Computer Science and Application 计算机科学与应用, 2025, 15(9), 52-62
Published Online September 2025 in Hans. https://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2025.159223

Hans汉斯

图约减技术在松弛团问题中的应用与研究

李明格*，张玉成，张耕硕，王佳艺

西京学院计算机学院，陕西 西安

收稿日期：2025年7月25日；录用日期：2025年8月23日；发布日期：2025年9月1日

摘　要

松弛团问题是图论中一个关键问题，旨在寻找图中满足相应约束条件的松弛团模型。伴随着数据规模的日益增长，如何有效地对松弛团问题进行约减，以降低问题的求解难度显得尤为重要。图约减技术由此应运而生，本文主要研究了图约减技术在松弛团中的应用与研究。具体来说，本文以经典的最大k-club问题和最大k-defective clique问题为例，根据各自问题的结构特点，设计实现了高效的约减算法G_NP和G_KD。Network Data Repository数据集以及Facebook社交网络数据集上的实验结果表明：所提出的G_NP和G_KD算法能够有效地对不同参数要求下的最大k-club问题和最大k-defective clique问题进行求解。特别是当参数k较小时，对于部分用例，能够约减大约20%的顶点，大幅度地降低了问题后续的求解难度。而当参数k较大时，所设计的上下界函数较为宽松，导致约减的效果有所下降，但即便如此，依旧能够约减掉一部分的顶点。

关键词

松弛团，图约减技术，启发式策略，k-Defective Clique，k-Club

The Application and Research of Graph Reduction Techniques in the Relaxation Group Problem

Mingge Li*, Yucheng Zhang, Gengshuo Zhang, Jiayi Wang

School of Computer Science, Xijing University, Xi'an Shaanxi

Received: Jul. 25th, 2025; accepted: Aug. 23rd, 2025; published: Sep. 1st, 2025

Abstract

The relaxed clique problem is a key problem in graph theory, which aims to find a relaxed clique

*通讯作者。

文章引用: 李明格, 张玉成, 张耕硕, 王佳艺. 图约减技术在松弛团问题中的应用与研究[J]. 计算机科学与应用, 2025, 15(9): 52-62. DOI: 10.12677/csa.2025.159223

model that satisfies the corresponding constraints in the graph. With the increasing size of data, it is particularly important to effectively reduce the relaxed clique problem to reduce the difficulty of solving the problem. Graph reduction technology has emerged as a result. This paper mainly studies the application and research of graph reduction technology in relaxed cliques. Specifically, this paper takes the classic maximum k-club problem and the maximum k-defective clique problem as examples, and designs and implements efficient reduction algorithms G_NP and G_KD according to the structural characteristics of each problem. The experimental results on the Network Data Repository dataset and the Facebook social network dataset show that the proposed G_NP and G_KD algorithms can effectively solve the maximum k-club problem and the maximum k-defective clique problem under different parameter requirements. In particular, when the parameter k is small, for some use cases, about 20% of the vertices can be reduced, which greatly reduces the difficulty of solving the problem later. When the parameter k is large, the designed upper and lower bound functions are relatively loose, resulting in a decrease in the reduction effect, but even so, some vertices can still be reduced.

Keywords

Relaxed Clique, Graph Reduction Techniques, Heuristic Strategy, k-Defective Clique, k-Club

1. 绪论

1.1. 引言

在社交网络分析中，完全图(团)是一个重要的概念，它表示网络中的每个个体都与所有其他个体直接相连，从而形成了一个紧密联系的社交圈。然而，在现实世界中，并非所有社区成员都认识彼此，这种不完全的社交关系促使了松弛团模型的产生[1]。松弛团模型放宽了完全图的严格限制，允许图中存在一定数量的缺失边，从而更真实地反映了社交网络的复杂性。根据应用的场景不同，研究人员对不同的团性质进行松弛进而提出了各种松弛团模型。在松弛团模型的众多变体中，k-club 和 k-defective clique 是两种常见的模型，它们通过引入参数 k 来控制图的紧密程度[2]。这些模型在大规模图数据中具有重要的应用价值，但同时也带来了巨大的计算挑战。当前对于松弛团模型的研究主要集中于大规模图用例上，但是数量繁多的顶点和边对相应问题的求解造成了巨大的挑战。因此，如何对大规模的图用例进行降维，以缩减输入图的规模显得尤为重要。为了解决上述问题，本文将以松弛团中重要的两类问题：最大 k-club 问题以及最大 k-defective clique 问题为核心，研究设计针对这两个问题的图约减技术。

