掉点子集 $X \subseteq V(\mathbf{G})$ 后, 图的连通分量数量增加, 则称 X 为图 **G** 的点割 (**Separator**)。如果删掉边子集 $Y \subseteq E(\mathbf{G})$ 后, 图的连通分量数量增加, 则称 Y 为图 **G** 的边割 (**Cut**)。特别地, 如果一条边自身是一条边割, 则称其为图 **G** 的桥 (**Bridge**)。

如果一个简单无向图的任意两个顶点间恰有一条边, 则称其为完全图 (**Complete Graph**), n 个顶点的完全图记作 \mathbf{K}_n 。在无向图 **G** 中, 若点子集 $K \subseteq V(\mathbf{G})$ 的导出子图是完全图, 则称 K 为 **G** 的团 (**Clique**)。如果一个简单有向图的任意两个顶点之间有且仅有一条有向边, 那么称它为竞赛图 (**Tournament**),

本文中用 \mathbf{T} 表示竞赛图。

在图 \mathbf{G} 中, 若点子集 $I \subseteq V(\mathbf{G})$ 的导出子图无边 (Edgeless), 即 $E(\mathbf{G}[I]) = \emptyset$, 则称 I 为图 \mathbf{G} 的独立集 (**Independent Set**)。在无向图 \mathbf{G} 中, 如果边子集 $M \subseteq E(\mathbf{G})$ 中的边两两没有公共端点, 则称 M 为 \mathbf{G} 的匹配。如果图 \mathbf{G} 中没有环, 则称 \mathbf{G} 是无环的 (**Acyclic**)。特别地, 在无向图中, 无环的连通图称为树 (**Tree**), 而无环的图称为森林 (**Forest**)。如果图 \mathbf{G} 中的每一条边都和集合 $X \subseteq V(\mathbf{G})$ 有公共顶点, 则称 X 为图 \mathbf{G} 的顶点覆盖 (**Vertex Cover**); 换言之, X 是图 \mathbf{G} 的顶点覆盖当且仅当 $\mathbf{G} - X$ 无边。如果图 \mathbf{G} 中的每一个环都和集合 $F \subseteq V(\mathbf{G})$ 有公共顶点, 则称 F 为图 \mathbf{G} 的反馈点集 (**Feedback Vertex Set**); 换言之, F 是图 \mathbf{G} 的反馈点集当且仅当 $\mathbf{G} - F$ 无环。如果图 \mathbf{G} 中的每一个环都和集合 $F \subseteq E(\mathbf{G})$ 有公共边, 则称 F 为图 \mathbf{G} 的反馈边集 (**Feedback Edge/Arc Set**); 换言之, F 是图 \mathbf{G} 的反馈边集当且仅当 $\mathbf{G} - F$ 无环。

给定点子集或边子集 T , 经过 T 中元素的环称为 T -环 (**T -cycle**), 经过 T 中元素的三角形称为 T -三角形 (**T -triangle**)。如果图 \mathbf{G} 中没有 T -环, 则称 \mathbf{G} 是 T -无环的 (**T -Acyclic**)。如果图 \mathbf{G} 中的每一个 T -环都和集合 $S \subseteq V(\mathbf{G})$ 有公共顶点, 则称 S 为图 \mathbf{G} 的 T -反馈点集 (**T -Feedback Vertex Set**); 换言之, S 是图 \mathbf{G} 的 T -反馈点集当且仅当 $\mathbf{G} - S$ 是 T -无环的。如果图 \mathbf{G} 中的每一个环都和集合 $S \subseteq E(\mathbf{G})$ 有公共边, 则称 S 为图 \mathbf{G} 的 T -反馈边集 (**T -Feedback Edge/Arc Set**); 换言之, S 是图 \mathbf{G} 的 T -反馈边集当且仅当 $\mathbf{G} - S$ 是 T -无环的。

2.2 精确与参数算法

本文主要研究反馈集相关问题的确定性算法设计, 本节将介绍精确算法和参数算法的相关的理论基础。

2.2.1 精确算法

本小节介绍精确算法以及相关复杂度理论的基本概念。

在本文使用的 Landau 记号中, 符号 $\mathcal{O}(\cdot)$ 表示渐近上界 (Asymptotic Upper Bound), 符号 $o(\cdot)$ 表示渐近严格上界 (Asymptotically Strict Upper Bound), 符号 $\Theta(\cdot)$ 表示渐近紧界 (Asymptotically Tight Bound)。例如, 多项式函数可表示为 $n^{\mathcal{O}(1)}$, 亚指数函数可表示为 $2^{o(n)}$ 。此外, 本文将多次使用符号 $\mathcal{O}^*(\cdot)$ 以省略多项式渐近部分: 对于任意函数 $f(n)$, 有

$$\mathcal{O}^*(f(n)) := f(n) \cdot n^{\mathcal{O}(1)}.$$

在计算理论中，判定性问题（**Decision Problem**）是一类答案只有“是”和“否”（“有解”和“无解”）两种可能性的问题。

本文参照了专著 [2, 92] 中对计算问题的简写规范：“变体名称-问题名称(输入类)”。例如，当问题 P 输入限制为类 \mathcal{G} 时，该问题称为 \mathcal{G} 上 P 问题，简写为 $P(\mathcal{G})$ ；问题 P 的名为“Special”的变体问题，称为 Special 版本的 P 问题，简写为 Special- P 。

某一问题的实例（**Instance**）是此问题的一个具体的输入，表示这个问题需要解决的一个具体对象。本文通常使用大写花体字母 \mathcal{I} 表示问题实例。

在复杂度理论中， P 表示多项式时间（**Polynomial-Time**）可判定的判定性问题类，若某问题属于 P 类，本文也称该问题能在多项式时间内求解。 NP 表示非确定性多项式时间（**Non-Deterministic Polynomial-Time**）可判定，或等价地，多项式时间可验证（**Verifiable**）的判定性问题类，若某个问题属于 NP 类，本文也称该问题是 NP 问题。例如，匹配问题^[93]、反馈边集问题（即生成树问题^[94]）等都可在多项式时间内求解，亦可在多项式时间内验证。

定义 2.1 (多项式时间归约) 给定两个判定性问题 P_1 和 P_2 ，若存在算法 \mathcal{R} ，可将问题 P_1 的任何实例 \mathcal{I}_1 作为输入，转换并输出问题 P_2 的实例 \mathcal{I}_2 ，满足以下条件

- 算法 \mathcal{R} 是多项式时间算法；
- \mathcal{I}_1 和 \mathcal{I}_2 等价，即 \mathcal{I}_1 有解当且仅当 \mathcal{I}_2 有解。

则称 \mathcal{R} 是问题 P_1 到问题 P_2 的多项式时间归约（**Polynomial-Time Reduction**），也称问题 P_1 可多项式时间归约（**Polynomial-Time Reducible**）到问题 P_2 。

若所有 NP 问题都可以多项式时间归约到某个问题 P ，则称问题 P 是 NP -难的（**NP-Hard**）。既是 NP -难又是 NP 的判定性问题称为 NP -完全（**NP-Complete**）问题。例如，旅行商问题、顶点覆盖问题、团问题、独立集问题、反馈点集问题等都是 NP -完全问题^[3]。在计算机科学中，精确算法是指确定性地求解某个问题，并在有限时间内给出准确答案的算法。在 $P \neq NP$ 的假设下，任何 NP -难问题及其对应的优化问题均无多项式时间的精确算法。

从基本的定义出发，算法的运行时间是输入长度的函数。在计算复杂性理论中，输入实例的长度定义为通过给定字母表编码实例后的字符串比特数，例如，图的输入长度应为 $\Theta(n + m \log n)$ 。值得强调的是，对于绝大多数自然问题而言，输入实例的长度与我们通常讨论的输入规模并不完全相同。对图论问题而言，传统衡量输入规模的参数是输入图的点数 n 或边数 m 。无论使用点数 n 或是边数 m 作为参数来衡量输入实例的规模，在多项式意义下都是等价的，即，它们的大小能够相互被多项式控制（**Polynomially Bounded**）。但是对于 NP -难问题而言，以不同

参数作为输入实例的规模可能导致时间复杂度呈现较大的差异。在 NP-难图问题的精确算法领域，都遵循既定的传统，以图的点数 n （较少情况使用图的边数 m ）来衡量输入规模。例如，旅行商问题目前最快精确算法复杂度为 $\mathcal{O}(2^n n^2)$ ^[35, 95]，顶点覆盖问题/团问题/独立集问题目前最快精确算法复杂度为 $\mathcal{O}(1.1996^n)$ ^[96]，而反馈点集目前最快精确算法复杂度为 $\mathcal{O}(1.7110^n)$ ^[52, 97]。

2.2.2 参数算法与核心化

精确的指数时间算法并不是唯一涉及困难问题准确解的算法领域。Downey 和 Fellows^[43, 48]在引入参数化复杂性理论时提出了一个精细化分析困难问题的通用算法和复杂性框架。本小节将介绍参数算法以及核心化算法的基本概念。

在参数算法领域，算法复杂度不仅与输入长度有关，而且与问题预设的参数有关，该参数可能不依赖于输入长度。粗略地讲，参数算法领域研究的是：对于 NP-难问题，如何设计相对高效的算法，其时间复杂度的超多项式部分仅是参数的函数，而与输入长度无关；换言之，此算法的时间复杂度中，与输入长度有关的部分可以被多项式控制。特别地，可以认为精确算法是参数算法的特例，图论问题的精确算法可以视为以顶点数为参数的参数算法。但是，从参数复杂性的角度而言，将顶点数作为参数并不具有太大的意义。这是因为在参数算法研究的问题中，参数与输入长度往往没有明确的数量关系。因此，参数算法与精确算法的相关算法设计技术具有一定差异性。

定义 2.2 (参数化) 经典判定性问题 P 的参数化 (Parameterization) 定义为将 P 的任一实例 \mathcal{I} 映射到自然数的可计算函数 (Computable Function) $\kappa: \mathcal{I} \mapsto k$ 。问题 P 关于参数化 κ 的参数化问题定义为 $\bar{P} = (P, \kappa)$ ，称 \bar{P} 为 P 以 κ 为参数的参数化问题。参数化问题 \bar{P} 的实例为 $\bar{\mathcal{I}} = (\mathcal{I}, \kappa(\mathcal{I})) = (\mathcal{I}, k)$ 。

在上下文不引起混淆的情况下，本文通常使用 $k = \kappa(\mathcal{I})$ 表示输入实例的参数，使用 n 表示输入实例规模。参数化具有强大的描述能力，在理论层面足够刻画问题的结构特征^[92]。例如，针对特定图类的图论问题可视为原图问题在特定参数设定下的参数化形式。因此，从某种程度上讲，针对特定图类的图论问题的算法研究，实质上是对某一特定参数算法的探究。

定义 2.3 (固定参数可解) 给定参数化问题 (P, κ) ，若存在可计算函数 $f(\cdot)$ ，使得对于任意的实例 (\mathcal{I}, k) ，可在 $\mathcal{O}^*(f(k)) = f(k)n^{\mathcal{O}(1)}$ 时间内求解，则称问题 P （在以 κ 为参数下）是固定参数可解的。

所有固定参数可解的参数化问题构成的全体称作固定参数可解类，记作 FPT，而属于 FPT 类的参数化问题是 FPT 问题。一个经典问题是 P 问题当且仅当在以

$\kappa = \mathcal{O}(1)$ 为参数下是 FPT 问题的。对于众多图问题而言，最常见的参数是解集大小 k ，例如，顶点覆盖和反馈点集问题均为 FPT 的^[39,40]，并且顶点覆盖问题目前最快参数算法的时间复杂度为 $\mathcal{O}^*(1.2529^k)$ ^[98]，而反馈点集问题目前最快参数算法的时间复杂度为 $\mathcal{O}^*(3.46^k)$ ^[52]。

类似于经典复杂性理论，在参数复杂理论中，FPT 被视为是高效的求解算法。同时也可以利用参数化归约（Parameterized Reduction）定义其他参数复杂性。

定义 2.4 (参数化归约) 设 \bar{P}_1 和 \bar{P}_2 为两个参数化问题。若算法 \mathcal{R} 将问题 \bar{P}_1 的实例 (\mathcal{I}_1, k_1) 作为输入，转换并输出问题 \bar{P}_2 的实例 (\mathcal{I}_2, k_2) ，满足以下条件

- 算法 \mathcal{R} 的运行时间为 $\mathcal{O}^*(f(k_1))$ ，其中 $f(\cdot)$ 为可计算函数；
- \mathcal{I}_1 和 \mathcal{I}_2 等价，即 \mathcal{I}_1 有解当且仅当 \mathcal{I}_2 有解；
- $k_2 \leq g(k_1)$ ，其中 $g(\cdot)$ 是可计算函数。

则称 \mathcal{R} 是问题 \bar{P}_1 到问题 \bar{P}_2 的 FPT-归约 (FPT-Reduction)，或问题 \bar{P}_1 可以 FPT-归约到问题 \bar{P}_2 。

若算法 \mathcal{R} 满足

- 算法 \mathcal{R} 是多项式时间算法；
- \mathcal{I}_1 和 \mathcal{I}_2 等价，即 \mathcal{I}_1 有解当且仅当 \mathcal{I}_2 有解；
- $k_2 \leq k_1$ 。

则称 \mathcal{R} 是问题 \bar{P}_1 到问题 \bar{P}_2 的保持参数（不增）的多项式时间归约，或 \bar{P}_1 可以多项式时间地归约到问题 \bar{P}_2 ，且保持参数（不增）。

利用 FPT-归约，可以定义常用的参数复杂性类。使用复杂度类 $W[d, t]$ 表示能够 FPT-归约到（以权重为参数的）Weighted-SAT(d -Depth t -Weft CIRC) 问题的参数化问题类。定义复杂性类

$$W[t] = \bigcup_{d \in \mathbb{N}} W[d, t].$$

使用复杂度类 $W[\text{SAT}]$ 和 $W[\text{P}]$ 分别表示能够 FPT-归约到 Weighted-SAT(PROP) 问题和 Weighted-SAT(CIRC) 问题的参数化问题类（问题定义见附录 A）。则有包含关系

$$\text{FPT} = W[0] \subseteq W[1] \subseteq \dots \subseteq W[t] \subseteq \dots \subseteq W[\text{SAT}] \subseteq W[\text{P}].$$

$\text{FPT} \neq W[1]$ 蕴含了 $\text{P} \neq \text{NP}$ ^[41,42]。若所有 $W[t]$ 中的问题都可以 FPT-归约到某个参数化问题，则称此问题是 $W[t]$ 难的 ($W[t]$ -Hard)。既是 $W[t]$ -难又属于 $W[t]$ 的参数化问题称为 $W[t]$ -完全的 ($W[t]$ -Complete)。例如，独立集问题和团问题均为 $W[1]$ -难的^[39,40]，碰撞集问题和集合覆盖问题是 $W[2]$ -难的^[39,99]。在 $\text{FPT} \neq W[1]$

的假设下, $\mathsf{W}[1]$ -难的问题没有 FPT 算法, 即不存在时间复杂度形如 $\mathcal{O}^*(f(k))$ 的参数算法。

相比 FPT-归约, 保持参数的多项式时间归约的定义更强, 因此能够更准确地刻画参数化问题之间的关系。特别地, 若存在问题 \bar{P}_1 到 \bar{P}_2 的保持参数的多项式时间归约, \bar{P}_2 可在 $\mathcal{O}^*(f(k_2))$ 时间内求解蕴含着 \bar{P}_1 可在 $\mathcal{O}^*(f(k_1))$ 时间内求解。

如果一个参数化问题 \bar{P} 能够在多项式时间内归约到该问题自身, 并且保证输出实例的参数和长度均能够被输入实例的参数控制, 那么这种归约称 \bar{P} 的核心化 (Kernelization)。核心化是一种研究多项式时间预处理算法的系统性方法, 也是设计参数化算法的重要工具。预处理或数据压缩 (Data Reduction) 普遍应用于实际的计算机算法实现中。在处理 NP-难问题时, 预处理算法可以高效地解决问题实例中“容易的部分”, 并将实例的简化到仅剩下“困难的核心部分”。换言之, 该方法的思想是在多项式时间内压缩 (但不一定解决) 实例, 从而得到规模更小且等价的实例, 称为核 (Kernel)。必要时, 只需调用精确算法的对核进行求解便能够判定原问题实例。

如何衡量预处理算法的压缩能力是需要首先明确的。事实上, 对于 NP-难问题, 如果存在多项式算法, 可以将任意实例的长度压缩至少一个比特, 则意味着 $\mathsf{P} = \mathsf{NP}$ 。因此, 在很长一段时间里, 无法对预处理这一重要概念给出严格且有价值的刻画。这导致多项式时间预处理方法缺乏理论支撑, 从而无法进行有效的理论分析。但是在参数复杂性框架下, 使用参数的某一函数控制实例规模能够完美地填补相关理论空白。至此, 被称为核心化算法的数据压缩技术成为了参数算法领域重要的研究对象。现给出核心化的严格定义。

定义 2.5 (核心化) 给定参数化问题 (P, κ) , 若存在算法 \mathcal{K} , 可以将问题 (P, κ) 的任意实例 (\mathcal{I}, k) 作为输入, 转换并输出问题 (P, κ) 的另一个实例 (\mathcal{I}', k') , 满足以下条件

- 算法 \mathcal{K} 是多项式时间算法;
- 存在可计算函数 $f(\cdot)$, 使得参数 $k' \leq f(k)$;
- 存在可计算函数 $g(\cdot)$, 使得输出实例的长度 $n' \leq g(k)$ 。

则称算法 \mathcal{K} 为问题 P (关于参数化 κ) 的核心化算法, \mathcal{I}' 为问题 P (关于参数化 κ) 的核, 核的大小为 $g(\cdot)$ 。

一个参数化问题是 FPT 的当且仅当其存在核心化算法^[2, 92]。

此外, 核的大小刻画了核心化算法的压缩能力。对于问题 P 及其大小为 $g(k)$ 的 (关于参数化 κ 的) 核, 如果 $g(k) = k^{\mathcal{O}(1)}$, 则称问题 P 存在多项式核 (**Polynomial Kernel**); 如果 $g(k) = \mathcal{O}(k)$, 则称问题 P 存在线性核 (**Linear Kernel**)。

例如，顶点覆盖问题有点集大小不超过 $2k$ 的线性核^[100,101]，反馈集点问题有点集大小不超过 $2k^2 + k$ 的平方核^[102]。

另一方面，核心化领域也存在一些关于下界的研究方向，目前这些下界结果均以比特数来衡量核的大小。例如，在 $\text{NP} \subsetneq \text{coNP/poly}$ 假设下，对于任意小的正实数 $\varepsilon \in \mathbb{R}^+$ ，顶点覆盖问题和反馈点集问题均不存在比特数为 $\mathcal{O}(k^{2-\varepsilon})$ 的核^[103]。

2.2.3 精确算法与参数算法的联系

本小节将介绍精确算法与参数算法的联系，旨在阐明参数算法可以作为设计精确算法的强大的工具。事实上，许多精确算法的设计都建立在一个或多个参数算法的基础之上。

特别是 2016 年 Fomin 等人^[97,104]提出的基于单调局部搜索的精确算法框架。在该框架中，如果给定一个单指数时间的参数算法，可以获得一个非平凡单指数时间的精确算法，这为当时大量 NP-难题的精确算法带来了显著的改进。直至今日，相当大一部分 NP-难题的最快精确算法记录依然是由此框架下发展出的精确算法所保持。

碰撞集问题是适用于单调局部搜索 (Monotone Local Search) 框架的一个十分重要且典型的例子^[3]。考虑一个集合系统 (U, \mathcal{F}) ，其中 \mathcal{F} 为集合 U 上的子集族。若子集 $S \subseteq U$ 与集族 \mathcal{F} 的每一个子集合均有公共元素，则称其为此集合系统 (U, \mathcal{F}) 的碰撞集。设 Π 是一个关于集族的具有遗传性 (Hereditary) 的性质，这里遗传性是指：若 \mathcal{F} 满足性质 Π ，则对任意子集 $W \subseteq U$ ，集族 $\{X \in \mathcal{F} : X \subseteq W\}$ 也满足性质 Π 。则可定义 Π 上碰撞集问题。

Π 上碰撞集问题 (Hitting Set (Π))

输入：一个元素集合 U ，一个子集族 $\mathcal{F} \subseteq 2^U$ 和一个自然数 $k \in \mathbb{N}$ ，其中 \mathcal{F} 满足性质 Π 。

输出：判断是否存在一个大小不超过 k 的碰撞集。

若性质 Π 为 “ U 的任意子集”，即不对 \mathcal{F} 加以任何限制时，Hitting Set (Π) 问题简记为 Hitting Set 问题。Hitting Set 问题是一个高度难解的组合优化问题，当以解集大小 k 为参数时，精确求解此问题甚至获得常数近似解也是 W[2]-难的^[39,99]。目前，Hitting Set 问题只存在平凡的计算复杂度为 $\mathcal{O}^*(2^n)$ 的精确算法。而在 SETH 下，对任意正实数 $\varepsilon > 0$ ，不存在运行时间为 $\mathcal{O}((2 - \varepsilon)^n)$ 的精确算法^[33]。

当性质 Π 是 “ \mathcal{F} 中的子集的大小不超过 d ” 时，称这类问题为 d -碰撞集问题。

***d*-碰撞集问题 (*d*-Hitting Set)**

输入: 一个元素集合 U , 一个子集族 $\mathcal{F} \subseteq 2^U$ 和一个自然数 $k \in \mathbb{N}$, 其中 \mathcal{F} 中的任意集合最多包含 d 个元素。

输出: 判断是否存在一个大小不超过 k 的碰撞集。

当 d 取 2 时, 2-碰撞集问题等价于经典的顶点覆盖问题。对于给定的 $d \geq 2$, d -碰撞集问题均未 FPT 的, 并且假定能在 $\mathcal{O}^*(\xi_d^k)$ 的时间内求解。目前已经证明, $\xi_2 \leq 1.2529^{[98]}$ 、 $\xi_3 \leq 2.0755^{[54]}$ 、 $\xi_4 \leq 3.0755^{[61, 62]}$ 以及 $\xi_5 \leq 4.0755^{[61, 62]}$, 且当 $d \geq 6$ 时, ξ_d 不超过三次方程

$$x^3 - dx^2 + (d-1)x - 1 = 0$$

的最大实根^[53], 进而可得渐近性 $\xi_d \leq d - 1 + \mathcal{O}(d^{-2})$ 。

定理 2.6 (单调局部搜索 [97, 104]) 设 Π 是集合系统上一个具有遗传性 (Hereditary) 的集族。若 $\text{Hitting Set}(\Pi)$ 可在 $\mathcal{O}^*(\alpha^k)$ 时间, 则 $\text{Hitting Set}(\Pi)$ 可在 $(2 - \alpha^{-1})^n \cdot 2^{o(n)}$ 时间内求解。

定理3.6表明, 若某问题可以用 $\text{Hitting Set}(\Pi)$ 问题进行描述, 那么只要存在单指数时间的参数算法, 都能够基于单调局部搜索框架得到非平凡的精确算法。并且在某一算法范式下, 已达理论最优^[?]。例如, 对于任意给定的正整数 d , 都可以设计出 d -Hitting Set 打破 2^n 的精确算法。具体地, 若 d -Hitting Set 可以在 $\mathcal{O}^*(\xi_d^k)$ 时间内求解, 则也可以在 $\varsigma_d^n \cdot 2^{o(n)}$ 时间内求解, 其中 $\varsigma_d = 2 - \xi_d^{-1} = 2 - (d-1)^{-1} + \mathcal{O}(d^{-4})$ (参见表3-1)。

表 2-1 d -碰撞集的参数与精确算法复杂度

d 值	3	4	5	6	7	8	9	10	100
$\xi_d \leq$	2.0755	3.0755	4.0755	5.0490	6.0330	7.0237	8.0178	9.0139	99.0002
$\varsigma_d \leq$	1.5182	1.6749	1.7547	1.8020	1.8343	1.8577	1.8753	1.8891	1.9899

特别地, 若对于任何满足性质 Π 的集族 \mathcal{F} 通过隐式子的方式表示, 则称 $\text{Hitting Set}(\Pi)$ 问题是隐式碰撞集问题。例如, 可以将 \mathcal{F} 用图结构进行编码, 并且可通过图结构在多项式时间内判定一个集合是否属于 \mathcal{F} , 那么对应的图问题就属于隐式碰撞集问题。设 Π 是具有遗传性的图族, 即若 $\mathbf{G} \in \Pi$, 则 \mathbf{G} 的所有导出子图都包含在 Π 中。对于任意的 Π , 都存在关于 Π 的顶点删除问题 (Vertex Deletion Problem), 定义如下:

***\Pi* 上顶点删除问题 (Vertex Deletion (Π))**

输入: 一个无向或有向图 \mathbf{G} 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在一个大小不超过 k 的点集 $S \subseteq V(\mathbf{G})$, 使得 $\mathbf{G} - S \in \Pi$ 。

Vertex Deletion (Π) 涵盖了大量经典的图论问题，并且所有 Π -顶点删除问题均属于隐式碰撞集问题，故也适用于单调局部搜索框架。例如，当 Π 取所有无边图时，Vertex Deletion (Π) 为顶点覆盖问题；当 Π 取所有森林时，Vertex Deletion (Π) 为反馈点集问题。

除此之外，基于多个不同参数算法之间的均摊 (Trade-off) 思想，也能够设计出效果良好的精确算法。这种均摊思想在众多问题的精确算法设计中得到了广泛应用^[58, 105]。

定理 2.7 (多个参数算法的均摊) 设 k 为可计算问题 P 的一个参数，并且在 P 的任意实例中，都满足 $k \leq n$ 。假设 P 可以分别在 $\mathcal{O}^*(\alpha^k)$ 和 $\mathcal{O}^*(\beta^{n-k})$ 的时间内求解，那么存在时间复杂度为 $\mathcal{O}^*(2^{\gamma n})$ 的精确算法求解问题 P ，其中 $\alpha, \beta > 1$ 且均为常数， $\gamma = \log \alpha \log \beta / \log \alpha \beta$ 。

2.3 精确与参数算法设计技术

本节详述了本文所使用的精确和参数算法设计技术。当前，解决许多基础算法问题的高效算法均基于本节所讨论的技术范畴^①。

2.3.1 皇冠分解技术

皇冠分解 (Crown Decomposition) 是一种基于匹配理论的通用核心化与参数算法设计技术。该技术发展于以 König 和 Hall 的奠基性工作^[106–108]。

给定无向图 \mathbf{G} 不相交的点集 U 和 W ，如果匹配 $M \subseteq E(\mathbf{G})$ 覆盖了 U 中的每一个顶点，且 M 中每一条边恰好连接了 U 和 W 中的各一个顶点，则称 M 是 U 到 W 的匹配。

定义 2.8 (皇冠分解) 无向图 \mathbf{G} 的皇冠分解是满足以下条件的点集 $V(\mathbf{G})$ 的三划分 (C, H, R) 。

- 点集 C 是非空独立集；
- 点集 C 和 R 之间没有边相连，即点集 H 割开了 C 和 R ；
- 图 \mathbf{G} 中存在连接 H 到 C 的匹配。

皇冠分解可能不是唯一的，也可能不存在，如三角形 \mathbf{C}_3 不存在皇冠分解。但可利用 Hopcroft-Karp 算法^[109]确保在一定条件下总能够快速找到一个皇冠分解。

定理 2.9 (Hopcroft-Karp 算法) 设 \mathbf{G} 为二部图，其中 A 和 B 分别为点集 $V(\mathbf{G})$ 两个独立集部分。可以在 $\mathcal{O}(m\sqrt{n})$ 时间内得到 \mathbf{G} 的一个最大匹配和一个最小顶点覆盖。进一步，在 $\mathcal{O}(m\sqrt{n})$ 时间内，要么找到一个 A 到 B 的匹配，要么找到子

^① 除了动态规划算法，本文涉及的所有算法设计技术均能够保证在多项式空间完成，因此在无特殊说明下，本文提出的算法的空间复杂度均为多项式的。

集 $A' \subseteq A$ 满足 $|N(A')| < |A|$ 成立。

引理 2.10 (皇冠引理) 设 \mathbf{G} 是不包含孤立点的无向图。若图 \mathbf{G} 的顶点数 $n \geq 3k + 1$, 则存在多项式时间算法, 要么找到一个大小为 $k + 1$ 的匹配; 要么找到 \mathbf{G} 的一个皇冠分解。

皇冠分解有诸多变体, 如平衡皇冠分解^[110]。基于匹配的图分解也可视为皇冠分解的变体, 如 Dulmage-Mendelsohn 分解^[111, 112]和 Gallai-Edmonds 分解^[93, 113, 114]。在本文第七章中, 将使用 Dulmage-Mendelsohn 分解设计参数算法。

2.3.2 树分解技术

树分解技术在现代图论与组合优化领域占据着至关重要的地位。此技术是一种能够将任意无向图映射到树的图结构分解方法, 旨在刻画一个图与树结构的相似度, 并能为许多图论计算问题的高效算法设计提供支持。

定义 2.11 (树分解) 无向图 \mathbf{G} 的树分解 (**Tree Decomposition**) 是一个满足以下条件的二元组 $(\mathcal{T}, \{B_t\}_{t \in V(\mathcal{T})})$, 其中 \mathcal{T} 是一颗树, 并且 \mathcal{T} 的每一个顶点 t 都对应一个点集 $B_t \subseteq V(\mathbf{G})$, 称作包 (**Bag**)。

- 图 \mathbf{G} 中的每一个顶点都至少在一个包中, 即

$$V(\mathbf{G}) = \bigcup_{t \in V(\mathcal{T})} B_t;$$

- 图 \mathbf{G} 中的每一条边都至少在一个包的导出子图中, 即

$$E(\mathbf{G}) = \bigcup_{t \in V(\mathcal{T})} E(G[B_t]);$$

- 令 $X_v = \{t \in V(\mathcal{T}) : v \in B_t\}$ 表示所有包含了顶点 v 的包所对应的树上的节点集。那么图 \mathbf{G} 中任何顶点 v 所对应的集合 X_v 导出的子图都是一颗树, 即 $\mathcal{T}[X_v]$ 是连通的。

直观上讲, 树分解将给定图 \mathbf{G} 的顶点表示为一棵树的子树, 使得图 \mathbf{G} 中任意两个相邻顶点所对应的子树交集非空, 从而使得图 \mathbf{G} 成为了这棵树上子树的相交图 (一种弦图) 的一个子图。基于树分解, 可以通过树宽 (Treewidth) 表征一个图与树结构的相似程度。

定义 2.12 (树宽) 设无向图 \mathbf{G} 的一个树分解为 $(\mathcal{T}, \{B_t\}_{t \in V(\mathcal{T})})$ 。树分解的宽度, 定义为

$$\max_{t \in V(\mathcal{T})} |B_t| - 1.$$

无向图 \mathbf{G} 的树宽, 是指 \mathbf{G} 的所有可能树分解中树宽度的最小值, 记作 $\text{tw} = \text{tw}(\mathbf{G})$ 。

特别地，连通图 \mathbf{G} 的树宽为 1 当且仅当 \mathbf{G} 是一颗树。计算一般图的树宽是 NP-难的^[115]，目前可以在 $\mathcal{O}(2^{\text{tw}^2} n^4)$ 时间内精确求解树宽^[116]，或者在 $\mathcal{O}(2^{\text{tw}/\varepsilon} n^4)$ 时间内得到近似率为 $1 + \varepsilon$ 的近似解^[116]。在弦图等一些特殊图上，能够在多项式时间内进行树分解并精确计算树宽。事实上，一些高效的实验算法正是通过弦图计算^[117]任意图的树宽。

当树宽有界时，可以使用树分解上的标准动态规划算法求解许多图论问题，著名的 Courcelle 定理^[118]就是一个经典的例子。粗略地讲，Courcelle 定理指出，可以被一元二阶逻辑（Monadic Second-Order）描述的图问题以树宽为参数是 FPT 的。在本文第八章中，将使用树分解设计参数算法和精确算法。

2.3.3 分支搜索技术与度量治之

分支搜索技术，也被称为分支约简（Branch and Reduce）技术，是一个最为通用的精确算法与参数算法设计技术。许多首次突破 2^n 或 2^k 计算复杂度瓶颈的非凡算法均采用了分支搜索技术。分支搜索技术一般通过递归算法形式进行描述，算法中包括一些约简规则（**Reduction Rule**）和分支规则（**Branching Rule**）。每一个规则都会对实例进行一些操作，然后进行递归调用。为了估计每步规则对求解问题的贡献，需要为实例选取一个度量 $\mu(\cdot)$ 。在算法的每一个步骤中，需要保证度量不增，并且度量为非正值的实例可在多项式时间内解决。

约简规则将当前实例转换为一个“更小”的实例。若约简规则保证输入实例有解当且仅当输出实例有解，则称该约简规则是正确的。分支规则将当前实例 \mathcal{I} 分为多个子实例 $\mathcal{I}_1, \mathcal{I}_2, \dots, \mathcal{I}_r$ ，($i \in [r]$) 并且满足 $\mu(\mathcal{I}_i) < \mu(\mathcal{I})$ 。若实例 \mathcal{I} 有解当且仅当至少一个子实例有解，则称该分支规则是正确的。

为了分析运行时间，需要采用度量治之分析手段控制搜索树的大小。用 $\mathcal{T}(\mu)$ 表示算法关于度量 μ 的搜索树的大小上界。具体地，为每一个分支规则建立如下形式的递归关系式

$$\mathcal{T}(\mu) \leq 1 + \sum_{i \in [r]} \mathcal{T}(\mu_i),$$

其中 $\mu_i = \mu(\mathcal{I}_i)$ ($i \in [r]$)。该递归关系可使用分支向量（**Branching Vector**）

$$(\mu - \mu_1, \mu - \mu_2, \dots, \mu - \mu_r)$$

表示。方程 $x^\mu - \sum_{i=1}^r x^{\mu_i} = 0$ 的唯一正实根称为该分支规则的分支因子（**Branching Factor**）。设 $\mathbf{a} = (a_1, a_2, \dots, a_p)$ 和 $\mathbf{b} = (b_1, b_2, \dots, b_q)$ 为两个分支向量。若 $p \leq q$ 并且 $a_i \geq b_i$ ($i \in [p]$) 成立，则称 \mathbf{a} 不劣于（**not worse than**） \mathbf{b} 。一个简单的性质是：

如果分支向量 \mathbf{a} 不劣于 \mathbf{b} ，那么 \mathbf{a} 的分支因子不大于 \mathbf{b} 的分支因子。在基于分支搜索技术的算法中，如果所有分支向量中的最大分支因子是 α ，那么算法的搜索树大小能够被 $\mathcal{O}^*(\alpha^\mu)$ 所控制。在本文后续章节中，将充分使用分支搜索技术联合其他相关技术设计参数算法与精确算法。

2.3.4 迭代压缩技术

迭代压缩技术，由 Reed 等人^[119]在 2004 年被首次提出，标志着参数算法领域的一个重大进步。该技术问世后，包括有向反馈点集问题在内的多个当时被认为难以攻克的问题均相继获得了首个 FPT 时间的算法，部分问题甚至能在单指数时间内求解^[67, 120–125]。

迭代压缩技术是一个适用于遗传性质 Π 上顶点删除问题的参数算法框架。回顾定义，这类问题旨在给定图中找出一个大小不超过 k 的点集，使得将其移除后的图满足某个全局性质 Π 。当以解集大小 k 为参数时，可以通过反复求解一个压缩版变体问题，从而得到原问题的最优解。

首先，定义 Vertex Deletion (Π) 问题的压缩版本 (Compression Version) 和分离版本 (Disjoint Version)，分别记作 Compression-Vertex Deletion (Π) 和 Disjoint-Vertex Deletion (Π)。

压缩版本 Π 上顶点删除问题 (Compression-Vertex Deletion (Π))

输入: 一个无向或有向图 \mathbf{G} ，一个点集 \bar{S} 使得 $\mathbf{G} - \bar{S} \in \Pi$ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在一个大小不超过 k 的点集 $S \subseteq V(\mathbf{G})$ ，使得 $\mathbf{G} - S \in \Pi$ 。

分离版本 Π 上顶点删除问题 (Disjoint-Vertex Deletion (Π)))

输入: 一个无向或有向图 \mathbf{G} ，一个不可删点集 U 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在一个大小不超过 k 的点集 $S \subseteq V(\mathbf{G}) \setminus U$ ，使得 $\mathbf{G} - S \in \Pi$ 。

接下来阐述如何使用压缩版本和分离版本构建算法框架求解 Vertex Deletion (Π) 问题的实例 $\mathcal{I} = (\mathbf{G}, k)$ (参见图3-1)。对图 \mathbf{G} 中的顶点编号 $V(\mathbf{G}) = \{v_1, v_2, \dots, v_n\}$ 。令 $V_i = \{v_1, v_2, \dots, v_i\}$ 以及 $\mathbf{G}_i = \mathbf{G}[V_i]$ 。然后，通过遍历 $i = 1, 2, \dots, n$ 并反复计算子实例 $\mathcal{I}_i = (\mathbf{G}_i, k)$ 的解。注意到， \mathbf{G}_i 作为图 \mathbf{G} 的导出子图，如果实例 \mathcal{I}_i 无解，则表明实例 \mathcal{I} 无解。对于 \mathcal{I}_1 ，因为顶点个数是常数，所以可以在常数时间内求解实例 \mathcal{I}_1 得到解集 S_1 。对于实例 \mathcal{I}_i ($i \in [n - 1]$)，设 S_i 为 \mathcal{I}_i 的解，令 $\bar{S}_i = S_i \cup \{v_{i+1}\}$ 。这意味着 $(\mathbf{G}_{i+1}, \bar{S}_{i+1}, k)$ 是 Compression-Vertex Deletion (Π) 问题的实例，且满足 $|\bar{S}_{i+1}| \leq k + 1$ 。根据定义，实例 $(\mathbf{G}_{i+1}, \bar{S}_{i+1}, k)$ 的解也是实例 \mathcal{I}_{i+1} 的解。因此，只需要迭代地多次求解压缩版本，便能够最终实现对原问题的求解，从而得到了压缩引理。

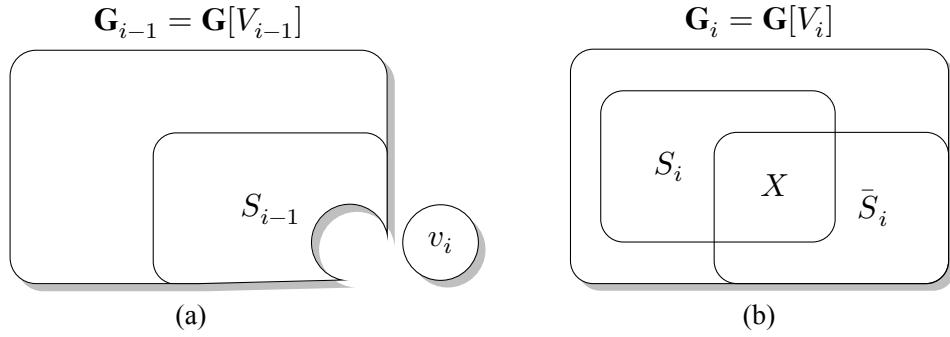


图 2-1 迭代压缩技术框架示意图。(a) 原问题的子实例示意图 $\mathcal{I}_i = (\mathbf{G}_{i-1}, k)$ 。

其中图 $\mathbf{G}_{i-1} = \mathbf{G}[V_{i-1}]$ 且 S_{i-1} 是 \mathcal{I}_i 的一个解。(b) 压缩版本的实例示意图 $(\mathbf{G}_i, \bar{S}_i, k)$ 。其中图 $\mathbf{G}_i = \mathbf{G}[V_i]$, $\bar{S}_i = S_{i-1} \cup \{v_i\}$ 是一个大小为 $k+1$ 的“近似解”， S_i 是一个精确解满足 $X = S_i \cap S'_i$ 。

引理 2.13 (压缩引理) 集合 \bar{S} 大小不超过 $k+1$ 时，若压缩版本能在 $\mathcal{T}(k, n)$ 时间内求解，则原问题能在 $\mathcal{O}(n) \cdot \mathcal{T}(k, n)$ 时间内求解。

接下来考虑压缩版本的求解。考虑 Compression-Vertex Deletion (Π) 问题的实例 (\mathbf{G}, \bar{S}, k) ，其中 $|\bar{S}| \leq k+1$ 。当 $|\bar{S}| \leq k$ 时， \bar{S} 本身就是一个解，因此不妨设 $|\bar{S}| = k+1$ 。枚举解集与 \bar{S} 的交集（共有 2^{k+1} 种情况），对于每一个交集 $X \subseteq \bar{S}$ ，将 X 删除， \bar{S} 中剩余顶点均不可放入解集中。设 $U = \bar{S} \setminus X$ ，那么 $\mathcal{I}_U = (\mathbf{G}', U, k')$ 是 Disjoint-Vertex Deletion (Π) 问题的实例，其中 $\mathbf{G}' = \mathbf{G} - X$, $k' = k - |X|$ 且 $n' = n - |X|$ 。在实例 \mathcal{I}_U 中， U 的一个大小为 $k'+1$ 且 $\mathbf{G}' - U \in \Pi$ 。因此，只需要多次求解分离版本的实例，便能够完成对压缩版本的求解，进而求解原问题。

定理 2.14 (迭代压缩) 不可删点集 U 的大小不超过 $k+1$ 且 $\mathbf{G} - U \in \Pi$ 时，如果分离版本能在 $\mathcal{T}(k, n)$ 时间内求解，那么求解原问题的运行时间上界为

$$\sum_{i=0}^k \binom{k+1}{i} \cdot \mathcal{T}(k, n) \cdot n^{\mathcal{O}(1)}.$$

特别地，若 $\mathcal{T}(k, n) = \alpha^k n^{\mathcal{O}(1)}$ ，那么原问题能在 $(\alpha+1)^k n^{\mathcal{O}(1)}$ 时间内求解。

迭代压缩的强大之处在于，一旦满足特定条件的分离版本被证明是 FPT 的，则能直接说明原问题是 FPT 的。然而，为分离版本设计参数算法相比原问题要容易许多。进一步地，若分离版本能在单指数时间 $\mathcal{O}^*(\alpha^k)$ 内求解，那么原问题也可在单指数时间内求解，且时间复杂度仅增加至 $\mathcal{O}^*((\alpha+1)^k)$ 。另一方面，在迭代压缩框架中，由于没有引入相关问题的其他结构信息，仅通过穷举解集与 \bar{S} 的交，标准的迭代压缩算法无法突破 2^k 的时间复杂度瓶颈。本文第六章将结合迭代压缩和其他相关技术设计参数算法突破 2^k 。

2.3.5 平衡割技术

平衡割分为平衡点割 (Balanced Separator) 和平衡边割 (Balanced Cut)，是设计分治算法的重要且基础的工具。平衡割能够将一个图的顶点或边均匀地分割开，同时自身的大小能被控制住。存在多种不同版本的平衡割技术，它们的大小能够被不同的参数所控制，本文仅介绍使用最大度控制平衡边割的结论。

定理 2.15 (平衡边割 [75]) 设 \mathbf{G} 为一个没有自环和孤立点的多重无向图，设其最大度为 $\Delta(\mathbf{G}) = d$ ，存在点集的二划分 (P_0, P_1) 满足

$$\begin{cases} \frac{m}{4} - \frac{d}{2} \leq |E(\mathbf{G}[P_0])| \leq \frac{m}{4} + \frac{d}{2}; \\ \frac{m}{4} - \frac{d}{2} \leq |E(\mathbf{G}[P_1])| \leq \frac{m}{4} + \frac{d}{2}; \\ \frac{m}{2} - d \leq |E(P_0, P_1)| \leq \frac{m}{2} + d, \end{cases}$$

其中 $E(P_0, P_1)$ 表示端点分别在点集 P_0 和 P_1 的边集。此外，划分 (P_0, P_1) 可以在多项式时间内计算得到。

上述定理表明，当一个无向图的最大度有界时，能够高效地找到足够小的平衡边割，在此基础上则可以考虑采用分治思想设计高效算法。本文第六章使用了基于平衡边割的分治思想设计参数算法。

2.4 本章小结

本章第一部分主要介绍了图论的基础概念与符号体系，以及经典计算理论和参数复杂性理论的基本知识。作为计算复杂性理论中相对年轻的分支，参数复杂性理论使得一些在经典复杂性理论中的难解问题可以在参数化的视角下得到相对有效地解决。因此，参数复杂性理论不仅为 NP-难问题提供了新的可解性框架，也为精确算法设计开辟了新的技术路线。

本章第二部分对全文涉及的精确与参数算法的基础性技术框架进行了详细阐述。这些技术均为领域内成熟且强大算法工具，在大量图论或组合优化问题的算法设计中展现出广泛的适应能力。在后续第六章至第八章中，本文将针对特定的反馈集问题，结合多种技术设计，提出快速的精确和参数算法。此外，本文也将创新性地提出针对性的技术，这些技术并未在本章列出，而将在后续各章中逐一展开讨论。

第三章 理论基础

本章首先在3.1节对涉及全文的图论基础和主要术语进行了介绍，然后在3.2和3.3节分别概述了精确算法和参数算法的基本概念与关键技术。本文假定读者对图论和复杂度理论的基础概念已基本了解。全文所涉及的图论基本概念可参见专著 [90,91]，所涉及的精确与参数算法基本概念可参见专著 [1,2,92]。

3.1 图论基础

本文研究的反馈集相关问题属于经典图论问题，本节首先介绍全文统一使用的图论基本概念与符号系统。

在图论中，一个图 (Graph) 通常使用符号 \mathbf{G} 来表示，由其顶点集和边集唯一决定。图 \mathbf{G} 的顶点集表示为 $V(\mathbf{G})$ ，边集表示为 $E(\mathbf{G})$ 。在上下文不产生混淆的情况下，本文简记 $V = V(\mathbf{G})$ 和 $E = E(\mathbf{G})$ ，同时使用正整数 n 表示点集 $V(\mathbf{G})$ 的大小，用 m 表示边集 $E(\mathbf{G})$ 的大小。一个图中的顶点 (Vertex) 一般使用小写字母 u, v, w 等来表示。若 \mathbf{G} 是无向图 (Undirected Graph)，图中的 (无向) 边 (Edge) 表示为一些无序点对；若 \mathbf{G} 是有向图 (Directed Graph 或 Digraph)，图中的有向边 (Arc) 则表示为一些有序点对。图中的边表示为小写字母 e 或二元对 $(v, w) \in V(\mathbf{G})^2$ ，简记为 vw 。本文通常使用符号 \mathbf{D} 表示有向图。

若一条边的两个端点均为同一顶点 v ，则称此边为顶点 v 上的自环 (Loop)。无向图或有向图的边集是一个多重集合 (Multiset)，即集合可以包含多个相同的元素。若图中的边集包含两条或多条与同一对顶点相连接的边，则称这些边为重边或平行边 (Multiple Edges)。没有自环和平行边的图称为简单图 (Simple Graph)。

本文使用符号 $[n]$ 简记前 n 个正整数构成的集合 $\{1, 2, \dots, n\}$ 。图 \mathbf{G} 中的一条路或路径 (Path) 是至少包含一个顶点的子图 \mathbf{P} ，表示为

$$\mathbf{P} = (v_1, v_2, \dots, v_\ell),$$

其中连续的两个顶点相邻，即 $E(\mathbf{P}) := \{v_i v_{i+1} : i \in [\ell - 1]\} \subseteq E(\mathbf{G})$ 。路 \mathbf{P} 的长度是 \mathbf{P} 中顶点的个数 ℓ ，记作 $|\mathbf{P}|$ 。图 \mathbf{G} 中的一个环或圈 (Cycle) 是一个至少包含三个顶点的子图 \mathbf{C} ，表示为

$$\mathbf{C} = (v_1, v_2, \dots, v_\ell),$$

其中连续的两个顶点以及首尾顶点相邻，即 $E(\mathbf{C}) := \{v_i v_{i+1}, v_\ell v_1 : i \in [\ell - 1]\} \subseteq E(\mathbf{G})$ 。环 \mathbf{C} 的长度是 \mathbf{C} 中顶点的数量 ℓ ，记作 $|\mathbf{C}|$ 。图 \mathbf{G} 中连接了路 \mathbf{P} (或环 \mathbf{C})

中不相邻顶点的边称为路 **P** (或环 **C**) 的弦 (**Chord**)。长度为 ℓ 的路径和环分别记作 \mathbf{P}_ℓ 和 \mathbf{C}_ℓ 。特别地, 长度为 3 的环 \mathbf{C}_3 亦被称为三角形 (**Triangle**)。本文中的路和环要求其中的顶点互不同, 在一些文献中也被分别称为简单路和简单环。

一个图中的点子集或边子集一般使用 A, B, X, Y 等大写字母表示。给定图 **G**, 点子集 $X \subseteq V(\mathbf{G})$ 的导出子图 (**Induced Subgraph**), 记作 $\mathbf{G}[X]$, 是 **G** 的一个子图, 其点集为 X , 边集为两个端点都在 X 中的边构成的集合。从图 **G** 中删掉点子集 $X \subseteq V(\mathbf{G})$ 以及与 X 中顶点邻接的边得到的子图用 $\mathbf{G} - X := \mathbf{G}[V(\mathbf{G}) \setminus X]$ 表示。当某一点集为单元集时, 通常将省略集合括号, 如 $\mathbf{G} - \{v\}$ 可简写为 $\mathbf{G} - v$ 。

在图 **G** 中, 点子集 $X \subseteq V(\mathbf{G})$ 的 (开) 邻域 (**Neighborhood**) $N_{\mathbf{G}}(X)$ 是所有与 X 中的点相邻且不属于 X 中的点集, 即

$$N_{\mathbf{G}}(X) := \{w \in V(\mathbf{G}) \setminus X : v \in X, wv \in E(\mathbf{G}) \text{ 或 } vw \in E(\mathbf{G})\}$$

集合 $N_{\mathbf{G}}(v)$ 中的元素称为 v 的 (开) 邻居 (**Neighbor**)。点子集 X 的闭邻域 (**Closed Neighborhood**) 定义为

$$N_{\mathbf{G}}[X] := N_{\mathbf{G}}(X) \cup X.$$

集合 $N_{\mathbf{G}}[v]$ 中的元素称为 v 的闭邻居 (**Closed Neighbor**)。对于有向图 **D**, 点子集 $X \subseteq V(\mathbf{D})$ 的入邻域 (**In-Neighborhood**) 和出邻域 (**Out-Neighborhood**) 分别是不属于 X 的点中, 连入 X 的顶点构成的点集和 X 连出的顶点构成的点集, 分别记作

$$\begin{cases} N_{\mathbf{D}}^-(X) := \{w \in V(\mathbf{D}) \setminus X : v \in X, wv \in E(\mathbf{D})\}; \\ N_{\mathbf{D}}^+(X) := \{w \in V(\mathbf{D}) \setminus X : v \in X, vw \in E(\mathbf{D})\}. \end{cases}$$

入邻域和出邻域内的元素分别称为 v 的入邻居 (**In-Neighbor**) 与出邻居 (**Out-Neighbor**)。类似地, X 的闭入邻域 (**Closed In-Neighborhood**) 和闭出邻域 (**Closed Out-Neighborhood**) 分别定义为

$$\begin{cases} N_{\mathbf{D}}^-[X] := N_{\mathbf{D}}^-(X) \cup X; \\ N_{\mathbf{D}}^+[X] := N_{\mathbf{D}}^+(X) \cup X. \end{cases}$$

在图 **G** 中, 顶点 v 的度 $\deg_{\mathbf{G}}(v)$ 定义为以 v 为端点的边的数量 (自环计算两次)。那么当 **G** 为简单图时, 有 $\deg_{\mathbf{G}}(v) = |N_{\mathbf{G}}(v)|$ 。类似地, 在简单有向图 **D** 中, 顶点 v 的入度 (**In-Degree**) $\deg_{\mathbf{D}}^-(v)$ 和 v 的出度 (**Out-Degree**) $\deg_{\mathbf{D}}^+(v)$, 分别定义为其入邻域和出邻域的大小, 即 $\deg_{\mathbf{D}}^-(v) := |N_{\mathbf{D}}^-(v)|$, $\deg_{\mathbf{D}}^+(v) := |N_{\mathbf{D}}^+(v)|$ 。在上下文不产生混淆的情况下, 本文将省略邻域符号和度符号的下标。

图 **G** 的最大度 (**Maximum Degree**) 定义为图中所有顶点度的最大值, 即

$$\Delta(\mathbf{G}) := \max_{v \in V(\mathbf{G})} \deg_{\mathbf{G}}(v).$$

有向图 **D** 的最大入度 (**Maximum In-Degree**) 和最大出度 (**Maximum Out-Degree**) 分别定义为

$$\begin{cases} \Delta^-(\mathbf{D}) := \max_{v \in V(\mathbf{D})} \deg_{\mathbf{D}}^-(v); \\ \Delta^+(\mathbf{D}) := \max_{v \in V(\mathbf{D})} \deg_{\mathbf{D}}^+(v), \end{cases}$$

同时, 在部分文献中会使用到符号

$$\Delta^*(\mathbf{D}) := \max_{v \in V(\mathbf{D})} \min \{ \deg_{\mathbf{D}}^-(v), \deg_{\mathbf{D}}^+(v) \},$$

本文将其定义为有向图的定向最大度。

在无向图 **G** 中, 如果两个顶点 v 和 w 之间存在一条路连接它们, 则称它们是连通的 (**Connected**), 否则称它们是不连通的 (**Disconnected**)。如果无向图 **G** 的任意两个点均连通, 则称 **G** 为连通图 (**Connected Graphs**)。如果 $C \subseteq V(\mathbf{G})$ 是一个极大的点子集, 使得其导出子图是连通的, 则称其是 **G** 的连通分量 (**Connected Component**)。在本文中, 满足某种性质的极大/极小集合均是指其在包含意义下极大/极小 (Inclusion-Wise Maximal/Minimal), 换言之, 在所有满足该性质的集合中, 没有此集合的真超集/真子集 (Proper Superset/Proper Subset)。一个有向图 **D**, 如果去掉有向边的方向所得到的无向图是连通的, 则称有向图 **D** 是连通的或弱连通的 (**Connected**, **Weakly Connected**)。对于有向图 **D** 中的两个顶点 v 和 w , 如果存在从 v 到 w 的 (有向) 路, 则称 w 关于 v 可达 (**Reachable**) 或 v 可到达 w 。若有向图 **D** 中的顶点两两相互可达, 则称 **D** 是强连通的 (**Strongly Connected**)。如果 $C \subseteq V(\mathbf{D})$ 是一个极大的点子集, 使得其导出子图是强连通的, 则称 C 为 **D** 的强连通分量 (**Strongly Connected Component**)。

在图 **G** 中, 如果删掉点子集 $X \subseteq V(\mathbf{G})$ 后, 图的连通分量数量增加, 则称 X 为图 **G** 的点割 (**Separator**)。如果删掉边子集 $Y \subseteq E(\mathbf{G})$ 后, 图的连通分量数量增加, 则称 Y 为图 **G** 的边割 (**Cut**)。特别地, 如果一条边自身是一条边割, 则称其为图 **G** 的桥 (**Bridge**)。

如果一个简单无向图的任意两个顶点间恰有一条边, 则称其为完全图 (**Complete Graph**), n 个顶点的完全图记作 \mathbf{K}_n 。在无向图 **G** 中, 若点子集 $K \subseteq V(\mathbf{G})$ 的导出子图是完全图, 则称 K 为 **G** 的团 (**Clique**)。如果一个简单有向图的任意两个顶点之间有且仅有一条有向边, 那么称它为竞赛图 (**Tournament**),

本文中用 \mathbf{T} 表示竞赛图。

在图 \mathbf{G} 中, 若点子集 $I \subseteq V(\mathbf{G})$ 的导出子图无边 (Edgeless), 即 $E(\mathbf{G}[I]) = \emptyset$, 则称 I 为图 \mathbf{G} 的独立集 (**Independent Set**)。在无向图 \mathbf{G} 中, 如果边子集 $M \subseteq E(\mathbf{G})$ 中的边两两没有公共端点, 则称 M 为 \mathbf{G} 的匹配。如果图 \mathbf{G} 中没有环, 则称 \mathbf{G} 是无环的 (**Acyclic**)。特别地, 在无向图中, 无环的连通图称为树 (**Tree**), 而无环的图称为森林 (**Forest**)。如果图 \mathbf{G} 中的每一条边都和集合 $X \subseteq V(\mathbf{G})$ 有公共顶点, 则称 X 为图 \mathbf{G} 的顶点覆盖 (**Vertex Cover**); 换言之, X 是图 \mathbf{G} 的顶点覆盖当且仅当 $\mathbf{G} - X$ 无边。如果图 \mathbf{G} 中的每一个环都和集合 $F \subseteq V(\mathbf{G})$ 有公共顶点, 则称 F 为图 \mathbf{G} 的反馈点集 (**Feedback Vertex Set**); 换言之, F 是图 \mathbf{G} 的反馈点集当且仅当 $\mathbf{G} - F$ 无环。如果图 \mathbf{G} 中的每一个环都和集合 $F \subseteq E(\mathbf{G})$ 有公共边, 则称 F 为图 \mathbf{G} 的反馈边集 (**Feedback Edge/Arc Set**); 换言之, F 是图 \mathbf{G} 的反馈边集当且仅当 $\mathbf{G} - F$ 无环。

给定点子集或边子集 T , 经过 T 中元素的环称为 T -环 (**T -cycle**), 经过 T 中元素的三角形称为 T -三角形 (**T -triangle**)。如果图 \mathbf{G} 中没有 T -环, 则称 \mathbf{G} 是 T -无环的 (**T -Acyclic**)。如果图 \mathbf{G} 中的每一个 T -环都和集合 $S \subseteq V(\mathbf{G})$ 有公共顶点, 则称 S 为图 \mathbf{G} 的 T -反馈点集 (**T -Feedback Vertex Set**); 换言之, S 是图 \mathbf{G} 的 T -反馈点集当且仅当 $\mathbf{G} - S$ 是 T -无环的。如果图 \mathbf{G} 中的每一个环都和集合 $S \subseteq E(\mathbf{G})$ 有公共边, 则称 S 为图 \mathbf{G} 的 T -反馈边集 (**T -Feedback Edge/Arc Set**); 换言之, S 是图 \mathbf{G} 的 T -反馈边集当且仅当 $\mathbf{G} - S$ 是 T -无环的。

3.2 精确与参数算法

本文主要研究反馈集相关问题的确定性算法设计, 本节将介绍精确算法和参数算法的相关的理论基础。

3.2.1 精确算法

本小节介绍精确算法以及相关复杂度理论的基本概念。

在本文使用的 Landau 记号中, 符号 $\mathcal{O}(\cdot)$ 表示渐近上界 (Asymptotic Upper Bound), 符号 $o(\cdot)$ 表示渐近严格上界 (Asymptotically Strict Upper Bound), 符号 $\Theta(\cdot)$ 表示渐近紧界 (Asymptotically Tight Bound)。例如, 多项式函数可表示为 $n^{\mathcal{O}(1)}$, 亚指数函数可表示为 $2^{o(n)}$ 。此外, 本文将多次使用符号 $\mathcal{O}^*(\cdot)$ 以省略多项式渐近部分: 对于任意函数 $f(n)$, 有

$$\mathcal{O}^*(f(n)) := f(n) \cdot n^{\mathcal{O}(1)}.$$

在计算理论中，判定性问题（**Decision Problem**）是一类答案只有“是”和“否”（“有解”和“无解”）两种可能性的问题。

本文参照了专著 [2, 92] 中对计算问题的简写规范：“变体名称-问题名称(输入类)”。例如，当问题 P 输入限制为类 \mathcal{G} 时，该问题称为 \mathcal{G} 上 P 问题，简写为 $P(\mathcal{G})$ ；问题 P 的名为“Special”的变体问题，称为 Special 版本的 P 问题，简写为 Special- P 。

某一问题的实例（**Instance**）是此问题的一个具体的输入，表示这个问题需要解决的一个具体对象。本文通常使用大写花体字母 \mathcal{I} 表示问题实例。

在复杂度理论中， P 表示多项式时间（**Polynomial-Time**）可判定的判定性问题类，若某问题属于 P 类，本文也称该问题能在多项式时间内求解。 NP 表示非确定性多项式时间（**Non-Deterministic Polynomial-Time**）可判定，或等价地，多项式时间可验证（**Verifiable**）的判定性问题类，若某个问题属于 NP 类，本文也称该问题是 NP 问题。例如，匹配问题^[93]、反馈边集问题（即生成树问题^[94]）等都可在多项式时间内求解，亦可在多项式时间内验证。

定义 3.1 (多项式时间归约) 给定两个判定性问题 P_1 和 P_2 ，若存在算法 \mathcal{R} ，可将问题 P_1 的任何实例 \mathcal{I}_1 作为输入，转换并输出问题 P_2 的实例 \mathcal{I}_2 ，满足以下条件

- 算法 \mathcal{R} 是多项式时间算法；
- \mathcal{I}_1 和 \mathcal{I}_2 等价，即 \mathcal{I}_1 有解当且仅当 \mathcal{I}_2 有解。

则称 \mathcal{R} 是问题 P_1 到问题 P_2 的多项式时间归约（**Polynomial-Time Reduction**），也称问题 P_1 可多项式时间归约（**Polynomial-Time Reducible**）到问题 P_2 。

若所有 NP 问题都可以多项式时间归约到某个问题 P ，则称问题 P 是 NP -难的（**NP-Hard**）。既是 NP -难又是 NP 的判定性问题称为 NP -完全（**NP-Complete**）问题。例如，旅行商问题、顶点覆盖问题、团问题、独立集问题、反馈点集问题等都是 NP -完全问题^[3]。在计算机科学中，精确算法是指确定性地求解某个问题，并在有限时间内给出准确答案的算法。在 $P \neq NP$ 的假设下，任何 NP -难问题及其对应的优化问题均无多项式时间的精确算法。

从基本的定义出发，算法的运行时间是输入长度的函数。在计算复杂性理论中，输入实例的长度定义为通过给定字母表编码实例后的字符串比特数，例如，图的输入长度应为 $\Theta(n + m \log n)$ 。值得强调的是，对于绝大多数自然问题而言，输入实例的长度与我们通常讨论的输入规模并不完全相同。对图论问题而言，传统衡量输入规模的参数是输入图的点数 n 或边数 m 。无论使用点数 n 或是边数 m 作为参数来衡量输入实例的规模，在多项式意义下都是等价的，即，它们的大小能够相互被多项式控制（**Polynomially Bounded**）。但是对于 NP -难问题而言，以不同

参数作为输入实例的规模可能导致时间复杂度呈现较大的差异。在 NP-难图问题的精确算法领域，都遵循既定的传统，以图的点数 n （较少情况使用图的边数 m ）来衡量输入规模。例如，旅行商问题目前最快精确算法复杂度为 $\mathcal{O}(2^n n^2)$ ^[35, 95]，顶点覆盖问题/团问题/独立集问题目前最快精确算法复杂度为 $\mathcal{O}(1.1996^n)$ ^[96]，而反馈点集目前最快精确算法复杂度为 $\mathcal{O}(1.7110^n)$ ^[52, 97]。

3.2.2 参数算法与核心化

精确的指数时间算法并不是唯一涉及困难问题准确解的算法领域。Downey 和 Fellows^[43, 48]在引入参数化复杂性理论时提出了一个精细化分析困难问题的通用算法和复杂性框架。本小节将介绍参数算法以及核心化算法的基本概念。

在参数算法领域，算法复杂度不仅与输入长度有关，而且与问题预设的参数有关，该参数可能不依赖于输入长度。粗略地讲，参数算法领域研究的是：对于 NP-难问题，如何设计相对高效的算法，其时间复杂度的超多项式部分仅是参数的函数，而与输入长度无关；换言之，此算法的时间复杂度中，与输入长度有关的部分可以被多项式控制。特别地，可以认为精确算法是参数算法的特例，图论问题的精确算法可以视为以顶点数为参数的参数算法。但是，从参数复杂性的角度而言，将顶点数作为参数并不具有太大的意义。这是因为在参数算法研究的问题中，参数与输入长度往往没有明确的数量关系。因此，参数算法与精确算法的相关算法设计技术具有一定差异性。

定义 3.2(参数化) 经典判定性问题 P 的参数化 (Parameterization) 定义为将 P 的任一实例 \mathcal{I} 映射到自然数的可计算函数 (Computable Function) $\kappa: \mathcal{I} \mapsto k$ 。问题 P 关于参数化 κ 的参数化问题定义为 $\bar{P} = (P, \kappa)$ ，称 \bar{P} 为 P 以 κ 为参数的参数化问题。参数化问题 \bar{P} 的实例为 $\bar{\mathcal{I}} = (\mathcal{I}, \kappa(\mathcal{I})) = (\mathcal{I}, k)$ 。

在上下文不引起混淆的情况下，本文通常使用 $k = \kappa(\mathcal{I})$ 表示输入实例的参数，使用 n 表示输入实例规模。参数化具有强大的描述能力，在理论层面足够刻画问题的结构特征^[92]。例如，针对特定图类的图论问题可视为原图问题在特定参数设定下的参数化形式。因此，从某种程度上讲，针对特定图类的图论问题的算法研究，实质上是对某一特定参数算法的探究。

定义 3.3(固定参数可解) 给定参数化问题 (P, κ) ，若存在可计算函数 $f(\cdot)$ ，使得对于任意的实例 (\mathcal{I}, k) ，可在 $\mathcal{O}^*(f(k)) = f(k)n^{\mathcal{O}(1)}$ 时间内求解，则称问题 P (在以 κ 为参数下) 是固定参数可解的。

所有固定参数可解的参数化问题构成的全体称作固定参数可解类，记作 FPT，而属于 FPT 类的参数化问题是 FPT 问题。一个经典问题是 P 问题当且仅当在以

$\kappa = \mathcal{O}(1)$ 为参数下是 FPT 问题的。对于众多图问题而言，最常见的参数是解集大小 k ，例如，顶点覆盖和反馈点集问题均为 FPT 的^[39,40]，并且顶点覆盖问题目前最快参数算法的时间复杂度为 $\mathcal{O}^*(1.2529^k)$ ^[98]，而反馈点集问题目前最快参数算法的时间复杂度为 $\mathcal{O}^*(3.46^k)$ ^[52]。

类似于经典复杂性理论，在参数复杂理论中，FPT 被视为是高效的求解算法。同时也可以利用参数化归约（Parameterized Reduction）定义其他参数复杂性。

定义 3.4 (参数化归约) 设 \bar{P}_1 和 \bar{P}_2 为两个参数化问题。若算法 \mathcal{R} 将问题 \bar{P}_1 的实例 (\mathcal{I}_1, k_1) 作为输入，转换并输出问题 \bar{P}_2 的实例 (\mathcal{I}_2, k_2) ，满足以下条件

- 算法 \mathcal{R} 的运行时间为 $\mathcal{O}^*(f(k_1))$ ，其中 $f(\cdot)$ 为可计算函数；
- \mathcal{I}_1 和 \mathcal{I}_2 等价，即 \mathcal{I}_1 有解当且仅当 \mathcal{I}_2 有解；
- $k_2 \leq g(k_1)$ ，其中 $g(\cdot)$ 是可计算函数。

则称 \mathcal{R} 是问题 \bar{P}_1 到问题 \bar{P}_2 的 FPT-归约 (FPT-Reduction)，或问题 \bar{P}_1 可以 FPT-归约到问题 \bar{P}_2 。

若算法 \mathcal{R} 满足

- 算法 \mathcal{R} 是多项式时间算法；
- \mathcal{I}_1 和 \mathcal{I}_2 等价，即 \mathcal{I}_1 有解当且仅当 \mathcal{I}_2 有解；
- $k_2 \leq k_1$ 。

则称 \mathcal{R} 是问题 \bar{P}_1 到问题 \bar{P}_2 的保持参数（不增）的多项式时间归约，或 \bar{P}_1 可以多项式时间地归约到问题 \bar{P}_2 ，且保持参数（不增）。

利用 FPT-归约，可以定义常用的参数复杂性类。使用复杂度类 $W[d, t]$ 表示能够 FPT-归约到（以权重为参数的）Weighted-SAT(d -Depth t -Weft CIRC) 问题的参数化问题类。定义复杂性类

$$W[t] = \bigcup_{d \in \mathbb{N}} W[d, t].$$

使用复杂度类 $W[\text{SAT}]$ 和 $W[\text{P}]$ 分别表示能够 FPT-归约到 Weighted-SAT(PROP) 问题和 Weighted-SAT(CIRC) 问题的参数化问题类（问题定义见附录 A）。则有包含关系

$$\text{FPT} = W[0] \subseteq W[1] \subseteq \dots \subseteq W[t] \subseteq \dots \subseteq W[\text{SAT}] \subseteq W[\text{P}].$$

$\text{FPT} \neq W[1]$ 蕴含了 $\text{P} \neq \text{NP}$ ^[41,42]。若所有 $W[t]$ 中的问题都可以 FPT-归约到某个参数化问题，则称此问题是 $W[t]$ 难的 ($W[t]$ -Hard)。既是 $W[t]$ -难又属于 $W[t]$ 的参数化问题称为 $W[t]$ -完全的 ($W[t]$ -Complete)。例如，独立集问题和团问题均为 $W[1]$ -难的^[39,40]，碰撞集问题和集合覆盖问题是 $W[2]$ -难的^[39,99]。在 $\text{FPT} \neq W[1]$

的假设下, $\mathsf{W}[1]$ -难的问题没有 FPT 算法, 即不存在时间复杂度形如 $\mathcal{O}^*(f(k))$ 的参数算法。

相比 FPT-归约, 保持参数的多项式时间归约的定义更强, 因此能够更准确地刻画参数化问题之间的关系。特别地, 若存在问题 \bar{P}_1 到 \bar{P}_2 的保持参数的多项式时间归约, \bar{P}_2 可在 $\mathcal{O}^*(f(k_2))$ 时间内求解蕴含着 \bar{P}_1 可在 $\mathcal{O}^*(f(k_1))$ 时间内求解。

如果一个参数化问题 \bar{P} 能够在多项式时间内归约到该问题自身, 并且保证输出实例的参数和长度均能够被输入实例的参数控制, 那么这种归约称 \bar{P} 的核心化 (Kernelization)。核心化是一种研究多项式时间预处理算法的系统性方法, 也是设计参数化算法的重要工具。预处理或数据压缩 (Data Reduction) 普遍应用于实际的计算机算法实现中。在处理 NP-难问题时, 预处理算法可以高效地解决问题实例中“容易的部分”, 并将实例的简化到仅剩下“困难的核心部分”。换言之, 该方法的思想是在多项式时间内压缩 (但不一定解决) 实例, 从而得到规模更小且等价的实例, 称为核 (Kernel)。必要时, 只需调用精确算法的对核进行求解便能够判定原问题实例。

如何衡量预处理算法的压缩能力是需要首先明确的。事实上, 对于 NP-难问题, 如果存在多项式算法, 可以将任意实例的长度压缩至少一个比特, 则意味着 $\mathsf{P} = \mathsf{NP}$ 。因此, 在很长一段时间里, 无法对预处理这一重要概念给出严格且有价值的刻画。这导致多项式时间预处理方法缺乏理论支撑, 从而无法进行有效的理论分析。但是在参数复杂性框架下, 使用参数的某一函数控制实例规模能够完美地填补相关理论空白。至此, 被称为核心化算法的数据压缩技术成为了参数算法领域重要的研究对象。现给出核心化的严格定义。

定义 3.5 (核心化) 给定参数化问题 (P, κ) , 若存在算法 \mathcal{K} , 可以将问题 (P, κ) 的任意实例 (\mathcal{I}, k) 作为输入, 转换并输出问题 (P, κ) 的另一个实例 (\mathcal{I}', k') , 满足以下条件

- 算法 \mathcal{K} 是多项式时间算法;
- 存在可计算函数 $f(\cdot)$, 使得参数 $k' \leq f(k)$;
- 存在可计算函数 $g(\cdot)$, 使得输出实例的长度 $n' \leq g(k)$ 。

则称算法 \mathcal{K} 为问题 P (关于参数化 κ) 的核心化算法, \mathcal{I}' 为问题 P (关于参数化 κ) 的核, 核的大小为 $g(\cdot)$ 。

一个参数化问题是 FPT 的当且仅当其存在核心化算法^[2, 92]。

此外, 核的大小刻画了核心化算法的压缩能力。对于问题 P 及其大小为 $g(k)$ 的 (关于参数化 κ 的) 核, 如果 $g(k) = k^{\mathcal{O}(1)}$, 则称问题 P 存在多项式核 (**Polynomial Kernel**); 如果 $g(k) = \mathcal{O}(k)$, 则称问题 P 存在线性核 (**Linear Kernel**)。

例如，顶点覆盖问题有点集大小不超过 $2k$ 的线性核^[100,101]，反馈集点问题有点集大小不超过 $2k^2 + k$ 的平方核^[102]。

另一方面，核心化领域也存在一些关于下界的研究方向，目前这些下界结果均以比特数来衡量核的大小。例如，在 $\text{NP} \subsetneq \text{coNP/poly}$ 假设下，对于任意小的正实数 $\varepsilon \in \mathbb{R}^+$ ，顶点覆盖问题和反馈点集问题均不存在比特数为 $\mathcal{O}(k^{2-\varepsilon})$ 的核^[103]。

3.2.3 精确算法与参数算法的联系

本小节将介绍精确算法与参数算法的联系，旨在阐明参数算法可以作为设计精确算法的强大的工具。事实上，许多精确算法的设计都建立在一个或多个参数算法的基础之上。

特别是 2016 年 Fomin 等人^[97,104]提出的基于单调局部搜索的精确算法框架。在该框架中，如果给定一个单指数时间的参数算法，可以获得一个非平凡单指数时间的精确算法，这为当时大量 NP-难题的精确算法带来了显著的改进。直至今日，相当大一部分 NP-难题的最快精确算法记录依然是由此框架下发展出的精确算法所保持。

碰撞集问题是适用于单调局部搜索 (Monotone Local Search) 框架的一个十分重要且典型的例子^[3]。考虑一个集合系统 (U, \mathcal{F}) ，其中 \mathcal{F} 为集合 U 上的子集族。若子集 $S \subseteq U$ 与集族 \mathcal{F} 的每一个子集合均有公共元素，则称其为此集合系统 (U, \mathcal{F}) 的碰撞集。设 Π 是一个关于集族的具有遗传性 (Hereditary) 的性质，这里遗传性是指：若 \mathcal{F} 满足性质 Π ，则对任意子集 $W \subseteq U$ ，集族 $\{X \in \mathcal{F} : X \subseteq W\}$ 也满足性质 Π 。则可定义 Π 上碰撞集问题。

Π 上碰撞集问题 (Hitting Set (Π))

输入：一个元素集合 U ，一个子集族 $\mathcal{F} \subseteq 2^U$ 和一个自然数 $k \in \mathbb{N}$ ，其中 \mathcal{F} 满足性质 Π 。

输出：判断是否存在一个大小不超过 k 的碰撞集。

若性质 Π 为 “ U 的任意子集”，即不对 \mathcal{F} 加以任何限制时，Hitting Set (Π) 问题简记为 Hitting Set 问题。Hitting Set 问题是一个高度难解的组合优化问题，当以解集大小 k 为参数时，精确求解此问题甚至获得常数近似解也是 W[2]-难的^[39,99]。目前，Hitting Set 问题只存在平凡的计算复杂度为 $\mathcal{O}^*(2^n)$ 的精确算法。而在 SETH 下，对任意正实数 $\varepsilon > 0$ ，不存在运行时间为 $\mathcal{O}((2 - \varepsilon)^n)$ 的精确算法^[33]。

当性质 Π 是 “ \mathcal{F} 中的子集的大小不超过 d ” 时，称这类问题为 d -碰撞集问题。

***d*-碰撞集问题 (*d*-Hitting Set)**

输入: 一个元素集合 U , 一个子集族 $\mathcal{F} \subseteq 2^U$ 和一个自然数 $k \in \mathbb{N}$, 其中 \mathcal{F} 中的任意集合最多包含 d 个元素。

输出: 判断是否存在一个大小不超过 k 的碰撞集。

当 d 取 2 时, 2-碰撞集问题等价于经典的顶点覆盖问题。对于给定的 $d \geq 2$, d -碰撞集问题均未 FPT 的, 并且假定能在 $\mathcal{O}^*(\xi_d^k)$ 的时间内求解。目前已经证明, $\xi_2 \leq 1.2529^{[98]}$ 、 $\xi_3 \leq 2.0755^{[54]}$ 、 $\xi_4 \leq 3.0755^{[61, 62]}$ 以及 $\xi_5 \leq 4.0755^{[61, 62]}$, 且当 $d \geq 6$ 时, ξ_d 不超过三次方程

$$x^3 - dx^2 + (d-1)x - 1 = 0$$

的最大实根^[53], 进而可得渐近性 $\xi_d \leq d - 1 + \mathcal{O}(d^{-2})$ 。

定理 3.6 (单调局部搜索 [97, 104]) 设 Π 是集合系统上一个具有遗传性 (Hereditary) 的集族。若 Hitting Set (Π) 可在 $\mathcal{O}^*(\alpha^k)$ 时间, 则 Hitting Set (Π) 可在 $(2 - \alpha^{-1})^n \cdot 2^{o(n)}$ 时间内求解。

定理3.6表明, 若某问题可以用 Hitting Set (Π) 问题进行描述, 那么只要存在单指数时间的参数算法, 都能够基于单调局部搜索框架得到非平凡的精确算法。并且在某一算法范式下, 已达理论最优^[126]。例如, 对于任意给定的正整数 d , 都可以设计出 d -Hitting Set 打破 2^n 的精确算法。具体地, 若 d -Hitting Set 可以在 $\mathcal{O}^*(\xi_d^k)$ 时间内求解, 则也可以在 $\varsigma_d^n \cdot 2^{o(n)}$ 时间内求解, 其中 $\varsigma_d = 2 - \xi_d^{-1} = 2 - (d-1)^{-1} + \mathcal{O}(d^{-4})$ (参见表3-1)。

表 3-1 d -碰撞集的参数与精确算法复杂度

d 值	3	4	5	6	7	8	9	10	100
$\xi_d \leq$	2.0755	3.0755	4.0755	5.0490	6.0330	7.0237	8.0178	9.0139	99.0002
$\varsigma_d \leq$	1.5182	1.6749	1.7547	1.8020	1.8343	1.8577	1.8753	1.8891	1.9899

特别地, 若对于任何满足性质 Π 的集族 \mathcal{F} 通过隐式子的方式表示, 则称 Hitting Set (Π) 问题是隐式碰撞集问题。例如, 可以将 \mathcal{F} 用图结构进行编码, 并且可通过图结构在多项式时间内判定一个集合是否属于 \mathcal{F} , 那么对应的图问题就属于隐式碰撞集问题。设 Π 是具有遗传性的图族, 即若 $\mathbf{G} \in \Pi$, 则 \mathbf{G} 的所有导出子图都包含在 Π 中。对于任意的 Π , 都存在关于 Π 的顶点删除问题 (Vertex Deletion Problem), 定义如下:

***\Pi* 上顶点删除问题 (Vertex Deletion (Π))**

输入: 一个无向或有向图 \mathbf{G} 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在一个大小不超过 k 的点集 $S \subseteq V(\mathbf{G})$, 使得 $\mathbf{G} - S \in \Pi$ 。

Vertex Deletion (Π) 涵盖了大量经典的图论问题，并且所有 Π -顶点删除问题均属于隐式碰撞集问题，故也适用于单调局部搜索框架。例如，当 Π 取所有无边图时，Vertex Deletion (Π) 为顶点覆盖问题；当 Π 取所有森林时，Vertex Deletion (Π) 为反馈点集问题。

除此之外，基于多个不同参数算法之间的均摊 (Trade-off) 思想，也能够设计出效果良好的精确算法。这种均摊思想在众多问题的精确算法设计中得到了广泛应用^[58, 105]。

定理 3.7 (多个参数算法的均摊) 设 k 为可计算问题 P 的一个参数，并且在 P 的任意实例中，都满足 $k \leq n$ 。假设 P 可以分别在 $\mathcal{O}^*(\alpha^k)$ 和 $\mathcal{O}^*(\beta^{n-k})$ 的时间内求解，那么存在时间复杂度为 $\mathcal{O}^*(2^{\gamma n})$ 的精确算法求解问题 P ，其中 $\alpha, \beta > 1$ 且均为常数， $\gamma = \log \alpha \log \beta / \log \alpha \beta$ 。

3.3 精确与参数算法设计技术

本节详述了本文所使用的精确和参数算法设计技术。当前，解决许多基础算法问题的高效算法均基于本节所讨论的技术范畴^①。

3.3.1 皇冠分解技术

皇冠分解 (Crown Decomposition) 是一种基于匹配理论的通用核心化与参数算法设计技术。该技术发展于以 König 和 Hall 的奠基性工作^[106–108]。

给定无向图 \mathbf{G} 不相交的点集 U 和 W ，如果匹配 $M \subseteq E(\mathbf{G})$ 覆盖了 U 中的每一个顶点，且 M 中每一条边恰好连接了 U 和 W 中的各一个顶点，则称 M 是 U 到 W 的匹配。

定义 3.8 (皇冠分解) 无向图 \mathbf{G} 的皇冠分解是满足以下条件的点集 $V(\mathbf{G})$ 的三划分 (C, H, R) 。

- 点集 C 是非空独立集；
- 点集 C 和 R 之间没有边相连，即点集 H 割开了 C 和 R ；
- 图 \mathbf{G} 中存在连接 H 到 C 的匹配。

皇冠分解可能不是唯一的，也可能不存在，如三角形 \mathbf{C}_3 不存在皇冠分解。但可利用 Hopcroft-Karp 算法^[109]确保在一定条件下总能够快速找到一个皇冠分解。

定理 3.9 (Hopcroft-Karp 算法) 设 \mathbf{G} 为二部图，其中 A 和 B 分别为点集 $V(\mathbf{G})$ 两个独立集部分。可以在 $\mathcal{O}(m\sqrt{n})$ 时间内得到 \mathbf{G} 的一个最大匹配和一个最小顶点覆盖。进一步，在 $\mathcal{O}(m\sqrt{n})$ 时间内，要么找到一个 A 到 B 的匹配，要么找到子

^① 除了动态规划算法，本文涉及的所有算法设计技术均能够保证在多项式空间完成，因此在无特殊说明下，本文提出的算法的空间复杂度均为多项式的。

集 $A' \subseteq A$ 满足 $|N(A')| < |A|$ 成立。

引理 3.10 (皇冠引理) 设 \mathbf{G} 是不包含孤立点的无向图。若图 \mathbf{G} 的顶点数 $n \geq 3k + 1$, 则存在多项式时间算法, 要么找到一个大小为 $k + 1$ 的匹配; 要么找到 \mathbf{G} 的一个皇冠分解。

皇冠分解有诸多变体, 如平衡皇冠分解^[110]。基于匹配的图分解也可视为皇冠分解的变体, 如 Dulmage-Mendelsohn 分解^[111, 112]和 Gallai-Edmonds 分解^[93, 113, 114]。在本文第七章中, 将使用 Dulmage-Mendelsohn 分解设计参数算法。

3.3.2 树分解技术

树分解技术在现代图论与组合优化领域占据着至关重要的地位。此技术是一种能够将任意无向图映射到树的图结构分解方法, 旨在刻画一个图与树结构的相似度, 并能为许多图论计算问题的高效算法设计提供支持。

定义 3.11 (树分解) 无向图 \mathbf{G} 的树分解 (**Tree Decomposition**) 是一个满足以下条件的二元组 $(\mathcal{T}, \{B_t\}_{t \in V(\mathcal{T})})$, 其中 \mathcal{T} 是一颗树, 并且 \mathcal{T} 的每一个顶点 t 都对应一个点集 $B_t \subseteq V(\mathbf{G})$, 称作包 (**Bag**)。

- 图 \mathbf{G} 中的每一个顶点都至少在一个包中, 即

$$V(\mathbf{G}) = \bigcup_{t \in V(\mathcal{T})} B_t;$$

- 图 \mathbf{G} 中的每一条边都至少在一个包的导出子图中, 即

$$E(\mathbf{G}) = \bigcup_{t \in V(\mathcal{T})} E(G[B_t]);$$

- 令 $X_v = \{t \in V(\mathcal{T}) : v \in B_t\}$ 表示所有包含了顶点 v 的包所对应的树上的节点集。那么图 \mathbf{G} 中任何顶点 v 所对应的集合 X_v 导出的子图都是一颗树, 即 $\mathcal{T}[X_v]$ 是连通的。

直观上讲, 树分解将给定图 \mathbf{G} 的顶点表示为一棵树的子树, 使得图 \mathbf{G} 中任意两个相邻顶点所对应的子树交集非空, 从而使得图 \mathbf{G} 成为了这棵树上子树的相交图 (一种弦图) 的一个子图。基于树分解, 可以通过树宽 (Treewidth) 表征一个图与树结构的相似程度。

定义 3.12 (树宽) 设无向图 \mathbf{G} 的一个树分解为 $(\mathcal{T}, \{B_t\}_{t \in V(\mathcal{T})})$ 。树分解的宽度, 定义为

$$\max_{t \in V(\mathcal{T})} |B_t| - 1.$$

无向图 \mathbf{G} 的树宽, 是指 \mathbf{G} 的所有可能树分解中树宽度的最小值, 记作 $\text{tw} = \text{tw}(\mathbf{G})$ 。

特别地, 连通图 \mathbf{G} 的树宽为 1 当且仅当 \mathbf{G} 是一颗树。计算一般图的树宽是 NP-难的^[115], 目前可以在 $\mathcal{O}(2^{\text{tw}^2} n^4)$ 时间内精确求解树宽^[116], 或者在 $\mathcal{O}(2^{\text{tw}/\varepsilon} n^4)$ 时间内得到近似率为 $1 + \varepsilon$ 的近似解^[116]。在弦图等一些特殊图上, 能够在多项式时间内进行树分解并精确计算树宽。事实上, 一些高效的实验算法正是通过弦图计算^[117]任意图的树宽。

当树宽有界时, 可以使用树分解上的标准动态规划算法求解许多图论问题, 著名的 Courcelle 定理^[118]就是一个经典的例子。粗略地讲, Courcelle 定理指出, 可以被一元二阶逻辑 (Monadic Second-Order) 描述的图问题以树宽为参数是 FPT 的。在本文第八章中, 将使用树分解设计参数算法和精确算法。

3.3.3 分支搜索技术与度量治之

分支搜索技术, 也被称为分支约简 (Branch and Reduce) 技术, 是一个最为通用的精确算法与参数算法设计技术。许多首次突破 2^n 或 2^k 计算复杂度瓶颈的非凡算法均采用了分支搜索技术。分支搜索技术一般通过递归算法形式进行描述, 算法中包括一些约简规则 (**Reduction Rule**) 和分支规则 (**Branching Rule**)。每一个规则都会对实例进行一些操作, 然后进行递归调用。为了估计每步规则对求解问题的贡献, 需要为实例选取一个度量 $\mu(\cdot)$ 。在算法的每一个步骤中, 需要保证度量不增, 并且度量为非正值的实例可在多项式时间内解决。

约简规则将当前实例转换为一个“更小”的实例。若约简规则保证输入实例有解当且仅当输出实例有解, 则称该约简规则是正确的。分支规则将当前实例 \mathcal{I} 分为多个子实例 $\mathcal{I}_1, \mathcal{I}_2, \dots, \mathcal{I}_r$, ($i \in [r]$) 并且满足 $\mu(\mathcal{I}_i) < \mu(\mathcal{I})$ 。若实例 \mathcal{I} 有解当且仅当至少一个子实例有解, 则称该分支规则是正确的。

