

最大团问题报告

问题分析

问题背景

最大团问题 (Maximum Clique Problem, MCP) 是图论中的一个经典的组合优化问题，是图论中的一个 NP 完全问题。最大团问题是给定一个无向图 G ，找到一个最大的完全子图，即一个最大的团。一个团是图 G 的一个子图，其中任意两个顶点之间有一条边。

本次作业中，我们将使用 ULSA 解决最大团问题。ULSA 在具有挑战性的 frb100-40 基准实例中获得了满足 100 个变量中 99 个的新记录最佳解决方案。

常用算法

最大团问题又称为最大独立集问题 (Maximum Independent Set Problem)。确定性算法有回溯法、分支限界法等，启发式算法有蚁群算法、顺序贪婪算法、DLS-MC 算法和智能搜索算法等。

ULSA 算法

算法思想

ULSA 算法是一种基于局部搜索的启发式算法，它通过不断地在当前解的邻域中搜索更优的解来逐步优化当前解。

它通过简单、无权的随机局部搜索方法，在二元约束满足问题 (CSP) 中快速找到解决方案。其设计关键点是通过强制变量值变化和时间戳机制，有效避免陷入局部最优，从而加速搜索过程。

对于要更新的变量，ULSA 会选择一个将冲突数量最小化的值。这类似于著名的最小冲突启发式 (Minton 等人, 1990)。然而，minconflicts 爬山启发式方法可能会陷入局部最优解，在不暂时恶化冲突总数的情况下无法逃脱 (Minton 等人, 1990)。ULSA 采用了一种替代的未加权方法，首先，ULSA 要求变量偏离其当前值 (即使这会加剧冲突数量)。其次，如果最初选择的端点确实会迫使冲突总数恶化，则扩展邻域以允许从最初选择的冲突的另一个端点进行候选更新。

其输入为所求解图的补图，输出为一个 100 个点的最大近似团。根据提前设置的参数不同，可以得到不同精确程度的近似团。本题中按照得到 98 个点的完全图来设置参数。

算法流程

ULSA 算法的流程如下：

- 初始化：随机设置变量的初始值，并计算初始的违反约束数量。
- 局部搜索：在每次迭代中，选择一个违反约束的变量，并尝试找到一个最佳替换值，使得违反约束的数量最小化。
- 部分成功检查：在一定条件下，检查当前解是否已经足够接近最优解。
- 更新：根据局部搜索的结果更新变量的取值，并记录当前最好的解。
- 终止条件：当达到最大迭代次数或找到最优解时，算法终止

整体算法的流程如下：

- 读取数据，构造图的补图；
- 使用 ULSA 算法，得到最大近似团；

- 遍历最大近似团中的点，去除不在完全图中的点，得到大小为98的完全图；
- 检查完全图是否为最大团，若是则输出，否则继续搜索。

运行环境

- 操作系统：Windows
- IDE：CLion 2024.3.1.1
- 编程语言：C++
- 依赖库：x86gprintrin.h, immintrin.h
- 编译器：MinGW
- 编译选项：-std=gnu99 -mbmi -mavx2 -m64 -funroll-loops -lm

运行时间

使用ULSA算法求解最大团问题，得到了98个点的最大团。在本地环境中，运行时间约为100秒。

参考文献

- <https://www.zgbk.com/ecph/words?SiteID=1&Name=%E6%9C%80%E5%A4%A7%E5%9B%A2%E7%AE%97%E6%B3%95&Type=bkzyb&subSourceType=000003000007000014&SourceID=46849>
- Jin Y, Hao J K. General swap-based multiple neighborhood tabu search for the maximum independent set problem[J]. Engineering Applications of Artificial Intelligence, 2015, 37: 20-33.
- Rosin C D. Unweighted stochastic local search can be effective for random CSP benchmarks[J]. arXiv preprint arXiv: 1411.7480, 2014.