1.2. 松弛团问题的定义及研究现状

1.2.1. 常见松弛团模型和相关问题

图约减技术是图论中用于简化问题复杂度的关键手段，其核心是在保留问题关键信息的前提下，通过删除冗余顶点、边或子图，降低图的规模与密度，从而提升后续算法的求解效率。常见的图约减方法包括基于顶点度数的约减、基于子图连通性的约减、基于上下界估计的剪枝等，这些方法广泛应用于团问题、顶点覆盖问题、图染色问题等组合优化问题中[3]。

松弛团问题是团问题的扩展，其核心是通过引入参数松弛“任意两顶点必相连”的严格约束，以适应现实网络的稀疏性与不完全性。从理论上看，松弛团问题的计算复杂性与参数密切相关：多数松弛团

问题(如最大 k-club、最大 k-defective clique)均被证明为 NP 难问题[4]，这意味着在大规模图上直接求解的计算成本极高，因此图约减技术成为提升求解效率的关键支撑。

常见的松弛团模型包括：

k-plex：子图中每个顶点至少与其他 k-1 个顶点相连(即顶点度数不小于|C|-k，其中 C 为子图顶点集)；

k-club：子图中任意两顶点的最短路径长度不超过 k；

k-defective clique：子图的缺失边数不超过 k (与同规模完全图相比)。

这些模型通过不同维度的松弛(如距离、边密度)，为不同场景的社区检测提供了灵活工具，例如 k-club 适用于识别“多步可达”的紧密社区，k-defective clique 适用于允许少量关系缺失的稠密子图挖掘。

1.2.2. 最大 k-Club 问题

2000 年，Bourjolly 等首次提出最大 k-club 问题并设计三种启发式搜索策略。2002 年，该问题被证明具有 NP 难度，同时出现基于整数线性规划(ILP)和分支定界(BB)的早期精确算法[5]，此后多数精确算法均基于这两种技术发展而来。Pajouh 和 Balasundaram 进一步证明 k-club 极大性问题的 NP 难度，提出结合图着色技术的上界估计 BB 算法；Shahinpour 和 Butenko [6]开发变邻域局部搜索算法，改进了 Bourjolly 等人[7]的 BB 算法初始下界。

ILP 领域中，Veremyev 和 Boginskii 提出基于路径枚举的创新模型；Buchanan 和 Salemi 的模型因变量少、效率高受关注；Almeida 和 Carvalho 针对 3-club 问题提供上界分析与 ILP 模型[8]；Pajouh 等则深入研究最大 2-club 问题的多面体结构，在《Operations Research》发表相关 ILP 模型[9]。

启发式算法方面，Shahinpour 和 Butenko 的变邻域局部搜索融合顶点增减等策略；2014 年 Almeida 和 Carvalho 的改进型两阶局部搜索算法显著提升效率[10]。2022 年 Chen 等人引入动态图约减技术，通过生成树保持解一致性，设计分层阈值格局检测策略避免循环[11]，据此开发的 NukCP 算法在多数场景下计算时间更短、解更优，且能在 NDR 数据集上为所有测试案例提供有效解[12]，展现出强实用性。

近年研究中，针对大规模图的分布式求解成为新方向。例如，2023 年 Li 等人基于 MapReduce 框架设计并行约减策略，将 k-club 问题分解为子图任务，在百万级顶点图上实现了高效求解，为最大 k-club 问题的工程应用提供了新思路。

1.2.3. 最大 k-Defective Clique 问题

尽管最大 k-defective clique 问题的研究相对较少，但已有系列重要进展[13]。2013 年，Trukhanov 等通过遗传学特性分析，开发出首个基于俄罗斯套娃搜索策略的完备算法；2018 年，Gschwind 等人引入交替递增验证过程，对该算法进行显著改进。

2020 年，Stozhkov 等基于 Motzkin-Straus 二阶公式[14]提出创新连续平方公式，首次将其应用于 k-defective clique 模型，为问题解决提供新视角。2021 年，Chen 等人设计多种图约减技术，开发出针对该问题的高效分支限界算法 MADEC；2022 年，Gao 等人扩展 MADEC 的约减策略，引入新边约减技术，提出当前最先进的完备算法 KDBB [15]。

此外，Chen 等人结合顶点信息开发混合收益值启发式方法，优化候选顶点选择，并设计新哈希函数与约束条件，构建解禁忌搜索算法 CSTDC [16]。2023 年，Wang 等人针对动态图场景，提出增量式约减策略，通过维护顶点度数与缺失边数的动态变化，实现了动态图中最大 k-defective clique 的高效更新，拓展了该问题的应用场景。