为了分析运行时间, 需要采用度量治之分析手段控制搜索树的大小。用 $\mathcal{T}(\mu)$ 表示算法关于度量 μ 的搜索树的大小上界。具体地, 为每一个分支规则建立如下形式的递归关系式

$$\mathcal{T}(\mu) \leq 1 + \sum_{i \in [r]} \mathcal{T}(\mu_i),$$

其中 $\mu_i = \mu(\mathcal{I}_i)$ ($i \in [r]$)。该递归关系可使用分支向量 (**Branching Vector**)

$$(\mu - \mu_1, \mu - \mu_2, \dots, \mu - \mu_r)$$

表示。方程 $x^\mu - \sum_{i=1}^r x^{\mu_i} = 0$ 的唯一正实根称为该分支规则的分支因子 (**Branching Factor**)。设 $\mathbf{a} = (a_1, a_2, \dots, a_p)$ 和 $\mathbf{b} = (b_1, b_2, \dots, b_q)$ 为两个分支向量。若 $p \leq q$ 并且 $a_i \geq b_i$ ($i \in [p]$) 成立, 则称 \mathbf{a} 不劣于 (**not worse than**) \mathbf{b} 。一个简单的性质是:

如果分支向量 \mathbf{a} 不劣于 \mathbf{b} ，那么 \mathbf{a} 的分支因子不大于 \mathbf{b} 的分支因子。在基于分支搜索技术的算法中，如果所有分支向量中的最大分支因子是 α ，那么算法的搜索树大小能够被 $\mathcal{O}^*(\alpha^\mu)$ 所控制。在本文后续章节中，将充分使用分支搜索技术联合其他相关技术设计参数算法与精确算法。

3.3.4 迭代压缩技术

迭代压缩技术，由 Reed 等人^[119]在 2004 年被首次提出，标志着参数算法领域的一个重大进步。该技术问世后，包括有向反馈点集问题在内的多个当时被认为难以攻克的问题均相继获得了首个 FPT 时间的算法，部分问题甚至能在单指数时间内求解^[67, 120–125]。

迭代压缩技术是一个适用于遗传性质 Π 上顶点删除问题的参数算法框架。回顾定义，这类问题旨在给定图中找出一个大小不超过 k 的点集，使得将其移除后的图满足某个全局性质 Π 。当以解集大小 k 为参数时，可以通过反复求解一个压缩版变体问题，从而得到原问题的最优解。

首先，定义 Vertex Deletion (Π) 问题的压缩版本 (Compression Version) 和分离版本 (Disjoint Version)，分别记作 Compression-Vertex Deletion (Π) 和 Disjoint-Vertex Deletion (Π)。

压缩版本 Π 上顶点删除问题 (Compression-Vertex Deletion (Π))

输入: 一个无向或有向图 \mathbf{G} ，一个点集 \bar{S} 使得 $\mathbf{G} - \bar{S} \in \Pi$ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在一个大小不超过 k 的点集 $S \subseteq V(\mathbf{G})$ ，使得 $\mathbf{G} - S \in \Pi$ 。

分离版本 Π 上顶点删除问题 (Disjoint-Vertex Deletion (Π)))

输入: 一个无向或有向图 \mathbf{G} ，一个不可删点集 U 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在一个大小不超过 k 的点集 $S \subseteq V(\mathbf{G}) \setminus U$ ，使得 $\mathbf{G} - S \in \Pi$ 。

接下来阐述如何使用压缩版本和分离版本构建算法框架求解 Vertex Deletion (Π) 问题的实例 $\mathcal{I} = (\mathbf{G}, k)$ (参见图3-1)。对图 \mathbf{G} 中的顶点编号 $V(\mathbf{G}) = \{v_1, v_2, \dots, v_n\}$ 。令 $V_i = \{v_1, v_2, \dots, v_i\}$ 以及 $\mathbf{G}_i = \mathbf{G}[V_i]$ 。然后，通过遍历 $i = 1, 2, \dots, n$ 并反复计算子实例 $\mathcal{I}_i = (\mathbf{G}_i, k)$ 的解。注意到， \mathbf{G}_i 作为图 \mathbf{G} 的导出子图，如果实例 \mathcal{I}_i 无解，则表明实例 \mathcal{I} 无解。对于 \mathcal{I}_1 ，因为顶点个数是常数，所以可以在常数时间内求解实例 \mathcal{I}_1 得到解集 S_1 。对于实例 \mathcal{I}_i ($i \in [n-1]$)，设 S_i 为 \mathcal{I}_i 的解，令 $\bar{S}_i = S_i \cup \{v_{i+1}\}$ 。这意味着 $(\mathbf{G}_{i+1}, \bar{S}_{i+1}, k)$ 是 Compression-Vertex Deletion (Π) 问题的实例，且满足 $|\bar{S}_{i+1}| \leq k+1$ 。根据定义，实例 $(\mathbf{G}_{i+1}, \bar{S}_{i+1}, k)$ 的解也是实例 \mathcal{I}_{i+1} 的解。因此，只需要迭代地多次求解压缩版本，便能够最终实现对原问题的求解，从而得到了压缩引理。

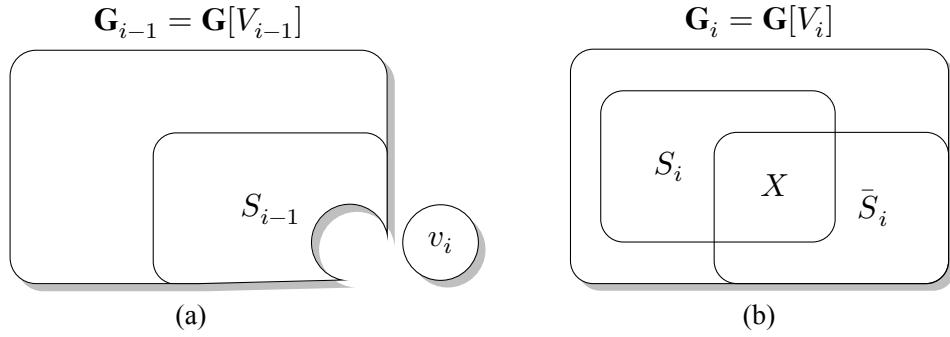


图 3-1 迭代压缩技术框架示意图。(a) 原问题的子实例示意图 $\mathcal{I}_i = (\mathbf{G}_{i-1}, k)$ 。

其中图 $\mathbf{G}_{i-1} = \mathbf{G}[V_{i-1}]$ 且 S_{i-1} 是 \mathcal{I}_i 的一个解。(b) 压缩版本的实例示意图 $(\mathbf{G}_i, \bar{S}_i, k)$ 。其中图 $\mathbf{G}_i = \mathbf{G}[V_i]$, $\bar{S}_i = S_{i-1} \cup \{v_i\}$ 是一个大小为 $k+1$ 的“近似解”， S_i 是一个精确解满足 $X = S_i \cap S'_i$ 。

引理 3.13 (压缩引理) 集合 \bar{S} 大小不超过 $k+1$ 时，若压缩版本能在 $\mathcal{T}(k, n)$ 时间内求解，则原问题能在 $\mathcal{O}(n) \cdot \mathcal{T}(k, n)$ 时间内求解。

接下来考虑压缩版本的求解。考虑 Compression-Vertex Deletion (Π) 问题的实例 (\mathbf{G}, \bar{S}, k) ，其中 $|\bar{S}| \leq k+1$ 。当 $|\bar{S}| \leq k$ 时， \bar{S} 本身就是一个解，因此不妨设 $|\bar{S}| = k+1$ 。枚举解集与 \bar{S} 的交集（共有 2^{k+1} 种情况），对于每一个交集 $X \subseteq \bar{S}$ ，将 X 删除， \bar{S} 中剩余顶点均不可放入解集中。设 $U = \bar{S} \setminus X$ ，那么 $\mathcal{I}_U = (\mathbf{G}', U, k')$ 是 Disjoint-Vertex Deletion (Π) 问题的实例，其中 $\mathbf{G}' = \mathbf{G} - X$, $k' = k - |X|$ 且 $n' = n - |X|$ 。在实例 \mathcal{I}_U 中， U 的一个大小为 $k'+1$ 且 $\mathbf{G}' - U \in \Pi$ 。因此，只需要多次求解分离版本的实例，便能够完成对压缩版本的求解，进而求解原问题。

定理 3.14 (迭代压缩) 不可删点集 U 的大小不超过 $k+1$ 且 $\mathbf{G} - U \in \Pi$ 时，如果分离版本能在 $\mathcal{T}(k, n)$ 时间内求解，那么求解原问题的运行时间上界为

$$\sum_{i=0}^k \binom{k+1}{i} \cdot \mathcal{T}(k, n) \cdot n^{\mathcal{O}(1)}.$$

特别地，若 $\mathcal{T}(k, n) = \alpha^k n^{\mathcal{O}(1)}$ ，那么原问题能在 $(\alpha+1)^k n^{\mathcal{O}(1)}$ 时间内求解。

迭代压缩的强大之处在于，一旦满足特定条件的分离版本被证明是 FPT 的，则能直接说明原问题是 FPT 的。然而，为分离版本设计参数算法相比原问题要容易许多。进一步地，若分离版本能在单指数时间 $\mathcal{O}^*(\alpha^k)$ 内求解，那么原问题也可在单指数时间内求解，且时间复杂度仅增加至 $\mathcal{O}^*((\alpha+1)^k)$ 。另一方面，在迭代压缩框架中，由于没有引入相关问题的其他结构信息，仅通过穷举解集与 \bar{S} 的交，标准的迭代压缩算法无法突破 2^k 的时间复杂度瓶颈。本文第六章将结合迭代压缩和其他相关技术设计参数算法突破 2^k 。

3.3.5 平衡割技术

平衡割分为平衡点割 (Balanced Separator) 和平衡边割 (Balanced Cut)，是设计分治算法的重要且基础的工具。平衡割能够将一个图的顶点或边均匀地分割开，同时自身的大小能被控制住。存在多种不同版本的平衡割技术，它们的大小能够被不同的参数所控制，本文仅介绍使用最大度控制平衡边割的结论。

定理 3.15 (平衡边割 [75]) 设 \mathbf{G} 为一个没有自环和孤立点的多重无向图，设其最大度为 $\Delta(\mathbf{G}) = d$ ，存在点集的二划分 (P_0, P_1) 满足

$$\begin{cases} \frac{m}{4} - \frac{d}{2} \leq |E(\mathbf{G}[P_0])| \leq \frac{m}{4} + \frac{d}{2}; \\ \frac{m}{4} - \frac{d}{2} \leq |E(\mathbf{G}[P_1])| \leq \frac{m}{4} + \frac{d}{2}; \\ \frac{m}{2} - d \leq |E(P_0, P_1)| \leq \frac{m}{2} + d, \end{cases}$$

其中 $E(P_0, P_1)$ 表示端点分别在点集 P_0 和 P_1 的边集。此外，划分 (P_0, P_1) 可以在多项式时间内计算得到。

上述定理表明，当一个无向图的最大度有界时，能够高效地找到足够小的平衡边割，在此基础上则可以考虑采用分治思想设计高效算法。本文第六章使用了基于平衡边割的分治思想设计参数算法。

3.4 本章小结

本章第一部分主要介绍了图论的基础概念与符号体系，以及经典计算理论和参数复杂性理论的基本知识。作为计算复杂性理论中相对年轻的分支，参数复杂性理论使得一些在经典复杂性理论中的难解问题可以在参数化的视角下得到相对有效地解决。因此，参数复杂性理论不仅为 NP-难问题提供了新的可解性框架，也为精确算法设计开辟了新的技术路线。

本章第二部分对全文涉及的精确与参数算法的基础性技术框架进行了详细阐述。这些技术均为领域内成熟且强大算法工具，在大量图论或组合优化问题的算法设计中展现出广泛的适应能力。在后续第六章至第八章中，本文将针对特定的反馈集问题，结合多种技术设计，提出快速的精确和参数算法。此外，本文也将创新性地提出针对性的技术，这些技术并未在本章列出，而将在后续各章中逐一展开讨论。

第四章 反馈集相关问题梳理与研究进展分析

本章首先在4.1节介绍反馈集问题和子集反馈集问题的定义，然后归纳了各类版本间的归约关系，最后在4.2节综述了这两类问题在各图类上的计算复杂性和算法的研究进展。

4.1 反馈集与子集反馈集问题

反馈集问题是计算机科学中研究最为广泛和深入的图上 NP-完全问题之一，而子集反馈集问题是反馈集问题更一般化的形式，更加具备普适性。这两个问题均属于隐式碰撞集问题，而隐式碰撞集问题是参数与精确算法的核心研究对象之一，长期以来受到广泛的关注。近年，关于反馈集和子集反馈集问题在计算复杂性上的分类工作已逐步完善，在算法领域中也已出现许多关键的突破性工作。本文的主要工作集中在反馈集和子集反馈集问题的计算复杂性和确定性算法研究方面，本节将给出这两个问题的严格定义和相关理论结果。

4.1.1 反馈集问题

回顾3.1节给出的反馈点集和反馈边集的定义。图 \mathbf{G} 的反馈点集 $S \subseteq V(\mathbf{G})$ (或反馈边集 $S \subseteq E(\mathbf{G})$) 是指的图 \mathbf{G} 的点子集 (或边子集)，在删除 S 以后，剩余的图不含有任何环。等价地，点集 (或边集) S 是图 \mathbf{G} 的反馈点集 (或边集) 当且仅当对 \mathbf{G} 中的任意环都经过了 S 中的顶点 (或边)。反馈集问题旨在找到不超过给定大小 k 的反馈集。反馈集的输入实例可表示为 $\mathcal{I} = (\mathbf{G}, k)$ ，输入图 \mathbf{G} 可以是无向图或有向图。若图 \mathbf{G} 中存在大小不超过 k 的反馈集 S ，则称实例 \mathcal{I} 有解且 S 为 \mathcal{I} 的一个解；否则称实例 \mathcal{I} 无解。图 \mathbf{G} 的最小反馈集称为实例 \mathcal{I} 的最小解 (**Minimum Solution**) 或最优解 (**Optimal Solution**)。根据图 \mathbf{G} 是有向还是无向图，以及反馈集是边集还是点集，可以定义以下四种不同版本的问题：

- (无向) 反馈点集问题 (Feedback Vertex Set Problem)，简称 FVS 问题；
- (无向) 反馈边集问题 (Feedback Edge Set Problem)，简称 FES 问题；
- 有向反馈点集问题 (Directed Feedback Vertex Set Problem)，简称 DFVS 问题；
- 有向反馈边集问题 (Directed Feedback Arc Set Problem)，简称 DFAS 问题。

见表4-1对四种问题的归类。

FES 问题等价于生成树 (Spanning Tree) 问题，因此 FES 问题能够在多项式时间内求解。其他三种版本的问题都是 NP-完全的，有向图上的两种反馈集问题

表 4-1 反馈集问题类别

图类型	反馈集	问题名	简称
无向	点集	反馈点集问题	FVS
无向	边集	反馈边集问题	FES
有向	点集	有向反馈点集问题	DFVS
有向	边集	有向反馈边集问题	DFAS

更是在 1972 年 Karp 的 21 个经典的 NP-完全问题之列^[3]。

FVS 问题和 DFVS 问题的 NP-难性基于顶点覆盖问题的归约^[3]；DFAS 问题的 NP-难性基于 DFVS 问题到 DFAS 问题的归约，此归约由 Lawler 和 Karp 提出^[3]。

4.1.2 反馈集问题的求解算法

反馈集问题早期的算法研究集中在近似算法领域。1981 年，Monien 和 Schulz^[127]提出了 FVS 问题的 $\sqrt{\log n}$ 倍近似算法。随后，Bar-Yehuda^[22, 26]给出了 4 倍近似算法，首次证明了 FVS 问题可以常数倍近似。目前，最优的近似算法由 20 世纪 90 年代的三项独立的工作给出^[128-130]。这三项研究均提出了 FVS 问题的 2 倍近似算法，且核心思想类似。具体地，Becker 和 Geiger^[128]提出了基于组合的贪心算法；Bafna 等人^[129]给出的算法的近似率为 $2 - 2/(3\Delta - 2)$ 且分析更加简洁；而 Chudak 等人则是^[130]给出了基于线性规划的 2 倍近似算法。

20 世纪在 90 年代，Bodlaender^[37, 38]提出了 FVS 问题在解集大小 k 为常数时的线性时间算法。同一时期，Downey 和 Fellows 初步建立参数复杂性理论，FVS 问题作为典型的 NP-难问题，被证明是 FPT 的^[41]。至此之后，FVS 问题参数算法的研究逐步深入，算法时间复杂度也持续改进^[43, 131-133]。在 2005 年和 2006 年，Dehne 等人^[134]和 Guo 等人^[135]分别利用了当时面世不久的迭代压缩技术，独立地提出了 FVS 问题的单指数参数算法。

FVS 问题的单指数参数算法的时间复杂度在本世纪得到了进一步研究，并经历了多轮改进^[52, 136-140]。目前最快的参数算法是由 Iwata 和 Kobayashi^[51, 52]在 2019 年提出，其运行时间达到了 $\mathcal{O}^*(3.46^k)$ 。该算法在 Cao^[141]提出的最大度分支算法（High-Degree Branching）的基础上进行了改进，同时引入了度量治之分析手段精细地分析时间复杂度上界。表 4-2 展示了 FVS 问题的参数算法的研究历程。

在核心化方面，Burrage 等^[143]首次在 2006 年为 FVS 问题得到了顶点数为 $\mathcal{O}(k^{11})$ 的多项式核。随后，Bodlaender 和 van Dijk^[144]将 FVS 问题的核大小改进到了立方级。直至目前，Thomassé^[145, 146]和 Iwata 等人^[102]分别给出了顶点数为 $4k^2$ 和 $2k^2 + k$ 的平方级大小的核。值的强调的是，Iwata 等人提出的核心化算法具有线性时间复杂度，从而使得 FVS 问题可在关于顶点数 n 线性的时间内求解。

表 4-2 反馈集问题参数算法研究历程

作者	参数算法	发表年份
Bodlaender ^[37,38]	$\mathcal{O}^*((17(k^4)!)$	1991
Downey 和 Fellows ^[41]	$\mathcal{O}^*((17(k^4)!)$	1992
Downey 和 Fellows ^[43]	$\mathcal{O}^*((2k+1)^k)$	1999
Raman 等 ^[131]	$\mathcal{O}^*((4 \log k)^k)$	2002
Kanj 等 ^[132]	$\mathcal{O}^*((2 \log k + 2 \log \log k + 18)^k)$	2004
Raman 等 ^[133]	$\mathcal{O}^*((12 \log k / \log \log k + 6)^k)$	2006
Guo 等 ^[135]	$\mathcal{O}^*(37.7^k)$	2006
Dehne 等 ^[134,136]	$\mathcal{O}^*(10.6^k)$	2005
Cao ^[141]	$\mathcal{O}^*(8^k)$	2018
Chen 等 ^[137,138]	$\mathcal{O}^*(5^k)$	2007
Cao 等 ^[139,142]	$\mathcal{O}^*(3.83^k)$	2010
Kociumaka 和 Pilipczuk ^[140]	$\mathcal{O}^*(3.619^k)$	2014
Iwata 和 Kobayashi ^[51,52]	$\mathcal{O}^*(3.46^k)$	2019

在精确算法方面, Razgon^[147]首次于 2006 年得到了 FVS 问题的非平凡精确算法, 其时间复杂度为 $\mathcal{O}(1.8899^n)$ 。随后, 包括 Fomin 和 Razgon 在内的几位合著者^[148]将时间复杂度降低至 $\mathcal{O}(1.7548^n)$ 。2013 年, Xiao 和 Nagamochi^[149,150]提出了运行时间为 $\mathcal{O}(1.7266^n)$ 的精确算法。在此之后, Fomin 等人^[97,104]提出的单调局部搜索技术进一步改进了这一结果。结合 Kociumaka 和 Pilipczuk 的参数算法^[140], 可以得到运行时间为 $\mathcal{O}(1.7237^n)$ 的精确算法。Gaspers 和 Lee^[151]通过推广单调局部搜索技术, 给出了运行时间为 $\mathcal{O}(1.7117^n)$ 的精确算法。此外, Iwata 和 Kobayashi 的参数算法^[51,52]也意味着 FVS 问题的精确算法运行时间可以进一步降低至 $\mathcal{O}(1.7110^n)$ 。表4-3展示了 FVS 问题的精确算法的研究历程。

表 4-3 反馈集问题精确算法研究历程

作者	精确算法	发表年份
Razgon ^[147]	$\mathcal{O}(1.8899^n)$	2006
Fomin 等 ^[148]	$\mathcal{O}(1.7548^n)$	2008
Xiao 和 Nagamochi ^[149,150]	$\mathcal{O}(1.7266^n)$	2013
Kociumaka 和 Pilipczuk ^[140] 与 Fomin 等 ^[97,104]	$\mathcal{O}(1.7237^n)$	2016
Gaspers 和 Lee ^[151]	$\mathcal{O}(1.7117^n)$	2017
Iwata 和 Kobayashi ^[51,52] 与 Fomin 等 ^[97,104]	$\mathcal{O}(1.7110^n)$	2019

相比于无向图, 有向图上的反馈集问题更加困难, DFVS 问题目前仅有 1995 年 Seymour^[152]提出的近似率为 $\mathcal{O}(\log k \log \log k)$ 的近似算法。事实上, 在独特游戏猜想 (Unique Games Conjecture) 成立的假设下, DFVS 问题不存在常数近似率的近似算法^[153]。此外, 直到 2008 年, Chen 等人^[120,121]通过给出时间复杂度为 $\mathcal{O}(k!4^k k^4 nm)$ 的参数算法, 首次证明了 DFVS 问题是 FPT 的。关于参数算法的唯

一改进来自于 Lokshtanov 等人^[154]的工作，他们给出了关于 $n + m$ 线性时间的参数算法，其运行时间为 $\mathcal{O}(k!4^k k^5(n + m))$ 。DFVS 问题是否存在单指数时间的参数算法是该领域内一个极其重要的公开性难题^[155]。需要强调的是，由于 DFVS 问题和 DFAS 问题之间存在保持解集大小不变的多项式时间归约，因此 DFAS 问题能在 $\mathcal{O}(k!4^k k^6(n + m))$ 时间内求解^[154]。此外，DFVS 问题和 DFAS 问题是否存在多项式核也是一个长期公开性的难题^[155]。

就 DFVS 问题的精确算法而言，Razgon^[156]提出了运行时间为 $\mathcal{O}(1.9977^n)$ 的分支搜索算法，这是目前已知的唯一一例非平凡精确算法。相较之下，DFAS 问题目前最快的精确算法是运行时间为 $\mathcal{O}^*(2^n)$ 的动态规划算法^[157, 158]，并且此算法需要指数级大小的空间。目前四类反馈集问题的总体研究现状如表4-4所示。

表 4-4 反馈集问题算法研究现状

问题	复杂性	近似率	参数算法	多项式核	精确算法
FVS	NP-难	$2^{[128-130]}$	$\mathcal{O}^*(3.46^k)^{[52]}$	$2k^2 + k^{[102]}$	$\mathcal{O}(1.7110^n)^{[52, 97]}$
FES	P	—	—	—	—
DFVS	NP-难	$\mathcal{O}(\log k \log \log k)^{[152]}$	$\mathcal{O}^*(k!4^k)^{[154]}$	未知 ^[155]	$\mathcal{O}(1.9977^n)^{[156]}$
DFAS	NP-难	$\mathcal{O}(\log k \log \log k)^{[152]}$	$\mathcal{O}^*(k!4^k)^{[154]}$	未知 ^[155]	$\mathcal{O}(2^n)^{[157, 158]}$

4.1.3 子集反馈集问题

在许多经典的图论问题中，其目标都可以归结为寻找足够小的点集（或边集）以经过特定结构，例如反馈集问题目标中的特定结构是所有环。但在一些实际应用案例中，图的规模十分庞大，通常不存在足够小的点集（或边集）经过所有特定结构。基于这样的事实，研究者们转而探究更具普遍性的问题，称作原图论问题的子集版本（Subset Version）或关键集版本（Terminal Set Version）^[57, 58]。此类问题的研究焦点是一个给定的顶点集（或边集），称作关键集，其中的元素称为关键点（或关键边）。子集版本的目标则松弛为寻找足够小的点集（或边集）以经过包含了关键点或边的特定结构。当关键集取为全部顶点（或边）时，子集版本问题退化为原图论问题，因此子集问题的更加具备一般性和普适性。

反馈集问题所对应的子集问题称为子集反馈集问题。给定图 \mathbf{G} 和点子集 $T \subseteq V(\mathbf{G})$ （或边子集 $T \subseteq E(\mathbf{G})$ ），子集反馈集问题旨在判定是否存在大小不超过 k 的点集或边集，使得关键点均不出现在剩余图的环中；即是判定是否存在大小不超过 k 的 T -反馈点集（或 T -反馈边集）。本文中，关键集均记作 T ，其大小记作 $\tau = |T|$ ，若 T 是点子集，其中的元素称为关键点；若 T 是边子集，其中的元素称为关键边。子集反馈集的输入实例可以表示为 $\mathcal{I} = (\mathbf{G}, T, k)$ 。若图 \mathbf{G} 存在大小不超过 k 的 T -反馈集 S ，则称实例 \mathcal{I} 有解且 S 为 \mathcal{I} 的一个解；否则称实例 \mathcal{I} 无解。

图 **G** 的最小 T -反馈集称为实例 \mathcal{I} 的最小解或最优解。

根据图 **G** 是有向或是无向图、反馈集是边集或是点集以及关键集 T 是边集或是点集，可定义以下八种不同版本的问题：

- (点) 子集反馈点集问题 (Subset Feedback Vertex Set Problem)，简称 SFVS 问题；
- (点) 子集反馈边集问题 (Subset Feedback Edge Set Problem)，简称 SFES 问题；
- 边子集反馈点集问题 (Edge Subset Feedback Vertex Set Problem)，简称 ESFVS 问题；
- 边子集反馈边集问题 (Edge Subset Feedback Edge Set Problem)，简称 ESFES 问题；
- 有向(点)子集反馈点集问题 (Directed Subset Feedback Vertex Set Problem)，简称 DSFVS 问题；
- 有向(点)子集反馈边集问题 (Directed Subset Feedback Arc Set Problem)，简称 DSFAS 问题；
- 有向边子集反馈点集问题 (Directed Arc Subset Feedback Vertex Set Problem)，简称 DASFVS 问题；
- 有向边子集反馈边集问题 (Directed Arc Subset Feedback Arc Set Problem)，简称 DASFAS 问题。

此外，在子集反馈集问题所涉及的一些实际场景中，关键集中的元素不允许被删除，换言之，反馈集中不允许包含关键集中的元素。这类问题被称为限制版 (Restricted Version) 问题。当关键集与 T -反馈集同为点集或边集时，均存在限制版本，共分为以下四种限制版本问题：

- 限制版子集反馈点集问题 (Restricted Subset Feedback Vertex Set Problem)，简称 R-SFVS 问题；
- 限制版边子集反馈边集问题 (Restricted Edge Subset Feedback Edge Set Problem)，简称 R-ESFES 问题；
- 限制版有向子集反馈点集问题 (Restricted Directed Subset Feedback Vertex Set Problem)，简称 R-DSFVS 问题；
- 限制版有向边子集反馈边集问题 (Restricted Directed Arc Subset Feedback Arc Set Problem)，简称 R-DASFAS 问题。

对于没有上述限制要求的子集反馈集问题，统称之为非限制版本 (Unrestricted Version)。综上，子集反馈集问题共分为十二种，其中有向图和无向图上的子集反

馈集问题各六种（参见附录 A）。见表4-5对 12 种问题的归类。

表 4-5 反馈集问题类别

图类型	关键集	反馈集	限制与否	问题名	简称
无向	点集	点集	非限制	子集反馈点集问题	SFVS
无向	点集	点集	限制	限制版子集反馈边集问题	R-SFVS
无向	点集	边集	非限制	子集反馈边集问题	SFES
无向	点集	边集	限制	—	—
无向	边集	点集	非限制	边子集反馈点集问题	ESFVS
无向	边集	点集	限制	—	—
无向	边集	边集	非限制	边子集反馈边集问题	ESFVS
无向	边集	边集	限制	限制版边子集反馈边集问题	R-ESFES
有向	点集	点集	非限制	有向子集反馈点集问题	DSFVS
有向	点集	点集	限制	限制版有向子集反馈边集问题	R-DSFVS
有向	点集	边集	非限制	有向子集反馈边集问题	DSFAS
有向	点集	边集	限制	—	—
有向	边集	点集	非限制	有向边子集反馈点集问题	DASFVS
有向	边集	点集	限制	—	—
有向	边集	边集	非限制	有向边子集反馈边集问题	DASFAS
有向	边集	边集	限制	限制版有向边子集反馈边集问题	R-DASFAS

子集反馈集问题最初由 Even 等人^[5-9]在 2000 年前后提出。鉴于 DFVS 问题和 DFAS 问题均是 NP-难的，因此有向图上六种非限制版子集反馈集问题也都是 NP-难的。在无向图上，类似于证明 FVS 问题 NP-难性的归约思想，也可得出 SFVS 问题、ESFVS 问题和 R-SFVS 问题的 NP-难性^[159]。对于无向图上的三种反馈边集问题，除限制版本 R-ESFES 问题为 NP-难外，另外两个非限制版本 SFES 问题和 ESFES 问题均可在多项式时间求解^[160]。具体地，Xiao 和 Nagamochi 在文献 [160] 中证明了 SFES 问题可以在线性时间内求解，该算法首先处理远离关键点的边，生成一个关键点“稠密”的图，然后根据该图的生成树性质设计出多项式时间算法求解 SFES 问题，同时将此算法扩展应用于求解 ESFES 问题，证明了 ESFES 问题仍然可以在线性时间内求解。

尽管子集反馈集问题有多达十二种不同的版本，且其中十个版本是 NP-难问题，但是这些问题之间存在非常紧密的多项式时间归约关系^[159]。这些归约关系能够保持解集大小 k 不增，部分归约甚至能同时保持关键集大小 τ 不增（参见图4-1）。凭借这些归约关系，相关算法研究只需聚焦于少量代表性的问题上，因为代表性问题的求解算法能很好地应用于其他版本的问题。这种归约关系的存在不仅揭示了不同子集反馈集问题之间的内在联系，还指明了研究的方向与方法，为这类 NP-难问题的算法研究提供了有效的指导意义。

具体地，在无向图上，SFVS 问题、ESFVS 问题和 R-SFVS 问题之间存在保持

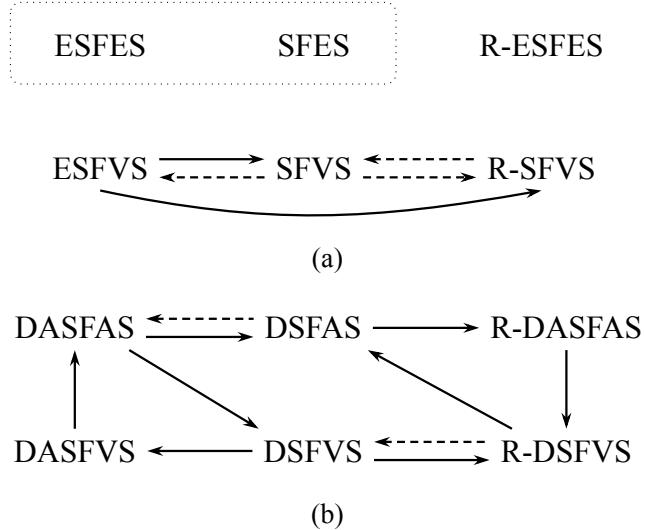


图 4-1 十二种子集反馈集问题的归约关系。(a) 六类无向图上的子集反馈集问题的归约关系。(b) 六类有向图上的子集反馈集问题的归约关系。实线箭头表示保持解集大小 k 和关键集大小 τ 的多项式时间归约，而虚线箭头表示只保持解集大小 k 的多项式时间归约。虚线内的问题多项式时间可求解的，其他问题则是 NP- 难的。

解集大小的多项式时间归约。因此，相关算法研究主要集中在 SFVS 问题上。相比之下，R-ESFES 问题较为特殊，目前尚未有关于 R-ESFES 与子集反馈点集问题间保持解集大小或关键集大小的归约。在有向图上六个版本之间均可互相归约且保持解集大小^[5, 161, 162]。因此，相关算法研究集中于三个代表性问题：R-ESFES 问题、SFVS 问题和 DSFVS 问题。

4.1.4 子集反馈集问题的求解算法

本小节将归纳整理子集反馈集问题的确定性算法，由4.1.3小节的论述，相关研究主要集中于 R-ESFES 问题、SFVS 问题和 DSFVS 问题三个版本。

Even 等人^[5]在提出子集反馈集问题时，给出了 SFVS 问题和 DSFVS 问题的近似算法，其中 SFVS 问题的近似率为常数 8，DSFVS 问题的近似率为 $\mathcal{O}(\log k \log \log k)$ 和 $\mathcal{O}(\log^2 \tau)$ ^[163, 164]。随后，Even 等人^[8, 9]又将 DSFVS 问题的近似率改进到 $\mathcal{O}(\log \tau \log \log \tau)$ 。对于 R-ESFES 问题而言，虽然尚未有针对此问题的近似算法，但是 Even 等人^[6, 7]提供了带边权的 SFES 问题的 2 倍近似算法，而 R-ESFES 问题可以被视为此问题的特例。

无向图上的子集反馈集问题是否属于 FPT 类一度是一个公开性问题^[165]。然而，三个独立的团队^[166-168]通过不同方法分别给出了肯定答案。具体地，

Guillemot^[166, 169]研究了一个更具一般性的问题，为该问题设计了运行时间为 $\mathcal{O}^*((8k+1)^k)$ 的参数算法；Cygan 等人^[167, 170]给出了时间复杂度为 $\mathcal{O}^*(2^{\mathcal{O}(k \log k)})$ 的参数算法；随后 Kawarabayashi 和 Kobayashi^[168]利用图子式理论（Graph Minor Theory）也证明了 SFVS 问题是 FPT 的。直到 2014 年，Iwata 等人^[171]首次给出单指数时间的参数算法，该算法基于线性规划技术且时间复杂度为 $\mathcal{O}^*(4^k)$ 。随后，两个独立的工作^[172, 173]分别给出了关于 $n+m$ 线性时间的参数算法，其中 Iwata 等人^[173]提出的算法以 $\mathcal{O}(4^k k(n+m))$ 的时间复杂度成为了当前已知最快的参数算法。目前，SFVS 问题是否存在确定性的多项式核仍然是一大公开性难题^[165]。表4-6展示了 SFVS 问题的参数算法研究历程。

表 4-6 子集反馈点集问题参数算法研究历程

作者	参数算法	发表年份
Kawarabayashi 和 Kobayashi ^[168]	$\mathcal{O}(f(k)n^2m)$	2012
Guillemot ^[166, 169]	$\mathcal{O}^*((8k+1)^k)$	2008
Cygan 等 ^[167, 170]	$\mathcal{O}^*(2^{\mathcal{O}(k \log k)})$	2011
Lokshtanov 等 ^[172, 174]	$2^{\mathcal{O}(k \log k)}(n+m)$	2015
Iwata 等 ^[171, 175]	$\mathcal{O}^*(4^k)$	2014
Iwata 等 ^[173]	$\mathcal{O}(4^k km)$	2018

SFVS 问题的精确算法研究可追溯至 Fomin 等人^[176, 177]的研究工作，他们提出了运行时间为 $\mathcal{O}(1.8638^n)$ 的精确算法，该算法不仅能够找到最小 T -反馈集，还可以枚举所有最小 T -反馈集。随后的研究进一步改进了算法的时间复杂度，将其降低到了 $\mathcal{O}(1.7743^n)$ ^[178]。

值得注意的是，在 Iwata 等人^[175]提出的 $\mathcal{O}^*(4^k)$ 时间复杂度的参数算法基础上，采用单调局部搜索技术可以获得时间复杂度为 $1.75^n 2^{\mathcal{O}(n)}$ 的精确算法。这是目前最快的 SFVS 问题的精确算法。表4-7展示了 SFVS 问题精确算法的研究历程。

表 4-7 子集反馈点集问题参数算法研究历程

作者	精确算法	发表年份
Fomin 等 ^[176, 177]	$\mathcal{O}(1.8638^n)$	2014
周晓清和肖鸣宇 ^[178]	$\mathcal{O}(1.7743^n)$	2018
Iwata 等 ^[171, 175] 与 Fomin 等 ^[97, 104]	$1.75^n 2^{\mathcal{O}(n)}$	2016

对于 R-ESFES 问题而言，由于缺乏其他子集反馈集问题保持解集大小不变的归约关系，这一问题的相关研究较为有限。Xiao 和 Nagamochi^[160]首次给出该问题的 FPT 算法，其运行时间为 $\mathcal{O}((12k)^{6k} 2^k nm)$ 。尽管这是一个重要的进展，但针对 R-ESFES 问题的非平凡精确算法尚未出现。

相较于无向图，有向图上子集反馈集问题的算法研究更具有挑战性，该问题的

参数和精确算法在近年内取得一些突破性进展。Chitnis 等人^[161, 162]证明了 DSFVS 问题是 FPT 的，并且给出了时间复杂度为 $\mathcal{O}^*(2^{\mathcal{O}(k^3)})$ 的参数算法。尽管其时间复杂度相对较高，但这是目前已知最快的参数算法。至于 DSFVS 问题是否存在多项式核，这仍是一个公开性的难题^[155]。在精确算法方面，Chitnis 等人^[57, 58]提出了针对包括 DSFVS 问题在内的一系列子集版本问题的精确算法。依据 Razgon^[156]提出的时间复杂度为 $\mathcal{O}(1.9977^n)$ 的精确算法，Chitnis 等人证明了 DSFVS 问题可以在 $\mathcal{O}(1.9993^n)$ 时间内求解。表4-8展示了三个代表性子集反馈集问题的算法研究现状。

表 4-8 子集反馈集问题算法研究现状

问题	复杂性	近似率	参数算法	多项式核	精确算法
SFVS	NP-难	$8^{[5]}$	$\mathcal{O}^*(4^k)^{[173]}$	未知	$1.75^{n+o(n)}^{[97, 175]}$
R-ESFES	NP-难	$2^{[7]}$	$\mathcal{O}^*(2^{\mathcal{O}(k \log k)})^{[160]}$	未知	$\mathcal{O}(2^m)$
DSFVS	NP-难	$\mathcal{O}(\log k \log \log k)^{[164]}$	$\mathcal{O}^*(2^{\mathcal{O}(k^3)})^{[162]}$	未知 ^[155]	$\mathcal{O}(1.9993^n)^{[58]}$

4.2 特殊图类上的反馈集与子集反馈集问题

特殊图上反馈集与子集反馈集问题的计算复杂性和算法也有大量的研究积累。在某些场景中，问题的输入图具有一些特定的性质。这使得求解特殊图上的反馈集问题相较容易，故针对特殊图的上反馈集与子集反馈集问题的研究具有更高的实际价值。本研究关注的图类涉及度有界图（Degree-Bounded Graph）图类、平面图（Planar Graph）、竞赛图（Tournament）图类、相交图（Intersection Graph）图类。因此，本节对上述图类上的反馈集和子集反馈集问题的计算复杂性、参数及精确算法的研究进展进行了详细总结和整理。

4.2.1 平面图与度有界的图类

平面图是指可以嵌入在平面上使得边不交叉的图。度有界图则顾名思义是指最大度不超过某一常数的图。特别地，最大度不超过 d 的图称为 d 度图（ d -Degree Graph）。无论是在图论还是算法领域，平面图和度有界的图类均十分常见。包括反馈集问题在内的众多图问题的算法研究都依赖于对顶点最大度的分析和利用^[52, 96, 141, 179, 180]。平面图与度有界图属于稀疏图类，故在探究反馈集问题的计算复杂性分类时，平面性和度有界的特征通常被同时纳入考虑。因此，本小节将同时对两个图类上反馈集问题的计算复杂性分类的研究进展进行总结。表4-9展示了平面图以及度有界图类上的反馈集与子集反馈集问题的研究现状。

已知 3 度平面图上顶点覆盖问题是 NP-难的^[181]，并且顶点覆盖问题到反馈

表 4-9 平面图与度有界的图类上的反馈集相关问题的计算复杂性

问题	图类	复杂性	多项式时间算法或参数/精确算法
FVS	平面图	NP-难 ^[181]	$\mathcal{O}^*(2^{15.11\sqrt{k}})$, $2^{\mathcal{O}(\sqrt{n})}$ ^[182, 183]
FVS	3 度图	P ^[184, 185]	$\mathcal{O}(n^2 \log^6 n)$ ^[186]
FVS	4 度、二部、平面图	NP-难 ^[187]	—
FVS	4 正则、哈密顿、平面图	NP-难 ^[188]	—
FVS	补平面图	P ^[189-191]	$\mathcal{O}(n^8)$ ^[159]
SFVS	补平面图	P ^[159]	$\mathcal{O}(n^{16})$ ^[159]
DFVS	1 出度（或 1 入度）图	P (5.2 节)	$\mathcal{O}(n)$ (5.2 节)
DFAS	1 出度（或 1 入度）图	P (5.2 节)	$\mathcal{O}(n)$ (5.2 节)
DFVS	2 出度且 2 入度图	NP-难 (5.2 节)	—
DFAS	2 出度且 2 入度图	NP-难 (5.2 节)	—
DFVS	2 出度且 2 入度、平面图	NP-难 (5.2 节)	—
DFAS	平面图	P ^[184, 185]	$\mathcal{O}(n^{2.5} \log n)$ ^[192]

点集问题的归约能够保证平面性^[3]，所以 6 度平面图上的 FVS 问题、DFVS 问题和 DFAS 问题均是 NP-难的。更准确地，3 入度且 3 出度有向图上的 DFVS 问题和 DFAS 问题都是 NP-难的。另一方面，2 度图上的上述三个问题均能在多项式时间内求解。因此，进一步的研究目标是缩小复杂性上下界的差距，明确多项式时间可解和 NP-难的分界线，完备关于最大度的复杂性分类。

对无向图而言，Speckenmeyer^[187]在博士论文中进一步证明了 4 度图上 FVS 问题仍旧保持 NP-难性。完备的计算复杂性分类结果直到 1988 年才得以圆满实现。当年，两个独立的研究团队同时证明了 3 度图上 FVS 问题可在多项式时间内解决。其中，Ueno 等人^[193]和 Furst 等人^[194]分别采用采用拟阵（Matroid）技术和分析最大亏格（Maximum Genus）给出了多项式时间算法。目前，借助图拟阵上的相关算法，3 度图上 FVS 问题可在 $\mathcal{O}(n^2 \log^6 n)$ 时间内求解。

对有向图而言，DFVS 问题和 DFAS 问题的计算复杂性分类相对困难。此前，唯一的研究成果出自 Garey 和 Johnson 于 1979 年出版的专著^[195]，其中未经证明地指出，2 出度且 2 入度图以及 3 出度且 3 入度图上的 DFVS 问题是 NP-难的。本文第五章的主要工作之一是完整地对一般图和平面图上的 DFVS 问题和 DFAS 问题按照最大度进行计算复杂性分类，证明了 2 出度且 2 入度图上的 DFVS 问题和 DFAS 问题均为 NP-难的。同时证明了即使在 2 出度且 2 入度平面图上，DFVS 问题依然是 NP-难的。值得注意的是，虽然一般有向图上的 DFVS 问题和 DFAS 问题可互相归约，但是在平面图上它们的计算复杂性有所不同。事实上，基于线性规划的对偶性^[184, 185]，平面图上 DFAS 问题可以在 $\mathcal{O}(n^{2.5} \log n)$ 时间内求解^[192]。此外，考虑到平面图上 FVS 问题是 NP-难的，该问题的参数算法也受到了广泛研究。通过应用二维化技术（Bidimensionality），可以得到亚指数运行时间 $\mathcal{O}^*(2^{15.11\sqrt{k}})$

的参数算法^[182, 183]，由此也表明此问题可存在亚指数时间的精确算法。

即便在平面二部图上，FVS 问题和 DFVS 问题依然是 NP-难的。这种归约操作是平凡的：仅需将每条边替换为一个二度点，这个点连接该边的两个端点。此操作既保持了输入图的平面性，也不改变最小解集的大小，同时实现了将任意图转化为二部图的目的。此外，Cavallaro 在硕士论文^[188]中研究了哈密顿图（Hamiltonian Graph）上的 FVS 问题，证明了即使在已知哈密顿环的条件下，4-正则图（4-Regular Graph）上 FVS 问题依然是 NP-难的。

此外，补平面图（Coplanar Graph）上 FVS 问题是多项式时间可计算的。如果一个图的补图是平面图，则称其为补平面图。根据四色定理^[189, 190]，补平面图的点集可以划分为四个团，且此划分可在平方级时间内完成^[191]。因此，补平面图上最大导出森林的至多超过 8 个顶点，故通过枚举所有大小不超过 8 的顶点集，补平面图上 FVS 问题可以在多项式时间内求解。进一步扩展此算法，能够证明补平面图上 SFVS 问题可在 $\mathcal{O}(n^{16})$ 时间内求解^[159]。

4.2.2 竞赛图图类

竞赛图是由完全图进行边定向得到的有向图。二部竞赛图（Bitournament）则是由完全二部图进行边定向得到的有向图。竞赛图因用于刻画循环赛制的胜负关系而得名。竞赛图上 DFVS 问题和 DFAS 问题的概念通常应用于排名算法^[12-15]。表4-10展示了竞赛图和二部竞赛图上反馈集和子集反馈集问题的研究现状。

表 4-10 竞赛图和二部竞赛图上的反馈集与子集反馈集问题的研究现状

问题	图类	复杂性	参数/精确算法
DFVS	竞赛图	NP-难 ^[196]	$\mathcal{O}^*(1.6181^k)^{[75]}$, $\mathcal{O}(1.3820^n)^{[97, 175]}$
DSFVS	竞赛图	NP-难 ^[196]	$\mathcal{O}^*(1.6181^k)$, $\mathcal{O}(1.3820^n)$ (6.4节)
R-DSFVS	竞赛图	NP-难 (6.5节)	$\mathcal{O}^*(1.2529^k)$, $\mathcal{O}(1.1996^n)$ (6.5节)
DFAS	竞赛图	NP-难 ^[197-199]	$\mathcal{O}^*(2^{\mathcal{O}(\sqrt{k})})$, $2^{\mathcal{O}(\sqrt{n})}$ ^[200]
DFVS	二部竞赛图	NP-难 ^[201, 202]	$\mathcal{O}^*(1.6181^k)^{[203]}$, $\mathcal{O}(1.3820^n)^{[97, 175]}$
DFAS	二部竞赛图	NP-难 ^[204]	$\mathcal{O}^*(3.3723^k)^{[62, 67]}$, $\mathcal{O}(1.7035^n)^{[97, 175]}$

竞赛图上 DFVS 问题的 NP-难性在 1989 年已得到了证明^[196]。然而，直到 21 世纪初，研究人员才解决竞赛图上 DFAS 问题以及二部竞赛图上 DFVS 问题和 DFAS 问题的计算复杂性。具体地，Cai 等人^[201, 202]在 1999 年证明了二部竞赛图上的 DFVS 问题是 NP-难的。此后，通过不同的归约策略，三个独立的研究^[197-199]分别证明了竞赛图上 DFAS 问题是 NP-难的。同年，Guo 等人^[204]证明了二部竞赛图上 DFAS 问题也是 NP-难的。

竞赛图类上反馈集问题的算法研究也在同一时期成为了热点^[67, 205, 206]。目前，

竞赛图和二部竞赛图上 DFVS 问题的最快参数算法均由同一个研究组提出，运行时间达到了 $\mathcal{O}^*(1.6181^k)$ ，超越了 2^k 的时间复杂度瓶颈。有趣的是，竞赛图上 DFAS 问题比其他三类问题更为简单，Alon 等人^[207]首次于 2009 年证明了该问题可以在亚指数时间内求解，当前最快参数算法运行时间为 $\mathcal{O}^*(2^{\mathcal{O}(\sqrt{k})})$ ^[200]，这也意味着 DFAS 问题存在亚指数时间的精确算法。其他三类问题的最快精确算法记录均由基于单调局部搜索的精确算法保持。

竞赛图上 SFVS 问题也是 NP-难问题，本文第六章进一步证明了竞赛图上 R-SFVS 问题是 NP-难的。此外，本文第六章给出了这两个问题目前最快的参数与精确算法。特别地，本文证明了竞赛图上 SFVS 问题也可以在 $\mathcal{O}^*(1.6181^k)$ 时间内求解，不仅打破了 2^k 的时间复杂度瓶颈，还与竞赛图上 DFVS 问题最快参数算法的时间复杂度持平。

4.2.3 相交图图类

相交图的概念源于一组集合（通常为几何图形）之间的相交关系。在此模型中，每个集合对应图中的一个顶点，两个顶点之间存在一条边当且仅当它们所对应的集合有非空交集。典型的相交图包括补比较图（Co-Comparability Graph）、弦图（Chordal Graph）、分裂图（Split Graph）、梯形图（Trapezoid Graph）、置换图（Permutation Graph）、弧线图（Circular Arc Graph）、圆图（Circle Graph）、区间图（Interval Graph）等等^[208]。相交图模型广泛应用于各类调度问题中^[209]，不同类型的相交图各具独特的结构特征，为解决特定问题提供了有力的图论工具。表4-11展示了相交图上反馈集和子集反馈集问题的研究现状。

相交图上 FVS 问题的算法研究可追溯至 20 世纪 80 年代弦图上的可染色子图问题^[210]。弦图是一棵树的子树构成的相交图，基于树分解上的动态规划思想，Yannakakis 和 Gavril^[210]提出了弦图上 FVS 问题的首个多项式时间算法。经过适当的状态预处理，该算法可以在 $\mathcal{O}(n^4)$ 时间内求解弦图上 FVS 问题。区间图和分裂图分别是路和星图（Star Graph）上的子树构成的相交图，作为弦图的子类，在这些子图类上的 FVS 问题同样可在多项式时间内求解的。需要强调的是，分裂图是弦图和补弦图的交类，基于分裂图的结构特性，文献 [210] 中的算法可以在立方时间复杂度内求解分裂图上 FVS 问题。此后，Lu 和 Tang^[211]提出了一种动态规划算法，将求解区间图上 FVS 问题的时间复杂度改进至线性时间复杂度。关于子集反馈点集问题，Fomin 等人^[176]于 2011 年证明了弦图甚至分裂图上 SFVS 问题和 R-SFVS 问题均属于 NP-难问题。本文在第七章和第八章分别研究了分裂图和弦图上 SFVS 问题以及 R-SFVS 问题，提出了当前最快的参数与精确算法。其中，参数

表 4-11 相交图类上的反馈集和子集反馈集问题的研究现状

问题	图类	计算复杂性	多项式时间算法或参数/精确算法
FVS	弦图	P ^[210]	$\mathcal{O}(n^4)$ ^[210]
SFVS	弦图	NP-难 ^[176, 177]	$\mathcal{O}^*(1.8192^k)$, $\mathcal{O}(1.3788^n)$ (8.3节)
R-SFVS	弦图	NP-难 ^[176, 177]	$\mathcal{O}^*(1.2529^k)$, $\mathcal{O}(1.1520^n)$ (8.4节)
FVS	分裂图	P ^[210]	$\mathcal{O}(n^3)$ ^[210]
SFVS	分裂图	NP-难 ^[176, 177]	$\mathcal{O}^*(1.8192^k)$, $\mathcal{O}(1.3488^n)$ (7.3节)
R-SFVS	分裂图	NP-难 ^[176, 177]	$\mathcal{O}^*(1.2529^k)$, $\mathcal{O}(1.1520^n)$ (7.4节)
FVS	区间图	P ^[210]	$\mathcal{O}(m + n)$ ^[211]
SFVS	区间图	P ^[212, 213]	$\mathcal{O}(n^3)$ ^[212, 213]
FVS	置换图	P ^[214, 215]	$\mathcal{O}(nm)$ ^[216]
SFVS	置换图	P ^[212, 213]	$\mathcal{O}(m^2(m + n))$ ^[212, 213]
FVS	二部梯形图	P ^[214, 215]	$\mathcal{O}(m + n)$ ^[217]
FVS	补比较图	P ^[218]	$\mathcal{O}(n^2m)$ ^[219]
FVS	圆-d-拱形图	P ^[220]	$\mathcal{O}(n^{2d+5})$ ^[220]
FVS	圆图	P ^[220]	$\mathcal{O}(n^7)$ ^[220]
FVS	圆-梯形图	P ^[220]	$\mathcal{O}(n^9)$ ^[220]
SFVS	弧线图	P ^[212, 213]	$\mathcal{O}(n^4)$ ^[212, 213]

算法均突破了 2^k 的计算复杂性瓶颈。

置换图上 FVS 问题也是早期的研究课题之一。 n 阶置换图由一个 n 阶置换所定义, 图中的每个顶点代表置换中的一个元素, 若两个元素的置换形成逆序对, 则它们对应的顶点之间存在一条边。Brandstädt 和 Kratsch^[214, 215]为置换图上 FVS 问题提出了首个多项式时间算法, 其运行时间为 $\mathcal{O}(n^6)$ 。随后, Brandstädt^[221]改进了此算法, 将时间复杂度降低至 $\mathcal{O}(nm(n^2 - m))$ 。目前, Liang^[216]提出的动态规划算法, 运行时间为 $\mathcal{O}(nm)$, 是求解置换图上 FVS 问题已知最快的算法。

梯形图是由两条平行线之间的梯形构成的相交图。置换图是梯形图的子类。Honma 等人^[222]通过枚举极大团并计算 4 长导出环的技术路线, 提出了一种求解梯形图上 FVS 问题的算法, 该算法运行时间为 $\mathcal{O}(n^{2.68} + \gamma n)$, 其中 γ 为枚举所有极大团的编码长度。需要强调的是, γ 的值关于点数 n 可能是指数级的。此外, 置换图的子类, 二部梯形图 (Bipartite Trapezoid Graph), 也是重要的相交图类, 其上的 FVS 问题已被证明可在线性时间内求解^[217]。

补比较图是由两条平行线之间 (从一条直线到平行的另一条) 的连续曲线构成的相交图。梯形图是补比较图的子类。Coorg 和 Rangan^[218]首次给出了该问题的 $\mathcal{O}(n^4)$ 时间的算法。Liang 和 Chang^[219]随后提出了更高效的动态规划算法, 将运行时间优化至 $\mathcal{O}(n^2m)$ 。

圆-d-拱形图 (Circle-d-Gon Graph) 是圆-d-拱形 (Circle-d-Gon) 构成的相交图, 其中圆-d-拱形是圆内不超过 d 个两两不相交的弦与圆弧围成的中间区域。

针对固定的正整数 d , Gavril^[220]给出了 $\mathcal{O}(n^{2d+5})$ 时间的动态规划算法, 证明了圆- d -拱形图上 FVS 问题是多项式时间可解的。圆- d -拱形图包含了多种常见的相交图类, 如梯形图、区间图、弧线图等。特别地, 圆-1-拱形图称为圆图, 圆-2-拱形图称为圆-梯形图 (Circle-Trapezoid Graph)。因此在这两个图类上的 FVS 问题分别可在 $\mathcal{O}(n^7)$ 和 $\mathcal{O}(n^9)$ 时间内求解。在这些图类上也有一些关于 SFVS 问题的可解性结果, Papadopoulos 和 Tzimas^[212,213]给出了运行时间为 $\mathcal{O}(n^3)$ 的动态规划算法, 用于求解区间图上 SFVS 问题, 该算法需要同时维护三个与关键集相关的集合大小以建立状态转移关系。基于此算法, Papadopoulos 和 Tzimas^[212,213]进一步提出了求解弧线图和置换图上 SFVS 问题的多项式时间算法, 时间复杂度分别为 $\mathcal{O}(n^4)$ 和 $\mathcal{O}(m^3(m+n))$ 。

4.3 本章小结

本章首先明确了反馈集问题和子集反馈集问题的定义。反馈集问题共分为四种版本, 而子集反馈集问题共可细分多达十二种不同的版本 (参见附录 A)。本文归纳并分析了这些版本间存在紧密的归约关系。基于这些归约关系, 相关算法研究工作通常聚焦于反馈点集问题和非限制版子集反馈点集问题。这两个问题也是本文的主要研究对象。

其次, 本章对全文涉及图类上的反馈集和子集反馈集问题的计算复杂性进行了整理和综述。值的强调的是, 除了本章所罗列的图类外, 禁止子图图类 (Forbidden Graph)、二部图图类上也有大量关于反馈集和子集反馈集问题的算法研究^[159]。由此可见, 反馈集及其相关问题在各图类中均具有显著的研究价值且引起了广泛的关注。

第五章 反馈集与子集反馈集问题的计算复杂性分类

反馈集与子集反馈集问题的计算复杂性分类研究是计算理论领域的核心课题。目前，针对这两个问题的分类研究已逐渐完善。本章对度有界（平面）图上反馈集问题以及小关键集下子集反馈集问题的计算复杂性进行了分类研究，解决了数十年来遗留下来的公开性难题。

5.1 研究背景与主要贡献

反馈集问题是图论中极为重要而基础的图问题。特别是在有向图上，DFVS 问题和 DFAS 问题均属于 Karp 的 21 个 NP-完全问题^[3]。鉴于反馈集问题在多众多领域中都具有广泛的应用，它们在平面图等特殊图类上的计算复杂性受到了大量关注。本文的主要研究内容之一是平面图上反馈集问题的计算复杂性，平面图上三种反馈集问题的严格定义如下。

平面图上反馈点集问题 (FVS(P))

输入: 一个平面图 \mathbf{G} 和一个自然数 k 。

输出: 判断平面图 \mathbf{G} 是否存在一个大小不超过 k 的反馈点集 $S \subseteq V(\mathbf{G})$ 。

平面图上有向反馈点集问题 (DFVS(P))

输入: 一个平面有向图 \mathbf{D} 和一个自然数 k 。

输出: 判断平面有向图 \mathbf{D} 是否存在一个大小不超过 k 的反馈点集 $S \subseteq V(\mathbf{D})$ 。

平面图上有向反馈边集问题 (DFAS(P))

输入: 一个平面有向图 \mathbf{D} 和一个自然数 k 。

输出: 判断平面有向图 \mathbf{D} 是否存在一个大小不超过 k 的反馈边集 $S \subseteq E(\mathbf{D})$ 。

实际上，在早期对反馈集问题进行的系统研究中，Younger 提出了关于平面图上有向反馈边集的著名猜想：DFAS(P) 是多项式时间可解的。该猜想随后被 Lucchesi 和 Younger 的研究成果所证实^[184, 185]。

关于图的最大度如何影响反馈集问题的计算复杂性也是早期研究的主要内容。对无向图而言，Speckenmeyer^[187]在其博士论文中证明了 4 度图上 FVS(P) 问题的 NP-难性。随后，Speckenmeyer 进一步专注于 3 度图上 FVS 问题^[223, 224]，为后续研究奠定了基础。Ueno 等人^[193]和 Furst 等人^[194]分别采用拟阵技术和最大亏格方法证明了 3 度图上 FVS 问题是多项式时间可解的，解决了无向图和平面无向图上反

馈点集问题的计算复杂性分类问题。有趣的是，4度图上 FVS 问题的 NP-难性证明早年并不广为人知，直到 2015 年，Speckenmeyer 的博士论文电子副本才能够在相关社区检索。在此之前，尽管一些团队^[225,226]对低度图上 FVS 问题的计算复杂性也展开过研究，但他们所设计的组件均无法保证平面性。

相较于无向图，有向图上反馈集问题的计算复杂性分类显得更为困难。除了证明 6 度图上 DFVS(P) 问题是 NP-难的平凡归约方法外，仅在 1979 年出版的一部专著^[195]中提及了一个未附证明的结论：3 度图上 DFVS 问题是 NP-难的。

子集反馈集问题也是重要的图连通性问题，它与多路割问题（Multiway Cut Problem）、多割问题（Multicut Problem）等经典图论问题高度相关。给定图 \mathbf{G} 和关键集 T ，多路割问题旨在图中找到不超过给定大小的割，使得关键元素被两两割开。给定图 \mathbf{G} 和关键点对集合 T ，多割问题旨在图中找到大小不超过给定值的割，使得每对关键元素均被割开（具体定义见附录附录 A）。当关键点数量为常数时，这类问题的参数复杂性在近年取得了重要进展^[85–89,227,228]。

研究现状与挑战 FVS 问题和 DFVS 问题的 NP-难性结果归功于 Cornell 大学的算法研讨会^[3]，是通过顶点覆盖问题的归约来实现的。具体地，此归约将 \mathbf{G} 中每条边替换为两条重边或两条相反方向的有向边^①。本章将此归约称为倍边操作。

DFAS 问题的 NP-难性建立在由 Lawler 和 Karp 提出的从 DFVS 问题到 DFAS 问题的归约基础之上^[3]。具体而言，对于 DFVS 问题中的有向图 \mathbf{D} ，将每个顶点 $v \in V(\mathbf{D})$ 替换为一条有向边 v^-v^+ ，同时将原图中的每条边 $vu \in E(\mathbf{D})$ 替换为 v^+u^- 。本章将此操作称为裂点操作。

基于 3 度平面图上顶点覆盖问题的 NP-难性^[181]以及倍边操作保证平面性的特点，可推断出 6 度图上的 FVS(P) 问题和 DFVS(P) 问题是 NP-难问题。然而，裂点操作并不能确保平面性。进一步地，FVS(P) 问题关于最大度 Δ 的计算复杂性的二分定理已经在 20 世纪 80 年代得出。具体地，当 $\Delta \leq 3$ 时，FVS(P) 问题是多项式时间可解的^[193,194]；而当 $\Delta \geq 4$ 时，FVS(P) 问题则是 NP-难的^[187]。关于 NP-难性，Speckenmeyer 设计了极其精巧的组件，通过反复替换高度顶点（参见图 5-1）证明了 4 度图上 DFVS(P) 问题是 NP-难的。

在平面有向图上，关于最大度 Δ 的 DFVS(P) 问题的计算复杂性分类工作长期进展缓慢。一个主要的原因在于，利用倍边操作所得到的有向图在采用组件替换技术时，似乎无法避免完全二部图 $\mathbf{K}_{3,3}$ 的非平面性事实。另一方面，通过倍边操作得到的有向图是三连通图，而根据 Whitney 嵌入定理^[229]，三连通平面图具有本

^① 若要求归约后的图保持为简单图，仅需对一条边进行分段（Subdivide），即通过添加一个度为 2 的顶点来连接该边的两个端点。

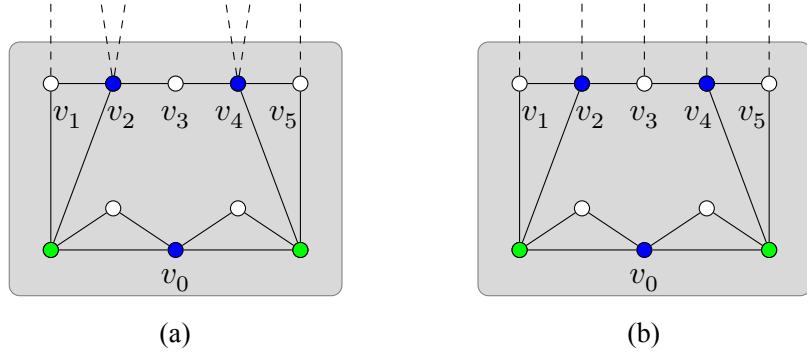


图 5-1 Speckenmeyer 的替换组件。(a) 替换 6 度点的替换组件, 其中包含 10 个顶点和 14 条边; (b) 替换 5 度点的替换组件^[187], 其中包含 10 个顶点和 14 条边; 连向原图其他顶点的边使用虚线表示并按照围绕顺序连接。任意最小反馈点集至少包含组件中的两个顶点, 且最多包含组件中的三个顶点。

质上唯一的嵌入, 这似乎宣告组件替换技术无法有效应用于该问题上。

常数大小关键点条件下子集反馈集问题的计算复杂性分类工作无论在理论上还是在实践中, 都具有重要意义。对于大多数非限制版本, 通过穷举搜索可以在 $\mathcal{O}(n^\tau)$ 时间内解决, 其中 τ 表示关键集大小。因此, 这类研究主要关注限制版本, 即解中不允许包含关键元素的变体问题。在子集反馈集问题被首次提出时, Even 等人^[163, 164]通过将节点多路割问题在多项式时间内归约至包含 1 个关键点的 R-SFVS 问题, 证明了当 $\tau \geq 1$ 时, R-SFVS 问题是 NP-难的。随后, Xiao 和 Nagamochi^[160]指出 R-SEFES 问题在 $\tau \leq 1$ 时可在多项式时间内解决, 而当 $\tau \geq 2$ 时, 问题转为 NP-难。

R-DSFVS 问题和 R-DASFAS 问题与有向多割问题极为相似, 均属于经典图连通性问题。对于有向多割问题, 当关键点对数 $\tau \leq 1$ 时, 等价于经典的流问题, 因此能在多项式时间内求解; 当关键点对数 $\tau = 2$ 时, 问题等价于有向多路割问题, 于 2012 年被证实是 FPT 的^[85, 86]; 当关键点对数 $\tau \geq 4$ 时, 该问题于 2016 年被证明为 W[1]-难的^[87, 88]; 对于 $\tau = 3$ 的情况, 最近的研究表明存在随机 FPT 算法求解此问题^[89]。

相较而言, R-DSFVS 问题和 R-DASFAS 问题更为简单, 无论关键点数量如何, 都被证明是 FPT 的^[161, 162]。事实上, 存在一种平凡的多项式时间归约算法, 可将 τ 个关键点的 R-DSFVS 问题 (或 τ 个关键边的 R-DASFAS 问题) 归约至 τ 个关键点的有向多割问题。然而, 反向的归约在 $FPT \neq W[1]$ 的假设下, 并不能保持关键元素数量。直观上, 将有向多割问题归约到 DSFAS 问题或 R-DASFAS 问题的方法是为每对关键点添加不可删除的反向边, 从而使得任何点对间的有向路径都转换

成了包含反向边的环。但这种添加反向边的操作同时增强了图的连通性。因此，相较于两个点对的有向多割问题来说，证明 DSFAS 问题或 R-DASFAS 问题在 $\tau = 2$ 时的 NP- 难性的难度更大。关于 R-DSFVS 问题在 $\tau = 2$ 时的 NP- 难性，出现在文献 [5] 中，但未给出完整证明。

主要贡献 本章研究了 DSFVS 问题、DSFAS 问题以及 DSFVS(P) 问题关于最大度的计算复杂性（见表 5-1）。

本章的核心贡献在于通过引入非正规平面嵌入的概念，全局调整了倍边操作中两条反向边的平面嵌入方式；同时，创新性地构造了有向图上高度顶点替换组件，给出了 SFVS(P) 问题关于最大度 Δ 的计算复杂性的二分定理。

表 5-1 度有界图上的反馈集问题的计算复杂性分类结果

问题	P	NP-完全
FVS	$\Delta \leq 3$ (文献 [193, 194])	$\Delta \geq 4$ (文献 [187])
FVS(P)	$\Delta \leq 3$ (文献 [193, 194])	$\Delta \geq 4$ (文献 [187])
DFVS	$\Delta \leq 2$ 或 $\Delta^* = 0$ (定理 5.1)	$\Delta \geq 3$ 且 $\Delta^* \geq 1$ (定理 5.1)
DFAS	$\Delta \leq 2$ 或 $\Delta^* = 0$ (定理 5.1)	$\Delta \geq 3$ 且 $\Delta^* \geq 1$ (定理 5.1)
DFVS(P)	$\Delta \leq 3$ 或 $\Delta^* \leq 1$ (定理 5.2)	$\Delta \geq 4$ 且 $\Delta^* \geq 2$ (定理 5.2)
DFAS(P)	任意 Δ (文献 [184, 185])	—

本章接着探讨了常数大小关键点集条件下子集反馈集问题的计算复杂性（见表 5-2）。建立了 DSFAS 问题、R-DSFVS 问题以及 R-DASFAS 问题之间的保持关键集和解集大小的多项式时间归约，证明了三者在关键点集大小相等时的 NP- 难性完全一致。为了保证图的连通性不受到归约的破坏，本研究摒弃了以有向多割问题分析 R-DSFVS 问题 NP- 难性的技术路线，而选用无向图上多路割问题进行归约，进一步证明了当且仅当关键点集大小 $\tau \geq 2$ 时，DSFAS 问题、R-DSFVS 问题和 R-DASFAS 问题均属于 NP- 难问题。给出了子集反馈集问题关于关键点集大小的计算复杂性二分定理。

表 5-2 关键集为常数的子集反馈集问题的计算复杂性分类结果

问题	P	NP-完全
R-SFVS	$\tau = 0$ (平凡的)	$\tau \geq 1$ (文献 [163, 164])
R-SEFES	$\tau = 1$ (文献 [160])	$\tau \geq 2$ (文献 [160])
R-DSFVS	$\tau = 1$ (文献 [5])	$\tau \geq 2$ (定理 5.15)
R-DASFAS	$\tau = 1$ (文献 [5])	$\tau \geq 2$ (定理 5.15)
DSFAS	$\tau = 1$ (文献 [5])	$\tau \geq 2$ (定理 5.15)
其余 7 类子集反馈集问题	$\tau = \mathcal{O}(1)$ (平凡的)	—

5.2 一般图与平面图上反馈集问题

回顾第三章给出的定义, 符号 Δ 表示有向图的最大度; Δ^* 表示有向图的最大入度和最大出度中的较小者。本章研究 DFVS 问题、DFAS 问题和 DFVS(P) 问题关于最大度 Δ 和定向最大度 Δ^* 的计算复杂性分类, 主要结论如下。

定理 5.1 当 $\Delta \leq 2$ 或 $\Delta^* \leq 0$ 时, DFVS 问题可以在线性时间内求解。当 $\Delta \geq 3$, $\Delta^* \geq 1$ 且均为常数时, DFVS 问题和 DFAS 问题均是 NP-完全的。

定理 5.2 当 $\Delta \leq 3$ 或 $\Delta^* \leq 1$ 时, DFVS(P) 问题可以在多项式时间内求解。当 $\Delta \geq 4$, $\Delta^* \geq 2$ 且均为常数时, DFVS(P) 问题是 NP-完全的。

定义两个简单的归约操作, 它们将多次应用于后文构造的归约中。

定义 5.3 (倍边操作) 将无向图 **G** 中的每条边替换为两条重边, 或将有向图 **D** 中每条边替换为两个相反方向的有向边。

由于每条边将被替换为一条长度为 2 的无向环或有向环, 所以归约后图中的任何反馈点集必然包含相邻两个顶点的其中之一, 因而也是原图的顶点覆盖。

定义 5.4 (裂点操作) 将有向图 **D** 中的每个顶点 v 替换为一条有向边 v^-v^+ , 使得 v^+u^- 是一条有向边当且仅当 $vu \in E(\mathbf{D})$ 。

在裂点操作操作完成以后, 每个顶点至多有一条入边或一条出边。因此, 图中的一条经过边 v^-v^+ 的环对应于原图一条经过顶点 v 的环, 而一条环经过 v^- 或 v^+ 当且仅当它经过有向边 v^-v^+ 。由此可见, 若 **D** 经过裂点操作以后得到有向图 **D'**, 那么图 **D** 的最小反馈点集大小等于图 **D'** 的最小反馈点集大小, 也等于图 **D'** 的最小反馈边集大小。

相比无向图而言, 控制有向图的最大度相对简单。采用以下归约思想, 足以确定最大度数为 3 时 DFVS 问题的 NP-难性。注意到, 裂点操作能够将顶点 v 分裂为两个顶点 v^- 和 v^+ , 并且有一条从 v^- 到 v^+ 的有向边。因此, 裂点操作能够分离顶点 v 的入邻居和出邻居, 即 v^- 有入邻居 u^+ 当且仅当 v 有入邻居 u ; v^+ 有出邻居 u^- 当且仅当 v 有出邻居 u 。如果 v^- 有超过两个人邻居, 那么可以进一步将 v^- 分裂为多个顶点, 使得每个顶点恰好连接 v^- 的一条入边; 同理可对顶点 v^+ 采取类似的处理方式。这种归约同时适用于 DFVS 问题和 DFAS 问题, 并且能够保证最终得到的图所有顶点的度相同。

定义 5.5 (多重裂点操作) 给定有向图 **D**, 对每个顶点 v 的入邻居和出邻居进行编号

$$\begin{cases} N^-(v) = \{ v'_1, v'_2, \dots, v'_{d^-} \}; \\ N^+(v) = \{ v''_1, v''_2, \dots, v''_{d^+} \}, \end{cases}$$

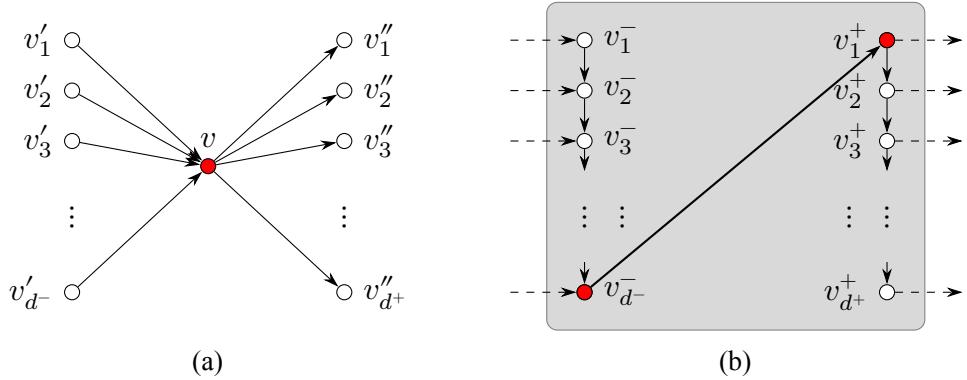


图 5-2 多重裂点操作示意图。(a) 一个入度 $\deg^-(v) = d^-$ 且出度 $\deg^+(v) = d^+$ 的顶点 v 。红色顶点表示 v , 白色顶点为 v 的邻居, 其中 $N^-(v) = \{v'_1, v'_2, \dots, v'_{d^-}\}$ 并且 $N^+(v) = \{v''_1, v''_2, \dots, v''_{d^+}\}$ 。(b) 顶点 v 的替换组件。将 v 替换为一条长度为 $\deg(v) = d^- + d^+$ 的路 $\mathbf{P}(v)$ 。

其中 $d^- = \deg^-(v)$ 且 $d^+ = \deg^+(v)$ 。然后将顶点 v 替换为一条长度为 $d = d^- + d^+$ 的路

$$\mathbf{P}(v) = (u_1^-, v_2^-, \dots, v_{d^-}^-, v_1^+, v_2^+, \dots, v_{d^+}^+),$$

使得 $v_i^+ u_j^-$ 是一条有向边当且仅当 $v = u_i'$ 且 $u = v_j''$ (参见图5-2)。

引理 5.6 设 \mathbf{D}' 是将有向图 \mathbf{D} 进行多重裂点操作得到的有向图。那么, 图 \mathbf{D} 的最小反馈点集大小与图 \mathbf{D}' 的最小反馈点集大小相同, 也与图 \mathbf{D}' 的最小反馈边集大小相同。

证明: 注意到对于任意顶点 $v \in V(\mathbf{D})$, 从路 $\mathbf{P}(v)$ 中移除顶点 v_1^+ 后, 没有环能经过这条路中任何剩余顶点。因此, \mathbf{D}' 的最小反馈点集至多包含 $\mathbf{P}(v)$ 中的一个顶点。此外, 如果 \mathbf{D}' 的最小反馈点集包含来自路 $\mathbf{P}(v)$ 的顶点, 总可以将其替换为 v_1^+ 。现在证明 $S \subseteq V(\mathbf{D})$ 是实例 \mathcal{I} 的最小反馈点集当且仅当

$$S' = \{v_1^+ : v \in S\}$$

是实例 \mathcal{I}' 的最小反馈点集。

首先证明必要性: 若 S 是图 \mathbf{D} 的最小反馈点集, 则 S' 是图 \mathbf{D}' 的最小反馈点集。采用反证法, 若 \mathbf{C}' 是 $\mathbf{D}' - S'$ 中的环。由于路 $\mathbf{P}(\cdot)$ 都不包含环, 因此 \mathbf{C}' 必须经过多条路。按顺序列出 \mathbf{C}' 经过的所有路

$$\mathbf{P}(v_1), \mathbf{P}(v_2), \dots, \mathbf{P}(v_\ell),$$

并且令 $v_{\ell+1} = v_1$ 。

注意到, 如果 \mathbf{C}' 经过了路 $\mathbf{P}(v)$ 上的顶点, 那么 \mathbf{C}' 必然经过了有向边 $v_{\deg^-(v)}^- v_1^+$ 。根据多重裂点操作规则, 有 $v_i v_{i+1} \in E(\mathbf{D})$ ($i \in [\ell]$)。因此, 可以得到 \mathbf{D} 的一个 ℓ 长环

$$\mathbf{C} = (v_1, v_2, \dots, v_\ell).$$

由于 S 是图 \mathbf{D} 的最小反馈点集, 它至少包含 \mathbf{C} 上的一个顶点, 记作 v , 但这样就说明 $v_1^+ \in S'$, 矛盾。

其次证明充分性: 若 S' 是图 \mathbf{D}' 的最小反馈点集, 则 S 是图 \mathbf{D} 的最小反馈点集。采用反证法, 若 \mathbf{C} 是 $\mathbf{D} - S$ 中的一个环。设 uv 是 \mathbf{C} 上的一条有向边。并且假设 v 是 u 的第 i 个出邻居, u 是 v 的第 j 个入邻居。根据多重裂点操作规则, 可知 $u_i^+ v_j^-$ 是 \mathbf{D}' 中的一条有向边。对 \mathbf{C} 上的所有有向边 uv , 替换为路径

$$u_{\deg^-(u)}^- u_1^+ u_2^+ \cdots u_i^+ v_j^- v_{j+1}^- \cdots v_{\deg^-(v)}^-.$$

依次拼接便得到 \mathbf{D}' 中的环 \mathbf{C}' 。由于 S' 是图 \mathbf{D}' 的最小反馈点集, 它至少包含 \mathbf{C}' 上的一个顶点, 但这样就说明 $v \in S$, 矛盾。

由于经过路 $\mathbf{P}(v)$ 的环一定经过有向边 $v_{\deg^-(v)}^- v_1^+$, 因此使用同样的论证方式, 也可以证明 \mathbf{D} 和 \mathbf{D}' 的最小反馈边集大小相同。引理得证。 ■

引理 5.7 当 $\Delta = 3$ 且 $\Delta^* = 1$ 时, DFVS 问题和 DFAS 问题均是 NP-难的。

证明: 设 \mathbf{D} 为有向图。对 \mathbf{D} 中每个顶点的邻居任意编号, 然后对 \mathbf{D} 进行多重裂点操作得到有向图 \mathbf{D}' 。根据构造, \mathbf{D}' 中的任何一个顶点都恰好在某一条路 $\mathbf{P}(v)$ 中 ($v \in V(\mathbf{D})$)。不难发现, 每个顶点的入度和出度恰有一个为 1, 一个为 2, 所以有 $\Delta(\mathbf{D}') = 3$ 且 $\Delta^*(\mathbf{D}') = 1$ 。

再根据引理 5.6, 由于 DFVS 问题和 DFAS 问题均为 NP-难的, 故满足 $\Delta = 3$ 且 $\Delta^* = 1$ 的有向图上 DFVS 问题和 DFAS 问题亦是 NP-难的。引理得证。 ■

在 Garey 和 Johnson 的专著中^[195], 提及了入度或出度不超过 2 的有向图上 DFVS 问题是 NP-难的结论。然而, 专著中没有提供证明、提示或任何参考文献作为支持。尽管不知道他们的论证技术是什么, 但本文采用的方法似乎并非他们所考虑的。这是因为他们对 DFVS 问题和 DFAS 问题关于度有界图类上的 NP-难结论并不相同 (详见文献 [195] 附录 [GT7] 和 [GT8])。作者仅表明了对于入度或出度不超过 3 的有向图上 DFAS 问题是 NP-难的。

定理 5.1 当 $\Delta \leq 2$ 或 $\Delta^* \leq 0$ 时, DFVS 可以在线性时间内求解。当 $\Delta \geq 3$, $\Delta^* \geq 1$ 且均为常数时, DFVS, DFAS 问题均是 NP-完全的。

证明: 一方面, 当 $\Delta \leq 2$ 或 $\Delta^* = 0$ 时, 可以简单高效地解决反馈集问题。首先, 反复删除所有没有入边或出边的顶点: 这些顶点不会出现在任何有向环中。

那么, 剩余的图由不相交的环组成, 只需从每个环中取一个顶点或边即可。因此, 可以在线性时间内求解 DFVS 问题或 DFAS 问题。

另一方面, DFVS 问题和 DFAS 问题均为 NP 问题。又根据引理5.7, 可知当 $\Delta = 3$ 且 $\Delta^* = 1$ 时, DFVS 问题和 DFAS 问题均是 NP-完全的。对于 $\Delta > 3$ 或 $\Delta^* > 1$ 的情况, 可以额外构造一个无环的强连通分量, 满足最大度和定向最大度分别为 Δ 和 Δ^* , 定理得证。 ■

平面有向图的情况则要复杂许多, 这是因为引理5.7所使用的归约方式无法保证平面性。只有当顶点的边按照某种特定的方式嵌入平面时, 才能够保持平面性。粗略地讲, 如果有向图的嵌入可以保证顶点 v 的邻边顺时针排列时, 入边和出边是连续的, 那么上述归约算法能够保持平面性。

现给出形式化的论述。考虑 d 度顶点 v , 有向图的平面嵌入决定了 v 所有邻边的顺时针排列顺序 (e_1, e_2, \dots, e_d) 。注意到, (e_1, e_2, \dots, e_d) 和 $(e_2, e_3, \dots, e_d, e_1)$ 视为同样的顺时针排列顺序。定义函数

$$\varphi_v: \{e_i : i \in [d]\} \rightarrow \{\ominus, \oplus\}$$

使得若 e_i 是 v 的入边, 则 $\varphi_v(e) = \ominus$, 若 e_i 是 v 的出边, 则 $\varphi_v(e) = \oplus$ 。

定义 5.8 (嵌入方向) 给定有向图 **D** 的平面嵌入, 设 d 度顶点 v 所有邻边的顺时针排列顺序为 (e_1, e_2, \dots, e_d) 。向量

$$(\varphi_v(e_1), \varphi_v(e_2), \dots, \varphi_v(e_d))$$

称为顶点 v 的嵌入方向。

定义 5.9 (双极嵌入) 如果有向图 **D** 的平面嵌入使得任意顶点 v 都有嵌入方向 $(\oplus, \oplus, \dots, \oplus, \ominus, \ominus, \dots, \ominus)$, 则称此嵌入为双极的 (**bipolar**)。

引理 5.10 若有向图 **D** 能够双极嵌入到平面上, 那么 SFVS(P) 问题实例 $\mathcal{I} = (\mathbf{D}, k)$ 能够在多项式时间内求解。

证明: 设 **D** 为有向图。对每个 d 度顶点 v 的入度邻居和出度邻居编号, 使得

$$v'_{d^-}, v'_{d^-1}, \dots, v'_1, v''_1, v''_2, \dots, v''_{d^+}$$

是一个顺时针排序, 其中 $d^- = \deg^-(v)$ 且 $d^+ = \deg^+(v)$ 。

$$\begin{cases} N^-(v) = \{v'_1, v'_2, \dots, v'_{d^-}\}; \\ N^+(v) = \{v''_1, v''_2, \dots, v''_{d^+}\}, \end{cases}$$

然后按照此编号, 对 **D** 进行多重裂点操作得到有向图 **D'**。不难发现 **D'** 是一个平面图。根据引理5.6, 图 **D** 的最小反馈点集大小与图 **D'** 的最小反馈边集大小相等。

又因为 \mathbf{D}' 是平面图，而 DFAS(\mathbf{P}) 能在多项式时间内求解，所以实例 \mathcal{I} 亦能在多项式时间内求解。引理得证。 ■

值得一提的是，引理5.10中的算法不需要额外给定平面图的双极嵌入。事实上，只需首先应用多重分裂操作，然后计算任意平面嵌入即可，原因在于满足 $\Delta^*(\mathbf{D}) \leq 1$ 的有向图 \mathbf{D} 的任何平面嵌入总是双极的。正是基于这一性质，可以得到以下推论。

推论 5.11 若有向图 \mathbf{D} 满足 $\Delta^*(\mathbf{D}) \leq 1$ ，那么 SFVS(\mathbf{P}) 问题实例 $\mathcal{I} = (\mathbf{D}, k)$ 能够在多项式时间内求解。

推论 5.12 若有向图 \mathbf{D} 满足 $\Delta(\mathbf{D}) \leq 3$ ，那么 SFVS(\mathbf{P}) 问题实例 $\mathcal{I} = (\mathbf{D}, k)$ 能够在多项式时间内求解。

接下来说明推论5.11和推论5.12给出的上界是紧的。如果有向图 \mathbf{D} 中每个顶点 v 的入度和出度均为 3，则称 \mathbf{D} 是 3-正则 (**3-Regular**) 的。由于无向 3-正则平面图上的顶点覆盖问题是 NP-难的^[181]，那么通过倍边操作进行归约可以得出 3-正则平面图上 DFVS 问题亦是 NP-难的。一个自然的尝试是类似于 Speckenmeyer 的方法设计一个平面的组件逐个替换顶点。然而，这一替换过程必须符合每一个顶点的嵌入方向，但这并非易事。特别地，当某顶点的嵌入方向是 $(\oplus, \ominus, \oplus, \ominus, \oplus, \ominus)$ 时，通过倍边操作得到的有向图上对应顶点的嵌入方向一定不是双极嵌入的。

另外，3-正则平面图上顶点的还存在两种对称的嵌入方向

$$(\ominus, \ominus, \oplus, \oplus, \ominus, \oplus) \text{ 或 } (\oplus, \oplus, \ominus, \ominus, \oplus, \ominus).$$

如果 3-正则平面有向图 \mathbf{D} 中每个顶点的嵌入方向是上述二者之一，则称此嵌入为非正规的 (**Irregular**)。有趣的是，对于通过加倍操作得到的 3-正则平面有向图，我们总是可以为其构造一个非正规嵌入。注意到，对于一般的 3-正则平面有向图而言此结论不一定成立。事实上，Whitney 嵌入定理指出三连通平面图有本质上唯一的嵌入^[229]。本文基于无向图的线性森林 (Linear Forest) 分解^[230]，调节并交换某些点对间两条相反方向有向边的顺序，得到了以下引理，这里线性森林是一些不相交路径的并。

引理 5.13 设 \mathbf{G} 为 3-正则平面无向图。存在由 \mathbf{G} 经过倍边操作得到的 3-正则平面有向图 \mathbf{D} ，使得 \mathbf{D} 能非正规嵌入到平面中，并且此嵌入可以在多项式时间内得到。

证明：首先，在多项式时间内找到图 \mathbf{G} 的平面嵌入并固定下来。根据 Duncan 等人^[230]的研究， \mathbf{G} 可以被两个边不相交的线性森林 \mathbf{F} 和 \mathbf{F}' 覆盖，即满足 $V(\mathbf{F}) = V(\mathbf{F}') = V(\mathbf{G})$ 并且 $(E(\mathbf{F}), E(\mathbf{F}'))$ 是边集 $E(\mathbf{G})$ 的一个划分。并且此分解能在多项式时间内找到。