1.3. 相关技术和工具介绍

1.3.1. VMware Workstation

VMware Workstation 是一款桌面虚拟化软件，支持在单一物理机上创建并运行多个独立虚拟机(可安

装不同操作系统)，为 IT 开发、软件测试等场景提供高效测试平台。其虚拟网络模拟、实时快照、共享文件夹等功能简化了实验流程，是本研究的首选虚拟机工具。

1.3.2. Linux 系统

Linux 是开源操作系统内核，基于 GNU GPL 协议分发，支持进程管理、内存调度等核心功能，由全球开发者社区维护。本研究采用 CentOS 发行版，其稳定的命令行环境和高效资源管理能力，适合大规模图数据的计算需求。

1.3.3. C++语言

C++是支持面向对象、过程化及泛型编程的高效语言，兼具底层硬件控制能力和跨平台特性，适合实现图约减算法中的邻接表操作、迭代逻辑等核心功能。其丰富的标准库(如容器、算法)可加速开发，故为本研究的编程语言。

1.4. 主要研究目标与方法

本文的主要目标是探索图约减技术在松弛团问题中的应用，特别是针对最大 k-club 问题和最大 k-defective clique 问题。为了实现这一目标，本文首先回顾了图约减技术的基本原理，然后分别针对最大 k-club 问题和最大 k-defective clique 问题设计了相应的高效约减算法——G_NP 和 G_KD，用于对大规模图进行有效约减。

1.4.1. 算法部分

将根据最大 k-club 问题和最大 k-defective clique 问题的结构特征设计有效的图约减策略，进而降低输入图的复杂度来加速松弛团问题的求解过程。其中涉及到的主要研究内容为设计基于图的连通性、度数、子图结构等方面的约减策略，从而开发高效的算法来解决松弛团问题。这些算法包括基于贪婪策略、启发式方法等技术。

1.4.2. 实验部分

将通过是否调用相关问题的约减方法，观察分析约减方法在这两个问题求解过程中对于输入图规模的影响，进而检验所提出的策略是否能够有效地对最大 k-club 问题以及最大 k-defective 问题进行约减。

2. 约减策略

2.1. 最大 k-Club 问题的约减策略

最大 k-club 问题的约减主要围绕计算图中 k-club 的下界值以及各个顶点的上界值展开。通常来说，任意一个 k-club 的大小就可以作为图的下界值，而每个顶点的 k 阶闭邻居数量便是一个简单的上界值。文献中针对该问题的约减策略主要分为 TRIM 方法和 DRM 方法，两者均是采用上述的上下界计算方式，区别的地方在于各自的维护方法不同，导致计算的效率也有所影响。

2.1.1. TRIM 方法

TRIM 是一种专门针对 k-club 问题设计的约减方法。它基于动态规划的思想，迭代地删除图中的冗余顶点。具体来说，TRIM 方法包括以下步骤：

1) 计算每个顶点的上界(k 阶邻域大小)和下界(启发式估计的最大 k-club 规模)；

2) 迭代删除上界 ≤ 下界的顶点及关联边，并更新其 k 阶邻域内顶点的上界；

3) 重复步骤 2 直至无顶点可删除。

TRIM 方法的优点在于它能够有效地减少搜索空间，但缺点是计算上界的过程可能会非常耗时，尤其

是对于大规模图来说，这可能会导致算法运行时间较长。

2.1.2. DRM 方法

DRM (Dynamic Reduction Method)是 TRIM 方法的一个改进版本，旨在提高约减过程的效率，因此在处理大规模图时更具效率。DRM 方法的核心思想是在删除顶点的同时动态更新上界，而不是重新计算。这减少了不必要的计算开销，并加快了约减过程。DRM 方法的主要步骤如下：

1) 初始计算所有顶点上界，后续通过增量更新替代重新计算；

2) 按顶点度数排序，优先处理高 degree 顶点，若其上限 ≤ 下界则删除，并将受影响的邻域顶点加入删除队列；

3) 基于队列批量处理待删顶点，减少迭代次数。

DRM 方法通过减少上界的重新计算和动态更新上界的方式，显著提高了约减过程的效率。尤其适用于大规模图，因此，本文提出的 G_NP 算法将基于 DRM 方法进行优化改进，实现对于输入图的单次约减，进而初步缩减问题规模，降低后续问题求解难度。

2.2. 最大 k-Defective Clique 问题的约减策略

最大 k-defective clique 问题的约减基于顶点度数与缺失边约束，通过规则化筛选删除不可能属于最优解的顶点和边。核心思路是：若顶点度数小于“当前下界-k”，则必然不属于最大 k-defective clique，可直接删除。在此基础上，衍生出以下关键约减规则：

顶点删除规则：若顶点 v 的度数 + 邻域顶点数 + 可添加顶点数 < 当前下界，则删除 v；若顶点度数 < 允许缺失边数 k，直接删除。

边删除规则：若移除边(u, v)不影响最大 k-defective clique 的搜索(即边的存在与否不改变上界与下界的关系)，则删除该边。

约减过程通过分支定界框架实现：初始化下界后，迭代应用上述规则删除冗余元素，同步更新上下界，直至无法进一步约减。该策略能有效压缩搜索空间，尤其适用于稀疏图场景。

3. 约减算法的设计

3.1. 最大 k-Club 问题的约减算法

3.1.1. NukCP 约减算法

NukCP 是求解最大 k-club 问题的高效局部搜索算法，适用于大规模实例，核心包含两项策略：

动态约减策略(Dynamic Reduction Strategy, DRM)：通过动态更新顶点上界(而非重新计算)减少迭代次数，在简化图上执行计算以提升效率。

分层阈值配置检查策略(Stratified Threshold Configuration Checking, STCC)：针对 k-club 需多级邻域信息的特性，为不同层级邻域设置优先级，结合生成树动态维护邻域信息，解决局部搜索循环问题，提升搜索多样性。

NukCP 通过融合上述策略，有效提高了最大 k-club 问题的求解效率与质量，为大规模图处理提供了可行方案，并为同类问题研究提供了新思路。

3.1.2. G_NP 算法的设计思路

G_NP 算法基于 NukCP 与 DRM 策略优化，聚焦于快速缩减图规模，设计要点包括：

顶点排序：按顶点度数降序排列，优先处理高连通性顶点；

k 阶邻居统计：计算每个顶点的 k 阶邻居数量，作为筛选依据；

约减逻辑：若顶点 k 阶邻居数 ≤lb-1，则判定为冗余并删除；

动态维护：通过数组记录剩余顶点，删除冗余顶点后同步更新其邻域顶点的 k 阶邻居数，确保约减后图结构的一致性。

该算法可简化原始图，且保留寻找最大 k-club 的关键信息，缩减搜索空间并提升效率，这对大规模图至关重要，但因其顶点和边众多，直接搜索会在计算上非常耗时。

3.1.3. G_NP 算法中图的约减过程

算法通过 Reduce GraphInit 函数实现，分为两个核心阶段：

1) 动态计算阶段：遍历顶点并计算 k 阶邻居数，存储于 g_degree_k 数组；

2) 迭代删除阶段：对 k 阶邻居数 ≤lb-1 的顶点，加入删除队列并从剩余顶点集(g_remaining)中移除；删除顶点后，更新其邻域顶点的 g_degree_k 值，若更新后满足删除条件则加入队列，直至无冗余顶点。

该过程通过单次遍历与队列批量处理，减少了重复计算，在大规模图上表现出较高时间效率。

Figure 1. G_NP code section (part one)
图 1. G_NP 代码部分(一)

3.1.4. G_NP 的代码部分

本算法的主要作用是简化图的结构，完成一次约减过程，便于后续的求解及算法更高效地运行，为之后的最大 k-club 问题求解完成预处理部分，其中关键的代码如图 1、图 2 和图 3 所示。

Figure 2. G_NP code section (part two)
图 2. G_NP 代码部分(二)