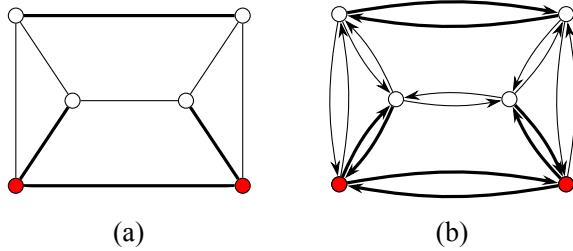


图 5-3 基于线性森林分解的非正规平面嵌入示意图。(a) 一个 3-正则图 \mathbf{G} 。

\mathbf{G} 可以被两个边不相交的线性森林 \mathbf{F} 覆盖, 其中一个线性森林 \mathbf{F} 的边集用粗线表示。红色顶点连接了两条 \mathbf{F} 中的边, 白色顶点连接了一条 \mathbf{F} 中的边。(b) 图 G 基于线性森林分解的非正规平面嵌入。图中粗箭头构成了顺时针的 2 长环, 细箭头构成了逆时针的 2 长环, 其中红色顶点的嵌入方向是 $(\ominus, \ominus, \oplus, \oplus, \ominus, \oplus)$, 而白色顶点的嵌入方向是 $(\oplus, \oplus, \ominus, \ominus, \oplus, \ominus)$ 。

现按照下述方式完成倍边操作 (参见图5-3)。对图 \mathbf{G} 中的每条边 e , 若 $e \in \mathbf{F}$, 则按照顺时针顺序添加两条方向相反的有向边; 否则按照逆时针顺序添加两条方向相反的有向边。令 \mathbf{D} 表示倍边操作后得到的 3-正则平面有向图。

下证此时的平面嵌入是非正规的。注意到, 线性森林 \mathbf{F} 恰好覆盖每个顶点的一条或两条邻居边。特别地, 如果顶点 v 正好与 \mathbf{F} 中的一条边相连, 则 v 的嵌入方向是 $(\oplus, \oplus, \ominus, \ominus, \oplus, \ominus)$; 否则是 $(\ominus, \ominus, \oplus, \oplus, \ominus, \oplus)$ 。引理得证。 ■

根据非正规嵌入的结构, 每个顶点恰好有两条顺时针相连的入边和两条顺时针相连的出边, 故可将它们分别进行“捆绑”然后构造一个替换组件。基于这一思想, 便可以证明本节的核心结论。

引理 5.14 当 $\Delta = 4$ 且 $\Delta^* = 2$ 时, DFVS 问题是 NP-难的。

证明: 设 $\mathcal{I} = (\mathbf{G}, k)$ 是顶点覆盖问题的实例。根据引理5.13, 可以将 \mathbf{G} 进行倍边操作得到 3-正则平面有向图 \mathbf{D} , 使得 \mathbf{D} 存在非正规平面嵌入, 并且此平面嵌入可在多项式时间内得到。考虑 \mathbf{D} 中嵌入方向为 $(\ominus, \ominus, \oplus, \oplus, \ominus, \oplus)$ 的 6 度顶点 v , 将其替换为组件 $\mathbf{H}(v)$ (参见图5-4)。组件 $\mathbf{H}(v)$ 共包含 11 个顶点。

$$V(\mathbf{H}(v)) = \{v_{12}^-, v_{12}^+, v_3^-, v_3^+, w_1^-, w_1^+, w_2, w_3, w_4, w_5, w_6\},$$

以及 16 条有向边, 其中

- $v_3^- v_3^+$ 是一条有向边;
- $(v_{12}^-, w_1^-, w_1^+, v_{12}^+)$ 是一条长度为 4 的有向路径;
- (w_2, w_4, w_4) , (w_4, w_6, w_5) , (w_1^-, v_3^+, w_2) , $(v_3^- w_1^+ w_6)$ 是四个有向三角形。

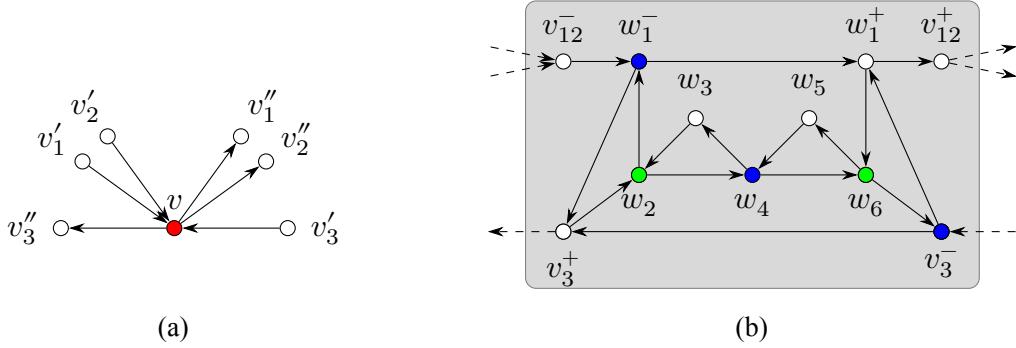


图 5-4 证明 4 度平面图上 DFVS 问题 NP-难的替换组件。(a) 一个嵌入方向为 $(\ominus, \ominus, \oplus, \oplus, \ominus, \oplus)$ 的顶点 v 。红色顶点表示 v , 白色顶点为 v 的邻居, 其中 $N^-(v) = \{v'_1, v'_2, v'_3\}$ 且 $N^+(v) = \{v''_1, v''_2, v''_3\}$ 。(b) 顶点 v 的替换组件。连向原图其他顶点的边使用虚线表示并按照平面嵌入的顺时针顺序连接。注意到, 任意最小反馈点集至少包含每个组件中的两个顶点, (因为如果恰好包含两个顶点, 则这两个顶点必须是 w_2 和 w_6)。并且最多只会包含组件中的三个顶点 (因为移除顶点 w_1^-, w_4, w_3^- 后, 组件中任何余下的顶点都不在环中)。

按照顶点 v 的嵌入方向, 设其邻居边为

$$v'_1v, v'_2v, vv''_1, vv''_2, v'_3v, vv''_3.$$

那么添加有向边

$$v'_1v_{12}^-, v'_2v_{12}^-, v_{12}^+v''_1, v_{12}^+v''_2, v'_3v_3^-, v_3^+v''_3.$$

对于 **D** 中嵌入方向为 $(\oplus, \oplus, \ominus, \ominus, \oplus, \ominus)$ 的 6 度顶点 u , 则可以通过类似地方式替换为组件 **H**(u)。这里 **H**(u) 是通过将图5-3中的组件中的所有边调换方向得到的组件。由于每次替换都会将一个 6 度顶点替换为一些新的 4 度顶点, 而不会改变其他顶点的度数和嵌入方向。因此, 可以通过反复迭代 n 次替换, 得到最终的有向图 **D'**, 使得 $\Delta(\mathbf{D}') = 4$ 且 $\Delta^*(\mathbf{D}') = 2$ 。

令 $n = |V(\mathbf{G})| = |V(\mathbf{D})|$ 以及 $k' = 2n + k$ 。只需证明 DFVS(P) 问题实例 $\mathcal{I} = (\mathbf{D}, k)$ 有解当且仅当 DFVS(P) 问题实例 $\mathcal{I}' = (\mathbf{D}', k')$ 即可。

首先考虑必要性: 若 \mathcal{I} 有解, 则 \mathcal{I}' 有解。设 S 是实例 \mathcal{I} 的解, 下证集合

$$S' = \{w_2, w_6 : v \in V(\mathbf{D}) \setminus S\} \cup \{w_1^-, w_4, v_3^- : v \in S\}$$

是实例 \mathcal{I}' 的解。一方面，有大小关系

$$|S'| = 2|V(\mathbf{D}) - S| + 3|S| = 2|V(\mathbf{D})| + |S| = 2n + k.$$

另一方面，若 $\mathbf{D}' - S'$ 中存在环 \mathbf{C}' 。由于删除了集合 $\{w_1^-, w_4, v_3^-\}$ 之后，没有有向环经过组件 $\mathbf{H}(v)$ 中的任何顶点。所以环 \mathbf{C}' 包含了 $\mathbf{H}(v)$ 中顶点，当且 $\mathbf{H}(v)$ 中恰有两个顶点属于 S' 。按顺序列出 \mathbf{C}' 经过的所有组件

$$\mathbf{H}(v_1), \mathbf{H}(v_2), \dots, \mathbf{H}(v_\ell),$$

不难发现， $(v_1, v_2, \dots, v_\ell)$ 构成了 $\mathbf{D} - S$ 中的环，这与 S 是 \mathcal{I} 的解矛盾。因此， S' 必然是 \mathcal{I}' 的解。

其次考虑充分性：若 \mathcal{I}' 有解，则 \mathcal{I} 有解。设 S' 是实例 \mathcal{I}' 的解，下证集合

$$S = \{v : |S' \cap V(\mathbf{H}(v))| \geq 3\}$$

是实例 \mathcal{I} 的解。一方面，由于 (v_1^-, w_2, v_3^+) 和 (v_3^-, v_1^+, w_6) 是组件 $\mathbf{H}(v)$ 中点不相交的有向三角形，所以 S' 至少包含每个组件中的两个顶点。故可得出大小关系

$$|S'| \geq 3|S| + 2|V(\mathbf{D}) - S| = 2n + |S|,$$

这意味着 $|S| \leq k$ 。另一方面，设 $\mathbf{D} - S$ 中存在环 \mathbf{C} 。考虑 \mathbf{C} 中的一条子路径 (v'_i, v, v''_j) ($i, j \in [3]$)，这里允许 $v'_i = v''_j$ 的情况出现。由于 $v \notin S$ ，那么 S' 恰好包含了组件 $\mathbf{H}(v)$ 中的两个顶点 w_2 和 w_6 。所以，无论 i 和 j 的取值如何， $\mathbf{H}(v)$ 中存在一条从 v_i^- 到 v_j^+ 的路。具体地，以下四条路均存在于 $\mathbf{D}' - S'$ 中

- $(v_{12}^-, w_1^-, w_1^+, v_{12}^+)$;
- (v_{12}^-, w_1^-, v_3^+) ;
- (v_3^-, w_1^+, v_{12}^+) ;
- (v_3^-, v_3^+) 。

综上所述，引理得证。 ■

定理 5.2 当 $\Delta \leq 3$ 或 $\Delta^* \leq 1$ 时，DFVS(P) 可以在多项式时间内求解。当 $\Delta \geq 4$ ， $\Delta^* \geq 2$ 且均为常数时，DFVS(P) 是 NP-完全的。

证明：一方面，根据推论5.11和推论5.12，当 $\Delta \leq 3$ 或 $\Delta^* \leq 1$ 时，DFVS(P) 可以在多项式时间内求解。另一方面，DFVS 问题属于 NP 问题，又根据引理5.14，当 $\Delta = 4$ ， $\Delta^* = 2$ 且均为常数时，DFVS(P) 是 NP-完全的。对于 $\Delta > 4$ 或 $\Delta^* > 2$ 的情况，可以额外构造一个无环的强连通分量，满足最大度和定向最大度分别为 Δ 和 Δ^* ，定理得证。 ■

5.3 常数关键点的子集反馈集问题

本节将研究关键点数量 τ 为常数时, 子集反馈集问题的计算复杂性的分类, 主要结论如下。

定理 5.15 当关键集大小 $\tau \leq 1$ 时, DSFAS 问题、DSFAS 问题和 R-DSFVS 问题均能在多项式时间内求解。当关键集大小 τ 是大小至少为 2 的常数时, DSFAS 问题、DSFAS 问题和 R-DSFVS 问题均为 NP-完全的。

注意到, 对于非限制版本的反馈点集和边子集反馈边集问题而言, 解集大小 $k \geq \tau$ 时, 一定是有解, 因为关键集自身就是一个大小为 τ 的 T -反馈集。所以这六类问题均可在 $\mathcal{O}(n^\tau) = n^{\mathcal{O}(1)}$ 时间通过穷举搜索计算。在剩下两类非限制版本中, Xiao 和 Nagamochi^[160] 证明了无向图上子集反馈边集问题能在多项式时间内求解。所以, 在六种所有非限制版本的子集反馈集问题中, 仅需考虑有向图上子集反馈边集问题, 即 DSFAS 问题。

事实上, DSFAS 问题和有向图上的两类限制版子集反馈集问题有着紧密的归约关系。具体地, DSFAS 问题、R-DSFVS 问题以及 R-DASFAS 问题两两之间均存在多项式时间归约, 并且保持解集大小和关键集大小。可采用文献 [5] 和文献 [162] 中使用归约技术完成论证。

引理 5.16 R-DSFVS 问题的实例 $\mathcal{I} = (\mathbf{D}, T, k)$ 可以在多项式时间内归约到 DSFAS 问题的实例 $\mathcal{I}' = (\mathbf{D}', T', k')$, 满足 $k' \leq k$ 且 $\tau' \leq \tau$ 。

证明: 设 $\mathcal{I} = (\mathbf{D}, T, k)$ 为 R-DSFVS 问题的实例, 其中 $n = |V(\mathbf{D})|$ 以及 $\tau = |T|$ 。对 \mathcal{I} 中的每个顶点进行裂点操作得到有向图 \mathbf{D}' , 其中

$$V(\mathbf{D}') = \{v^+, v^- : v \in V(\mathbf{D})\}.$$

然后定义关键点集 $T' = \{t^+ : t \in T\}$ 。最后令 $k' = k$, 得到 DSFAS 问题的实例 $\mathcal{I}' = (\mathbf{D}', T', k')$ 。不难发现, $n' = |V(\mathbf{D}')| = 2n$ 且 $\tau' = |T'| = \tau$ 。

下证实例 \mathcal{I} 有解当且仅当 \mathcal{I}' 有解。注意到在图 \mathbf{D}' 中, 一个环 \mathcal{C}' 经过顶点 v^+ 当且仅当经过有向边 v^-v^+ 。所以,

$$\mathbf{C} = (v_1, v_2, \dots, v_\ell)$$

是 \mathbf{D} 中的 T -环, 当且仅当

$$\mathbf{C}' = (v_1^-, v_1^+, v_2^-, v_2^+, \dots, v_\ell^-, v_\ell^+)$$

是 \mathbf{D}' 中的 T' -环。 \mathbf{D} 中任意点子集 S 都一一对应 \mathbf{D}' 的点集 $S' = \{v^-v^+ \in E(\mathbf{D}') : v \in S\}$ 。不难发现, 经过有向边 u^+v^- 的环必然经过有向边 v^-v^+ , 因此, 不妨设 S'

仅包含了形如 v^-v^+ 的有向边。所以 $\mathbf{D} - S$ 是 T -无环的当且仅当 $\mathbf{D}' - S'$ 是 T' -无环的。又因为 $|S| = |S'|$, 所以 \mathcal{I} 有解当且仅当 \mathcal{I}' 有解。

综上所述, 引理得证。 ■

引理 5.17 DSFAS 问题的实例 $\mathcal{I} = (\mathbf{D}, T, k)$ 可以在多项式时间内归约到 R-DASFAS 问题的实例 $\mathcal{I}' = (\mathbf{D}', T', k')$, 满足 $k' \leq k$ 且 $\tau' \leq \tau$ 。

证明: 设 $\mathcal{I} = (\mathbf{D}, T, k)$ 为 DSFAS 问题的实例, 其中 $n = |V(\mathbf{D})|$ 以及 $\tau = |T|$ 。对 \mathcal{I} 中的每个关键点进行裂点操作得到有向图 \mathbf{D}' , 其中

$$V(\mathbf{D}') = \{v' : v \in V(\mathbf{D}) \setminus T\} \cup \{t^+, t^- : t \in T\}.$$

然后定义关键边集 $T' = \{t^+t^- : t \in T\}$ 。最后令 $k' = k$, 得到 R-DASFAS 问题的实例 $\mathcal{I}' = (\mathbf{D}', T', k')$ 。不难发现, $n' = |V(\mathbf{D}')| = n + \tau$ 并且 $\tau' = |T'| = \tau$ 。

下证实例 \mathcal{I} 有解当且仅当 \mathcal{I}' 有解。为方便表述, 对于 \mathbf{D}' 的非关键点 v , 令 $v^+ = v^- = v'$ 。注意到在图 \mathbf{D}' 中, 一个环 \mathcal{C}' 经过顶点 v^+ 当且仅当经过有向边 v^-v^+ 。所以,

$$\mathbf{C} = (v_1, v_2, \dots, v_\ell)$$

是 \mathbf{D} 中的 T -环, 当且仅当由 ℓ 段子路径拼接得到的环

$$\mathbf{C}' = (\mathbf{C}'_1, \mathbf{C}'_2, \dots, \mathbf{C}'_\ell)$$

是 \mathbf{D}' 中的 T' -环, 其中对于顶点 v_i ($i \in [\ell]$), 若为非关键点, 则 $\mathbf{C}'_i = (v'_i)$; 若为关键点, 则 $\mathbf{C}'_i = (v_i^-, v_i^+)$ 。 \mathbf{D} 中任意边子集 S 都一一对应 \mathbf{D}' 的边子集 $S' = \{v^+u^- \in E(\mathbf{D}') : v^+u^- \in S\}$ 。所以 $\mathbf{D} - S$ 是 T -无环的当且仅当 $\mathbf{D}' - S'$ 是 T' -无环的。又因为 $|S| = |S'|$, 所以 \mathcal{I} 有解当且仅当 \mathcal{I}' 有解。

综上所述, 引理得证。 ■

引理 5.18 R-DASFAS 问题的实例 $\mathcal{I} = (\mathbf{D}, T, k)$ 可以在多项式时间内归约到 R-DSFVS 问题的实例 $\mathcal{I}' = (\mathbf{D}', T', k')$, 满足 $k' \leq k$ 且 $\tau' \leq \tau$ 。

证明: 设 $\mathcal{I} = (\mathbf{D}, T, k)$ 为 R-DASFAS 问题的实例, 其中 $n = |V(\mathbf{D})|$, $m = |E(\mathbf{D})|$ 以及 $\tau = |T|$ 。现构造 \mathbf{D} 的“有向线图” \mathbf{D}' 如下。对 \mathcal{I} 中的每条边 e , 引入一个顶点 x'_e , 得到顶点集

$$V(\mathbf{D}') = \{x'_e : e \in E(\mathbf{D})\}.$$

若两条边首尾相连, 则对应顶点按照相同方向连一条有向边, 得到有向边集

$$E(\mathbf{D}') = \{x'_{uv}x'_{vw} : uv, vw \in E(\mathbf{D})\}.$$

然后定义关键点集 $T' = \{x'_e : e \in T\}$ 。最后令 $k' = k$, 得到 R-DSFVS 问题的实例 $\mathcal{I}' = (\mathbf{D}', T', k')$ 。不难发现, $n' = |V(\mathbf{D}')| = m$ 并且 $\tau' = |T'| = \tau$ 。

下证实例 \mathcal{I} 有解当且仅当 \mathcal{I}' 。根据构造,

$$\mathbf{C} = (v_1, v_2, \dots, v_\ell)$$

是 \mathbf{D} 中的 T -环, 当且仅当

$$\mathbf{C}' = (x'_{v_1 v_2}, x'_{v_1 v_2}, \dots, x'_{v_\ell v_1})$$

是 \mathbf{D}' 中的 T' -环。 \mathbf{D} 中任意边子集 S 都一一对应 \mathbf{D}' 的点子集 $S' = \{x'_e \in V(\mathbf{D}') : e \in S\}$ 。所以 $\mathbf{D} - S$ 是 T -无环的当且仅当 $\mathbf{D}' - S'$ 是 T' -无环的。又因为 $|S| = |S'|$, 所以 \mathcal{I} 有解当且仅当 \mathcal{I}' 有解。

综上所述, 引理得证。 ■

根据引理5.16、引理5.17以及引理5.18, DSFAS 问题、R-DSFVS 问题以及 R-DASFAS 问题在关键集大小为常数时, 它们的 NP-难性是一致的。因此, 接下来只需讨论三个问题其中之一的 NP-难性即可。

Even 等人^[5]系统性地对 DSFAS 问题和有向节点多割问题 (Directed Node Multicut Problem) 的计算复杂性和可近似性展开了研究。有向多割问题是关于图连通性的一个核心问题之一。给定一个有向图 \mathbf{D} 和 τ 个有向关键点对 T , 有向多割问题旨在找到大小不超过 k 的边割能够割开所有点对 (详见附录附录 A)。有向多割问题和 DSFAS 问题非常相似, Even 等人^[5]说明了可以通过将 DSFAS 问题的每个关键点执行裂点操作归约到有向多割问题上。由于 $\tau = 1$ 时有向多割问题使用最大流算法进行求解, 所以 DSFAS 问题在 $\tau = 1$ 时候是多项式时间可解的。

当 $\tau \geq 2$ 时, Even 等人^[5]引用了文献 [231] 证明有向多割问题 NP-难性的归约技术, 以阐述 DSFAS 问题也是 NP-难的。然而, 文献 [231] 和文献 [5] 的论述均不完整。值得强调的是, 一个基本的尝试是将有向多割问题归约到 DSFAS 问题或 R-DASFAS 问题: 为每个关键点对添加不可删的反向边, 使得点对之间的有向路都对应了经过反向边的环。然而, 添加反向边将带来了连通性上的增强。例如, 考虑有向多割问题中的点对 (t_1, s_1) 和 (t_2, s_2) , 添加反向边 $s_1 t_1$ 和 $s_2 t_2$ 之后, 从 t_1 到 s_2 的有向路与从 t_2 到 s_1 的有向路也会构成一条经过 $s_1 t_1$ 的有向环; 从 t_2 到 s_1 的有向路与从 t_1 到 s_2 的有向路也会构成一条经过 $s_2 t_2$ 的有向环, 这将破坏两个问题的等价性。这是一个十分容易忽视的障碍。

为了解决上述问题, 本文采用无向图上的节点多路割问题 (Node Multiway Cut Problem) 归约到有向节点多割问题 (Directed Node Multicut Problem) 的技术路线^[232,233], 通过证明 R-DSFVS 问题的 NP-难性解决了这一棘手的难题。给定无

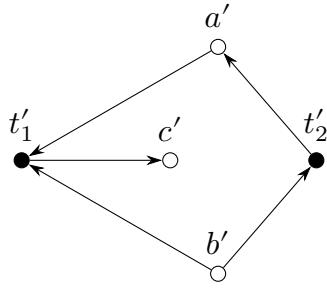


图 5-5 论证 2 个关键点的 R-DSFVS 问题 NP-难性的归约示意图。黑色顶点 $\{t'_1, t'_2\}$ 表示 R-DSFVS 问题实例 \mathcal{I}' 中的关键点，白色顶点 $\{a', b', c'\}$ 表示节点多路割问题实例中关键点在 \mathcal{I}' 中对应的顶点。

向图 \mathbf{G} 和大小为 τ 的关键集 T ，节点多路割问题旨在找到大小不超过 k 的点割能够割开任意两个关键点。

定理 5.19 当关键集大小 τ 为常数且至少为 2 时，R-DSFVS 问题是 NP-难的。

证明：由于添加孤立的关键点不影响实例的 NP-难性。所以，只需证明 $\tau = 2$ 时，R-DSFVS 问题是 NP-难的即可。使用包含 3 个关键点的无向图上的节点多路割问题进行归约。设 $\mathcal{I} = (\mathbf{G}, T, k)$ 为节点多路割问题，其中 $T = \{a, b, c\} \subseteq V(\mathbf{G})$ 。

现在构造 R-DSFVS 问题的实例 $\mathcal{I}' = (\mathbf{D}', T', k')$ (参见图 5-5)。首先对 \mathbf{G} 应用倍边操作得到有向图 \mathbf{D} ，其中

$$V(\mathbf{D}) = \{v' : v \in V(\mathbf{G})\},$$

且

$$V(\mathbf{D}) = \{v'u', u'v' : vu \in E(\mathbf{G})\}.$$

引入 2 个关键点 t'_1, t'_2 ，使得 (b', t'_1, c') 和 (b', t'_2, a', t'_1) 是两条路。然后对 a', b', c' 分别赋值 k 份拷贝，得到 $\{a'_i, b'_i, c'_i : i \in [k]\}$ ，使得 a'_i, b'_i, c'_i ($i \in [k]$) 的出入邻居分贝与 a', b', c' 相同。这便得到了有向图 \mathbf{D}' ，其点集为

$$V(\mathbf{D}') = V(\mathbf{D}) \cup T' \cup \{a'_i, b'_i, c'_i : i \in [k]\},$$

其中 $T' = \{t'_1, t'_2\}$ 。为方便表述，令 $a'_0 = a'$ ， $b'_0 = b'$ ， $c'_0 = c'$ 。有向图 \mathbf{D}' 的边集为

$$\begin{aligned} E(\mathbf{D}') = & E(\mathbf{D}) \cup \{b'_i t'_1, t'_1 c'_i, b'_i t'_2, t'_2 a'_i, a'_i t'_1 : 0 \leq i \leq k\} \cup \\ & \{x'_i v, v x'_i : x \in \{a, b, c\}, i \in [k], v' x' \in E(\mathbf{D})\}. \end{aligned}$$

最后，令 $k' = k$ ，便得到了 R-DSFVS 问题的实例 $\mathcal{I}' = (\mathbf{D}', T', k')$ 。

往证实例 \mathcal{I} 有解当且仅当 \mathcal{I}' 有解。

首先考虑必要性：若 \mathcal{I} 有解，则 \mathcal{I}' 有解。设 S 是实例 \mathcal{I} 的解，下证集合

$$S' = \{v' : v \in V(\mathbf{D}) \cap T\}$$

是实例 \mathcal{I}' 的解。若 $\mathbf{D}' - S'$ 中存在 T' -环 \mathbf{C}' 。根据构造， \mathbf{C}' 包含子路 (a', t'_1, c') , (b', t'_1, c') , (b', t'_2, a') 至少其中之一 ($0 \leq i \leq k$)。所以 $\mathbf{D}' - S'$ 中存在 b' 到 c' 的有向路，或 a' 到 c' 的有向路，或 a' 到 b' 的有向路。这意味着 $\mathbf{G} - S$ 中从 a 到 b 的路，或 a 到 c 的路，或 a 到 b 的有向路，这与 S 是实例 \mathcal{I} 的解矛盾。所以， $\mathbf{D}' - S'$ 中不存在 T' -环。又因为 $|S'| = |S| = k = k'$ ，所以 S' 是实例 \mathcal{I}' 的解。

其次考虑充分性：若 \mathcal{I}' 有解，则 \mathcal{I} 有解。设 S' 是实例 \mathcal{I}' 的解，由于顶点 a' , b' , c' 各有 k 份拷贝，不妨设 S 不包含它们及其任何一份拷贝。下证集合

$$S = \{v : v' \in S'\}$$

是实例 \mathcal{I} 的解。假设 $\mathbf{G} - S$ 中存在以关键点 a, b, c 为端点的路，设为 \mathbf{P} 。若 \mathbf{P} 的端点为 a 和 b ，则 $\mathbf{D}' - S'$ 中存在从 b' 到 a' 的有向路，这说明 $\mathbf{D}' - S'$ 中存在经过 t'_2 的环，矛盾；若 \mathbf{P} 的端点为 a 和 c ，则 $\mathbf{D}' - S'$ 中存在从 a' 到 c' 的有向路，这说明 $\mathbf{D}' - S'$ 中存在经过 t'_1 的环，矛盾；若 \mathbf{P} 的端点为 b 和 c ，则 $\mathbf{D}' - S'$ 中存在从 b' 到 c' 的有向路，这说明 $\mathbf{D}' - S'$ 中存在经过 t'_1 的环，矛盾。因此， $\mathbf{G} - S$ 中三个关键点两两之间没有路。又因为 $|S| = |S'| = k' = k$ ，所以 S 是实例 \mathcal{I} 的解。

综上所述，引理得证。 ■

定理 5.15 当关键集大小 $\tau \leq 1$ 时，DSFAS 问题、DSFAS 问题和 R-DSFVS 问题均能在多项式时间内求解。当关键集大小 τ 是大小至少为 2 的常数时，DSFAS 问题、DSFAS 问题和 R-DSFVS 问题均为 NP-完全的。

至此，我们可以最终得到 DSFAS 问题、R-DSFVS 问题以及 R-DASFAS 问题计算复杂性关于关键集大小的二分定理。最终完成了对所有十二种子集反馈集问题在常数大小关键集条件下的计算复杂性分类工作。

证明：根据引理5.16、引理5.17以及引理5.18，DSFAS 问题、R-DSFVS 问题以及 R-DASFAS 问题在关键集大小为常数时，它们的 NP-难性是一致的。

一方面，而当关键集大小 $\tau \leq 1$ 时，三个问题均可通过流算法在多项式时间内求解。另一方面，DSFAS 问题、DSFAS 问题和 R-DSFVS 问题均属于 NP 问题，又根据引理5.19，当关键集大小 τ 是大小至少为 2 的常数时，三个问题均是 NP-完全的。定理得证。 ■

5.4 本章小结

本章深入研究了图计算问题的一个核心课题——反馈集问题和子集反馈集问题的计算复杂性分类，这两个问题均被认为是极具应用和理论价值的 NP-难问题。

本章借鉴 Speckenmeyer 的替换组件等归约技术，解决了 DFVS 问题、DFAS 问题以及 DFVS(P) 问题的计算复杂性分类难题，其中 DSFVS(P) 问题关于最大度的计算复杂性分类工作是长期未被解决的开放性难题。本章证明了，即使是三度图，有向反馈点集问题依然是 NP-难的，但是如果限制输入图是平面图，则只有当最大度至少为 4 时，有向反馈点集问题才是 NP-难的。但对于无向图而言，将输入图限制为平面图不会改变最大度对反馈点集问题计算复杂性的影响。

此外，本章研究了常数大小关键点集条件下子集反馈集问题。本章采用了新的技术路线，通过无向图上节点多路割问题进行归约，证明了 R-DSFVS 问题在关键集大小至少是 2 的常数时是 NP-难的。进一步地，本章通过建立同时保持关键集和解集大小的多项式时间规约，完备了十二种子集反馈集问题关于关键集大小的计算复杂性分类工作。

第六章 竞赛图上子集反馈点集问题的参数与精确算法

竞赛图是循环赛制或社交网络的常用图模型。本章研究了竞赛图上子集反馈点集问题的参数与精确算法，对现有的算法结果进行了改进。特别地，对于竞赛图上的非限制版子集反馈点集问题，本文提出的参数算法首次突破了 2^k 的计算复杂度瓶颈。

6.1 研究背景与主要贡献

竞赛图是通过对完全图的边定向而得到有向图，其每对顶点间恰有一条有向边。竞赛图的名称源自于对循环赛制的建模^[234]。在包含 n 个队伍的循环赛中，每个队伍都会和其他 $n - 1$ 个队伍完成一场比赛且会决出胜负。在循环赛的有向图模型中，每个队伍用一个顶点表示，而队伍之间的比赛结果通过从获胜队伍指向落败队伍的有向边来表示。可见，一个循环赛对应的有向图恰好是竞赛图。对竞赛图的系统研究可追溯至 1953 年 Landau 对动物社会中的支配关系^[235]。目前，竞赛图在选举理论、社会选择理论中得到了广泛应用^[236,237]。

在本章中，竞赛图上反馈点集和子集反馈点集问题分别简记为 DFVS(T) 问题和 DSFVS(T) 问题。这两个问题在选举系统 (Voting Systems) 和排名聚合 (Rank Aggregation)^[13,238,239] 等应用的推动下，吸引了研究者的广泛关注。例如，在循环赛中，若需要确定最多能为多少队伍进行排名，则可建模为 DFVS(T) 问题进行求解；若是对特定的种子队伍或赛区队伍进行选拔排名，则等同于求解 DSFVS(T) 问题。特别地，根据不同的排名需求，DSFVS(T) 问题也存在限制版本，简记为 R-DSFVS(T)，严格定义如下。

竞赛图上非限制版子集反馈点集问题 (DSFVS(T))

输入: 一个竞赛图 T ，一个大小为 τ 的关键点集 $T \subseteq V(T)$ 和一个自然数 k 。

输出: 判断竞赛图 T 是否存在一个大小不超过 k 的 T -反馈点集 $S \subseteq V(T)$ 。

竞赛图上限制版子集反馈点集问题 (R-DSFVS(T))

输入: 一个竞赛图 T ，一个大小为 τ 的关键点集 $T \subseteq V(T)$ 和一个自然数 k 。

输出: 判断竞赛图 T 是否存在一个大小不超过 k 的 T -反馈点集 $S \subseteq V(T)$ 。

竞赛图上分离版本的子集反馈点集问题是非限制版和限制版的推广，该问题在迭代压缩技术和解集分离技术中起着至关重要的作用，本章也给出分离版本的具体定义。

竞赛图上分离版子集反馈点集问题 (Disjoint-DSFVS(T))

输入: 一个竞赛图 \mathbf{T} , 一个大小为 τ 的关键点集 $T \subseteq V(\mathbf{T})$, 一个不可删点集 $U \subseteq V(\mathbf{T})$ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断竞赛图 \mathbf{T} 是否存在一个大小不超过 k 的 T -反馈点集 $S \subseteq V(\mathbf{T}) \setminus U$ 。

当不可删点集为空集时, Disjoint-DSFVS(T) 问题退化为 DSFVS(T) 问题; 当不可删点集恰好为关键点集时, Disjoint-DSFVS(T) 问题退化为 R-DSFVS(T) 问题。当关键集 $T = V(\mathbf{T})$, Disjoint-DSFVS(T) 问题退化为竞赛图上分离版本的反馈点集问题 Disjoint-DFVS(T)。Dom 等人^[62]证明了, 当 $\mathbf{T} - U$ 无环时, Disjoint-DFVS(T) 问题能够使用动态规划算法在多项式时间内求解。基于迭代压缩算法框架, 可以得出 DFVS(T) 问题能在 $\mathcal{O}^*(2^k)$ 时间内求解。令人遗憾的是, 当 $\mathbf{T} - U$ 是 T -无环时, Disjoint-DSFVS(T) 仍然是 NP-难的, 无法使用标准迭代压缩技术得到 $\mathcal{O}^*(2^k)$ 时间复杂度的求解算法。由此可见, 竞赛图上的子集反馈点集和反馈点集有着本质上的差异性。

研究现状与挑战 DFVS(T) 问题是著名的隐式 3-碰撞集问题。因此, 该问题可以在 $\mathcal{O}^*(2.0755^k)$ 时间内求解^[54], 也存在点集大小为 $\mathcal{O}(k^2)$ 的核^[240, 241]。基于标准的迭代压缩技术, Dom 等人^[62]证明了当不可删点集是反馈点集时, 分离版本 Disjoint-DFVS(T) 问题可在多项式时间内求解, 进而给出了运行时间为 $\mathcal{O}^*(2^k)$ 的参数算法, 改进了基于 3-碰撞集问题的参数算法的时间复杂度上界。目前, DFVS(T) 问题最快的参数算法由 Kumar 等人^[75]提出, 其运行时间为 $\mathcal{O}^*(1.6181^k)$, 首次打破了 2^k 的计算复杂性瓶颈。基于单调局部搜索技术^[97, 104], DFVS(T) 问题可以在 $\mathcal{O}(1.3820^n)$ 时间内求解。在核心化方面, Fomin 等人^[242, 243]也首次得到了顶点数为 $\mathcal{O}(k^{3/2})$ 的亚平方核。

DSFVS(T) 问题是 DFVS(T) 问题的推广。简单的观察可以发现 DSFVS(T) 问题也是隐式 3-碰撞集问题, 这意味着 DSFVS(T) 问题也可在 $\mathcal{O}^*(2.0755^k)$ ^[54] 和 $\mathcal{O}(1.5182^n)$ ^[97, 104] 时间内求解。然而, 目前针对 DFVS(T) 问题的算法均无法直接应用于 DSFVS(T) 问题上。一方面, 本章证明了当不可删点集为一个 T -反馈点集时, Disjoint-DSFVS(T) 问题依然是 NP-难的, 这直接导致了标准迭代压缩技术无法有效应用于 DSFVS(T) 问题。另一方面, 非关键点的存在增强了图的连通性, 这导致平衡割技术无法直接应用于 DSFVS(T) 问题。综上所述, 改进 DSFVS(T) 问题的参数算法, 甚至突破时间复杂度瓶颈 2^k , 面临着诸多挑战。

主要贡献 本章系统性地开展了 DSFVS(T) 问题和 R-DSFVS(T) 问题的算法研究, 提出了当前最快的参数与精确算法, 并给出了 R-DSFVS(T) 问题的核心化算法。

本章首先探讨了 DSFVS(T) 问题和 R-DSFVS(T) 问题的推广版本 Disjoint-DSFVS(T) 问题。有趣的是, Disjoint-DSFVS(T) 问题是 NP-难的, 但 Disjoint-DFVS(T) 问题是多项式时间可解的。本章找到了介于两者之间的特殊变体问题, 并为该问题提出一个多项式时间的动态规划算法, 其中的一些基本约简规则有助于对 DSFVS(T) 问题的算法研究。

本章的核心贡献在于结合多种近年来提出的前沿技术, 同时改进了 DSFVS(T) 问题的参数算法与精确算法, 证明了 DSFVS(T) 问题可以在 $\mathcal{O}^*(1.6181^k)$ 和 $\mathcal{O}(1.3820^n)$ 时间内求解, 其中参数算法首次突破 2^k 。该参数算法引入了亚指数时间的解集分离技术, 将 DSFVS(T) 问题的实例在亚指数时间内转化为关键点分布“均匀”的 Disjoint-DSFVS(T) 问题实例。值得强调的是, 本章通过迭代压缩技术, 对解集分离技术进行了改良, 精简了算法框架, 解除了对可近似性的依赖, 也优化了时间复杂度。基于此改良, 进一步使用穷举搜索算法便可得到时间复杂度为 $\mathcal{O}^*(2^{k+\mathcal{O}(\sqrt{k} \log k)})$ 的参数算法, 改进了基于 3-碰撞集问题的参数算法结果。接着, 为了在后续环节中应用平衡割技术进行进一步改进, 本章创新性地提出了基于分割块的分支技术, 将运行时间上界降低至 $\mathcal{O}^*(1.6181^k)$ 。

针对 R-DSFVS(T) 问题, 本章建立了该问题与顶点覆盖问题之间紧密的归约关系。基于此归约关系, 本章给出了顶点数为 $3k$ 的线性核, 并且证明了在基本复杂度假设下, 核的大小是渐近紧的。基于同样的归约关系, 本章证明了顶点覆盖问题和 R-DSFVS(T) 问题的参数算法和精确算法能在相同的指数时间内求解。因此, 可以得出 R-DSFVS(T) 问题能在 $\mathcal{O}^*(1.2529^k)$ 和 $\mathcal{O}(1.1996^n)$ 时间内求解。此外, 本章证明了当 τ 为常数且至少为 2 时, R-DSFVS(T) 问题是 NP-难的, 故在 $P \neq NP$ 假设下, 不存在以 τ 为参数的 FPT 算法。主要结果见表6-1。

表 6-1 竞赛图上子集反馈点集问题的主要算法结果

	DSFVS(T) 问题	R-DSFVS(T) 问题
核心化 (顶点数)	$\mathcal{O}(k^2)$ (推论6.17)	$3k$ (定理6.35)
参数算法 (以 k 为参数)	$\mathcal{O}^*(1.6181^k)$ (定理6.15)	$\mathcal{O}^*(1.2529^k)$ (定理6.30)
参数算法 (以 τ 为参数)	$\mathcal{O}^*(1.6181^\tau)$ (推论6.29)	NP-难 ($\tau \geq 2$) (定理6.34)
精确算法	$\mathcal{O}(1.3820^n)$ (定理6.16)	$\mathcal{O}(1.1996^n)$ (定理6.31)

6.2 竞赛图的基本结构与性质

竞赛图是一类任意两个顶点恰有一条有向边相连的有向图, 本文使用 \mathbf{T} 表示竞赛图。有向图的拓扑序 (Topological Ordering) 是它的顶点所构成的线性序, 使得对于每条有向边 vu , v 关于线性序在 u 之前。竞赛图的强连通分类也存在类似

于拓扑序的概念，称为典型序 (**Canonical Ordering**)。

定义 6.1 (典型序) 给定竞赛图 \mathbf{T} ，其点子集构成的一个序列 $\mathcal{X} = (X_1, X_2, \dots, X_\ell)$ 如果满足以下条件，则称为典型序 (**Canonical Order**)：

1. X_1, X_2, \dots, X_ℓ 是点集 $V(\mathbf{T})$ 的划分；
2. X_i ($i \in [\ell]$) 是极大强连通分量；
3. 对于任意顶点 $v \in X_i$ 和 $w \in X_j$ ($i < j$)，存在一条从 v 到 w 的有向边。

竞赛图均存在唯一的典型序，且可以在 $\mathcal{O}(n^2)$ 时间内计算得到^[235, 244]。引理6.2推广了竞赛图上反馈点集的性质。该引理表明，与所有 T -三角形相交的点集均为 DSFVS(T) 问题的 T -反馈点集。

引理 6.2 设 \mathbf{T} 为竞赛图和 $T \subseteq V(\mathbf{T})$ 为其上的关键点集。 \mathbf{T} 是 T -无环的当且仅当 \mathbf{T} 不包含任何 T -三角形。

证明：容易看出，如果图 \mathbf{T} 是 T -无环的，那么 \mathbf{T} 不包含任何 T -三角形。往证若 \mathbf{T} 中没有 T -三角形，则 \mathbf{T} 是 T -无环的。

采用反证法，假设 \mathbf{T} 中没有 T -三角形，但存在一个 T -环 \mathbf{C} 。不失一般性，设 \mathbf{C} 是 \mathbf{T} 中长度最小的 T -环，令 $\mathbf{C} = (t, x_1, x_2, \dots, x_{\ell-1})$ ($\ell \geq 4$)，其中 $t \in T$ 为关键点。考虑 \mathbf{C} 中的两个顶点 t 和 x_2 。由于 \mathbf{T} 中没有 T -三角形，故 $x_2t \notin E(\mathbf{T})$ ；否则， (x_1, x_2, t) 是 T -三角形。此外， \mathbf{T} 是竞赛图，意味着 \mathbf{C} 中的每一对顶点都存在一条有向边连接。所以， tx_2 是一条有向边，这说明了 $\mathbf{C}' = (t, x_2, x_3, \dots, x_{\ell-1})$ 是长度为 $\ell - 1$ 的 T -环。要么 \mathbf{C}' 是 T -三角形，矛盾；要么 \mathbf{C}' 是长度为 $\ell - 1$ 的环，与 \mathbf{C} 的最小性矛盾。综上所述，引理得证。 ■

引理6.2表明，DSFVS(T) 问题是一种隐式 3-碰撞集问题。此外，借助引理6.2，可利用典型序快速判断 DSFVS(T) 问题实例是否为 T -无环的。

引理 6.3 设 \mathbf{T} 为竞赛图， $T \subseteq V(\mathbf{T})$ 为其上的关键点集。 \mathbf{T} 是 T -无环的当且仅当对于任意关键点 $t \in T$ ，单点集 $\{t\}$ 是图 \mathbf{T} 典型序中的一个元素。

证明：令 $\mathcal{X} = (X_1, X_2, \dots, X_\ell)$ 是 \mathbf{T} 唯一的典型序。一方面，若每个关键点均单独构成强连通分量，那么图 \mathbf{T} 一定是 T -无环的。另一方面，因为竞赛图上没有包含恰好 2 个顶点的强连通分量，所以，任一极大强连通分量要么包含 1 个顶点，要么包含至少 3 个顶点。对于任意一个包含了关键点 $t \in T$ 的强连通分量 X_i ($i \in [\ell]$)，可知 $X_i = \{t\}$ 必成立；否则， X_i 中存在 T -环，与 \mathbf{T} 是 T -无环的相矛盾。引理得证。 ■

6.3 一种多项式时间可解情形

本节讨论竞赛图上子集反馈点集问题的一种多项式时间可解的情形。根据定义, Disjoint-DSFVS(T) 问题为 DSFVS(T) 问题和 R-DSFVS(T) 问题的推广。特别地, 对于 Disjoint-DSFVS(T) 问题的实例 $\mathcal{I} = (\mathbf{T}, T, U, k)$, 当 $U = \emptyset$ 时, 退化为 DSFVS(T) 问题的实例; 当 $U = T$ 时, 退化为 R-DSFVS(T) 问题的实例。本节将证明 Disjoint-DSFVS(T) 问题的一个特殊变体时多项式时间可解的, 主要结论如下。

定理 6.4 设 $\mathcal{I} = (\mathbf{T}, T, U, k)$ 为 Disjoint-DSFVS(T) 问题的实例。若 $\mathbf{T} - U$ 的每个强连通分量中的任意两个顶点不在同一个 T -三角形中时, 则可以在多项式时间内求解。

算法的整体设计思想如下: 首先, 计算 $\mathbf{T}[U]$ 的典型序 \mathcal{X} 和 $\mathbf{T} - U$ 的典型序 \mathcal{Y} 。其次, 需要考察 $V(\mathbf{T}) - U$ 中任何一对顶点与 U 中的哪些顶点会形成 T -三角形, 并且刻画 U 中的这些顶点在典型序 \mathcal{X} 中的位置。基于这些位置信息, 采用动态规划思想求解 \mathcal{Y} 的最大子序列, 使得与 \mathcal{X} 合并之后不会产生任何 T -三角形。

6.3.1 基本的约简规则

本节将首先引入最基本的两条约简规则以简化输入实例, 这两个规则对 Disjoint-DSFVS(T) 问题的任意实例均适用, 它们也将应用于 DSFVS(T) 问题和 R-DSFVS(T) 问题的算法之中。

定义 6.5 (冲突点) 若顶点 $v \in V(\mathbf{T}) - U$ 与 U 中的两个顶点构成 T -三角形, 则称 v 为冲突点 (**Conflict Vertex**)。

约简规则 6.3.1 (U -约简规则) 若 $G[U]$ 包含 T -三角形, 则直接返回 \mathcal{I} “无解”。

约简规则 6.3.2 (冲突点约简规则) 若存在冲突点 v , 将其删除, 并使 k 减少 1。

根据定义, U 中的顶点不会放入解中, 因此 U -约简规则6.3.1的正确的。基于同样的原因, 由于冲突点与两个 U 中的顶点构成 T -三角形, 所以 v 必在任意解集中, 故冲突点约简规则6.3.1亦正确。若 U -约简规则6.3.1和冲突点约简规则6.3.2均无法执行, 则 $\mathbf{T}[U]$ 不包含 T -三角形, 且 \mathbf{T} 中无冲突点。

此外, 为方便表述, 可以在原图中新添加两个关键点 s_1 和 s_2 , 使得它们分别处于图 \mathbf{T} 的拓扑序的首尾。由于 s_1 和 s_2 不在任何 T -三角形中, 添加关键点 s_1 和 s_2 并将它们放入 U 中能够保持与原实例的等价性。

定义 6.6 (约简实例) 若 Disjoint-DSFVS(T) 问题的实例满足以下条件, 则称为约简实例。

1. 在 $\mathbf{T} - U$ 的每个强连通分量中, 任意两分顶点均不出现于同一个 T -三角形;
2. 子图 $\mathbf{T}[U]$ 不包含 T -三角形, 且 \mathbf{T} 中无冲突点。

3. 存在关键点 $s_1, s_2 \in U \cap T$ 满足 $N^+(s_1) = V(\mathbf{T}) \cup \{s_2\}$ 且 $N^-(s_2) = V(\mathbf{T}) \cup \{s_1\}$ 。

设 $\mathcal{I} = (\mathbf{T}, T, U, k)$ 为 Disjoint-DSFVS(T) 问题的实例，满足 $\mathbf{T} - U$ 的任何强连通分量中，任意两个顶点均不出现于同一个 T -三角形。总可以对 \mathcal{I} 执行 U -约简规则6.3.1和冲突点约简规则6.3.1，并添加 s_1 和 s_2 两个关键点，使得 \mathcal{I} 是约简的。注意到， U -约简规则6.3.1和冲突点约简规则6.3.1只有删点操作，不难看出可以在 $\mathcal{O}(|U|^2 \cdot n)$ 时间内得到约简实例。综上所述，本节总假设实例 \mathcal{I} 是约简的。显然，删掉约简实例中的一些顶点得到的子实例依然是约简的。这一基本性质将多次应用于本节的分析中。

引理 6.7 设 $\mathcal{I} = (\mathbf{T}, T, U, k)$ 是约简实例。导出子图 $\mathbf{T}[U]$ 和 $\mathbf{T} - U$ 均是 T -无环的。进一步，每个 T -三角形恰好包含 1 个 U 中顶点和 2 个 $V(\mathbf{T}) - U$ 中顶点。

证明：令 \mathbf{C} 是图 \mathbf{T} 中的 T -三角形。若 \mathbf{C} 在 $\mathbf{T} - U$ 中，那么 \mathbf{C} 包含在某个强连通分量内。因此 \mathbf{C} 中的 2 个顶点出现在同一个 T -三角形中，与 \mathcal{I} 是约简的这一条件矛盾。又由于 $\mathbf{T}[U]$ 也不包含 T -三角形，所以 \mathbf{C} 中的 3 个顶点不会同时出现在 U 或 $V(\mathbf{T}) - U$ 中。根据引理6.2，导出子图 $\mathbf{T}[U]$ 和 $\mathbf{T} - U$ 均是 T -无环的。进一步，由于 \mathbf{T} 中无冲突点，因而 \mathbf{C} 不可能只包含 1 个 $V(\mathbf{T}) - U$ 中的顶点。所以，唯一的可能性便是 \mathbf{C} 恰好包含 1 个 U 中顶点和 2 个 $V(\mathbf{T}) - U$ 中顶点。引理得证。 ■

6.3.2 典型序的相容性

考虑约简实例 $\mathcal{I} = (\mathbf{T}, T, U, k)$ ，令 $\mathcal{X} = (X_1, X_2, \dots, X_\ell)$ 为 $\mathbf{T}[U]$ 的典型序。对于任意顶点 $v \in V(\mathbf{T}) - U$ ，定义以下三种下标函数：

$$\alpha(v) := \max\{i \in [\ell] : N^-(v) \cap X_i \neq \emptyset\},$$

$$\beta(v) := \min\{i \in [\ell] : N^+(v) \cap X_i \neq \emptyset\},$$

和

$$\gamma(v) := \begin{cases} \beta(v), & \text{若 } v \text{ 是关键点;} \\ \min\{i \in [\ell] : N^+(v) \cap X_i \cap T \neq \emptyset\}, & \text{若 } v \text{ 是非关键点.} \end{cases}$$

粗略地讲， $\alpha(v)$ 表示 v 的入邻居在 \mathcal{X} 中的最大位置； $\beta(v)$ 表示 v 的出邻居在 \mathcal{X} 中的最小位置；而 $\gamma(v)$ 表示 v 的出邻居且与 v 至少有 1 个为关键点的顶点在 \mathcal{X} 中的最大位置。

由于关键点 $s_1 \in N^-(v)$ 且 $s_2 \in N^+(v)$ ，因此 $\alpha(v)$, $\beta(v)$, $\gamma(v)$ 均为良定义的。注意到，在约简实例 \mathcal{I} 中，若两个非关键点 v 和 w 在同一个强连通分量里，必有

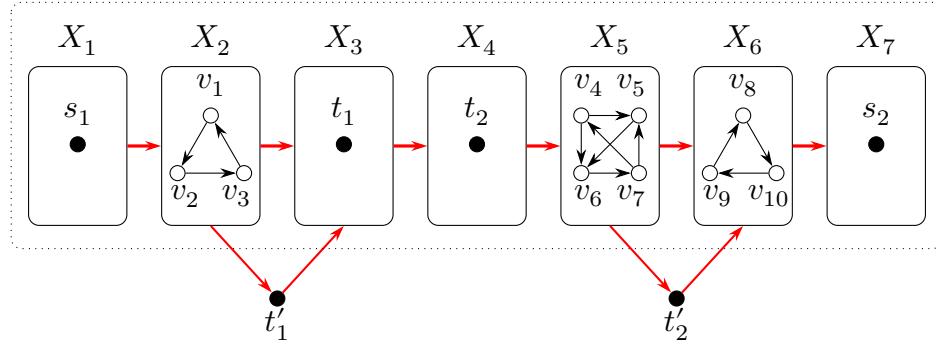


图 6-1 导出子图 $\mathbf{T}[U]$ 与 $\mathbf{T}-U$ 中关键点的关系示意图。 $\mathcal{X} = (X_1, X_2, \dots, X_7)$ 为 $\mathbf{T}[U]$ 的典型序，其中关键点集为 $T = \{t_i, t'_i, s_i\}_{i=1}^2$ ，不可删点集为 $U = \{t_i, s_i\}_{i=1}^2 \cup \{v_i\}_{i=1}^{10}$ ，并且 $X_1 = \{s_1\}$ 以及 $X_2 = \{s_2\}$ 。图中黑色顶点 $\{t_i, t'_i, s_i\}_{i=1}^2$ 表示关键点，白色顶点 $\{v_i\}_{i=1}^{10}$ 表示非关键点。红色粗箭头表示不同点集之间有向边的方向。在此示意图中，可以得到 $\alpha(t'_1) = 2$, $\beta(t'_1) = \gamma(t'_1) = 3$, $\alpha(t'_2) = 5$, 且 $\beta(t'_2) = \gamma(t'_2) = 6$ 。

$\gamma(w) = \gamma(v)$ 。否则，不妨设 $\gamma(w) < \gamma(v)$ 且令 $X_{\gamma(w)} = \{t\}$ ，则有 $tv \in E(\mathbf{T})$ 。那么存在包含 w, v, t 的 T -环，出现矛盾。

引理 6.8 设 $\mathcal{I} = (\mathbf{T}, T, U, k)$ 为约简实例。令 $\mathcal{X} = (X_1, X_2, \dots, X_\ell)$ 是导出子图 $\mathbf{T}[U]$ 的典型序。对任意关键点 $t \in V(\mathbf{T}) - U$ ，其下标函数均满足以下性质（参见图6-1）：

- 对任意 $i \leq \alpha(t)$ ，有 $X_i \subseteq N^-(t)$ ；
- 对任意 $i \geq \beta(t)$ ，有 $X_i \subseteq N^+(t)$ ；
- $\alpha(t) + 1 = \beta(t) = \gamma(t)$ 。

证明：根据引理6.7，导出子图 $\mathbf{T}[U \cup \{t\}]$ 是 T -无环的。考虑典型序 \mathcal{X} 中的任意元素 X_i ，要么 $X_i \subseteq N^-(t)$ 成立，要么 $X_i \subseteq N^+(t)$ 成立。否则，对于顶点 $v \in X_i \cap N^-(t)$ 和顶点 $w \in X_i \cap N^+(t)$ ， X_i 中存在一条从 w 到 v 的路，进而存在一条包含了 w, v, t 三个顶点的 T -环。所以， $X_i \cap N^-(t) \neq \emptyset$ 意味着 $X_i \subseteq N^-(t)$ 成立；同时 $X_i \cap N^+(t) \neq \emptyset$ 也意味着 $X_i \subseteq N^+(t)$ 成立。根据函数 $\alpha(\cdot)$ 和 $\beta(\cdot)$ 的定义，有 $X_i \subseteq N^+(t)$ ($i < \beta(t)$) 且 $X_j \subseteq N^-(t)$ ($j > \alpha(t)$)。最后便得到关系式 $\beta(t) \leq \alpha(t) + 1$ 。

往证 $\beta(t) = \alpha(t) + 1$ 。采用反证法，假设 $\beta(t) < \alpha(t) + 1$ 成立。令顶点 $b \in X_{\beta(t)} \cap N^+(t)$ 以及顶点 $a \in X_{\alpha(t)} \cap N^-(t)$ 。注意到 $\beta(t) \leq \alpha(t)$ ，故考虑 $\beta(t)$ 和 $\alpha(t)$ 的两种大小关系。如果 $\beta(t) < \alpha(t)$ ，则存在一条从 b 到 a 的有向边。如果 $\beta(t) = \alpha(t)$ ，则 $\mathbf{T}[U]$ 中存在从 b 到 a 的路，这是因为 $X_{\alpha(t)} = X_{\beta(t)}$ 且是 $\mathbf{T}[U]$ 中的极大强连通分量。无论哪种大小关系， $\mathbf{T}[U \cup \{t\}]$ 中均存在一条包含 b, a, t 的

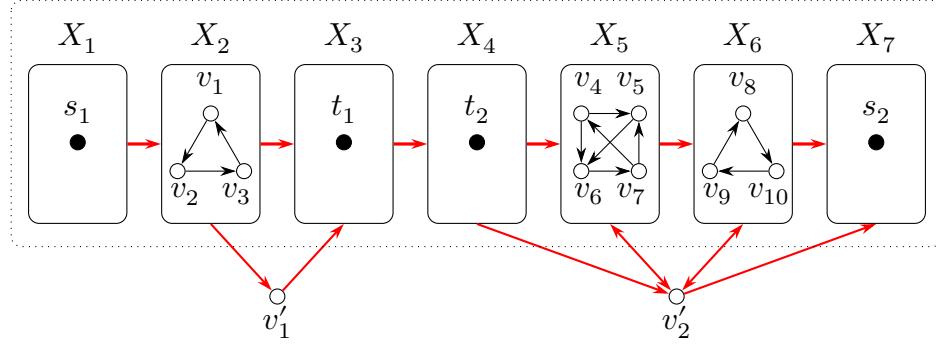


图 6-2 子图 $\mathbf{T}[U]$ 与 $\mathbf{T} - U$ 中非关键点的关系示意图。 $\mathcal{X} = (X_1, X_2, \dots, X_7)$ 为 $\mathbf{T}[U]$ 的典型序，其中关键点集为 $T = \{t_i, s_i\}_{i=1}^2$ ，不可删点集为 $U = \{t_i, s_i\}_{i=1}^2 \cup \{v_i\}_{i=1}^{10}$ ，并且 $X_1 = \{s_1\}$ 以及 $X_2 = \{s_2\}$ 。图中黑色顶点 $\{t_i, s_i\}_{i=1}^2$ 表示关键点，白色顶点 $\{v_i\}_{i=1}^{10} \cup \{v'_i\}_{i=1}^2$ 表示非关键点。红色粗箭头表示不同点集之间有向边的方向，其中双向箭头表示点集间存在两个方向的有向边。在此示意图中，可以得到 $\alpha(v'_1) = 2$, $\beta(v'_1) = \gamma(v'_1) = 3$, $\alpha(v'_2) = 6$, $\beta(v'_2) = 5$, 且 $\gamma(v'_2) = 7$ 。

T -环，矛盾。

综上可知， $\beta(t) = \alpha(t) + 1$ 成立。此外，对于任意下标 $i \geq \alpha(t) + 1 = \beta(t)$ ，都有 $X_i \subseteq N^+(t)$ ，且对于任意下标 $j \leq \beta(t) - 1 = \alpha(t)$ ，都有 $X_j \subseteq N^-(t)$ 。引理得证。 ■

引理 6.9 设 $\mathcal{I} = (\mathbf{T}, T, U, k)$ 为约简实例。令 $\mathcal{X} = (X_1, X_2, \dots, X_\ell)$ 是导出子图 $\mathbf{T}[U]$ 的典型序。对于任意非关键点 $v \in V(\mathbf{T}) \setminus (U \cup T)$ ，其下标函数均满足以下性质（参见图6-2）：

- 对任意 $i < \beta(v)$ ，有 $X_i \subseteq N^-(v)$ ；
- 对任意 $i > \alpha(v)$ ，有 $X_i \subseteq N^+(v)$ ；
- $\beta(v) \leq \alpha(v) + 1 \leq \gamma(v)$ ；
- 对任意满足 $\beta(v) \leq i < \gamma(v)$ 的下标 i ，有 $X_i \cap T = \emptyset$ 。

证明：根据引理6.7，导出子图 $\mathbf{T}[U \cup \{v\}]$ 是 T -无环的。根据函数 $\alpha(\cdot)$ 和 $\beta(\cdot)$ 的定义，有 $X_i \subseteq N^+(v)$ ($i < \beta(v)$) 且 $X_j \subseteq N^-(v)$ ($j > \alpha(v)$)。这意味着关系式 $\beta(v) \leq \alpha(v) + 1$ 成立。

往证 $\alpha(v) + 1 \leq \gamma(v)$ 。由函数 $\alpha(\cdot)$ 的定义，只需证明 $N^-(v) \cap X_i = \emptyset$ 对于 $i \geq \gamma(v)$ 均成立。设 $X_{\gamma(v)} = \{t\}$ ，其中 $t \in T \cap U$ 为一关键点。自然有 $t \in N^+(v)$ 但 $t \notin N^-(v)$ 。采用反证法，假设存在顶点 $w \in N^-(v) \cap X_i$ ($i > \gamma(v)$)。那么，根据典型序的定义，必有一条从 t 到 w 的有向边。注意到 $vw \in E(\mathbf{T})$ 为一条有向边，从而说明 (t, w, v) 是 T -三角形，矛盾。所以关系式 $\beta(v) \leq \alpha(v) + 1 \leq \gamma(v)$ 得证。

最后, 根据函数 $\beta(\cdot)$ 和 $\gamma(\cdot)$ 的定义, 对于任意满足 $\beta(v) \leq i < \gamma(v)$ 的下标 i , 可以得出 $X_i \cap T = \emptyset$ 。引理得证。 ■

考虑一个集合构成的序列 $\mathcal{Y} = (Y_1, Y_2, \dots, Y_r)$, 令 $V(\mathcal{Y}) = \bigcup_{i=1}^r Y_i$ 表示 \mathcal{Y} 中所有顶点构成的集合, 并且用 $|V(\mathcal{Y})|$ 表示 \mathcal{Y} 的大小。如果存在单调不减的函数 $\varphi: [r'] \rightarrow [r]$ 使得序列 $\mathcal{Y}' = (Y'_1, Y'_2, \dots, Y'_{r'})$ 满足 $Y'_i \subseteq Y_{\varphi(i)}$ 且 $\varphi(i) < \varphi(j)$ 当且仅当 $i < j$, 则称 \mathcal{Y}' 为 \mathcal{Y} 的子序列。

定义 6.10 (相容性) 设 $\mathcal{I} = (\mathbf{T}, T, U, k)$ 为约简实例。令 $\mathcal{X} = (X_1, X_2, \dots, X_\ell)$ 是导出子图 $\mathbf{T}[U]$ 的典型序。对于点子集 $V' \subseteq V(\mathbf{T}) \setminus U$, 如果 $\mathbf{T}[V']$ 中的任意一条有向边 $vw \in E(\mathbf{T}[V'])$ 满足

1. 若顶点 v 和 w 中至少有 1 个是关键点, 则 $\alpha(v) \leq \alpha(w)$ 且 $\beta(v) \leq \beta(w)$;
2. 若顶点 v 和 w 均非关键点, 则 $\gamma(v) \leq \gamma(w)$ 。

则称 V' 与 \mathcal{X} 相容 (**Compatible**)。

引理 6.11 设 $\mathcal{I} = (\mathbf{T}, T, U, k)$ 为约简实例。令 $\mathcal{X} = (X_1, X_2, \dots, X_\ell)$ 是导出子图 $\mathbf{T}[U]$ 的典型序。点集 $S \subseteq V(\mathbf{T}) \setminus U$ 是实例 \mathcal{I} 的 T -反馈点集当且仅当 $V(\mathbf{T}) \setminus (S \cup U)$ 与 \mathcal{X} 相容。

证明: 首先证明, 若 $S \subseteq V(\mathbf{T}) \setminus U$ 是实例 \mathcal{I} 的 T -反馈点集, 则集合 $V(\mathbf{T}) \setminus (S \cup U)$ 与 \mathcal{X} 相容。考察 $V(\mathbf{T}) \setminus (S \cup U)$ 中的两个顶点 v 和 w , 不妨设 $vw \in E(\mathbf{T})$ 。根据它们是否为关键点, 讨论两种情形。

情形 1.1: 顶点 v 和 w 至少有 1 个为关键点。在此情形中, 子图 $\mathbf{T} - S$ 中不存在同时包含 v 和 w 的环。观察到 $X_{\alpha(v)} \cap N^-(v)$ 是非空的。所以 $X_{\alpha(v)} \cap N^+(w)$ 为空, 否则在 $\mathbf{T}[X_{\alpha(v)} \cup \{v, w\}]$ 中存在包含 v 和 w 的环, 这是因为 $X_{\alpha(v)}$ 是 $\mathbf{T}[U]$ 中的极大强连通分量。那么, 有 $X_{\alpha(v)} \subseteq N^-(w)$ 成立, 故有关系式 $\alpha(v) \leq \alpha(w)$ 。类似地, 由于 $X_{\beta(v)} \cap N^+(v)$ 非空, 故 $X_{\beta(v)} \cap N^-(w)$ 为空, 并且 $X_{\beta(v)} \subseteq N^+(w)$ 。这也可以得出关系式 $\alpha(v) \leq \alpha(w)$ 。综上, 无论何种情况, 均说明 $V(\mathbf{T}) \setminus (S \cup U)$ 与 \mathcal{X} 相容。

情形 1.2: 顶点 v 和 w 均非关键点。在此情形中, 子图 $\mathbf{T}[U \cup \{v, w\}]$ 是 T -无环的。设 $X_{\gamma(w)} = \{t\}$, 其中 $t \in T \cap U$ 为一关键点。自然有 $vt \in E(\mathbf{T})$ 且 $wt \in E(\mathbf{T})$, 否则 (v, w, t) 是 T -三角形, 矛盾。所以 $t \in N^+(v)$ 并且关系式 $\gamma(v) \leq \gamma(w)$ 成立。

往证若 $V(\mathbf{T}) \setminus (S \cup U)$ 与 \mathcal{X} 相容, 那么 S 必是 \mathcal{I} 的 T -反馈点集。引理 6.7, 只需证明对于任意的二元集合 $\{v, w\} \subseteq V(\mathbf{T}) \setminus (S \cup U)$, 子图 $\mathbf{T}[U \cup \{v, w\}]$ 不会包含 T -三角形。不失一般性, 假设 vw 为一条有向边, 并且根据 v 和 w 是否为关键点, 讨论两种情形。

情形 2.1: 顶点 v 和 w 均非关键点。在此情形下, 只需证明集合 $N^-(v) \cap$

$N^+(w) \cap (U \cap T)$ 为空。由引理6.8, 极大强连通分量 X_i ($i \geq \gamma(v)$) 中的任意关键点均属于 v 的出邻居。类似地, 由引理6.8, 极大强连通分量 X_j ($j < \gamma(w)$) 中的任意关键点均属于 w 的入邻居。所以, 若 $N^-(v) \cap N^+(w) \cap (U \cap T)$ 中存在关键点 t , 其必在极大强连通分量 X_i 内, 其中 $\gamma(w) \leq i < \gamma(v)$ 。但是, 根据定义6.10, 有 $\gamma(v) \leq \gamma(w)$ 成立。这意味着关键点 t 不存在, 矛盾。

情形 2.2: 顶点 v 和 w 至少有 1 个为关键点。在此情形中, 只需证明 $N^-(v) \cap N^+(w) \cap U$ 为空。根据引理6.8和引理6.9, 对于任意的下标 $i > \alpha(v)$ 和 $j < \beta(w)$, 有 $X_i \subseteq N^+(v)$ 且 $X_j \subseteq N^-(w)$ 成立。若顶点 u 属于 $N^-(v) \cap N^+(w) \cap X_i$, 可以得出

$$\beta(w) \leq i \leq \alpha(v).$$

因为 $\alpha(v) \leq \alpha(w)$ 且 $\beta(v) \leq \beta(w)$, 若 w 是关键点, 则可得到

$$\alpha(v) + 1 \leq \alpha(w) + 1 = \beta(w),$$

而若 v 是关键点, 也可得到

$$\alpha(v) + 1 = \beta(v) \leq \beta(w).$$

这说明了不存在满足 $\beta(w) \leq i \leq \alpha(v)$ 的下标 $i \in [r]$, 进而表明集合 $N^-(v) \cap N^+(w)$ 里没有 U 中顶点, 也即 $N^-(v) \cap N^+(w) \cap U$ 为空。

综上所述, 引理得证。 ■

引理 6.12 设 $\mathcal{I} = (\mathbf{T}, T, U, k)$ 为约简实例。令 $\mathcal{X} = (X_1, X_2, \dots, X_\ell)$ 是导出子图 $\mathbf{T}[U]$ 的典型序, $\mathcal{Y} = (Y_1, Y_2, \dots, Y_r)$ 是 $\mathbf{T} - U$ 的典型序。设 $\mathcal{Y}' = (Y'_1, Y'_2, \dots, Y'_{r'})$ 是 \mathcal{Y} 的子序列, v 是 $Y'_{r'}$ 中的顶点。点子集 $V(\mathcal{Y}')$ 与 \mathcal{X} 相容当且仅当

- $V(\mathcal{Y}') \setminus \{v\}$ 与 \mathcal{X} 相容;
- 对任一 $V(\mathcal{Y}') \setminus \{v\}$ 中的关键点 t , 有 $\alpha(t) < \beta(v)$;
- 对任一 $V(\mathcal{Y}') \setminus \{v\}$ 中的非关键点 w , 有 $\alpha(w) < \gamma(v)$ 。

证明: 首先证明充分性。假设 $V(\mathcal{Y}')$ 与 \mathcal{X} 相容。那么 $V(\mathcal{Y}')$ 的任何子集 (包括 $V(\mathcal{Y}') \setminus \{v\}$ 在内) 也与 \mathcal{X} 相容。令 v' 是 $V(\mathcal{Y}')$ 中与 v 不同的另一顶点。注意到 v 属于 \mathcal{Y}' 的最后一个元素, 因此根据 v' 所属集合以及 v' 是否为关键点, 分三种情形进行讨论。

情形 1.1: 顶点 v' 在与 $Y'_{r'}$ 不同的另一集合 Y'_i 中, 且 v' 为关键点。在此情形中, 有 $i < r'$ 并且 $v'v$ 构成一条有向边。因为 v' 是关键点, 根据引理6.8, 可知 $\alpha(v') < \beta(v')$ 。结合定义6.10以及 $v'v$ 是一条有向边的事实, 可以得出 $\beta(v') \leq \beta(v)$ 。所以关系式 $\alpha(v') < \beta(v)$ 成立。

情形 1.2: 顶点 v' 在与 Y'_r 不同的另一集合 Y'_i 中, 且 v' 不是关键点。在此情形中, 也有 $i < r'$ 且 $v'v$ 构成一条有向边。由于 v' 为非关键点, 根据引理6.9, 可知 $\alpha(v') < \gamma(v')$ 。结合定义6.10以及 $v'v$ 是一条有向边的事实, 可以得出 $\gamma(v') \leq \gamma(v)$ 。所以关系式 $\alpha(v') < \gamma(v)$ 成立。

情形 1.3: 顶点 v' 恰与 v 同在集合 Y'_r 中。在此情形中, 由引理6.3, 顶点 v 与 v' 均非关键点。所以根据引理6.9, 可得 $\alpha(v') < \gamma(v')$ 。又因为 \mathcal{I} 是约简实例, $\gamma(v') = \gamma(v)$ 成立, 所以有 $\alpha(v') < \gamma(v)$ 。

下证必要性。假设 $V(\mathcal{Y}') \setminus \{v\}$ 与 \mathcal{X} 相容的, 则对任意关键点 $t \in V(\mathcal{Y}') \setminus \{v\}$, 都有 $\alpha(t) < \beta(v)$, 并且对任意非关键点 $w \in V(\mathcal{Y}') \setminus \{v\}$, 都有 $\alpha(w) < \gamma(v)$ 。

由于集合 $V(\mathcal{Y}') \setminus \{v\}$ 与 \mathcal{X} 相同, 只需验证顶点 v 与集合 $V(\mathcal{Y}') \setminus \{v\}$ 中另一顶点 v' 的关系即可。根据顶点 v 和顶点 v' 是否为关键点, 分两种情形进行讨论。

情形 2.1: 顶点 v 与 v' 中至少有 1 个是关键点。由相容性的定义 (定义6.10), 只需证明 $\alpha(v') \leq \alpha(v)$ 以及 $\beta(v') \leq \beta(v)$ 即可。由引理6.3, 存在下标 $i < r'$ 使得 $v' \in Y_i$, 这意味着 $v'v$ 是一条有向边。根据引理条件, 有 $\alpha(v') < \beta(v)$, 进而可知

$$\alpha(v') + 1 \leq \beta(v).$$

再根据引理6.8和引理6.9, 无论 v 是关键点与否, 都有

$$\beta(v) \leq \alpha(v) + 1.$$

类似地, 无论 v' 是关键点与否, 亦都有

$$\beta(v') \leq \alpha(v') + 1.$$

结合上述关系式, 可得

$$\beta(v') \leq \alpha(v') + 1 \leq \beta(v) \leq \alpha(v) + 1,$$

从而说明了 $\alpha(v') \leq \alpha(v)$ 以及 $\beta(v') \leq \beta(v)$ 是成立的。

情形 2.2: 顶点 v' 和 v 均非关键点。若 v' 和 v 都属同一集合 Y'_r , 那么函数值 $\gamma(v')$ 和 $\gamma(v)$ 相等。现考虑 v' 属于集合 Y'_i , 而 $i < r'$, 此时, $v'v$ 是一条有向边。根据引理条件, 有 $\alpha(v') < \gamma(v)$ 。由于 v' 是非关键点, 由引理6.9, 可知 $\beta(v') \leq \alpha(v') + 1$ 。进而可得

$$\beta(v') \leq \alpha(v') + 1 \leq \gamma(v).$$

又由于 v 也为非关键点, 根据引理6.9, 对于所有满足 $\beta(v') \leq i < \gamma(v')$ 的极大强连通分量 X_i 都不可能包含关键点。注意到 $X_{\gamma(v)}$ 必是由 1 个关键点构成的集合。

所以下标 $\gamma(v)$ 满足

$$\gamma(v) < \beta(v') \text{ 或 } \gamma(v) \geq \gamma(v').$$

结合上述关系式, 最终可得 $\gamma(v') \leq \gamma(v)$ 。

综上所述, 引理得证。 ■

6.3.3 基于典型序的动态规划算法

本节将提出基于典型序的动态规划算法求解约简实例 $\mathcal{I} = (\mathbf{T}, T, U, k)$ 。

设 $\mathcal{X} = (X_1, X_2, \dots, X_\ell)$ 为 $\mathbf{T}[U]$ 的典型序, $\mathcal{Y} = (Y_1, Y_2, \dots, Y_r)$ 为 $\mathbf{T} - U$ 的典型序。事实上, 本节提出的动态规划算法将计算与 \mathcal{X} 相容的 \mathcal{Y} 的最大子序列 \mathcal{Y}' 。根据引理6.11, 集合 $V(\mathbf{T}) - U$ 中不属于子序列 \mathcal{Y}' 的顶点构成的点集就是实例 \mathcal{I} 的最小 T -反馈点集。

首先按照如下操作得到 $V(\mathbf{T}) - U$ 中顶点的线性序: 对每个集合 Y_i , 以任意顺序对 Y_i 中的顶点排序得到 $\tau_i = (v_{i,1}, v_{i,2}, \dots, v_{i,|Y_i|})$; 拼接 $\tau_1, \tau_2, \dots, \tau_r$ 便可得到集合 $V(\mathbf{T}) - U$ 全体顶点的线性序 τ 。然后对顶点重新编号使得 $\tau = (v_1, v_2, \dots, v_{n-|U|})$ 。

回顾前文, 子序列 \mathcal{Y}' 的大小定义为 \mathcal{Y}' 中包含的顶点数。用符号 $\text{dp}[i, p, q]$ ($0 \leq i \leq n - |U|$, $1 \leq p \leq q \leq |U|$) 表示 \mathcal{Y} 满足如下条件的最大子序列 \mathcal{Y}' 的大小:

1. 点集 $V(\mathcal{Y}')$ 与 \mathcal{X} 相容;
2. 点集 $V(\mathcal{Y}')$ 仅包含 τ 中的前 i 个顶点;
3. 点集 $V(\mathcal{Y}')$ 中的任意关键点 t 均满足 $\alpha(t) \leq p$;
4. 点集 $V(\mathcal{Y}')$ 中的任意非关键点 v 均满足 $\alpha(v) \leq q$ 。

下面介绍两个用于设计动态规划的引理。

引理 6.13 若 v_i 是关键点, 那么

$$\text{dp}[i, p, q] = \begin{cases} 0, & \text{如果 } i = 0, \\ \text{dp}[i - 1, p, q], & \text{如果 } i \neq 0, \alpha(v_i) > p, \\ \max\{\text{dp}[i - 1, p, q], 1 + \text{dp}[i - 1, p', q']\}, & \text{如果 } i \neq 0, \alpha(v_i) \leq p, \end{cases}$$

其中 $p' = q' = \alpha(v_i)$ 。

证明: 设 \mathcal{Y}' 是 $\text{dp}[i, p, q]$ 所对应的 \mathcal{Y} 的子序列, 即 \mathcal{Y}' 的大小为 $\text{dp}[i, p, q]$ 。当 $i = 0$ 时, 显然对任意的 p 和 q , 都有 $\text{dp}[i, p, q] = 0$ 。接下来考虑 $i > 0$ 的情况。

考虑序列 (v_1, v_2, \dots, v_i) 的最后一个元素, 顶点 v_i 。根据 v_i 是否在序列 \mathcal{Y}' 中, 分两种情形进行讨论。

情形 1: v_i 不属于集合 $V(\mathcal{Y}')$ 。在此情形中, 子序列 \mathcal{Y}' 仅包含序列 τ 的前 $i - 1$ 个顶点, 这意味着 $\text{dp}[i, p, q] = \text{dp}[i - 1, p, q]$ 。特别地, 当 $\alpha(v_i) > p$ 时, 顶点

v_i 必然不在集合 $V(\mathcal{Y}')$ 之中。

情形 2: v_i 属于集合 $V(\mathcal{Y}')$ 。在此情形中, 首先由引理6.8, 可知 $\beta(v_i) - 1 = \gamma(v_i) - 1 = \alpha(v_i)$ 。考虑集合 $V(\mathcal{Y}') \setminus \{v_i\}$ 中的任意顶点 $v_{i'}$ 。由于 v_i 是关键点, 则根据引理6.3可知 $v_{i'}$ 和 v_i 不在子序列 \mathcal{Y}' 的同一极大强连通分量中。根据引理6.12, 若 $v_{i'}$ 是关键点, 有

$$\alpha(v_{i'}) \leq \beta(v_i) - 1 = \alpha(v_i).$$

类似地, 若 $v_{i'}$ 为非关键点, 有

$$\alpha(v_{i'}) \leq \gamma(v_i) - 1 = \alpha(v_i).$$

所以, 只需计算 $\text{dp}[i-1, p', q']$, 其中 $p' = \min \{p, \alpha(v_i)\}$ 且 $q' = \min \{q, \alpha(v_i)\}$ 。注意到, 根据此情形的条件, 有 $\alpha(v_i) \leq p$, 所以 $p' = \alpha(v_i)$; 同时, 根据 $\text{dp}[\cdot, \cdot, \cdot]$ 的定义以及 $\alpha(v_i) \leq p$, 有 $p \leq q$, 可得 $q' = \alpha(v_i)$ 。

综上, 无论哪种情形, 均符合递归关系式 (6.13), 引理得证。 ■

引理 6.14 若 v_i 是非关键点, 那么

$$\text{dp}[i, p, q] = \begin{cases} 0, & \text{如果 } i = 0, \\ \text{dp}[i-1, p, q], & \text{如果 } i \neq 0, \alpha(v_i) > q, \\ \max\{\text{dp}[i-1, p, q], 1 + \text{dp}[i-1, p', q']\}, & \text{如果 } i \neq 0, \alpha(v_i) \leq q, \end{cases}$$

其中

$$p' = \min \{p, \beta(v_i) - 1\} \text{ 且 } q' = \min \{q, \gamma(v_i) - 1\}.$$

证明: 此证明类似于引理6.13的论证技巧。设 \mathcal{Y}' 是 $\text{dp}[i, p, q]$ 所对应的 \mathcal{Y} 的子序列, 即 \mathcal{Y}' 的大小为 $\text{dp}[i, p, q]$ 。当 $i = 0$ 时, 显然对任意的 p 和 q , 都有 $\text{dp}[i, p, q] = 0$ 。接下来考虑 $i > 0$ 的情况。

考虑序列 (v_1, v_2, \dots, v_i) 的最后一个元素, 顶点 v_i 。根据 v_i 是否在序列 \mathcal{Y}' 中分两种情形进行讨论。

情形 1: v_i 不在集合 $V(\mathcal{Y}')$ 中。在此情形中, 子序列 \mathcal{Y}' 仅包含序列 τ 的前 $i-1$ 个顶点, 这意味着 $\text{dp}[i, p, q] = \text{dp}[i-1, p, q]$ 。特别地, 当 $\alpha(v_i) > q$ 时, 顶点 v_i 必然不在集合 $V(\mathcal{Y}')$ 之中。

情形 2: v_i 在集合 $V(\mathcal{Y}')$ 中。考虑集合 $V(\mathcal{Y}') \setminus \{v_i\}$ 中的任意顶点 $v_{i'}$ 。若 $v_{i'}$ 与 v_i 不在子序列 \mathcal{Y}' 的同一极大强连通分量中, 则 $v_{i'}v_i$ 是一条有向边。根据引理6.12, 若 $v_{i'}$ 是关键点, 有

$$\alpha(v_{i'}) \leq \beta(v_i) - 1.$$

类似地, 若 $v_{i'}$ 为非关键点, 有

$$\alpha(v_{i'}) \leq \gamma(v_i) - 1.$$

另一方面, 若顶点 $v_{i'}$ 和 v_i 在 \mathcal{Y}' 的同一极大强连通分量中, 由引理6.3, 顶点 $v_{i'}$ 和 v_i 均非关键点。根据线性序的构造, 满足关系 $\gamma(v_{i'}) \leq \gamma(v_i)$ 。再由引理6.12, 可得 $\alpha(v_{i'}) < \gamma(v_i)$ 。所以, 可以推出

$$\alpha(v_{i'}) \leq \gamma(v_{i'}) - 1 \leq \gamma(v_i) - 1.$$

综上, 无论哪种情形, 均符合引理6.14表述, 引理得证。 ■

依据引理6.11, 约简实例 $\mathcal{I} = (\mathbf{T}, T, U, k)$ 的最小 T -反馈点集大小必为 $n - |U| - \text{dp}[n - |U|, |U|, |U|]$ 。最后给出采用标准自底向上方法的动态规划算法计算 $\text{dp}[n - |U|, |U|, |U|]$ 。

算法 6-1 Disjoint-DSFVS(T) 问题一个特殊变体的动态规划算法

输入: Disjoint-DSFVS(T) 问题的实例 $\mathcal{I} = (\mathbf{T}, T, U, k)$, 其中子图 $\mathbf{T} - U$ 的每个强连通块中任意两个顶点不在同一个 T -三角形中。

输出: 实例 \mathcal{I} 有解或无解。

- 1 执行 U -约简规则6.3.1。// 此步骤后, $\mathbf{T}[U]$ 无 T -三角形。
- 2 执行冲突点约简规则6.3.2。// 此步骤后, \mathbf{T} 中无冲突点。
- 3 在竞赛图 \mathbf{T} 的拓扑序首尾各添加 1 个关键点 s_1 和 s_2 , 并将它们加入集合 U 中。// 此步骤后得到约简实例。
- 4 求出导出子图 $\mathbf{T}[U]$ 的典型序 \mathcal{X} 以及子图 $\mathbf{T} - U$ 的典型序 \mathcal{Y} 。
- 5 构造一个 $V(\mathbf{T}) - U$ 的线性序, 使其符合 \mathcal{Y} 的顺序。
- 6 初始化 $\text{dp}[0, p, q] = 0$, 其中 $1 \leq p \leq q \leq |U|$ 。
- 7 **for** $i = 1$ **to** $n - |U|$ **do**
- 8 | 对任意的正整数 p 和 q ($1 \leq p \leq q \leq |U|$), 使用函数值 $\text{dp}[i - 1, \cdot, \cdot]$ 根据引理6.13和引理6.14计算 $\text{dp}[i, p, q]$ 。
- 9 **end**
- 10 返回函数值 $\text{dp}[n - |U|, |U|, |U|]$ 。

定理 6.4 设 $\mathcal{I} = (\mathbf{T}, T, U, k)$ 为 Disjoint-DSFVS(T) 问题的实例。若 $\mathbf{T} - U$ 的每个强连通分量中的任意两个顶点不在同一个 T -三角形中时, 则可以在多项式时间内求解。

证明: 计算 $\text{dp}[i, p, q]$ 并存储结果需要常数时间, 且至多有 $|U|^2 \cdot n$ 个候选函数值 $\text{dp}[\cdot, \cdot, \cdot]$ 需进行计算和存储, 故函数值 $\text{dp}[n - |U|, |U|, |U|]$ 可在 $\mathcal{O}(|U|^2 \cdot n)$ 时间内求解。在执行6-1前, 需在 $\mathcal{O}(n^2)$ 时间内执行 U -约简规则和冲突点约简规则, 还需在线性时间内添加关键点 s_1 和 s_2 , 从而获得约简实例。综上, Disjoint-DSFVS(T) 问题可以在 $\mathcal{O}(|U|^2 \cdot n + n^2)$ 时间内计算, 定理得证。 ■

6.4 竞赛图上非限制版子集反馈点集问题

本节将提出 DSFVS(T) 问题的参数算法与精确算法，其中参数算法首次打破了 2^k 的瓶颈，改进了当前最快的 $\mathcal{O}^*(2.0755^k)$ 时间的参数算法。然后基于参数算法，得到了 DSFVS(T) 问题的精确算法，改进了当前最快的 $\mathcal{O}(1.5182^n)$ 时间的精确算法，主要结论如下。

定理 6.15 DSFVS(T) 问题可以在 $\mathcal{O}^*(1.6181^k)$ 时间内求解。

定理 6.16 DSFVS(T) 问题可以在 $\mathcal{O}(1.3820^n)$ 时间内求解。

DSFVS(T) 问题的参数算法包括三个部分。

第一部分使用了亚指数时间的解集分离技术。该技术能够将 DSFVS(T) 问题的实例 \mathcal{I} 在亚指数时间内转化为竞赛图上分离版子集反馈点集问题的实例 \mathcal{I}' 。实例 \mathcal{I}' 中，不可删点集 U 的拓扑序能够将剩下的顶点分割为 $|U| + 1$ 个部分，解集分离技术保证了每一部分包含足够少的不在解中的关键点。

第二部分采用传统的分支算法求解 \mathcal{I}' ，若一条有向边与 U 中的顶点构成了 T -三角形，则称其为冲突边。冲突边的至少 1 个端点必然在解集中，因此对冲突边进行简单分支操作后，便可独立地对每一“分割块”进行穷举搜索求解，其运行时间是亚指数的。最终能够得到运行时间为 $\mathcal{O}^*(2^{k+o(k)})$ 的参数算法。

第三部分进一步提出了更具全局性的分支操作，用以处理连接了较多冲突边的“分割块”，减弱图的连通性。在此基础上使用基于平衡割技术的分治算法，得到了运行时间为 $\mathcal{O}^*(1.6181^k)$ 的参数算法，成功突破了 2^k 的瓶颈。

最后，DSFVS(T) 问题的精确算法可以直接通过单调局部搜索技术获得。定理 6.15 和定理 6.16 所得到的算法时间复杂度已与 DFVS(T) 问题目前最快的参数算法与精确算法复杂度一致。