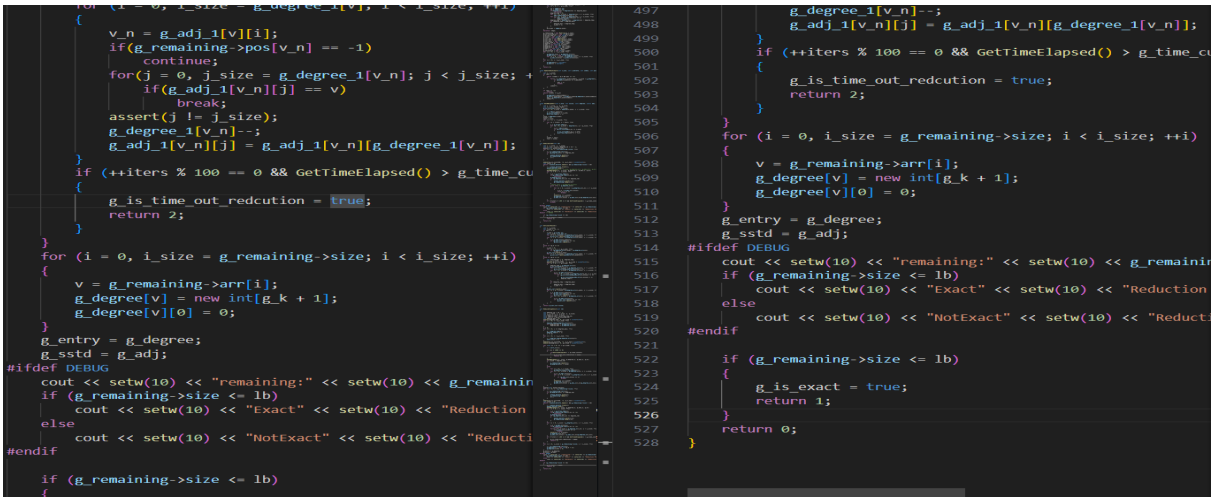

Figure 3. Part (3) of the G_NP code
图 3. G_NP 代码部分(三)

函数 ReduceGraphInit 实现图的约减过程，通过移除冗余顶点简化图结构并保留关键性质，关键执行逻辑包括：

(1) 阈值初始化：设置 degree_lb = lb-1 作为顶点筛选阈值；

(2) k 阶邻居计算：通过广度优先搜索(BFS)计算各顶点 k 阶邻居数，并结合时间限制动态终止计算；

(3) 队列驱动的约减：利用 FIFO 队列批量处理待删顶点，同步更新邻接表与剩余顶点数组，确保约减后图仍保留最大 k-club 的关键信息。

3.2. 最大 k-Defective Clique 问题的约减算法

3.2.1. KDBB 约减算法

KDBB 算法是专为求解最大 k-defective clique 问题设计的精确算法，基于分支定界框架，通过引入新技术提升大规模稀疏图的处理效率。该算法系统枚举候选解并剪枝不可行分支，提出新的上下界估计方法及规约规则以移除冗余顶点和边，减少搜索空间；同时结合顶点度数等启发式信息排序候选顶点，优化搜索方向。KDBB 在大规模稀疏图的 MDCP 问题上表现出显著优势，尤其在效率和性能方面。

3.2.2. G_KD 算法的设计思路

G_KD 算法用于图的预处理约减，旨在移除不可能属于最大 k-defective clique 的顶点和边，以缩减搜索空间。该算法借鉴 KDBB 算法及约减规则，通过以下步骤实现单次约减(为后续求解预处理)：

删除标记初始化：通过 v_del 数组记录待删顶点；

冗余顶点识别：利用 vertex_judge 函数筛选度数 ≤ lb-k 的顶点(此类顶点不可能属于最大 k-defective clique)；

图结构更新：建立 v_map 数组映射剩余顶点的新索引，重建邻接表(t_graph_tbl)与边集(t_edge)，仅保留剩余顶点间的边；

内存优化：释放原图内存，更新顶点数、边数及指针至新图结构。

3.2.3. G_KD 算法中图的约减过程

1) 顶点和边的删除：通过队列批量处理待删顶点，删除时同步更新邻接顶点的度数与剩余边数；

2) 更新图的拓扑结构：基于 v_map 映射剩余顶点，重建邻接表与边集，自动剔除与删除顶点相关的

边；

3) 最终更新：替换原图指针为新图结构，确保后续求解基于简化后的图执行。

```
void first_reduction() {
    v_del  = (bool*)malloc(num_vertex * sizeof(bool));
    for(int i = 0; i < num_vertex; i++) v_del[i] = false;

    queue <int> q;
    int t_num_vertex = num_vertex;
    int t_num_edge  = num_edge;
    for(int i = 0; i < num_vertex; i++) {
        if(!vertex_judge(i)) q.push(i);
    }

    while(!q.empty()) {
        int v = q.front(); q.pop();
        if(v_del[v]) continue;
        v_del[v] = true;
        t_num_vertex--;
        for(int i = 0; i < tbl_len[v]; i++) {
            int u = graph_tbl[v][i];
            if(v_del[u]) continue;
            pre_deg[u]--; pre_deg[v]--;
            t_num_edge--;
            if(!v_del[u] && !vertex_judge(u)) q.push(u);
        }
    }

    int *v_map = new int[num_vertex], tot = 0;
    for(int i = 0; i < num_vertex; i++) {
        if(v_del[i]) continue;
        v_map[i] = tot++;
    }
    assert(tot == t_num_vertex);
```