6.4.1 亚指数时间的解集分离技术

本节将介绍基于拓扑序的亚指数时间的解集分离技术 (Solution-Disjoint Technique)。针对 DSFVS(T) 问题的实例 $\mathcal{I} = (\mathbf{T}, T, k)$ ，解集分离技术能够在亚指数时间内得到与解集不相交的关键点子集 $U \subseteq T$ 。因此， $\mathcal{I}' = (\mathbf{T}, T, U, k)$ 恰好为 Disjoint-DSFVS(T) 问题的实例。解集分离技术依赖于常数近似率的近似解和多项式核心化算法。依据引理 6.2，DSFVS(T) 问题是隐式 3-碰撞问题。所以，隐式 3-碰撞集问题的核心化结果^[240,241]可以直接适用于 DSFVS(T) 问题。

推论 6.17 DSFVS(T) 问题存在点数为 $\mathcal{O}(k^2)$ 的核。

回顾 6.3.1 节，冲突点是集合 $V(\mathbf{T}) \setminus U$ 中的顶点，该点与 U 中两个顶点构成 T -三角形。删除 $\mathcal{I}' = (\mathbf{T}, T, U, k)$ 中的所有冲突点后，竞赛图 \mathbf{T} 的典型序将被 U

分割为 $|U| + 1$ 个部分, 每一部分被称为分割块。现给出分割块的严格定义。

定义 6.18 (分割块) 令 $U \subseteq T$ 为一个子关键集, 满足其导出子图 $\mathbf{T}[U]$ 是无环的。如果 B 是满足对 $v, u \in B$ 有

$$\begin{cases} N^-(v) \cap U = N^-(u) \cap U; \\ N^+(v) \cap U = N^+(u) \cap U. \end{cases}$$

的极大点子集, 则称其为 U -分割块。

换言之, U -分割块是 $V(\mathbf{T}) \setminus U$ 的极大点子集, 满足所有顶点在 U 中的入邻居以及出邻居相同。注意到, 由于 $\mathbf{T}[U]$ 是无环的, 故存在唯一的拓扑序。若 U -分割块 B 满足 $\mathbf{T}[U \cup B]$ 是 T -无环的, 那么存在位置 $i \in \{0, 1, \dots, |U|\}$ 使得 $\mathbf{T}[U]$ 拓扑序的前 i 个顶点属于 $N^-(B)$ 且后 $|U| - i + 1$ 个顶点属于 $N^+(B)$ 。

引理 6.19 是本节的核心结论。它表明, 可以使用亚指数的时间分离出一个关键点子集 U , U 与某个最优解 S 交为空, 且在子图 $\mathbf{T} - S$ 中, U -分割块仅包含亚线性个关键点。

引理 6.19 (解集分离技术) 令 $\mathcal{I} = (\mathbf{T}, T, k)$ 是 DSFVS(T) 问题的顶点数量为 $\mathcal{O}(k^2)$ 的实例。已知大小为 $k + 1$ 的 T -反馈点集 $\bar{S} \subseteq V(\mathbf{T})$, 存在运行时间为 $2^{\mathcal{O}(\sqrt{k} \log k)}$ 的算法, 输出一个大小不超过 $2^{\mathcal{O}(\sqrt{k} \log k)}$ 关键点集族 \mathcal{F} , 使得对于任意 T -反馈点集 S , 均存在一个关键点子集 $U \in \mathcal{F}$ 满足以下性质:

1. U 与 S 的交为空;
2. 在子图 $\mathbf{T} - S$ 中, 任意 U -分割块至多包含 $2\sqrt{k}$ 个关键点。

证明: \bar{S} 是大小为 $k + 1$ 的 T -反馈点集, 所以 $\mathbf{T} - \bar{S}$ 是 T -无环的, 进而说明子图 $\mathbf{T}[T \setminus \bar{S}]$ 是无环的。所以, $\mathbf{T}[T \setminus \bar{S}]$ 存在唯一的拓扑序 $t_1, t_2, \dots, t_{|T \setminus \bar{S}|}$ 。现对 $T \setminus \bar{S}$ 进行染色: 使用 $\ell = \lceil \sqrt{k} \rceil$ 种颜色, 编号为 $0, 1, \dots, \ell - 1$, 按照拓扑序依次循环染色。可知, 用颜色 r 进行染色的顶点构成的集合为

$$Z_r := \{t_i \in T \setminus \bar{S} : i \bmod \ell = r\}, \quad r = 0, 1, \dots, \ell - 1.$$

最后便得到了 $T \setminus \bar{S}$ 的划分 $(Z_0, Z_1, \dots, Z_{\ell-1})$ 。

集族 \mathcal{F} 中的任意元素将使用四元组 (r', S', T'_1, T'_2) 进行表示, 其中

- r' 是颜色编号 $0 \leq r' \leq \ell - 1$;
- $S' \subseteq Z_{r'}$ 是染色为 r' 的关键点子集, 其大小不超过 \sqrt{k} ;
- $T'_1 \subseteq T \setminus \bar{S}$ 是不在近似解中的关键点子集, 其大小不超过 \sqrt{k} ;
- $T'_2 \subseteq T \cap \bar{S}$ 是在近似解中的关键点子集, 其大小也不超过 \sqrt{k} 。

对于给定的四元组 (r', S', T'_1, T'_2) , 其对应的关键点子集 U 定义为

$$U = (Z_{r'} \cap S') \cup T'_1 \cup T'_2.$$

接下来, 首先估计 \mathcal{F} 的大小以及算法的运行时间。根据条件, 图 \mathbf{T} 有 $\mathcal{O}(k^2)$ 个顶点。此外, \bar{S} 有 $k+1$ 个顶点, 故可得到 $|\mathcal{F}|$ 的上界:

$$\begin{aligned} & \lceil \sqrt{k} \rceil \cdot \binom{\mathcal{O}(k^2)}{\sqrt{k}} \cdot \binom{\mathcal{O}(k^2)}{\sqrt{k}} \cdot \binom{k+1}{\sqrt{k}} \\ & \leq (\sqrt{k}+1) \cdot \mathcal{O}((k^2)^{\sqrt{k}}) \cdot \mathcal{O}((k^2)^{\sqrt{k}}) \cdot (k+1)^{\sqrt{k}} \\ & = 2^{\mathcal{O}(\sqrt{k} \log k)}. \end{aligned}$$

所以, 枚举算法的运行时间也控制在 $2^{\mathcal{O}(\sqrt{k} \log k)}$ 内。

最后说明算法的正确性, 即对于任意解集 S , 均存在关键点子集 $U \in \mathcal{F}$ 满足

- U 与 S 的交为空;
- 在子图 $\mathbf{T} - S$ 中, 任意 U -分割块至多包含 $2\sqrt{k}$ 个顶点。

注意到 S 至多包含 k 个关键点, 根据抽屉原理, 必然存在颜色 r^* 使得

$$|Z_{r^*} \cap S| \leq \frac{k}{\lceil \sqrt{k} \rceil} \leq \sqrt{k}.$$

成立。令 $r' = r^*$ 且 $S' = Z_{r^*} \cap S$, 自然有 $0 \leq r' \leq \ell-1$ 且 $|S'| \leq \sqrt{k}$ 。

然后构造 $T'_1 \subseteq T \cap \bar{S}$: 关键点 $t'_1 \in T'_1$ 当且仅当存在关键点 $t' \in S'$ 使得 t'_1 是 $T \cap (S \cup \bar{S})$ 中拓扑序在 t' 之后的第一个关键点。由于 S' 中每个关键点最多对应一个 T'_1 中的关键点, 因此有 $|T'_1| \leq |S'| \leq \sqrt{k}$ 。此外, 根据 T'_1 的构造, 不难看出在子图 $\mathbf{T} - (S \cup \bar{S})$ 中, 每个 $(Z_{r'} \cap S') \cup T'_1$ -分割块至多包含 $\ell-1 \leq \sqrt{k}$ 个关键点。

最后构造 $T'_2 \subseteq T \cap \bar{S}$ 。考虑在近似解 \bar{S} 却不在解集 S 中的关键点。这些关键点所导出的子图必是无环的, 这是因为 S 是图 \mathbf{T} 的 T -反馈点集。这说明了 $\mathbf{T}[(T \cap \bar{S}) \setminus S]$ 也存在唯一的拓扑序。所以可以如下构造 T'_2 : 关键点 $t'_2 \in T'_2$ 当且仅当它是拓扑序中的第 $i \cdot \lceil \sqrt{k} \rceil$ 个顶点, 其中 $i \in \mathbb{N}$ 取遍不超过 $k+1$ 的自然数。同理可知, 在子图 $\mathbf{T}[(T \cap \bar{S}) \setminus S]$ 中, 每个 $(T \cap \bar{S}) \setminus S$ -分割块也至多包含 $\ell-1 \leq \sqrt{k}$ 个关键点。

设 $U = (Z_{r'} \cap S') \cup T'_1 \cup T'_2$, 便可得出, 在子图 $\mathbf{T} - S$ 中, 每个 U -分割块至多包含 $2\sqrt{k}$ 个关键点。综上所述, 引理得证。 ■

值得强调的是, 根据集族 \mathcal{F} 中各关键点子集的四元组表示, 依次枚举 \mathcal{F} 中各元素仅需要多项式空间。

如果 Disjoint-DSFVS(\mathbf{T}) 问题的实例 $\mathcal{I} = (\mathbf{T}, T, U, k)$ 满足 $U \subseteq T$ 是一个关键点子集且所有 U -分割块至多包含 $2\sqrt{k}$ 个关键点, 则称其为 $2\sqrt{k}$ -均匀的。那么,

算法 6-2 基于亚指数时间解集分离技术的 DSFVS(T) 问题计算框架

输入: DSFVS(T) 问题的实例 $\mathcal{I} = (\mathbf{T}, T, k)$ 。

输出: 实例 \mathcal{I} 有解或无解。

1 根据推论6.17, 计算 \mathcal{I} 的核, 并对 \mathcal{I} 进行更新 // 此步骤后, 点集大小满足 $|V(\mathbf{T})| = \mathcal{O}(k^2)$ 。

2 对实例 \mathcal{I} 采用迭代压缩技术。初始化, 对 \mathbf{T} 中顶点进行编号 $V(\mathbf{T}) = \{v_1, v_2, \dots, v_n\}$, 然后令 $V_i = \{v_1, v_2, \dots, v_i\}$ 并且 $S_k = V_k$ 。

3 **for** $i = k + 1$ **to** n **do**

4 更新图 \mathbf{T}_i 的大小为 $k + 1$ 的 T -反馈点集 $\bar{S}_i = S_{i-1} \cup \{v_i\}$ ($S_k = V_k$)。

5 根据引理6.20, 计算 Compression-DSFVS(T) 问题实例 $\mathcal{I}_i = (\mathbf{T}_i, T_i, \bar{S}_i, k)$, 其中 $\bar{S}_k = V_k$

使用亚指数时间的解集分离技术得到关键点集族 \mathcal{F} 。不会对集族 \mathcal{F} 进行存储, 但能依次获得其中的元素。// 集族 \mathcal{F} 大小满足 $|\mathcal{F}| = 2^{\mathcal{O}(\sqrt{k} \log k)}$ 。

6 **while** 依次选取关键点子集 $U \in \mathcal{F}$ **do**

7 由引理6.19, 得到 Disjoint-DSFVS(T) 问题实例 $\mathcal{I}_U = (\mathbf{T}_i, T_i, U, k)$ 。// 实例 \mathcal{I}_U 是 $2\sqrt{k}$ -均匀的。

8 在实例 \mathcal{I}_U 中, 图 \mathbf{T}_i 的大小为 $|V(\mathbf{T}_i)| = i \leq n$ 。因此, 在 $\mathcal{O}(f(k) \cdot n^{g(k)})$ 时间内求解实例 \mathcal{I}_U 。// 将在6.4.2节和6.4.3节进行介绍。

9 **if** 实例 \mathcal{I}_U 有解 **then**

10 实例 \mathcal{I}_U 的解就是 \mathcal{I}_i 的解 S_i , 然后 **break**。// 解集 S_i 将用于计算实例 \mathcal{I}_{i+1} 中的 \bar{S}_i 。

11 **end**

12 **end**

13 若对任意 $U \in \mathcal{F}$, 实例 \mathcal{I}_U 均无解, 则直接返回 \mathcal{I} “无解”。

14 **end**

15 返回实例 \mathcal{I} “有解”。// 实例 \mathcal{I}_n 的解 S_n 即为 \mathcal{I} 的解。

通过解集分离技术, 可以在亚指数时间内枚举得到一个等价的 $2\sqrt{k}$ -均匀的实例。

引理 6.20 若 $2\sqrt{k}$ -均匀的实例能在 $\mathcal{O}(f(k) \cdot n^{g(k)})$ 时间内求解, 则 DSFVS(T) 问题可以在 $\mathcal{O}^*(f(k) \cdot 2^{\mathcal{O}((g(k) + \sqrt{k}) \log k)})$ 时间内求解。

证明: 首先给出求解 DSFVS(T) 问题的参数算法 (参见算法6-2)。根据定理6.17, 调用核心化算法得到 DSFVS(T) 问题的核 $\mathcal{I} = (\mathbf{T}, T, k)$, 满足 $|V(\mathbf{T})| = \mathcal{O}(k^2)$ 。实例 \mathcal{I} 有解当且仅当 \mathcal{I}_0 有解。然后使用迭代压缩技术如下: 任给 \mathbf{T} 中的顶点进行编号 $V(\mathbf{T}) = \{v_1, v_2, \dots, v_n\}$, 其中 $n = |V(\mathbf{T})|$ 。用 $V_i = \{v_1, v_2, \dots, v_i\}$ 表示前 i 个顶点构成的点子集, 且令 $T_i = T \cap V_i$ 和 $\mathbf{T}_i = \mathbf{T}[V_i]$ ($i \in [n]$)。考察压缩版本, 即在迭代压缩框架中 Compression-DSFVS(T) 问题。实例 $\mathcal{I}_k = (\mathbf{T}_k, T_k, \bar{S}_k = V_k, k)$ 有解, 其中 \bar{S}_k 表示一个大小不超过 $k + 1$ 的 T_k -反馈点集。实例 \mathcal{I}_k 一定有解, 因为图 \mathbf{T}_k 仅包含 k 个顶点, 故 $S_k = \bar{S}_k$ 就是 \mathcal{I}_k 的解。对于 $i = k + 1, k + 2, \dots, n$, 都有压缩版本的实例 $\mathcal{I}_i = (\mathbf{T}_i, T_i, \bar{S}_i, k)$, 其中 $\bar{S}_i = S_{i-1} \cup \{v_i\}$, 这里 S_{i-1} 表示

实例 \mathcal{I}_{i-1} 的解。那么 \bar{S}_i 均为实例 \mathcal{I}_i 的大小不超过 $k+1$ 的 T_i -反馈点集。由 \mathcal{I}_i ($i \in \{k+1, k+2, \dots, n\}$) 的构造可知, \mathcal{I}_i 可由 \mathcal{I}_{i-1} 的解 S_{i-1} 得到。由于 \mathcal{I}_i 是 \mathcal{I} 的子实例, 所以 \mathcal{I}_i 无解蕴含着 \mathcal{I} 无解, 否则可以依次求解实例 $\mathcal{I}_k, \mathcal{I}_{k+1}, \dots, \mathcal{I}_n$ 。最终得到 \mathcal{I}_n 的解 S_n , 即是 \mathcal{I} 的解。所以, 若 Compression-DSFVS(T) 问题能在 $\mathcal{O}^*(h(k))$ 时间内求解, 则 DSFVS(T) 问题也能在

$$(n-k) \cdot \mathcal{O}^*(h(k)) = \mathcal{O}(k^2) \cdot \mathcal{O}^*(h(k)) = \mathcal{O}^*(h(k))$$

时间内求解。因此, 只需证明 Compression-DSFVS(T) 问题能在

$$\mathcal{O}^*(f(k) \cdot 2^{\mathcal{O}((g(k)+\sqrt{k}) \log k)})$$

时间内求解即可。

设 $\mathcal{I}' = (\mathbf{T}', T', \bar{S}, k)$ 为 Compression-DSFVS(T) 问题的实例, 满足 $V(\mathbf{T}') = \mathcal{O}(k^2)$ 。已知 \bar{S} 是 \mathcal{I}' 的大小为 $k+1$ 的 T' -反馈点集 (若大小不足 $k+1$, 可以任意添加顶点使其大小恰为 $k+1$)。接着使用引理6.19中亚指数时间的解集分离算法, 得到一族关键点子集 \mathcal{F} 。对于 \mathcal{F} 中的每个关键点子集 $U \subseteq T'$, 有对应的实例 $\mathcal{I}_U = (\mathbf{T}', T', U, k)$, 这是一个 Disjoint-DSFVS(T) 问题的实例。

依据引理6.19, 解集分离算法的时间复杂度为 $\mathcal{O}^*(2^{\mathcal{O}(\sqrt{k} \log k)})$, 并且对于任意 $U \in \mathcal{F}$, 实例 \mathcal{I}_U 是 $2\sqrt{k}$ -均匀的。于是对于每个 $U \in \mathcal{F}$, 可以在 $\mathcal{O}(f(k) \cdot n^{g(k)})$ 时间内进行求解。若存在点集 U , 其对应的实例 \mathcal{I}_U 为有解, 则 \mathcal{I}_0 有解, 否则 \mathcal{I}_0 无解。总运行时间为

$$\mathcal{O}^*(2^{\sqrt{k} \log k}) \cdot \mathcal{O}^*((f(k) \cdot (k^2)^{g(k)})) = \mathcal{O}^*(f(k) \cdot 2^{\mathcal{O}((g(k)+\sqrt{k}) \log k)}).$$

引理得证。 ■

6.4.2 基于简单分支搜索技术的参数算法

本节将给出运行时间为 $\mathcal{O}^*(2^{k+\mathcal{O}(\sqrt{k} \log k)})$ 的一个简单参数算法求解 DSFVS(T) 问题。基于6.4.1小节的引理6.20, 本节假设输入实例 $\mathcal{I} = (\mathbf{T}, T, U, k)$ 是 $2\sqrt{k}$ -均匀的, 且不可删点集 $U \subseteq T$ 为关键点子集。

类似于冲突点的定义, 现给出冲突边的定义, 冲突边将用于本节与6.4.3节的算法描述。

定义 6.21 (冲突边) 若顶点 $v, u \in V(\mathbf{T})$ U 与 U 中 1 个顶点构成 T -三角形, 且 vu 是一条有向边, 则称 vu 为一条冲突边 (**Conflict Arc**)。

根据分割块的定义, 冲突边的两个端点不会同时包含在同一个 U -分割块内。显然, 冲突边的至少 1 个端点会在解集中, 这引出了一条简单的分支规则, 冲突

边分支规则。

分支规则 6.4.1 (冲突边分支规则) 若存在冲突边 $e = uv$, 则分别计算以下两个子分支。若两个子分支均“无解”, 则返回“无解”; 否则返回“有解”。

- 删去 v , 并使 k 减 1;
- 删去 u , 并使 k 减 1。

引理 6.22 设 $\mathcal{I} = (\mathbf{T}, T, U, k)$ 为 Disjoint-DSFVS(T) 问题的实例。若实例 \mathcal{I} 中无冲突点与冲突边, 则任意 T -三角形要么在一个 U -分割块中, 要么在导出子图 $\mathbf{T}[U]$ 中。

证明: 设 \mathbf{C} 为一个不在导出子图 $\mathbf{T}[U]$ 中的 T -三角形。往证 \mathbf{C} 不包含 U 中顶点。若 \mathbf{C} 中恰有 1 个 U 中顶点, 则存在一条冲突边, 与引理条件矛盾。若 \mathbf{C} 中恰有 2 个 U 中顶点, 则存在一条冲突点, 也与引理条件矛盾。所以, \mathbf{C} 中所有顶点均未出现在 U 中, 引理得证。 ■

依据引理 6.22, 若 U -约简规则 6.3.1、冲突点约简规则 6.3.2 和冲突边分支规则 6.4.1 均不能执行, 则任意 T -三角形均出现在某个 U -分割块的导出子图中。所以可以单独计算 U -分割块的导出子图的最小 T -反馈点集。注意到存在一个最优解, 使得每个 U -分割块中至多只有 $2\sqrt{k}$ 个关键点不在其中。现给出求解 $2\sqrt{k}$ -均匀实例的完整分支算法。

算法 6-3 求解 $2\sqrt{k}$ -均匀实例的简单分支算法

输入: 一个 $2\sqrt{k}$ -均匀的实例 $\mathcal{I} = (\mathbf{T}, T, U, k)$, 其中 $U \subseteq T$ 是关键点子集。

输出: 实例 \mathcal{I} 有解或无解。

- 1 执行 U -约简规则 6.3.1。// 此步骤后, $\mathbf{T}[U]$ 无 T -三角形。
- 2 执行冲突点约简规则 6.3.2。// 此步骤后, \mathbf{T} 中无冲突点。
- 3 执行冲突边分支规则 6.4.1。// 此步骤后, \mathbf{T} 中无冲突边。
- 4 计算所有 U -分割块。对于每个 U -分割块 B_i ($i \in [\ell]$), 穷举搜索所有大小不超过 $|B_i \cap T|$ 且包含至多 $2\sqrt{k}$ 个非关键点的点集, 判断是否为实例 $\mathcal{I}_i = (\mathbf{T}[B_i], T \cap B_i, \emptyset, k)$ 的解, 进而得到实例 \mathcal{I}_i 的最优解集 S_i 。若存在某个实例无解, 则直接返回“无解”。
- 5 计算 $S = S_1 \cup S_2 \cup \dots \cup S_\ell$, 将 $|S|$ 与 k 进行比较, 若 $k \geq |S|$, 则返回“有解”; 若 $k < |S|$, 则返回“无解”。

引理 6.23 算法 6-3 是正确的, 并且该算法可在 $2^k \cdot n^{\mathcal{O}(\sqrt{k} \log k)}$ 时间内求解 $2\sqrt{k}$ -均匀的实例。

证明: 设 $\mathcal{I} = (\mathbf{T}, T, U, k)$ 为 $2\sqrt{k}$ -均匀实例。 U -约简规则以及冲突点约简规则对于 DSFVS(T) 问题的实例都能正确执行。并且由于冲突边的至少 1 个端点在解集中, 冲突边分支规则也是正确的。在三个规则执行完成以后, 由引理 6.22 可

知, 任意 T -三角形仅出现在某个 U -分割块的导出子图中。设所有 U -分割块为 B_1, B_2, \dots, B_ℓ 。只需独立地求解实例 $\mathcal{I}_i = (\mathbf{T}[B_i], T \cap B_i, \emptyset, k)$ 的最优解, 注意到 U 中的顶点不出现在任何 T -三角形中, 因此将 \mathcal{I}_i 中的不可删点集设为空集是正确的。对于任意 U -分割块 B_i ($i \in [\ell]$), 设 $T_i = T \cap B_i$, 令 $\mathcal{I}_i = (\mathbf{T}[B_i], T_i, \emptyset, k)$ 的最优解集为 S_i , 且满足 $|T_i \setminus S_i| \leq 2\sqrt{k}$ 。由于 T_i 是 \mathcal{I}_i 的 T_i -反馈点集, 所以 \mathcal{I}_i 的最优解大小满足关系式 $|S_i| \leq |T_i|$ 。此外, \mathcal{I} 是 $2\sqrt{k}$ -均匀的, 有 $|S_i \cap T| \geq |T_i| - 2\sqrt{k}$, 说明了 S_i 包含至多 $2\sqrt{k}$ 个非关键点。因此穷举搜索所有大小不超过 $|T_i|$ 且包含至多 $2\sqrt{k}$ 个非关键点的点集一定能够找到最优解。若某个实例 \mathcal{I}_i ($i \in [\ell]$) 无解, 那么原实例的最小 T_i -反馈点集的大小必然大于 k , 故也无解。若每个实例 \mathcal{I}_i ($i \in [\ell]$) 均有最优解 S_i , 由于不存在某个 T -三角形同时与两个 U -分割块相交, 所以并集 $S = S_1 \cup S_2 \cup \dots \cup S_\ell$ 一定是 \mathcal{I} 的最优解。综上所述, 算法6-3是正确的。

现在估计算法的时间复杂度。首先, U -约简规则和冲突点约简规则可在多项式时间内完成, 且最多执行 n 次。冲突边分支规则的分支向量为 $(1, 1)$, 对应的分支因子为 2。此外, 不难发现 U -分割块也可在多项式时间内计算得到。接下来估计穷举搜索实例 $\mathcal{I}_i = (\mathbf{T}[B_i], T_i, \emptyset, k)$ 的最小 T_i -反馈点集 S_i 的运行时间上界。一方面, 由于 \mathcal{I} 是 $2\sqrt{k}$ -均匀的, 有 $|S_i \cap T| \geq |B_i \cap T| - 2\sqrt{k}$ 。另一方面, S_i 包含至多 $2\sqrt{k}$ 个非关键点, 即 $|S_i \setminus T| \leq 2\sqrt{k}$ 。所以, 穷举搜索的运行时间至多为

$$\begin{aligned} & \sum_{i=1}^{\ell} \sum_{j=0}^{\lfloor 2\sqrt{k} \rfloor} \binom{|B_i \cap T|}{|B_i \cap T| - j} \cdot \sum_{j=0}^{\lfloor 2\sqrt{k} \rfloor} \binom{|B_i \cap T|}{j} \cdot |B_i|^{\mathcal{O}(1)} \\ & \leq \ell \cdot (2\sqrt{k} + 1) \cdot n^{\mathcal{O}(\sqrt{k})} \cdot (2\sqrt{k} + 1) \cdot n^{\mathcal{O}(\sqrt{k})} \cdot n^{\mathcal{O}(1)} \\ & = n^{\mathcal{O}(\sqrt{k})}. \end{aligned}$$

综上, 算法整体的时间复杂度为 $2^k \cdot n^{\mathcal{O}(\sqrt{k})}$, 引理得证。 ■

结合引理6.20和引理6.23, 便可得到定理6.24。

定理 6.24 DSFVS(T) 问题可以在 $\mathcal{O}^*(2^{k+\mathcal{O}(\sqrt{k} \log k)})$ 时间内求解。

6.4.3 基于平衡割技术的参数算法

考虑 Disjoint-DSFVS(T) 问题的 $2\sqrt{k}$ -均匀的实例 $\mathcal{I} = (\mathbf{T}, T, U, k)$ 。本节假设 U -约简规则6.3.1以及冲突点约简规则6.3.2均已不可执行。算法6-3的思想是首先处理掉所有冲突点和冲突边, 然后针对各 U -分割块所导出的子图在 $n^{\mathcal{O}(\sqrt{k} \log k)}$ 时间内计算最小 T -反馈点集再求并集。事实上, 并不需要完全处理掉所有冲突边再求解各连通块导出子图的最小 T -反馈点集。可以优先处理掉一部分构成割的冲突边, 再利用分治思想来设计算法。基于此思想, 需要构造实例 $\mathcal{I} = (\mathbf{T}, T, U, k)$ 的辅助

图 **F**。辅助图 **F** 是一个多重无向图并且满足以下性质：

- 对于 $\mathbf{T} - U$ 中的任意 U -分割块 B , 都存在顶点 $x_B \in V(\mathbf{F})$;
- 对于端点在不同 U -分割块的冲突边 uv ($u \in B$ 且 $v \in B'$), 都存在一条无向边 $e_{uv} = x_B x_{B'}$ 。

根据辅助图的构造, **F** 没有自环。以下引理6.25能够说明, 若 **F** 不连通, 则可以使用分治思想构造算法。

引理 6.25 设 $\mathcal{I} = (\mathbf{T}, T, U, k)$ 为 Disjoint-DSFVS(\mathbf{T}) 问题的实例, \mathcal{I} 中无冲突点且 $\mathbf{T}[U]$ 中无 T -三角形。若 \mathcal{I} 的辅助图 **F** 不连通, 令 (C_0, C_1) 为 $V(\mathbf{F})$ 的划分, 且满足 C_0 与 C_1 不连通, 再令

$$X_0 = \bigcup_{x_B \in C_0} B \text{ 且 } X_1 = \bigcup_{x_B \in C_1} B.$$

设 S_0 为子实例 $\mathcal{I}_0 = (\mathbf{T}[X_0], X_0 \cap T, U, k)$ 的最优解, S_1 为子实例 $\mathcal{I}_1 = (\mathbf{T}[X_1], X_1 \cap T, U, k)$ 的最优解。那么 $S_0 \cup S_1$ 则为实例 \mathcal{I} 的最优解。

证明: 设 S 为实例 \mathcal{I} 的最优解。由于 S_0 为子实例 \mathcal{I}_0 的最优解, S_1 为子实例 \mathcal{I}_1 的最优解, 所以 $|S_0| \leq |S \cap X_0|$ 且 $|S_1| \leq |S \cap X_1|$ 。那么有 $|S_0 \cup S_1| = |S_0| + |S_1| \leq |S|$ 。所以, 只需证明 $S_0 \cup S_1$ 为实例 \mathcal{I} 的一个解即可。

采用反证法, 假设存在 T -三角形 **C** 在 $\mathbf{T} - (S_0 \cup S_1)$ 中。显然, **C** 中既有 X_0 里的顶点, 也有 X_1 里的顶点。若 **C** 中的三条有向边均非冲突边, 那么 **C** 一定在某个 U -分割块 B 中。则要么 $B \subseteq X_0$ 要么 $B \subseteq X_1$, 矛盾。所以 **C** 中一定存在冲突边 vu 。但是, 由于 **C** 是一些连通块的并, 说明没有分别以 X_0 中顶点和 X_1 中顶点为端点的冲突边。因此, $\mathbf{T} - (S_0 \cup S_1)$ 不存在冲突边, 同样产生矛盾。

综上所述, $S_0 \cup S_1$ 是实例 \mathcal{I} 的最优解。引理得证。 ■

引理6.25表明, 若 **F** 不连通, 可以独立地计算 \mathcal{I}_1 和 \mathcal{I}_2 的最优解, 然后求并集。这保证了基于平衡割的分治思想的可行性。

引理 6.26 设 $\mathcal{I} = (\mathbf{T}, T, U, k)$ 为 $2\sqrt{k}$ -均匀的实例, \mathcal{I} 中无冲突点且 $\mathbf{T}[U]$ 中无 T -三角形。若 \mathcal{I} 的辅助图 **F** 的最大度为 d , 则 \mathcal{I} 可以在 $\mathcal{O}^*(2^{2|E(\mathbf{F})|/3} \cdot 2^{\mathcal{O}(d \log |E(\mathbf{F})|)} \cdot n^{\mathcal{O}(\sqrt{k})})$ 时间内求解。

证明: 以下给出基于平衡割的分治算法。不妨设 **F** 是连通的, 否则, 根据引理6.25, 可以独立地求解每个连通分量对应的子实例的最优解, 然后求并集。特别地, 若 U -分割块 B 不与任何冲突边相连, 则可以用类似于算法6-3的思想穷举搜索实例 $\mathcal{I}_B = (\mathbf{T}[B], T \cap B, \emptyset, k)$ 的最优解。具体地, 枚举所有大小不超过 $|B_i \cap T|$ 且包含至多 $2\sqrt{k}$ 个非关键点的点集, 判断其是否为 \mathcal{I}_B 的最优解, 进而可以得到 \mathcal{I}_B 的最优解。

若某个子实例无解，则原实例的最小 T -反馈点集的大小一定超过 k ，因此原实例也无解。首先，由于 \mathbf{F} 是没有自环和孤立点的多重无向图，故可在多项式时间内计算划分 (P_0, P_1) ，使其满足

$$\begin{cases} \frac{|E(\mathbf{F})|}{4} - \frac{d}{2} \leq |E(\mathbf{F}[P_i])| \leq \frac{|E(\mathbf{F})|}{4} + \frac{d}{2}, \forall i \in \{0, 1\}; \\ \frac{|E(\mathbf{F})|}{2} - d \leq |E(P_0, P_1)| \leq \frac{|E(\mathbf{F})|}{2} + d, \end{cases}$$

令

$$X_0 = \bigcup_{x_B \in P_0} B \text{ 且 } X_1 = \bigcup_{x_B \in P_1} B.$$

那么 (X_0, X_1) 是 $V(\mathbf{T}) \setminus U$ 的划分。一个端点在 X_0 而另一个端点在 X_1 的冲突边的集合记作 $E(X_0, X_1)$ ，那么根据辅助图的定义，有 $|E(X_0, X_1)| = |E(P_0, P_1)|$ 。初始化 $S' = \emptyset$ ，依次选择 $E(X_0, X_1)$ 中的每条边 $e = vu$ ，枚举哪一个端点在解集中，将该顶点放入集合 S' ，直至 $E(X_0, X_1)$ 中的每条边都至少有 1 个端点在 S' 中。显然有 $|S'| \leq |E(X_0, X_1)| = |E(P_0, P_1)|$ 。考虑实例 $\mathcal{I}' = (\mathbf{T} - S', T \setminus S', U, k - |S'|)$ ，若存在某个 S' ，使得 \mathcal{I}' 有解，则 \mathcal{I} 亦有解；反之 \mathcal{I} 则无解。

接下来，对于每个固定的 S' ，将对实例 \mathcal{I}' 进行分治操作。显然， \mathcal{I}' 对应的辅助图 \mathbf{F}' 是 \mathbf{F} 的子图，即 $V(\mathbf{F}') \subseteq V(\mathbf{F})$ 且 $E(\mathbf{F}') \subseteq E(\mathbf{F})$ 。令 $P'_0 = P_0 \cap V(\mathbf{F}')$ 并且 $P'_1 = P_1 \cap V(\mathbf{F}')$ ，可知 P'_0 和 P'_1 不连通。根据引理6.25，分别计算两个实例 $\mathcal{I}'_0 = (\mathbf{T}[X'_0], T \cap X'_0, U, k - |S'|)$ 和 $\mathcal{I}'_1 = (\mathbf{T}[X'_1], T \cap X'_1, U, k - |S'|)$ 的最优解 S'_0 和 S'_1 ，那么 $S' = S'_0 \cup S'_1$ 则为实例 \mathcal{I}' 的最优解。若 \mathcal{I}'_0 或 \mathcal{I}'_1 其中无解，则 \mathcal{I}' 的最小 $T \setminus S'$ -反馈点集的大小一定超过 $k - |S'|$ ，因此 \mathcal{I}' 亦无解。

现分析分治算法的时间复杂度。设 $|V(\mathbf{T})| = n$, $|E(\mathbf{F})| = l$ 且 $\Delta(\mathbf{F}) = d$ 。用符号 $\mathcal{T}(l, d, n)$ 表示计算实例 \mathcal{I} 的运行时间。

若 \mathbf{F} 不连通，则独立地求解两个子实例的最优解。设 l_0 和 l_1 为两个子实例所对应的辅助图的边数， d_0 和 d_1 为两个子实例所对应的辅助图的最大度， n_0 和 n_1 为两个子实例中竞赛图的顶点数。显然有 $l_0 + l_1 = l$, $d_0, d_1 \leq d$, $n_0 + n_1 = n$ ，那么

$$\begin{aligned} \mathcal{T}(l, d, n) &\leq \mathcal{T}(l_0, d_0, n_0) + \mathcal{T}(l_1, d_1, n_1) + n^{\mathcal{O}(1)} \\ &\leq \mathcal{T}(l, d, n_0) + \mathcal{T}(l, d, n - n_0) + n^{\mathcal{O}(1)}. \end{aligned}$$

此时，有 $\mathcal{T}(l, d, n) = n^{\mathcal{O}(1)}$ 。

若 \mathbf{F} 连通，则需要在多项式时间内找到辅助图 \mathbf{F} 的顶点划分 (P_0, P_1) ，并计算 $V(\mathbf{T}) \setminus U$ 的划分 (X_0, X_1) 。由于集合 S' 的大小不超过 $|E(P_0, P_1)| \leq l/2 + d$ ，

故不同的集合 S' 至多有 $2^{l/2+d}$ 种可能性。对于每个固定的 S' ，分治算法需要求解两个子实例 \mathcal{I}'_0 和 \mathcal{I}'_1 ，设它们的点集大小分别为 n'_0 和 n'_1 ，辅助图边数分别为 l'_0 和 l'_1 ，最大度分别为 d'_0 和 d'_1 。有关系式 $n'_0 + n'_1 + |S'| \leq n$, $d_0, d_1 \leq d$, 且 $l_0, l_1 \leq l/2 + d/4$, 所以可得

$$\begin{aligned}\mathcal{T}(l, d, n) &\leq 2^{l/2+d} \cdot (\mathcal{T}(l_0, d_0, n_0) + \mathcal{T}(l_1, d_1, n_1) + n^{\mathcal{O}(1)}) + n^{\mathcal{O}(1)} \\ &\leq 2^{l/2+d} \cdot \mathcal{T}\left(\frac{l}{4} + \frac{d}{2}, d, n\right) + 2^{l/2+d} \cdot n^{\mathcal{O}(1)},\end{aligned}$$

解得

$$\mathcal{T}(l, d, n) = 2^{2l/3} \cdot 2^{\mathcal{O}(d \log \ell)} \cdot \mathcal{T}(0, d, n).$$

当 $l = 0$, 采用的穷举搜索的时间复杂度为

$$\begin{aligned}\mathcal{T}(0, d, n) &= \sum_{i=0}^{2\sqrt{k}} \binom{\tau}{\tau-i} \cdot \sum_{i=0}^{2\sqrt{k}} \binom{n-\tau}{i} \cdot n^{\mathcal{O}(1)} \\ &\leq (2\sqrt{k}+1) \cdot \binom{n}{2\sqrt{k}} \cdot (2\sqrt{k}+1) \cdot \binom{n}{2\sqrt{k}} \cdot n^{\mathcal{O}(1)} \\ &= n^{\mathcal{O}(\sqrt{k})}.\end{aligned}$$

所以总时间复杂度为

$$\mathcal{T}(l, d, n) = 2^{2l/3} \cdot 2^{\mathcal{O}(d \log l)} \cdot n^{\mathcal{O}(\sqrt{k})}.$$

引理得证。 ■

余下的工作只需考虑如何控制 $|E(\mathbf{F})|$ 和 $\Delta(\mathbf{F})$ 的大小。接下来将引入两条分支规则和两条约简规则，其中某条规则只有在前述所有规则均不可执行时才会开始执行。这四条规则执行完成之后，可以保证 $|E(\mathbf{F})| \leq d$ 且 $\Delta(\mathbf{F}) \leq k/\log^2 k$ ，此时 $2\sqrt{k}$ -均匀的实例可以在

$$2^{2k/3} \cdot 2^{\mathcal{O}(k/\log k)} \cdot n^{\mathcal{O}(\sqrt{k})} = 1.5874^k \cdot n^{\mathcal{O}(\sqrt{k})}.$$

的时间内求解。

分支规则 6.4.2 (相交冲突边分支规则) 若存在顶点 v 连接了至少两条冲突边，则分别计算以下两个子分支。若两个子分支均“无解”，则返回“无解”；否则返回“有解”。

- 删去 v , 并使 k 减 1;
- 对于每个与 v 之间存在冲突边的顶点 u , 删去 u , 并使 k 减 1。

约简规则 6.4.3 (冲突边约简规则) 若冲突边的数量超过 k , 则返回 \mathcal{I} “无解”。

由于每条冲突边的至少 1 个端点在解集中，因此相交冲突边分支规则6.4.2是

正确的。若相交冲突边分支规则6.4.2不可执行，那么任意两条冲突边没有公共端点。因此只有当冲突边的数量不超过 k 时，实例有解，故冲突边约简规则6.4.3亦是正确的。

约简规则 6.4.4 (分割块约简规则)若存在 U -分割块 B 不包含关键点，令 $Z \subseteq V(\mathbf{T})$ ， B 为所有与 B 中某一顶点构成了冲突边的顶点集，删去 $B \cup Z$ ，并使得 k 减去 $|Z|$ 。

引理 6.27 分割块约简规则6.4.4是正确的。

证明：令 $\bar{B} = B \cup Z$ 。考虑实例 $\mathcal{I}' = (\mathbf{T} - Z, T \setminus Z, U, k - |Z|)$ 。在其辅助图 \mathbf{F}' 上， B 对应的顶点 x_B 是孤立点。依据引理6.25， $S' = S'_0 \cup S'_1$ 是 \mathcal{I}' 的 $T \setminus Z$ -反馈点集，其中 S'_0 为子实例 $\mathcal{I}'_0 = (\mathbf{T}[B], B \cap T, U, k)$ 的最优解，并且 S'_1 为实例 $\mathcal{I}'_1 = (\mathbf{T} - \bar{B}, T \setminus \bar{B}, U, k)$ 的最优解。注意到， B 为 U -分割块，所以 B 中无冲突边。又根据假设， U -约简规则以及冲突点约简规则均已不可执行。所以 \mathcal{I}'_0 不包含任何 T -三角形，这说明了 S'_0 为空。因此， S'_1 是 \mathcal{I}' 的最优解，进而表明 $S'_1 \cup Z$ 是实例 \mathcal{I} 的最优解。

下证 $S'_1 \cup Z$ 是实例 \mathcal{I} 的最优解。设 S 为实例 \mathcal{I} 的最优解。一方面， S'_1 是 \mathcal{I}' 的最优解意味着 $|S'_1| \leq |S \cap \bar{B}|$ 。另一方面，在导出子图 $\mathbf{T}[\bar{B}]$ 中，恰好包含了 $|Z|$ 条互不相交的冲突边，故 $|Z| \leq |S \cap \bar{B}|$ 。所以有

$$|S'_1 \cup Z| \leq |S'_1| + |Z| \leq |S \cap \bar{B}| + |Z| \leq |S|.$$

这意味着 $S'_1 \cup Z$ 是 \mathcal{I} 的最优解。

综上所述， Z 总在最优解中，且删掉 Z 之后，只需求解实例 $\mathcal{I}'_1 = (\mathbf{T} - \bar{B}, T \setminus \bar{B}, U, k)$ 即可。引理得证。 ■

接下来引入基于分割块的分支规则，称为最大度分支规则，此分支规则是本节算法的核心操作。

分支规则 6.4.5 (最大度分支规则)若 $2\sqrt{k}$ -均匀实例 \mathcal{I} 的辅助图 \mathbf{F} 的最大度超过 $k/\log^2 k$ ，任选一个度数最大的顶点 $x_B \in V(\mathbf{F})$ 及其对应的 U -分割块 B 。首先枚举 B 中不在解中的关键点，设它们构成的集合为 U' ($|U'| \leq 2\sqrt{k}$)。然后针对点集 B ，计算子分支：删去其余关键点 $(T \cap B) \setminus U'$ ，并使 k 减去 $|(T \cap B) \setminus U'|$ 。最后将集合 U' 加入 U ，更新冲突点和冲突边，依次执行冲突点约简规则、冲突边约简规则和分割块约简规则。若在上述三条规则执行过程中， k 的值减少了至少 $3k/\log^2 k$ ，直接返回无解。

最大度分支规则6.4.5是正确的，因为在 $2\sqrt{k}$ -均匀实例 \mathcal{I} 中， U -分割块 B 至多只有 $2\sqrt{k}$ 个关键点不在解中。所以，枚举 B 中所有大小不超过 $2\sqrt{k}$ 的关键点集合，一定能够找到恰好不在解中的所有关键点。注意到，在最大度分支规则6.4.5执

行完成后，实例依然是 $2\sqrt{k}$ -均匀的，并且 U -约简规则6.3.1，冲突点约简规则6.3.2，相交冲突边分支规则6.4.2，分割块约简规则6.4.4均可能会再次可执行。

显然，若冲突边约简规则不可执行则 $|E(\mathbf{F})| \leq k$ 成立，若最大度分支规则不可执行，则 $\Delta(\mathbf{F}) \leq k/\log^2 k$ 成立，此时借助引理6.26，便可以利用分治算法求解 $2\sqrt{k}$ -均匀的实例了。以下给出完整的求解算法。

算法 6-4 求解 $2\sqrt{k}$ -均匀的实例的基于平衡割技术的分支算法

输入: 一个 $2\sqrt{k}$ -均匀的实例 $\mathcal{I} = (\mathbf{T}, T, U, k)$ ，其中 $U \subseteq T$ 是关键点子集。

输出: 实例 \mathcal{I} 有解或无解

- 1 执行 U -约简规则6.3.1。// 此步骤后， $\mathbf{T}[U]$ 无 T -三角形。
- 2 执行冲突点约简规则6.3.2。// 此步骤后， \mathbf{T} 中无冲突点。
- 3 执行相交冲突边分支规则6.4.2。// 此步骤后， \mathbf{T} 中冲突边两两无公共端点。
- 4 执行冲突边约简规则6.4.3。// 此步骤后，辅助图 \mathbf{F} 的边数不超过 k 。
- 5 执行分割块约简规则6.4.4。// 此步骤后，任意分割块必包含关键点。
- 6 执行最大度分支规则6.4.5。// 此步骤后，辅助图 \mathbf{F} 最大度不超过 $k/\log^2 k$ 。
- 7 根据引理6.26直接求解实例 \mathcal{I} 并返回结果。

引理 6.28 算法6-4是正确的，且能够在 $\varphi^k n^{\mathcal{O}(\sqrt{k} \log^3 k)}$ 时间内求解 $2\sqrt{k}$ -均匀的实例，其中 $\varphi = (1 + \sqrt{5})/2$ 。

证明: 算法6-4中各规则的正确性已经论证，所以算法6-4是正确的。以下估计算法6-4的时间复杂度。首先， U -约简规则、冲突点约简规则和分割块约简规则可在多项式时间内完成。又由于这三条约简规则只有删点操作，所以它们最多执行 n 次。然后，冲突边约简规则可在多项式时间内完成，且最多执行 1 次。相交冲突边分支规则的分支向量为 $(1, 2)$ ，对应的分支因子为 $\varphi = (1 + \sqrt{5})/2$ 。

接下来，分析最大度分支规则。需要将该分支规则与后续的其他规则作为一个整体进行分析。设实例 $\mathcal{I} = (\mathbf{T}, T, U, k)$ 可以执行最大度分支规则，且前序步骤均不可执行。设 S 为实例 \mathcal{I} 的最优解。在最大度分支规则中，令其所操作的 U -分割块为 B ，其在辅助图对应的顶点 x_B 满足 $\deg_{\mathbf{F}}(x_B) = \Delta(\mathbf{F}) > k/\log^2 k$ 。首先，算法会枚举 B 中不在解中的关键点，设它们构成的集合为 U' ，则有 $|U'| \leq 2\sqrt{k}$ 。共有

$$\sum_{i=0}^{\lfloor 2\sqrt{k} \rfloor} \binom{|T \cap B|}{i} \leq (2\sqrt{k} + 1) \cdot \binom{n}{2\sqrt{k}} = n^{\mathcal{O}(\sqrt{k})}.$$

种可能性。然后，关键点集合 $(T \cap B) \setminus U'$ 将被删去， k 减少了 $|(T \cap B) \setminus U'|$ 。最后将集合 U' 加入 U 得到 $\bar{U} = U' \cup U$ ，并在多项式时间内更新冲突点和冲突边。由于当前实例是 $2\sqrt{k}$ -均匀的， B 中在解集里的顶点数量至多为 $2\sqrt{k} + k/\log^2 k \leq 3k/\log^2 k$ 。此时， $B \setminus T$ 成为多个新的 \bar{U} -分割块的并集，但这些分割块均不包含任何关键点。

设集合 $Z \subseteq V(\mathbf{T})$ B 是所有与 Z 中某一顶点构成了冲突边的顶点集，根据最大度分支规则的执行条件，有 $|Z| > k / \log^2 k$ 。令 $\bar{B} = Z \cup B$ ，直至分割块约简规则不可执行， \bar{B} 中所有顶点都将被删除。由于导出子图 $\mathbf{T}[\bar{B}]$ 恰好包含了 $|Z|$ 个两两无公共端点的冲突边，所以 \bar{B} 中至少包含了 $k / \log^2 k$ 个 S 中的顶点。因此，参数 k 至少减少了 $k / \log^2 k$ 。所以，在最大度分支规则中， k 至少需要减少 $k / \log^2 k$ 才可能再一次执行此规则。而剩下的分支规则仅有分支因子为 φ 的相交冲突边分支规则。所以，更新冲突点和冲突边后，依次执行冲突点约简规则、冲突边约简规则和分割块约简规则使得 k 至少减少了 $k / \log^2 k$ ，至多减少了 $3k / \log^2 k$ 。那么，整体的分支因子至多为

$$(\varphi^{3k / \log^2 k} \cdot n^{\mathcal{O}(\sqrt{k})})^{\log^2 k / k} = \varphi \cdot n^{\mathcal{O}(\log^2 k / \sqrt{k})} > \varphi.$$

最后，根据引理6.26，若 $\Delta(\mathbf{F}) \leq k / \log^2 k$ ，实例 \mathcal{I} 能够在 $1.5874^k \cdot n^{\mathcal{O}(\sqrt{k})}$ 时间内求解。

综上所述，存在整数 $k' \leq k$ 使得算法6-4得总运行时间为

$$\begin{aligned} & \left(\varphi \cdot n^{\mathcal{O}(\log^2 k / \sqrt{k})} \right)^{k'} \cdot 1.5874^{k-k'} \cdot n^{\mathcal{O}(\sqrt{k})} \\ & \leq \varphi^k \cdot n^{\mathcal{O}(\sqrt{k} \log^2 k)} \cdot n^{\mathcal{O}(\sqrt{k})} \\ & \leq \varphi^k \cdot n^{\mathcal{O}(\sqrt{k} \log^2 k)}. \end{aligned}$$

引理得证。 ■

定理 6.15 DSFVS(T) 问题可以在 $\mathcal{O}^*(1.6181^k)$ 时间内求解。

证明：结合引理6.20和引理6.28，可以得出 DSFVS(T) 问题可以在

$$\mathcal{O}^*(\varphi^k \cdot 2^{\mathcal{O}((\sqrt{k} \log^2 k + \sqrt{k}) \log k)}) = \mathcal{O}^*(\varphi^k \cdot 2^{\mathcal{O}(\sqrt{k} \log^3 k)}) \leq \mathcal{O}^*(1.6181^k).$$

的时间内求解，定理得证。 ■

推论 6.29 DSFVS(T) 问题可以在 $\mathcal{O}^*(1.6181^\tau)$ 时间内求解。

证明：当 $\tau \leq k$ 时，实例必然有解，可以在线性时间内判断。当 $\tau > k$ 时，根据定理6.15，可以在 $\mathcal{O}^*(1.6181^k) \leq \mathcal{O}^*(1.6181^\tau)$ 时间内求解。推论成立。 ■

定理 6.16 DSFVS(T) 问题可以在 $\mathcal{O}(1.3820^n)$ 时间内求解。

证明：基于单调局部搜索技术，得出 DSFVS(T) 问题可以在

$$\left(2 - \frac{1}{1.6181}\right)^n \cdot 2^{o(n)} = \mathcal{O}(1.3820^n)$$

时间内求解，定理得证。 ■

6.5 竞赛图上限制版子集反馈点集问题

本节将给出 R-DSFVS(T) 问题的参数算法与精确算法, 改进了当前最快的时间复杂度, 主要结论如下。

定理 6.30 R-DSFVS(T) 问题能在 $\mathcal{O}^*(1.2529^k)$ 时间内求解。

定理 6.31 R-DSFVS(T) 问题能在 $\mathcal{O}(1.1996^n)$ 时间内求解。

6.5.1 与顶点覆盖问题的归约关系

引理 6.32 R-DSFVS(T) 问题的实例 $\mathcal{I} = (\mathbf{T}, T, k)$ 可以多项式时间归约到 r -部图上顶点覆盖问题的实例 $\mathcal{I}' = (\mathbf{G}, k')$, 并且满足 $|V(\mathbf{G})| \leq |V(\mathbf{T})| - \tau$, $r \leq \tau + 1$ 和 $k' \leq k$ 。

证明: 给定 R-DSFVS(T) 问题的实例 $\mathcal{I} = (\mathbf{T}, T, k)$, 其中 $n = |V(\mathbf{T})|$ 且 $\tau = |T|$ 。首先给出归约算法构造顶点覆盖问题的实例 $\mathcal{I}' = (\mathbf{G}, k')$ 。不妨设 $\mathbf{T}[T]$ 是无环的, 否则 \mathcal{I} 无解, 可以直接归约到顶点覆盖问题的一个平凡无解实例。因为导出子图 $\mathbf{T}[T]$ 是无环的, 可在多项式时间内计算得到 $\mathbf{T}[T]$ 的拓扑序 $(t_1, t_2, \dots, t_\tau)$ 。定义

$$\begin{cases} B_0 = \{v \in V(\mathbf{T}) : t_1 \in N^+(v)\}; \\ B_i = \{v \in V(\mathbf{T}) : t_i \in N^-(v), t_{i+1} \in N^+(v)\}, \forall i \in [\tau - 1]; \\ B_\tau = \{v \in V(\mathbf{T}) : t_\tau \in N^-(v)\}. \end{cases}$$

并且令 $B = B_0 \cup B_1 \cup \dots \cup B_\tau$ 。对于每个 B_i , 构造点集 $V'_i = \{v' : v \in B_i\}$ 。然后得到

$$V(\mathbf{G}) = V'_1 \cup V'_2 \cup \dots \cup V'_{\tau-1}.$$

任意两个顶点 $v' \in V'_i$ 和 $u' \in V'_j$ 之间有一条无向边, 当且仅当 vu 是一条有向边且 $i > j$ 。即,

$$E(\mathbf{G}) = \{v'u' : vu \in E(\mathbf{T}), v \in B_i, u \in B_j, i > j\}.$$

最后设 $k' = k - n + |B|$, 进而得到了实例 $\mathcal{I}' = (\mathbf{G}, k')$ 。根据边集的构造可知, 无向图 \mathbf{G} 是 r -部图, 其中的每一部均为 $V'_0, V'_1, \dots, V'_\tau$ 中的非空集合之一, 所以有 $r \leq \tau + 1$ 。此外, 由于 B_i 中的顶点均为非关键点, 所以有

$$|V(\mathbf{G})| = \sum_{i=1}^{\tau-1} |V'_i| = \sum_{i=1}^{\tau-1} |B_i| \leq |V(\mathbf{T})| - \tau.$$

下证 \mathcal{I} 有解当且仅当 \mathcal{I}' 有解。

首先论证充分性：若 S 为实例 \mathcal{I} 的解，则

$$S' = \{v' \in V(\mathbf{G}) : v \in S \cap B\}$$

是 \mathcal{I}' 的解。考虑图 \mathbf{G} 中的任意一条边 $v'u'$ ($v' \in V'_i$ 且 $u' \in V'_j$)，根据构造，有 $i > j$ 且 vu 是一条有向边。不难发现， (t_i, v, u) 是 T -三角形，这意味着 v 和 u 至少其中之一在 S 中，也即说明了 v' 和 u' 至少其中之一在 S' 中。所以 S' 是图 \mathbf{G} 的顶点覆盖。注意到

$$|S'| = |S| - n + |B| \leq k - |V(\mathbf{T})| + |B| = k'.$$

所以， S' 是 \mathcal{I}' 的解。

接着论证必要性：若 S' 为实例 \mathcal{I}' 的解，则

$$S = \{v \in V(\mathbf{T}) : v' \in S'\} \cup (V(\mathbf{T}) \setminus B)$$

是 \mathcal{I} 的解。不难发现，有 $V(\mathbf{T}) \setminus S \subseteq B$ 。因此，若 T -三角形在 $\mathbf{T} - S$ 中，则其中的非关键点均来自某个 B_i 。设 $(t_{j'}, t_{j''}, v)$ 是 T -三角形 ($j', j'' \in [\tau]$)，其中 $v \in B_i$ 。由于 $t_{j'}t_{j''}$ 是一条有向边，所以 $j' > j''$ 。由 $t_{j'} \in N^+(v)$ 可知 $i \geq j'$ ；由 $t_{j''} \in N^-(v)$ 可知 $i < j''$ ，矛盾。设 (t_j, v, u) 是 T -三角形 ($j \in [\tau]$)，其中 $v \in B_{i'}$ 且 $u \in B_{i''}$ 。由 $t_j \in N^-(v)$ 可知 $i' < j$ ；由 $t_j \in N^+(u)$ 可知 $i'' \geq j$ 。这意味着 $i' < i''$ 。又因为 vu 是一条有向边，那么 $v'u'$ 是 \mathbf{G} 中的一条无向边，进而说明 v' 和 u' 至少其中之一在 S' 中，即是说 v 和 u 至少其中之一在 S 中。所以， $\mathbf{T} - S$ 中没有 T -三角形，故 S 是 \mathbf{T} 的 T -反馈点集。注意到

$$|S| = |S'| + n - |B| \leq k' + n - |B| = k.$$

所以， S 是 \mathcal{I} 的解，引理得证。 ■

引理 6.33 $(r+1)$ -部图上顶点覆盖问题的实例 $\mathcal{I}' = (\mathbf{G}, k')$ 可以多项式时间归约到 R-DSFVS(\mathbf{T}) 问题的实例 $\mathcal{I} = (\mathbf{T}, T, k)$ ，并且满足 $|V(\mathbf{T})| = |V(\mathbf{G})| + r$ ， $\tau = r$ 和 $k = k'$ 。

证明：设 \mathbf{G} 是 $(r+1)$ -部图，每一部分别为 V'_0, V'_1, \dots, V'_r 。构造图 \mathbf{T} 的点集

$$V(\mathbf{G}) = \{t_i : i \in [r]\} \cup \{v : v' \in V(\mathbf{G})\},$$

其中 $T = \{t_i : i \in [r]\}$ 是关键点集。令 $V_i = \{v \in V(\mathbf{G}) : v' \in V'_i\}$ 表示 V'_i 对应的竞赛图 \mathbf{T} 中的非关键点集。现在构造图 \mathbf{T} 的边集：

- 导出子图 $\mathbf{T}[V_i]$ ($i \in \{0, 1, \dots, r\}$) 为任意竞赛图。
- 导出子图 $\mathbf{T}[T]$ 为拓扑序是 (t_1, t_2, \dots, t_r) 的竞赛图。

- 对于任意关键点 t_i 和任意非关键点 $v \in V_i$ ($i \in [r]$, $j \in \{0, 1, \dots, r\}$), 若 $i \leq j$, 则添加有向边 $t_i v$; 若 $i < j$, 则添加有向边 $v t_i$ 。
- 对于任意两个非关键点 $v \in V_i$ 和 $u \in V_j$ 满足 $i < j$ ($i, j \in \{0, 1, \dots, r\}$), 若 \mathbf{G} 中 v' 和 u' 之间有一条边, 则添加有向边 uv ; 否则添加有向边 vu 。

最终得到

$$\begin{aligned} E(\mathbf{T}) = & \bigcup_{i=0}^r E(\mathbf{T}[V_i]) \cup \{t_i t_j : 1 \leq i < j \leq r\} \\ & \cup \{t_i v : v \in V_i, 1 \leq i \leq j \leq r\} \cup \{v t_i : v \in V_i, 0 \leq j \leq i \leq r\} \\ & \cup \{uv : u'v' \in E(\mathbf{G}), v \in V_i, u \in V_j, 0 \leq i \leq j \leq r\}. \end{aligned}$$

其中导出子图 $\mathbf{T}[V_i]$ ($i \in \{0, 1, \dots, r\}$) 为任意竞赛图。最后令 $k = k'$, 从而得到了实例 $\mathcal{I} = (\mathbf{T}, T, k)$ 。此外, 可以看出 $\tau = r$, 且

$$|V(\mathbf{T})| = |V(\mathbf{G})| + \tau.$$

下证 \mathcal{I} 有解当且仅当 \mathcal{I}' 有解。

首先论证充分性: 若 S 为实例 \mathcal{I} 的解, 则

$$S' = \{v \in V(\mathbf{T}) : v' \in S\}$$

是 \mathcal{I}' 的解。考虑图 \mathbf{G} 中的任意一条边 $v'u'$ ($v' \in V'_i$ 且 $u' \in V'_j$), 不妨设 $i > j$ 。此时, 图 \mathbf{T} 中存在有向边 vu , 所以 (t_i, v, u) 是一个 T -三角形。这意味着 v 和 u 至少有 1 个顶点在 S 中, 也即说明了 v' 和 u' 至少其中之一在 S' 。所以, S' 是图 \mathbf{G} 的顶点覆盖。注意到

$$|S'| = |S| \leq k = k'.$$

所以, S' 是 \mathcal{I}' 的解。

接着论证必要性: 若 S' 为实例 \mathcal{I}' 的解, 则

$$S = \{v \in V(\mathbf{T}) : v' \in S'\}$$

是 \mathcal{I} 的解。不难发现, \mathbf{T} 中的 T -三角形均包含两个非关键点。设 (t_j, v, u) 是 T -三角形 ($j \in [\tau]$), 其中 $v \in V_{i'}$ 且 $u \in V_{i''}$ 。根据构造, 有 $i' > i''$ 且 $v'u' \in E(\mathbf{G})$ 。所以, v' 和 u' 至少其中之一在 S' 中, 这意味着 v 和 u 至少其中之一在 S 中。所以, T -三角形 (t_j, v, u) 不在 $\mathbf{T} - S$ 中, 故 S 是 \mathbf{T} 的 T -反馈点集。注意到

$$|S| = |S'| \leq k' = k.$$

所以, S 是 \mathcal{I} 的解, 引理得证。 ■

定理 6.34 当关键点数量至多为 1 时, R-DSFVS(T) 问题是多项式时间可解的。当关键点数量是至少为 2 的常数时, R-DSFVS(T) 问题是 NP-难的。

证明: 设 $\mathcal{I} = (\mathbf{T}, T, k)$ 为 R-DSFVS(T) 问题的实例。若 $\tau \leq 1$, 根据引理 6.32, 可以在多项式时间内归约到二部图上的顶点覆盖问题。而二部图上的顶点覆盖问题可以在多项式时间内求解^[106, 107, 109]。所以, 当关键点数量至多为 1 时, R-DSFVS(T) 问题是多项式时间可解的。

另一方面, 根据引理 6.33, $(r + 1)$ -部图上的顶点覆盖问题可以多项式时间归约到关键集大小为 r 的 R-DSFVS(T) 问题的实例 $\mathcal{I} = (\mathbf{T}, T, k)$ 。由于三部图上的顶点覆盖问题是 NP-难的^[245], 所以当关键点数量为 2 时, R-DSFVS(T) 问题是 NP-难的。当 $\tau \geq 3$ 时, 可以从 $\tau = 2$ 的 R-DSFVS(T) 问题的实例进行归约: 添加 $\tau - 2$ 个孤立的关键点。综上所述, 关键点数量是至少为 2 的常数时, R-DSFVS(T) 问题是 NP-难的, 定理得证。 ■

6.5.2 基于归约的参数与精确算法

定理 6.35 R-DSFVS(T) 问题存在点数为 $3k$ 边数为 $4.5k^2 - 1.5k$ 核。

证明: 设 $\mathcal{I} = (\mathbf{T}, T, k)$ 为 R-DSFVS(T) 问题的实例, 设 \mathbf{T} 包含 $n = |V(\mathbf{T})|$ 个顶点。首先, 依据引理 6.32, 可以在多项式时间内将 \mathcal{I} 归约到顶点覆盖问题的实例 $\mathcal{I}_1 = (\mathbf{G}_1, k_1)$, 其中 $n_1 = |V(\mathbf{G}_1)| \leq n - \tau$ 且 $k_1 \leq k$ 。然后, 使用已有的顶点覆盖问题的核心化算法^[100, 101], 得到实例 \mathcal{I}_1 的线性核 $\mathcal{I}_2 = (\mathbf{G}_2, k_2)$, 其中 $n_2 = |V(\mathbf{G}_2)| \leq 2k_1$ 且 $k_2 \leq k_1$ 。反复检查 \mathbf{G}_2 中是否存在度数至少为 $k_2 + 1$ 的顶点 v , 若存在, 则其必然在解中可将 v 直接删除, 然后使 k_2 减少 1; 直至实例 $\mathcal{I}_3 = (\mathbf{G}_3, k_3)$, 使得 $\Delta(\mathbf{G}_3) \leq k_3$ 。上述操作是正确的, 若 v 不在解中, 则其所有邻居均在解中, 但是 v 有至少 $k_2 + 1$ 个邻居, 矛盾。可以得出 $n_3 = |V(\mathbf{G}_3)| \leq n_2$, 且 $k_3 \leq k_2$ 。由于图 \mathbf{G}_3 的最大度不超过 k_3 , 可以使用 $d = k_3 + 1$ 种颜色 (编号从 1 至 d) 对图 \mathbf{G}_3 贪心地点染色: 任选一个未染色的顶点 v , 由于其度数不超过 d , 总存在与它已染色的邻居全部不相同的颜色, 用此颜色对 v 进行染色。第 i 种颜色的顶点集合记作 V_i ($i \in [d]$)。那么 \mathbf{G}_3 是 d -部图, 并且每一部分别为 V_1, V_2, \dots, V_d (可能其中有一些集合为空)。根据引理 6.33, 可以在多项式时间内将 \mathcal{I}_3 归约到 R-DSFVS(T) 问题的实例 $\mathcal{I}' = (\mathbf{T}, T', k')$, 其中 $n' = |V(\mathbf{T}')| = n_3 + d - 1$, 并且 $k' = k_3$ 。

现分析实例 \mathcal{I}' 的大小, 根据五个实例 $\mathcal{I}, \mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3, \mathcal{I}'$ 的大小关系, 可以得出

$$n' = n_3 + d - 1 = n_3 + k_3 \leq n_2 + k_2 \leq 2k_1 + k_1 = 3k_1 \leq 3k,$$

且

$$k' \leq k_3 \leq k_2 \leq k_1 \leq k.$$

所以, 实例 \mathcal{I}' 是 R-DSFVS(T) 问题的点数为 $3k$, 边数为 $3k(3k-1)/2 = 4.5k^2 - 1.5k$ 的多项式核。定理得证。 ■

根据现有的结论, 除非 $\text{NP} \subseteq \text{coNP/poly}$, 对于任意正实数 $\varepsilon > 0$, 顶点覆盖问题不存在边数为 $\mathcal{O}(k^{2-\varepsilon})$ 的多项式核^[246]。所以, 引理6.33表明 R-DSFVS(T) 问题的核心化存在下界。

定理 6.36 除非 $\text{NP} \subseteq \text{coNP/poly}$, 对于任意正实数 $\varepsilon > 0$, R-DSFVS(T) 问题不存在点数为 $\mathcal{O}(k^{1-\varepsilon})$ 和边数为 $\mathcal{O}(k^{2-\varepsilon})$ 的核。

定理6.35和定理6.36共同说明了 R-DSFVS(T) 问题关于点数的线性核是紧的。

同样依赖于引理6.32和引理6.33, 接下来将证明在指数时间的意义下, R-DSFVS(T) 问题和顶点覆盖问题的参数和精确算法的时间复杂度是相同的。

引理 6.37 对于任意实数 $\alpha > 1$, R-DSFVS(T) 问题能在 $\mathcal{O}^*(\alpha^k)$ 时间内求解当且仅当顶点覆盖问题能在 $\mathcal{O}^*(\alpha^{k'})$ 时间内求解, 其中 k 和 k' 分别为 R-DSFVS(T) 问题和顶点覆盖问题实例的解集大小。

证明: 首先证明必要性: 若顶点覆盖问题的实例 $\mathcal{I}' = (\mathbf{G}, k')$ 能在 $\mathcal{O}^*(\alpha^{k'})$ 时间内求解 ($\alpha > 1$), 则 R-DSFVS(T) 问题的实例 $\mathcal{I} = (\mathbf{T}, T, k)$ 能在 $\mathcal{O}^*(\alpha^k)$ 时间内求解。依据引理6.32, 实例 \mathcal{I} 可在多项式时间内归约到 \mathcal{I}' , 满足 $k' \leq k$ 。所以可以调用顶点覆盖问题的参数算法在 $\mathcal{O}^*(\alpha^{k'}) = \mathcal{O}^*(\alpha^k)$ 时间内求解判定 \mathcal{I} 是否有解。

其次证明充分性: 若 R-DSFVS(T) 问题实例 \mathcal{I} 能在 $\mathcal{O}^*(\alpha^k)$ 时间内求解 ($\alpha > 1$), 则顶点覆盖问题的实例 \mathcal{I}' 能在 $\mathcal{O}^*(\alpha^{k'})$ 时间内求解。设 $\mathcal{I}' = (\mathbf{G}, k)$ 。注意到 α 是常数, 故存在足够大的常数 d , 使得分支向量 $(1, d)$ 的分支因子不超过 α 。根据图 \mathbf{G} 的最大度, 考虑两种情形。

情形 1: $\Delta(\mathbf{G}) \geq d$ 。执行以下分支规则以处理度数至少为 d 的顶点: 选择一个度数至少为 d 的顶点 $v' \in V(\mathbf{G})$, 计算以下两个分支。若两个实例均无解, 则 \mathcal{I}' 无解, 否则 \mathcal{I}' 有解。

- 删去 v' , 并使 k 减少 1;
- 删去 $N[v']$, 并使 k 减少 $\deg(v')$ 。

此分支规则是正确的, 因为每条边都至少有 1 个端点在解集中, 并且关于度量 k 的分支向量不劣于 $(1, d)$ 。

情形 2: $\Delta(\mathbf{G}) \leq d - 1$ 。使用 d 种颜色 (编号从 1 至 d) 对图 \mathbf{G} 贪心地点染色: 任选一个未染色的顶点 v' , 由于其度数至多为 d , 总存在与它已染色的邻居全部不相同的颜色, 用此颜色对 v' 进行染色。第 i 种颜色的顶点集合记作 V'_i

($i \in [d]$)。那么 \mathbf{G} 是 d -部图，并且每一部分别为 V'_1, V'_2, \dots, V'_d (可能其中有一些集合为空)。根据引理6.33，可以在多项式时间内将 \mathcal{I}' 归约到 R-DSFVS(T) 问题的实例 $\mathcal{I} = (\mathbf{T}, T, k)$ 。最后，在 $\mathcal{O}^*(\alpha^k)$ 时间内求解实例 \mathcal{I} 。

由于情形 1 中分支规则的分支因子不超过 α ，所以实例 \mathcal{I}' 能在 $\mathcal{O}^*(\alpha^{k'})$ 。

综上所述，引理得证。 ■

引理 6.38 对于任意实数 $\alpha > 1$ ，R-DSFVS(T) 问题能在 $\mathcal{O}(\alpha^n)$ 时间内求解当且仅当顶点覆盖问题能在 $\mathcal{O}(\alpha^{n'})$ 时间内求解，其中 n 和 n' 分别为 R-DSFVS(T) 问题和顶点覆盖问题实例的顶点数量。

证明：首先证明必要性：若顶点覆盖问题的包含 n' 个顶点的实例 $\mathcal{I}' = (\mathbf{G}, k')$ 能在 $\mathcal{O}(\alpha^{n'})$ 时间内求解 ($\alpha > 1$)，则 R-DSFVS(T) 问题的包含 n 个的实例 $\mathcal{I} = (\mathbf{T}, T, k)$ 能在 $\mathcal{O}(\alpha^n)$ 时间内求解。依据引理6.32，实例 \mathcal{I} 可以在多项式时间内归约到顶点覆盖问题的实例 \mathcal{I}' ，满足 $n' = |V(\mathbf{G})| \leq n - \tau \leq n$ 。所以可以调用顶点覆盖问题的精确算法在 $\mathcal{O}(\alpha^{n'}) = \mathcal{O}(\alpha^n)$ 时间内求解 \mathcal{I} 。

其次证明充分性：若 R-DSFVS(T) 问题能在 $\mathcal{O}(\alpha^n)$ 时间内求解 ($\alpha > 1$)，则顶点覆盖问题能在 $\mathcal{O}(\alpha^{n'})$ 时间内求解。设 $\mathcal{I}' = (\mathbf{G}, k)$ 为顶点覆盖问题的实例，且 \mathbf{G} 包含了 $n' = |V(\mathbf{G})|$ 个顶点。注意到 α 是常数，故存在足够大的常数 d ，使得分支向量 $(1, d + 1)$ 的分支因子不超过 α 。根据图 \mathbf{G} 的最大度，考虑两种情形。

情形 1： $\Delta(\mathbf{G}) \geq d$ 。执行以下分支规则以处理度数至少为 d 的顶点：选择一个度数至少为 d 的顶点 $v' \in V(\mathbf{G})$ ，计算以下两个分支。若两个实例均无解，则 \mathcal{I}' 无解，否则 \mathcal{I}' 有解。

- 删去 v' ，并使 k 减少 1，返回实例 $(\mathbf{G} - v', k - 1)$ ；
- 删去 $N[v']$ ，并使 k 减少 $\deg(v')$ ，返回实例 $(\mathbf{G} - N[v'], k - \deg(v'))$ 。

显然此分支规则是正确的，因为每条边都至少有 1 个端点在解集中，并且关于度量 n' 的分支向量不劣于 $(1, d + 1)$ 。

情形 2： $\Delta(\mathbf{G}) \leq d - 1$ 。使用 d 种颜色 (编号从 1 至 d) 对图 \mathbf{G} 贪心地点染色：任选一个未染色的顶点 v' ，由于其度数至多为 d ，总存在与它已染色的邻居全部不相同的颜色，用此颜色对 v' 进行染色。第 i 种颜色的顶点集合记作 V'_i ($i \in [d]$)。那么 \mathbf{G} 是 d -部图，并且每一部分别为 V'_1, V'_2, \dots, V'_d (可能其中有一些集合为空)。根据引理6.33，可以在多项式时间内将 \mathcal{I}' 归约到 R-DSFVS(T) 问题的实例 $\mathcal{I} = (\mathbf{T}, T, k)$ ，满足 $|V(\mathbf{T})| = |V(\mathbf{G})| + d$ 。最后，在 $\mathcal{O}(\alpha^n) = \mathcal{O}(\alpha^{n'+d})$ 时间内求解 \mathcal{I} 。

由于情形 1 中分支规则的分支因子不超过 α ，并且 d 是依赖于 α 的常数，所以 \mathcal{I}' 能在 $\mathcal{O}(\alpha^{n'+d}) = \mathcal{O}(\alpha^{n'})$ 时间内求解。

综上所述, 引理得证。 ■

定理 6.30 R-DSFVS(T) 问题能在 $\mathcal{O}^*(1.2529^k)$ 时间内求解。

证明: 顶点覆盖问题最快参数算法的运行时间为 $\mathcal{O}^*(1.2529^k)$ ^[98]。结合引理6.37, R-DSFVS(T) 问题能在 $\mathcal{O}^*(1.2529^k)$ 时间内求解, 定理得证。 ■

定理 6.31 R-DSFVS(T) 问题能在 $\mathcal{O}(1.1996^n)$ 时间内求解。

证明: 顶点覆盖问题最快精确算法的运行时间为 $\mathcal{O}(1.1996^n)$ ^[34]。结合引理6.38, R-DSFVS(T) 问题能在 $\mathcal{O}(1.1996^n)$ 时间内求解, 定理得证。 ■

6.6 本章小结

本章深入研究了竞赛图上的子集反馈点集问题及其限制版本。

针对 DSFVS(T) 问题的参数算法, 本章扩展并结合了多种前沿技术。具体地, 本章首先利用解集分离技术, 在亚指数时间内获得一个性质足够好且与某一最优解不相交的关键点子集。然后本章引入了冲突边的概念, 通过多种分支技术, 逐步减弱关于冲突边的连通性。最后采用平衡割技术得到了运行时间为 $\mathcal{O}^*(1.6181^k)$ 的参数算法。该算法时间复杂度首次突破 2^k , 且该算法时间复杂度与 DFVS(T) 问题当前最快参数算法的时间复杂度持平。这意味着, 若 DFVS(T) 问题的参数算法时间复杂度无法得到进一步优化, 那么 DSFVS(T) 问题的参数算法也将无法进一步得到改进。

值的强调的是, 本章的核心创新点在于引入了解集分离技术以及基于分割块的分支搜索技术。采用这两个技术的时间消耗是亚指数的, 理论上的效果显著的, 但是其时间复杂度十分接近单指数, 在解决实际问题时, 需要考虑结合其他实验算法技术。

此外, 本章研究了 R-DSFVS(T) 问题, 证明了即使在仅有两个关键点的条件下, R-DSFVS(T) 问题也是 NP-难的。同时本章建立了 R-DSFVS(T) 问题和顶点覆盖问题的联系, 发现能够在多项式时间内将顶点覆盖问题多项式时间归约到 R-DSFVS(T) 问题, 并且其中仅包含常数数量的关键点。这一个归约明确刻画了 R-DSFVS(T) 问题和顶点覆盖问题的高度相似性。因此, 本章说明了 R-DSFVS(T) 问题与顶点覆盖问题存在相同指数时间的参数与精确算法, 也证明了 R-DSFVS(T) 问题存在线性核, 且在基本计算复杂性假设下达到了理论最优。

第七章 分裂图上子集反馈点集问题的参数与精确算法

分裂图是构成了众多关键图类的基础子结构，在社交网络和系统生物学（Systems Biology）等领域也发挥了显著作用。本章聚焦于分裂图上子集反馈点集问题的参数算法和精确算法研究，改进了当前最快算法结果。特别地，对于分裂图上的非限制版子集反馈点集问题，本文提出的参数算法首次突破了 2^k 的计算复杂度瓶颈。此外，本章将这一参数算法应用于超图上带惩罚的独立集问题，提出了首个非平凡的精确算法，成功突破 2^n 。

7.1 研究背景与主要贡献

分裂图是一种能够将点集划分为一个独立集和一个团的无向图。分裂图结构的简洁性并不掩盖其在理论研究与实践应用上的重要性。Földes 和 Hammer^[247,248]以及 Tyshkevich 和 Chernyak^[249]在 20 世纪 70 年代的工作奠定了分裂图在图论中的重要地位。Trotter^[250]进一步刻画了分裂图的重要特征：一个图是分裂图当且仅当该图及其补图都是弦图。分裂图可以视为星图上子树的相交图模型，而弦图是树的子图的相交模型，这一结构性质使得分裂图成为理解弦图及其基于树分解上动态规划算法的关键。分裂图在社交网络^[251,252]和系统生物学^[253]中的广泛应用也体现了其在解决实际问题方面的巨大潜力。

根据分裂图的定义，分裂图上的独立集问题和团问题能够在多项式时间内求解。值得注意的是，分裂图甚至弦图上的反馈点集问题是多项式可计算的^[210]；但分裂图上的子集反馈点集问题，无论是非限制版还是限制版均被证明是 NP-难的^[176,177]。本章将分裂图上非限制版子集反馈点集问题简记为 SFVS(S)；分裂图上限制版子集反馈点集问题简记为 R-SFVS(S)。对 SFVS(S) 问题和 R-SFVS(S) 问题的研究有助于理解子集反馈集问题与反馈集问题在计算复杂性上的差异，这凸显了它们在计算复杂性领域的研究价值。SFVS(S) 问题和 R-SFVS(S) 问题的具体定义如下。

分裂图上非限制版子集反馈点集问题 (SFVS(S))

输入：一个分裂图 \mathbf{G} ，一个大小为 τ 的关键点集 $T \subseteq V(\mathbf{T})$ 和一个自然数 $k \in \mathbb{N}$ 。

输出：判断分裂图 \mathbf{G} 是否存在一个大小不超过 k 的 T -反馈点集 $S \subseteq V(\mathbf{G})$ 。

分裂图上限制版子集反馈点集问题 (R-SFVS(S))

输入: 一个分裂图 \mathbf{G} , 一个大小为 τ 的关键点集 $T \subseteq V(\mathbf{G})$ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断分裂图 \mathbf{G} 是否存在一个大小不超过 k 的 T -反馈点集 $S \subseteq V(\mathbf{G}) \setminus T$ 。

SFVS(S) 问题可以视为超图上带惩罚的独立集问题的一种推广。超图上带惩罚的独立集问题是独立集问题的松弛化, 简记为 PC-MIS(H)。超图 (hypergraph) 是经典图模型的延伸, 其中的超边允许连接任意数量的顶点。相比之下, 经典无向图中的一条边恰好连接两个顶点。超图中的独立集是一些两两不包含于同一超边的顶点构成的点集。超图上的独立集问题 (Maximum Independent Set Problem on Hypergraphs), 简记作 MIS(H), 旨在寻找一个最大的独立集。而 PC-MIS(H) 问题是 MIS(H) 问题的松弛化推广, 更具有普适性和实用性。PC-MIS(H) 问题旨在找到一个最大价值的点集, 其中, 一个点集的价值定义为该点集大小和违背独立性限制的超边数量之差。直观地讲, PC-MIS(H) 问题需要平衡解集大小与其违背了独立性的超边的数量。MIS(H) 问题和 PC-MIS(H) 问题的定义如下。

超图上独立集问题 (MIS(H))

输入: 一个超图 \mathbf{H} 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在大小不超过 k 的独立集 $S \subseteq V(\mathbf{H})$ 。

超图上带惩罚的独立集问题 (PC-MIS(H))

输入: 一个超图 \mathbf{H} 和一个自然数 $p \in \mathbb{N}$ 。

输出: 判断是否存在价值不超过 p 的点集 $S \subseteq V(\mathbf{H})$ 。

若将 PC-MIS(H) 问题实例中的每一条超边复制足够多份后, 该实例则退化为 MIS(H) 问题的一个实例。

事实上, 许多重要而基础的问题都存在带惩罚的版本并得到了深入研究, 例如带惩罚的旅行商问题 (Prize-Collecting Traveling Salesman Problem)^[254], 带惩罚的斯坦纳树问题 (Prize-Collecting Steiner Tree Problem)^[255,256] 带惩罚的网络激活问题 (Prize-Collecting Network Activation Problem)^[257,258]。这类问题的研究不仅展现了理论计算机领域探索可计算性问题的持续努力, 也为解决复杂的实际问题提供了新的方法和思路。

研究现状与挑战 SFVS(S) 问题是一种隐式 3-碰撞集问题^[80]。因此, 该问题存在点集大小为 $\mathcal{O}(k^2)$ 的核^[240,241], 并且可以在 $\mathcal{O}^*(2.0755^k)$ 时间内求解^[54]。最近, Philip 等人^[80]针对弦图上子集反馈点集问题, 提出了更快速的参数算法, 能够将

时间复杂度降低至 $\mathcal{O}^*(2^k)$ 。尽管他们的工作聚焦于更一般化的弦图，但所采用的分支搜索技术总是对弦图的某个局部子结构进行搜索。由于弦图的局部子结构均可视为分裂图，意味着即便应用于 SFVS(S) 问题上，其时间复杂度亦无法进一步降低。此外，突破 2^k 的时间复杂度瓶颈对于 SFVS(S) 问题而言尤为困难。一方面，Philip 等人提出的分支搜索算法极其复杂，其中两个分支规则包含多达 7 个子分支，仅靠深入的结构分析和情况讨论难以取得实质性突破。另一方面，分裂图的结构简单，但在此结构上的诸多经典问题的依然是 NP-难的，其中包括支配集问题^[259]，哈密顿问题^[260,261]等等。分裂图的简洁性导致算法难以挖掘出更多可利用的图结构，故为分裂图上这些问题设计快速的参数和精确算法仍面临着巨大挑战。因此，得到快于 $\mathcal{O}^*(2^k)$ 的参数算法面临着诸多障碍。

就独立集问题而言，MIS(H) 问题能在 $\mathcal{O}(1.1996^n)$ 时间内求解^[34]。相比之下，求解 PC-MIS(H) 问题目前最快的算法仍然是运行时间为 $\mathcal{O}^*(2^n)$ 的穷举搜索算法，尚未发现任何突破 2^n 的精确算法。值得注意的是，SFVS(S) 问题的 $\mathcal{O}^*(\alpha^k)$ 时间参数算法蕴含着 PC-MIS(H) 问题的 $\mathcal{O}^*(\alpha^n)$ 时间精确算法。遗憾的是，尽管针对 SFVS(S) 问题，目前最快的参数算法设计精巧且复杂，但无法为 PC-MIS(H) 问题突破 2^n 时间复杂度瓶颈提供有效帮助。

在 SFVS(S) 问题的精确算法方面，Golovach 等人^[262,263]在 2012 年给出了运行时间为 $\mathcal{O}(1.6708^n)$ 的精确算法。随后，Chitnis 等人^[58]与 Fomin 等人^[97]分别将运行时间改进到了 $\mathcal{O}(1.6181^n)$ 和 $\mathcal{O}(1.5182^n)$ 。目前为止，最快的精确算法时间复杂度达到了 $1.5^n 2^{o(n)}$ 。由于这些算法均为分支搜索算法，因此它们同时能够适用于分裂图上的限制版和非限制版的子集反馈点集问题。令人惊讶的是，在多个团队先后进行了研究之后，弦图和分裂图上的子集反馈点集问题的精确算法运行时间上界一直保持一致，分裂图更加简单的结构没有带来更加高效的精确算法。

主要贡献 本章系统性地开展了 SFVS(S) 问题和 R-SFVS(S) 问题的算法研究，提出了当前最快的参数与精确算法，并给出了 R-SFVS(S) 问题的核心化算法。本章同时建立了 SFVS(S) 问题与 PC-MIS(H) 问题之间的联系，为 PC-MIS(H) 问题提出了首个非平凡的精确算法。更重要的是，本章的算法结果为第八章研究弦图上子集反馈点集问题奠定了重要的基础。

本章的核心贡献在于采用 Dulmage-Mendelsohn 分解刻画图的结构性质，并基于该性质创新性地引入了一种非传统度量函数，以此开展时间复杂度分析。这一方法论的应用为 SFVS(S) 问题提供了运行时间为 $\mathcal{O}^*(1.8192^k)$ 的参数算法，首次突破 2^k 。相较而言，以传统的解集大小 k 为度量的分析方法将陷入过于庞大的分类讨论而难以奏效。此外，本章通过建立 SFVS(S) 与 PC-MIS(H) 问题的桥梁，将

PC-MIS(H) 问题的精确求解时间降至 $\mathcal{O}^*(1.8192^n)$ ，首次突破了 2^n 时间复杂度的关键性瓶颈。基于均摊思想，本章进一步改进了 SFVS(S) 问题的精确算法运行时间上界，证明了 SFVS(S) 问题能在 $\mathcal{O}(1.3488^n)$ 时间内求解。该算法的正确性高度依赖分裂图的结构特征，不能直接适用于弦图上的子集反馈点集问题。

针对 R-SFVS(S) 问题，本章建立了该问题和顶点覆盖问题之间紧密的归约关系。基于此归约关系，本章给出了一个顶点数为 $k^2 + 2k$ 且边数为 $5k^2$ 的平方核，并且证明了在基本复杂度假设下，核的大小是渐近紧的。基于同样的归约关系，顶点覆盖问题和 R-SFVS(S) 问题的参数算能够在相同的指数时间复杂度内求解，这意味着 R-SFVS(S) 问题能在 $\mathcal{O}^*(1.2529^k)$ 时间内求解。此外，本章通过精细的度量治之分析，为 R-SFVS(S) 问题给出了比求解顶点覆盖问题更快的精确算法，其时间复杂度为 $\mathcal{O}(1.1520^n)$ ，刷新了当前精确算法的最佳纪录。此外，本章为 R-SFVS(S) 问题提供了一个以关键点数量 τ 为参数的参数算法，其运行时间上界 $\mathcal{O}^*(2^\tau)$ 。作为直接的推论，本章证明了独立集问题以团覆盖数为参数是 FPT 的。主要结果见表 7-1。

表 7-1 分裂图上子集反馈点集问题的主要算法结果

	SFVS(S) 问题	R-SFVS(S) 问题
核心化（顶点数）	$\mathcal{O}(k^2)$ (文献 [240, 241])	$k^2 + 2k$ (定理 7.27)
参数算法（以 k 为参数）	$\mathcal{O}^*(1.8192^k)$ (定理 7.4)	$\mathcal{O}^*(1.2529^k)$ (定理 7.23)
参数算法（以 τ 为参数）	$\mathcal{O}^*(1.8192^\tau)$ (推论 7.20)	$\mathcal{O}^*(2^\tau)$ (定理 7.31)
精确算法	$\mathcal{O}(1.3488^n)$ (定理 7.5)	$\mathcal{O}(1.1520^n)$ (定理 7.24)

7.2 分裂图的基本结构与性质

如果图 \mathbf{G} 的点集 $V(\mathbf{G})$ 存在划分 (I, K) 使得 I 是一个独立集且 K 是一个团，则称 \mathbf{G} 为分裂图，其中划分 (I, K) 称为分裂图 \mathbf{G} 的分裂划分 (Split Partition)。注意到分裂图的分裂划分不一定是唯一的。但分裂图的分裂划分可以在线性时间内找到，且分裂图识别问题可在线性时间内解决^[247]。给定分裂图 \mathbf{G} 的分裂划分 (I, K) ， \mathbf{G} 中的极大团至多只有 $|I| + 1$ 个，且下述三种情况必满足其一。

- 存在一个顶点 $x \in I$ 使得 $K \cup \{x\}$ 是一个团，此时， $K \cup \{x\}$ 是一个最大团且 I 是一个最大独立集。
- 存在一个顶点 $x \in K$ 使得 $I \cup \{x\}$ 是一个独立集，此时 $I \cup \{x\}$ 是一个最大独立集且 K 是一个最大团。
- K 一个极大团且 I 是一个极大独立，此时 (I, K) 是图 \mathbf{G} 唯一的分裂划分。

分裂图是一类禁止子图图类，一个图是分裂图当且仅当没有以下三个导出子

图：一对不相交的边 $2\mathbf{K}_2$ ，4长环 \mathbf{C}_4 和 5长环 \mathbf{C}_5 ^[247]。所以，分裂图的导出子图也是分裂图，这意味着，从分裂图中删去一些顶点得到的子图也是分裂图。值得强调的是，从分裂图中删去一些边后，得到的可能不再是分裂图。

引理7.1表明 SFVS(S) 问题是隐式 3-碰撞集问题。由于分裂图是弦图的子类，所以引理7.1已在文献 [80] 中给出。

引理 7.1 设 \mathbf{G} 为分裂图， $T \subseteq V(\mathbf{G})$ 为 \mathbf{G} 的关键点集。 \mathbf{G} 是 T -无环的当且仅当 \mathbf{G} 不包含任何 T -三角形。

基于引理7.1，在设计算法时，只需要将目标确定为删除顶点以碰撞给定分裂图中所有 T -三角形即可。

引理 7.2 设 \mathbf{G} 为分裂图， $T \subseteq V(\mathbf{G})$ 为 \mathbf{G} 的关键点集。令 (I, K) 为图 \mathbf{G} 的分裂划分。满足以下条件之一的最小 T -反馈点集至多有 $n(n+1)/2$ 个。

- 包含了 I 中的某个非关键点；
- 没有包含 K 中的某个关键点。

并且可以在多项式时间内枚举满足此条件的所有最小 T -反馈点集。

证明：设 S 是 \mathbf{G} 的最小 T -反馈点集。若 S 包含了非关键点 $v \in I$ ，那么必然存在关键点 $t \in K$ 不属于 S ；否则 $S \setminus \{v\}$ 也是 T -反馈点集，与 S 的最小性矛盾。因此，只需要考虑没有包含 K 中某个关键点的最小 T -反馈点集数量。

假设 S 没有包含某个关键点 $t \in K$ ，那么 K 中至多存在另外一个不在 S 中的顶点，否则 $K \setminus S$ 中的任意两个顶点将与 t 构成 T -三角形，矛盾。所以 $|K \setminus S| \leq 2$ 。此外，不难发现，给定集合 $K \setminus S$ 的最小 T -反馈点集 S 是唯一的。

最后， T -反馈点集的大小不会超过 $|K| - 1$ ，这是因为，删除 K 中任意 $|K| - 1$ 个顶点将得到一个星图。综上所述，满足引理条件的最小 T -反馈点集至多有

$$\binom{|K|}{1} + \binom{|K|}{2} \leq n + \frac{1}{2}n(n-1) = \frac{1}{2}n(n+1).$$

并且可以在多项式时间内进行枚举。引理得证。 ■

基于引理7.2，给定分裂划分 (I, K) ，图 \mathbf{G} 的最小 T -反馈点集中，包含了 I 中的某个非关键点或没有包含 K 中的某个关键点的数量只有 $\mathcal{O}(n^2)$ 个。由于此类最小 T -反馈点集可以在多项式时间内验证进而可以简化分裂图的结构，因此这里引入 T -可分的概念。

定义 7.3 (T -可分) 设 \mathbf{G} 为分裂图， $T \subseteq V(\mathbf{G})$ 为 \mathbf{G} 的关键点集。若 $(T, V(\mathbf{G}) \setminus T)$ 是图 \mathbf{G} 的分裂划分，则称 \mathbf{G} 是 T -可分的。

若 SFVS(S) 问题或 R-SFVS(S) 问题实例的输入图是 T -可分的，那么也称该实例是 T -可分的。

7.3 分裂图上非限制版子集反馈点集问题

本节将提出 SFVS(S) 问题的参数算法，改进了当前最快的 $\mathcal{O}^*(2^k)$ 时间的参数算法。以参数算法的基础，本节将进一步给出 SFVS(S) 问题的精确算法，改进了当且最快的 $1.5^n 2^{o(n)}$ 时间的精确算法。主要结论如下。

定理 7.4 SFVS(S) 问题可以在 $\mathcal{O}^*(1.8192^k)$ 时间内求解。

定理 7.5 SFVS(S) 问题可以在 $\mathcal{O}(1.3488^n)$ 时间内求解。

SFVS(S) 问题的参数算法包括三个部分。

第一部分基于 Dulmage-Mendelsohn 分解技术，提出了 Dulmage-Mendelsohn 约简规则。

第二部分引入了衡量实例的非传统度量 $\mu(\cdot)$ 并详尽地分析了度量函数关于图上基本操作的变化量，这里度量 $\mu(\cdot)$ 不仅与解集大小 k 有关，也依赖于 Dulmage-Mendelsohn 分解结构。

第三部分设计了基于分支搜索技术的参数算法，成功突破了 2^k 的瓶颈，得到了运行时间为 $\mathcal{O}^*(1.8192^k)$ 的参数算法。

最后，利用分裂图的分裂划分，为 SFVS(S) 问题设计了更加快速的精确算法，得到了运行时间为 $\mathcal{O}(1.3488^n)$ 的参数算法。

7.3.1 基于 Dulmage-Mendelsohn 分解的约简规则

本节将介绍二部图上的 Dulmage-Mendelsohn 分解^[111, 112]，并引入基于该分解的重要约简规则，称作 **Dulmage-Mendelsohn 约简规则**，简称 DM 约简规则。DM 约简规则是后文求解 SFVS(S) 问题参数算法的核心操作。

定义 7.6 (Dulmage-Mendelsohn 分解^[264, 265]) 令 \mathbf{F} 为二部图，其中 A 和 B 分别为点集 $V(\mathbf{F})$ 两个独立集部分。图 \mathbf{F} 的 **Dulmage-Mendelsohn 分解**是满足以下条件的点集 $V(\mathbf{F})$ 的三划分 (C, H, R) (参见图7-1)。

- 点集 C 是独立集且 $H = N(C)$ ；
- 点集 R 导出的子图 $\mathbf{F}[R]$ 存在完美匹配；
- 点集 H 是图 \mathbf{F} 所有最小顶点覆盖的交集；
- 点集 $R \cup H$ 是存在于 \mathbf{F} 的某个最大匹配的顶点全体。

事实上，Dulmage-Mendelsohn 分解是皇冠分解的变体，但是该分解是唯一的，具有更细致的结构特征。二部图 \mathbf{F} 的 Dulmage-Mendelsohn 分解可以通过在 $\mathcal{O}(m\sqrt{n})$ 时间内计算 \mathbf{F} 的任一最大匹配得到^[109]。

为了更好地使用 Dulmage-Mendelsohn 分解进行刻画，需要对 SFVS(S) 问题实例中的一些边进行标记，然后构建由部分标记边组成的二部图，并进行 Dulmage-

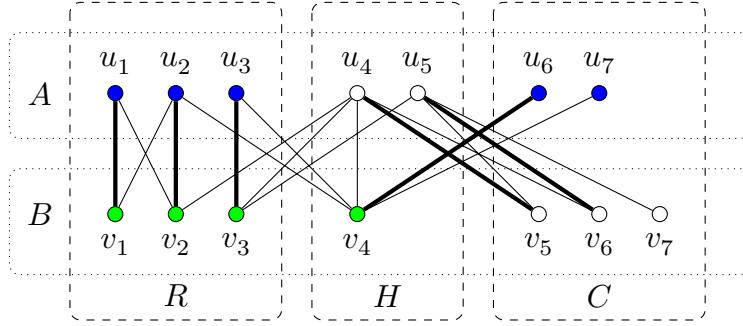


图 7-1 二部图 \mathbf{F} 的 Dulmage-Mendelsohn 分解示意图。二部图 \mathbf{F} 的两个独立集为 $A = \{u_i\}_{i=1}^7$ 和 $B = \{v_i\}_{i=1}^7$ 粗实线为图 \mathbf{F} 的一个最大匹配。图 \mathbf{F} 的 Dulmage-Mendelsohn 分解为 (C, H, R) , 其中 $C = \{u_6, u_7, v_5, v_6, v_7\}$, $H = \{u_4, u_5, v_4\}$, $R = \{u_1, u_2, u_3, v_1, v_2, v_3\}$ 。如果 \mathbf{F} 是 SFVS(S) 问题实例 \mathcal{I} 的辅助图, 那么有 $\hat{A} = \{u_1, u_2, u_3, u_6, u_7\}$ (蓝色顶点) 以及 $\hat{B} = \{v_1, v_2, v_3, v_4\}$ (绿色顶点)。

Mendelsohn 分解。因此, 需要定义 SFVS(S) 问题的广义版本, 称为分裂图上广义子集反馈点集问题 (Generalized Subset Feedback Vertex Set Problems in Split Graphs), 简记为 Generalized-SFVS(S)。具体定义如下。

分裂图上广义子集反馈点集问题 (Generalized-SFVS(S))

输入: 一个分裂图 \mathbf{G} , 一个关键点集 $T \subseteq V(\mathbf{G})$, 一条标记边集 $M \subseteq E(\mathbf{G})$, 一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在一个大小不超过 k 的 T -反馈点集 S , 使得 M 中的任何一条边都至少存在一个端点在 S 中。

在实例中加入标记边的考量来源于一个自然的观察: 在一个有解实例中, 设 (t, v, u) 为包含了 2 度关键点 t 的 T -三角形。不难发现, 总存在不包含 t 而包含 t 的某个邻居的解, 因此总可以将 t 删掉, 然后将边 uv 进行标记, 得到 Generalized-SFVS(S) 问题的等价实例。相反, 也可以对每条标记边添加一个 2 度关键点, 然后去掉所以边的标记, 从而也可以得到一个 SFVS(S) 问题的一个等价实例。以上事实表明了 Generalized-SFVS(S) 问题和 SFVS(S) 问题具有非常直观地等价性。后文将直接针对 Generalized-SFVS(S) 问题设计参数算法。

考虑 Generalized-SFVS(S) 问题的实例 $\mathcal{I} = (\mathbf{G}, T, M, k)$ 。令 (I, K) 为图 \mathbf{G} 的分裂划分, 其中 I 是独立集, K 是团。基于分裂划分 (I, K) , 可以唯一地确定实例 \mathcal{I} 的一个辅助图 \mathbf{F} 。辅助图 \mathbf{F} 是图 \mathbf{G} 的一个二部子图, 其中的两个独立集分别记作为 A 和 B , 并且满足以下性质:

- 集合 A 是独立集 I 中仅与标记边相连接的 (非孤立) 顶点;

- 集合 B 是 A 的邻居, 即 $B = N_{\mathbf{G}}(A)$;
- 边集 $E(\mathbf{F})$ 是连接了 A 和 B 中顶点边的全体, 即 $E(\mathbf{F}) := \{ab \in E(\mathbf{G}) : a \in A, b \in B\}$ 。

注意到, \mathbf{F} 不包含孤立点, 并且 \mathbf{F} 中的边均为标记边。

令 (R, H, C) 为辅助图 \mathbf{F} 的 Dulmage-Mendelsohn 分解。定义两个点子集 (参见图7-1)

$$\begin{cases} \hat{A} := A \cap (R \cup C); \\ \hat{B} := B \cap (R \cup H). \end{cases}$$

根据定义7.6, 不难发现 \hat{B} 恰为 \hat{A} 的邻居, 即 $\hat{B} = N_{\mathbf{G}}(\hat{A})$; 并且一定存在 \hat{B} 到 \hat{A} 匹配, 即 \hat{B} 中的任何顶点均为匹配中某条边的端点。这意味着 \mathcal{I} 的每个解在集合 $\hat{A} \cup \hat{B}$ 中必定包含了至少 $|\hat{B}|$ 个顶点。此外, \hat{B} 是 $\hat{A} \cup \hat{B}$ 导出的子图的最小顶点覆盖。所以, 总存在 \mathcal{I} 的最优解包含了点子集 \hat{B} 。基于上述观察, 现引入 DM 约简规则的正式定义。

约简规则 7.3.1 (DM 约简规则) 设 \mathcal{I} 为 Generalized-SFVS(S) 问题的实例。令图 \mathbf{F} 是实例 \mathcal{I} 的辅助图, 其中的两个独立集分别记作为 A 和 B , 并用 (R, H, C) 表示 \mathbf{F} 的 Dulmage-Mendelsohn 分解。若集合 \hat{A} 和 \hat{B} 非空, 则从图 \mathbf{G} 中删去 \hat{A} 和 \hat{B} , 然后使 k 减少 $|\hat{B}| = |R|/2 + |H \cap B|$ 。

引理 7.7 DM 约简规则7.3.1是正确的。

证明: 依据 Dulmage-Mendelsohn 分解的定义, 有 $N_{\mathbf{F}}(\hat{A}) = \hat{B}$, 并且 \hat{B} 是子图 $\mathbf{F}[\hat{A} \cup \hat{B}]$ 的最小顶点覆盖。考虑实例 \mathcal{I} 的解集 S , 因为 S 需要与所有标记边相交, 所以集合 $S \cap (\hat{A} \cup \hat{B})$ 的大小不小于 $|\hat{B}|$ 。令集合 $S' = (S \setminus \hat{A}) \cup \hat{B}$ 。显然, S' 是图 \mathbf{G} 的 T -反馈点集, 这是因为删除 S' 后 A 是独立集。注意到 $|\hat{A}| > |\hat{B}|$, 否则 A 是辅助图 \mathbf{F} 的最小顶点覆盖, 但 H 不是 A 的子集, 矛盾。因此, 可以得到 $|S'| \leq |S|$, 进而说明 S' 也为 \mathcal{I} 的解, 引理得证。 ■

引理 7.8 设 $\mathcal{I} = (\mathbf{G}, T, M, k)$ 为 Generalized-SFVS(S) 问题的实例。令 \mathbf{F} 是实例 \mathcal{I} 的辅助图, 其中的两个独立集分别记作为 A 和 B 。若 DM 约简规则7.3.1不可执行, 那么对于任意点子集 $A' \subseteq A$, 总有 $|A'| < |N_{\mathbf{G}}(A')|$ 成立。

证明: 若 DM 约简规则在 \mathcal{I} 上不可执行, 那么辅助图 \mathbf{F} 的 Dulmage-Mendelsohn 分解必然为 $(R, H, C) = (\emptyset, A, B)$ 。

由 Dulmage-Mendelsohn 分解的定义, A 是辅助图 \mathbf{F} 的顶点覆盖, 并且 $H = A$ 是 \mathbf{F} 所有最小顶点覆盖的交集。由此可知, A 是 \mathbf{F} 唯一的最小顶点覆盖。

假设 DM 约简规则不可执行, 下证对于任意点子集 $A' \subseteq A$, 总有 $|A'| < |N_{\mathbf{G}}(A')|$ 。采用反证法, 假设存在 $A' \subseteq A$, 使得 $|A'| \geq |N_{\mathbf{G}}(A')|$ 。那么 $(A \setminus A') \cup$

$N_{\mathbf{G}}(A')$ 是 \mathbf{F} 的最小顶点覆盖, 这与 A 是唯一最小顶点覆盖矛盾, 引理得证。 ■

引理 7.9 设 $\mathcal{I} = (\mathbf{G}, T, M, k)$ 为 Generalized-SFVS(S) 问题的实例, \mathbf{F} 是实例 \mathcal{I} 的辅助图, 其中的两个独立集分别记作为 A 和 B 。若 DM 约简规则 7.3.1 不可执行并且 $k < |A|$, 那么实例 \mathcal{I} 无解。

证明: 由于 \mathbf{F} 是图 \mathbf{G} 的子图, 并且 $E(\mathbf{F}) \subseteq M$, 所以 \mathcal{I} 的解集大小必然不小于 \mathbf{F} 的最小顶点覆盖大小。若 DM 约简规则不可执行, 根据引理 7.8, A 是辅助图 \mathbf{F} 的唯一最小顶点覆盖。所以, \mathcal{I} 的解集大小必然至少为 $|A|$ 。因此, 当 $k < |A|$ 时, \mathcal{I} 无解, 引理得证。 ■

基于 DM 约简规则 7.3.1, 现引入约简实例的概念。

定义 7.10 (约简实例) 若 Generalized-SFVS(S) 问题的实例 $\mathcal{I} = (\mathbf{G}, T, M, k)$ 满足以下条件, 则称其为约简实例。

1. 图 \mathbf{G} 是 T -可分的, 即 $(T, V(\mathbf{G}) \setminus T)$ 是 \mathbf{G} 的分裂划分;
2. 每条标记边连接了一个关键点和一个非关键点;
3. DM 约简规则 7.3.1 在 \mathbf{F} 上不可执行, 其中 \mathbf{F} 是由分裂划分 $(T, V(\mathbf{G}) \setminus T)$ 决定的辅助图。

接下来证明在指数时间意义下, 求解约简实例并不会比求解任意 Generalized-SFVS(S) 问题实例更加容易。

引理 7.11 对于任意实数 $\alpha > 1$, Generalized-SFVS(S) 问题能在 $\mathcal{O}^*(\alpha^k)$ 时间内求解当且仅当约简实例能在 $\mathcal{O}^*(\alpha^k)$ 时间内求解。

证明: 由于约简实例是 Generalized-SFVS(S) 问题的一种特殊实例, 因此只需证明, 若约简实例能在 $\mathcal{O}^*(\alpha^k)$ 时间内求解, 那么 Generalized-SFVS(S) 问题便可在 $\mathcal{O}^*(\alpha^k)$ 时间内求解。

设 $\mathcal{I} = (\mathbf{G}, T, M, k)$ 是 Generalized-SFVS(S) 问题的任意实例。注意到 α 是常数, 因此存在充分大的常数 c , 使得分支向量 $(1, c, c)$ 的分支因子不超过 α 。现构造算法求解 \mathcal{I} 。

首先, 在多项式时间内计算图 \mathbf{G} 的一个分裂划分 (I, K) , 其中 I 为独立集, K 为团。若有 $|K| \leq 2c$, 则可以在常数时间内穷举 K 中的哪些点在解集中, 进而在线性时间内找到 \mathcal{I} 的最优解。因此, 不妨设 $|K| \geq 2c + 1$ 。接下来, 根据 K 中是否存在关键点分为两种情形讨论。

情形 1: K 中存在关键点 t 。在此情形中, 先将集合 $K \setminus \{t\}$ 划分为两部分 (K', K'') , 使得 $|K'| \geq C$ 以及 $|K''| \geq C$ 。若 t 不在解集中, 团 $K \setminus \{t\}$ 中至多还有一个顶点不在解中。所以, 要么 K' 在解中, 要么 K'' 在解中。所以可以分别计算以下三个子分支, 若三个子分支均无解, 则返回“无解”; 否则返回“有解”。

- 删除关键点 t , 使 k 减少 1;
- 删除集合 K' , 使 k 减少 $|K'|$;
- 删除集合 K'' , 使 k 减少 $|K''|$ 。

若以 k 度量, 这一分支规则的分支向量均为 $(1, |K'|, |K''|)$, 由 $|K'| \geq c$ 和 $|K''| \geq c$ 可知, 此分支向量对应的分支因子不超过 α 。

情形 2: K 中无关键点。在此情形中, 非关键点 $v \in I$ 不在任何 T -三角形中。注意到, 此时并不能将 v 移除, 因为 v 可能连接了某些标记边。但是, 我们可以执行如操作: 添加非关键点 v 和 $N_G(v) \cap K$ 中的所有顶点之间的边。不难发现, 此操作不增加新的 T -三角形或标记边, 并且得到的是以 $(I \setminus \{v\}, K \cup \{v\})$ 为分裂划分的分裂图。此外, 若 K 中存在两个非关键点 v 和 u , 它们之间的边是标记边, 则添加一个新的 2 度关键点 t_{uv} 连接 v 和 u , 然后去掉边 uv 的标记。

最终, 我们得到了一个分裂划分为 $(T, V(\mathbf{G}) \setminus T)$ 的分裂图, 并且所有标记边均连接一个关键点和一个非关键点。此时应用 DM 约简规则, 便得到了一个约简实例, 然后在 $\mathcal{O}^*(\alpha^k)$ 时间内求解此实例。所以, 整体求解实例 \mathcal{I} 的计算复杂度为 $\mathcal{O}^*(\alpha^k)$, 引理得证。 ■

基于引理 7.11, 在接下来的算法设计与分析中, 只需聚焦于约简实例即可。

7.3.2 一个非传统度量函数及其性质

借助于辅助子图及其上的 DM 约简规则 7.3.1, 可定义以下非传统的度量用于后文的算法时间复杂度分析。

定义 7.12 (约简实例的度量) 设 $\mathcal{I} = (\mathbf{G}, T, M, k)$ 为 Generalized-SFVS(S) 问题的实例。令 \mathbf{F} 是实例 \mathcal{I} 的辅助图, 其中的两个独立集分别记作为 A 和 B 。实例 \mathcal{I} 的度量定义为函数

$$\mu(\mathcal{I}) := k - \frac{2}{3}|A|.$$

基于定义 7.12, 便可以得到关于点集 A 的两条约简规则。

约简规则 7.3.2 (无解约简规则) 若 $|A| > k$ 成立, 甚至 $\mu(\mathcal{I}) < 0$ 成立, 则直接返回 “无解”。

约简规则 7.3.3 (有解约简规则) 若 $\tau \leq k$ 成立, 则直接返回 “有解”。

无解约简规则 7.3.2 的正确性是引理 7.9 的推论; 有解约简规则 7.3.3 的正确性则是根据一个简单的事实: 关键点集是一个 T -反馈点集。

本文提出参数算法的核心思想是对一个非关键点 $v \in V(\mathbf{G})$ 是否在解集中进行分支搜索: 若顶点 v 在解集中, 则将其删除, 并使 k 减少 1; 若顶点 v 不在解集中, 则将其隐藏 (hide)。这里, 隐藏操作的定义依赖于 v 是否为关键点:

隐藏关键点 t : 删除所有与关键点 t 通过标记边相邻的顶点, 即删除集合 $N_M(t)$, 并使 k 减少 $|N_M(t)|$;

隐藏非关键点 v : 删除所有与非关键点 v 通过标记边相邻的顶点, 即删除集合 $N_M(v)$, 然后使 k 减少 $|N_M(v)|$, 接着标记任何与 v 构成 T -三角形的边 ut (即 (u, t, v) 为 T -三角形), 最后从图中删除顶点 v 。

其中, 符号 $N_M(v) := \{u \in N(v) : uv \in M\}$ 表示所有与 v 通过标记边相连的顶点集。可见, 算法中包含两个基本操作: 删除顶点与隐藏顶点。

为使得度量 $\mu(\cdot)$ 能够良好地应用于分析时间复杂度, 必须保证算法的每个基本操作后, 得到的仍是约简实例, 并且其度量不增。因此, 首先需要在每个基本操作完成后, 均应用 DM 约简规则7.3.1, 即总是检查改规则是否可执行, 若可执行则执行 DM 约简规则。所以, 需要整体分析执行基本操作及后续 DM 约简规则7.3.1的度量整体变化情况。

引理 7.13 设 $\mathcal{I} = (\mathbf{G}, T, M, k)$ 为约简实例。若存在解集包含顶点 v , 则执行以下三个步骤是正确的: 首先删除 v , 然后使 k 减少 1, 最后应用 DM 约简规则7.3.1。若存在解集不包含顶点 v , 则执行以下两个步骤是正确的: 隐藏 v , 然后应用 DM 约简规则7.3.1。进一步地, 上述任意一种操作序列完成后得到的实例仍然是约简实例。

证明: 设 S 为约简实例 $\mathcal{I} = (\mathbf{G}, T, M, k)$ 的解。首先论证存在解集 S 包含顶点 v 的情况。在此情况中, $S \setminus \{v\}$ 是子实例 $(\mathbf{G} - v, T \setminus \{v\}, k - 1)$ 的解集。此外, 由引理7.7可知, DM 约简规则是正确的。所以, 操作序列“删除 v , 然后使 k 减少 1, 最后应用 DM 约简规则”是正确的。

接下来分析解集 S 不包含顶点 v 的情况。由于 S 覆盖了 M 中的边, 那么所有通过标记边相连与 v 相邻的顶点均在 S 中, 即 $N_M(v) \subseteq S$ 。这直接说明了 v 是关键点时, 隐藏 v 的正确性。当 v 是非关键点时, 对于任意 T -三角形 (u, v, t) , S 都至少包含 u 和 t 中至少一个顶点, 其中 u 是非关键点, t 是关键点。因此, “将边 ut 进行标记并加入边集 M 中”是正确的。此外, 由引理7.7可知, DM 约简规则是正确的。所以, 操作序列“隐藏关键点 v , 然后应用 DM 约简规则”是正确的。

最后, 由于两种操作序列都是由“删掉一个顶点”和“标记一条连接关键点和非关键点的边”这两个基本操作组合而成。不难发现, 执行完成每个基本操作后, 关键点集依然是独立集, 非关键点集依然是团, 而且每条标记边依然连接一个关键点和一个非关键点。此外, 两种操作序列都会在应用 DM 约简规则后结束, 所以, 最终得到的实例均是约简实例。

综上所述, 引理得证。 ■

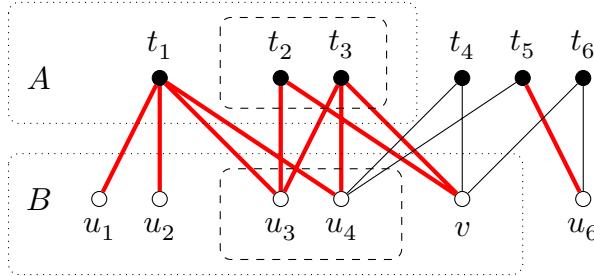


图 7-2 DM 约简规则在删除顶点后的执行情况示意图。在图 **G** 中, 黑色顶点表示关键点, 白色顶点表示非关键点, 红色粗线表示标记边, 此外省略了两个非关键点之间的边。在辅助图 **F** 中, 两部独立集分别为 $A = \{t_1, t_2, t_3\}$ 和 $B = \{u_1, u_2, u_3, u_4, v\}$ (图中短虚线框中的顶点)。再删除顶点 v 后, DM 约简规则可执行, 并且在执行完成后, 点集 $\hat{A} = \{t_2, t_3\}$ 以及 $\hat{B} = \{u_3, u_4\}$ 被删除 (图中长虚线框中的顶点)。

引理 7.14 设 $\mathcal{I} = (\mathbf{G}, T, M, k)$ 为约简实例。令图 **F** 是实例 \mathcal{I} 的辅助图, 其中的两个独立集分别记作为 A 和 B 。对于顶点 $v \in V(\mathbf{G}) \setminus A$, 实例 \mathcal{I}_1 表示删除 v 后应用 DM 约简规则7.3.1得到的实例 (参见图7-2)。那么, \mathcal{I}_1 是约简实例, 并且满足 $\mu(\mathcal{I}) - \mu(\mathcal{I}_1) \geq 0$ 。

证明: 设 $\mathcal{I}_0 = (\mathbf{G}_0, T_0, M_0, k_0)$ 是通过从实例 \mathcal{I} 中删除顶点 v 后得到的实例。设 $\mathcal{I}_1 = (\mathbf{G}_1, T_1, M_1, k_1)$ 是通过在实例 \mathcal{I}_0 上应用 DM 约简规则后得到的实例。同时, 令 \mathbf{F}_i 表示实例 \mathcal{I}_i 的辅助图, 且它的两部分独立集为 A_i 和 B_i ($i \in \{0, 1\}$)。

由于没有新标记的边且 $k_0 = k$, 容易得出 $\mu(\mathcal{I}_0) = \mu(\mathcal{I})$ 成立。假设最终从 \mathcal{I}_0 中移除了 $\hat{A}_0 \subseteq A_0$ 和 $\hat{B}_0 \subseteq B_0$ 。那么, DM 约简规则不执行当且仅当有 $\hat{A}_0 = \hat{B}_0 = \emptyset$ 。注意到关系 $A_1 = A_0 \setminus \hat{A}_0$, $B_1 = B_0 \setminus \hat{B}_0$ 以及 $k_1 = k_0 - |\hat{B}_0|$ 。所以, 可以得出度量函数的变化量为

$$\mu(\mathcal{I}) - \mu(\mathcal{I}_1) = \mu(\mathcal{I}_0) - \mu(\mathcal{I}_1) = \left(k_0 - \frac{2}{3}|A_0|\right) - \left(k_1 - \frac{2}{3}|A_1|\right) = |\hat{B}_0| - \frac{2}{3}|\hat{A}_0|.$$

当 DM 约简规则不可执行时, 自然有 $\hat{A}_0 = \hat{B}_0 = \emptyset$, 这说明着度量是不增的, 即 $\mu(\mathcal{I}) - \mu(\mathcal{I}_1) = 0$ 。否则, 删除顶点 v 后, DM 约简规则是可执行的。此时, 必然有 $v \in B$ 。此外, 根据引理7.8, \mathcal{I} 是约简实例意味着

$$N_{\mathbf{G}}(\hat{A}_0) = \hat{B}_0 \cup \{v\},$$

以及

$$|\hat{B}_0 \cup \{v\}| > |\hat{A}_0|.$$

所以可以推出 $|\hat{A}_0| = |\hat{B}_0|$, 进而有 $\mu(\mathcal{I}) - \mu(\mathcal{I}_1) \geq 0$ 。引理得证。 \blacksquare

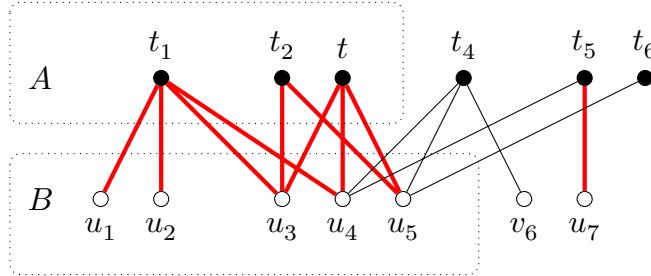


图 7-3 DM 约简规则在隐藏关键点后的执行情况示意图。在图 **G** 中, 黑色顶点表示关键点, 白色顶点表示非关键点, 红色粗线表示标记边, 此外省略了两个非关键点之间的边。在辅助图 **F** 中, 两部独立集分别为 $A = \{t_1, t_2, t_3\}$ 和 $B = \{u_1, u_2, u_3, u_4, u_5\}$ (图中短虚线框中的顶点)。在隐藏关键点 t 后, 关键点 t_2 变为孤立点, 但是 DM 约简规则不可执行。

引理 7.15 设 $\mathcal{I} = (\mathbf{G}, T, M, k)$ 为约简实例。令图 **F** 是实例 \mathcal{I} 的辅助图, 其中的两个独立集分别记作为 A 和 B 。对于关键点 $t \in T$, 实例 \mathcal{I}_1 表示隐藏 t 后应用 DM 约简规则7.3.1得到的实例 (参见图7-3)。那么, \mathcal{I}_1 是约简实例, 并且满足

- 若 $t \in A$, 则 $\mu(\mathcal{I}) - \mu(\mathcal{I}_1) \geq 4/3$;
- 若 $t \notin A$, 则 $\mu(\mathcal{I}) - \mu(\mathcal{I}_1) \geq 0$ 。

证明: 设 $\mathcal{I}_0 = (\mathbf{G}_0, T_0, M_0, k_0)$ 是通过从实例 \mathcal{I} 中隐藏关键点 t 后得到的实例。设 $\mathcal{I}_1 = (\mathbf{G}_1, T_1, M_1, k_1)$ 是通过在实例 \mathcal{I}_0 上应用 DM 约简规则后得到的实例。同时, 令 \mathbf{F}_i 表示实例 \mathcal{I}_i 的辅助图, 且它的两部分独立集为 A_i 和 B_i ($i \in \{0, 1\}$)。

在隐藏了关键点 t 的操作中, 一个顶点被删除当且仅当它是非关键点且与 t 之间有一条标记边。所以, 有关系 $B_0 = B \setminus N_M(t)$ 以及 $k_0 = k - |N_M(t)|$ 。进而得出

$$\mu(\mathcal{I}) - \mu(\mathcal{I}_0) = \left(k - \frac{2}{3}|A|\right) - \left(k_0 - \frac{2}{3}|A_0|\right) = |N_M(t)| - \frac{2}{3}(|A| - |A_0|).$$

注意到, 在隐藏 t 的操作过程中, 仅有 t 自身被删除了。此外, 关键点 $t' \neq t$ 从 A 中移除当且仅当 t' 变成了孤立点。所以, 若 $t' \in A \setminus A_0$, 必然有 $N_{\mathbf{G}}(t') \subseteq N_M(t)$ 。

假设最终从 \mathcal{I}_0 中移除了 $\hat{A}_0 \subseteq A_0$ 和 $\hat{B}_0 \subseteq B_0$ 。那么, DM 约简规则不执行当且仅当有 $\hat{A}_0 = \hat{B}_0 = \emptyset$ 。注意到关系 $k_1 = k_0 - |\hat{B}_0|$ 以及 $|A_1| = |A_0| - |\hat{A}_0|$ 。所以, 可以得出度量函数的变化量为

$$\begin{aligned} \mu(\mathcal{I}) - \mu(\mathcal{I}_1) &= |N_M(t)| - \frac{2}{3}(|A| - |A_0|) + |\hat{B}_0| - \frac{2}{3}|\hat{A}_0| \\ &= |N_M(t) \cup \hat{B}_0| - \frac{2}{3}(|A| - |A_1|) \\ &\geq |N_M(t) \cup \hat{B}_0| - \frac{2}{3}|A \setminus A_1|. \end{aligned}$$

考虑关键点 $t' \in A \setminus A_1$ 。若有 $t' \in A \setminus A_0$, 那么 $N_{\mathbf{G}}(t') \subseteq N_M(t)$ 。否则, 必然有

$t' \in \hat{A}_0$ 并且 t' 在图 \mathbf{G} 中的所有邻居都被删除了, 这说明了 $N_{\mathbf{G}}(t') \subseteq \hat{B}_0 \cup N_M(t)$ 。所以, 可以得出 $N_{\mathbf{G}}(A \setminus A_1) \subseteq \hat{B}_0 \cup N_M(t)$ 。又因为实例 \mathcal{I} 是约简实例, 根据引理7.8, 要么 $A \setminus A_1 = \emptyset$ 成立, 要么 $|N_{\mathbf{G}}(A \setminus A_1)| > |A \setminus A_1|$ 成立。

若 $A \setminus A_1 = \emptyset$ 成立, 那么每个 A 中的关键点均未被删除, 这意味着 $t \notin A$ 。此时有

$$\mu(\mathcal{I}) - \mu(\mathcal{I}_1) \geq |N_M(t) \cup \hat{B}_0| - \frac{2}{3}|A \setminus A_1| \geq |N_M(t) \cup \hat{B}_0| \geq 0.$$

若 $|N_{\mathbf{G}}(A \setminus A_1)| > |A \setminus A_1|$ 成立, 有

$$\mu(\mathcal{I}) - \mu(\mathcal{I}_1) \geq |N_M(t) \cup \hat{B}_0| - \frac{2}{3}|A \setminus A_1| \geq |N_{\mathbf{G}}(A \setminus A_1)| - \frac{2}{3}|A \setminus A_1| \geq \frac{4}{3}.$$

综上所述, 引理得证。 \blacksquare

引理 7.16 设 $\mathcal{I} = (\mathbf{G}, T, M, k)$ 为约简实例。令图 \mathbf{F} 是实例 \mathcal{I} 的辅助图, 其中的两个独立集分别记作为 A 和 B 。对于非关键点 $t \in V(\mathbf{G}) \setminus T$, 实例 \mathcal{I}_1 表示隐藏 v 后应用 DM 约简规则7.3.1得到的实例 (参见图7-4)。那么, \mathcal{I}_1 是约简实例, 并且满足 $\mu(\mathcal{I}) - \mu(\mathcal{I}_1) \geq 0$ 。

进一步地, 若每个关键点 (或非关键点) 连接了至少两个非关键点 (或关键点), 并且所有 2 度关键点在 \mathbf{G} 中的邻居都不完全相同, 那么有

- 若 $v \in B$, 则

$$\mu(\mathcal{I}) - \mu(\mathcal{I}_1) \geq \min \left\{ \frac{2}{3}|N_{\mathbf{G}}(v) \cap T| - \frac{1}{3}|N_M(v) \cap A|, \frac{4}{3} \right\}.$$

- 若 $v \notin B$, 则

$$\mu(\mathcal{I}) - \mu(\mathcal{I}_1) \geq \min \left\{ \frac{2}{3}|N_{\mathbf{G}}(v) \cap T| + \frac{1}{3}|N_M(v) \cap A|, \frac{4}{3} \right\} = \frac{4}{3}.$$

证明: 设 $\mathcal{I}_0 = (\mathbf{G}_0, T_0, M_0, k_0)$ 是通过从实例 \mathcal{I} 中隐藏非关键点 v 后得到的实例。设 $\mathcal{I}_1 = (\mathbf{G}_1, T_1, M_1, k_1)$ 是通过在实例 \mathcal{I}_0 上应用 DM 约简规则后得到的实例。同时, 令 \mathbf{F}_i 表示实例 \mathcal{I}_i 的辅助图, 且它的两部分独立集为 A_i 和 B_i ($i \in \{0, 1\}$)。

在隐藏了关键点 v 的操作中, 一个顶点被删除当且仅当它是一个关键点且和 v 之间有一条标记边。所以, 有关系 $k_0 = k - |N_M(v)|$ 以及 $A \setminus A_0 = A \cap N_M(v)$ 。所以, 可以得出

$$\begin{aligned} \mu(\mathcal{I}) - \mu(\mathcal{I}_0) &= (k - \frac{2}{3}|A|) - (k_0 - \frac{2}{3}|A_0|) \\ &= (k - k_0) + \frac{2}{3}|A_0 \setminus A| - \frac{2}{3}|A \setminus A_0| \\ &= |N_M(v)| + \frac{2}{3}|A_0 \setminus A| - \frac{2}{3}|A \cap N_M(v)|. \end{aligned}$$

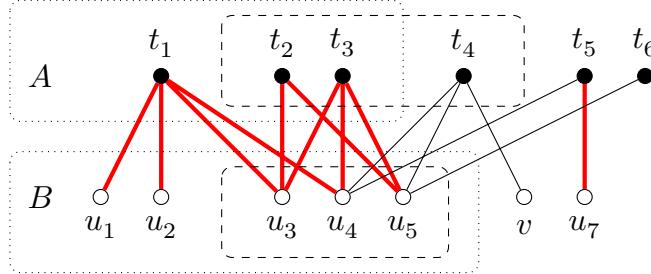


图 7-4 DM 约简规则在隐藏非关键点后的执行情况示意图。在图 **G** 中, 黑色顶点表示关键点, 白色顶点表示非关键点, 红色粗线表示标记边, 此外省略了两个非关键点之间的边。在辅助图 **F** 中, 两部独立集分别为 $A = \{t_1, t_2, t_3\}$ 和 $B = \{u_1, u_2, u_3, u_4, u_5\}$ (图中短虚线框中的顶点)。在隐藏非关键点 v 后, DM 约简规则不可执行, 并且在执行完成后, 点集 $\hat{A} = \{t_2, t_3, t_4\}$ 以及 $\hat{B} = \{u_3, u_4, u_5\}$ 被删除 (图中长虚线框中的顶点)。

因为有 $|A \cap N_M(v)| \leq |N_M(v)|$, 进而有 $\mu(\mathcal{I}) - \mu(\mathcal{I}_0) \geq 0$ 。这也说明, 当 DM 约简规则在实例 \mathcal{I}_0 上不可执行时, 自然有 $\mu(\mathcal{I}_0) = \mu(\mathcal{I}_1)$, 最终得出度量函数不增, 即 $\mu(\mathcal{I}) - \mu(\mathcal{I}_1) \geq 0$ 。

接下来, 考察 $A_0 \setminus A$ 中的关键点 t 。一方面, 关键点 t 必然是通过一条未标记边与 v 相连。另一方面, 在隐藏了 v 之后, 关键点 t 不是孤立点, 这说明了关键点 t 连接了与 v 不同的另一个顶点。所以, 若每个关键点都连接了至少两个非关键点, 则有 $A_0 \setminus A = (N_{\mathbf{G}}(v) \cap T) \setminus N_M(v)$ 。此时, 可以得出度量函数的变化量为

$$\begin{aligned}\mu(\mathcal{I}) - \mu(\mathcal{I}_0) &= |N_M(v)| + \frac{2}{3}|A_0 \setminus A| - \frac{2}{3}|A \cap N_M(v)| \\ &= |N_M(v)| + \frac{2}{3}|(N_{\mathbf{G}}(v) \cap T) \setminus N_M(v)| - \frac{2}{3}|A \cap N_M(v)| \\ &= |N_M(v)| + \frac{2}{3}(|N_{\mathbf{G}}(v) \cap T| - |N_M(v)|) - \frac{2}{3}|A \cap N_M(v)| \\ &\geq \frac{2}{3}|N_{\mathbf{G}}(v) \cap T| - \frac{1}{3}|A \cap N_M(v)|.\end{aligned}$$

若约简规则在 \mathcal{I}_0 上不可执行, 有

$$\mu(\mathcal{I}) - \mu(\mathcal{I}_1) = \mu(\mathcal{I}) - \mu(\mathcal{I}_0) \geq \frac{2}{3}|N_{\mathbf{G}}(v) \cap T| - \frac{1}{3}|A \cap N_M(v)|.$$

此外, 若 $v \notin B$, 则说明 v 不与 A 中的任何顶点相连, 从而 $A \cap N_M(v)$ 为空。此时, 若 v 属于 B 而且连接了至少两个关键点, 能够进一步得到度量函数的变化量为

$$\mu(\mathcal{I}) - \mu(\mathcal{I}_1) = \mu(\mathcal{I}) - \mu(\mathcal{I}_0) \geq \frac{2}{3}|N_{\mathbf{G}}(v) \cap T| \geq \frac{4}{3}.$$

到此为止, DM 约简规则在实例 \mathcal{I}_0 上不可执行时的度量函数变化情况已完全得出。

现分析 DM 约简规则在实例 \mathcal{I}_0 上可执行时的度量函数变化情况。假设最终从 \mathcal{I}_0 中移除了 $\hat{A}_0 \subseteq A_0$ 和 $\hat{B}_0 \subseteq B_0$ 。自然 \hat{A}_0 和 \hat{B}_0 都是非空的，并且 $k_1 = k_0 - |\hat{B}_0| = k - |N_M(v)| - |\hat{B}_0|$ 成立。考察 $A \setminus A_1$ 中的关键点 t' 。若 $t' \in A \cap N_M(v)$ ，那么在隐藏 v 时 t' 已被删除；否则将在执行 DM 约简规则时被删除，此时必有 $t' \in A \cap \hat{A}_0$ 。所以，可以得出度量函数的变化量为

$$\begin{aligned}\mu(\mathcal{I}) - \mu(\mathcal{I}_1) &= (k - \frac{2}{3}|A|) - (k_1 - \frac{2}{3}|A_1|) \\ &= |N_M(v)| + |\hat{B}_0| + \frac{2}{3}|A_1 \setminus A| - \frac{2}{3}|A \setminus A_1| \\ &\geq |N_M(v)| + |\hat{B}_0| + \frac{2}{3}|A_1 \setminus A| - \frac{2}{3}(|A \cap \hat{A}_0| + |A \cap N_M(v)|) \\ &\geq \frac{1}{3}|N_M(v)| + |\hat{B}_0| + \frac{2}{3}|A_1 \setminus A| - \frac{2}{3}|A \cap \hat{A}_0|.\end{aligned}$$

现分析 $\mu(\mathcal{I}) - \mu(\mathcal{I}_1)$ 的下界。根据 $A \cap \hat{A}_0$ 是否为空集，考虑两种情形。

情形 1.1: $A \cap \hat{A}_0$ 非空。观察到，在图 **G** 中， v 不会与 \hat{A}_0 中的任何关键点相连。原因在于：任何与 v 通过标记边相连的关键点均会在隐藏 v 时删除掉，因而不会出现在图 **G** 中。所以 $\hat{B}_0 = N_{\mathbf{G}}(\hat{A}_0)$ 成立。此外，由于 $B = N_{\mathbf{G}}(A)$ ，所以有 $B \cap \hat{B}_0 = N_{\mathbf{G}}(A \cap \hat{A}_0)$ 。又因为 \mathcal{I} 是约简实例，并且 $A \cap \hat{A}_0$ 非空，那么有 $|B \cap \hat{B}_0| > |A \cap \hat{A}_0|$ 。所以，可以得出度量函数的变化量为

$$\mu(\mathcal{I}) - \mu(\mathcal{I}_1) \geq |\hat{B}_0| - \frac{2}{3}|A \cap \hat{A}_0| \geq 1 + \frac{1}{3}|A \cap \hat{A}_0| \geq \frac{4}{3}.$$

情形 1.2: $A \cap \hat{A}_0$ 为空。在此情形中，可以直接得到

$$\mu(\mathcal{I}) - \mu(\mathcal{I}_1) \geq \frac{1}{3}|N_M(v)| + |\hat{B}_0| + \frac{2}{3}|A_1 \setminus A| \geq \frac{1}{3}|N_M(v)| + 1 \geq 1.$$

这说明了度量是不增的。

最后证明，当每个关键点（或非关键点）连接了至少两个非关键点（关键点），并且所有 2 度关键点在 **G** 中的邻居都不完全相同时， $\mu(\mathcal{I}) - \mu(\mathcal{I}_1) \geq 4/3$ 总成立。根据顶点 v 是否属于 B ，考虑两种情形。

情形 2.1: $v \in B$ 。在此情形下， v 至少连接了 A 中的一个顶点，并且是通过标记边相连。因此，集合 $N_M(v)$ 非空，同时有 $\mu(\mathcal{I}) - \mu(\mathcal{I}_1) \geq 1/3 + 1 = 4/3$ 。所以，当 $v \in B$ 时，度量函数至少减少了 $4/3$ 。

情形 2.2: $v \notin B$ 。采用反证法，假设 DM 约简规则可执行并且度量函数满足 $\mu(\mathcal{I}) - \mu(\mathcal{I}_1) < 4/3$ 。此时有 $|N_M(v)| = 0$ ， $|\hat{B}_0| = 1$ 并且 $|A_1 \setminus A| = 0$ 。这意味着 t 与 v 相连。首先根据引理条件，有 $\deg_G(t) \geq 2$ 。因为 $N_M(v)$ 是空集，则 tv 是未标记边，同时 $t \notin A$ 。又因为 $\deg_G(t) \geq 2$ ，并且在隐藏 v 的操作中 v 是唯一被删除

的非关键点, 所以 t 属于 A_0 。注意到 $A_1 \subseteq A$ 但 $t \notin A$, 所以在执行 DM 约简规则时, t 被删除了, 即 $t \in \hat{A}_0$ 。由于 \hat{B}_0 是单元集, 所以 t 在图 \mathbf{G} 中恰好有两个邻居, 其中一个顶点是 v , 另一个顶点是 \hat{B}_0 集合中的唯一顶点, 即 $N_{\mathbf{G}}(t) = \{v\} \cup \hat{B}_0$ 。综合可知, $N_{\mathbf{G}}(v)$ 中的所有关键点都是 2 度顶点并且他们的邻居完全相同, 与引理条件矛盾所以, 对于任意非关键点 $v \notin B$, 都有

$$\mu(\mathcal{I}) - \mu(\mathcal{I}_1) \geq \min\left\{\frac{2}{3}|N_{\mathbf{G}}(v) \cap T|, \frac{4}{3}\right\} = \frac{4}{3}.$$

综上所述, 引理得证。 ■

7.3.3 基于分支搜索技术的参数算法

7.3.2节完成了对度量 $\mu(\cdot)$ 的全部分析工作, 本节将给出算法并分析时间复杂度。接下来将引入三条简单的约简规则和两条分支规则, 任意一条规则只有在前述所有规则均不可执行时才会开始执行。

约简规则 7.3.4 (无效顶点约简规则) 若顶点 v 既不属于任何 T -三角形中, 也不连接任何标记边, 则直接删除顶点 v , 然后应用 DM 约简规则7.3.1。

约简规则 7.3.5 (非关键点约简规则) 若非关键点 v 只与一个关键点相连, 则隐藏顶点 v , 然后应用 DM 约简规则7.3.1。

无效顶点约简规则7.3.4和非关键点约简规则7.3.5的正确性是平凡的。若顶点 v 不在任何 T -三角形或标记边中, 那么任何最优解均不包含 v 。结合引理7.13, 无效顶点约简规则7.3.4是正确的。若非关键点 v 只与唯一一个关键点 t 相连, 那么任何包含顶点 v 的 T -三角形或标记边均包含了关键点 t , 因此, 对于任何一个包含顶点 v 的解集, 将其中的 v 替换为 t 也会得到一个解。结合引理7.13, 非关键点约简规则7.3.5是正确。

基于引理7.14和引理7.16可知, 无效顶点约简规则7.3.4和非关键点约简规则7.3.5执行完成后, 得到的依然是约简实例且度量不增。此外, 容易发现, 若上述两条约简规则不再能执行, 那么

约简规则 7.3.6 (二度关键点约简规则) 设 $t, t' \in T \setminus A$ 为不在 A 中的任意两个 2 度关键点, 即 $\deg_{\mathbf{G}}(t) = \deg_{\mathbf{G}}(t') = 2$ 。

1. 若 t 恰好连接了一条标记边, 则隐藏 t , 然后应用 DM 约简规则7.3.1。
2. 若关键点 t 和 t' 的邻居完全相同 (即 $N_{\mathbf{G}}(t) = N_{\mathbf{G}}(t')$) 且它们均不与标记边相连接, 则删除 t , 然后应用 DM 约简规则7.3.1。

现论证二度关键点约简规则7.3.6的正确性。

引理 7.17 二度关键点约简规则7.3.6是正确的。在此规则执行完成后, 得到的实例是约简实例, 并且度量 $\mu(\cdot)$ 不增。

证明: 依据引理7.13, 规则执行完成后得到的实例是约简实例。又根据引理7.14, 引理7.15和引理7.16, 度量函数 $\mu(\cdot)$ 不增。因此, 只需证明二度关键点约简规则的正确性。

设输入实例为 $\mathcal{I} = (\mathbf{G}, T, k)$ 。首先考虑二度关键点约简规则中的第一种情况。设关键点 t 恰好在 T -三角形 (t, v, u) 中, 并且标记边为 $tv \in M$, 则 tu 不是标记边。若存在包含关键点 t 的解集 S , 那么 $S' = (S \setminus \{t\}) \cup \{v\}$ 也是一个包含 t 的解集。原因在于 T -三角形 (t, v, u) 和标记边 tv 都包含了顶点 v , 而 v 在 S' 中。所以, 总存在不包含 t 的解集, 再结合引理7.13, 二度关键点约简规则中的第一种情况是正确的。

接下来考虑二度关键点约简规则中的第二种情况。一方面, \mathcal{I} 的解必然是子实例 $(\mathbf{G} - t, T \setminus \{t\}, k)$ 的解。另一方面, 设 S 为子实例 $(\mathbf{G} - t, T \setminus \{t\}, k)$ 的解。设 t 只存在于一个 T -三角形中, 记作 (t, v, u) 。那么 (t', v, u) 是一个在子图 $\mathbf{G} - t$ 中的 T -三角形。由于 t' 是 2 度点, 不妨设 S 不包含 t' , 所以 v 和 u 中至少其中之一在 S 中。所以 S 也是 \mathcal{I} 的解, 这说明了二度关键点约简规则中的第二种情况也是正确的。

综上所述, 引理得证。 ■

至此为止, 所有约简规则已全部引入。不难看出, 若所有约简规则不再能执行, 引理7.16中的条件必然满足, 即每个关键点(或非关键点)连接了至少两个非关键点(或关键点); 并且所有 2 度关键点的邻居互不相同。现介绍两条分支规则, 它们用于处理与未标记边相连的非关键点。第一条分支规则处理集合 B 中的一种特殊 2 度非关键点, 称为 B -非关键点分支规则; 第二条分支规则处理任意一个连接了未标记边的非关键点, 称为非关键点分支规则。

分支规则 7.3.7 (B -非关键点分支规则) 若非关键点 $v \in B$ 恰好与两个关键点 t 和 t' 相邻, 并且 vt 是标记边而 vt' 不是标记边, 则分别计算以下两个子分支。若两个子分支均无解, 则返回“无解”; 否则返回“有解”。

- 隐藏顶点 v , 然后应用 DM 约简规则7.3.1。
- 隐藏顶点 t , 然后应用 DM 约简规则7.3.1。

分支规则 7.3.8 (非关键点分支规则) 若非关键点 $v \in V(\mathbf{G})$ T 连接了一个非标记边, 则分别计算以下两个子分支。若两个子分支均无解, 则返回“无解”; 否则返回“有解”。

- 删除顶点 v , 使 k 减少 1, 然后应用 DM 约简规则7.3.1。
- 隐藏顶点 v , 然后应用 DM 约简规则7.3.1 (参见图7-5)。

现论证 B -非关键点分支规则7.3.7和非关键点分支规则7.3.8的正确性。

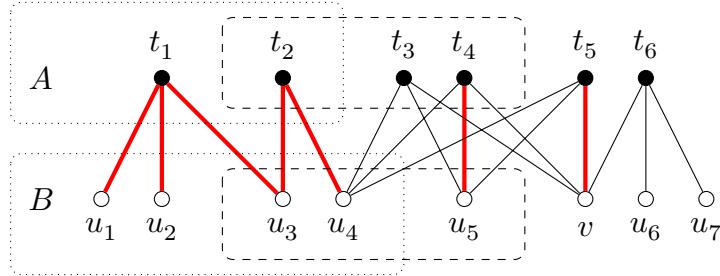


图 7-5 非关键点分支规则示意图。在图 \mathbf{G} 中, 黑色顶点表示关键点, 白色顶点表示非关键点, 红色粗线表示标记边, 此外省略了两个非关键点之间的边。在辅助图 \mathbf{F} 中, 两部独立集分别为 $A = \{t_1, t_2, t_3, t_4\}$ 和 $B = \{u_1, u_2, u_3, u_4, v_5\}$ (图中短虚线框中的顶点)。对于非关键点 $v \in V(\mathbf{G}) \setminus T$, 若将其删除, DM 约简规则不可执行; 若将其隐藏, DM 约简规则可执行, 并且在执行完成后, 点集 $\hat{A} = \{t_2, t_3, t_4\}$ 以及 $\hat{B} = \{v_3, v_4, v_5\}$ 被删除 (图中长虚线框中的顶点)。

引理 7.18 B -非关键点分支规则7.3.7和非关键点分支规则7.3.8是正确的, 且执行完成后得到的任意实例均是约简实例。此外, 这两条分支规则的分支向量均不劣于 $(1, 4/3)$ 。

证明: 依据引理7.13, B -非关键点分支规则和非关键点分支规则得到的任何子实例均为约简实例。

接下来分析 B -非关键点分支规则和非关键点分支规则的正确性。显然, 一个非关键点要么在解中, 要么不在解中, 所以非关键点分支规则是正确的。而 B -非关键点分支规则的正确性基于以下事实: 总存在解恰好包含 v 和 t 其中之一。原因在于, 一方面 vt 是标记边, 所以 v 和 t 至少其中之一在解中; 另一方面, 若解 S 包含了 t , 那么在删去顶点 t 后, v 恰好连接了一个关键点, 此时总存在解不包含 v 。

最后, 我们分析 B -非关键点分支规则和非关键点分支规则的分支向量。在 B -非关键点分支规则中, 顶点 v 是属于 B 的非关键点, 且满足 $|N_M(v)| = 1$ 以及 $|N_{\mathbf{G}}(v) \cap T| = 2$ 。此外, 引理7.16中的条件满足, 因为所有约简规则均无法执行。所以, 在 B -非关键点分支规则的第一个分支中, 根据引理7.16可知度量函数至少减少了 1。在 B -非关键点分支规则的第二个分支中, 顶点 $v \in B$ 意味着 $t \in A$, 再根据引理7.16, 可知度量函数至少减少了 $4/3$ 。最终得到 B -非关键点分支规则分支向量为 $(1, 4/3)$ 。

在非关键点分支规则中的第一个分支中, 由引理7.14可知, 删除 v 再应用 DM 约简规则不会使得度量函数增加。又因为顶点 v 放入解中导致了 k 减少 1, 因此度量函数整体至少减少了 1。在非关键点分支规则中的第二个分支中, 若 v 属于 B ,

有 $|N_{\mathbf{G}}(v) \cap T| \geq 3$, 这是因为 B -非关键点分支规则此时不可执行。根据顶点 v 的选择, 集合 $(N_{\mathbf{G}}(v) \cap T) \setminus A$ 非空。此外, 因为所有约简规则均无法执行, 引理7.16中的条件满足。所以, 由引理7.16可知度量函数至少减少了 $2/3 \times 3 - 1/3 \times 2 = 4/3$ 。最终得到非关键点分支规则分支向量为 $(1, 4/3)$ 。

综上所述, B -非关键点分支规则和非关键点分支规则的分支向量均不劣于 $(1, 4/3)$, 引理得证。 ■

当所有分支规则均不可执行时, 连接关键点和非关键点的边均是未标记边, 这意味着无解约简规则7.3.2或有解约简规则7.3.3能够直接返回结果。至此, 得到了以下求解 Generalized-SFVS(S) 问题的参数算法。

算法 7-1 求解 Generalized-SFVS(S) 问题的基于分支搜索技术的参数算法

输入: Generalized-SFVS(S) 问题的实例 $\mathcal{I} = (\mathbf{G}, T, M, k)$ 。

输出: 实例 \mathcal{I} 有解或无解。

- 1 **if** \mathcal{I} 不是约简实例 **then**
- 2 根据引理7.11, 在多项式时间内归约到约简实例并更新 \mathcal{I} 。// 此步骤至多执行一次, 并且在此步骤后的实例 \mathcal{I} 总是一个约简实例。
- 3 **end**
- 4 执行无解约简规则7.3.2和有解约简规则7.3.3。// 此步骤为算法的终止条件。
- 5 执行无效顶点约简规则7.3.4。// 此步骤后, 所有顶点均在某个 T -三角形或标记边中, 这意味着关键点的度至少为 2。
- 6 执行非关键点约简规则7.3.5。// 此步骤后, 任何非关键点至少与两个关键点相邻。
- 7 执行二度关键点约简规则7.3.6。// 此步骤后, 所有 2 度关键点的邻居互不相同, 这意味着引理7.16中的条件成立。
- 8 执行 B -非关键点分支规则7.3.7。// 此步骤后, 集合 B 中任何与未标记边相连的非关键点至少与三个关键点相邻。
- 9 执行非关键点分支规则7.3.8。// 此步骤后, 关键点和非关键点间的边均为标记边, 因此无解约简规则7.3.2或有解约简规则7.3.3可以执行。

引理 7.19 算法7-1是正确的, 且能够在 $\mathcal{O}^*(1.8192^k)$ 时间内求解 Generalized-SFVS(S) 问题的约简实例。

证明: 依据引理7.17和引理7.18, 算法7-1中包含的各条规则均是正确的。此外, 根据引理7.18, 算法7-1中的两条分支规则的分支向量为 $(1, 4/3)$, 其分支因子为 1.8192。所以, Generalized-SFVS(S) 问题的约简实例均能在

$$\mathcal{O}^*(1.8192^\mu) = \mathcal{O}^*(1.8192^{k-2|A|/3}) \leq \mathcal{O}^*(1.8192^k)$$

时间内求解, 引理得证。 ■

定理 7.4 SFVS(S) 问题可以在 $\mathcal{O}^*(1.8192^k)$ 时间内求解。

证明: 结合引理7.11, 可以得出 Generalized-SFVS(S) 问题能在 $\mathcal{O}^*(1.8192^k)$ 时间内求解。那么, SFVS(S) 问题也可以在 $\mathcal{O}^*(1.8192^k)$ 时间内求解。定理得证。 ■

推论 7.20 SFVS(S) 问题可以在 $\mathcal{O}^*(1.8192^\tau)$ 时间内求解。

证明: 当 $\tau \leq k$ 时, 实例必然有解, 可以在线性时间内判断。当 $\tau > k$ 时, 根据定理7.4, 可以在 $\mathcal{O}^*(1.8192^k) \leq \mathcal{O}^*(1.8192^\tau)$ 时间内求解。推论成立。 ■

7.3.4 基于均摊思想的精确算法

本节将给出 SFVS(S) 问题的精确算法。根据引理7.2, 只需考虑求解 SFVS(S) 问题的 T -可分实例。

引理 7.21 SFVS(S) 问题能在 $\alpha^n n^{\mathcal{O}(1)}$ 时间内求解, 当且仅当 SFVS(S) 问题的 T -可分实例能在 $\alpha^n n^{\mathcal{O}(1)}$ 时间内求解。

证明: 由于 T -可分实例是 SFVS(S) 问题的特殊实例, 因此只需证明, 若约简实例能 $\alpha^n n^{\mathcal{O}(1)}$ 时间内求解, 那么 SFVS(S) 问题便可在 $\alpha^n n^{\mathcal{O}(1)}$ 时间内求解。

设 $\mathcal{I} = (\mathbf{G}, T, k)$ 是 SFVS(S) 问题的实例。首先在多项式时间内计算 \mathbf{G} 的一个分裂划分 (I, K) , 其中 I 为独立集, K 为团。记 $I' = I \setminus T$ 且 $K' = K \cap T$ 。

根据引理7.2, 可以在多项式时间内判断是否存在一个最小 T -反馈点集, 它包含了某些 I 中的某个非关键点, 或者不包含 K 中某个关键点。若存在, 则直接返回 “有解”; 否则可以将集合 I' 和集合 K' 直接删去, 然后使 k 减少 $|K'|$ 。那么最后得到一个 T -可分实例 $\mathcal{I}' = (\mathbf{G} - I' - K', T \setminus K', k - |K'|)$ 。最后, 调用 T -可分实例的算法, 在 $\alpha^n n^{\mathcal{O}(1)}$ 时间内求解。总时间复杂度为

$$\alpha^n n^{\mathcal{O}(1)} + n^{\mathcal{O}(1)} = \alpha^n n^{\mathcal{O}(1)}.$$

引理得证。 ■

现采用一个简单的均摊思想, 当 k 较小时, 参数算法的运行时间较短, 而当 k 较大时, 也存在能够快速判定是否有解的方法。

引理 7.22 设 $\mathcal{I} = (\mathbf{G}, T, \emptyset, k)$ 为 T -可分实例。当 $2k \geq n$ 时, \mathcal{I} 必有解。

证明: $\mathcal{I} = (\mathbf{G}, T, \emptyset, k)$ 为 T -可分实例, 所以 $(I, K) = (T, V(\mathbf{G}) \setminus T)$ 是图 \mathbf{G} 的分裂划分。一方面, 根据分裂划分的结构, K 是 T -反馈点集。另一方面, I 是 T -反馈点集, 因为 K 中所有顶点均为非关键点。因此, 可以得出 $|I| \leq k$ 或 $|K| \leq k$ 时, \mathcal{I} 均有解。故当 $n = |I| + |K| \geq 2k$ 时, 点集 I 和点集 K 至少其中之一的大小不超过 k , 此时 \mathcal{I} 必然有解。

综上所述, 引理得证。 ■

基于引理7.21和引理7.22, 便可以得到一个基于均摊思想的精确算法。

算法 7-2 求解 SFVS(S) 问题的基于均摊思想的精确算法

```

输入: SFVS(S) 问题的实例  $\mathcal{I} = (\mathbf{G}, T, k)$ 。
输出: 实例  $\mathcal{I}$  有解或无解
1 if  $\mathcal{I}$  不是  $T$ -可分的 then
2   | 根据引理7.21, 在多项式时间内归约至  $T$ -可分实例并更新  $\mathcal{I}$ 。// 此步骤
   | 只会执行一次且执行完成后,  $\mathcal{I}$  总是  $T$ -可分的。
3 end
4 if  $n \leq 2k$  then
5   | 依据引理7.22, 直接返回“有解”。
6 else
7   | 依据引理7.4, 调用运行时间为  $\mathcal{O}^*(1.8192^k)$  的参数算法求解实例  $\mathcal{I}$ 。
8 end

```

定理 7.5 SFVS(S) 问题可以在 $\mathcal{O}(1.3488^n)$ 时间内求解。

证明: 由于当 $n \leq 2k$ 时, 可以在常数时间内得求解。所以, 在调用 SFVS(S) 问题运行时间为 $\mathcal{O}^*(1.8192^k)$ 的参数算法时, 有 $2k \leq n$ 。进而得出算法7-2的时间复杂度为

$$\mathcal{O}^*(1.8192^k) \leq \mathcal{O}^*(1.8192^{n/2}) = \mathcal{O}(1.3488^n) \leq \mathcal{O}(1.3488^n).$$

最后, 结合引理7.11, SFVS(S) 问题可以在 $\mathcal{O}(1.3488^n)$ 时间内求解, 定理得证。 ■

7.4 分裂图上限制版子集反馈点集问题

本节将提出分裂图上限制版子集反馈点集问题的参数算法与精确算法。改进了前人的结果, 并且证明了在指数时间意义上, 该问题与顶点覆盖问题参数算法的时间复杂度相同。同时, 通过给出了以关键点集大小为参数的单指数 FPT 算法, 证明当 $\tau = \mathcal{O}(\log n)$, 分裂图上限制版子集反馈点集问题是多项式时间可解的。作为推论, 本文证明了经典无向图上独立集问题在以团覆盖数为参数时是 FPT 的。主要结论如下。

定理 7.23 R-SFVS(S) 问题可以在 $\mathcal{O}^*(1.2529^k)$ 时间内求解。

定理 7.24 R-SFVS(S) 问题可以在 $\mathcal{O}(1.1520^n)$ 时间内求解。

7.4.1 与顶点覆盖问题的归约关系

对于无向图 \mathbf{G} , 若一族团 \mathcal{C} 覆盖了 \mathbf{G} 中的任何一条边, 称 \mathcal{C} 是 \mathbf{G} 的团覆盖。图 \mathbf{G} 团覆盖数, 记作 $\text{ecc}(\mathbf{G})$, 是其最小团覆盖的大小。团覆盖数是图的基本参数之一。本节将建立 R-SFVS(S) 问题与顶点覆盖问题之间的归约关系, 其中顶点覆盖问题实例的团覆盖数和 R-SFVS(S) 问题的关键点集大小 τ 存在重要的关联性。

引理 7.25 R-SFVS(S) 问题的实例 $\mathcal{I} = (\mathbf{G}, T, k)$ 可以在多项式时间内归约到顶

点覆盖问题的实例 $\mathcal{I}' = (\mathbf{G}', k')$, 满足 $|V(\mathbf{G}')| \leq |V(\mathbf{G})| - \tau$, $k' \leq k$ 。此外, \mathbf{G}' 存在大小不超过 τ 的团覆盖 \mathcal{C} 。

证明: 设 $\mathcal{I} = (\mathbf{G}, T, k)$ 为 R-SFVS(S) 问题的实例, 其中 $|V(\mathbf{G})| = n$ 且 $\tau = |T|$ 。不失一般性, 设 T 是独立集且 $V(\mathbf{G}) \setminus T$ 是团。事实上, 一方面, 若 T 中有 3 个两两相连的关键点, 则 \mathcal{I} 一定无解, 可以直接归约到 MIS(H) 问题的一个平凡无解的实例。另一方面, 若 T 中有 2 个相连的关键点, 那么, 它们的每个共同非关键点邻居 v 必然在解集中, 可以直接删去 v 并使 k 减少 1。所以, 存在 \mathbf{G} 的分裂划分 (I, K) , 满足 $T \subseteq I$ 。此外, 可以进一步假设 $I = T$ 。这是因为, I 中的非关键点不在任何经过关键点的三角形中, 将其删除不会改变实例的等价性。综上, 我们总假设存在分裂划分 (I, K) 使得 $I = T$ 且 $K = V(\mathbf{G}) \setminus T$ 。

现构造顶点覆盖问题的实例 $\mathcal{I}' = (\mathbf{G}', k')$ 。首先, 对于点集 $V(\mathbf{G})$ 中的每个非关键点 $v \in K$, 构造一个顶点 v' , 得到图 \mathbf{G}' 的点集

$$V(\mathbf{G}') = \{v' : v \in K\}.$$

然后, 对于每个关键点 $t \in T$, 构造一个团 $Q_t = \{v' : v \in N_{\mathbf{G}}(t)\}$, 进而得到

$$E(\mathbf{G}') = \bigcup_{t \in T} Q_t = \bigcup_{t \in T} \{v' : v \in N_{\mathbf{G}}(t)\}.$$

根据构造, \mathbf{G}' 是一个包含 $n' = n - \tau$ 个顶点的无向图, 且存在大小为 τ 的团覆盖 $\mathcal{C} = \{Q_t : t \in T\}$ 。因此, 有 $\text{ecc}(\mathbf{G}') \leq \tau$ 。最后令 $k' = k$, 得到了顶点覆盖问题的实例 $\mathcal{I}' = (\mathbf{G}', k')$ 。

下证 \mathcal{I}' 有解当且仅当 \mathcal{I} 有解。首先论证充分性: 若 S' 为实例 \mathcal{I}' 的解, 则

$$S = \{v \in V(\mathbf{G}) : v' \in S'\}$$

是 \mathcal{I} 的解。考虑图 \mathbf{G} 中的任意 T -三角形 \mathbf{C} , 由于关键点集是独立集 I , 那么 \mathbf{C} 恰好包含一个关键点 $t \in I$ 和两个非关键点 $v, u \in K$ 。根据构造, 有 $v', u' \in Q_t$ 。由于 S' 是图 \mathbf{G}' 的顶点覆盖, 因此顶点 v' 和 u' 至少其中之一在 S' 中, 这意味着顶点 v 和 u 至少其中之一在 S 中。所以, $\mathbf{G} - S$ 中没有 T -三角形, 进而说明 S 是 T -反馈点集。注意到

$$|S| = |S'| \leq k' = k.$$

所以, S 是 \mathcal{I} 的解。

接着论证必要性: 若 S 为实例 \mathcal{I} 的解, 则

$$S' = \{v' \in V(\mathbf{G}') : v \in S\}$$

是 \mathcal{I}' 的解。考虑图 \mathbf{G}' 的团覆盖 \mathcal{C} 中任意一个团 Q_t , 设 $v', u' \in Q_t$ 。根据构造, 可知 (t, v, u) 是 T -三角形。由于 S 仅包含非关键点, 所以顶点 v 和 u 至少其中之一在 S 中, 这意味着顶点 v' 和 u' 至少其中之一在 S' 中。所以, S 是独立集。注意到

$$|S'| = |S| \leq k = k'.$$

所以, S' 是 \mathcal{I}' 的解。

综上所述, 引理得证。 \blacksquare

另一方向, 顶点覆盖问题也可以归约到 R-SFVS(S) 问题。Fomin 等人^[177]给出了一个简单归约, 从而证明了 R-SFVS(S) 是 NP-难的。但在本文中, 需要给出一个更强的归约构造。

引理 7.26 给定顶点覆盖问题实例 $\mathcal{I}' = (\mathbf{G}', k')$ 的团覆盖 \mathcal{C} , \mathcal{I}' 可以在多项式时间内归约到 R-SFVS(S) 问题的实例 $\mathcal{I} = (\mathbf{G}, T, k)$, 满足 $|V(\mathbf{G})| \leq |V(\mathbf{G}')| + |\mathcal{C}|$, $\tau \leq |\mathcal{C}|$ 以及 $k \leq k'$ 。此外, $(T, V(\mathbf{G}) \setminus T)$ 是图 \mathbf{G} 的一个分裂划分, 并且 T 中关键点的度数不超过 \mathcal{C} 中最大团的大小。

证明: 设 $\mathcal{I}' = (\mathbf{G}', k')$ 为 MIS(H) 问题的实例, 其中 $|V(\mathbf{G}')| = n'$ 且 $|\mathcal{C}| = \tau'$ 。现构造 R-SFVS(S) 问题的实例 $\mathcal{I} = (\mathbf{G}, T, k)$ 。对于点集 $V(\mathbf{G}')$ 中的每个顶点 v' , 构造一个顶点 v , 得到集合 $K = \{v : v' \in V(\mathbf{G}')\}$; 团覆盖 \mathcal{C} 中每个团 Q , 构造一个顶点 t_Q , 得到集合 $I = \{t_Q : Q \in \mathcal{C}\}$ 。进而得到图 \mathbf{G} 的点集

$$V(\mathbf{G}) = K \cup I = \{v : v' \in V(\mathbf{G}')\} \cup \{t_Q : Q \in \mathcal{C}\}.$$

令 K 为团, I 为独立集, 并且对于 K 中任意顶点 v 和 I 中任意顶点 t_Q , 满足 v 与 t_Q 相连当且仅当 $v' \in Q$, 即

$$E(\mathbf{G}) = \{vu : v, u \in K\} \cup \{vt_Q : v' \in Q\}.$$

根据构造, \mathbf{G} 是包含 $n' + \tau'$ 个顶点的分裂图, 并且 (I, K) 是分裂划分。此外, 有关系 $|Q| = \deg_{\mathbf{G}}(t_Q)$, 所以 \mathbf{G} 中的每个关键点的度不超过 \mathcal{C} 中最大团的大小。最后令 $T = I$ 且 $k = k'$, 得到了 R-SFVS(S) 问题的实例 $\mathcal{I} = (\mathbf{G}, T, k)$ 。

下证 \mathcal{I}' 有解当且仅当 \mathcal{I} 有解。首先论证充分性: 若 S' 为实例 \mathcal{I}' 的解, 则

$$S = \{v \in V(\mathbf{G}) : v' \in S'\}$$

是 \mathcal{I} 的解。考虑图 \mathbf{G} 中的任意 T -三角形 \mathbf{C} , 由于关键点构成独立集, 那么 \mathbf{C} 恰好包含一个关键点 $t_Q \in I$ 和两个非关键点 $v, u \in K$ 。根据构造, 有 $v', u' \in Q$ 。由于 S' 是图 \mathbf{G}' 的顶点覆盖, 因此顶点 v' 和 u' 至少其中之一在 S' 中, 这意味着顶点 v 和 u 至少其中之一在 S 中。所以, $\mathbf{G} - S$ 中没有 T -三角形, 进而说明 S 是子集反

馈点集。注意到

$$|S| = |S'| \leq k' = k.$$

所以， S 是 \mathcal{I} 的解。

接着论证必要性：若 S 为实例 \mathcal{I} 的解，则

$$S' = \{v' \in V(\mathbf{G}') : v \in S\}$$

是 \mathcal{I}' 的解。考虑图 \mathcal{C} 中的任意一个团 Q ，设 $v', u' \in Q$ 。根据构造，可知 (t_Q, v, u) 是 T -三角形。由于 S 仅包含非关键点，所以顶点 v 和 u 至少其中之一在 S 中，这意味着顶点 v' 和 u' 至少其中之一在 S' 中。所以， S' 是顶点覆盖。注意到

$$|S'| = |S| \leq k = k'.$$

所以， S' 是 \mathcal{I}' 的解。

综上所述，引理得证。 ■

7.4.2 基于归约的参数算法

定理 7.27 R-SFVS(S) 问题存在顶点数为 $k^2 + 2k$ 且边数为 $5k^2$ 的核。

证明：设 $\mathcal{I} = (\mathbf{G}, T, k)$ 为 R-SFVS(S) 问题的实例，设 \mathbf{G} 包含 $n = |V(\mathbf{G})|$ 个顶点，其中有 $\tau = |T|$ 个关键点。首先，依据引理 7.25，可以在多项式时间内将 \mathcal{I} 归约到顶点覆盖问题的实例 $\mathcal{I}_1 = (\mathbf{G}_1, k_1)$ ，其中 $n_1 = |V(\mathbf{G}_1)| \leq n - \tau$ 且 $k_1 \leq k$ 。然后，使用顶点覆盖问题已有的核心化算法^[100, 101]，得到实例 \mathcal{I}_1 的线性核 $\mathcal{I}_2 = (\mathbf{G}_2, k_2)$ ，其中 $n_2 = |V(\mathbf{G}_2)| \leq 2k_1$ 且 $k_2 \leq k_1$ 。根据 Erdős 等人^[209]的工作，无向图 \mathbf{G}' 可以被不超过 $|V(\mathbf{G}')|^2/4$ 个三角形或边覆盖，这些三角形或边能在多项式时间计算得到。因此，图 \mathbf{G}' 总存在大小不超过 $|V(\mathbf{G}')|^2/4$ 的团覆盖 \mathcal{C} ，其中的团为一些三角形或边。最后，依据引理 7.26，在已知 \mathcal{C} 的情况下，可以在多项式时间内将 \mathcal{I}_2 归约到 R-SFVS(S) 问题的实例 $\mathcal{I}' = (\mathbf{G}', T', k')$ ，满足 $|V(\mathbf{G}')| \leq n_2 + |\mathcal{C}|$ ， $|T'| \leq |\mathcal{C}|$ 以及 $k' \leq k_2$ 。

现分析实例 \mathcal{I}' 的大小，根据四个实例 \mathcal{I} , \mathcal{I}_1 , \mathcal{I}_2 , \mathcal{I}' 的大小关系，可以得出

$$|V(\mathbf{G}')| \leq n_2 + |\mathcal{C}| = n_2 + \frac{1}{4}n_2^2 \leq 2k_1 + \frac{1}{4}(2k_1)^2 \leq k^2 + 2k.$$

此外, 由于 \mathcal{C} 中的团为一些三角形和边, 因此 T' 中关键点的度数至多为 3, 故有

$$\begin{aligned} |E(\mathbf{G}')| &\leq 3|T'| + \frac{1}{2}|V(\mathbf{G}')|(|V(\mathbf{G}')| - 1) \\ &\leq 3|\mathcal{C}| + \frac{1}{2}|V(\mathbf{G}')|^2 \\ &\leq \frac{5}{4}|V(\mathbf{G}')|^2 \leq \frac{5}{4}n_2^2 \leq \frac{5}{4}(2k_1)^2 \leq 5k^2. \end{aligned}$$

综上所述, 定理得证。 ■

根据现有的结论, 除非 $\text{NP} \subseteq \text{coNP/poly}$, 对于任意正实数 $\varepsilon > 0$, 顶点覆盖问题不存在边数为 $\mathcal{O}(k^{2-\varepsilon})$ 的核^[246]。所以, 引理7.26表明 R-SFVS(S) 问题存在核心化下界。

定理 7.28 除非 $\text{NP} \subseteq \text{coNP/poly}$, 对于任意正实数 $\varepsilon > 0$, R-SFVS(S) 问题不存在边数为 $\mathcal{O}(k^{2-\varepsilon})$ 的核。

定理7.27和定理7.28共同说明了 R-SFVS(S) 问题关于边数的平方核是紧的。

引理 7.29 对于任意实数 $\alpha > 1$, R-SFVS(S) 问题能在 $\mathcal{O}^*(\alpha^k)$ 时间内求解当且仅当顶点覆盖问题能在 $\mathcal{O}^*(\alpha^{k'})$ 时间内求解, 其中 k 和 k' 分别为 R-SFVS(S) 问题和顶点覆盖问题实例的解集大小。

证明: 首先证明必要性: 若顶点覆盖问题的实例 $\mathcal{I}' = (\mathbf{G}', k')$ 能在 $\mathcal{O}^*(\alpha^{k'})$ 时间内求解 ($\alpha > 1$), 则 R-SFVS(S) 问题的实例 $\mathcal{I} = (\mathbf{G}, T, k)$ 能在 $\mathcal{O}^*(\alpha^k)$ 时间内求解。根据引理7.25, 实例 \mathcal{I} 可以在多项式时间内归约到 \mathcal{I}' , 满足 $k' \leq k$ 。故可调用顶点覆盖问题的参数算法在 $\mathcal{O}^*(\alpha^{k'}) = \mathcal{O}^*(\alpha^k)$ 时间内求解判定 \mathcal{I} 是否有解。

接着证明充分性: 若 R-SFVS(S) 问题的实例 $\mathcal{I} = (\mathbf{G}, T, k)$ 能在 $\mathcal{O}^*(\alpha^k)$ 时间内求解 ($\alpha > 1$), 则顶点覆盖问题的实例 $\mathcal{I}' = (\mathbf{G}', k')$ 能在 $\mathcal{O}^*(\alpha^{k'})$ 时间内求解。显然, 图 \mathbf{G}' 存在大小为 $|E(\mathbf{G}')|$ 的团覆盖 \mathcal{C} , 即 \mathbf{G}' 中的每条边构成 \mathcal{C} 中的团。根据引理7.26, 实例 \mathcal{I}' 可以在多项式时间内归约到 \mathcal{I} , 满足 $k \leq k'$ 。所以可以调用 R-SFVS(S) 问题的参数算法在 $\mathcal{O}^*(\alpha^k) = \mathcal{O}^*(\alpha^{k'})$ 时间内求解判定 \mathcal{I}' 是否有解。

综上所述, 引理得证。 ■

定理 7.23 R-SFVS(S) 问题可以在 $\mathcal{O}^*(1.2529^k)$ 时间内求解。

证明: 顶点覆盖问题最快的参数算法的运行时间为 $\mathcal{O}^*(1.2529^k)$ ^[98]。结合引理7.29, 可以说明 R-SFVS(S) 能在 $\mathcal{O}^*(1.2529^k)$ 时间内求解。定理得证。 ■

7.4.3 基于动态规划的参数算法

本节将研究 R-SFVS(S) 问题以关键点数量为参数的参数算法。引理7.25表明, R-SFVS(S) 问题可在多项式时间内归约至顶点覆盖问题, 且实例中的团覆盖数被关键集大小所控制。基于此观察, 可考虑以团覆盖数为参数的独立集问题。

引理 7.30 给定无向图 \mathbf{G} 的团覆盖 \mathcal{C} , 可以在 $\mathcal{O}^*(2^{|\mathcal{C}|})$ 时间和空间内求解图 \mathbf{G} 的一个最大独立集。

证明: 设 $\mathcal{C} = \{Q_1, Q_2, \dots, Q_\kappa\}$, 其中 κ 为团覆盖 \mathcal{C} 的大小。定义顶点的编码 $\sigma: v \mapsto [\sigma_1(v), \sigma_2(v), \dots, \sigma_\kappa(v)]$, 满足

$$\sigma_i(v) = \begin{cases} 0, & v \in Q_i; \\ 1, & v \notin Q_i. \end{cases} \quad (7-1)$$

同时, 对于点子集 $V' \subseteq V(\mathbf{G})$, 定义

$$\sigma(V') := \sum_{v \in V'} \sigma(v) = \left[\sum_{v \in V'} \sigma_1(v), \sum_{v \in V'} \sigma_2(v), \dots, \sum_{v \in V'} \sigma_\kappa(v) \right]. \quad (7-2)$$

若两个向量 $\mathbf{x} = (x_1, x_2, \dots, x_\kappa)$ 和 $\mathbf{y} = (y_1, y_2, \dots, y_\kappa)$ 满足 $x_i \leq y_i$ ($i \in [\kappa]$), 则写作 $\mathbf{x} \leq \mathbf{y}$; 否则写作 $\mathbf{a} \not\leq \mathbf{b}$ 。并且用符号 $\mathbf{0}$ 和 $\mathbf{1}$ 表示全零和全一向量。

考虑图 \mathbf{G} 中的一对顶点 v 和 u 。一方面, 若 v 和 u 之间有一条边, 那么根据团覆盖的定义, 边 uv 必然存在于 \mathcal{C} 中的某个团中。所以 v 和 u 必然同时出现在 \mathcal{C} 中的某个团中, 进而存在下标 $i \in [\kappa]$, 使得 $\sigma_i(\{v, u\}) \geq 2$ 成立。另一方面, 若指标 $i \in [\kappa]$ 满足 $\sigma_i(\{v, u\}) \geq 2$, 那么根据编码 σ 的定义, 当顶点 v 和 u 同在团 Q_i 中时, 意味着 vu 为一条无向边。所以, 我们得出结论: 点子集 $V' \subseteq V(\mathbf{G})$ 是独立集当且仅当 $\sigma(V') \leq \mathbf{1}$ 成立, 基于此结论, 可以将独立集问题转化为找到一个最大的点子集 $V' \subseteq V(\mathbf{G})$ 使得 $\sigma(V') \leq \mathbf{1}$ 成立。

可以在指数时间和空间内采用动态规划求解该问题。对顶点进行编号 $V(\mathbf{G}) = \{v_1, v_2, \dots, v_n\}$ 。令 $V_0 := \emptyset$ 以及 $V_i := V_{i-1} \cup \{v_i\}$ ($i \in [n]$)。用 \mathcal{X} 表示所有 κ 维 0-1 向量, 那么 \mathcal{X} 包含了 2^κ 个元素。

对于任何下标 $i \in [n]$ 和任意向量 $\mathbf{x} \in \mathcal{X}$, 用符号 $\text{dp}[\mathbf{x}, i]$ 表示最大的点子集 $V' \subseteq V_i$ 使得 $\sigma(V') = \mathbf{x}$ 成立。可以建立 $\text{dp}[\cdot, \cdot]$ 的递推关系式如下: 首先有边界条件 $\text{dp}[\mathbf{0}, 0] = 0$ 以及 $\text{dp}[\mathbf{x} \neq \mathbf{0}, 0] = \infty$; 然后, 对任意下标 $i \geq 1$ 有

$$\text{dp}[\mathbf{x}, i] = \begin{cases} \max \{ \text{dp}[\mathbf{x} - \sigma(v_i), i-1] + 1, \text{dp}[\mathbf{x}, i-1] \}, & \sigma(v_i) \leq \mathbf{x}; \\ \text{dp}[\mathbf{x}, i-1], & \sigma(v_i) \not\leq \mathbf{x}. \end{cases}$$