Figure 4. Part of G_KD code (part one)
图 4. G_KD 代码部分(一)

```
    assert(tot == t_num_vertex);

    Edge* t_edge = (Edge*) malloc(t_num_edge * sizeof(Edge));

    int *t_pre_deg = (int*) malloc(t_num_vertex * sizeof(int));
    for(int i = 0; i < t_num_vertex; i++) t_pre_deg[i] = 0;

    int *t_tbl_len     = (int*)  malloc(t_num_vertex * sizeof(int));
    int **t_graph_tbl = (int**) malloc(t_num_vertex * sizeof(int*));

    tot = 0;
    for(int i = 0; i < num_edge; i++) {
        int v1 = edge[i].v1, v2 = edge[i].v2;
        if(v_del[v1] || v_del[v2]) continue;
        t_edge[tot].v1 = v_map[v1];
        t_edge[tot].v2 = v_map[v2];
        t_pre_deg[v_map[v1]]++;
        t_pre_deg[v_map[v2]]++;
        tot++;
    }
    assert(tot == t_num_edge);

    for(int i = 0; i < t_num_vertex; i++) {
        t_graph_tbl[i] = (int*) malloc (t_pre_deg[i] * sizeof(int));
        t_tbl_len[i] = 0;
    }

    for(int i = 0; i < t_num_edge; i++) {
        int v1 = t_edge[i].v1, v2 = t_edge[i].v2;
        t_graph_tbl[v1][t_tbl_len[v1]++] = v2;
        t_graph_tbl[v2][t_tbl_len[v2]++] = v1;
    }
```

Figure 5. Part of G_KD code (part two)
图 5. G_KD 代码部分(二)

```
        t_pre_deg[v_map[v1]]++;
        t_pre_deg[v_map[v2]]++;
        tot++;
    }
    assert(tot == t_num_edge);

    for(int i = 0; i < t_num_vertex; i++) {
        t_graph_tbl[i] = (int*) malloc (t_pre_deg[i] * sizeof(int));
        t_tbl_len[i] = 0;
    }

    for(int i = 0; i < t_num_edge; i++) {
        int v1 = t_edge[i].v1, v2 = t_edge[i].v2;
        t_graph_tbl[v1][t_tbl_len[v1]++] = v2;
        t_graph_tbl[v2][t_tbl_len[v2]++] = v1;
    }

    free(edge);
    free(pre_deg);
    free(tbl_len);
    free(graph_tbl);

    edge  = t_edge;
    pre_deg = t_pre_deg;
    tbl_len = t_tbl_len;
    graph_tbl = t_graph_tbl;
    num_vertex = t_num_vertex;
    num_edge   = t_num_edge;
}
```

Figure 6. Part of G_KD code (part three)
图 6. G_KD 代码部分(三)

3.2.4. G_KD 的代码部分

G_KD 算法通过删除度数 ≤ lb-k 的顶点、重编号剩余顶点及更新边连接，实现图的简化，以提升后续操作效率(关键代码见图 4~6 所示)。其核心函数 first_reduction 的关键执行逻辑包括：

1) 状态初始化：创建删除队列与顶点状态数组，标记待处理顶点；

2) 迭代约减：基于队列迭代删除满足度数条件的顶点，同步更新邻域顶点的度数与边信息；

3) 图结构重建：通过顶点映射数组重建邻接表，仅保留有效边，结合内存释放与指针更新完成图简化。

4. 实验结果与分析

4.1. 实验运行环境配置及用例

4.1.1. 相关数据集的介绍

Facebook 社交网络数据集：Facebook 数据集来源于 Facebook 社交网络转换形成的图，其中顶点表示一个账号，边表示账号之间相互关注。这部分用例被 Gao 等人[8]最初收集用于验证最大 k-defective clique 算法的求解性能。该数据集总共拥有 114 个测试用例，本文从中挑选了 10 个用于测试所提出的 G_NP 和 G_KD 算法。

Network Data Repository 数据集：Network Data Repository 收集了来自各个领域的网络图模型，这些图模型具有不同的规模和属性，并且被广泛应用于各类科学研究中。该数据集总共拥有 139 个测试用例，本文从中挑选了具有代表性的 10 个用例，这些用例也在最小顶点覆盖、图染色等问题中被作为基准用例。