因此, 通过自底向上的计算方式, 可以用函数值 $\text{dp}[\mathbf{x}', i']$ ($\mathbf{x}' \in \mathcal{X}$ 和 $i' < i$) 来计算 $\text{dp}[\mathbf{x}, i]$ 。显然, 对于固定的向量 \mathbf{x} 和下标 i , 只需要常数时间计算并存储 $\text{dp}[\mathbf{x}, i]$ 。下标 i 一共有 $n+1$ 个取值而 \mathbf{x} 共有 2^κ 个候选向量, 所以一共需要花费 $\mathcal{O}(n2^\kappa)$ 时间进行计算。

在得到了所有形如 $\text{dp}[\mathbf{x}, i]$ 的函数值和, 便可以通过求 $\max_{\mathbf{x} \in \mathcal{X}} \text{dp}[\mathbf{x}, n]$ 得到图 \mathbf{G} 的最大独立集, 引理得证。 ■

结合引理7.25和引理7.30，可得出定理7.31。

定理 7.31 SFVS(S) 问题可以在 $\mathcal{O}^*(2^\tau)$ 时间和空间内求解。

注意到，可以在 $\mathcal{O}^*(2^{\kappa})$ 时间内找到无向图的团覆盖数 κ ^[266]。这说明了当以团覆盖数为参数时，独立集问题是 FPT 的。值得强调的是，以解集大小为参数时，独立集问题是 W[1]-难的^[39,40]。

定理 7.32 独立集问题在以团覆盖数为参数是 FPT 的。

7.4.4 基于分支搜索技术的精确算法

本节将深入研究 R-SFVS(S) 的快速精确算法。

首先，使用简单的均摊思想便可快速得到一个时间复杂度较好的精确算法。根据引理7.25，可使用顶点覆盖问题的 $\mathcal{O}(1.19951^n)$ 时间的精确算法^[34]求解 R-SFVS(S) 问题，从而得到了引理7.33。

引理 7.33 R-SFVS(S) 问题可以在 $\mathcal{O}^*(1.19951^{n-\tau})$ 时间内求解。

再结合定理7.31，便可得到引理7.34。

引理 7.34 R-SFVS(S) 问题可以在 $\mathcal{O}(1.1550^n)$ 时间和 $\mathcal{O}^*(2^\tau)$ 空间内求解。

注意到，引理7.34中的精确算法是一个指数空间的精确算法。本文将基于分支搜索技术，采用度量治之分析手段得到一个更高效的多项式空间精确算法。

引理 7.35 对于任意实数 $\alpha > 1$ ，R-SFVS(S) 问题能在 $\alpha^n n^{\mathcal{O}(1)}$ 时间内求解，当且仅当 SFVS(S) 问题的 T -可分实例能在 $\alpha^n n^{\mathcal{O}(1)}$ 时间内求解。

证明：由于 T -可分实例是 R-SFVS(S) 问题的特殊实例，因此只需证明，若约简实例能 $\alpha^n n^{\mathcal{O}(1)}$ 时间内求解，那么 R-SFVS(S) 问题便可在 $\alpha^n n^{\mathcal{O}(1)}$ 时间内求解。

设 $\mathcal{I} = (\mathbf{G}, T, k)$ 是 R-SFVS(S) 问题的实例。首先在多项式时间内计算 \mathbf{G} 的分裂划分 (I, K) ，其中 I 为独立集， K 为团。记 $I' = I \setminus T$ 且 $K' = K \cap T$ 。

根据引理7.2，可以在多项式时间内判断是否存在一个最小 T -反馈点集，它包含了某些 I 中的某个非关键点，或者不包含 K 中某个关键点。若存在，则直接返回“有解”；否则可以将集合 I' 和集合 K' 直接删去，然后使 k 减少 $|K'|$ 。这便得到了一个 T -可分实例 $\mathcal{I}' = (\mathbf{G} - I' - K', T \setminus K', k - |K'|)$ 。最后，调用 T -可分实例的算法，在 $\alpha^n n^{\mathcal{O}(1)}$ 时间内求解。总时间复杂度亦为 $\alpha^n n^{\mathcal{O}(1)}$ ，引理得证。 ■

根据引理7.35，本节接下来总假设输入实例是 T -可分的。

本小节用符号 $N_T(v) := \{u \in V(\mathbf{G}) : \exists t \in T, u, v \in N(t)\}$ 表示与非关键点 v 构成了某个 T -三角形的非关键点集合。本节提出的算法的核心思想是对一个非关键点 v 进行分支，要么 v 在解集中，要么 $N_T(v)$ 在解集中。

现给出 R-SFVS(S) 问题的 T -可分实例 $\mathcal{I} = (\mathbf{G}, T, k)$ 的度量函数。首先定义各

顶点的权重：非关键点的权重为 1；度数为 d 关键点的权重为 $\omega_d = \min \{ 1, \lambda \cdot d \}$ ，其中 $\lambda = 1/7$ 。然后，实例 \mathcal{I} 的度量函数定义为各顶点权重之和，即

$$\mu(\mathcal{I}) = n - \tau + \sum_{r \in \mathbb{N}} \omega_r \cdot n_r,$$

其中 n 表示顶点数量， τ 表示关键点数量， n_d 表示为度为 d 的关键点数量。

现给出 6 条约简规则和 3 条分支规则，任意一条规则只有在前述的规则均不可执行时才会开始执行。所有规则中仅有删除顶点操作，而一个 T -可分实例在删除顶点后，所得到的实例仍然是 T -可分实例。此外，由于每个顶点的权重均为非负值，所以删除顶点不会使得度量函数增加。综上所述，每条规则得到的实例必然是 T -可分实例，且度量函数不增。

约简规则 7.4.1 (无解约简规则) 若 $k < 0$ 或，则返回“无解”。

约简规则 7.4.2 (有无解约简规则) 若图 \mathbf{G} 为空图且 $k \geq 0$ ，则返回“有解”。

约简规则 7.4.3 (无效顶点约简规则) 若存在顶点不在任何 T -三角形中，则将其删除。

前三条约简规则的正确性是平凡的。若上述三条约简规则均不可执行，那么每个关键点的度至少为 2。

约简规则 7.4.4 (无效关键点约简规则) 若存在两个关键点 t 和 t' ，它们的公共邻居完全相同，则删除 t 。

引理 7.36 无效关键点约简规则 7.4.4 是正确的。

证明：设输入实例为 $\mathcal{I} = (\mathbf{G}, T, k)$ 。一方面， \mathcal{I} 的解必然是子实例 $(\mathbf{G} - t, T \setminus \{t\}, k)$ 的解。另一方面，设 S 为子实例 $(\mathbf{G} - t, T \setminus \{t\}, k)$ 的解，那么至多只有 t' 的一个邻居不在 S 中。由于 $N(t) = N(t')$ ，所以 $\mathbf{G} - S$ 中 t 是一个 1 度点。这说明了 S 也为 \mathcal{I} 的解。

综上所述，引理得证。 ■

约简规则 7.4.5 (非关键点约简规则) 若非关键点 v 只与一个关键点相连，则删除 $N_T(v)$ 并放入解，并使 k 减少 $|N_T(v)|$ 。

引理 7.37 非关键点约简规则 7.4.5 是正确的。

证明：设输入实例为 $\mathcal{I} = (\mathbf{G}, T, k)$ 。令 S 为 \mathcal{I} 的包含顶点 v 的解。若 $S \setminus \{v\}$ 依然为解，那么有 $N_T(v) \subseteq S$ ，非关键点约简规则正确。若 $S \setminus \{v\}$ 不再是解，存在 t 的唯一的另一个邻居 $u \neq v$ 不在解 S 中。那么 $S' = (S \setminus \{u\}) \cup \{v\}$ 是 T -反馈点集。此外，由于 $|S'| = |S| \leq k$ ，所以 S' 也是解。

其次，注意到 v 仅与 t 相连，所以所有经过 v 的 T -三角形都经过 t 。由于 $N_T(v) \subseteq S'$ ，所以在图 $\mathbf{G} - S'$ 中， v 是 t 唯一的邻居， t 不在 T -三角形中。因此，

S' 也是实例 \mathcal{I} 的不包含 v 的解。最终说明了总存在解不包含顶点 v ，此时集合 $N_T(v)$ 必然在解中，所以非关键点约简规则是正确的。

综上所述，引理得证。 ■

约简规则 7.4.6 (低度关键点约简规则) 若关键点的度均不超过 7，则使用引理7.33中的算法直接求解。

引理 7.38 低度关键点约简规则7.4.6是正确的，并且可以在 $\mathcal{O}(1.15199^\mu)$ 时间内求解。

证明：设 $D = \sum_{t \in T} \deg(t)$ 为所有关键点度数之和。根据条件，关键点的度数均不超过 7。所以

$$\mu = n + \sum_{t \in T} \lambda \cdot \deg(t) = n + \lambda \cdot D.$$

进一步，每个非关键点至少与两个关键点相邻，那么 $D \geq 2n$ ，这意味着

$$\mu = n + \lambda \cdot D \geq n + 2\lambda n = \frac{9n}{7}.$$

所以，关键点的度均不超过 7 时，实例 \mathcal{I} 可在

$$\mathcal{O}(1.19951^n) \leq \mathcal{O}(1.19951^{7\mu/9}) = \mathcal{O}(1.15199^\mu).$$

时间内求解。 ■

分支规则 7.4.7 (多公共邻居高度关键点分支规则) 如果存在两个度至少为 8 的关键点 $t_1, t_2 \in T$ 满足 $|N(t_1) \cap N(t_2)| \geq 2$ ，则令 $X = N(t_1) \cap N(t_2)$ 且 $Y = (N(t_1) \cup N(t_2)) \setminus X$ ，然后分别计算以下两个子分支。若两个子分支均无解，则返回“无解”；否则返回“有解”。

- 删除集合 X ，使 k 减少 $|X|$ ；
- 删除集合 Y 和关键点 t ，使 k 减少 $|Y|$ 。

引理 7.39 多公共邻居高度关键点分支规则7.4.7是正确的，且分支因子至多为 1.13863。

证明：令 $Y_1 = N(t_1) \setminus X$ 以及 $Y_2 = N(t_2) \setminus X$ 。对于任意解集 S ， $N(t_1)$ 中至多有一个顶点不在 S 中。因为 (X, Y_1) 是 $N(t_1)$ 的划分，所以要么 $X \subseteq S$ 成立，要么 $Y_1 \subseteq S$ 成立。类似地，可以得出 $X \subseteq S$ 成立，要么 $Y_2 \subseteq S$ 成立。联合可得，要么 $X \subseteq S$ 成立，要么 $Y = Y_1 \cup Y_2 \subseteq S$ 成立。因此，多公共邻居高度关键点分支规则是正确的。

现分析多公共邻居高度关键点分支规则的分支因子，令 δ_1 和 δ_2 分别表示第一个和第二个分支中度量函数的减少数值。设 $X_i = N(t_i) \setminus X$ ($i \in [2]$)。由于无效关键点约简规则不可执行，可知 X_1 和 X_2 均是非空集合。

在第一个分支中, 集合 X 被删除, 且 t_i 的度减少至 $|X_i|$ ($i \in [2]$)。故可得

$$\delta_1 = |X| + 2 - \min \{ 1, \lambda \cdot |X_1| \} - \min \{ 1, \lambda \cdot |X_2| \} \geq |X|.$$

在第二个分支中, $Y = X_1 \cup X_2$ 以及 t_1 被删除, 并且 t_2 的度减少至 $|X|$ 。由于 t_1 和 t_2 的度至少为 8, 可以得出

$$\delta_2 = |X_1| + |X_2| + 2 - \min \{ 1, \lambda \cdot |X| \} \geq 18 - 2|X| - \min \{ 1, \lambda \cdot |X| \}.$$

在多公共邻居高度关键点分支规则中, X 的大小至少为 2 且至多为 7, 总共 7 种情况的分支向量及其分支因子如表7-2所示。

表 7-2 多公共邻居高度关键点分支规则中的分支向量及其分支因子

$ X $ 的值	$ X_1 $ 的值	$ X_2 $ 的值	分支向量 (不劣于)	分支因子 (至多)
2	≥ 6	≥ 6	$(2, 14 - 2\lambda)$	1.12216
3	≥ 5	≥ 5	$(3, 12 - 3\lambda)$	1.11607
4	≥ 4	≥ 4	$(4, 10 - 4\lambda)$	1.11600
5	≥ 3	≥ 3	$(5, 8 - 5\lambda)$	1.12100
6	≥ 2	≥ 2	$(6, 6 - 6\lambda)$	1.13278
7	1	1	$(9 - 2\lambda, 3)$	1.13863
7	1	≥ 2	$(7, 4)$	1.13819

当 $2 \leq |X| \leq 6$ 时, 分支向量不劣于 $(2, 14 - 2\lambda)$, $(3, 12 - 3\lambda)$, $(4, 10 - 4\lambda)$, $(5, 8 - 5\lambda)$ 或 $(6, 6 - 6\lambda)$, 它们的分支因子至多为 1.13278。

当 $|X| \geq 7$ 时, 有 $\delta_1 \geq 7$ 且 $\delta_2 = |X_1| + |X_2| + 1$ 。若 $|X_1| = |X_2| = 1$, 那么分支向量为 $(9 - 2\lambda, 3)$, 对应的分支因子为 1.13863。否则有 $|X_1| + |X_2| \geq 3$, 进而分支向量不劣于 $(7, 4)$, 对应的分支因子为 1.13819。

综上所述, 多公共邻居高度关键点分支规则的分支因子至多为 1.13863, 引理得证。 ■

分支规则 7.4.8 (唯一公共邻居高度关键点分支规则) 如果存在一个顶点 v 与两个度数至少为 8 的关键点相连, 则分别计算以下两个子分支。若两个子分支均“无解”, 则返回“无解”; 否则返回“有解”。

- 删除顶点 v , 使 k 减少 1;
- 删除集合 $N_T(v)$ 以及集合 $N(v) \cap T$, 使 k 减少 $|N_T(v)|$ 。

引理 7.40 唯一公共邻居高度关键点分支规则7.4.8是正确的, 且分支因子至多为 1.13391。

证明: 唯一公共邻居高度关键点分支规则的正确性是平凡的, 现分析其分支因子。

由于非关键点约简规则不可执行，每个非关键点至少连接了两个关键点。令 $t_1, t_2 \in N(v)$ 为连接了 v 的两个关键点。又由于多公共邻居高度关键点分支规则不可执行，任意两个度至少为 8 的关键点至多有一个公共邻居。因此有 $N(t_1) \cap N(t_2) = \{v\}$ 。所以，

$$|N_T(v)| \geq |N(t_1) \cup N(t_2)| = |N(t_1)| + |N(t_2)| - |N(t_1) \cap N(t_2)| \geq 8 + 8 - 1 = 15.$$

此外，在删除了 $N_T(v)$ 以后，所有与 v 相邻的关键点都变成了一度点，所以可以将 $N_v \cap T$ 中的顶点删除。因为 $t_1, t_2 \in N_v \cap T$ ，所以 $|N_v \cap T| \geq 2$ ，进而得到

$$|N_T(v)| + |N(v) \cap T| \geq 15 + 2 \geq 17.$$

综上，唯一公共邻居高度关键点分支规则的分支向量不劣于 $(1, 17)$ ，其对应的分支因子为 1.13391，引理得证。 ■

分支规则 7.4.9 (高度关键点分支规则) 选择一个度数至少为 8 的关键点 t ，则将 t 的邻居进行二划分 $N(t) = X \cup Y$ 使得 $|X| = 4$ 且 $|Y| = \deg(t) - 4$ ，则分别计算以下两个子分支。若两个子分支均“无解”，则返回“无解”；否则返回“有解”。

- 删除集合 X ，使 k 减少 $|X|$ ；
- 删除集合 Y ，使 k 减少 $|Y|$ 。

引理 7.41 高度关键点分支规则 7.4.9 是正确的，且分支因子至多为 1.14870。

证明：对于关键点 t ，其邻居中至多有一个顶点不在解集中。由于 X 和 Y 是 $N(t)$ 的划分，所以要么 X 在解集中，要么 Y 在解集中。所以高度关键点分支规则是正确的。

现分析高度关键点分支规则的分支因子。在高度关键点分支规则中，关键点的度至少为 8，即 $\deg(t) = |X| + |Y| \geq 8$ ，并且满足 $|X| = 4$ 和 $|Y| \geq 4$ 。由于非关键点约简规则不可执行，每个非关键点至少连接了两个关键点。又由于唯一公共邻居高度关键点分支规则不可执行，任意两个度至少为 8 的关键点均无公共邻居。所以，对于任意顶点 $v \in N(t)$ ，存在 t 不同的关键点 t' 与 v 相连，并且总有 $\deg(t') \leq 7$ 。因此，当 v 被删除时，关键点 t' 的度减少了 1，从而度量函数 $\mu(\cdot)$ 至少减少了 λ 。

令 δ_1 和 δ_2 表示第一个和第二个分支中度量 $\mu(\cdot)$ 减少的数值。在第一个分支中，点集 X 中的 4 个顶点被删除，关键点 t 的度变为 $|Y|$ 。所以可以得出

$$\delta_1 = |X| + \lambda \cdot |X| + 1 - \min \{1, \lambda \cdot |Y|\} = 5 + 4\lambda - \min \{1, \lambda \cdot |Y|\}.$$

类似地，在第二个分支中，点集 Y 中的顶点被删除，关键点 t 的度变为 $|X|$ 。

$$\delta_2 = |Y| + \lambda \cdot |Y| + 1 - \min \{1, \lambda \cdot |X|\} = (1 + \lambda) \cdot |Y| + 3\lambda.$$

集合 Y 的大小至少为 4，根据 Y 的大小共分为两种情形，总共 4 种情况的分支向量及其分支因子如表7-3所示。

表 7-3 高度关键点分支规则中的分支向量及其分支因子

$ N_1 $ 的值	$ N_2 $ 的值	分支向量（不劣于）	分支因子（至多）
4	4	(5, 5)	1.14870
4	5	(5 - λ , 6 + λ)	1.13500
4	6	(5 - 2 λ , 7 + 2 λ)	1.12459
4	≥ 7	(4 + 4 λ , 7 + 10 λ)	1.11637

情形 1: $4 \leq |Y| \leq 6$ 。在此情形下，分支向量分别为 $(5, 5)$, $(5 - \lambda, 6 + \lambda)$ 以及 $(5 - 2\lambda, 7 + 2\lambda)$ ，它们的分支因子至多为 1.14870。

情形 2: $|Y| \geq 7$ 。在此情形下，有 $\delta_1 = 4 + 4\lambda$ 且 $\delta_2 \geq 7 + 10\lambda$ ，此时的分支因子至多为 1.11637。

综上所述，高度关键点分支规则的分支因子至多为 1.14870，引理得证。 ■

根据上述规则，现在给出以下分支搜索算法求解 R-SFVS(S) 问题。

算法 7-3 求解 R-SFVS(S) 问题的基于分支搜索技术的精确算法

输入: R-SFVS(S) 问题的实例 $\mathcal{I} = (\mathbf{G}, T, k)$ 。

输出: 实例 \mathcal{I} 有解或无解

- 1 **if** \mathcal{I} 不是 T -可分的 **then**
- 2 根据引理7.35，在多项式时间内归约到一个 T -可分实例并更新 \mathcal{I} 。// 此步骤只会执行一次且此步骤后， \mathcal{I} 总是 T -可分的。
- 3 **end**
- 4 执行无解约简规则7.4.1和有解约简规则7.4.2。// 此步骤为算法的终止条件。
- 5 执行无效顶点约简规则7.4.3。// 此步骤后，每个关键点至少连接了两个非关键点，每个非关键点至少连接了一个关键点。
- 6 执行无效关键点约简规则7.4.4。// 此步骤后，关键点的邻居两两不同。
- 7 执行非关键点约简规则7.4.5。// 此步骤后，每个非关键点至少连接了两个关键点。
- 8 执行低度关键点约简规则7.4.6。// 此步骤后，必然存在度至少 8 的关键点。
- 9 执行多公共邻居高度关键点分支规则7.4.7。// 此步骤后，任意两个度至少 8 的关键点最多有一个公共邻居。
- 10 执行唯一公共邻居高度关键点分支规则7.4.8。// 此步骤后，任意两个度至少 8 的关键点没有公共邻居。
- 11 执行高度关键点分支规则7.4.9。

定理 7.24 R-SFVS(S) 问题可以在 $\mathcal{O}(1.1520^n)$ 时间内求解。

证明: 基于引理7.39, 引理7.40和引理7.41, 算法7-3中三条分支规则的分支因子均不超过 1.15199。注意到

$$\mu = \mu(\mathcal{I}) = n - \tau + \sum_{r \in \mathbb{N}} \omega_d \cdot n_d \leq n - \tau + \tau = n.$$

因此,结合引理7.38可以得出, R-SFVS(S) 问题的 T -可分实例可以在 $\mathcal{O}(1.15199^\mu) \leq \mathcal{O}(1.15199^n)$ 时间内求解。最后, 根据引理7.35, R-SFVS(S) 问题可以在 $\mathcal{O}(1.1520^n)$ 时间内求解。定理得证。 ■

7.5 超图上带惩罚的独立集问题

由于分裂图和超图有着非常相似的结构。本节将说明分裂图上子集反馈点集问题可以用于求解超图上的一类独立集问题, 称为超图上带惩罚的独立集问题, 简记为 PC-MIS(H)。

借助 SFVS(S) 的参数算法, 本文给出了 PC-MIS(H) 问题的首个非平凡的精确算法。主要结论如下。

定理 7.42 PC-MIS(H) 问题可以在 $\mathcal{O}^*(1.8192^n)$ 时间内求解, 其中 n 为超图的顶点数量。

在图论中, 超图是经典图模型的一种推广, 其中的超边可以连接任意数量的顶点。本节中, 超图用符号 \mathbf{H} 表示, 其顶点集表示为 $V(\mathbf{H})$; 超边集是一个多重集, 表示为 $E(\mathbf{H})$, 其中的元素称为超边 (Hyperedge)。每条超边 $e \in E(\mathbf{H})$ 对应 $V(\mathbf{H})$ 的一个子集。为区分重边, 使用 $N_{\mathbf{H}}(e)$ 表示超边 e 所包含的顶点集合, 该顶点集的大小称为超边 e 的秩 (Rank), 记为 $\text{rank}(e) = |N_{\mathbf{H}}(e)|$ 。若超图 \mathbf{H} 中超边的秩最大为 r , 则称为这个 \mathbf{H} 的秩为 r , 或称 \mathbf{H} 为 r -秩图。通常, 超图中允许存在秩为 0 或 1 的超边, 也允许存在重边, 即允许存在超边 $e_1 \neq e_2$ 满足 $N_{\mathbf{H}}(e_1) = N_{\mathbf{H}}(e_2)$ 。在不产生混淆时, 超图 \mathbf{H} 的顶点个数记作 $n = |V(\mathbf{H})|$, 超图 \mathbf{H} 的超边个数记作 $m = |E(\mathbf{H})|$ 。

经典无向图上独立集问题简记为 MIS, 是最基础的 NP-难问题之一。包括 MIS 问题在内的许多经典无向图上的连通性问题都可以自然地推广到超图上^[267-273]。在超图 \mathbf{H} 中, 如果点子集 X 的任意两个顶点不同在一条超边中, 那么 X 称为 \mathbf{H} 的一个大小为 $|X|$ 的独立集或强独立集。换言之, 若超图 \mathbf{H} 删去点集 $V(\mathbf{H}) \setminus X$ 之后的秩至多为 1, 则称 X 为超图 \mathbf{H} 的独立集。因此, MIS 问题可以自然地推广到超图上, 称为超图上的独立集问题, 简记为 MIS(H)。

显然, MIS 问题可以在多项式时间内直接归约到一个秩为 2 的 MIS(H) 问题,

因此 $\text{MIS}(\mathbf{H})$ 是 NP -难的。另一个方向, $\text{MIS}(\mathbf{H})$ 问题也可以多项式时间归约到 MIS 问题。此归约基于一个简单的观察: 对于超图上的一条超边 e , 给 $N_{\mathbf{H}}(e)$ 中任意两个顶点添加一条秩为 2 的边, 然后删去 e (如果 e 的秩为 0 或 1, 则直接删掉), 超边 e 就被替换成了一些秩为 2 的边。如果对超图的所有超边都进行上述操作, 便将超图 \mathbf{H} 转换成了一个 n 阶无向图 \mathbf{G} 。因此, 在以 n 为参数的精确算法意义下, 区分 $\text{MIS}(\mathbf{H})$ 问题和 MIS 问题是无意义的。可以直接使用 MIS 问题 $\mathcal{O}(1.1996^n)$ 时间的精确算法^[34]求解 $\text{MIS}(\mathbf{H})$ 问题, 从而得到了推论 7.43。

推论 7.43 $\text{MIS}(\mathbf{H})$ 问题可以在 $\mathcal{O}^*(1.1996^n)$ 时间内求解。

超图上带惩罚的独立集问题, 即 $\text{PC-MIS}(\mathbf{H})$ 问题, 是 $\text{MIS}(\mathbf{H})$ 问题的松弛化推广。在 $\text{MIS}(\mathbf{H})$ 问题中, 所求的集合必需为独立集 (即满足所有超边上独立性的约束), 这一要求在很多场合过于严格。一个自然的松弛方式是: 可以通过花费一定的惩罚代价获得一个违背少量超边上独立性约束但是更大的点集。因此, $\text{PC-MIS}(\mathbf{H})$ 问题应运而生。给定超图 \mathbf{H} , 如果点子集 $S \subseteq V(\mathbf{H})$ 满足有 $p = \text{pen}(S)$ 条超边包含了 S 中至少两个顶点, 则称 S 为 \mathbf{H} 的价值 (**prize**) 为 $|S| - \text{pen}(S)$ 的松弛独立集, 称 $\text{pen}(S)$ 为 S 的惩罚值。 $\text{PC-MIS}(\mathbf{H})$ 问题旨在允许寻找一个价值最大的点集。

这里强调, 在 $\text{PC-MIS}(\mathbf{H})$ 问题中, 违背任意一条超边上独立性约束的惩罚代价均为 1, 若需将违背某一超边 e 上独立性约束的惩罚代价定义为任意正整数, 可以通过添加多条 e 的重边, 归约到 $\text{PC-MIS}(\mathbf{H})$ 问题上。此外, 在经典的无向图上也可以定义带惩罚的独立集问题, 但是此问题与经典的独立集问题是等价的。 $\text{PC-MIS}(\mathbf{H})$ 问题是一个十分困难的问题, 引理 7.44 证明了 p 为一个不小于 2 的常数时 $\text{PC-MIS}(\mathbf{H})$ 问题也是 NP -难的。

引理 7.44 当 p 的值不超过 1 时, $\text{PC-MIS}(\mathbf{H})$ 问题是多项式时间可解的。当 p 的值是至少为 2 的常数时, $\text{PC-MIS}(\mathbf{H})$ 问题是 NP -难的。

证明: 根据点集的价值的定义, 任何单点集的价值都为 1。所以, 当 $p \leq 1$ 时, $\text{PC-MIS}(\mathbf{H})$ 问题总是有解的, 自然可以在多项式时间内求解。

往证即使 p 为常数且 $p \geq 2$ 时, $\text{PC-MIS}(\mathbf{H})$ 问题依然是 NP -难的。采用 MIS 问题的实例 $\mathcal{I} = (\mathbf{G}, k)$ 进行归约。不妨设 $k \geq p$, 否则, 因为 p 是常数, 可以在 $\mathcal{O}(n^p)$ 时间内判断 \mathcal{I} 是否有解, 然后归约到一个平凡的 $\text{PC-MIS}(\mathbf{H})$ 问题实例。令 $|V(\mathbf{G})| = n$ 以及 $|E(\mathbf{G})| = m$ 。现构造 $\text{PC-MIS}(\mathbf{H})$ 问题的实例 $\bar{\mathcal{I}} = (\mathbf{H}, p)$, 其中 \mathbf{H} 是一个包含 n 个顶点和 $nm + k - p$ 条超边的超图。粗略地讲, 归约的思路通过对原无向图的每条边复制为 n 份拷贝使得它们的独立性限制必须被满足, 然后添加 $k - p$ 条包含所有顶点的超边。形式化地构造如下, 对于任一顶点 $v \in V(\mathbf{G})$, 构造

一个顶点 x_v , 进而得到点集

$$V(\mathbf{H}) = \{x_v : v \in V(\mathbf{G})\}.$$

然后, 对于每条无向边 $uv \in E(\mathbf{G})$, 构造 n 条完全相同的秩为 2 的超边 $e_i^{uv} = \{u, v\}$ ($i \in [n]$), 最后再添加 $k - p$ 条秩为 n 的超边 $\bar{e}_i = V(\mathbf{H})$ ($i \in [k - p]$), 最终得到超边集

$$E(\mathbf{H}) = \{e_i^{uv} = \{u, v\} : uv \in E(\mathbf{G}), i \in [n]\} \cup \{\bar{e}_i = V(\mathbf{H}) : i \in [k - p]\}.$$

显然, 超图 \mathbf{H} 一共包含 $|V(\mathbf{H})| = n$ 个顶点和 $|E(\mathbf{H})| = nm + (k - p)$ 条超边。

下证 \mathcal{I} 有解当且仅当 $\bar{\mathcal{I}}$ 有解。首先论证充分性: 若 S 为实例 \mathcal{I} 的解, 则

$$X = \{x_v \in V(\mathbf{H}) : v \in S\}$$

是 $\bar{\mathcal{I}}$ 的解。由于 S 是图 \mathbf{G} 的独立集, \mathbf{H} 中所有秩为 2 的超边至多包含 X 中的一个顶点。此外, 任意秩为 n 的超边则包含了恰好 $|S| = k$ 个顶点。注意到 $k \geq p \geq 2$, 所以可以计算出 X 的价值至少为 $k - (k - p) = p$, 说明了 X 是 (\mathbf{H}, p) 的解。

接着论证必要性: 设 X 为实例 $\bar{\mathcal{I}}$ 的解, 现说明

$$S = \{v \in V(\mathbf{G}) : x_v \in X\}$$

是 \mathcal{I} 的解。令集合 $\bar{E} \subseteq E(\mathbf{H})$ 表示包含了 X 中至少两个顶点的超边集, 那么有 $|X| - |\bar{E}| \geq p$ 。注意到, 若超边 e_i^{vu} 在超边子集 \bar{E} 中, 那么所有与 e_i^{vu} 包含顶点相同的超边都在 \bar{E} 中, 所以有 $p \leq |\bar{E}| - n \leq 0$, 矛盾。因此, 根据构造, S 是图 \mathbf{G} 的独立集。又因为 \bar{E} 只包含一些秩为 n 的超边, 有 $|\bar{E}| \geq k - p$, 可得

$$|S| = |X| \geq p + |\bar{E}| \geq p + (k - p) \geq k.$$

所以, S 是 \mathcal{I} 的解。 ■

综上所述, 引理得证。 ■

目前尚未已知有比 $\mathcal{O}(2^n)$ 更快的 PC-MIS(H) 问题的精确算法。本节剩余部分将建立 SFVS(S) 问题与 PC-MIS(H) 问题的联系, 试图利用 SFVS(S) 问题的求解算法解决 PC-MIS(H) 问题。

引理 7.45 PC-MIS(H) 问题的实例 $\bar{\mathcal{I}} = (\mathbf{H}, p)$ 可以在多项式时间归约到 R-SFVS(S) 问题的 T -可分实例 $\mathcal{I} = (\mathbf{G}, T, k)$, 并且满足, $|V(\mathbf{G})| \leq |V(\mathbf{H})| + |E(\mathbf{H})|$, $\tau \leq |E(\mathbf{H})|$ 以及 $k \leq |V(\mathbf{H})| - p$ 。

证明: 设 $\bar{\mathcal{I}} = (\mathbf{H}, p)$ 为 PC-MIS(H) 问题的实例, 其中 $|V(\mathbf{H})| = n$ 且 $|E(\mathbf{H})| = m$ 。现构造 SFVS(S) 问题的实例 $\mathcal{I} = (\mathbf{G}, T, k)$ 。对于点集 $V(\mathbf{H})$ 中的每个顶点 x ,

构造一个顶点 v_x , 得到集合 $K = \{v_x : x \in V(\mathbf{H})\}$; 对于超边集 $E(\mathbf{H})$ 中每条超边 e , 构造一个顶点 t_e , 得到集合 $I = \{t_e : e \in E(\mathbf{H})\}$ 。进而得到图 \mathbf{G} 的点集

$$V(\mathbf{G}) = K \cup I = \{v_x : v \in V(\mathbf{H})\} \cup \{t_e : e \in E(\mathbf{H})\}.$$

令 K 为团, I 为独立集, 并且对于 K 中任意顶点 v_x 和 I 中任意顶点 t_e , 满足 v_x 与 t_e 相连当且仅当 $x \in N_{\mathbf{H}}(e)$, 即

$$E(\mathbf{G}) = \{v_x v_y : v_x, v_y \in K\} \cup \{v_x t_e : x \in N_{\mathbf{H}}(e)\}.$$

根据构造, \mathbf{G} 是 $n+m$ 阶分裂图, 并且 (I, K) 是分裂划分。最后令 $T = I$ 且 $k = n-p$, 最终得到了 R-SFVS(S) 问题的 T -可分实例 $\mathcal{I} = (\mathbf{G}, T, k)$ 。

下证 $\bar{\mathcal{I}}$ 有解当且仅当 \mathcal{I} 有解。首先论证充分性: 若 X 为实例 $\bar{\mathcal{I}}$ 的解, 则

$$S = \{v_x \in V(\mathbf{G}) : v \in V(\mathbf{H}) \setminus X\} \cup \{t_e \in T : e \in \bar{E}\}$$

是 \mathcal{I} 的解, 其中 $\bar{E} \subseteq E(\mathbf{H})$ 表示包含了至少 2 个 X 中顶点的超边子集。考虑图 \mathbf{G} 中的任意 T -三角形 \mathbf{C} , 由于关键点构成独立集, 则 T -三角形 \mathbf{C} 恰好包含一个关键点 $t_e \in I$ 和两个非关键点 $v_x, v_y \in K$ 。根据构造, 必然有 $x, y \in N_{\mathbf{H}}(e)$ 。若顶点 x 和 y 均在 X 中, 那么 $t_e \in S$; 若 x 和 y 至多其中之一在 X 中, 那么顶点 v_x 和 v_y 至少其中之一在 S 中。所以, $\mathbf{G} - S$ 中没有 T -三角形, 进而说明 S 是 T -反馈点集。注意到

$$|S| = |V(\mathbf{H})| - |X| + |\bar{E}| \leq n - p = k.$$

所以, S 是 \mathcal{I} 的解。

接着论证必要性: 若 S 为实例 \mathcal{I} 的解, 则

$$X = \{x \in V(\mathbf{H}) : v_x \in K \setminus S\}$$

是 $\bar{\mathcal{I}}$ 的解。令 $\bar{E} = \{e \in E(\mathbf{H}) : t_e \in S\}$ 。考虑图 \mathbf{H} 中的任意超边 $e \in E(\mathbf{H}) \setminus \bar{E}$, 设 $x, y \in N_{\mathbf{H}}(e)$ 。根据构造, 可知 (t_e, v_x, v_y) 是 T -三角形。由于 S 不包含 t_e , 所以顶点 v_x 和 v_y 至少其中之一在 S 中, 这意味着顶点 x 和 y 至多其中之一在 X 中。所以, $E(\mathbf{H}) \setminus \bar{E}$ 中的任何一条超边只包含 X 中的至多一个顶点, 这意味着 X 的惩罚值至多为 $\text{pen}(X) = |\bar{E}|$ 。可以得出 X 的价值至少为

$$|X| - |\bar{E}| = (|K| - |S \setminus T|) - |S \cap T| = |V(\mathbf{G})| - \tau - |S| = n - k = p.$$

所以, X 是 $\bar{\mathcal{I}}$ 的解。

综上所述, 引理得证。 ■

引理 7.46 SFVS(S) 问题的实例 $\mathcal{I} = (\mathbf{G}, T, k)$ 可以在多项式时间归约到 PC-MIS(H) 问题的实例 $\bar{\mathcal{I}} = (\mathbf{H}, p)$ ，并且满足 $|V(\mathbf{H})| \leq |V(\mathbf{G})| - \tau$, $|E(\mathbf{H})| \leq \tau$ 以及 $p \geq |V(\mathbf{H})| - k$ 。

证明：设 $\mathcal{I} = (\mathbf{G}, T, k)$ 为 SFVS(S) 问题的实例，其中 $\tau = |T|$ 。首先在多项式时间内找到 \mathbf{G} 的一个分裂划分 (I, K) 。根据引理 7.2，可以在多项式时间内判断是否存在一个最小 T -反馈点集，它包含了某些 I 中的某个非关键点，或者不包含 K 中某个关键点。若存在满足这一条件的最小 T -反馈点集，那么 \mathcal{I} 有解，可以平凡地归约到一个 PC-MIS(H) 问题的有解实例上。所以，不妨设 \mathcal{I} 是 T -可分实例，即 $I = T$ 且 $K = V(\mathbf{G}) \setminus T$ 。

现构造 MIS(H) 问题的实例 $\bar{\mathcal{I}} = (\mathbf{H}, p)$ 。首先，对于点集 $V(\mathbf{G})$ 中的每个非关键点 $v \in K$ ，构造一个顶点 x_v ，得到图 \mathbf{H} 的点集

$$V(\mathbf{H}) = \{x_v : v \in K\}.$$

然后，对于每个关键点 $t \in T$ ，构造一个超边 e_t ，使得 $N_{\mathbf{H}}(e_t) = \{x_v : v \in N_{\mathbf{G}}(t)\}$ ，进而得到

$$E(\mathbf{H}) = \{e_t = \{x_v : v \in N_{\mathbf{G}}(t)\} : t \in E(\mathbf{G})\}.$$

根据构造， \mathbf{H} 是一个包含 $n = |V(\mathbf{G})| - |I| = |V(\mathbf{G})| - \tau$ 个顶点和 τ 条超边超图。最后令 $p = n - k$ ，得到了 MIS(H) 问题的实例 $\bar{\mathcal{I}} = (\mathbf{H}, p)$ 。

下证 $\bar{\mathcal{I}}$ 有解当且仅当 \mathcal{I} 有解。首先论证充分性：若 X 为实例 $\bar{\mathcal{I}}$ 的解，则

$$S = \{v \in V(\mathbf{G}) : x_v \in V(\mathbf{H}) \setminus X\}$$

是 \mathcal{I} 的解，其中 $\bar{E} \subseteq E(\mathbf{H})$ 表示包含了至少 2 个 X 中顶点的超边子集。考虑图 \mathbf{G} 中的任意 T -三角形 \mathbf{C} ，由于关键点集是独立集，那么 \mathbf{C} 恰好包含一个关键点 $t \in T \subseteq I$ 和两个非关键点 $v, u \in K$ 。根据构造，有 $x_v, x_u \in N_{\mathbf{H}}(e_t)$ 。若顶点 x_v 和 x_u 均在 X 中，那么 $t \in S$ ；若 x_v 和 x_u 至多其中之一在 X 中，那么顶点 v 和 u 至少其中之一在 S 中。所以， $\mathbf{G} - S$ 中没有 T -三角形，进而说明 S 是 T -反馈点集。注意到

$$|S| = |V(\mathbf{H})| - |X| + |\bar{E}| \leq n - p = k.$$

所以， S 是 \mathcal{I} 的解。

接着论证必要性：设 S 为实例 \mathcal{I} 的解，现说明

$$X = \{x_v \in V(\mathbf{H}) : v \in K \setminus S\}$$

是 $\bar{\mathcal{I}}$ 的解。令 $\bar{E} = \{e_t \in E(\mathbf{H}) : t \in S\}$ 。考虑图 \mathbf{H} 中的任意超边 $e_t \in E(\mathbf{H}) \setminus \bar{E}$ ，设 $x_v, x_u \in N_{\mathbf{H}}(e)$ 。根据构造，可知 (t, v, u) 是 T -三角形。由于 S 不包含 t ，所以顶点 v 和 u 至少其中之一在 S 中，这意味着顶点 x_v 和 x_u 至多其中之一在 X 中。所以， $E(\mathbf{H}) \setminus \bar{E}$ 中的任何一条超边只包含 X 中的至多一个顶点，这意味着 X 的惩罚值至多为 $\text{pen}(X) = |\bar{E}|$ 。可以得出 X 的价值至少为

$$|X| - |\bar{E}| = (|K| - |S \cap T|) - |S \cap T| = |V(\mathbf{G})| - \tau - |S| = n - k = p.$$

所以， X 是 $\bar{\mathcal{I}}$ 的解。 ■

综上所述，引理得证。 ■

目前，尚未已知 PC-MIS(H) 问题的运行时间快于 $\mathcal{O}(2^n)$ 的精确算法。但借助引理 7.45，能够得到该问题以 $n - p$ 为参数的参数算法。特别地，若 SFVS(S) 问题存在快于 $\mathcal{O}^*(2^k)$ 的参数算法，那么便找到了 PC-MIS(H) 问题打破 2^n 瓶颈的非平凡精确算法。

推论 7.47 SFVS(S) 问题可以在 $\mathcal{O}^*(\alpha^k)$ 时间内求解当且仅当 PC-MIS(H) 问题可以在 $\mathcal{O}^*(\alpha^{n-p})$ 时间内求解，其中 k 为 SFVS(S) 问题的解集大小， n 和 p 分别为 PC-MIS(H) 问题实例顶点数和解集价值。

定理 7.42 PC-MIS(H) 问题可以在 $\mathcal{O}^*(1.8192^n)$ 时间内求解。

证明：根据引理 7.47，和本章 7.3.3 节给出的 SFVS(S) 问题的参数算法，PC-MIS(H) 问题可以在 $\mathcal{O}^*(1.8192^{n-p})$ 时间内求解。所以 PC-MIS(H) 问题能在 $\mathcal{O}^*(1.8192^n)$ 时间内求解。 ■

7.6 本章小结

本章深入探讨了分裂图上子集反馈点集问题及其限制版，在参数和精确算法方面实现了显著的改进。

针对 SFVS(S) 问题的参数算法，本章使用了与顶点覆盖问题相关的分解技术以及分支搜索技术。具体地，本章首先引入标记边的概念，并基于此定义了顶点的隐藏操作。然后基于 Dulmage-Mendelsohn 分解技术，提出了一种新颖的约简规则，称为 Dulmage-Mendelsohn 约简规则，用以处理实例的局部可解结构。最后提出了一种非传统的度量函数，采用度量治之分析手段得到了运行时间为 $\mathcal{O}^*(1.8192^k)$ 的参数算法，首次突破 2^k 。

值的强调的是，本章的核心创新点在于引入了 Dulmage-Mendelsohn 分解技术以及将度量治之方法应用于参数算法的时间复杂度分析中。事实上，许多参数算法的设计难以使用度量治之分析方法，而对 SFVS(S) 问题而言，若以传统的解集

大小 k 为度量建立分支算法，容易陷入过于庞大的分类讨论而难以奏效。本文通过引入新颖约简方法，仅使用了少量简单的分支规则，便成功地突破了 2^k 的计算复杂度瓶颈。此外，本章将所得算法应用于超图上带惩罚的独立集问题，得到了首个非平凡的精确算法，成功突破 2^n 。此项研究将为后文研究弦图上子集反馈点集问题奠定基础，也为超图上带惩罚的独立集问题开辟了新的算法研究思路，是一次创新性的突破。

此外，针对 R-SFVS(S) 问题，本章也进行了全面的算法研究。一方面，本章通过建立与顶点覆盖问题的归约关系，证明了 R-SFVS(S) 问题和顶点覆盖问题之间存在保持解集大小不变的多项式时间归约。这一归约说明了 R-DSFVS(S) 问题与顶点覆盖问题存在相同指数时间的参数算法。另一方面，本章为 R-DSFVS(S) 问题设计了比顶点覆盖问题更快的精确算法，这体现了 R-SFVS(S) 问题与 R-DSFVS(T) 问题显著的差异性。事实上，虽然 R-DSFVS(T) 问题和 R-SFVS(S) 问题均为特殊图类上的限制版子集反馈点集问题，但是当关键点数量为常数且至少为 2 时，R-DSFVS(T) 是 NP-难的；而 R-SFVS(S) 问题在 $\tau = \mathcal{O}(\log n)$ 时，依然是多项式时间可解的。这一发现也强调了 R-SFVS(S) 问题与 R-DSFVS(T) 问题在计算复杂性上也有本质上的不同。

作为推论，本文证明了 MIS 问题以团覆盖数为参数是 FPT 的。值得强调的是，对于完美图（Perfect Graph）上 MIS 问题，目前唯一已知的多项式时间算法是基于椭圆法实现的^[274]，但目前尚未已知任何基于组合的多项式时间甚至是 FPT 时间的算法^[275]。由于完美图上团覆盖数与独立集数相等，所以本章提出的算法间接成为了首个解决完美图上 MIS 问题基于组合的 FPT 算法。

第八章 弦图上子集反馈点集问题的参数与精确算法

弦图是能够和树以及平面图相齐名的重要图类，在组合优化，贝叶斯推理，关系式数据库，寄存器分配等领域具有广泛的应用价值。本章聚焦于弦图上子集反馈点集问题的参数与精确算法研究，改进了当前最快算法结果。特别地，对于弦图上的非限制版子集反馈点集问题，本文提出的参数算法首次突破了 2^k 的计算复杂度瓶颈。

8.1 研究背景与主要贡献

弦图是任意长度不小于 4 的环都具有弦的无向图，在组合学、图论和算法等领域中均为基本且核心的图类。现代组合学与图论创始人之一 Claude Berge 在其奠基性著作《Graphes Et Hypergraphes》中，对弦图进行了深入的系统性研究^[276,277]。同一时期，Berge 提出了著名的完美图猜想和强完美图猜想，后续分别被 Lovász^[278-280]与 Chudnovsky 和 Robertson^[281]等证明，分别形成了完美图定理 (Perfect Graph Theorem) 和强完美图定理 (Strongly Perfect Graph Theorem)。Berge 对完美图的子图类——弦图的研究工作不仅深化了大量有价值的图论研究，也开启了并推动了完美图理理论的发展^[260]。

弦图早期曾被称之为刚性环路图 (Rigid Circuit Graph)^[282]，无环图 (Acyclic Graph)^[283]，完美消除图 (Perfect Elimination Graph)^[284]，三角图 (Triangulated Graph)^[260,276,277]，后来统一沿用了 Gavril 根据原始定义的命名方式^[285]。弦图以其简洁的图结构特征和数学性质著称，例如一个图是弦图当且仅当该图的所有极小点割是团^[282]。弦图及其延伸图类，如分裂图，强弦图 (Strongly Chordal Graph)，弱弦图 (Weakly Chordal Graph)，探测弦图 (Probe Chordal Graph)，双弦图 (Doubly Chordal Graph)，对偶弦图 (Dually Chordal Graph) 等^[208]，吸引了广泛的学术关注，这些研究不仅丰富了图论的理论体系，也对解决实际问题提供了重要的理论工具。在应用领域，弦图的理论与技术被广泛运用于超图理论^[286]、组合优化^[210,287-289]、关系式数据库 (Relational Databases)^[290]、完美系统分类学 (Perfect Phylogeny)^[291,292]、概率推理 (Probabilistic Reasoning) 中的 Bayesian 网络^[293-295]、半正定规划 (Semidefinite Optimization)^[296]、编译器中寄存器分配 (Register Allocation)^[297]等领域。

在现代图论和算法领域中，弦图的树状结构不仅推动了树分解技术^[117]和树宽等图参数的研究^[298-300]，也促进了对弦图上许多关键问题算法研究的深

入^[210,285,301–303]。其中，弦图上反馈集相关问题近年来成为研究的焦点，需要删除多少顶点可以将一个树状结构的图转化为真正的树是一个自然而有趣的问题。弦图上的反馈点集问题可在多项式时间内求解^[210]，但是无论是非限制版还是限制版，弦图甚至分裂图上的子集反馈点集问题均属于 NP-难问题^[176,177]。在本章中，弦图上非限制版子集反馈点集问题简记为 SFVS(C)；弦图上限制版子集反馈点集问题简记为 R-SFVS(C)，具体定义如下。

弦图上非限制版子集反馈点集问题 (SFVS(C))

输入: 一个弦图 \mathbf{G} ，一个大小为 τ 的关键点集 $T \subseteq V(\mathbf{G})$ 和一个自然数 k 。

输出: 判断弦图 \mathbf{G} 是否存在一个大小不超过 k 的 T -反馈点集 $S \subseteq V(\mathbf{G})$ 。

弦图上限制版子集反馈点集问题 (R-SFVS(C))

输入: 一个弦图 \mathbf{G} ，一个大小为 τ 的关键点集 $T \subseteq V(\mathbf{G})$ 和一个自然数 k 。

输出: 判断弦图 \mathbf{G} 是否存在一个大小不超过 k 的 T -反馈点集 $S \subseteq V(\mathbf{G}) \setminus T$ 。

研究现状与挑战 SFVS(C) 问题是著名隐式 3-碰撞集问题^[80]。因此，该问题存在点集大小为 $\mathcal{O}(k^2)$ 的核^[240,241]，并且可以在 $\mathcal{O}^*(2.0755^k)$ 时间内求解^[54]。尽管如此，为 SFVS(C) 问题突破 3-碰撞集的算法瓶颈是一项长期的挑战。直到 2019 年，Philip 等人^[80]首次提出了改进的参数算法，将求解时间降低至 $\mathcal{O}^*(2^k)$ 。然而，进一步改进 SFVS(C) 问题的参数算法，即使将输入限制为分裂图上，依旧是困难的。一方面，弦图上传统的技术是使用其典型的树状结构，采用动态规划算法完成求解，面对非限制版子集反馈点集问题时，动态规划的状态数量难以控制。另一方面，Philip 等人提出的分支搜索算法极其复杂，其中两个分支规则包含多达 7 个子分支，仅靠深入的结构分析和情况讨论难以取得实质性突破。此外， 2^k 是许多算法技术的天然障碍，如简单穷举搜索算法，动态规划算法，容斥原理算法，子集卷积算法，迭代压缩算法等众多算法通常都无法打破 2^k 的瓶颈。因此，寻找快于 $\mathcal{O}^*(2^k)$ 的参数算法面临多重障碍。

在精确算法方面，Golovach 等人^[262,263]针对弦图上的子集反馈点集问题，于 2012 年给出了运行时间为 $\mathcal{O}(1.6708^n)$ 的精确算法。此后，Chitnis 等人^[58]，Fomin 等人^[97]分别将运行时间改进到了 $\mathcal{O}(1.6181^n)$ 和 $\mathcal{O}(1.5182^n)$ 。目前最快的精确算法^[80]时间复杂度为 $1.5^n 2^{o(n)}$ 。由于这些算法均为分支搜索算法，因此它们能够同时适用于限制版和非限制版的子集反馈点集问题。

反馈点集与子集反馈点集问题在弦图及其子图上的计算复杂性也得到了充分的研究。Yannakakis 和 Gavril 证明了弦图上的反馈点集问题是多项式时间可解的。而 Fomin 等人^[176,177]证明了，即使是在分裂图上，非限制版和限制版子集反馈点

集问题均是 NP-难的。此外, Papadopoulos 和 Tzimas^[212,213]证明了区间图上的反馈点集问题是多项式时间可解的。

主要贡献 本章系统性地开展了 SFVS(C) 问题和 R-SFVS(C) 问题的算法研究, 提出了当前最快的参数与精确算法, 并给出了 R-SFVS(C) 问题的核心化算法。本章的算法研究建立在第七章针对分裂图上子集反馈点集问题的研究基础之上。

本章的核心贡献在于通过树分解技术, 建立了 SFVS(C) 问题和 SFVS(S) 问题之间的关系, 证明了对于任意实数 $\alpha > 1$, 若 SFVS(S) 能在 $\mathcal{O}^*(\alpha^k)$ 时间内求解, 那么 SFVS(C) 能在 $\mathcal{O}^*(\alpha^k + 1.6181^k)$ 时间内求解。本章所采用的技术路线摒弃了传统的动态规划技术路线, 而是在树分解中寻找具有关键性质的极小点割, 通过多次调用 SFVS(S) 问题的参数算法, 能够直接将一部分子结构完全删除, 逐步缩小问题实例, 直至将弦图退化成分裂图。最终证明了 SFVS(C) 问题可以在 $\mathcal{O}^*(1.1820^k)$ 时间内求解, 首次突破 2^k 。这一算法成果不仅避免了动态规划等算法难以打破 2^k 的事实, 而且保证了能够在多项式空间内完成计算。

此外, 基于类似的树分解技术, 本章揭示了 R-SFVS(C) 问题和 R-SFVS(S) 问题之间的关系, 得出了相比非限制版本更强的命题: 无论是以解集大小 k 或关键集大小 τ 为参数的参数算法, 还是以点集大小 n 为参数的精确算法, R-SFVS(S) 问题和 R-SFVS(C) 问题均能在相同指数时间内求解。基于此关系, 直接得到了 R-SFVS(C) 问题的快速参数和精确算法, 改进了当前所有的算法结果。注意到, 直接将 R-SFVS(C) 问题归约到 R-SFVS(S) 问题是困难的, 依然需要结合精细的分支搜索技术与分治技术, 通过多次调用 R-SFVS(S) 问题的算法才能最终建立联系。主要结果见表8-1。

表 8-1 弦图上子集反馈点集问题的主要算法结果

	SFVS(C) 问题	R-SFVS(C) 问题
核心化 (顶点数)	$\mathcal{O}(k^2)$ (文献 [240,241])	$k^2 + 2k$ (定理8.30)
参数算法 (以 k 为参数)	$\mathcal{O}^*(1.8192^k)$ (定理8.2)	$\mathcal{O}^*(1.2529^k)$ (定理8.23)
参数算法 (以 τ 为参数)	$\mathcal{O}(1.8192^\tau)$ (推论8.21)	$\mathcal{O}^*(2^\tau)$ (定理8.36)
精确算法	$\mathcal{O}(1.3788^n)$ (定理8.3)	$\mathcal{O}(1.1520^n)$ (定理8.24)

8.2 弦图的基本结构与性质

如果一个图中的任何环都存在弦, 则称这个图为弦图。回顾弦的定义, 环上不连续的两个顶点间相连的边称为弦。弦图是一类禁止子图图类, 一个图是弦图当且仅当这个图没有任何长度至少为 4 的导出环。在一些文献中, 将长度至少为 4 的导出环称为洞 (Hole), 因此一个图是弦图当且仅当没有洞 $\mathcal{H} = \{\mathbf{C}_\ell : \ell \geq 4\}$ 。

弦图的导出子图仍然为弦图，并且弦图的每个极小点割必然为一个团^[282]。弦图是树的一些子树构成的相交图，所以将弦图的一条边进行合并操作得到的图也是弦图。此外，分裂图和区间图均为弦图的子图，其中分裂图是星图的一些子树构成的相交图；区间图是路的一些子路构成的相交图。

考虑连通弦图 \mathbf{G} ，令 $\mathcal{Q}_{\mathbf{G}}$ 表示 \mathbf{G} 中所有极大团构成的集族。如果某顶点只属于弦图 \mathbf{G} 中唯一一个极大团，那么称这个顶点为单纯点 (**Simplicial Vertex**)；包含这个单纯点的极大团称为单纯团 (**Simplicial Clique**)。弦图至少存在一个单纯团。弦图 \mathbf{G} 的团树 (Clique Tree)，记作 $\mathcal{T}_{\mathbf{G}}$ ，是图 \mathbf{G} 的一个树分解，并且其中的每个节点一一对应一个极大团，即 $V(\mathcal{T}_{\mathbf{G}}) = \mathcal{Q}_{\mathbf{G}}$ 。由树分解的结构可知，弦图的团树满足以下性质^[285, 304, 305]：

- 团树的每个叶子节点必然为单纯团；
- 对于团树中的任何一条边 $Q_1Q_2 \in \mathcal{T}_{\mathbf{G}}$ ，交集 $Q_1 \cap Q_2$ 是极小点割，且将 $Q_1 \cup Q_2$ 完整地分割在 $\mathbf{G} - (Q_1 \cap Q_2)$ 中的不同连通块里。

这一性质在本章给出的参数算法中起着重要作用。弦图上极大团的数量不会超过总点数^[284]，并且这些极大团可以在线性时间内枚举得到^[285]。基于线图上极大团的枚举算法，可以在线性时间内判断一个图是否为弦图^[306]。

引理8.1表明 SFVS(C) 问题是隐式 3-碰撞集问题。此结果在文献 [80] 中给出。

引理 8.1 设 \mathbf{G} 为弦图， $T \subseteq V(\mathbf{G})$ 为 \mathbf{G} 的关键点集。 \mathbf{G} 是 T -无环的当且仅当 \mathbf{G} 不包含任何 T -三角形。

基于引理8.1，在设计算法时，只需将目标确定为删除顶点以碰撞给定弦图中所有 T -三角形即可。

8.3 弦图上非限制版子集反馈点集问题

本节将提出弦图上非限制版子集反馈点集问题的参数算法，首次打破了 2^k 的瓶颈，改进了当前最快的 $\mathcal{O}^*(2^k)$ 时间的参数算法。以参数算法为基础，本节将进一步给出 SFVS(C) 问题的精确算法，改进了当前最快的 $1.5^n 2^{o(n)}$ 时间的精确算法。主要结论如下。

定理 8.2 SFVS(C) 问题可以在 $\mathcal{O}^*(1.8192^k)$ 时间内求解。

定理 8.3 SFVS(C) 问题可以在 $\mathcal{O}(1.3788^n)$ 时间内求解。

本章给出的 SFVS(C) 问题的参数算法包括三个部分。

第一部分引入了基于点割的替换约简规则：若存在小的点割，可以将该点割所对应的某一连通块替换为更小的连通块。这一操作能够在不考虑连通块具体结构的情况下完成约简操作，极大地简化了算法的分析工作。

第二部分则是通过常规分支搜索技术简化弦图的结构，以便 SFVS(S) 问题的参数算法能够使用。具体地，若一个实例关于第二部分所提出的所有约简规则和分支规则都不可执行，则称该实例是约简实例。在约简实例中，如果所有关键点都是单纯点，便能够调用 SFVS(S) 问题的参数算法完成求解。

第三部分则是考虑存在内部关键点（即不是单纯的关键点）的情形。在第三部分中，将首先找到一个包含内部关键点的极小点割，然后多次调用 SFVS(S) 问题的参数算法求解若干子实例。接着，基于这些子实例的解集大小，设计出一个基于分治思想的参数算法。

最后，利用弦图上反馈点集问题的多项式时间算法，通过均摊思想为 SFVS(S) 问题设计了更加快速的精确算法，得到了运行时间为 $\mathcal{O}(1.3488^n)$ 的参数算法。

8.3.1 基于点割的替换约简规则

本节将提出一个约简规则用于处理输入弦图中存在大小不超过 2 的点割的实例。基本思想如下：考虑点割 Q ，根据弦图的性质， Q 必为团。对于 $\mathbf{G} - Q$ 中的连通块 C ，可以将其替换为至多两个仅包含 1 个关键点的连通块，进而实现对实例的简化工作。

为了保证在给定任何连通块的条件下均能实现替换，这里首先引入 SFVS(C) 问题的一个广义版本，称为弦图上广义子集反馈点集问题（Generalized Subset Feedback Vertex Set Problems in Chordal Graphs），简记为 Generalized-SFVS(C)。具体定义如下。

弦图上广义子集反馈点集问题（Generalized-SFVS(C)）

输入：一个弦图 \mathbf{G} ，一个关键点集 $T \subseteq V(\mathbf{G})$ ，一条标记边集 $M \subseteq E(\mathbf{G})$ ，一个自然数 $k \in \mathbb{N}$ 。

输出：判断是否存在一个大小不超过 k 的 T -反馈点集 S ，使得 M 中的任何一条边都至少存在一个端点在 S 中。

在本节中，仍然使用符号 $N_M(v) := \{u \in N(v) : uv \in M\}$ 表示所有与 v 有标记边相连的顶点集。对于任意点击 $X \subseteq \mathbf{G}$ ，将实例 $(\mathbf{G}[X], T \cap X, M \cap E(\mathbf{G}[X]), k)$ 称为由 X 导出的子实例。

设 $\mathcal{I} = (\mathbf{G}, T, M, k)$ 为 Generalized-SFVS(C) 问题的实例。设 $Q = \{v_1, v_2\}$ 是大小为 2 的点割，并且令 C 是 $\mathbf{G} - Q$ 中的连通块。考虑四个点子集； $C_0 = C$ ， $C_1 = C \cup \{v_1\}$ ， $C_2 = C \cup \{v_2\}$ ， $C_3 = C \cup Q$ 。为方便记忆，子集 $C_j = C \cup Q'$ 满足 Q' 中的顶点下标和恰好为 j 。

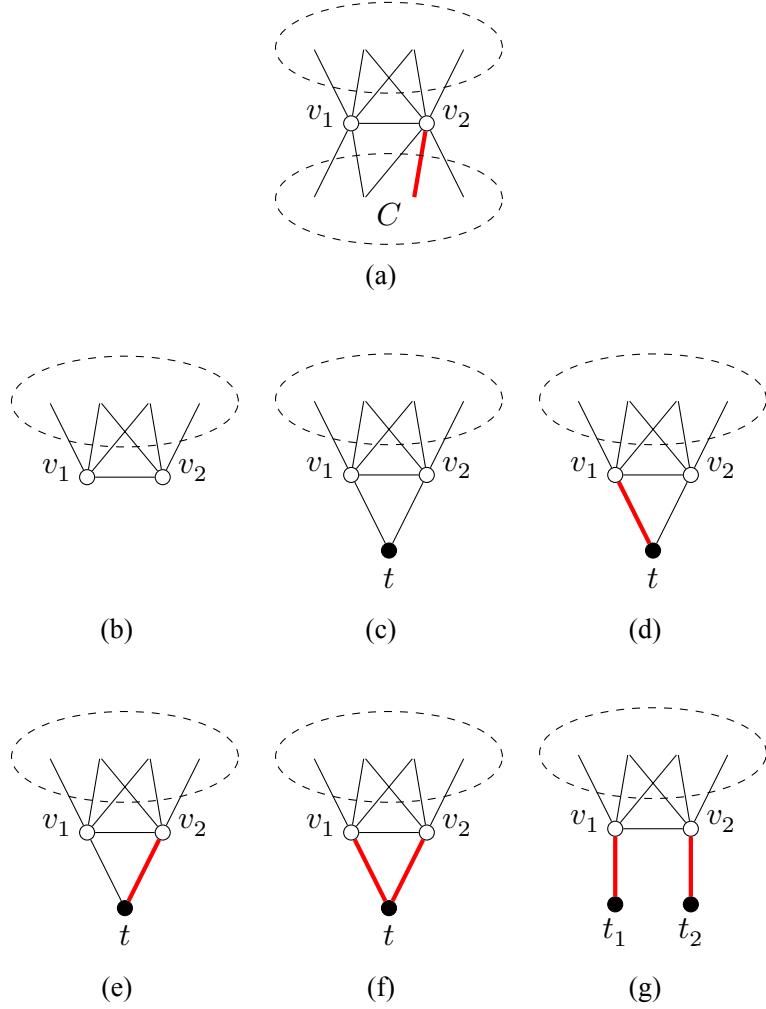


图 8-1 实例 \mathcal{I} 替换为实例 $\bar{\mathcal{I}}_i$ ($i \in [6]$) 的示意图。(a) 为输入实例 $\mathcal{I} = (\mathbf{G}, T, M, k)$ 。(b)-(g) 分别为 6 种替换后得到的实例 $\bar{\mathcal{I}}_1, \bar{\mathcal{I}}_2, \dots, \bar{\mathcal{I}}_6$ 。中黑色顶点表示关键点, 白色顶点表示非关键点, 虚线椭圆表示子图 $\mathbf{G} - Q$ 的连通块。点割 $Q = \{v_1, v_2\}$ 是弦图 \mathbf{G} 中大小为 2 的点割。

用 \mathcal{I}_j 表示由 C_j 导出的子实例 ($j = \{0, 1, 2, 3\}$)。显然, 若某个子实例无解, 那么 \mathcal{I} 也无解。因此设 S_j 为 \mathcal{I}_j 的最优解。注意到 $|C_0| + 2 = |C_1| + 1 = |C_2| + 1 = |C_3|$, 那么有

$$\mathbf{s} = [|S_1|, |S_2|, |S_3|]^\top = |S_0| \cdot \mathbf{1} + \mathbf{x}$$

其中, $\mathbf{1}$ 表示全一向量, 且 $\mathbf{x} = [x_1, x_2, x_3]^\top$ 必然恰好属于以下六个向量之一:

$$\mathbf{x}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{x}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{x}_3 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \quad \mathbf{x}_4 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{x}_5 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad \mathbf{x}_6 = \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}.$$

因此, 需要分别找到六种替换规则替换六种情况之一的连通块 C 进行替换, 最终得到实例 $\bar{\mathcal{I}}_i$ ($i \in [6]$) (参见图8-1)。

实例 $\bar{\mathcal{I}}_1$: 将 C 从实例 \mathcal{I} 中直接删除;

实例 $\bar{\mathcal{I}}_2$: 将 C 替换为一个关键点 t 并连接 v_1 和 v_2 ;

实例 $\bar{\mathcal{I}}_3$: 将 C 替换为一个关键点 t 并连接 v_1 和 v_2 , 标记 v_1t ;

实例 $\bar{\mathcal{I}}_4$: 将 C 替换为一个关键点 t 并连接 v_1 和 v_2 , 标记 v_2t ;

实例 $\bar{\mathcal{I}}_5$: 将 C 替换为一个关键点 t 并连接 v_1 和 v_2 , 标记 v_1t, v_2t ;

实例 $\bar{\mathcal{I}}_6$: 将 C 替换为两个关键点 t_1 和 t_2 分别连接 v_1 和 v_2 , 标记 v_1t_1, v_2t_2 。

现给出引理8.4刻画替换规则的正确性。

引理 8.4 设实例 $\bar{\mathcal{I}}_i$ 的最优解为 \bar{S}_i ($i \in [6]$), 实例 \mathcal{I} 的最优解为 S 。若向量 $\mathbf{x} = \mathbf{x}_i$ ($i \in [6]$), 那么有 $|S| = |S_0| + |\bar{S}_i|$ 。

证明: 由于 $\mathbf{x} = \mathbf{x}_i$, 那么有大小关系

$$|S_0| = |S_1| - x_1 = |S_2| - x_2 = |S_3| - x_3,$$

其中 $\mathbf{x} = [x_1, x_2, x_3]^\top$ 。

在实例 $\bar{\mathcal{I}}_i$ 中, 用 C' 表示替换 C 的点集。具体地, 当 $i = 1$ 时, $C' = \emptyset$; 当 $i = 2, 3, 4, 5$ 时, $C' = \{t\}$; 当 $i = 6$ 时, $C' = \{t_1, t_2\}$ 。用 C' 替换了 C 后, C_j 相应地变为了 $C'_j = (C_j \setminus C) \cup C'$ ($j \in \{0, 1, 2, 3\}$)。考虑子实例 $\mathcal{I}'_j = (\mathbf{G}[C'_j], T \cap C'_j, M \cap E(\mathbf{G}[C'_j]), k)$ ($j = \{0, 1, 2, 3\}$)。令 S'_i 表示实例 \mathcal{I}'_i 的最优解。容易验证

$$0 = |S'_0| = |S'_1| - x_1 = |S'_2| - x_2 = |S'_3| - x_3.$$

因此, 对于任意的 $j = \{0, 1, 2, 3\}$, 都有 $|S_j| - |S'_j| = |S_0|$ 。

设 S 是实例 \mathcal{I} 的最优解, 令 $C_{j^*} = C \cup (Q \setminus S)$ 。那么 $S \cap C_{j^*}$ 是子实例 \mathcal{I}_{j^*} 的解。所以有 $|S \cap C_{j^*}| \geq |S_{j^*}|$, 进而有 $|S \setminus C_{j^*}| \leq |S|$ 。注意到, 因为 Q 是一个点割, 所以 $(S \setminus C_{j^*}) \cup S_{j^*}$ 也是实例 \mathcal{I} 的解。又因为

$$|(S \setminus C_{j^*}) \cup S_{j^*}| = |S \setminus C_{j^*}| + S_{j^*}| \leq |S| - |S_{j^*}| + |S_{j^*}| = |S|.$$

所以, $(S \setminus C_{j^*}) \cup S_{j^*}$ 也是 \mathcal{I} 的最优解, 因此有

$$|(S \setminus C_{j^*}) \cup S_{j^*}| = |S|.$$

类似地, 可以得出 $(S \setminus C_{j^*}) \cup S'_{j^*}$ 也是替换后的实例 $\bar{\mathcal{I}}_{j^*}$ 的最优解, 并且

$$|(S \setminus C_{j^*}) \cup \bar{S}_{j^*}| = |S| - |S_{j^*}| + |S'_{j^*}|.$$

结合上述关系式, 可得

$$|S| - |(S \setminus C_{j^*}) \cup S'_{j^*}| = |S| - (|S| - |S_{j^*}| + |S'_{j^*}|) = |S_{j^*}| - |S'_{j^*}| = |S_0|,$$

引理得证。 ■

基于引理8.4, 便可以得到以下约简规则。

约简规则 8.3.1 (小点割约简规则) 设 Q 是大小为 2 的点割, C 是 $\mathbf{G} - Q$ 中的连通块且 $|C| \geq 2$ 。若由 C 导出的子实例的最优解不超过 5, 按照引理8.4的条件, 将 C 用一个或两个关键点进行替换, 得到更小的实例 $\bar{\mathcal{I}}_i$ ($i \in [6]$)。

注意到, 为了使得小点割约简规则8.3.1在多项式时间内执行完成, 需要保证实例 \mathcal{I}_0 的解集大小为常数。此外, 若存在大小为 1 的点割 $Q = \{v\}$, 那么对于顶点 v 的任意邻居 u , $Q \cup \{u\}$ 都构成大小为 2 的点割, 引理8.4依然适用。因此, 当小点割约简规则8.3.1不可执行时, 对于任意大小不超过 2 的点割 Q 和任意 $\mathbf{G} - Q$ 上的连通块 C , 要么 C 导出的子实例的解超过 5, 要么 $|C| = 1$ 。

事实上, 小点割约简规则8.3.1执行后所得到的子实例必然存在一度或二度单纯关键点。这些结构能通过8.3.2节的约简规则进一步化简处理。

8.3.2 基本的约简与分支规则

考虑 Generalized-SFVS(C) 问题的实例 $\mathcal{I} = (\mathbf{G}, T, M, k)$ 。由于弦图的结构性质相比分裂图更弱, 无法采用 Generalized-SFVS(S) 问题参数算法所使用的技术进行求解。但是, 如果实例 \mathcal{I} 中的关键点都是单纯点, 则是可以在消耗相对快速的指数时间内归约至 Generalized-SFVS(S) 问题再进一步求解。

本小节将给出基本的七条约简和分支规则, 将实例 \mathcal{I} 化简, 并证明当所有关键点都是单纯点时, 可以通过调用 Generalized-SFVS(S) 问题的参数算法进行求解。这些约简规则和分支规则的技术细节相对简单, 其中一部分约简规则源自于对 Generalized-SFVS(S) 问题参数算法的推广。本小节将使用传统度量 k 进行分析, 由于每条规则都仅包含删点和删边操作, 因此每条规则均至多执行多项式次, 并且每次执行时 k 的值均不增。本小节假设任意一条规则只有在前述的规则均不可执行时才会开始执行。

约简规则 8.3.2 (无效顶点约简规则) 若顶点 v 既不属于任何 T -三角形中, 也不连接任何标记边, 则直接删除顶点 v 。

约简规则 8.3.3 (桥约简规则) 若非标记边 e 是桥, 则直接删除边 e 。

无效顶点约简规则8.3.2和桥约简规则8.3.3的正确性是平凡的, 因为无效顶点和桥都不出现在任何 T -三角形里。注意到, 不能直接将所有不在 T -三角形里的边都删除, 因为这一操作不能保证得到的图依然是弦图, 而删除桥则可以保证得到

的图依然是弦图。若无效顶点约简规则8.3.2和桥约简规则8.3.3均不可执行，那么图中不存在孤立点，并且所有非关键点必然连接着关键点，即 $N[T] = V(\mathbf{G})$ 。

分支规则 8.3.4 (相交标记边分支规则) 若顶点 v 与至少两条标记边相连接，那么分别计算以下两个子分支。若两个子分支均“无解”，则返回“无解”；否则返回“有解”。

- 删去 v ，并使 k 减 1；
- 对于每个与 v 之间存在标记边的顶点 u ，删去 u ，并使 k 减 1。

引理 8.5 相交标记边分支规则8.3.4是正确的，并且其分支向量不劣于 $(1, 2)$ 。

证明：由于每条标记边的至少一个端点在解中，故如果 v 不在解中，那么所有与 v 通过标记边相连的顶点都必然在解中。因此相交标记边分支规则是正确的，并且分支向量是 $(1, |N_M(v)|)$ 。根据规则执行条件， $N_M(v)$ 至少包含两个顶点，因此分支向量不劣于 $(1, 2)$ ，引理得证。 ■

相交标记边分支规则8.3.4不可执行时，标记边构成匹配，此时每个顶点至多为一条标记边的端点。

接下来考虑度数不超过 2 的顶点。

约简规则 8.3.5 (M -小度顶点约简规则) 若存在 1 度或 2 度顶点 v ，并且 v 与标记边 $vu \in M$ 相连接，则删除 u ，将其放入解集，并使 k 减少 1。

注意到，由于无效顶点约简规则8.3.2此时不可执行，一度点 v 所连接的唯一边必然是标记边。此外，由于相交标记边分支规则8.3.5此时也不可执行，顶点 v 至多与一条标记边相连接。

引理 8.6 M -小度顶点约简规则8.3.5是正确的。

证明：由于相交标记边分支规则此时不可执行， vu 是与顶点 v 唯一相连的标记边。又由于标记边的至少一个端点在解中，所以 v 和 u 至少其中之一在解中。此外，所有包含 v 的 T -三角形和标记边（即边 vu ）也都包含了 u 。所以，如果存在解 S 包含了顶点 v ，那么将其替换为 u 得到的点集 $S' = (S \setminus \{v\}) \cup \{u\}$ 也是解。综上所述，如果实例有解，一定存在解包含 u ，因此 M -小度顶点约简规则是正确的，引理得证。 ■

约简规则 8.3.6 (二度点约简规则) 若存在 2 度点 v ，它不在任何标记边中，将连接它两个邻居的边进行标记，然后删除顶点 v 。

事实上，由于 M -小度顶点约简规则8.3.5此时不可执行，因此任意 2 度点必然不与标记边连接。此外，由于桥约简规则8.3.3此时不可执行，它的两个邻居必然相邻。

引理 8.7 二度点约简规则8.3.6是正确的。

证明: 设 2 度点 v 的邻居为 $N_G(v) = \{v_1, v_2\}$, 那么 (v, v_1, v_2) 必然构成 T -三角形, 因为无效顶点约简规则此时不可执行。所以, 若存在解包含顶点 v , 那么可以将 v 替换为 v_1 或 v_2 , 从而得到另外一个解。但无论是何种情况, 将 v_1v_2 进行标记均正确。所以二度点约简规则是正确的, 引理得证。 ■

考虑某个单纯点 v , 根据定义, 顶点 v 的任何邻居都满足支配关系 $N[v] \subseteq N[u]$ 。接下来将引入一条约简规则和一条分支规则处理包含单纯点的局部结构。这两条规则均由点对支配关系所导出的, 属于高级分支搜索算法中的常用手段。

约简规则 8.3.7 (支配点对约简规则) 设 $vu \in M$ 为一条标记边, 并且 v 或者 u 是关键点。若 $N_G[v] \subseteq N_G[u]$ 成立, 则删除顶点 u , 将其放入解中, 并使 k 减少 1。

引理 8.8 支配点对约简规则 8.3.7 是正确的。

证明: 由于 vu 是一条标记边, 因此顶点 v 和 u 至少其中之一在解集中。下证若解集 S 包含 v 但不包含 u , 那么集合 $S' = (S \setminus \{v\}) \cup \{u\}$ 也是解, 且包含 t 。

采用反证法, 假设 S' 不是解。注意到 S' 和 S 的大小相同。 (v, a, b) 是一个在导出子图 $\mathbf{G} - S'$ 中的 T -三角形。由 $N_G[v] \subseteq N_G[u]$ 可知 (u, a, b) 是三角形。当 v 是非关键点时, a 和 b 至少其中之一是关键点; 当 v 是关键点时, 由引理条件可知 u 也是关键点。因此, 无论 v 是否为关键点, (u, a, b) 都是 T -三角形。这意味着在导出子图 $\mathbf{G} - S$ 中存在 T -三角形 (u, a, b) , 这与 S 是解矛盾。因此 S' 是解, 引理得证。 ■

分支规则 8.3.8 (单纯点分支规则) 设 Q 为大小至少为 4 的单纯团。若单纯点 $v \in Q$ 与关键点 t 相邻, 则分别计算以下两个子分支。若两个子分支均“无解”, 则返回“无解”; 否则返回“有解”。

- 删除关键点 t , 并使 k 减少 1;
- 删除点集 $Q \setminus \{t, v\}$, 并使 k 减少 $|Q| - 2 \geq 2$ 。

引理 8.9 单纯点分支规则 8.3.8 是正确的, 并且其分支向量不劣于 $(1, 2)$ 。

证明: 注意到, 支配点对约简规则此时不可执行, 因此 vt 不是一条标记边。只需证明若存在解 S 不包含关键点 t , 则一定存在另外一个解包含顶点 v 。

注意到, 由于 t 不在解集 S 中, 那么 S 至少包含 Q 中的 $|Q| - 2$ 个顶点, 即 Q 中除了 t 以外, 至多还有一个顶点不在 S 中。否则 $Q \setminus S$ 中存在两个顶点和 t 构成 T -三角形。将 $S \cap Q$ 替换为 $Q \setminus \{t, v\}$ 可以得到集合 $S' = (S \setminus Q) \cup (Q \setminus \{t, v\})$ 。现证明 S' 是解。一方面, 有大小关系 $|S'| \leq |S|$; 另一方面, 在子图 $\mathbf{G} - S'$ 中, v 仅与关键点 t 相连, 这说明了 v 不在任何 T -三角形或标记边中。所以, S' 必是解。

最后分析分支向量, 根据规则执行条件, Q 至少包含四个顶点, 所以在第二个分支中, k 至少减少了 $|Q| - 2 \geq 2$ 。这说明了单纯点分支规则的分支向量不劣

于 $(1, 2)$, 引理得证。 ■

结合本节提出的条约简或分支规则, 与8.3.1节所提出的小点割约简规则8.3.1, 现引入 Generalized-SFVS(C) 问题的约简实例的概念。

定义 8.10 (约简实例) 设 $\mathcal{I} = (\mathbf{G}, T, M, k)$ 是 Generalized-SFVS(C) 问题的实例。若上述 8 种约简规则和分支规则在 \mathcal{I} 上均不可执行, 则称 \mathcal{I} 为约简实例。

引理 8.11 若实例 $\mathcal{I} = (\mathbf{G}, T, M, k)$ 是约简实例, 则以下性质成立:

- (a) 图 \mathbf{G} 中无孤立点, 且 M 为一个匹配;
- (b) 每个单纯团至少包含 4 个顶点;
- (c) 每个单纯团仅包含一个单纯点, 并且也是该团中唯一的关键点;
- (d) 对于任意大小不超过 2 的点割 Q , 以及任意 $\mathbf{G} - Q$ 中的连通块 C , 由 C 导出的子实例的最优解大小至少为 5。

证明: 孤立点不在任何 T -三角形或标记边中, 所以如果存在孤立点, 无效顶点约简规则可执行。此外, 如果存在一个顶点连接了至少两个标记边, 那么相交标记边分支规则可执行。所以性质(a)是正确的。

现证明性质(b)的正确性。考虑单纯团 Q , 设 v 是 Q 中的单纯点。因为 v 不是孤立点, 所以 $\deg(v) \geq 1$ 。若 $\deg(v) = 1$ 且 v 不连接标记边, 无效顶点约简可执行, 矛盾。若 $\deg(v) = 1$ 且 v 连接了标记边, M -小度顶点约简规则可执行, 矛盾。若 $\deg(v) = 2$ 且 v 不连接标记边, 二度点约简规则可执行, 矛盾。若 $\deg(v) = 2$ 且 v 恰好连接了一条标记边, M -小度顶点约简规则可执行, 矛盾。若 $\deg(v) = 2$ 且 v 连接了两条标记边, 相交标记边分支规则可执行, 矛盾。因此, 可得 $\deg(v) \geq 3$, 这意味着 $|Q| \geq 4$ 。所以, 性质(b)是正确的。

接下来考虑性质(c)的正确性。设 u 是单纯团 Q 中的单纯点。由于相交标记边分支规则不可执行, u 至多连接一条标记边, 若 u 是非关键点且连接了一条标记边, 支配点对约简规则可执行, 矛盾。若 u 是非关键点但不连接标记边, 由于无效顶点约简不可执行, $Q \cap T$ 非空。因此设 t 是 Q 中的关键点, 此时单纯点分支规则可执行, 亦矛盾。所以, u 必然为关键点。此外, 若存在另一个单纯点 $u' \in Q$, 单纯点分支规则同样可执行。这说明了 u 是 Q 中唯一一个单纯点, 并且 u 必然是关键点。此外, 如果与 u 与另外一个关键点相邻, 单纯点分支可执行, 矛盾。所以性质(c)正确。

最后, 我们证明性质(d)的正确性。令 Q 为大小至多为 2 的点割。采用反证法, 设 C 是子图 $\mathbf{G} - Q$ 中的连通块且由 C 导出的子实例 $(\mathbf{G}[C], T \cap C, M \cap E(\mathbf{G}[C]), k)$ 的最优解大小少于 5。根据小点割约简规则的执行规则, C 必为单点集。此时, C 中的顶点的度数至多为 2, 这意味着 $C \cup Q$ 是大小仅为 3 的单纯团, 假设不成立。

故性质(d)正确。

综上所述, 引理得证。 ■

接下来介绍引理8.12, 这是本小节的核心结论。引理8.12表明, 如果约简实例中的关键点均是单纯的, 则可在多项式时间内归约到 SFVS(S) 问题进行求解。

引理 8.12 设 $\mathcal{I} = (\mathbf{G}, T, M, k)$ 为约简实例。若 \mathcal{I} 中的所有关键点均为单纯点, 则可以在多项式时间内将 \mathcal{I} 归约到 Generalized-SFVS(S) 问题的一个实例, 并且 k 不会增加。

证明: 首先构造 Generalized-SFVS(S) 问题的实例 $\mathcal{I}' = (\mathbf{G}', T', M', k')$ 。直观地讲, \mathcal{I}' 是通过将 \mathcal{I} 中的非关键点两两相连形成团而构建的。具体地, 对于 \mathbf{G} 中的每个顶点 $v \in V(\mathbf{G})$, 构造一个顶点 v' , 得到点集

$$V(\mathbf{G}') = \{v' : v \in V(\mathbf{G})\},$$

并且令关键点

$$T' = \{t' : t \in T\}.$$

关键点 t' 和非关键点 v' 相连当且仅当 t 和 v 相连, 然后令所有 $V(\mathbf{G}')$ 中非关键点形成团, 得到了边集

$$E(\mathbf{G}') = \{v'u' : vu \in E(\mathbf{G})\} \cup \{v't' : vt \in E(\mathbf{G})\}.$$

最后令 $k' = k$, 最终得到了实例 $\mathcal{I}' = (\mathbf{G}', T', M', k')$ 。

注意到, \mathcal{I} 是约简的, 所以关键点构成独立集, 并且每个关键点都是单纯点。所以, \mathbf{G} 中不相邻非关键点的公共邻居中无关键点。因此, 在非关键点之间连边不会增加经过关键点的三角形, 即 T' -三角形和 T -三角形是一一对应的。此外, 根据构造有 $k' = k$, 所以 k 的值不增。

引理得证。 ■

最后, 结合8.3.1小节提出的小点割约简规则8.3.1, 以及本小节给出的七条约简和分支规则, 本小节最后给出以下算法, 该算法将 Generalized-SFVS(C) 问题的任何一个实例转化为约简实例或直接判断实例是否有解。

具体地, 若一个关键点不是单纯点, 则称其是内部关键点 (**Inner Terminal**)。在以下算法中, 每个步骤仅由删点、删边和标记边三种基本操作组成。所以, 如果实例 \mathcal{I} 没有内部关键点, 那么通过该算法所得到的约简实例也没有内部关键点。那么根据引理8.12, 则可以归约至 Generalized-SFVS(S) 问题, 进而完成求解。故本节部分地解决了 Generalized-SFVS(C) 问题, 得到了引理8.13。

算法 8-1 Generalized-SFVS(C) 问题的约简实例转化算法

输入: Generalized-SFVS(C) 问题的实例 $\mathcal{I} = (\mathbf{G}, T, M, k)$ 。

输出: 输出一个约简实例或直接判断 \mathcal{I} 是否有解。

- 1 执行无效顶点约简规则8.3.2。// 此步骤后，非关键点必是某个关键点的邻居且没有孤立点。
- 2 执行桥约简规则8.3.3。// 此步骤后，任何极大团大小至少为 2，且极大团为 2 团中的边必然是一条标记边。
- 3 执行相交标记边分支规则8.3.4。// 此步骤后，标记边构成一个匹配。
- 4 执行 M -小度顶点约简规则8.3.5。// 此步骤后，所有顶点的度至少为 2，且 2 度顶点必然不连接标记边。
- 5 执行二度点约简规则8.3.6。// 此步骤后，所有顶点的度至少为 3，因此任何极大团大小至少为 4。
- 6 执行支配点对约简规则8.3.7。// 此步骤后，所有单纯非关键点均不与标记边相连接。
- 7 执行单纯点分支规则8.3.8。// 此步骤后，每个单纯团包含唯一一个单纯点，并且也是唯一的关键点。
- 8 执行小点割约简规则8.3.1。// 此步骤后，得到约简实例。
- 9 **if** 所有关键点均为单纯点 **then**
- 10 | 依据引理8.12，可在多项式时间内归约到 Generalized-SFVS(S) 问题，并调用运行时间为 $\mathcal{O}^*(1.8192^k)$ 的参数算法直接进行求解。
- 11 **end**
- 12 返回当前实例。// 此时的实例是一个约简实例且其中一定存在内部关键点。

引理 8.13 算法8-1是正确的，对于任意实数 $\alpha > 1$ ，若 Generalized-SFVS(S) 问题能在 $\mathcal{O}^*(\alpha^k)$ 时间内求解，那么算法8-1能在 $\mathcal{O}^*(1.6181^k)$ 时间将 Generalized-SFVS(C) 问题的任意实例转化为一个约简实例。进一步，若 Generalized-SFVS(S) 问题实例没有内部关键点，则算法8-1可以在 $\mathcal{O}^*(\alpha^k + 1.6181^k)$ 时间内求解。

证明: 算法8-1中的每条规则都是正确的，所以算法8-1是正确的。

因为每条分支规则的分支向量不劣于 $(1, 2)$ ，对应的分支因子是 $\varphi < 1.6181$ 。所以算法8-1能够在 $\mathcal{O}^*(1.6181^k)$ 时间内将 Generalized-SFVS(C) 问题的任意实例 \mathcal{I} 转化为一个约简实例 \mathcal{I}' 。

由于算法8-1中的每个步骤仅有删点、删边和标记边三种基本操作。所以，如果 \mathcal{I} 不包含内部关键点，则 \mathcal{I}' 中也没有内部关键点。进而算法8-1将依据引理8.12，在多项式时间内归约到 Generalized-SFVS(S) 问题的实例 \mathcal{I}'' 。然后在 $\mathcal{O}^*(\alpha^k)$ 时间内求解 \mathcal{I}'' 。综上，算法8-1整体的时间复杂度为 $\mathcal{O}^*(\alpha^k + 1.6181^k)$ ，引理得证。 ■

对于约简实例中存在内部关键点的情况，将在8.3.3小节中进行讨论解决。

8.3.3 基于团树上分治技术的参数算法

回顾8.3.2小节的结论, Generalized-SFVS(C) 问题的实例 $\mathcal{I} = (\mathbf{G}, T, M, k)$ 可以通过算法8-1转化为约简实例。如果约简实例中无内部关键点, 那么算法8-1会调用求解 Generalized-SFVS(S) 问题的参数算法直接进行求解。假设 Generalized-SFVS(S) 问题能在 $\mathcal{O}^*(\alpha^k)$ 时间内求解, 那么此时 \mathcal{I} 则能在 $\mathcal{O}^*(\alpha^k + 1.6181^k)$ 时间内求解。

在本小节中, 假设 \mathcal{I} 是已经通过算法8-1得到的约简实例, 并且 \mathcal{I} 中存在内部关键点。针对这种情况, 本小节将给出三条分支规则, 它们都基于分治思想而建立, 并且假设任意一条规则只有在前述的规则均不可执行时才会开始执行。注意到, 一条规则执行完成后, 得到的实例不一定是约简的, 因此需要重新调用算法8-1进一步转化为新的约简实例。

定义 8.14 (分治割与单纯块) 若一个极小点割 Q 满足以下条件, 则成为分治割 (Dividing Separator)

1. 点割 Q 包含一个内部关键点;
2. 团树 \mathcal{T}_G 中存在一条边 Q_1Q_2 满足 $Q = Q_1 \cap Q_2$;
3. 存在 $\mathbf{G} - Q$ 中的连通块 C_Q , 使得导出子图 $\mathbf{G}[C_Q \cup Q]$ 中没有内部关键点。

此外, 上述连通块 C_Q 称为 (关于 Q) 的单纯块 (Simplicial Component)。

根据定义, 若内部关键点存在, 则分治割必然存在。首先在线性时间内找到 \mathbf{G} 的所有极大团 Q_G , 然后计算一个团树 \mathcal{T}_G ^[285]。注意到, 团树中至多有 $n - 1$ 条边, 因此可以枚举并检查每条边, 从而能够在多项式时间内找到分治割 Q 及其对应的一个单纯块 $C_Q \subseteq V(\mathbf{G})$, 其中 $Q = Q_1 \cap Q_2$ 并且 Q_1Q_2 是团树种的一条边。

顾名思义, 后续的步骤将基于分治割上的分治技术实现。本小节将使用以下概念 (参见图8-2)。设 Q 包含了内部关键点 \hat{t} 。不失一般性, 假设 $Q \subsetneq Q_1 \subsetneq C_Q \cup Q$ 。由于 Q_1 和 Q_2 都包含内部关键点 \hat{t} , 所以 Q_1 和 Q_2 均不是单纯团。但是, $C_Q \setminus Q_1$ 至少包含一个单纯点, 这是因为团树中的每个叶子均为单纯团。

设 $Q_1 \setminus \{\hat{t}\} = \{v_1, v_2, \dots, v_\ell\}$ 以及 $C_0 := (C_Q \setminus Q_1) \cup \{\hat{t}\}$ 。对于任意顶点 $v_i \in Q_1 \setminus \{\hat{t}\}$, 定义子集 $C_i := C_0 \cup \{v_i\}$ 。用 \mathcal{I}_i 表示由 $\mathbf{G}[C_i]$ 导出的子实例 ($i \in \{0, 1, \dots, \ell\}$), 回顾导出子实例的定义, 有 $\mathcal{I}_i = (\mathbf{G}[C_i], C_i \cap T, M \cap E(\mathbf{G}[C_i]), k)$ 。不难发现, \mathcal{I}_i 完全由分治割和单纯块决定的。由于 $\mathbf{G}[C_i]$ 是 $\mathbf{G}[C_Q \cup Q]$ 的子图, 所以子实例 \mathcal{I}_i 中不包含内部关键点。因此根据引理8.13, 算法8-1可以在 $\mathcal{O}^*(\alpha^k + 1.6181^k)$ 时间内求解 \mathcal{I}_i 并得到最优解集的大小, 记作 s_i 。

由于计算每个子实例用时 $\mathcal{O}^*(\alpha^k + 1.6181^k)$, 并且一共需要计算 $\ell < n$ 个子实例, 所以求解 s_i ($i \in \{0, 1, \dots, \ell\}$) 总运行时间为 $\mathcal{O}^*(\alpha^k + 1.6181^k)$ 。

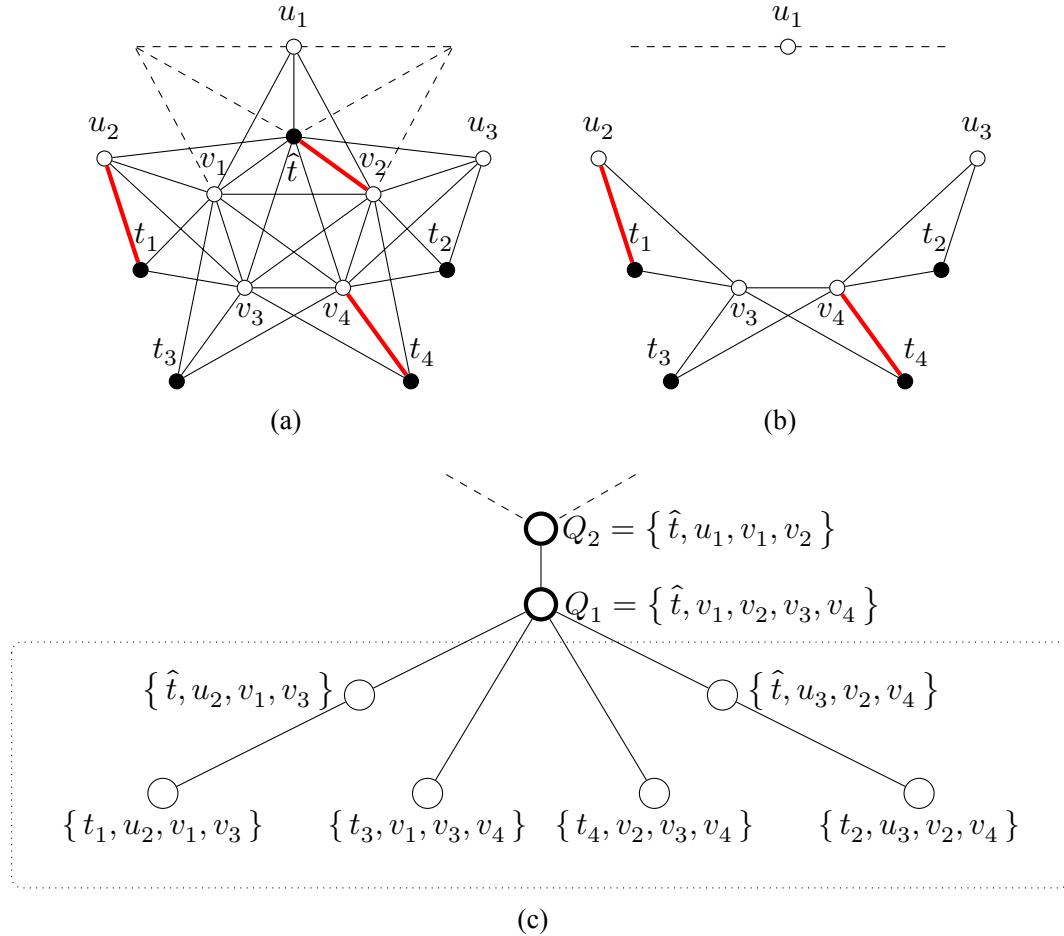


图 8-2 弦图中的内部关键点、分治割和单纯块示意图。(a) 为一个弦图 \mathbf{G} 的局部结构。(b) 为删去分治割 Q 之后的子图 $\mathbf{G} - Q$ 的局部结构。黑色顶点表示关键点 $\{\hat{t}\} \cup \{t_i\}_{i=1}^4$, 白色顶点表示 $\{u_i\}_{i=1}^3 \cup \{v_i\}_{i=1}^4$ 。红色粗线表示标记边, 虚线表示与局部结构之外的顶点的连边情况。(c) 表示图 \mathbf{G} 的一个团树 $\mathcal{T}_{\mathbf{G}}$, 其中的节点表示 \mathbf{G} 中极大团。关键点 \hat{t} 是一个内部关键点, $Q = Q_1 \cap Q_2 = \{\hat{t}, v_1, v_2\}$ 是分治割, 其中 Q_1 和 Q_2 是 (c) 中的粗轮廓节点。 $C_Q = \{t_i\}_{i=1}^4 \cup \{u_2, u_3, v_3, v_4\}$ 是关于 Q 的单纯块, 是由虚线框中的团中顶点的并再移除 Q 中顶点得到的。

注意到, 若某个子实例 \mathcal{I}_i 无解, 则有 $s_i > k$, 那么实例 \mathcal{I} 亦无解。因此, 后文假设 $s_i \leq k$ 总成立。此外, 对于每个下标 $i \in [\ell]$, 由于 $|C_i| = |C_0| + 1$, 所以有大小关系 $s_0 \leq s_i \leq s_0 + 1$ 。因此得到 $Q_1 \setminus \{\hat{t}\}$ 的二划分 (U_0, U_1) , 其中 $U_0 := \{v_i \in Q_1 \setminus \{\hat{t}\} : s_i = s_0\}$ 以及 $U_1 := \{v_i \in Q_1 \setminus \{\hat{t}\} : s_i = s_0 + 1\}$ (参见图8-3)。

本节提出的团树上的分治思想主要依赖于以下性质: 对于任意点割 Q' 以及

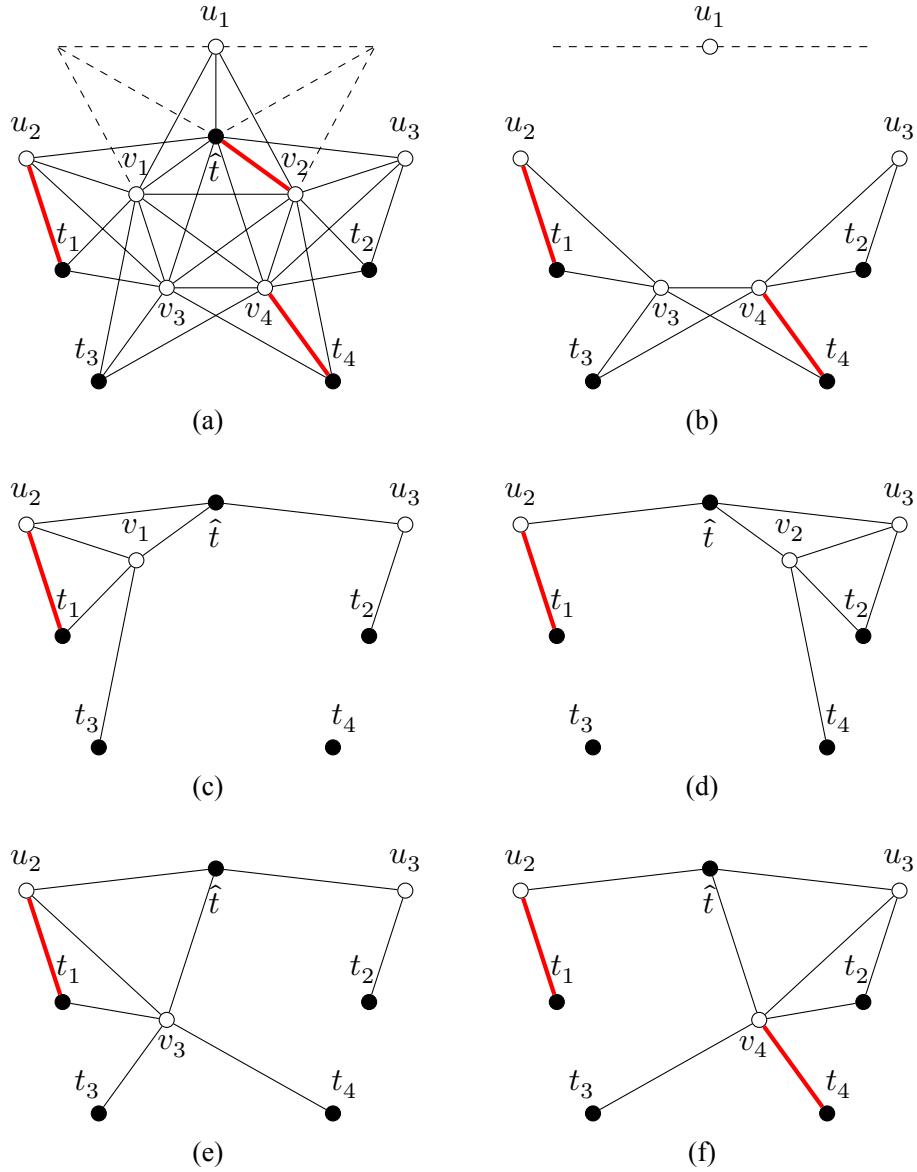


图 8-3 由分治割和单纯块决定的子实例示意图。(a) 为一个弦图 \mathbf{G} 的局部结构。(b) 为删去分治割 Q 之后的子图 $\mathbf{G}-Q$ 的局部结构。黑色顶点表示关键点 $\{\hat{t}\} \cup \{t_i\}_{i=1}^4$, 白色顶点表示 $\{u_i\}_{i=1}^3 \cup \{v_i\}_{i=1}^4$ 。红色粗线表示标记边, 虚线表示与局部结构之外的顶点的连边情况。 $C_Q = \{t_i\}_{i=1}^4 \cup \{u_2, u_3, v_3, v_4\}$ 是关于 Q 的单纯块, 是由虚线框中的团中顶点的并再移除 Q 中顶点得到的。点子集 $C_0 = (C_Q \setminus Q_1) \cup \{\hat{t}\} = \{t_i\}_{i=1}^4 \cup \{\hat{t}, u_2, u_3\}$ 并且由 C_0 导出的子实例的最优解集大小为 $s_0 = 1$ 。点子集 $C_i = C_0 \cup \{v_i\}$ 。(c)-(f) 分别是点子集 $\mathbf{G}[C_i]$ 导出的子图 ($i \in [4]$), 由 C_i 导出的子实例的最优解集大小满足 $s_1 = s_3 = 1$ 以及 $s_2 = s_4 = 2$ 。所以有 $U_0 = \{v_1, v_3\}$ 并且 $U_1 = \{v_2, v_4\}$ 。

$\mathbf{G} - Q'$ 中的连通分量 C' , 大小不超过 k 的点集 S 是 \mathcal{I} 的解当且仅当以下条件同时成立:

- $S \cap (X' \cup Q')$ 是由 $X' \cup Q'$ 导出的子实例的解;
- $S \setminus X'$ 是由 $V(\mathbf{G}) - C'$ 导出的子实例的解。

这一性质将多次应用于后文的论证中。

设 $\bar{\mathcal{I}}$ 是由 $V(\mathbf{G}) - C_0$ 导出的子实例。考虑 \hat{t} 不在解中的情况。此时除 \hat{t} 外, Q_1 中至多有一个顶点不在解中。考察顶点 v_i 是否在解中 ($i \in [\ell]$)。注意到, v_i 可能同时在两个子实例 $\bar{\mathcal{I}}$ 和 \mathcal{I}_i 的最优解中; 因此根据 s_i 的值能够部分判断 v_i 是否属于最优解。基于以上观察, 可以得到引理8.15。

引理 8.15 若约简实例 \mathcal{I} 中存在不包含内部关键点 \hat{t} 的最优解, 那么存在最优解 S , 它包含 U_1 中所有顶点, 但不包含 \hat{t} 。

证明: 采用反证法。假设 S 是 $\mathcal{I} = (\mathbf{G}, T, M, k)$ 的最优解, 且不包括顶点 $v_i \in U_1$ 以及内部关键点 $\hat{t} \in T$ 。以下通过构造另一个不包括关键点 \hat{t} 但包含 U_1 的最优解 S' 导出矛盾。

注意到, S 包含 $Q_1 \setminus \{\hat{t}, v_i\}$ 中的所有顶点; 否则在子图 $\mathbf{G} - S$ 中, 存在一个包含了 \hat{t} 和 v_i 的 T -三角形。所以, 可以得出

$$|S \cap Q_1| = |Q_1| - 2.$$

又因为 $S \cap C_i$ 是子实例 \mathcal{I}_i 的解, 进而有

$$|S \cap C_i| \geq s_i = s_0 + 1.$$

根据定义, 有 $C_i = C_0 \cup \{v_i\}$ 。又由于 $v_i \notin S$, 有

$$S \cap C_0 = S \cap C_i.$$

结合上述关系式, 可得

$$|S \cap (Q_1 \cup C_0)| = |Q_1| - 2 + s_i \geq s_0 + |Q_1| - 1.$$

令 S_0 是由 C_0 导出的子实例 \mathcal{I}_0 的解。那么,

$$S' = (S \setminus (Q_1 \cup C_0)) \cup (S_0 \cup Q_1 \setminus \{\hat{t}\})$$

仍然是 \mathcal{I} 的解。因为 $S \setminus C_0 \subseteq S'$, 所以 $S' \setminus C_0$ 是由 $V(\mathbf{G}) \setminus C_0$ 导出的子实例的解; 此外, 因为 $S \cap (S_0 \cup Q_1) \subseteq S'$, 所以 $S' \cap (S_0 \cup Q_1)$ 是由 $S_0 \cup Q_1$ 导出的子实例的解。进而说明 S' 是 \mathcal{I} 的 T -反馈子集。此外, 根据 S' 的构造, 有

$$|S'| \leq |S| - (s_0 + |Q_1| - 1) + (|S_0| + |Q_1| - 1) = |S|,$$

这说明了 S 是 \mathcal{I} 的最优解。

最后, 根据 S' 的构造, 有 $S' \cap Q_1 = Q_1 \setminus \{\hat{t}\}$, 进而说明了 S' 包含了 $Q_1 \setminus \{\hat{t}\}$ 中的所有顶点, 引理得证。 ■

依据引理8.15便可得到以下分支规则。

分支规则 8.3.9 (U_1 -分支规则) 若 $s_0 + |U_1| \geq 2$, 则分别计算以下两个子分支。若两个子分支均“无解”, 则返回“无解”; 否则返回“有解”。

- 删除内部关键点 \hat{t} , 并使 k 减少 1;
- 删除 $C_0 \setminus \{\hat{t}\}$ 和 U_1 , 并使 k 减少 $s_0 + |U_1| \geq 2$ 。

引理 8.16 U_1 -分支规则8.3.9是正确的, 且分支向量不劣于 $(1, 2)$ 。

证明: 只需证明, 若 \hat{t} 不在解时, 第二个分支的操作是正确的。第二个分支将实例 \mathcal{I} 转化为实例 $\mathcal{I}' = (\mathbf{G} - C_0 - U_1, T(C_0 \cup U_1), k - s_0 - |U_1|)$ 。根据引理8.15, 若存在解不包含 \hat{t} , 一定存在解 S 包含 U_1 , 因此可以将 U_1 删除, 然后使 k 减少 $|U_1|$ 。因此, 不妨设 U_1 为空, 此时有 $\mathcal{I}' = (\mathbf{G} - C_0, T(C_0), k - s_0)$ 。

设 S 是 \mathcal{I} 的解, 那么 $|S| \leq k$ 。首先, $S \cap C_0$ 是 \mathcal{I}_0 的解, 所以有 $|S \cap C_0| \geq s_0$ 。

$$|S \cap C_0| = |S| - |S \cap C_0| \leq |S| - s_0.$$

这说明了 \mathcal{I}' 有解。

另一个方向, 设 S' 是 \mathcal{I}' 的解, 则 $|S'| \leq k - s_0$ 。下证 \mathcal{I} 有解。显然有 $|Q| - 2 \leq |S' \cap Q| \leq |Q| - 1$ 。根据 $S' \cap Q$ 的大小, 考虑两种情形。

情形 1: $|S' \cap Q| = |Q| - 2$ 。在此情形中, Q 中恰有一个非关键点 v_i 不在 S' 中。令 $S = S_i \cup S'$, 其中 S_i 是子实例 \mathcal{I}_i 的解。一方面, 有大小关系

$$|S| = |S' \cup S_i| \leq (k - s_i) + s_i \leq k.$$

另一方面, 因为 $S' \subseteq S$ 是由 $V(\mathbf{G}) \setminus C_0$ 导出的子实例的解; 并且 S_i 是子实例 \mathcal{I}_i 的解。所以 S 是 \mathcal{I} 的解。

情形 2: $|S' \cap Q| = |Q| - 1$ 。在此情形中, 令 $S = S_0 \cup S'$, 其中 S_0 是子实例 \mathcal{I}_0 的解。一方面, 有大小关系

$$|S| = |S' \cup S_0| \leq (k - s_0) + s_0 \leq k.$$

另一方面, 因为 $S' \subseteq S$ 是由 $V(\mathbf{G}) \setminus C_0$ 导出的子实例的解; 并且 S_0 是子实例 \mathcal{I}_0 的解。所以 S 是 \mathcal{I} 的解。

最后分析 U_1 -分支规则的分支向量。在第一个分支中, 仅有 \hat{t} 被删除, k 减少了 1。第二个分支中, 集合 $C_0 \setminus \{\hat{t}\}$ 和 U_1 被删除, k 减少了 $s_0 + |U_1| \geq 2$ 。所以分支向量不劣于 $(1, 2)$, 引理得证。 ■

显然, 若 U_1 -分支规则8.3.9不可执行, 那么 U_1 至多包含一个顶点。仍然考虑 \hat{t} 不在解中的情形, 此时 Q_1 中除了 \hat{t} 外, 至多有一个顶点不在解中。注意到, U_1 -分支规则8.3.9不可执行时, Q_1 中绝大多数顶点都属于 U_0 。若 $U_0 \cap Q$ 中存在顶点 v_i , 那么 v_i 不在极小点割 Q 中, 所以不将 v_i 放入解中时, $Q \setminus \{\hat{t}\}$ 全体必在解中。那么, 此时所有经过 v_i 的标记边或 T -三角形必然在子实例 \mathcal{I}_i 中。又因为 $s_i = s_0$, 这表明 v_i 也不会出现于任何子实例 \mathcal{I}_i 的最优解中。所以总存在解不包含 v_i , 这就引入了以下分支规则。

分支规则 8.3.10 (Q_1 -分支规则) 若 $s_0 + |Q_1| \geq 4$ 且 $U_0 \cap Q$ 非空, 任选一个顶点 $v_i \in U_0 \cap Q$ 分别计算以下两个子分支。若两个子分支均“无解”, 则返回“无解”; 否则返回“有解”。

- 删除内部关键点 \hat{t} , 并使 k 减少 1;
- 删除 $(Q_1 \cup C_0) \setminus \{\hat{t}, v_i\}$, 并使 k 减少 $s_0 + |Q_1| - 2 \geq 2$ 。

引理 8.17 Q_1 -分支规则8.3.10是正确的, 且分支向量不劣于 $(1, 2)$ 。

证明: 只需证明, 若 \hat{t} 不在解时, 第二个分支的操作是正确的。设 S 是一个不包含 \hat{t} 的解, 那么有 $|Q_1| - 2 \leq |S \cap Q_1|$ 。令顶点 v_i 属于 $U_0 \cap Q$ 。若解集 S 包含了顶点 v_i , 现在证明 $S' = (S \setminus X_i) \cup (Q_1 \setminus \{\hat{t}, v_i\}) \cup S_i$ 也是解。

注意到, 在删去了 S' 后, $\{\hat{t}\}$ 是一个点割。一方面, $S \setminus C_i \subseteq S'$ 意味着 $S' \setminus C_i$ 是由 $V(\mathbf{G}) \setminus C_i$ 导出的子实例的解; 且 $S_i \subseteq S'$ 蕴含了 $S' \cap C_i$ 是子实例 \mathcal{I}_i 的解。所以 S' 是 \mathcal{I} 的 T -反馈点集。

另一方面, $S \cap X_i$ 是 \mathcal{I}_i 的解, 那么有 $s_i \leq |S \cap X_i|$ 。进而可以得出

$$\begin{aligned} |S'| &= |S \setminus (X_i \cup Q_1)| + (|Q_1| - 2) + s_i \\ &= |S \setminus (X_i \cup Q_1)| + |S \cap Q_1| + |S \cap X_i| \leq |S|. \end{aligned}$$

这便说明了 S' 是实例 \mathcal{I} 的解。注意到, 根据 S' 的构造, 已经说明存在解包含 $Q_1 \setminus \{\hat{t}, v_i\}$ 以及 S_i 。因此将它们删除并且使 k 减少 $s_0 + |Q_1| - 2$ 是正确的。此外, 在删除了点集 $Q_1 \setminus \{\hat{t}, v_i\}$ 后, $\{\hat{t}\}$ 成为了一个点割, 且 C_i 中剩下的顶点都不在 T -三角形或标记边中。故可将 C_i 中剩余的顶点全部删除。

最后分析 U_1 -分支规则的分支向量。在第一个分支中, 仅有 \hat{t} 被删除, k 减少了 1。第二个分支中, 集合 $C_0 \setminus \{\hat{t}\}$ 和 U_1 被删除, k 减少了 $s_0 + |U_1| \geq 2$ 。所以分支向量不劣于 $(1, 2)$ 。最后, 由于 $s_0 + |Q_1| \geq 4$, 可知 Q_1 -分支规则的分支向量不劣于 $(1, 2)$ 。引理得证。 ■

当 U_1 -分支规则8.3.9和 Q_1 -分支规则8.3.10均不可执行时, s_0 和 $|U_1|$ 的值都非常小, 并且 Q_1 具有十分明确的结构特征。接下来将刻画这一特征, 然后基于此特征, 本节将给出最后一个分支规则。

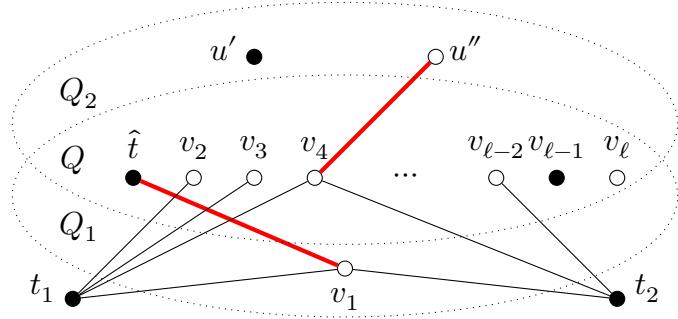


图 8-4 U_1 -分支规则和 Q_1 -分支规则均不可执行时的实例示意图。黑色顶点表示关键点 $\{\hat{t}, t_1, t_2, v_{\ell-1}, u'\}$ ，白色顶点表示非关键点 $\{v_1, v_2, \dots, v_{\ell}, u''\}$ 。
 $Q_1 = \{\hat{t}\} \cup \{v_i\}_{i=1}^{\ell}$ 和 $Q_2 = \{\hat{t}, u', u''\} \cup \{v_i\}_{i=2}^{\ell}$ 是包含内部关键点 \hat{t} 的极大团，用虚线椭圆框表示。 Q_1 和 Q_2 中的标记边用红色粗线表示，省略了未标记边。 $Q = Q_1 \cap Q_2$ 是一个分治割， $C_Q = \{\hat{t}, v_1, t_1, t_2\}$ 是单纯块，满足 $C_0 = \{\hat{t}, t_1, t_2\}$ 且 $s_0 = 0$ 。极大团 Q_1 中， $U_1 = \{v_1\} = Q_1 \setminus Q$ 且 $U_0 = \{v_i\}_{i=2}^{\ell}$ 。同时 t_1 和 t_2 是单纯点且 $C_Q \setminus Q_1 = \{t_1, t_2\}$ 。

引理 8.18 若一个约简实例包含内部关键点，且 U_1 -分支规则8.3.9和 Q_1 -分支规则8.3.10均不可执行，那么满足以下性质（参见图8-4）。

- (a) 由 C_0 导出的子实例不包含 T -三角形或标记边，即 $s_0 = 0$ ；
- (b) 点集 U_1 与 $Q_1 \setminus Q$ 相等，且恰好包含一个顶点；
- (c) 点集 $C_Q \setminus Q_1$ 非空，且其中的每个顶点均为单纯点。

证明：首先，由于 U_1 -分支规则不可执行，直接得到 $s_0 + |U_1| \leq 1$ 。若有 $s_0 + |Q_1| \leq 3$ ，那么有 $|Q| \leq |Q_1| - 1 \leq 2$ 。此外，由 $C_Q \cup (Q_1 \setminus Q)$ 导出的子实例的最优解集大小至多为 $s_0 + |Q_1| \leq 3$ 。这与引理8.11(d)矛盾。所以必然有 $s_0 + |Q_1| \geq 4$ 成立。又因为 Q_1 -分支规则不可执行，所以 $U_0 \setminus Q$ 必然为空。这意味着 $U_1 = Q_1 \setminus Q$ 并且 U_1 恰好包含一个顶点，所以性质(b)正确。此外，根据 $s_0 + |U_1| \leq 1$ 可知 $s_0 = 0$ ，进而得到了性质(a)。

接下来考察性质(b)。若 $C_Q \setminus Q_1$ 为空， $U_1 = Q_1 \setminus Q$ 中的唯一的顶点必然是单纯点，这意味着 u 是关键点。然而，此时关键点 \hat{t} 与单纯点 u 相邻，这与引理8.11(c)矛盾。所以(b)正确。

最后，采用反证法证明性质(c)。假设 $C_Q \setminus Q_1$ 中存在非单纯点。那么一定有单纯点 $t \in C_Q \setminus Q_1$ ，并且 t 有至少一个邻居不属于 Q_1 。假设 u 是 t 不在 Q_1 中的邻居。根据 t 的邻居的可能性，分三种情形讨论。

情形 1： $\hat{t} \in N_G(t)$ 。在此情形下，因为 t 是单纯点， \hat{t} 与 u 是相邻的。所以，顶点 \hat{t} , t , u 构成了一个在 $\mathbf{G}[C_0]$ 中的 T -三角形，与 $s_0 = 0$ 矛盾。

情形 2: 存在顶点 $v_i \in N_G(t) \cap Q$ ($i \in [\ell]$)。在此情形下, 顶点 v_i, t, u 构成一个 $\mathbf{G}[C_i]$ 中的 T -三角形, 与 $s_i = 0$ 矛盾。

情形 3: t 没有在 Q_1 中的邻居。根据引理8.11(b), 必然存在另一个与 u 和 v_1 都不相同的顶点 $u' \in N_G(t)$ 。这意味着顶点 u, u', t 构成 T -三角形, 这也与 $s_0 = 0$ 矛盾。

综上所述, 性质(c)是正确的, 引理得证。 ■

基于引理8.18, 现引入最后一个分支规则, 该分支规则将保证一定可以对集合 $Q_1 \cup Q_2$ 中的一些顶点进行删除操作, 使得图中的分治割数量减少。

分支规则 8.3.11 (分治割分支规则) 设 $U_1 = Q_1 \setminus Q = \{v_1\}$ 。若 v_1 与一个不同于 \hat{t} 的顶点 u 之间有一条标记边, 则分别计算以下两个子分支。若两个子分支均“无解”, 则返回“无解”; 否则返回“有解”。

- 删除 v_1 , 并使 k 减少 1;
- 删除 $\{\hat{t}, u\}$, 并使 k 减少 2.

否则, 分别计算以下两个子分支。若两个子分支均“无解”, 则返回“无解”; 否则返回“有解”(参见图8-4)。

- 删除 \hat{t} , 并使 k 减少 1;
- 删除 $(Q_1 \cup Q_2) \setminus Q$, 并使 k 减少 $|Q_1 \cup Q_2| - |Q| = 1 + |Q_2 \setminus Q| \geq 2$ 。

注意到, 根据分治割分支规则8.3.11的执行条件, 其中必然有一个子分支规则可执行, 取决于 v_1 是否与除了 \hat{t} 以外的某个顶点之间有标记边。

引理 8.19 分治割分支规则8.3.11是正确的, 且分支向量不劣于 $(1, 2)$ 。

证明: 首先考虑 v_1 与一个不同于 \hat{t} 的顶点 u 之间有一条标记边的情况。在此情况下只需证明若 v_1 不在解中时, 第二个分支的操作是正确的。此时, 由于 uv_1 是一条标记边, 那么 u 一定在解中。此外, 根据引理8.15, 一定存在最优解, 要么包含内部关键点 \hat{t} , 要么包含 $U_1 = \{v_1\}$ 。所以 v_1 不在解时, 将 \hat{t} 加入解是正确的。所以第二个分支是正确的, 并且 k 减少了 2。所以分支向量为 $(1, 2)$ 。

接下来考虑 v_1 不与除 \hat{t} 外的其他顶点通过标记边相连的情况。在此情况下, 只需证明若 \hat{t} 不在解中, 第二个分支的操作是正确的。设 S 是不包含 \hat{t} 的最优解, 必然有 $|Q_1 \setminus S| \leq 2$ 以及 $|Q_2 \setminus S| \leq 2$ 。下证, 若存在最优解 S 包含了 Q 中除了 \hat{t} 以外的所有顶点, 则存在解集 $S' = (S \setminus C_1) \cup \{\hat{t}\}$, 它包含了 \hat{t} 。这意味着一定存在最优解, 要么包含 \hat{t} ; 要么不包含 Q 中的两个顶点, 进而说明了这个解必定包含 $(Q_1 \cup Q_2) \setminus Q$ 。

回顾定义, $C_1 = C_0 \cup \{v_1\}$, \mathcal{I}_1 是由 C_1 导出的子实例, s_1 是 \mathcal{I}_1 的最优解大小。由于 U_1 -分支规则和 Q_1 -分支规则均不可执行, 根据引理8.18, 有 $s_0 = 0$ 以及

$s_1 = 1$ 。因为 $Q \setminus S = \{\hat{t}\}$, 所以有 $|S \cap C_1| \geq s_1 = 1$ 。这说明了

$$|S'| = |S \setminus C_1| + 1 = |S| - |S \cap C_1| + 1 \leq |S|.$$

进一步, 观察到 Q 是点割, 且 $Q \subseteq S'$ 。根据构造, $C_1 \setminus Q = C_Q \setminus Q$, 并且是 $\mathbf{G} - S'$ 中的连通块。根据引理8.18(c), $C_Q \setminus Q$ 中除了 v_1 以外的顶点均为单纯点。根据假设, v_1 与 $C_Q \setminus Q$ 中的其他顶点间没有标记边。这说明了导出子图 $\mathbf{G}[C_Q \setminus Q]$ 是星图, 不包含标记边和 T -三角形。又因为 $S \setminus C_Q \subseteq S'$, 所以 $S' \setminus C_Q$ 是由 $V(\mathbf{G}) - C_Q$ 导出的子实例的解。最终说明了 S' 是实例 \mathcal{I} 的解, 且包含 \hat{t} 。

最后分析该子分支规则的分支向量。第一个分支中, 仅有 \hat{t} 被删除, k 减少了 1。第二个分支中, 因为 $Q_1 \setminus Q$ 和 $Q_2 \setminus Q$ 均非空, 所以 k 减少了

$$|Q_1 \cup Q_2| - |Q| = |Q_1 \setminus Q| + |Q_2 \setminus Q| \geq 2.$$

这意味着分支向量不劣于 $(1, 2)$, 引理得证。 ■

根据上述规则, 现在给出以下分支搜索算法求解 Generalized-SFVS(C) 问题。

算法 8-2 Generalized-SFVS(C) 问题基于团树上分治技术的参数算法

输入: Generalized-SFVS(C) 问题的约简实例 $\mathcal{I} = (\mathbf{G}, T, M, k)$ 。
输出: 判断 \mathcal{I} 是否有解。

- 1 调用算法8-1返回结果, 或者得到约简实例并更新 \mathcal{I} 。// 此步骤后, \mathcal{I} 是一个总包含内部关键点的约简实例。
- 2 在线性时间内求解图 \mathbf{G} 的一个团树 \mathcal{T}_G 。
- 3 在多项式时间内找到一个分治割 $Q = Q_1 \cap Q_2$ 和关于 Q 的单纯块 C_Q , 其中 $Q_1 Q_2 \in E(\mathcal{T}_G)$ 。// 由于 $|E(\mathcal{T}_G)| \leq n$, 可以枚举每条边并再多项式时间内检查是否 $Q = Q_1 \cap Q_2$ 是分治割。
- 4 选取 Q 中的一个内部关键点 \hat{t} , 并对 $Q_1 \setminus \{\hat{t}\}$ 中的顶点编号: v_1, v_2, \dots, v_ℓ 。
- 5 计算子集 $C_0 = (C_Q \setminus Q_1) \cup \{\hat{t}\}$ 和 $C_i = C_0 \cup \{v_i\}$ ($i \in [\ell]$)。
- 6 **for** $i = 0$ **to** ℓ **do**
 - 7 依据引理8.13, 调用算法8-1计算由 C_i 导出的子实例 \mathcal{I}_i 。// 由于子实例的数量至多为 n , 故此步耗时 $\mathcal{O}^*(\alpha^k + 1.6181^k)$ 。
 - 8 **if** 实例 \mathcal{I}_i 无解 **then**
 - 9 | 直接返回“无解”。
 - 10 **end**
 - 11 | 令 s_i 为 \mathcal{I}_i 的最优解集大小。
- 12 **end**
- 13 执行 U_1 -分支规则8.3.9。// 此步骤后, $s_0 + |U_1| \leq 1$ 。
- 14 执行 Q_1 -分支规则8.3.10。// 此步骤后, 由引理8.18可知, $s_0 = 0$ 并且 $U_1 = Q_1 \setminus Q$ 大小为 1。
- 15 执行分治割分支规则8.3.11。// 此步骤包含两个子分支规则, 其中之一必然可执行。

定理 8.20 算法8-2是正确。若 Generalized-SFVS(S) 能在 $\mathcal{O}^*(\alpha^k)$ 时间内求解，则算法8-2能在 $\mathcal{O}^*(\alpha^k + 1.6181^k)$ 时间内求解 Generalized-SFVS(C) 问题。

证明：由于算法8-2中的三条分支规则都是正确的，所以算法8-2也是正确的。

依据引理8.13，算法8-2能在 $\mathcal{O}^*(1.6181^k)$ 时间内将 Generalized-SFVS(C) 问题的实例 \mathcal{I} 转化为约简实例 \mathcal{I}' 。此外，若 \mathcal{I}' 不包含内部关键点，则算法8-2能够在 $\mathcal{O}^*(\alpha^k + 1.6181^k)$ 时间内直接求解。若 \mathcal{I}' 包含了内部关键点，则将通过算法8-2后续步骤进行处理。

设 $\mathcal{T}(k)$ 表示实例 \mathcal{I} 的运行时间。由于算法8-2中的三条分支规则的分支向量不劣于 $(1, 2)$ ，所以可以得出

$$\mathcal{T}(k) \leq \mathcal{O}^*(\alpha^k + 1.6181^k) + \mathcal{T}(k-1) + \mathcal{T}(k-2).$$

最终可得 $\mathcal{T}(k) = \mathcal{O}^*(\alpha^k + 1.6181^k)$. ■

定理 8.21 SFVS(C) 问题可以在 $\mathcal{O}^*(1.8192^k)$ 时间内求解。

证明：结合定理7.4，可知 Generalized-SFVS(C) 问题能在 $\mathcal{O}^*(1.8192^k)$ 时间内求解。因为 Generalized-SFVS(C) 问题是 SFVS(C) 问题的广义形式，所以 SFVS(C) 问题也能在 $\mathcal{O}^*(1.8192^k)$ 时间内求解。定理得证。 ■

推论 8.21 SFVS(C) 问题可以在 $\mathcal{O}^*(1.8192^\tau)$ 时间内求解。

证明：当 $\tau \leq k$ 时，实例必然有解，可以在线性时间内判断。当 $\tau > k$ 时，根据定理8.2，可以在 $\mathcal{O}^*(1.8192^k) \leq \mathcal{O}^*(1.8192^\tau)$ 时间内求解。推论成立。 ■

8.3.4 基于均摊思想的精确算法

本节将利用 SFVS(C) 问题的参数算法设计精确算法。

根据推论8.21，SFVS(C) 可在 $\mathcal{O}^*(1.8192^\tau)$ 时间内求解。当 τ 足够小时，此算法足够快。因此需要为 τ 较大时的情况设计一个算法，这时非关键顶点较少。注意到，当所有顶点均为关键点时，SFVS(C) 问题退化为弦图上的反馈点集问题，而此问题是多项式时间可解的。基于上述观察，可以得到引理8.22。

引理 8.22 SFVS(C) 问题可以在 $\mathcal{O}^*(2^{n-\tau})$ 时间内求解。

证明：设 $\mathcal{I} = (\mathbf{G}, T, k)$ 为 SFVS(C) 问题的输入实例，其中顶点数为 $n = |V(\mathbf{G})|$ 且关键点数为 $\tau = |T|$ 。

首先枚举在解集中的非关键点，至多需要枚举 $2^{n-\tau}$ 个集合。对于每个枚举得到的集合 $W \subseteq V(\mathbf{G}) \setminus T$ ，将 W 从图中删除，并且余下的非关键点均不在解中。因此，若某个关键点和两个非关键点构成了三角形，则直接将该关键点删除，并使 k 减少 1。然后，由于相邻的两个非关键点 v 和 u 均不在解中，因此可以合并边 vu ：引入一个新的顶点 w_{vu} 使得 $N(w_{vu}) = N(\{v, u\})$ ，再删除顶点 v 和 u 。反复执行

合并操作直至图中的非关键点两两不相邻。注意到，由于弦图是相交图，合并任意边得到的图依然是弦图。

设当前的实例为 (\mathbf{G}', T', k') ，其中 $T = T'$ 且 $k' = k - |W|$ 。那么图 \mathbf{G}' 中的任意环都包含关键点。因此，实例 (\mathbf{G}', T', k') 的每个 T -反馈点集都是图 \mathbf{G}' 的反馈点集。由于所有非关键点都不在解中，因此为每个顶点按照以下方式赋权 $\omega(\cdot)$ ：关键点 $t \in T'$ 为单位权重 $\omega(t) = 1$ ，非关键点 $v \in V(\mathbf{G}')$ 为 T' 的权重设为 $\omega(v) = n$ 。不难发现，点集 $S' \subseteq T'$ 是图 \mathbf{G}' 的最小的只包含关键点的反馈点集，当且仅当 S' 是带权图 (\mathbf{G}', ω) 的最小权反馈点集。这便将实例 \mathcal{I}' 转化为了弦图上带权的反馈点集问题。而此问题能够在多项式时间完成求解^[210]。所以可以在多项式时间内计算 \mathcal{I}' 的只包含关键点的最小 T -反馈点集。最后，通过比较每个 W 所对应的最小 T -反馈点集的大小，便可判断实例 \mathcal{I} 是否有解。整体的时间复杂度为

$$2^{n-\tau} \cdot n^{\mathcal{O}(1)} = \mathcal{O}^*(2^{n-\tau}),$$

引理得证。 ■

根据推论8.21和引理8.22，可得到以下算法。

算法 8-3 SFVS(C) 问题基于均摊思想的精确算法

输入: SFVS(C) 问题的实例 $\mathcal{I} = (\mathbf{G}, T, k)$ 。

输出: 判断 \mathcal{I} 是否有解。

```

1 if  $\tau \leq 0.5367n$  then
2   | 依据推论8.21，在  $\mathcal{O}^*(1.8192^\tau)$  时间内求解并返回解。// 由于  $\tau \leq 0.5367n$ ，  

   | 运行时间为  $\mathcal{O}(1.3788^n)$ 。
3 else
4   | 依据引理8.22，在  $\mathcal{O}^*(2^{n-\tau})$  时间内求解并返回解。// 由于  $\tau > 0.5367n$ ，  

   | 运行时间为  $\mathcal{O}(1.3788^n)$ 。
5 end

```

定理 8.3 SFVS(C) 问题可以在 $\mathcal{O}(1.3788^n)$ 时间内求解。

证明: 根据推论8.21，SFVS(C) 问题能在 $\mathcal{O}^*(1.8192^\tau)$ 时间内求解。又根据引理8.22，SFVS(C) 问题可以在 $\mathcal{O}^*(2^{n-\tau})$ 时间内求解。结合定理3.7，可知 SFVS(C) 问题能在

$$\mathcal{O}^*(2^{(\log 1.8192 \log 2 / \log 1.8192 \times 2)n}) < \mathcal{O}^*(2^{0.4633n}) < \mathcal{O}(1.3788^n).$$

时间内求解。定理得证。 ■

8.4 弦图上限制版子集反馈点集问题

本节将探讨弦图上限制版子集反馈点集问题的参数算法与精确算法，改进了前人的结果。此外，本节证明了在指数时间意义下，求解 R-SFVS(S) 问题与 R-SFVS(C) 问题运行时间相同。主要结论如下。

定理 8.23 R-SFVS(C) 问题可以在 $\mathcal{O}^*(1.2529^k)$ 时间内求解。

定理 8.24 R-SFVS(C) 问题可以在 $\mathcal{O}(1.1520^n)$ 时间内求解。

8.4.1 基本的约简规则

类似于8.3.2节中提出的针对 Generalized-SFVS 问题的约简和分支规则，也可以为 R-SFVS(C) 问题设计相应的实例简化规则。特别地，由于在 R-SFVS(C) 问题中，解集不包含关键点，只需要引入少量简单的约简规则。

设 $\mathcal{I} = (\mathbf{G}, T, k)$ 为 R-SFVS(C) 问题的实例。本小节将给出基本的五条约简规则，将实例 \mathcal{I} 化简，并证明当所有关键点都是单纯点时，可以进一步将实例归约到 R-SFVS(S) 问题进行求解。本小节将使用传统度量 k 进行分析，由于所有规则仅包含删点或删边操作，因此每条规则都至多执行多项式次，并且每一次执行时参数 k 的值均不增加。本小节假设任意一条规则只有在前述的规则均不可执行时才会开始执行。

约简规则 8.4.1 (无解约简规则) 若存在全部由关键点构成的三角形，直接返回“无解”。

约简规则 8.4.2 (非关键点约简规则) 若存在一个三角形恰好包含两个关键点，则将该三角形中唯一的非关键点删除，并使 k 减少 1。

根据 R-SFVS(C) 问题的定义，关键点均不在解集中。因此无解约简规则8.4.1和非关键点约简规则8.4.2的正确性是直观的。

约简规则 8.4.3 (桥约简规则) 若存在桥，则直接将其删除。

约简规则 8.4.4 (无效顶点约简规则) 若顶点 v 不属于任何 T -三角形中，则直接删除顶点 v 。

桥约简规则8.4.3和无效顶点约简规则8.4.4也是平凡的，因为无效顶点和桥都不出现在任何 T -三角形里。此外，如果上述四条规则均无法执行，那么关键点集合必然是独立集，非关键点至少与一个关键点相连，并且极大团的大小至少为 3。

约简规则 8.4.5 (单纯点约简规则) 若存在单纯非关键点 v ，它属于单纯团 Q ，且与关键点 t 相连，则删除 $Q \setminus \{t, v\}$ ，并使 k 减少 $|Q| - 2$ 。

引理 8.25 单纯点约简规则8.4.5是正确的。

证明: 由于 t 不在解集 S 中，那么 S 至少包含 Q 中的 $|Q| - 2$ 个顶点，即 Q

中除了 t 以外, 至多还有一个顶点不在 S 中。否则 $Q \setminus S$ 中存在两个顶点和 t 构成 T -三角形。将 $S \cap Q$ 替换为 $Q \setminus \{t, v\}$ 可以得到集合 $S' = (S \setminus Q) \cup (Q \setminus \{t, v\})$ 。现证明 S' 也是解。一方面, 有大小关系 $|S'| \leq |S|$; 另一方面, 在子图 $\mathbf{G} - S'$ 中, v 仅与关键点 t 相连, 这说明在删除 S' 后, v 不在任何 T -三角形中。所以, S' 必为解。引理得证。 ■

事实上, 在单纯点约简规则8.4.5中, 非关键点 v 至少与一个关键点相邻, 这是因为无效顶点约简规则8.4.4不可执行。又因为无解约简规则8.4.1和非关键点约简规则8.4.2也不可执行, 所以 v 恰好与一个关键点相邻。

定义 8.26 (约简实例) 设 $\mathcal{I} = (\mathbf{G}, T, k)$ 是 R-SFVS(C) 问题的实例。若上述 5 种约简规则在 \mathcal{I} 上均不可执行, 则称 \mathcal{I} 为约简实例。

引理 8.27 若实例 $\mathcal{I} = (\mathbf{G}, T, M, k)$ 是约简实例, 则以下性质成立:

- (a) 图 \mathbf{G} 中无孤立点, 且关键集 T 是独立集;
- (b) 每个非关键点至少与一个关键点相邻;
- (c) 每个单纯团仅包含一个单纯点, 并且也是该团中唯一的键点。

证明: 孤立点不在任何 T -三角形中, 所以如果存在孤立点, 无效顶点约简规则可执行。此外, 如果两个关键点 t_1 与 t_2 相邻, 它们必然不在同 T -三角形中, 否则无解约简规则和非关键点约简规则其中之一可执行, 矛盾。但此时, $t_1 t_2$ 构成桥, 桥约简规则可执行, 亦矛盾。所以性质(a)是正确的。

性质(b)的正确性是直观的。这是因为不与任何关键点相邻的非关键点必然不在任何 T -三角形中, 此时无效顶点约简规则可执行。

最后考虑性质(c)的正确性。若单纯团中有一个非关键单纯点 v , 由性质(b), v 与某个关键点 t 相邻, 这意味着单纯点约简规则可执行, 矛盾。因此, Q 中的单纯点必为关键点。又由性质(a)可知, Q 仅包含一个关键点。所以性质(c)正确。

综上所述, 引理得证。 ■

基于约简实例的性质, 可以建立 R-SFVS(C) 问题与顶点覆盖问题之间的归约关系, 得到引理8.28。引理8.28是引理7.25的推广。

引理 8.28 R-SFVS(C) 问题的实例 $\mathcal{I} = (\mathbf{G}, T, k)$ 可以在多项式时间内归约到顶点覆盖问题的实例 $\mathcal{I}' = (\mathbf{G}', k')$, 满足 $|V(\mathbf{G}')| \leq |V(\mathbf{G})| - \tau$, $k' \leq k$ 。此外, \mathbf{G}' 存在大小不超过 κ 的团覆盖 \mathcal{C} , κ 为弦图 \mathbf{G} 中包含了关键点的极大团数量。

证明: 设 $\mathcal{I} = (\mathbf{G}, T, k)$ 为 R-SFVS(C) 问题的实例, 其中 $|V(\mathbf{G})| = n$ 且 $\tau = |T|$ 。令弦图 \mathbf{G} 中极大团构成的集族为 $\mathcal{Q}_{\mathbf{G}}$, 其中包含了关键点的极大团数量为 κ 。不妨设 \mathcal{I} 是约简实例。这是因为可以通过本小节五条约简规则将任意实例在多项式时间内转化为约简实例。由于这五条约简规则仅包含删点和删边两个基本操作, 所

以约简实例的关键点数量 τ , 非关键数量 $n - \tau$, 解集大小 k , 极大团数量 $|\mathcal{Q}_G|$, 包含了关键点的极大团数量为 κ 都是不增的。

根据引理8.27, 约简实例中 T 是独立集, 并且每个团至多包含一个关键点。所以有关系 $\tau \leq \kappa \leq |\mathcal{Q}_G|$, 其中 κ 为弦图 \mathbf{G} 中包含了关键点的极大团数量。

现构造顶点覆盖问题的实例 $\mathcal{I}' = (\mathbf{G}', k')$ 。首先, 对于点集 $V(\mathbf{G})$ 中的每个非关键点 $v \in K$, 构造一个顶点 v' , 得到图 \mathbf{G}' 的点集

$$V(\mathbf{G}') = \{v' : v \in V(\mathbf{G}) \setminus T\}.$$

然后, 对于每个包含了关键点的极大团 Q , 设其中的关键点为 $t \in T$, 构造一个团 $Q'_t = \{v' : v \in Q \setminus T\}$, 进而得到

$$E(\mathbf{G}') = \bigcup_{t \in Q, Q \in \mathcal{Q}_G} Q'_t = \bigcup_{t \in T} \{v' : v \in Q \setminus T\}.$$

根据构造, \mathbf{G}' 是一个包含 $n' = n - \tau$ 个顶点的无向图, 且存在大小为 κ 的团覆盖 $\mathcal{C} = \{Q'_t : t \in T\}$ 。因此, 有 $\text{ecc}(\mathbf{G}') \leq \kappa$ 。最后令 $k' = k$, 得到了顶点覆盖问题的实例 $\mathcal{I}' = (\mathbf{G}', k')$ 。

下证 \mathcal{I}' 有解当且仅当 \mathcal{I} 有解。首先论证充分性: 设 S' 为实例 \mathcal{I}' 的解, 现说明

$$S = \{v \in V(\mathbf{G}) : v' \in S'\}$$

是 \mathcal{I} 的解。考虑图 \mathbf{G} 中的任意 T -三角形 \mathbf{C} , 由于关键点集是独立集, 那么 \mathbf{C} 恰好包含一个关键点 $t \in T$ 和两个非关键点 $v, u \in V(\mathbf{G}) \setminus T$ 。根据构造, 有 $v', u' \in Q'_t$ 。由于 S' 是图 \mathbf{G}' 的顶点覆盖, 因此顶点 v' 和 u' 至少其中之一在 S' 中, 这意味着顶点 v 和 u 至少其中之一在 S 中。所以, $\mathbf{G} - S$ 中没有 T -三角形, 进而说明 S 是 T -反馈点集。注意到

$$|S| = |S'| \leq k' = k.$$

所以, S 是 \mathcal{I} 的解。

接着论证必要性: 设 S 为实例 \mathcal{I} 的解, 现说明

$$S' = \{v' \in V(\mathbf{G}') : v \in S\}$$

是 \mathcal{I}' 的解。考虑图 \mathbf{G}' 的团覆盖 \mathbf{C} 的任意一个团 Q'_t , 设 $v', u' \in Q'_t$ 。根据构造, 可知 (t, v, u) 是 T -三角形。由于 S 仅包含非关键点, 所以顶点 v 和 u 至少其中之一在 S 中, 这意味着顶点 v' 和 u' 至少其中之一在 S' 中。所以, S' 是顶点覆盖。

注意到

$$|S'| = |S| \leq k = k'.$$

所以, S' 是 \mathcal{I}' 的解。综上所述, 引理得证。 ■

根据引理8.28与引理7.26, 可以将 R-SFVS(C) 问题的任何实例多项式时间归约到 R-SFVS(S) 问题的实例。

推论 8.29 R-SFVS(C) 问题的实例 $\mathcal{I} = (\mathbf{G}, T, k)$ 可以在多项式时间内归约到 R-SFVS(S) 问题的实例 $\mathcal{I}' = (\mathbf{G}', T', k')$, 满足 $|V(\mathbf{G}')| \leq |V(\mathbf{G})| - \tau + \kappa$, $k' \leq k$ 以及 $|T'| \leq \kappa$, 其中 κ 为弦图 \mathbf{G} 中包含了关键点的极大团数量。

在推论8.29中 R-SFVS(C) 问题可以在多项式时间内归约到 R-SFVS(S) 问题, 保持解集大小不增, 但是总顶点数和关键点数可能增加。但这个结论足够将 R-SFVS(S) 的核心化算法推广到 R-SFVS(C) 问题上。注意到, 一般 R-SFVS(S) 问题和 R-SFVS(C) 问题之间的核心化结论没有蕴含关系。

定理 8.30 R-SFVS(C) 问题存在顶点数为 $k^2 + 2k$ 且边数为 $5k^2$ 的核。

证明: 设 $\mathcal{I} = (\mathbf{G}, T, k)$ 为 R-SFVS(C) 问题的实例, 设 \mathbf{G} 包含 $n = |V(\mathbf{G})|$ 个顶点, 其中有 $\tau = |T|$ 个关键点。首先, 依据推论8.29, 可以在多项式时间内将 \mathcal{I} 归约到 R-SFVS(S) 问题的实例 $\mathcal{I}_1 = (\mathbf{G}_1, T_1, k_1)$, 其中 $n_1 = |V(\mathbf{G}_1)|$ 且 $k_1 \leq k$ 。然后, 依据定理7.27, 使用 R-SFVS(S) 问题的核心化算法, 得到实例 \mathcal{I}_1 的线性核 $\mathcal{I}' = (\mathbf{G}', T', k')$, 其中 $n' = |V(\mathbf{G}')| \leq k_1^2 + 2k_1$ 且 $m' = |E(\mathbf{G}')| \leq 5k_1^2$ 。注意到 \mathcal{I}' 也可以视为 R-SFVS(C) 问题的实例。

现分析实例 \mathcal{I}' 的大小, 根据三个实例 \mathcal{I} , \mathcal{I}_1 , \mathcal{I}' 的大小关系, 可以得出

$$|V(\mathbf{G}')| \leq k_1^2 + 2k_1 \leq k^2 + 2k.$$

以及

$$|E(\mathbf{G}')| \leq 5k_1^2 \leq 5k^2.$$

综上所述, 定理得证。 ■

根据现有的结论, 除非 $\text{NP} \subseteq \text{coNP/poly}$, 对于任意正实数 $\varepsilon > 0$, 顶点覆盖问题不存在边数为 $\mathcal{O}(k^{2-\varepsilon})$ 的核^[246]。所以, 引理7.26表明 R-SFVS(C) 问题存在核心化下界。

定理 8.31 除非 $\text{NP} \subseteq \text{coNP/poly}$, 对于任意正实数 $\varepsilon > 0$, R-SFVS(C) 问题不存在边数为 $\mathcal{O}(k^{2-\varepsilon})$ 的核。

定理7.27和定理8.31共同说明了 R-SFVS(S) 问题关于边数的平方核是紧的。

虽然引理8.28和引理7.26建立了 R-SFVS(C) 问题和 R-SFVS(S) 问题多项式时

间归约关系。但是这个归约不保持关键集大小与点集大小，主要原因在于 κ 可能严格大于 τ ，因为可能存在某个关键点属于多个不同的极大团。所以，分裂图上的精确算法和以关键集大小为参数的参数算法都无法推广至 R-SFVS(C) 问题上。只有没有内部关键点存在时，有 $\kappa = \tau$ 成立，此时 R-SFVS(S) 问题的参数或精确算法都适用。回顾定义，若某个关键点不是单纯点，则称其是内部关键点。根据本节给出的规则，现在给出以下为具体算法要么直接求解 R-SFVS(C) 问题，要么得到一个 R-SFVS(C) 问题的约简实例。

算法 8-4 R-SFVS(C) 问题的约简实例转化算法

输入: R-SFVS(C) 问题的实例 $\mathcal{I} = (\mathbf{G}, T, k)$ 。

输出: 输出一个约简实例或直接判断 \mathcal{I} 是否有解。

- 1 执行无解约简规则8.4.1。// 此步骤后，不存在由关键点构成的三角形。
- 2 执行非关键点约简规则8.4.2。// 此步骤后，不存在的三角形中包含两个关键点。
- 3 执行桥约简规则8.4.3。// 此步骤后，关键点构成独立集，任意极大团的大小至少为 3。
- 4 执行无效顶点约简规则8.4.4。// 此步骤后，非关键点的邻居中均存在关键点。
- 5 执行单纯点约简规则8.4.5。// 此步骤后，每个单纯团包含唯一一个单纯点，并且也是唯一的关键点。
- 6 **if** 所有关键点均为单纯点 **then**
- 7 **if** 依据推论8.29，在多项式时间内归约到 SFVS(S) 问题，然后可以在 $\mathcal{O}^*(1.2529^k)$, $\mathcal{O}^*(2^\tau)$ 或 $\mathcal{O}(1.15199^n)$ 时间内进行求解。
- 8 **end**
- 9 返回当前实例。// 此时的实例必为约简实例且其中一定存在内部关键点。

注意到，算法8-4中的每个步骤仅有删点和删边两种基本操作组成。所以，如果实例 \mathcal{I} 没有内部关键点，那么通过算法8-4所得到的约简实例也没有内部关键点。所以本节部分地解决了 R-SFVS(C) 问题，得到了引理8.32。

引理 8.32 算法8-4是正确的，算法8-4能在多项式时间内将 R-SFVS(C) 问题的任意实例转化为约简实例。进一步地，若 R-SFVS(C) 问题实例没有内部关键点，则算法8-4可以在 $\mathcal{O}^*(1.2529^k)$, $\mathcal{O}^*(2^\tau)$ 或 $\mathcal{O}(1.15199^n)$ 时间内求解。

证明: 算法8-4中的每条规则都是正确的，所以算法8-4是正确的。

每条约简规则都能在多项式时间内执行完成，并且每条约简规则至多执行多项式次。所以算法8-4能够在多项式时间内将 R-SFVS(C) 问题的任意实例 \mathcal{I} 转化为约简实例 \mathcal{I}' 。

由于算法8-4中的每个步骤仅有删点和删边两种基本操作。所以，如果 \mathcal{I} 不包含内部关键点，则 \mathcal{I}' 中也没有内部关键点。进而算法8-4将依据引理8.28和引

理7.26在多项式时间内归约到 SFVS(S) 问题的实例 \mathcal{I}'' 。并且关键集大小, 解集大小和顶点数不增。依据定理7.23, 定理7.31和定理7.24, 可以在 $\mathcal{O}^*(1.2529^k)$, $\mathcal{O}^*(2^\tau)$ 或 $\mathcal{O}(1.15199^n)$ 时间内求解 \mathcal{I}'' 。

综上所述, 引理得证。 ■

对于约简实例中存在内部关键点的情况, 将在8.4.2小节中进行讨论并解决。

8.4.2 基于团树上分治技术的参数与精确算法

回顾8.4.1小节的结论, R-SFVS(C) 问题的实例 $\mathcal{I} = (\mathbf{G}, T, k)$ 可以通过算法8-4转化为约简实例。若约简实例中无内部关键点, 那么算法8-4会调用 R-SFVS(S) 问题的参数算法或精确算法直接进行求解。

在本小节中, 假设 \mathcal{I} 是已经通过算法8-4得到的约简实例, 并且 \mathcal{I} 中存在内部关键点。类似于8.3.3小节的讨论, 本小节将针对这种情况给出两条约简规则, 它们都基于分治思想而建立, 并且假设任意一条规则只有在前述的规则均不可执行时才会开始执行。注意到, 执行完一条规则之后, 得到的实例不一定是约简的, 因此需要重新调用算法8-4进一步转化为新的约简实例。

回顾8.3.3小节中的定义8.14。分治割是包含了内部关键点的极小点割 $Q = Q_1 \cap Q_2$, 满足 $Q_1 Q_2$ 是团树 \mathcal{T}_G 中的一条边, 并且 $\mathbf{G} - Q$ 中存在连通块 C_Q , 使得导出子图 $\mathbf{G}[C_Q \cup Q]$ 中没有内部关键点。连通块 C_Q 称为 (关于 Q) 的单纯块。根据定义, 若内部关键点存在, 则分治割必然存在。首先在线性时间内找到 \mathbf{G} 的所有极大团 \mathcal{Q}_G , 然后计算一个团树 \mathcal{T}_G ^[285]。注意到, 团树中至多有 $n - 1$ 条边, 因此可以枚举并检查每条边, 从而能够在多项式时间内找到分治割 Q 及其对应的一个单纯块 $C_Q \subseteq V(\mathbf{G})$, 其中 $Q = Q_1 \cap Q_2$ 并且 $Q_1 Q_2$ 是团树中的一条边。

后续的算法都讲使用以下概念。设 Q 包含了内部关键点 \hat{t} 。不失一般性, 假设 $Q \subsetneq Q_1 \subsetneq C_Q \cup Q$ 。由于 Q_1 和 Q_2 都包含内部关键点 \hat{t} , 所以 Q_1 和 Q_2 均不是单纯团。但是, $C_Q \setminus Q_1$ 中至少包含一个单纯点, 这是因为团树中的每个叶子均为单纯团。

设 $Q_1 \setminus \{\hat{t}\} = \{v_1, v_2, \dots, v_\ell\}$ 以及 $C_0 := (C_Q \setminus Q_1) \cup \{\hat{t}\}$ 。对于任意顶点 $v_i \in Q_1 \setminus \{\hat{t}\}$, 定义子集 $C_i := C_0 \cup \{v_i\}$ 。用 \mathcal{I}_i 表示由 $\mathbf{G}[C_i]$ 导出的子实例 ($i \in \{0, 1, \dots, \ell\}$), 回顾导出子实例的定义, 有 $\mathcal{I}_i = (\mathbf{G}[C_i], C_i \cap T, M \cap E(\mathbf{G}[C_i]), k)$ 。不难发现, \mathcal{I}_i 完全由分治割和单纯块决定。由于 $\mathbf{G}[C_i]$ 是 $\mathbf{G}[C_Q \cup Q]$ 的子图, 所以子实例 \mathcal{I}_i 中不包含内部关键点。因此根据引理8.32, 算法8-4可以在 $\mathcal{O}^*(1.2529^k)$, $\mathcal{O}^*(2^\tau)$ 或 $\mathcal{O}(1.15199^n)$ 时间内求解 \mathcal{I}_i 并得到最优解集的大小, 记作 s_i 。

由于一共需要计算 $\ell < n$ 个子实例, 所以求解 s_i ($i \in \{0, 1, \dots, \ell\}$) 总运行

时间依然为 $\mathcal{O}^*(1.2529^k)$, $\mathcal{O}^*(2^\tau)$ 或 $\mathcal{O}(1.15199^n)$ 。注意到, 若某个子实例 \mathcal{I}_i 无解, 则有 $s_i > k$, 那么实例 \mathcal{I} 也无解。因此, 后文假设 $s_i \leq k$ 总成立。此外, 对于每个下标 $i \in [\ell]$, 由于 $|C_i| = |C_0| + 1$, 所以有大小关系 $s_0 \leq s_i \leq s_0 + 1$ 。因此得到 $Q_1 \setminus \{\hat{t}\}$ 的二划分 (U_0, U_1) , 其中 $U_0 := \{v_i \in Q_1 \setminus \{\hat{t}\} : s_i = s_0\}$ 以及 $U_1 := \{v_i \in Q_1 \setminus \{\hat{t}\} : s_i = s_0 + 1\}$ 。

类似于引理8.15, 可以得到以下结论。

引理 8.33 若约简实例 \mathcal{I} 有解, 则存在最优解 S 包含 U_1 中所有顶点。

证明: 采用反证法。假设 S 是 $\mathcal{I} = (\mathbf{G}, T, k)$ 的最优解, 且不包括顶点 $v_i \in U_1$ 。以下通过构造另一个包含 U_1 的最优解 S' 导出矛盾。

注意到, S 包含 $Q_1 \setminus \{\hat{t}, v_i\}$ 中的所有顶点; 否则在子图 $\mathbf{G} - S$ 中, 存在包含了 \hat{t} 和 v_i 的 T -三角形。所以, 可以得出

$$|S \cap Q_1| = |Q_1| - 2.$$

又因为 $S \cap C_i$ 是由 C_i 导出的子实例 \mathcal{I}_i 的解, 进而有

$$|S \cap C_i| \geq s_i = s_0 + 1.$$

根据定义, 有 $C_i = C_0 \cup \{v_i\}$ 。又由于 $v_i \notin S$, 所以可得

$$S \cap C_0 = S \cap C_i.$$

结合上述关系式, 可得

$$|S \cap (Q_1 \cup C_0)| = |Q_1| - 2 + s_i \geq s_0 + |Q_1| - 1.$$

令 S_0 是由 C_0 导出的子实例 \mathcal{I}_0 的解。同时, 令

$$S' = (S \setminus (Q_1 \cup C_0)) \cup (S_0 \cup Q_1 \setminus \{\hat{t}\}).$$

注意到 Q_1 是点割, 同时 $S \setminus (Q_1 \cup C_0)$ 经过了 $\mathbf{G} - C_0$ 中的任意 T -三角形且 $S_0 \cup Q_1 \setminus \{\hat{t}\}$ 经过了 $\mathbf{G}[C_0 \cup Q_1]$ 中的任意 T -三角形。所以 S' 仍然是 \mathcal{I} 的解。

此外, 根据 S' 的构造, 有

$$|S'| \leq |S| - (s_0 + |Q_1| - 1) + (|S_0| + |Q_1| - 1) = |S|,$$

这说明了 S 是 \mathcal{I} 的最优解。

最后, 根据 S' 的构造, 有 $S' \cap Q_1 = Q_1 \setminus \{\hat{t}\}$, 进而说明了 S' 包含了 $Q_1 \setminus \{\hat{t}\}$ 中的所有顶点。

综上所述, 引理得证。 ■

基于引理8.33, 一定存在最优解包含集合 U_1 , 这直接导出了以下约简规则。

约简规则 8.4.6 (U_1 -约简规则) 若 U_1 非空, 则删除 U_1 , 并使 k 减少 $|U_1|$ 。

若 U_1 -约简规则8.4.6不可执行, 那么所有 Q_1 中的非关键点 v_i ($i \in [\ell]$) 均满足 $s_0 = s_i$ 。

约简规则 8.4.7 (单纯块约简规则) 若 U_1 为空, 则删除 C_0 并使 k 减少 s_0 。

引理 8.34 单纯块约简规则8.4.7是正确的。

证明: 单纯块约简规则将 \mathcal{I} 转化为实例 $\mathcal{I}' = (G - C_0, T \setminus C_0, k - s_0)$ 。设 S 是 \mathcal{I} 的解, 那么 $|S| \leq k$ 。首先, $S \cap C_0$ 经过了 $\mathbf{G}[C_0]$ 中的所有 T -三角形; 否则, S 不是 $\mathbf{G}[C_0]$ 的 T -反馈点集。所以 $S \cap C_0$ 是实例 \mathcal{I}_0 的解, 进而有 $|S \cap C_0| \geq s_0$ 。故可得

$$|S \setminus C_0| = |S| - |S \cap C_0| \leq |S| - s_0.$$

这说明了 \mathcal{I}' 有解。

另一个方向, 假设 S' 是 \mathcal{I}' 的解, 那么 $|S'| \leq k - s_0$ 。下证, \mathcal{I} 有解。由于 S' 经过了所有 Q_1 中的 T -三角形, 所以有

$$|Q| - 2 \leq |S' \cap Q| \leq |Q| - 1.$$

根据 $S' \cap Q$ 的大小, 考虑两种情形。

情形 1: $|S' \cap Q| = |Q| - 2$ 。在此情形中, Q 中恰有一个非关键点 v_j 不属于 S' 。令 $S = S_j \cup S'$, 其中 S_j 是子实例 \mathcal{I}_j 的解。一方面, 有大小关系

$$|S| = |S' \cup S_j| \leq (k - s_j) + s_j \leq k.$$

另一方面, 由于 Q_1 是点割, 同时 $S \setminus C_0$ 经过了 $\mathbf{G} - C_0$ 中的任意 T -三角形, 并且 $S_j \cup Q_1 \setminus \{\hat{t}\}$ 经过了 $\mathbf{G}[C_j \cup Q_1]$ 中的任意 T -三角形。所以 S 是 \mathcal{I} 的解。

情形 2: $|S' \cap Q| = |Q| - 1$ 。在此情形中, 令 $S = S_0 \cup S'$, 其中 S_0 是子实例 \mathcal{I}_0 的解。一方面, 有大小关系

$$|S| = |S' \cup S_0| \leq (k - s_0) + s_0 \leq k.$$

另一方面, 由于 Q_1 是点割, 同时 $S \setminus C_0$ 经过了 $\mathbf{G} - C_0$ 中的任意 T -三角形, 并且 $S_0 \cup Q_1 \setminus \{\hat{t}\}$ 经过了 $\mathbf{G}[C_0 \cup Q_1]$ 中的任意 T -三角形。所以 S 是 \mathcal{I} 的解。

综上所述, 引理得证。 ■

由于单纯块约简规则8.4.7总是可执行, 通过 U_1 -约简规则8.4.6和单纯块约简规则8.4.7, 可以给出以下算法求解 R-SFVS(C) 问题。

算法 8-5 R-SFVS(C) 问题基于团树上分治技术的参数或精确算法

输入: R-SFVS(C) 问题的约简实例 $\mathcal{I} = (\mathbf{G}, T, k)$ 。

输出: 判断 \mathcal{I} 是否有解。

- 1 调用算法8-4返回结果, 或者得到约简实例并更新 \mathcal{I} 。// 此步骤后, \mathcal{I} 是一个总包含内部关键点的约简实例。
- 2 在线性时间内求解图 \mathbf{G} 的团树 $\mathcal{T}_{\mathbf{G}}$ 。
- 3 在多项式时间内找到一个分治割 $Q = Q_1 \cap Q_2$ 和关于 Q 的单纯块 C_Q , 其中 $Q_1 Q_2 \in E(\mathcal{T}_{\mathbf{G}})$ 。// 由于 $|E(\mathcal{T}_{\mathbf{G}})| \leq n$, 可以枚举每条边并再多项式时间内检查是否 $Q = Q_1 \cap Q_2$ 是分治割。
- 4 选取 Q 中的一个内部关键点 \hat{t} , 并对 $Q_1 \setminus \{\hat{t}\}$ 中的顶点编号: $v_1, v_2, \dots, v_{\ell}$ 。
- 5 计算子集 $C_0 = (C_Q \setminus Q_1) \cup \{\hat{t}\}$ 和 $C_i = C_0 \cup \{v_i\}$ ($i \in [\ell]$)。
- 6 **for** $i = 0$ **to** ℓ **do**
- 7 | 依据引理8.32, 调用算法8-4计算由 C_i 导出的子实例 \mathcal{I}_i 。
- 8 | **if** 实例 \mathcal{I}_i 无解 **then**
- 9 | | 直接返回“无解”。
- 10 | **end**
- 11 | 令 s_i 为 \mathcal{I}_i 的最优解集大小。
- 12 **end**
- 13 执行 U_1 -约简规则8.4.6。// 此步骤后, U_1 为空, 即 $Q_1 = U_2 \cup \{\hat{t}\}$ 。
- 14 执行单纯块约简规则8.4.7。// 由于 U_1 -约简规则不可执行, 则 U_1 为空, 所以单纯块约简规则一定可执行。

定理 8.35 算法8-5是正确的。进一步, 对任意可计算函数 $f(\cdot, \cdot, \cdot)$, R-SFVS(S) 问题能在 $\mathcal{O}^*(f(n, k, \tau))$ 时间内求解当且仅当 R-SFVS(C) 问题能在 $\mathcal{O}^*(f(n, k, \tau))$ 时间内求解。

证明: 算法8-5中的两条约简规则都是正确的, 所以算法8-5是正确的。

只需证明, 若 R-SFVS(S) 问题能在 $\mathcal{O}^*(f(n, k, \tau))$ 时间内求解, 那么 R-SFVS(C) 问题也能在 $\mathcal{O}^*(f(n, k, \tau))$ 时间内求解。依据引理8.32, 算法8-5能在多项式时间内将 R-SFVS(C) 问题的实例 \mathcal{I} 转化为约简实例 \mathcal{I}' 。此外, 若 \mathcal{I}' 不包含内部关键点, 则算法8-5能够在 $\mathcal{O}^*(f(n, k, \tau))$ 时间内进行求解。若 \mathcal{I}' 包含了内部关键点, 则将通过算法8-5后续步骤进行处理, 并且后步骤中的两条约简规则都能在多项式时间内完成, 且至多执行 n 次。因此, R-SFVS(C) 问题能在

$$\mathcal{O}^*(f(n, k, \tau)) \cdot n^{\mathcal{O}(1)} = \mathcal{O}^*(f(n, k, \tau))$$

时间内进行求解。引理得证。 ■

定理 8.36 R-SFVS(C) 问题可以在 $\mathcal{O}^*(2^\tau)$ 时间内求解。

证明: 由于 R-SFVS(S) 问题可以在 $\mathcal{O}^*(2^\tau)$ 时间内求解, 根据定理8.35, R-SFVS(C) 问题也可以在 $\mathcal{O}^*(2^\tau)$ 时间内求解。定理得证。 ■

定理 8.23 R-SFVS(C) 问题可以在 $\mathcal{O}^*(1.2529^k)$ 时间内求解。

证明: 由于 R-SFVS(S) 问题可以在 $\mathcal{O}^*(1.2529^k)$ 时间内求解, 根据定理8.35, R-SFVS(C) 问题也可以在 $\mathcal{O}^*(1.2529^k)$ 时间内求解。定理得证。 ■

定理 8.24 R-SFVS(C) 问题可以在 $\mathcal{O}(1.1520^n)$ 时间内求解。

证明: 由于 R-SFVS(S) 问题可以在 $\mathcal{O}(1.15199^n)$ 时间内求解, 根据定理8.35, R-SFVS(C) 问题也可以在 $\mathcal{O}^*(1.15199^n) \leq \mathcal{O}(1.1520^n)$ 时间内求解。定理得证。 ■

8.5 本章小结

本章深入探讨了弦图上的子集反馈点集问题及其限制版, 在参数和精确算法方面实现了显著的改进。

针对 SFVS(C) 问题, 本章首次提出了突破 2^k 的参数算法, 该算法建立在第七章对 SFVS(S) 问题的算法研究基础之上。具体地, 本章首先论证了, 如果所有关键点均为单纯点, 则可以通过分支搜索技术得到约简实例, 然后归约至 SFVS(S) 问题完成求解, 运行时间为 $\mathcal{O}^*(1.8192^k)$ 。其次, 针对包含非单纯关键点的情况, 本文创新性地利用了树分解技术, 在弦图的团树上定义了分治割与单纯块等概念。由于单纯块中的所有关键点均为单纯点, 单纯块导出的子实例的最优解可在 $\mathcal{O}^*(1.8192^k)$ 时间内求解。最后, 利用子实例最优解的大小关系, 本章设计了基于分治割的分治算法, 成功建立了 SFVS(C) 问题与 SFVS(S) 问题之间的联系。基于这一联系, 本章证明了 SFVS(C) 问题可在 $\mathcal{O}^*(1.8192^k)$ 时间内求解, 成功地突破了 2^k 的计算复杂度瓶颈。进一步地, 此结果也表明, 除非找到求解 SFVS(S) 问题足够快的参数算法, 否则在指数时间意义下 SFVS(C) 问题和 SFVS(S) 问题均可在同样时间内求解。

值的强调的是, 本章的核心创新点在于引入了团树上的分治割与单纯块的概念, 摒弃了传统的动态规划思想。本文提出的概念有望应用于其他局部覆盖问题的子集版本或其他基于关键点的图连通性问题。

此外, 针对 R-SFVS(C) 问题, 本章也进行了全面的算法研究。同样采用弦图上的树分解技术, 本章亦建立了 R-SFVS(C) 问题与 R-SFVS(S) 问题之间的联系。在 R-SFVS(C) 问题中, 由于关键点不允许出现在解集中, 分裂图和弦图上限制版问题的关联性强于非限制版本。因此, 本文证明了 R-SFVS(S) 问题的精确与参数算法均可通过分治算法进行推广, 用以求解 R-SFVS(C) 问题, 且具有相同的指数时间复杂度。这一结论进一步揭示了 R-SFVS(C) 问题的求解难点完全来自于弦图的局部结构。

第九章 总结与展望

9.1 研究工作总结

本文全面系统地开展了反馈集相关问题的计算复杂性分类工作和参数与精确算法研究工作，取得了一系列完整丰富的理论研究成果。

计算复杂性综述与分类工作 本文的第一个工作属于本研究的前期工作，系统性整理了反馈集和子集反馈集问题各版本间的归约关系；综合梳理了这两个问题在一般图和特殊图类上的计算复杂性和算法研究成果。基于这一综述性成果，本文研究了反馈集和子集反馈集问题的计算复杂性分类，给出了相应的二分定理。

在无向图和平面无向图上，反馈集问题关于最大度的计算复杂性分类工作在1988年已基本完成。然而，在一般有向图和平面有向图上，反馈集问题的计算复杂性分类工作长期以来进展缓慢。本文首先证明，3度图上的有向反馈点集和有向反馈边集问题属于NP-难问题。其次，本文着手于研究平面有向图上的反馈点集问题。该问题的主要难点在于，有向图上需要分别考虑顶点的入度和出度，使得研究面临更复杂的挑战；同时，直接应用平面无向图上的相关技术难以避免完全二部图 $K_{3,3}$ 的非平面性。为克服这一难点，本文创新性地引入了非正规平面嵌入的概念，并提出了针对有向图上高度顶点的替换组件，成功证明了平面4度图上有向反馈点集问题是NP-难的。此外，本文还证明了3度平面图上有向反馈点集问题是多项式时间可解的，从而解决了（平面图上）有向反馈集问题关于最大度的计算复杂性分类难题。值得注意的是，与无向图上反馈点问题的计算复杂性分类结果不同，在平面有向图和一般有向图上，反馈点集问题的计算复杂性二分界限并不一致。具体地，3度图上有向反馈点集问题是NP-难的，而3度平面图上有向反馈点集问题能在多项式时间内求解。

子集反馈集问题作为图连通性问题中的一个核心研究对象，根据关键集大小的计算复杂性分类工作已取得了诸多成果。本文系统性归纳了子集反馈集问题的十二种不同版本。通过整理和建立各类问题之间的归约关系，本文发现，有向子集反馈边集问题、限制版有向子集反馈点集问题和限制版边子集反馈边集问题的计算复杂性分类尚未明确。本文证明了至少存在2个关键元素时，上述三类问题均为NP-难问题。这一成果完备了各版本子集反馈集问题关于关键集大小的计算复杂性分类工作。

三类隐式 3-碰撞集问题的算法研究工作 本文的第二个工作重点是竞赛图、分裂图和弦图上子集反馈点集问题的算法研究，这三个问题均属于著名的隐式 3-碰撞集问题。本文为这三个问题及其限制版设计了精确算法和以关键集大小 τ 和解集大小 k 为参数的参数算法。

(1) 三类非限制版子集反馈点集问题

本文改进了竞赛图、分裂图和弦图上的非限制版子集反馈点集问题现有的参数与精确算法，其中的参数算法运行时间均突破了 2^k 。

对于竞赛图上子集反馈点集问题，长期以来，研究人员未能发现比 3-碰撞集问题更快速的参数和精确算法。本文首先引入并改良了解集分离技术，克服了原技术对可近似性的依赖，成功将竞赛图上子集反馈点集问题的时间复杂度降至 $\mathcal{O}^*(2^{k+o(k)})$ 。进一步，本文借助“冲突边”的概念，创新性地提出了基于分割块的分支技术，以保证图结构具有足够弱的连通性。然后结合平衡割的分治思想，得到了的时间复杂度为 $\mathcal{O}^*(1.6181^k)$ 的参数算法。基于这一算法结果，本文进一步提出了运行时间为 $\mathcal{O}(1.3820^n)$ 的精确算法和 $\mathcal{O}^*(1.6181^\tau)$ 的参数算法。

对于分裂图和弦图上的子集反馈点集问题，近年内提出的参数算法将时间复杂度从 $\mathcal{O}^*(2.0755^k)$ 改进至 $\mathcal{O}^*(2^k)$ 。然而，该算法使用了复杂的分支搜索技术，其中的分支规则甚至包含了多达 7 个子分支，仅靠深入的结构分析和分类讨论难以取得实质性突破。此外，弦图上反馈点集问题能在多项式时间内求解，故也无法为分裂图或弦图上子集反馈点集问题的算法设计提供有效的帮助。本文引入“标记边”的概念，使用 Dulmage-Mendelsohn 分解对图结构进行深入刻画，采用度量治之分析手段证明了分裂图上子集反馈点集问题可以在 $\mathcal{O}^*(1.8192^k)$ 时间内求解。作为推论，本文为超图上带惩罚的独立集问题提出了首个非平凡的精确算法，运行时间为 $\mathcal{O}^*(1.8192^n)$ 。进一步，本文基于树分解上的分治技术，建立了分裂图和弦图上子集反馈点集问题的联系，为弦图上子集反馈点集问题设计了 $\mathcal{O}^*(1.8192^k)$ 时间的参数算法。基于此结果，本文也改进了以 n 和 τ 为参数的算法时间复杂度。

虽然本文考虑的三类隐式 3-碰撞集问题有显著的区别，但在算法设计过程中遇到了类似的难点：若已知某个顶点不在解集中，如何有效地将那些存储和利用？否则难以突破 3-碰撞集问题当前最快参数算法的运行时间上界。本文通过引入“冲突边”和“标记边”的概念，深入研究“冲突边”或“标记边”构成的子结构性质，同时建立这些子结构与原问题实例在算法意义上的联系，最终为三类隐式 3-碰撞集问题涉及了突破了 2^k 时间复杂度瓶颈的参数算法。上述技术路线有助于为更多不同的隐式 d -碰撞集问题实现 $(d-1)^k$ 的时间复杂度突破，为相关计算复杂性猜想的研究提供新的技术和证据。

(2) 三类限制版子集反馈点集问题

相比于非限制版问题，竞赛图、分裂图和弦图上的限制版子集反馈点集问题更为简单。本文针对这三类限制版子集反馈点集问题，提出了更快速的参数与精确算法。

本文主要的研究思路是建立三类限制版子集反馈点集问题与顶点覆盖问题的保持参数的归约关系。本文首先证明了三类限制版问题与顶点覆盖问题的参数算法在指数时间意义下，有相同的时间复杂度，这便得到了运行时间为 $\mathcal{O}^*(1.2529^k)$ 的参数算法。对于精确算法而言，本文证明了竞赛图上限制版问题与顶点覆盖问题依然能在相同指数时间 $\mathcal{O}(1.1996^n)$ 内求解；但对分裂图和弦图上限制版问题而言，此结论不成立。本文设计了精细地分支搜索算法，通过度量治之分析手段，证明了分裂图和弦图上的限制版本问题能够比顶点覆盖问题更快速地求解，运行时间为 $\mathcal{O}(1.1520^n)$ 。值得强调的是，尽管三类图上非限制版问题与顶点覆盖问题（即 2-碰撞集问题）有很多相似之处，但是不能将它们视为顶点覆盖问题的特例。主要原因在于非限制版问题的实例中，额外给定了一个不可删顶点集，因此顶点覆盖问题存在线性核不能推导出三类限制版问题也存在线性核。本文研究了三类限制版问题的核心化，给出了竞赛图上限制版问题 $3k$ 个顶点的线性核，以及分裂图和弦图上限制版问题的平方核。此外，本文给出的三类限制版问题核的比特数已达到理论最优。

本文进一步研究了以关键集大小 τ 为参数的参数算法。一方面，分裂图和弦图上限制版问题均可在 $\mathcal{O}^*(2^\tau)$ 时间内求解，故在 $\tau = \mathcal{O}(\log n)$ 的条件下该问题是多项式时间可解的。作为推论，本文证明了独立集问题以团覆盖数（Edge-Clique Cover Number）为参数是 FPT 的。另一方面，竞赛图上限制版自己反馈点集问题在 $\tau \geq 2$ 时是 NP-难的，因此不存在 FPT 算法。这标志着，三种限制版子集反馈点集问题之间具有显著的差异性。

9.2 后续工作展望

本论文系统全面地研究了反馈集相关问题的计算复杂性和参数与精确算法。鉴于已达博士学业年限，以下课题可作为未来研究方向，延续本文的研究工作。

特殊图类上反馈集相关问题计算复杂性 特殊图类上反馈集和子集反馈集问题的计算复杂性仍是理论计算机科学领域内一个活跃的研究方向。目前，弦图是已知其上反馈点集问题与子集反馈点集问题的计算复杂性不一致的极大图类。探究弦图的子图类（如强弦图）上的子集反馈点集问题是否可以在多项式时间内求解，

是一个具有重要研究价值的课题。

此外，单个禁止子图图类上反馈点集问题和子集反馈点集问题的计算复杂性分类工作是近年内备受关注的研究工作。一方面，若 \mathbf{H} -free 图上反馈点集或子集反馈点集问题是多项式的，那么 \mathbf{H} 必然是一个线性森林^[307,308]。若线性森林由 $s_1 + s_2 + \dots + s_r$ 条路组成，其中包括 s_i 条长度为 i 的路 \mathbf{P}_i ，则记 $\mathbf{H} = s_1\mathbf{P}_1 + s_2\mathbf{P}_2 + \dots + s_r\mathbf{P}_r$ 。目前，对子集反馈点集问题的相关计算复杂性分类工作已基本完成。具体地，当且仅当 \mathbf{H} 是 $2\mathbf{P}_1 + \mathbf{P}_4$ 的非空子图时， \mathbf{H} -free 图上子集反馈点集问题是多项式时间可解的^[309]。然而，对反馈点集问题的相关计算复杂性分类工作尚未完成。具体地，目前仅已知当 \mathbf{H} 是 $s\mathbf{P}_1 + \mathbf{P}_5$ 或 $s\mathbf{P}_3$ ($s \in \mathbb{N}$) 的非空子图时， \mathbf{H} -free 图上反馈点集问题是多项式时间可解的^[310-314]。但是， \mathbf{H} -free 图上反馈点集问题的计算复杂性下界依然是目前的公开性问题。

为隐式碰撞集问题突破 2^n 当前，仍然存在许多隐式碰撞集问题，它们没有比穷举搜索更高效的精确算法，其中包括子集奇圈交叉问题、有向奇圈交叉问题、有向子集奇圈交叉问题、有向多路割问题、有向多割问题等^[57,58]。这些隐式碰撞集问题都是经过长期研究的基础性问题，它们不仅是与工业生产和日常生活息息相关的组合优化问题，也是理论上亟待突破穷举搜索时间复杂度瓶颈的核心计算问题。一方面，近年来已有越来越多的研究发现，理论算法中的一些关键技术对实验算法的效率提升有着显著的作用。因此，通过提出非平凡的精确算法，可以为直接解决各领域面临的具体问题提供强大的理论和技术支持。另一方面，对此类问题的算法研究有助于理解 NP-完全问题以及相关的计算时间复杂度下界猜想。更重要的是，这一方向的研究成果能够为精确算法领域提供更为高效的工具和方法，从而为推动解决 NP-完全问题提供全新的算法设计思想。

为隐式 d -碰撞集问题突破 $(d-1)^k$ d -碰撞集问题能否在 $\mathcal{O}^*((d-1)^k)$ 时间内求解是一个备受关注的猜想。一方面，目前尚未发现计算复杂度为 $\mathcal{O}^*((d-1)^k)$ 的随机算法或近似率优秀的多项式甚至 FPT 近似算法。另一方面，目前许多隐式 d -碰撞集问题尚未突破 $(d-1)^k$ 的时间复杂度瓶颈。例如，三角形删除问题当前最高效的参数算法依然是 3-碰撞集问题的求解算法，其运行时间为 $\mathcal{O}^*(2.0755^k)$ 。

本文已为三类隐式 3-碰撞集问题设计了首个突破 2^k 的参数算法。未来将考虑为（隐式） d -碰撞集设计运行时间逼近 $\mathcal{O}^*((d-1)^k)$ 的确定性 FPT 算法、随机 FPT 算法或近似 FPT 算法。以期最终为 d -碰撞集问题突破 $(d-1)^k$ 做出贡献。

致 谢

夜已静，蛙齐鸣，笔渐停，思绪起。思来想去，还是决定采用这老套的开篇——舔犊情深，爱自此生。不忘父母养育之恩，永记父母启蒙之泽。你们无私的呵护是我成长中无尽的动力，你们默默的付出是我求学中坚实的后盾。父亲工作操劳栉风沐雨，上学放课早送晚接，夙兴夜寐含辛茹苦。母亲多年来的引导和教育，使我从小立志，走上愿为之不懈奋斗的学术之路。至今回想，难以相信，童蒙时从科普读物中了解的 Fair Cake-Cutting 竟与我博士研究中的机制设计工作不谋而合。

求学之路艰辛坎坷，承蒙恩师肖鸣宇教授知遇，没齿难忘，微意何曾有一毫，空携笔砚奉龙韬。恩师思想深邃，见闻广博，指导科研工作时，鱼渔皆授，学术嗅觉敏锐，见解独到深刻。学生多年受教铭刻于心获益匪浅。恩师治学严谨，尽心尽责，指导论文撰写时，殚精竭虑，逐字逐句考究，反复打磨推敲。学生初中稿的喜悦之情记忆犹新。恩师传道授业无以为报，自蒙半夜传衣后，不羨王祥得佩刀。学生学识浅薄初有所悟，昨夜西风凋碧树，独上高楼，望尽天涯路。

学术生涯路远苦多，感谢同门携手奋进的岁月；感谢课题组教授老师们在科研上的提携；感谢学院辅导老师和严管老师在工作上的关怀；感谢研究生院领导在学业上的支持；感谢中南大学、中国科学技术大学、香港城市大学、香港大学、香港理工大学、上海财经大学、上海交通大学、南京大学、斯坦福大学、卑尔根大学等世界顶级学府的访问学者在学术上的指导。

还要感谢我的家人和挚友们，谢谢你们平日里的嘘寒问暖，谢谢你们给予的鼓励、支持和帮助。常常记得长辈们挂念，尤念外祖多年的勉励，拙作谨献。

参考文献

- [1] Fomin F V, Kratsch D. *Exact exponential algorithms*[M]. Berlin: Springer, 2010.
- [2] Cygan M, Fomin F V, Kowalik L, et al. *Parameterized algorithms*[M]. Berlin: Springer, 2015.
- [3] Karp R M. Reducibility among combinatorial problems[C]. *Proceedings of a symposium on the Complexity of Computer Computations*, New York, New York, USA, 1972: 85-103.
- [4] Festa P, Pardalos P M, Resende M G. *Feedback set problems*[M]. Boston, MA: Springer US, 2009, 1005-1016.
- [5] Even G, Naor J, Schieber B, et al. Approximating minimum feedback sets and multicuts in directed graphs[J]. *Algorithmica*, 1998, 20(2): 151-174.
- [6] Even G, Naor J, Schieber B, et al. Approximating minimum subset feedback sets in undirected graphs with applications[C]. *Proceedings of the 40th Israel Symposium on Theory of Computing and Systems, ISTCS 1996*, Jerusalem, Israel, 1996: 78-88.
- [7] Even G, Naor J, Schieber B, et al. Approximating minimum subset feedback sets in undirected graphs with applications[J]. *SIAM Journal on Discrete Mathematics*, 2000, 13(2): 255-267.
- [8] Even G, Naor J, Rao S, et al. Divide-and-conquer approximation algorithms via spreading metrics (extended abstract)[C]. *Proceedings of the 36th Annual Symposium on Foundations of Computer Science, FOCS 1995*, Milwaukee, Wisconsin, USA, 1995: 62-71.
- [9] Even G, Naor J, Rao S, et al. Divide-and-conquer approximation algorithms via spreading metrics[J]. *Journal of the ACM*, 2000, 47(4): 585-616.
- [10] Unger S H. *A study of asynchronous logical feedback networks*[R]. Cambridge: Department of Electrical Engineering, Massachusetts Institute of Technology in Cambridge, 1957.
- [11] Minoura T. *Deadlock avoidance revisited*[J]. *Journal of ACM*, 1982, 29(4): 1023-1048.
- [12] Ailon N, Charikar M, Newman A. Aggregating inconsistent information: ranking and clustering[C]. *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, STOC 2005*, Baltimore, Maryland, USA, 2005: 684-693.
- [13] Ailon N, Charikar M, Newman A. Aggregating inconsistent information: Ranking and clustering[J]. *Journal of the ACM*, 2008, 55(5): 23:1-23:27.
- [14] Chen X, Ding G, Zang W, et al. Ranking tournaments with no errors I: structural description[J]. *Journal of Combinatorial Theory, Series B*, 2020, 141: 264-294.