4.1.2. G_NP 和 G_KD 算法实验环境

算法实现语言：C++

编译器：g++，使用‘-O3’选项进行编译

实验平台：Workstation17.虚拟机 8GB 运行内存，运行 CentOSLinux7.5 操作系统

时间限制：每个实例的截止时间为 300 秒(5 分钟)

参数设置：其中 G_NP 算法中的 k 取值为 2，3，4，运行截止时间为 300 s，随机种子为 3，而 G_KD 算法中的 k 取值分别为 1，3，5，10，15，20，运行截止时间为 100 s，随机种子为 3。

4.2. G_NP 算法实验结果

Table 1. The reduction results of G_NP on the Facebook social network dataset (partial)
表 1. G_NP 在 Facebook 社交网络数据集上的约减结果(部分)

instance		初始规模	约减后规模		
			K = 2	K = 3	K = 4
socfb-Caltech36.mtx	V	769	528	553	582
	E	16656	14954	15334	15715
socfb-Bowdoin47.mtx	V	2252	1893	1927	1967
	E	84387	80901	81562	82260
socfb-Swarthmore42.mtx	V	1659	1468	1499	1537
	E	61050	59549	59968	60406

(注：表中仅展示约减效果显著的用例，其余用例约减前后规模差异 ＜5%，故省略)

本文采用 Facebook 社交网络数据集验证 G_NP 算法在最大 k-club 问题上的约减性能，结果如表 1 所

示，约减效果受图结构影响显著：密集图(如多数 Facebook 子图)中约减效果有限，因顶点 k 阶邻居数普遍较大，上界筛选难以生效；而稀疏子图(如 socfb-Caltech36)中，k = 2 时顶点约减率达 31.3%，边约减率达 10.2%，效果更优。

Table 2. Reduction results of G_NP on the Network Data Repository dataset (partial)
表 2. G_NP 在 Network Data Repository 数据集上的约减结果(部分)

instance		初始规模	约减后规模		
			K = 2	K = 3	K = 4
coAuthorsDBLP	V	299067	315	94326	299067
	E	977676	7043	461616	977676
rgg_n_2_17_s0	V	131072	67	89645	22503
	E	728453	571	51504	131747
p2p-Gnutella04	V	10876	320	9012	9951
	E	39994	65	37992	39040

实验结果如表 2 所示：k 值增大时约减难度上升，顶点和边的删除量减少，如 coAuthorsDBLP 在 k = 2 时顶点从 30 万减至 315，k = 4 时无约减；不同数据集对 k 值的敏感度不同，图的密度、连通性等特性显著影响约减效果，稀疏图(如 rgg_n_2_17_s0)约减率远高于稠密图。

4.3. G_KD 算法实验结果

以下是运行截至时间为 100 s 的运行结果。

Table 3. Reduction results (partial) of the G_KD algorithm on the Facebook social network dataset (partial)
表 3. G_KD 算法在 Facebook 社交网络数据集上的约减结果(部分)

instance		初始规模	约减后规模					
			K = 1	K = 3	K = 5	K = 10	K = 15	K = 20
Socfb-Amherst41.mtx	V	2235	1894	1931	1967	2052	2127	2235
	E	90954	87720	88402	88988	90091	90689	90954
socfb-Mich67.mtx	V	3748	1739	1964	2123	2443	2882	3303
	E	81903	57526	62723	66094	71793	77348	80657
socfb-Reed98.mtx	V	962	637	708	754	866	962	962
	E	18812	16244	17175	17709	18578	18812	18812

(注：K = 20 时多数用例约减率 < 5%，故省略)

实验结果如表 3 所示 G_KD 算法的约减效果随 k 值增加而减弱：k = 1 时部分用例顶点约减率超 50% (如 socfb-Mich67)，k = 20 时约减率接近 0；受图结构影响，Haverford76 等稠密图在各 k 值下均无显著约减，因顶点度数普遍较高，难以满足“度数 ≤ lb-k”的删除条件。

5. 总结与展望

5.1. 研究成果及研究局限性

5.1.1. 研究成果

本文研究图约减技术在松弛团问题中的应用，设计了针对最大 k-club 的 G_NP 算法(通过动态更新顶点上界、维护多级邻域信息缩减图规模)和针对最大 k-defective clique 的 G_KD 算法(通过约减规则减少搜索空间与计算复杂度)。两种算法在大规模稀疏图上表现出效率优势，适用于大小规模图，且在标准测试