- [15] Chen X, Ding G, Zang W, et al. Ranking tournaments with no errors II: minimax relation[J]. *Journal of Combinatorial Theory, Series B*, 2020, 142: 244-275.
- [16] Kunzmann A, Wunderlich H. An analytical approach to the partial scan problem[J]. *Journal of Electronic Testing*, 1990, 1(2): 163-174.
- [17] Kim J, Peled U N. Explicit construction of families of LDPC codes with no 4-cycles[C]. *Proceedings of the 2004 IEEE International Symposium on Information Theory, ISIT 2004*, Chicago, Illinois, USA, 2004: 235.
- [18] Kim J, Peled U N, Perepelitsa I, et al. Explicit construction of families of LDPC codes with no 4-cycles[J]. *IEEE Transactions on Information Theory*, 2004, 50(10): 2378-2388.
- [19] Park H, Hong S, No J, et al. Design of multiple-edge protographs for QC LDPC codes avoiding short inevitable cycles[J]. *IEEE Transactions on Information Theory*, 2013, 59(7): 4598-4614.
- [20] Budak C, Agrawal D, Abbadi A E. Limiting the spread of misinformation in social networks[C]. *Proceedings of the 20th International Conference on World Wide Web, WWW 2011*, Hyderabad, India, 2011: 665-674.
- [21] Simpson M, Srinivasan V, Thomo A. Clearing contamination in large networks[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2016, 28(6): 1435-1448.
- [22] Bar-Yehuda R, Geiger D, Naor J, et al. Approximation algorithms for the feedback vertex set problem with applications to constraint satisfaction and bayesian inference[J]. *SIAM Journal on Computing*, 1998, 27(4): 942-959.
- [23] Kemeny J G. Mathematics without numbers[J]. *Daedalus*, 1959, 88(4): 577-591.
- [24] Jiang X, Lim L, Yao Y, et al. Statistical ranking and combinatorial hodge theory[J]. *Mathematical Programming*, 2011, 127(1): 203-244.
- [25] Dechter R. Enhancement schemes for constraint processing: Backjumping, learning, and cutset decomposition[J]. *Artificial Intelligence*, 1990, 41(3): 273-312.
- [26] Bar-Yehuda R, Geiger D, Naor J, et al. Approximation algorithms for the vertex feedback set problem with applications to constraint satisfaction and bayesian inference[C]. *Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 1994*, Arlington, Virginia, USA, 1994: 344-354.
- [27] Ott J. *Analysis of human genetic linkage*[M]. The John Hopkins University Press, 1999.
- [28] Pearl J. *Probabilistic reasoning in intelligent systems - networks of plausible inference*[M]. San Francisco, California: Morgan Kaufmann, 1988.

- [29] Kratsch S, Schweitzer P. Isomorphism for graphs of bounded feedback vertex set number[C]. Proceedings of the 12th Scandinavian Workshop on Algorithm Theory, SWAT 2010, Bergen, Norway, 2010: 81-92.
- [30] Williams V V. On some fine-grained questions in algorithms and complexity[C]. Proceedings of the International Congress of Mathematicians 2018, ICM 2018, Rio de Janeiro, Rio de Janeiro, Brazil, 2018.
- [31] Impagliazzo R, Paturi R. On the complexity of k -SAT[J]. Journal of Computer and System Sciences, 2001, 62(2): 367-375.
- [32] Cygan M, Dell H, Lokshtanov D, et al. On problems as hard as CNF-SAT[C]. Proceedings of the 27th Conference on Computational Complexity, CCC 2012, Porto, Portugal, 2012: 74-84.
- [33] Cygan M, Dell H, Lokshtanov D, et al. On problems as hard as CNF-SAT[J]. ACM Transactions on Algorithms, 2016, 12(3): 41:1-41:24.
- [34] Xiao M, Nagamochi H. Exact algorithms for maximum independent set[J]. Information and Computation, 2017, 255: 126-146.
- [35] Bellman R. Dynamic programming treatment of the travelling salesman problem[J]. Journal of the ACM, 1962, 9(1): 61-63.
- [36] Coppersmith D, Vishkin U. Solving NP-hard problems in 'almost trees': Vertex cover[J]. Discret. Appl. Math., 1985, 10(1): 27-45.
- [37] Bodlaender H L. On disjoint cycles[C]. Proceedings of the 17th International Workshop on Graph-Theoretic Concepts in Computer Science, WG 1991, Fischbachau, Germany, 1991: 230-238.
- [38] Bodlaender H L. On disjoint cycles[J]. International Journal of Foundations of Computer Science, 1994, 5(1): 59-68.
- [39] Downey R G, Fellows M R. Fixed-parameter tractability and completeness I: basic results[J]. SIAM Journal on Computing, 1995, 24(4): 873-921.
- [40] Downey R G, Fellows M R. Fixed-parameter tractability and completeness II: on completeness for $W[1]$ [J]. Theoretical Computer Science, 1995, 141(1&2): 109-131.
- [41] Downey R G, Fellows M R. Fixed-parameter tractability and completeness III: some structural aspects of the W hierarchy[C]. Workshop on Structure and Complexity Theory, Dagstuhl, Germany, 1992: 191-225.

- [42] Abrahamson K R, Downey R G, Fellows M R. Fixed-parameter tractability and completeness IV: on completeness for $W[P]$ and PSPACE analogues[J]. Annals of Pure and Applied Logic, 1995, 73(3): 235-276.
- [43] Downey R G, Fellows M R. Parameterized complexity[M]. Berlin: Springer, 1999.
- [44] Stege U. Resolving conflicts from problems in computational biology[D]. Eidgenössische Technische Hochschule Zürich, 1999.
- [45] Roth-Korostensky C. Algorithms for building multiple sequence alignments and evolutionary trees[D]. Eidgenössische Technische Hochschule Zürich, 2000.
- [46] Chen J, Kanj I A, Jia W. Vertex cover: Further observations and further improvements[C]. Proceedings of the 25th International Workshop on Graph-Theoretic Concepts in Computer Science, WG 1999, Ascona, Switzerland, 1999: 313-324.
- [47] Chen J, Kanj I A, Jia W. Vertex cover: Further observations and further improvements[J]. Journal of Algorithms, 2001, 41(2): 280-301.
- [48] Downey R G, Fellows M R. Fundamentals of parameterized complexity[M]. Berlin: Springer, 2013.
- [49] Akiba T, Iwata Y. Branch-and-reduce exponential FPT algorithms in practice: A case study of vertex cover[C]. Proceedings of the 17th Workshop on Algorithm Engineering and Experiments, ALENEX 2015, San Diego, California, USA, 2015: 70-81.
- [50] Akiba T, Iwata Y. Branch-and-reduce exponential FPT algorithms in practice: A case study of vertex cover[J]. Theoretical Computer Science, 2016, 609: 211-225.
- [51] Iwata Y, Kobayashi Y. Improved analysis of highest-degree branching for feedback vertex set[C]. Proceedings of the 14th International Symposium of Parameterized and Exact Computation, IPEC 2019, Munich, Germany, 2019: 22:1-22:11.
- [52] Iwata Y, Kobayashi Y. Improved analysis of highest-degree branching for feedback vertex set[J]. Algorithmica, 2021, 83(8): 2503-2520.
- [53] Fernau H. Parameterized algorithmics for d -hitting set[J]. International Journal of Computer Mathematics, 2010, 87(14): 3157-3174.
- [54] Wahlström M. Algorithms, measures and upper bounds for satisfiability and related problems[D]. Linköping Sweden: Linköping University, 2007.
- [55] Cai L, Juedes D W. On the existence of subexponential parameterized algorithms[J]. Journal of Computer and System Sciences, 2003, 67(4): 789-807.

- [56] Fomin F V, Kratsch D, Woeginger G J. Exact (exponential) algorithms for the dominating set problem[C]. Proceedings of the 30th International Workshop on Graph-Theoretic Concepts in Computer Science, WG 2004, Bad Honnef, Germany, 2004: 245-256.
- [57] Chitnis R H, Fomin F V, Lokshtanov D, et al. Faster exact algorithms for some terminal set problems[C]. Proceedings of the 8th International Symposium of Parameterized and Exact Computation, IPEC 2013, Sophia Antipolis, France, 2013: 150-162.
- [58] Chitnis R, Fomin F V, Lokshtanov D, et al. Faster exact algorithms for some terminal set problems[J]. Journal of Computer and System Sciences, 2017, 88(3): 195-207.
- [59] Cai L. Fixed-parameter tractability of graph modification problems for hereditary properties[J]. Information Processing Letters, 1996, 58(4): 171-176.
- [60] Niedermeier R, Rossmanith P. An efficient fixed-parameter algorithm for 3-hitting set[J]. Journal of Discrete Algorithms, 2003, 1(1): 89-102.
- [61] Fomin F V, Gaspers S, Kratsch D, et al. Iterative compression and exact algorithms[J]. Theoretical Computer Science, 2010, 411(7-9): 1045-1053.
- [62] Dom M, Guo J, Hüffner F, et al. Fixed-parameter tractability results for feedback set problems in tournaments[J]. Journal of Discrete Algorithms, 2010, 8(1): 76-86.
- [63] Fernau H. Parameterized algorithms for d -hitting set: The weighted case[J]. Theoretical Computer Science, 2010, 411(16-18): 1698-1713.
- [64] Xiao M. A parameterized algorithm for bounded-degree vertex deletion[C]. Proceedings of the 22nd Annual International Computing and Combinatorics, COCOON 2016, Ho Chi Minh City, Vietnam, 2016: 79-91.
- [65] Downey R G, Fellows M R, Stege U. Parameterized complexity: A framework for systematically confronting computational intractability[C]. Proceedings of a DIMACS Workshop: Contemporary Trends in Discrete Mathematics: From DIMACS and DIMATIA to the Future, 1997, Stirín Castle, Czech Republic, 1997: 49-99.
- [66] Gramm J, Guo J, Hüffner F, et al. Automated generation of search tree algorithms for hard graph modification problems[J]. Algorithmica, 2004, 39(4): 321-347.
- [67] Dom M, Guo J, Hüffner F, et al. Fixed-parameter tractability results for feedback set problems in tournaments[C]. Proceedings of the 6th International Conference on Algorithms and Complexity, CIAC 2006, Rome, Italy, 2006: 320-331.

- [68] Hüffner F, Komusiewicz C, Moser H, et al. Fixed-parameter algorithms for cluster vertex deletion[C]. Proceedings of the 8th Latin American Symposium on Theoretical Informatics, LATIN 2008, Búzios, Brazil, 2008: 711-722.
- [69] Hüffner F, Komusiewicz C, Moser H, et al. Fixed-parameter algorithms for cluster vertex deletion[J]. Theory of Computing Systems, 2010, 47(1): 196-217.
- [70] Fernau H. A top-down approach to search-trees: Improved algorithmics for 3-hitting set[J]. Algorithmica, 2010, 57(1): 97-118.
- [71] Boral A, Cygan M, Kociumaka T, et al. A fast branching algorithm for cluster vertex deletion[C]. Proceedings of the 9th International Computer Science Symposium in Russia - Computer Science - Theory and Applications, CSR 2014, Moscow, Russia, 2014: 111-124.
- [72] Boral A, Cygan M, Kociumaka T, et al. A fast branching algorithm for cluster vertex deletion[J]. Theory of Computing Systems, 2016, 58(2): 357-376.
- [73] Tu J. A fixed-parameter algorithm for the vertex cover p_3 problem[J]. Information Processing Letter, 2015, 115(2): 96-99.
- [74] Wu B Y. A measure and conquer approach for the parameterized bounded degree-one vertex deletion[C]. Proceedings of the 21st Annual International Computing and Combinatorics, COCOON 2015, Beijing, China, 2015: 469-480.
- [75] Kumar M, Lokshtanov D. Faster exact and parameterized algorithm for feedback vertex set in tournaments[C]. Proceedings of the 33rd Symposium on Theoretical Aspects of Computer Science, STACS 2016, Orléans, France, 2016: 49:1-49:13.
- [76] Katriel J. A faster FPT algorithm for 3-path vertex cover[J]. Information Processing Letter, 2016, 116(4): 273-278.
- [77] Chang M, Chen L, Hung L, et al. Fixed-parameter algorithms for vertex cover p_3 [J]. Discrete Optimization, 2016, 19: 12-22.
- [78] Xiao M, Kou S. Kernelization and parameterized algorithms for 3-path vertex cover[C]. Proceedings of the 14th Annual Conference on Theory and Applications of Models of Computation, TAMC 2017, Bern, Switzerland, 2017: 654-668.
- [79] Xiao M, Kou S. A $5k$ -vertex kernel for 3-path vertex cover[J]. Theoretical Computer Science, 2023, 959: 113872.
- [80] Philip G, Rajan V, Saurabh S, et al. Subset feedback vertex set in chordal and split graphs[J]. Algorithmica, 2019, 81(9): 3586-3629.

- [81] Tsur D. Parameterized algorithm for 3-path vertex cover[J]. *Theoretical Computer Science*, 2019, 783: 1-8.
- [82] Tsur D. Faster parameterized algorithm for cluster vertex deletion[J]. *Theory of Computing Systems*, 2021, 65(2): 323-343.
- [83] Cervený R, Suchý O. Generating faster algorithms for d-path vertex cover[C]. *Proceedings of the 49th International Workshop on Graph-Theoretic Concepts in Computer Science, WG 2023, Fribourg, Switzerland, 2023*: 157-171.
- [84] Tian K, Xiao M, Yang B. Parameterized algorithms for cluster vertex deletion on degree-4 graphs and general graphs[C]. *Proceedings of the 29th Annual International Computing and Combinatorics, COCOON 2023 Part I, Hawaii, Hawaii, USA, 2023*: 182-194.
- [85] Chitnis R H, Hajiaghayi M, Marx D. Fixed-parameter tractability of directed multiway cut parameterized by the size of the cutset[C]. *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, 2012*: 1713-1725.
- [86] Chitnis R H, Hajiaghayi M, Marx D. Fixed-parameter tractability of directed multiway cut parameterized by the size of the cutset[J]. *SIAM Journal on Computing*, 2013, 42(4): 1674-1696.
- [87] Pilipczuk M, Wahlström M. Directed multicut is W[1]-hard, even for four terminal pairs[C]. *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, Virginia, USA, 2016*: 1167-1178.
- [88] Pilipczuk M, Wahlström M. Directed multicut is W[1]-hard, even for four terminal pairs[J]. *ACM Transactions on Computation Theory*, 2018, 10(3): 13:1-13:18.
- [89] Hatzel M, Jaffke L, Lima P T, et al. Fixed-parameter tractability of directed multicut with three terminal pairs parameterized by the size of the cutset: twin-width meets flow-augmentation[C]. *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, 2023*: 3229-3244.
- [90] Kloks T, Xiao M. *A guide to graph algorithms*[M]. Berlin: Springer, 2022.
- [91] Lovász L, Plummer M D. *Matching theory*[M]. Providence: American Mathematical Society, 1986.
- [92] Flum J, Grohe M. *Parameterized complexity theory*[M]. Berlin: Springer, 2006.
- [93] Edmonds J. Paths, trees, and flowers[J]. *Canadian Journal of Mathematics*, 1965, 17: 449-467.
- [94] Graham R L, Hell P. On the history of the minimum spanning tree problem[J]. *IEEE Annals of the History of Computing*, 1985, 7(1): 43-57.

- [95] Held M, Karp R M. A dynamic programming approach to sequencing problems[J]. *Journal of the Society for Industrial and Applied mathematics*, 1962, 10(1): 196-210.
- [96] Xiao M, Nagamochi H. Exact algorithms for maximum independent set[J]. *Information and Computation*, 2017, 255: 126-146.
- [97] Fomin F V, Gaspers S, Lokshtanov D, et al. Exact algorithms via monotone local search[J]. *Journal of the ACM*, 2019, 66(2): 8:1-8:23.
- [98] Harris D G, Narayanaswamy N S. A faster algorithm for vertex cover parameterized by solution size[C]. *Proceedings of the 41st Symposium on Theoretical Aspects of Computer Science, STACS 2024, Clermont-Ferrand, France, 2024*: 40:1-40:18.
- [99] Lin B, Ren X, Sun Y, et al. Constant approximating parameterized k -SetCover is W[2]-hard[C]. *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, 2023*: 3305-3316.
- [100] Nemhauser G L, Jr L E T. Properties of vertex packing and independence system polyhedra[J]. *Mathematical Programming*, 1974, 6(1): 48-61.
- [101] Dehne F K H A, Fellows M R, Rosamond F A, et al. Greedy localization, iterative compression, modeled crown reductions: New FPT techniques, an improved algorithm for set splitting, and a novel $2k$ kernelization for vertex cover[C]. *Proceedings of the 1st International Workshop of Parameterized and Exact Computation, IWPEC 2004, Bergen, Norway, 2004*: 271-280.
- [102] Iwata Y. Linear-time kernelization for feedback vertex set[C]. *Proceedings of the 44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, Warsaw, Poland, 2017*: 68:1-68:14.
- [103] Dell H, van Melkebeek D. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses[J]. *Journal of the ACM*, 2014, 61(4): 23:1-23:27.
- [104] Fomin F V, Gaspers S, Lokshtanov D, et al. Exact algorithms via monotone local search[C]. *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, Massachusetts, USA, 2016*: 764-775.
- [105] Bai T, Xiao M. Exact and parameterized algorithms for restricted subset feedback vertex set in chordal graphs[C]. *Proceedings of the 17th Annual Conference on Theory and Applications of Models of Computation, TAMC 2022, Tianjin, China, 2022*: 249-261.
- [106] König D. über graphen und ihre anwendung auf determinantentheorie und mengenlehre[J]. *Mathematische Annalen*, 1916, 77(4): 453-465.
- [107] König D. Gráfok és mótrixok[J]. *Matematikai és Fizikai Lapok*, 1931, 38(1031): 116-119.

- [108] Hall P. On representatives of subsets[J]. *Journal of the London Mathematical Society*, 1935, 10: 26-30.
- [109] Hopcroft J E, Karp R M. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs[J]. *SIAM Journal on Computing*, 1973, 2(4): 225-231.
- [110] Casel K, Friedrich T, Issac D, et al. Balanced crown decomposition for connectivity constraints[C]. *Proceedings of the 29th Annual European Symposium on Algorithms, ESA 2021, Lisbon, Portugal (Virtual)*, 2021: 26:1-26:15.
- [111] Dulmage A L, Mendelsohn N S. Coverings of bipartite graphs[J]. *Canadian Journal of Mathematics*, 1958, 10: 517-534.
- [112] Dulmage A L. A structure theory of bipartite graphs of finite exterior dimension[J]. *The Transactions of the Royal Society of Canada, Section III*, 1959, 53: 1-13.
- [113] Gallai T. Kritische graphen II[J]. *Magyar Tudományos Akadémia Matematikai Kutató Intézetének Közleményei*, 1963, 8: 373-395.
- [114] Gallai T. Maximale systeme unabhängiger kanten[J]. *Magyar Tudományos Akadémia Matematikai Kutató Intézetének Közleményei*, 1964, 9: 401-413.
- [115] Arnborg S, Corneil D G, Proskurowski A. Complexity of finding embeddings in a k -tree[J]. *SIAM Journal on Algebraic and Discrete Methods*, 1987, 8(2): 277-284.
- [116] Korhonen T, Lokshtanov D. An improved parameterized algorithm for treewidth[C]. *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, Florida, USA*, 2023: 528-541.
- [117] Shoikhet K, Geiger D. A practical algorithm for finding optimal triangulations[C]. *Proceedings of the 14th National Conference on Artificial Intelligence, AAAI 1997, Providence, Rhode Island, USA*, 1997: 185-190.
- [118] Courcelle B. The monadic second-order logic of graphs. i. recognizable sets of finite graphs[J]. *Information and Computation*, 1990, 85(1): 12-75.
- [119] Reed B A, Smith K, Vetta A. Finding odd cycle transversals[J]. *Operations Research Letters*, 2004, 32(4): 299-301.
- [120] Chen J, Liu Y, Lu S, et al. A fixed-parameter algorithm for the directed feedback vertex set problem[C]. *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, STOC 2008, Victoria, British Columbia, Canada*, 2008: 177-186.
- [121] Chen J, Liu Y, Lu S, et al. A fixed-parameter algorithm for the directed feedback vertex set problem[J]. *Journal of the ACM*, 2008, 55(5): 21:1-21:19.

- [122] Razgon I, O'Sullivan B. Almost 2-sat is fixed-parameter tractable (extended abstract)[C]. Proceedings of the 35th International Colloquium on Automata, Languages, and Programming, ICALP 2008 Part I, Reykjavik, Iceland, 2008: 551-562.
- [123] Razgon I, O'Sullivan B. Almost 2-sat is fixed-parameter tractable[J]. Journal of Computer and System Sciences, 2009, 75(8): 435-450.
- [124] Cao Y, Marx D. Chordal editing is fixed-parameter tractable[C]. Proceedings of the 31st Symposium on Theoretical Aspects of Computer Science, STACS 2014, Lyon, France, 2014: 214-225.
- [125] Cao Y, Marx D. Chordal editing is fixed-parameter tractable[J]. Algorithmica, 2016, 75(1): 118-137.
- [126] Esmer B C, Kulik A, Marx D, et al. Approximate monotone local search for weighted problems[C]. Proceedings of the 18th International Symposium of Parameterized and Exact Computation, IPEC 2023, Amsterdam, The Netherlands, 2023: 17:1-17:23.
- [127] Monien B, Schulz R. Four approximation algorithms for the feedback vertex set problem[C]. Proceedings of the 7th Conference Graphtheoretic Concepts in Computer Science, WG 1981, Linz, Austria, 1981: 315-326.
- [128] Becker A, Geiger D. Optimization of pearl's method of conditioning and greedy-like approximation algorithms for the vertex feedback set problem[J]. Artificial Intelligence, 1996, 83(1): 167-188.
- [129] Bafna V, Berman P, Fujito T. A 2-approximation algorithm for the undirected feedback vertex set problem[J]. SIAM Journal on Discrete Mathematics, 1999, 12(3): 289-297.
- [130] Chudak F A, Goemans M X, Hochbaum D S, et al. A primal-dual interpretation of two 2-approximation algorithms for the feedback vertex set problem in undirected graphs[J]. Operations Research Letters, 1998, 22(4-5): 111-118.
- [131] Raman V, Saurabh S, Subramanian C R. Faster fixed parameter tractable algorithms for undirected feedback vertex set[C]. Proceedings of the 13th International Symposium on Algorithms and Computation, ISAAC 2002, Vancouver, British Columbia, Canada, 2002: 241-248.
- [132] Kanj I A, Pelsmajer M J, Schaefer M. Parameterized algorithms for feedback vertex set[C]. Proceedings of the 1st International Workshop of Parameterized and Exact Computation, IWPEC 2004, Bergen, Norway, 2004: 235-247.

- [133] Raman V, Saurabh S, Subramanian C R. Faster fixed parameter tractable algorithms for finding feedback vertex sets[J]. ACM Transactions on Algorithms, 2006, 2(3): 403-415.
- [134] Dehne F K H A, Fellows M R, Langston M A, et al. An $\mathcal{O}(2^{\mathcal{O}(k)}n^3)$ FPT algorithm for the undirected feedback vertex set problem[C]. Proceedings of the 11th Annual International Computing and Combinatorics, COCOON 2005, Kunming, China, 2005: 859-869.
- [135] Guo J, Gramm J, Hüffner F, et al. Compression-based fixed-parameter algorithms for feedback vertex set and edge bipartization[J]. Journal of Computer and System Sciences, 2006, 72(8): 1386-1396.
- [136] Dehne F K H A, Fellows M R, Langston M A, et al. An $\mathcal{O}(2^{\mathcal{O}(k)}n^3)$ FPT algorithm for the undirected feedback vertex set problem[J]. Theory of Computing Systems, 2007, 41(3): 479-492.
- [137] Chen J, Fomin F V, Liu Y, et al. Improved algorithms for the feedback vertex set problems[C]. Proceedings of the 10th International Workshop on Algorithms and Data Structures, WADS 2007, Halifax, Canada, 2007: 422-433.
- [138] Chen J, Fomin F V, Liu Y, et al. Improved algorithms for feedback vertex set problems[J]. Journal of Computer and System Sciences, 2008, 74(7): 1188-1198.
- [139] Cao Y, Chen J, Liu Y. On feedback vertex set: New measure and new structures[J]. Algorithmica, 2015, 73(1): 63-86.
- [140] Kociumaka T, Pilipczuk M. Faster deterministic feedback vertex set[J]. Information Processing Letters, 2014, 114(10): 556-560.
- [141] Cao Y. A naive algorithm for feedback vertex set[C]. Proceedings of the 1st Symposium on Simplicity in Algorithms, SOSA 2018, New Orleans, Louisiana, USA, 2018: 1:1-1:9.
- [142] Cao Y, Chen J, Liu Y. On feedback vertex set new measure and new structures[C]. Proceedings of the 12th Scandinavian Symposium and Workshops on Algorithm Theory, SWAT 2010, Bergen, Norway, 2010: 93-104.
- [143] Burrage K, Estivill-Castro V, Fellows M R, et al. The undirected feedback vertex set problem has a $\text{poly}(k)$ kernel[C]. Proceedings of the 2nd International Workshop of Parameterized and Exact Computation, IWPEC 2006, Zürich, Switzerland, 2006: 192-202.
- [144] Bodlaender H L, van Dijk T C. A cubic kernel for feedback vertex set and loop cutset[J]. Theory of Computing Systems, 2010, 46(3): 566-597.

- [145] Thomassé S. A quadratic kernel for feedback vertex set[C]. Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2009, New York, New York, USA, 2009: 115-119.
- [146] Thomassé S. A $4k^2$ kernel for feedback vertex set[J]. ACM Transactions on Algorithms, 2010, 6(2): 32:1-32:8.
- [147] Razgon I. Exact computation of maximum induced forest[C]. Proceedings of the 10th Scandinavian Workshop on Algorithm Theory, SWAT 2006, Riga, Latvia, 2006: 160-171.
- [148] Fomin F V, Gaspers S, Pyatkin A V, et al. On the minimum feedback vertex set problem: Exact and enumeration algorithms[J]. Algorithmica, 2008, 52(2): 293-307.
- [149] Xiao M, Nagamochi H. An improved exact algorithm for undirected feedback vertex set[C]. Proceedings of the 7th International Conference International Conference on Combinatorial Optimization and Applications, COCOA 2013, Chengdu, China, 2013: 153-164.
- [150] Xiao M, Nagamochi H. An improved exact algorithm for undirected feedback vertex set[J]. Journal of Combinatorial Optimization, 2015, 30(2): 214-241.
- [151] Gaspers S, Lee E J. Exact algorithms via multivariate subroutines[C]. Proceedings of the 44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, Warsaw, Poland, 2017: 69:1-69:13.
- [152] Seymour P D. Packing directed circuits fractionally[J]. Combinatorica, 1995, 15(2): 281-288.
- [153] Guruswami V, Lee E. Simple proof of hardness of feedback vertex set[J]. Theory Computing, 2016, 12(1): 1-11.
- [154] Lokshtanov D, Ramanujan M S, Saurabh S. When recursion is better than iteration: A linear-time algorithm for acyclicity with few error vertices[C]. Proceedings of the 29th Symposium on Simplicity in Algorithms, SODA 2018, New Orleans, Los Angeles, USA, 2018: 1916-1933.
- [155] Cygan M, Fomin F, Jansen B M, et al. Open problems for fpt schools[R]. Będlewo, Poland: Conference Center of Institute of Mathematics of Polish Academy of Sciences, 2014.
- [156] Razgon I. Computing minimum directed feedback vertex set in $\mathcal{O}(1.9977^n)$ [C]. Proceedings of the 10th Italian Conference on Theoretical Computer Science, ICTCS 2007s, Rome, Italy, 2007: 70-81.
- [157] Koster E. A comment on minimum feedback arc sets[J]. IEEE Transactions on Circuit Theory, 1964, 11(2): 296-297.
- [158] Bodlaender H L, Fomin F V, Koster A M C A, et al. A note on exact algorithms for vertex ordering problems on graphs[J]. Theory of Computing Systems, 2012, 50(3): 420-432.

- [159] 白天, 肖鸣宇. 反馈集与子集反馈集问题的计算复杂性研究进展 [J]. 计算机研究与发展, 2023, 61: 1-15.
- [160] Xiao M, Nagamochi H. An FPT algorithm for edge subset feedback edge set[J]. Information Processing Letters, 2012, 112(1-2): 5-9.
- [161] Chitnis R H, Cygan M, Hajiaghayi M T, et al. Directed subset feedback vertex set is fixed-parameter tractable[C]. Proceedings of the 39th International Colloquium on Automata, Languages, and Programming, ICALP 2012, Part I, Warwick, UK, 2012: 230-241.
- [162] Chitnis R H, Cygan M, Hajiaghayi M T, et al. Directed subset feedback vertex set is fixed-parameter tractable[J]. ACM Transactions on Algorithms, 2015, 11(4): 28:1-28:28.
- [163] Even G, Naor J, Zosin L. An 8-approximation algorithm for the subset feedback vertex set problem[C]. Proceedings of the 37th Annual Symposium on Foundations of Computer Science, FOCS 1996, Burlington, Vermont, USA, 1996: 310-319.
- [164] Even G, Naor J, Zosin L. An 8-approximation algorithm for the subset feedback vertex set problem[J]. SIAM Journal on Computing, 2000, 30(4): 1231-1252.
- [165] Demaine E D, Hajiaghayi M, Marx D. Open problems from dagstuhl seminar 09511[C]. Parameterized complexity and approximation algorithms, Dagstuhl, Germany, 2009: 1-14.
- [166] Guillemot S. FPT algorithms for path-transversal and cycle-transversal problems[J]. Discrete Optimization, 2011, 8(1): 61-71.
- [167] Cygan M, Pilipczuk M, Pilipczuk M, et al. Subset feedback vertex set is fixed-parameter tractable[J]. SIAM Journal on Discrete Mathematics, 2013, 27(1): 290-309.
- [168] Kawarabayashi K, Kobayashi Y. Fixed-parameter tractability for the subset feedback set problem and the S -cycle packing problem[J]. Journal of Combinatorial Theory, 2012, 102(4): 1020-1034.
- [169] Guillemot S. FPT algorithms for path-transversals and cycle-transversals problems in graphs[C]. Proceedings of the 3rd International Workshop of Parameterized and Exact Computation, IWPEC 2008, Victoria, Canada, 2008: 129-140.
- [170] Cygan M, Pilipczuk M, Pilipczuk M, et al. Subset feedback vertex set is fixed-parameter tractable[C]. Proceedings of the 38th International Colloquium on Automata, Languages, and Programming, ICALP 2011, Part I, Zurich, Switzerland, 2011: 449-461.
- [171] Wahlström M. Half-integrality, LP-branching and FPT algorithms[C]. Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, 2014: 1762-1781.

- [172] Lokshtanov D, Ramanujan M S, Saurabh S. Linear time parameterized algorithms for subset feedback vertex set[C]. Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming, ICALP 2015, Part I, Kyoto, Japan, 2015: 935-946.
- [173] Iwata Y, Yamaguchi Y, Yoshida Y. 0/1/All CSPs, half-integral A -path packing, and linear-time FPT algorithms[C]. 59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, 2018: 462-473.
- [174] Lokshtanov D, Ramanujan M S, Saurabh S. Linear time parameterized algorithms for subset feedback vertex set[J]. ACM Transactions on Algorithms, 2018, 14(1): 7:1-7:37.
- [175] Iwata Y, Wahlström M, Yoshida Y. Half-integrality, LP-branching, and FPT algorithms[J]. SIAM Journal on Computing, 2016, 45(4): 1377-1411.
- [176] Fomin F V, Heggernes P, Kratsch D, et al. Enumerating minimal subset feedback vertex sets[C]. Proceedings of the 12th International Symposium on Algorithms and Data Structures, WADS 2011, New York, New York, USA, 2011: 399-410.
- [177] Fomin F V, Heggernes P, Kratsch D, et al. Enumerating minimal subset feedback vertex sets[J]. Algorithmica, 2014, 69(1): 216-231.
- [178] 周晓清, 肖鸣宇. 无向图中子集反馈顶点集问题的精确算法 [J]. 计算机学报, 2018, 41(3): 493-505.
- [179] Xiao M, Nagamochi H. A refined algorithm for maximum independent set in degree-4 graphs[J]. Journal of Combinatorial Optimization, 2017, 34(3): 830-873.
- [180] Xiao M, Nagamochi H. An exact algorithm for maximum independent set in degree-5 graphs[J]. Discrete Applied Mathematics, 2016, 199: 137-155.
- [181] Garey M R, Johnson D S. The rectilinear steiner tree problem is NP complete[J]. SIAM Journal of Applied Mathematics, 1977, 32(4): 826-834.
- [182] Koutsonas A, Thilikos D M. Planar feedback vertex set and face cover: Combinatorial bounds and subexponential algorithms[C]. Proceedings of the 34th International Workshop on Graph-Theoretic Concepts in Computer Science, WG 2008, Durham, UK, 2008: 264-274.
- [183] Koutsonas A, Thilikos D M. Planar feedback vertex set and face cover: Combinatorial bounds and subexponential algorithms[J]. Algorithmica, 2011, 60(4): 987-1003.
- [184] Lucchesi C L. A minimax equality for directed graphs[D]. Waterloo, Ontario: University of Waterloo, 1976.
- [185] Lucchesi C L, Younger D H. A minimax theorem for directed graphs[J]. Journal of the London Mathematical Society, 1978, s2-17(3): 369-374.

- [186] Gabow H N, Stallmann M F M. Efficient algorithms for graphic matroid intersection and parity (extended abstract)[C]. Proceedings of the 12th International Colloquium on Automata, Languages, and Programming, ICALP 1985, Nafplion, Greece, 1985: 210-220.
- [187] Speckenmeyer E. Untersuchungen zum feedback vertex set problem in ungerichteten graphen[D]. Paderborn, Germany: Universität Paderborn, 1983.
- [188] v D Cavallaro. Hamiltonicity and the computational complexity of graph problem[D]. Berlin, Germany: Technische Universität Berlitz, 2019.
- [189] Appel K, Haken W. Every planar map is four colorable. part I. discharging[J]. Illinois Journal of Mathematics, 1977, 21(3): 429-490.
- [190] Appel K, Haken W, Koch J. Every planar map is four colorable. part II. reducibility[J]. Illinois Journal of Mathematics, 1977, 21(3): 491-567.
- [191] Robertson N, Sanders D P, Seymour P D, et al. Efficiently four-coloring planar graphs[C]. Proceedings of the 28th Annual ACM Symposium on Theory of Computing, STOC 1996, Philadelphia, Pennsylvania, USA, 1996: 571-575.
- [192] Gabow H N. A framework for cost-scaling algorithms for submodular flow problems[C]. Proceedings of the 34th Annual Symposium on Foundations of Computer Science, Palo Alto, California, USA, 1993: 449-458.
- [193] Ueno S, Kajitani Y, Gotoh S. On the nonseparating independent set problem and feedback set problem for graphs with no vertex degree exceeding three[J]. Discrete Mathematics, 1988, 72(1-3): 355-360.
- [194] Furst M L, Gross J L, McGeoch L A. Finding a maximum-genus graph imbedding[J]. Journal of the ACM, 1988, 35(3): 523-534.
- [195] Garey M R, Johnson D S. Computers and intractability: A guide to the theory of NP-completeness[M]. San Francisco: W. H. Freeman, 1979.
- [196] Speckenmeyer E. On feedback problems in diagraphs[C]. Proceedings of the 15th International Workshop on Graph-Theoretic Concepts in Computer Science, WG 1989, Castle Rolduc, The Netherlands, 1989: 218-231.
- [197] Alon N. Ranking tournaments[J]. SIAM Journal on Discrete Mathematics, 2006, 20(1): 137-142.
- [198] Conitzer V. Computing slater rankings using similarities among candidates[C]. Proceedings of the 21st National Conference on Artificial Intelligence, AAAI 2006, Boston, Massachusetts, USA, 2006: 613-619.

- [199] Charbit P, Thomassé S, Yeo A. The minimum feedback arc set problem is NP-hard for tournaments[J]. *Combinatorics, Probability & Computing*, 2007, 16(1): 1-4.
- [200] Karpinski M, Schudy W. Faster algorithms for feedback arc set tournament, kemeny rank aggregation and betweenness tournament[C]. *Proceedings of the 21st International Symposium on Algorithms and Computation, ISAAC 2010, Part I, Jeju Island, Korea, 2010*: 3-14.
- [201] Cai M, Deng X, Zang W. A min-max theorem on feedback vertex sets[C]. *Proceedings of the 7th International Conference on Integer Programming and Combinatorial Optimization, IPCO 1999, Graz, Austria, 1999*: 73-86.
- [202] Cai M, Deng X, Zang W. A min-max theorem on feedback vertex sets[J]. *morCaiDZ02*, 2002, 27(2): 361-371.
- [203] Kumar M, Lokshtanov D. Faster exact and parameterized algorithm for feedback vertex set in bipartite tournaments[C]. *Proceedings of the 36th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2016, Chennai, India, 2016*: 24:1-24:15.
- [204] Guo J, Hüffner F, Moser H. Feedback arc set in bipartite tournaments is NP-complete[J]. *Information Processing Letters*, 2007, 102(2-3): 62-65.
- [205] Truß A. Parameterized algorithms for feedback set problems in tournaments[D]. Jena, Thuringia, German: Friedrich-Schiller-Universität Jena, 2005.
- [206] Raman V, Saurabh S. Parameterized algorithms for feedback set problems and their duals in tournaments[J]. *Theoretical Computer Science*, 2006, 351(3): 446-458.
- [207] Alon N, Lokshtanov D, Saurabh S. Fast FAST[C]. *Proceedings of the 36th International Colloquium on Automata, Languages, and Programming, ICALP 2009, Part I, Rhodes, Greece, 2009*: 49-58.
- [208] Brandstädt A, Le V B, Spinrad J P. Graph classes: A survey[M]. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1999.
- [209] Erdős P, Goodman A W, Pósa L. The representation of a graph by set intersections[J]. *Canadian Journal of Mathematics*, 1966, 18: 106-112.
- [210] Yannakakis M, Gavril F. The maximum k -colorable subgraph problem for chordal graphs[J]. *Information Processing Letters*, 1987, 24(2): 133-137.
- [211] Lu C L, Tang C Y. A linear-time algorithm for the weighted feedback vertex problem on interval graphs[J]. *Information Processing Letters*, 1997, 61(2): 107-111.

- [212] Papadopoulos C, Tzimas S. Polynomial-time algorithms for the subset feedback vertex set problem on interval graphs and permutation graphs[C]. Proceedings of the 21th International Symposium on Fundamentals of Computation Theory, FCT 2017, Bordeaux, France, 2017: 381-394.
- [213] Papadopoulos C, Tzimas S. Polynomial-time algorithms for the subset feedback vertex set problem on interval graphs and permutation graphs[J]. Discrete Applied Mathematics, 2019, 258: 204-221.
- [214] Brandstädt A, Kratsch D. On the restriction of some np-complete graph problems to permutation graphs[C]. Proceedings of the 5th International Symposium on Fundamentals of Computation Theory, FCT 1985, Bezirk Cottbus, German, 1985: 53-62.
- [215] Brandstädt A, Kratsch D. On domination problems for permutation and other graphs[J]. Theoretical Computer Science, 1987, 54: 181-198.
- [216] Liang Y D. On the feedback vertex set problem in permutation graphs[J]. Information Processing Letters, 1994, 52(3): 123-129.
- [217] Takaoka A, Tayu S, Ueno S. On minimum feedback vertex sets in bipartite graphs and degree-constraint graphs[J]. IEICE Transactions on Information & Systems, 2013, 96-D(11): 2327-2332.
- [218] Coorg S R, Rangan C P. Feedback vertex set on cocomparability graphs[J]. Networks, 1995, 26(2): 101-111.
- [219] Liang Y D, Chang M. Minimum feedback vertex sets in cocomparability graphs and convex bipartite graphs[J]. Acta Informatica, 1997, 34(5): 337-346.
- [220] Gavril F. Minimum weight feedback vertex sets in circle n -gon graphs and circle trapezoid graphs[J]. Discrete Mathematics, Algorithms and Applications, 2011, 3(3): 323-336.
- [221] Brandstädt A. On improved time bounds for permutation graph problems[C]. Proceedings of the 18th International Workshop on Graph-Theoretic Concepts in Computer Science, WG 1992, Wiesbaden-Naurod, Germany, 1992: 1-10.
- [222] Honma H, Kitamura Y, Masuyama S. An algorithm for minimum feedback vertex set problem on a trapezoid graph[J]. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 2011, 94-A(6): 1381-1385.
- [223] Speckenmeyer E. Bounds on feedback vertex sets of undirected cubic graphs[M]. Lonton: Elsevier Science Ltd., 1986, 719-729.

- [224] Speckenmeyer E. On feedback vertex sets and nonseparating independent sets in cubic graphs[J]. *Journal of Graph Theory*, 1988, 12(3): 405-412.
- [225] Rizzi R. Minimum weakly fundamental cycle bases are hard to find[J]. *Algorithmica*, 2009, 53(3): 402-424.
- [226] Brandstädt A, Brito S, Klein S, et al. Cycle transversals in perfect graphs and cographs[J]. *Theoretical Computer Science*, 2013, 469: 15-23.
- [227] Bousquet N, Daligault J, Thomassé S. Multicut is FPT[C]. *Proceedings of the 43th Annual ACM Symposium on Theory of Computing, STOC 2011, San Jose, California, USA, 2011*: 459-468.
- [228] Bousquet N, Daligault J, Thomassé S. Multicut is FPT[J]. *SIAM Journal on Computing*, 2018, 47(1): 166-207.
- [229] Whitney H. Hassler whitney's collected papers[M]. Boston, Massachusetts: Birkhäuser, 1992.
- [230] Duncan C A, Eppstein D, Kobourov S G. The geometric thickness of low degree graphs[C]. *Proceedings of the 20th ACM Symposium on Computational Geometry, SoCG 2004, Brooklyn, New York, USA*, 2004: 340-346.
- [231] Yannakakis M, Kanellakis P C, Cosmadakis S S, et al. Cutting and partitioning a graph after a fixed pattern (extended abstract)[C]. *Proceedings of the 10th International Colloquium on Automata, Languages, and Programming, ICALP 1983, Barcelona, Spain, 1983*: 712-722.
- [232] Garg N, Vazirani V V, Yannakakis M. Multiway cuts in directed and node weighted graphs[C]. *Proceedings of the 21st International Colloquium on Automata, Languages, and Programming, ICALP 1994, Jerusalem, Israel, 1994*: 487-498.
- [233] Garg N, Vazirani V V, Yannakakis M. Multiway cuts in node weighted graphs[J]. *Journal of Algorithms*, 2004, 50(1): 49-61.
- [234] Bar-Noy A, Naor J. Sorting, minimal feedback sets, and hamilton paths in tournaments[J]. *SIAM Journal on Discrete Mathematics*, 1990, 3(1): 7-20.
- [235] Landau J G. On dominance relations and the structure of animal societies III: The condition for a score structure[J]. *Bulletin of Mathematical Biophysics*, 1953, 15(2): 143 -148.
- [236] Seedig H G. Majority relations and tournament solutions (mehrheitsrelationen und turnierlösungen)[D]. Technical University of Munich, Germany, 2015.
- [237] Brill M, Schmidt-Kraepelin U, Suksompong W. Margin of victory for tournament solutions[J]. *Artificial Intelligence*, 2022, 302: 103600.

- [238] Charon I, Hudry O. A survey on the linear ordering problem for weighted or unweighted tournaments[J]. 4OR: Quarterly Journal of the Belgian, French and Italian Operations Research Societies, 2007, 5(1): 5-60.
- [239] Veerathu V, Rajkumar A. On the structure of parametric tournaments with application to ranking from pairwise comparisons[C]. Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, Virtual, 2021: 12065-12076.
- [240] Abu-Khzam F N. Kernelization algorithms for d -hitting set problems[C]. Proceedings of the 10th International Workshop on Algorithms and Data Structures, WADS 2007, 2007: 434-445.
- [241] Abu-Khzam F N. A kernelization algorithm for d -hitting set[J]. Journal of Computer and System Sciences, 2010, 76(7): 524-531.
- [242] Le T, Lokshtanov D, Saurabh S, et al. Subquadratic kernels for implicit 3-hitting set and 3-set packing problems[C]. Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, Los Angeles, USA, 2018: 331-342.
- [243] Fomin F V, Le T, Lokshtanov D, et al. Subquadratic kernels for implicit 3-hitting set and 3-set packing problems[J]. ACM Transactions on Algorithms, 2019, 15(1): 13:1-13:44.
- [244] Harary F, Moser L. The theory of round robin tournaments[J]. The American Mathematical Monthly, 1966, 73(3): 231-246.
- [245] Poljak S. A note on stable sets and colorings of graphs[J]. Commentationes Mathematicae Universitatis Carolinae, 1974, 15: 307-309.
- [246] Dell H, van Melkebeek D. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses[J]. Journal of the ACM, 2014, 61(4): 23:1-23:27.
- [247] Stephane F, Hammer P. Split graphs[C]. Proceedings of the 8th south-east Combinatorics, Graph Theory, and Computing, SEICCGTC 1977, Boca Raton, Florida, USA, 1977: 311-315.
- [248] Foldes S, Hammer P L. Split graphs having dilworth number two[J]. Canadian Journal of Mathematics, 1977, 29(3): 666-672.
- [249] Tyškevič R I, Čhernyak A A. Canonical decomposition of a graph determined by the degrees of its vertices[J]. Vestsī Akad. Navuk BSSR Ser. Fīz.-Mat. Navuk, 1979, 5(5): 14-26,138.
- [250] Jr L E T. Algorithmic graph theory and perfect graphs[J]. Networks, 1983, 13(2): 304-305.
- [251] Olaizola N, Valenciano F. Dominance of weighted nested split graph networks in connections models[J]. International Journal of Game Theory, 2019, 49(1): 75-96.

- [252] Belik I. The analysis of split graphs in social networks based on the k-cardinality assignment problem[J]. International Journal of Network Science, 2016, 1(1): 53-62.
- [253] Kim S, Thapa I, Lu G, et al. A systems biology approach for modeling microbiomes using split graphs[C]. Proceedings of the 2017 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2017, Kansas City, Missouri, USA, 2017: 2062-2068.
- [254] Blauth J, Nägele M. An improved approximation guarantee for prize-collecting TSP[C]. Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, Florida, USA, 2023: 1848-1861.
- [255] Pedrosa L L C, Rosado H K K. A 2-approximation for the k-prize-collecting steiner tree problem[C]. Proceedings of the 14th Latin American Symposium on Theoretical Informatics, LATIN 2020, São Paulo, Brazil, 2020: 76-88.
- [256] Pedrosa L L C, Rosado H K K. A 2-approximation for the k -prize-collecting steiner tree problem[J]. Algorithmica, 2022, 84(12): 3522-3558.
- [257] Fukunaga T. Spider covers for prize-collecting network activation problem[C]. Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, California, USA, 2015: 9-24.
- [258] Fukunaga T. Spider covers for prize-collecting network activation problem[J]. ACM Transactions on Algorithms, 2017, 13(4): 49:1-49:31.
- [259] Bertossi A A. Dominating sets for split and bipartite graphs[J]. Information Processing Letter, 1984, 19(1): 37-40.
- [260] Golumbic M C. Algorithmic graph theory and perfect graphs[M]. Amsterdam: Academic Press, 1980.
- [261] Müller H. Hamiltonian circuits in chordal bipartite graphs[J]. Discrete Mathematics, 1996, 156(1-3): 291-298.
- [262] Golovach P A, Heggernes P, Kratsch D, et al. An exact algorithm for subset feedback vertex set on chordal graphs[C]. Proceedings of the 7th International Symposium of Parameterized and Exact Computation, IPEC 2012, Ljubljana, Slovenia, 2012: 85-96.
- [263] Golovach P A, Heggernes P, Kratsch D, et al. Subset feedback vertex sets in chordal graphs[J]. Journal of Discrete Algorithms, 2014, 26: 7-15.
- [264] Lovász L, Plummer M D. Matching theory[M]. London: Elsevier Science Ltd., 1986.
- [265] Chen J, Kanj I A. Constrained minimum vertex cover in bipartite graphs: complexity and parameterized algorithms[J]. Journal of Computer and System Sciences, 2003, 67(4): 833-847.

- [266] Gramm J, Guo J, Hüffner F, et al. Data reduction and exact algorithms for clique cover[J]. ACM Journal of Experimental Algorithms, 2008, 13.
- [267] Halldórsson M M, Losievskaja E. Independent sets in bounded-degree hypergraphs[C]. Proceedings of the 10th International Workshop on Algorithms and Data Structures, WADS 2007, Halifax, Canada, 2007: 263-274.
- [268] Halldórsson M M, Losievskaja E. Independent sets in bounded-degree hypergraphs[J]. Discrete Applied Mathematics, 2009, 157(8): 1773-1786.
- [269] Chekuri C, Xu C. Computing minimum cuts in hypergraphs[C]. Proceedings of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, 2017: 1085-1100.
- [270] Chekuri C, Xu C. Minimum cuts and sparsification in hypergraphs[J]. SIAM Journal on Computing, 2018, 47(6): 2118-2156.
- [271] Xu C. Cuts and connectivity in graphs and hypergraphs[D]. University of Illinois Urbana-Champaign, USA, 2018.
- [272] Fox K, Panigrahi D, Zhang F. Minimum cut and minimum k -cut in hypergraphs via branching contractions[C]. Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, 2019: 881-896.
- [273] Fox K, Panigrahi D, Zhang F. Minimum cut and minimum k -cut in hypergraphs via branching contractions[J]. ACM Transactions on Algorithms, 2023, 19(2): 13:1-13:22.
- [274] Grötschel M, Lovász L, Schrijver A. The ellipsoid method and its consequences in combinatorial optimization[J]. Combinatorica, 1981, 1(2): 169-197.
- [275] Husić E, Thomassé S, Trotignon N. The independent set problem is FPT for even-hole-free graphs[C]. Proceedings of the 14th International Symposium of Parameterized and Exact Computation, IPEC 2019, Dagstuhl, Germany, 2019: 21:1-21:12.
- [276] Berge C. Graphes et hypergraphes[M]. Malakoff, Hauts-de-Seine: Dunod, 1970.
- [277] Berge C. Graphs and hypergraphs[M]. London: Elsevier Science Ltd., 1973.
- [278] Lovász L. Normal hypergraphs and the perfect graph conjecture[J]. Discrete Mathematics, 1972, 2(3): 253-267.
- [279] Lovász L. Normal hypergraphs and the perfect graph conjecture[J]. Discrete Mathematics, 2006, 306(10-11): 867-875.
- [280] Lovász L. A characterization of perfect graphs[J]. Journal of Combinatorial Theory, Series B, 1972, 13(2): 95 -98.

- [281] Chudnovsky M, Robertson N, Seymour P, et al. A characterization of perfect graphs[J]. *Annals of Mathematics*, 2006, 164(1): 51 -22.
- [282] Dirac G A. On rigid circuit graphs[J]. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 1961, 25(1): 71-76.
- [283] Lekkerkerker C G, Boland J C. Representation of a finite graph by a set of intervals on the real line[J]. *Fundamenta Mathematicae*, 1962, 51(1): 45-64.
- [284] Fulkerson D, Gross O. Incidence matrices and interval graphs[J]. *Pacific Journal of Mathematics*, 1965, 15(3): 835-855.
- [285] Gavril F. The intersection graphs of subtrees in trees are exactly the chordal graphs[J]. *Journal of Combinatorial Theory, Series B*, 1974, 16(1): 47-56.
- [286] Emtander E. A class of hypergraphs that generalizes chordal graphs[J]. *Mathematica Scandinavica*, 2010, 106(1): 50-66.
- [287] Agnarsson G. On chordal graphs and their chromatic polynomials[J]. *Mathematica Scandinavica*, 2003, 93(2): 240 246.
- [288] Heggernes P, Mancini F, Papadopoulos C, et al. Strongly chordal and chordal bipartite graphs are sandwich monotone[J]. *Journal of Combinatorial Optimization*, 2011, 22(3): 438-456.
- [289] Ekim T, Shalom M, Seker O. The complexity of subtree intersection representation of chordal graphs and linear time chordal graph generation[J]. *Journal of Combinatorial Optimization*, 2021, 41(3): 710-735.
- [290] Beeri C, Fagin R, Maier D, et al. On the desirability of acyclic database schemes[J]. *Journal of ACM*, 1983, 30(3): 479-513.
- [291] Gysel R, Gusfield D. Extensions and improvements to the chordal graph approach to the multi-state perfect phylogeny problem[C]. *Proceedings of the 6th International Symposium Bioinformatics Research and Applications, ISBRA 2010, Storrs, Connecticut, USA*, 2010: 52-60.
- [292] Gysel R, Gusfield D. Extensions and improvements to the chordal graph approach to the multistate perfect phylogeny problem[J]. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2011, 8(4): 912-917.
- [293] Corander J, Janhunen T, Rintanen J, et al. Learning chordal markov networks by constraint satisfaction[C]. *Advances in Neural Information Processing Systems 26: Annual Conference on Neural Information Processing Systems 2013, NeurIPS 2013, Lake Tahoe, Nevada, USA*, 2013: 1349-1357.

- [294] Kangas K, Koivisto M, Niinimäki T M. Learning chordal markov networks by dynamic programming[C]. Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, NeurIPS 2014, Montreal, Quebec, Canada, 2014: 2357-2365.
- [295] Rantanen K, Hyttinen A, Järvisalo M. Learning chordal markov networks via branch and bound[C]. Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, NeurIPS 2017, Long Beach, California, USA, 2017: 1847-1857.
- [296] Vandenberge L, Andersen M S. Chordal graphs and semidefinite optimization[J]. Foundations and Trends in Optimization, 2015, 1(4): 241-433.
- [297] Hack S, Goos G. Optimal register allocation for ssa-form programs in polynomial time[J]. Information Processing Letter, 2006, 98(4): 150-155.
- [298] Bern M W, Lawler E L, Wong A L. Linear-time computation of optimal subgraphs of decomposable graphs[J]. Journal of Algorithms, 1987, 8(2): 216-235.
- [299] Bodlaender H L. Dynamic programming on graphs with bounded treewidth[C]. AProceedings of the 15th International Colloquium on Automata, Languages, and Programming, ICALP 1988, Tampere, Finland, 1988: 105-118.
- [300] Arnborg S, Proskurowski A. Linear time algorithms for NP-hard problems restricted to partial k -trees[J]. Discrete applied mathematics, 1989, 23(1): 11-24.
- [301] Konrad C, Zamaraev V. Distributed minimum vertex coloring and maximum independent set in chordal graphs[C]. Proceedings of the 44th International Symposium on Mathematical Foundations of Computer Science, MFCS 2019, Aachen, Germany, 2019: 21:1-21:15.
- [302] Konrad C, Zamaraev V. Distributed minimum vertex coloring and maximum independent set in chordal graphs[J]. Theoretical Computer Science, 2022, 922: 486-502.
- [303] Papadopoulos C, Tzimas S. Computing a minimum subset feedback vertex set on chordal graphs parameterized by leafage[C]. Proceedings of the 33rd International Workshop on Combinatorial Algorithms, IWOCA 2022, Trier, Germany, 2022: 466-479.
- [304] Buneman P. A characterisation of rigid circuit graphs[J]. Discrete Mathematics, 1974, 9(3): 205-212.
- [305] Walter J R. Representations of rigid cycle graphs[D]. Wayne State University, 1972.
- [306] Rose D J, Tarjan R E, Lueker G S. Algorithmic aspects of vertex elimination on graphs[J]. SIAM Journal on Computing, 1976, 5(2): 266-283.

- [307] Poljak S. A note on stable sets and colorings of graphs[J]. *Commentationes Mathematicae niversitatis Carolinae*, 2017, 15(2): 204-221.
- [308] Munaro A. On line graphs of subcubic triangle-free graphs[J]. *Discrete Mathematics*, 2017, 340(6): 1210-1226.
- [309] Paesani G, Paulusma D, Rzazewski P. Classifying subset feedback vertex set for H -free graphs[C]. *Proceedings of the 48th International Workshop on Graph-Theoretic Concepts in Computer Science, WG 2022*, Tübingen, Germany, 2022: 412-424.
- [310] Feghali C, Johnson M, Paesani G, et al. On cycle transversals and their connected variants in the absence of a small linear forest[C]. *Proceedings of the 22nd International Symposium on Fundamentals of Computation Theory, FCT 2019*, Copenhagen, Denmark, 2019: 258-273.
- [311] Dabrowski K K, Feghali C, Johnson M, et al. On cycle transversals and their connected variants in the absence of a small linear forest[J]. *Algorithmica*, 2020, 82(10): 2841-2866.
- [312] Abrishami T, Chudnovsky M, Pilipczuk M, et al. Induced subgraphs of bounded treewidth and the container method[C]. *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021*, Alexandria, Virginia, USA, 2021: 1948-1964.
- [313] Paesani G, Paulusma D, Rzazewski P. Feedback vertex set and even cycle transversal for h -free graphs: Finding large block graphs[C]. *Proceedings of the 46th International Symposium on Mathematical Foundations of Computer Science, MFCS 2021*, Tallinn, Estonia, 2021: 82:1-82:14.
- [314] Paesani G, Paulusma D, Rzazewski P. Feedback vertex set and even cycle transversal for h -free graphs: Finding large block graphs[J]. *SIAM Journal on Discrete Mathematics*, 2022, 36(4): 2453-2472.

附录 A 全文涉及的计算问题定义

本附录列举了正文中提及的所有图论、组合优化或可满足性问题的定义。

四类反馈集问题

在定义中，通过下划线突出强调各版本之间的不同之处。

反馈点集问题 (Feedback Vertex Set, FVS)

输入: 一个无向图 \mathbf{G} 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在一个大小不超过 k 的反馈点集 $S \subseteq V(\mathbf{G})$ ，即 $\mathbf{G} - S$ 无环。

反馈边集问题 (Feedback Edge Set, FES)

输入: 一个无向图 \mathbf{G} 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在一个大小不超过 k 的反馈边集 $S \subseteq E(\mathbf{G})$ ，即 $\mathbf{G} - S$ 无环。

有向反馈点集问题 (Directed Feedback Vertex Set, DFVS)

输入: 一个有向图 \mathbf{D} 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在一个大小不超过 k 的反馈点集 $S \subseteq V(\mathbf{D})$ ，即 $\mathbf{D} - S$ 无环。

有向反馈边集问题 (Directed Feedback Arc Set, DFAS)

输入: 一个有向图 \mathbf{D} 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在一个大小不超过 k 的反馈边集 $S \subseteq E(\mathbf{D})$ ，即 $\mathbf{D} - S$ 无环。

十二类子集反馈集问题

在定义中，通过下划线突出强调各版本之间的不同之处。

子集反馈点集问题 (Subset Feedback Vertex Set, SFVS)

输入: 一个无向图 \mathbf{G} ，一个关键点集 $T \subseteq V(\mathbf{G})$ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在一个大小不超过 k 的 T -反馈点集 $S \subseteq V(\mathbf{G})$ ，即 $\mathbf{G} - S$ 无经过 T 中元素的环。

限制版子集反馈点集问题 (Restricted Subset Feedback Vertex Set, R-SFVS)

输入: 一个无向图 \mathbf{G} ，一个关键点集 $T \subseteq V(\mathbf{G})$ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在一个大小不超过 k 的 T -反馈点集 $S \subseteq V(\mathbf{G})$ ，即 $\mathbf{G} - S$ 无经过 T 中元素的环。

子集反馈边集问题 (Subset Feedback Edge Set, SFES)

输入: 一个无向图 \mathbf{G} , 一个关键点集 $T \subseteq V(\mathbf{G})$ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在一个大小不超过 k 的 T -反馈边集 $S \subseteq E(\mathbf{G})$, 即 $\mathbf{G} - S$ 无经过 T 中元素的环。

边子集反馈点集问题 (Subset Edge Feedback Vertex Set, ESFVS)

输入: 一个无向图 \mathbf{G} , 一个关键边集 $T \subseteq E(\mathbf{G})$ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在一个大小不超过 k 的 T -反馈点集 $S \subseteq V(\mathbf{G})$, 即 $\mathbf{G} - S$ 无经过 T 中元素的环。

边子集反馈边集问题 (Subset Edge Feedback Edge Set, ESFES)

输入: 一个无向图 \mathbf{G} , 一个关键边集 $T \subseteq E(\mathbf{G})$ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在一个大小不超过 k 的 T -反馈边集 $S \subseteq E(\mathbf{G})$, 即 $\mathbf{G} - S$ 无经过 T 中元素的环。

限制版边子集反馈边集问题 (Restricted Edge Subset Feedback Edge Set, R-ESFES)

输入: 一个无向图 \mathbf{G} , 一个关键点集 $T \subseteq V(\mathbf{G})$ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在一个大小不超过 k 的 T -反馈边集 $S \subseteq E(\mathbf{G}) \setminus T$, 即 $\mathbf{G} - S$ 无经过 T 中元素的环。

有向子集反馈点集问题 (Directed Subset Feedback Vertex Set, DSFVS)

输入: 一个有向图 \mathbf{D} , 一个关键点集 $T \subseteq V(\mathbf{D})$ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在一个大小不超过 k 的 T -反馈点集 $S \subseteq V(\mathbf{D})$, 即 $\mathbf{D} - S$ 无经过 T 中元素的环。

限制版有向子集反馈点集问题 (Restricted Directed Subset Feedback Vertex Set, R-DSFVS)

输入: 一个有向图 \mathbf{D} , 一个关键点集 $T \subseteq V(\mathbf{D})$ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在一个大小不超过 k 的 T -反馈点集 $S \subseteq V(\mathbf{D}) \setminus T$, 即 $\mathbf{D} - S$ 无经过 T 中元素的环。

有向子集反馈边集问题 (Directed Subset Feedback Arc Set, DSFAS)

输入: 一个有向图 \mathbf{D} , 一个关键点集 $T \subseteq V(\mathbf{D})$ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在一个大小不超过 k 的 T -反馈边集 $S \subseteq E(\mathbf{D})$, 即 $\mathbf{D} - S$ 无经过 T 中元素的环。

有向边子集反馈点集问题 (Directed Subset Arc Feedback Vertex Set, DASFVS)

输入: 一个有向图 \mathbf{D} , 一个关键边集 $T \subseteq E(\mathbf{D})$ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在一个大小不超过 k 的 T -反馈点集 $S \subseteq V(\mathbf{D})$, 即 $\mathbf{D} - S$ 无经过 T 中元素的环。

有向边子集反馈边集问题 (Directed Subset Arc Feedback Arc Set, DASFAS)

输入: 一个有向图 \mathbf{D} , 一个关键边集 $T \subseteq E(\mathbf{D})$ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在一个大小不超过 k 的 T -反馈边集 $S \subseteq E(\mathbf{D})$, 即 $\mathbf{D} - S$ 无经过 T 中元素的环。

限制版有向边子集反馈边集问题 (Restricted Directed Arc Subset Feedback Arc Set, R-DASFAS)

输入: 一个有向图 \mathbf{D} , 一个关键点集 $T \subseteq V(\mathbf{D})$ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在一个大小不超过 k 的 T -反馈边集 $S \subseteq E(\mathbf{D}) \setminus T$, 即 $\mathbf{D} - S$ 无经过 T 中元素的环。

本文中研究的核心问题

平面图上反馈点集问题 (FVS in Planar Graphs, FVS(P))

输入: 一个平面图 \mathbf{G} 和一个自然数 k 。

输出: 判断平面图 \mathbf{G} 是否存在一个大小不超过 k 的反馈点集 $S \subseteq V(\mathbf{G})$ 。

平面图上有向反馈点集问题 (DFVS in Planar Graphs, DFVS(P))

输入: 一个平面有向图 \mathbf{D} 和一个自然数 k 。

输出: 判断平面有向图 \mathbf{D} 是否存在一个大小不超过 k 的反馈点集 $S \subseteq V(\mathbf{D})$ 。

平面图上有向反馈边集问题 (DFAS in Planar Graphs, DFAS(P))

输入: 一个平面有向图 \mathbf{D} 和一个自然数 k 。

输出: 判断平面有向图 \mathbf{D} 是否存在一个大小不超过 k 的反馈边集 $S \subseteq E(\mathbf{D})$ 。

竞赛图上非限制版子集反馈点集问题 (DSFVS in Tournaments, DSFVS(T))

输入: 一个竞赛图 \mathbf{T} , 一个大小为 τ 的关键点集 $T \subseteq V(\mathbf{T})$ 和一个自然数 k 。

输出: 判断竞赛图 \mathbf{T} 是否存在一个大小不超过 k 的 T -反馈点集 $S \subseteq V(\mathbf{T})$ 。

竞赛图上限制版子集反馈点集问题 (R-DSFVS in Tournaments, R-DSFVS(T))

输入: 一个竞赛图 \mathbf{T} , 一个大小为 τ 的关键点集 $T \subseteq V(\mathbf{T})$ 和一个自然数 k 。

输出: 判断竞赛图 \mathbf{T} 是否存在一个大小不超过 k 的 T -反馈点集 $S \subseteq V(\mathbf{T}) \setminus T$ 。

竞赛图上分离版子集反馈点集问题 (Disjoint-DSFVS in Tournaments, Disjoint-DSFVS(\mathbf{T}))

输入: 一个竞赛图 \mathbf{T} , 一个大小为 τ 的关键点集 $T \subseteq V(\mathbf{T})$, 一个不可删点集 $U \subseteq V(\mathbf{T})$ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断竞赛图 \mathbf{T} 是否存在一个大小不超过 k 的 T -反馈点集 $S \subseteq V(\mathbf{T}) \setminus U$ 。

分裂图上广义子集反馈点集问题 (Generalized-Subset Feedback Vertex Set in Split Graphs, Generalized-SFVS(\mathbf{S}))

输入: 一个分裂图 \mathbf{G} , 一个关键点集 $T \subseteq V(\mathbf{G})$, 一个标记边集 $M \subseteq E(\mathbf{G})$, 一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在一个大小不超过 k 的 T -反馈点集 S , 使得 M 中的任何一条边都至少存在一个端点在 S 中。

分裂图上非限制版子集反馈点集问题 (SFVS in Split Graphs, SFVS(\mathbf{S}))

输入: 一个分裂图 \mathbf{G} , 一个大小为 τ 的关键点集 $T \subseteq V(\mathbf{T})$ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断分裂图 \mathbf{G} 是否存在一个大小不超过 k 的 T -反馈点集 $S \subseteq V(\mathbf{G})$ 。

分裂图上限制版子集反馈点集问题 (R-SFVS in Split Graphs, R-SFVS(\mathbf{S}))

输入: 一个分裂图 \mathbf{T} , 一个大小为 τ 的关键点集 $T \subseteq V(\mathbf{T})$ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断分裂图 \mathbf{T} 是否存在一个大小不超过 k 的 T -反馈点集 $S \subseteq V(\mathbf{G}) \setminus T$ 。

超图上独立集问题 (Maximum Independent Set on Hypergraphs, MIS(\mathbf{H}))

输入: 一个超图 \mathbf{H} 和一个自然数 $k \in \mathbb{N}$.

输出: 判断是否存在大小不超过 k 的独立集 $S \subseteq V(\mathbf{H})$ 。

超图上带惩罚的独立集问题 (Prize Collecting Maximum Independent Set on Hypergraphs, PC-MIS(\mathbf{H}))

输入: 一个超图 \mathbf{H} 和一个自然数 $p \in \mathbb{N}$.

输出: 判断是否存在价值不超过 p 的点集 $S \subseteq V(\mathbf{H})$ 。

弦图上广义子集反馈点集问题 (Generalized-Subset Feedback Vertex Set in Chordal Graphs, Generalized-SFVS(\mathbf{C}))

输入: 一个弦图 \mathbf{G} , 一个关键点集 $T \subseteq V(\mathbf{G})$, 一个标记边集 $M \subseteq E(\mathbf{G})$, 一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在一个大小不超过 k 的 T -反馈点集 S , 使得 M 中的任何一条边都至少存在一个端点在 S 中。

弦图上非限制版子集反馈点集问题 (SFVS in Chordal Graphs, SFVS(C))

输入: 一个弦图 \mathbf{G} , 一个大小为 τ 的关键点集 $T \subseteq V(\mathbf{G})$ 和一个自然数 k 。

输出: 判断弦图 \mathbf{G} 是否存在一个大小不超过 k 的 T -反馈点集 $S \subseteq V(\mathbf{G})$ 。

弦图上限制版子集反馈点集问题 (R-SFVS in Chordal GraphsR-SFVS(C))

输入: 一个弦图 \mathbf{G} , 一个大小为 τ 的关键点集 $T \subseteq V(\mathbf{G})$ 和一个自然数 k 。

输出: 判断弦图 \mathbf{G} 是否存在一个大小不超过 k 的 T -反馈点集 $S \subseteq V(\mathbf{G})$ 且 $T \subseteq S$ 。

Karp 的 NP-完全问题

除了前文定义的有向反馈点集和有向反馈边集外, 本节列举全文涉及的 Karp 的 NP-完全问题。

布尔可满足性问题 (Satisfiability, SAT)

输入: 一个合取范式 φ 。

输出: 判断是否存在赋值 \mathbf{x} , 使得 $\varphi(\mathbf{x})$ 为真。

d -布尔可满足性问题 (d -Satisfiability, d -SAT)

输入: 一个合取范式 φ , 其中每一个子句至多包含 d 个文字。

输出: 判断是否存在赋值 \mathbf{x} , 使得 $\varphi(\mathbf{x})$ 为真。

顶点覆盖问题 (Vertex Cover, VC)

输入: 一个无向图 \mathbf{G} 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在一个大小不超过 k 的点覆盖 $S \subseteq V(\mathbf{G})$, 即 $\mathbf{G} - S$ 无边。

独立集问题 (Maximum Independent Set, MIS)

输入: 一个无向图 \mathbf{G} 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断图 \mathbf{G} 中存在一个大小至少为 k 的独立集。

团问题 (Clique)

输入: 一个无向图 \mathbf{G} 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断图 \mathbf{G} 中存在一个大小至少为 k 的团。

集合覆盖 (d -Set Cover)

输入: 一个元素集合 U , 一个子集族 $\mathcal{F} \subseteq 2^U$ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在不超过 k 个子集 $S_1, S_2, \dots, S_{k'} \subseteq \mathcal{F}$ ($k' \leq k$), 使得它们的并集为 U 。

碰撞集问题 (Hitting Set)

输入: 一个元素集合 U , 一个子集族 $\mathcal{F} \subseteq 2^U$ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在大小不超过 k 的集合 $S \subseteq U$, 使得 \mathcal{F} 中任意子集与集合 S 的交非空。

d -碰撞集问题 (d -Hitting Set)

输入: 一个元素集合 U , 一个子集大小不超过 d 的子集族 $\mathcal{F} \subseteq 2^U$ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在大小不超过 k 的集合 $S \subseteq U$, 使得 \mathcal{F} 中任意子集与集合 S 的交非空。

旅行商问题 (Travelling Salesman Problem)

输入: 一个无向图 \mathbf{G} 。

输出: 判断图 \mathbf{G} 中是否一条包含所有点的环。

隐式碰撞集问题

除了前文定义的反馈相关问题外, 本节列举全文提及的各类隐式碰撞集问题。主要包括奇圈交叉相关问题和割问题, 其中子集奇圈交叉问题仅列举了非限制版, 割问题近列举了限制版。

奇圈交叉问题 (Odd Cycle Transversal, OCT)

输入: 一个无向图 \mathbf{G} 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在大小不超过 k 的点集 $S \subseteq V(\mathbf{G})$, 使得 $\mathbf{G} - S$ 是二部图。

子集奇圈交叉问题 (Subset Odd Cycle Transversal, SOCT)

输入: 一个无向图 \mathbf{G} , 一个关键点集 $T \subseteq V(\mathbf{G})$ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在大小不超过 k 的点集 $S \subseteq V(\mathbf{G})$, 使得 $\mathbf{G} - S$ 中没有包含关键点的奇圈。

有向奇圈交叉问题 (Directed Odd Cycle Transversal, DOCT)

输入: 一个有向图 \mathbf{D} 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在大小不超过 k 的点集 $S \subseteq V(\mathbf{D})$, 使得 $\mathbf{D} - S$ 中无有向奇圈。

有向子集奇圈交叉问题 (Directed Subset Odd Cycle Transversal, DSOCT)

输入: 一个有向图 \mathbf{D} , 一个关键点集 $T \subseteq V(\mathbf{D})$ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在大小不超过 k 的点集 $S \subseteq V(\mathbf{D})$, 使得 $\mathbf{D} - S$ 中没有包含关键点的有向奇圈。

节点多路割问题 (Node Multiway Cut, NMC)

输入: 一个无向图 \mathbf{G} , 一个关键点集 $T \subseteq V(\mathbf{G})$ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在大小不超过 k 的点割 $S \subseteq V(\mathbf{G}) \setminus T$, 使得 T 中的关键点在 $\mathbf{G} - S$ 中两两不连通。

节点多割问题 (Node Multicut, NM)

输入: 一个无向图 \mathbf{G} , 一个关键点对集 $T \subseteq V(\mathbf{G}) \times V(\mathbf{G})$ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在大小不超过 k 的点割 $S \subseteq V(\mathbf{G}) \setminus \{s, t : (s, t) \in T\}$, 使得 T 中的关键点对在 $\mathbf{G} - S$ 中不连通。

有向节点多路割问题 (Directed Node Multiway Cut, DNMC)

输入: 一个有向图 \mathbf{D} , 一个关键点集 $T \subseteq V(\mathbf{D})$ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在大小不超过 k 的点割 $S \subseteq V(\mathbf{D}) \setminus T$, 使得 T 中的关键点在 $\mathbf{D} - S$ 中两两不连通。

有向节点多割问题 (Directed Node Multicut, DNM)

输入: 一个有向图 \mathbf{D} , 一个关键点对集 $T \subseteq V(\mathbf{D}) \times V(\mathbf{D})$ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在大小不超过 k 的点割 $S \subseteq V(\mathbf{D}) \setminus \{s, t : (s, t) \in T\}$, 使得 T 中任意一对关键点 (s, t) 在 $\mathbf{D} - S$ 中都没有从 s 到 t 的有向路。

隐式 3-碰撞集相关问题

除前文定义的反馈相关问题外, 本节列举全文提及的各类隐式 3-碰撞集问题。

三角形顶点删除问题 (Triangle-free Vertex Deletion)

输入: 一个无向图 \mathbf{G} 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在大小不超过 k 的点集 $S \subseteq V(\mathbf{G})$, 使得 $\mathbf{G} - S$ 无三角形。

3-长路顶点覆盖问题 (3-Path Vertex Cover Problem)

输入: 一个无向图 \mathbf{G} 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在大小不超过 k 的点集 $S \subseteq V(\mathbf{G})$, 使得 $\mathbf{G} - S$ 没有长度为 3 的路。

聚类图顶点删除问题 (Cluster Vertex Deletion)

输入: 一个无向图 \mathbf{G} 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在大小不超过 k 的点集 $S \subseteq V(\mathbf{G})$, 使得 $\mathbf{G} - S$ 是一些团的并。

竞赛图上有向反馈点集问题 (DFVS in Tournaments, DFVST)

输入: 一个竞赛图 \mathbf{D} 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在大小不超过 k 的点集 $S \subseteq V(\mathbf{D})$, 使得 $\mathbf{G} - S$ 无环。

参数复杂性类相关问题

本节给出全文涉及用于定义参数复杂性类的计算问题, 其中会涉及到电路的深度和纬度的概念。电路的深度是指, 任意输入门到输出门的电路上门电路的最大数量。电路的纬度是指任意输入门到输出门的电路上包含的多扇入电路门数量, 多扇入门是指输入变量至少为 3 的门电路。

d -深度 t -纬度电路上的带权布尔可满足性问题 (Weighted-SAT(d -Depth t -Weft CIRC))

输入: 一个深度为 d 纬度为 t 的电路 ψ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在权重为 k 的赋值 \mathbf{x} , 使得 $\psi(\mathbf{x})$ 为真。

电路上的带权布尔可满足性问题 (Weighted-SAT(CIRC))

输入: 一个电路 ψ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在权重为 k 的赋值 \mathbf{x} , 使得 $\psi(\mathbf{x})$ 为真。

带权布尔可满足性问题 (Weighted-SAT(PROP))

输入: 一个布尔式 φ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在权重为 k 的赋值 \mathbf{x} , 使得 $\varphi(\mathbf{x})$ 为真。

其他问题

图同构问题 (Isomorphism)

输入: 一个无向图 \mathbf{G} 和一个无向图 \mathbf{G}' 。

输出: 判断是否图 \mathbf{G} 和 \mathbf{G}' 是同构。

支配集问题 (Dominating Set)

输入: 一个无向图 \mathbf{G} 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在大小不超过 k 的点集 $S \subseteq V(\mathbf{G})$, 使得 $N_{\mathbf{G}}[S] = V(\mathbf{G})$ 。

边的团覆盖问题 (Edge Clique Cover)

输入: 一个无向图 \mathbf{G} 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在大小不超过 k 的团族 \mathcal{C} , 使得每一条边 $e \in E(\mathbf{G})$ 都至少出现在 \mathcal{C} 中的一个团中。

匹配问题 (Matching)

输入: 一个无向图 \mathbf{G} 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断图 \mathbf{G} 中是否存在一个大小至少为 k 的匹配。

最小边割问题 (Minimum Cut)

输入: 一个无向图 \mathbf{G} , 一对点 $(s, t) \in V(\mathbf{G}) \times V(\mathbf{G})$ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在大小不超过 k 的边割 $S \subseteq E(\mathbf{G})$, 使得在 $\mathbf{G} - S$ 中 s 和 t 不连通。

生成树问题 (Spanning Tree)

输入: 一个无向连通图 \mathbf{G} 。

输出: 找到一个包含 \mathbf{G} 中所有顶点的子图, 它是一棵树。

 d -集合覆盖 (Set Cover)

输入: 一个元素集合 U , 一个子集族 $\mathcal{F} \subseteq 2^U$ 和一个自然数 $k \in \mathbb{N}$, 其中 \mathcal{F} 包含的子集大小至多为 d 。

输出: 判断是否存在不超过 k 个子集 $S_1, S_2, \dots, S_{k'} \subseteq \mathcal{F}$ ($k' \leq k$), 使得它们的并集为 U 。

 Π 上顶点删除问题 (Vertex Deletion (Π))

输入: 一个无向或有向图 \mathbf{G} 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在一个大小不超过 k 的点集 $S \subseteq V(\mathbf{G})$, 使得 $\mathbf{G} - S \in \Pi$ 。

压缩版本 Π 上顶点删除问题 (Compression-Vertex Deletion (Π))

输入: 一个无向或有向图 \mathbf{G} , 一个点集 \bar{S} 使得 $\mathbf{G} - \bar{S} \in \Pi$ 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在一个大小不超过 k 的点集 $S \subseteq V(\mathbf{G})$, 使得 $\mathbf{G} - S \in \Pi$ 。

分离版本 Π 上顶点删除问题 (Disjoint-Vertex Deletion (Π))

输入: 一个无向或有向图 \mathbf{G} , 一个不可删点集 U 和一个自然数 $k \in \mathbb{N}$ 。

输出: 判断是否存在一个大小不超过 k 的点集 $S \subseteq V(\mathbf{G}) \setminus U$, 使得 $\mathbf{G} - S \in \Pi$ 。