用例上取得较好结果，为大规模图问题提供了新方法。

5.1.2. 研究局限性

实验表明，图约减技术能提升算法效率(尤其大规模稀疏图)，但性能受参数选择、图结构及 k 值影响；此外，约减技术虽减少计算量，但最坏情况下的解质量保证仍需突破。

5.2. 未来工作方向及可能的优化措施

未来工作将聚焦于提升算法的适应性与实用性，具体包括：开发结合图拓扑特征动态调整规则的自适应约减机制，降低对参数和图结构的依赖；构建跨 k-club 与 k-defective clique 的通用约减框架，实现多松弛团模型兼容；基于分布式框架设计并行算法，突破超大规模图的单机处理瓶颈；并在社交网络、生物网络等实际场景中验证优化，增强工程应用价值。

参考文献

[1] 汤小春, 周佳文, 田凯飞, 等. 大图中全部极大团的并行挖掘算法研究[J]. 计算机学报, 2019, 42(3): 513-531.

[2] 陈杰江. 最大松弛团问题求解算法研究[D]: [博士学位论文]. 长春: 东北师范大学, 2023.

[3] Gao, J., Xu, Z., Li, R. and Yin, M. (2022) An Exact Algorithm with New Upper Bounds for the Maximum K-Defective Clique Problem in Massive Sparse Graphs. *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, **36**, 10174-10183. https://doi.org/10.1609/aaai.v36i9.21257

[4] 林维博. 若干最大松弛团问题的精确算法和应用研究[D]: [硕士学位论文]. 成都: 电子科技大学, 2019.

[5] Bourjolly, J., Laporte, G. and Pesant, G. (2002) An Exact Algorithm for the Maximum K-Club Problem in an Undirected Graph. *European Journal of Operational Research*, **138**, 21-28. https://doi.org/10.1016/s0377-2217(01)00133-3

[6] Shahinpour, S. and Butenko, S. (2012) Algorithms for the Maximum K-Club Problem in Graphs. *Journal of Combinatorial Optimization*, **26**, 520-554. https://doi.org/10.1007/s10878-012-9473-z

[7] Veremyev, A. and Boginski, V. (2012) Identifying Large Robust Network Clusters via New Compact Formulations of Maximum K-Club Problems. *European Journal of Operational Research*, **218**, 316-326. https://doi.org/10.1016/j.ejor.2011.10.027

[8] 谭耀昌, 杨俊伟, 李昆洋, 等. 基于角度约束和极大团的点云配准算法[J]. 激光与光电子学进展, 2025, 62(12): 236-246.

[9] Almeida, M.T. and Carvalho, F.D. (2014) Two-Phase Heuristics for the K-Club Problem. *Computers & Operations Research*, **52**, 94-104. https://doi.org/10.1016/j.cor.2014.07.006

[10] Gschwind, T., Irnich, S. and Podlinski, I. (2018) Maximum Weight Relaxed Cliques and Russian Doll Search Revisited. *Discrete Applied Mathematics*, **234**, 131-138. https://doi.org/10.1016/j.dam.2016.09.039

[11] Stozhkov, V., Buchanan, A., Butenko, S. and Boginski, V. (2020) Continuous Cubic Formulations for Cluster Detection Problems in Networks. *Mathematical Programming*, **196**, 279-307. https://doi.org/10.1007/s10107-020-01572-4

[12] Chen, X., Zhou, Y., Hao, J. and Xiao, M. (2021) Computing Maximum K-Defective Cliques in Massive Graphs. *Computers & Operations Research*, **127**, Article 105131. https://doi.org/10.1016/j.cor.2020.105131

[13] Almeida, M.T. and Carvalho, F.D. (2012) Integer Models and Upper Bounds for the 3-Club Problem. *Networks*, **60**, 155-166. https://doi.org/10.1002/net.21455

[14] Motzkin, T.S. and Straus, E.G. (1965) Maxima for Graphs and a New Proof of a Theorem of Turán. *Canadian Journal of Mathematics*, **17**, 533-540. https://doi.org/10.4153/cjm-1965-053-6

[15] Cook, V.J., Sun, S.J., Tapia, J., Muth, S.Q., Argüello, D.F., Lewis, B.L., *et al.* (2007) Transmission Network Analysis in Tuberculosis Contact Investigations. *The Journal of Infectious Diseases*, **196**, 1517-1527. https://doi.org/10.1086/523109

[16] Pajouh, F.M., Balasundaram, B. and Hicks, I.V. (2016) On the 2-Club Polytope of Graphs. *Operations Research*, **64**, 1466-1481. https://doi.org/10.1287/opre.2016.1500