Discrete Applied Mathematics ■ (■■■■) ■■■–■■■

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Fast maximum weight clique extraction algorithm: Optimal tables for branch-and-bound

Satoshi Shimizu, Kazuaki Yamaguchi *, Toshiki Saitoh, Sumio Masuda

Graduate School of Engineering, Kobe University, Japan

ARTICLE INFO

Article history:
Received 28 January 2015
Received in revised form 27 January 2017
Accepted 30 January 2017
Available online xxxx

Keywords:
Maximum weight clique
Exact algorithm
Branch-and-bound
Upper bound calculation
NP-hard

ABSTRACT

A new branch-and-bound algorithm for the maximum weight clique problem is proposed. The proposed algorithm consists of two phases, a *precomputation phase* and a *branch-and-bound phase*. In the precomputation phase, the weights of maximum weight cliques in many small subgraphs are calculated and stored in *optimal tables*. In the branch-and-bound phase, each problem is divided into smaller subproblems, and unnecessary subproblems are pruned using the optimal tables. We performed experiments with the proposed algorithm and five existing algorithms for several types of graphs. The results indicate that only the proposed algorithm can obtain exact solutions for all graphs and that it performs much faster than other algorithms for nearly all graphs.

1. Introduction

A set of vertices V' in a graph $G = (V, E)$ is called a clique if any pair of vertices in V' is adjacent. The maximum clique problem (MCP) is to find the clique of maximum cardinality of a given graph. Here, let $w(v)$ denote the weight of $v \in V$. For a set of vertices $V' \subseteq V$, let $w(V') = \sum_{v \in V'} w(v)$. Given an undirected graph $G = (V, E)$ and weight of vertices $w(\cdot)$, the maximum weight clique problem (MWCP) is to find a clique C such that $w(C)$ is the maximum. Note that the MWCP is a generalization of the MCP.

The MCP and the MWCP are known to be NP-hard [10], and have many applications in coding theory [6], network design [30], computer vision [11], bioinformatics [12], economics [7], etc. The maximum independent set problem and the minimum vertex cover problem for general graphs are equivalent to the MCP and have been well studied.

The branch-and-bound technique is often used in exact algorithms. The branch procedure divides a problem into smaller subproblems and solves them in a recursive manner. During this process, the upper bound of each subproblem is calculated and pruned if it is proved that the subproblem does not contain the global optimum solution (the bounding procedure). In previous studies, several techniques have been investigated to obtain upper bounds for subproblems. For the MCP, vertex coloring is used in numerous algorithms [1,3,5,26,27,29,32,31,33,34]. They calculate vertex coloring in $O(|V|^2)$ or $O(|V|^3)$ time for each subproblem. For the MWCP, some algorithms calculate vertex coloring only once before starting branch-and-bound and use it to obtain upper bound in $O(|V|)$ for each subproblem [13,15,14,16,28]. Upper bound calculation of $O(1)$ time has also been proposed in [21,22]. In these methods, $|V|$ subproblems are solved sequentially. During the execution, an upper bound of subproblem P is calculated from an exact value of subproblems which are already solved. Some algorithms

* Corresponding author.
E-mail addresses: ss81054@gmail.com (S. Shimizu), ky@kobe-u.ac.jp (K. Yamaguchi), saitoh@eedept.kobe-u.ac.jp (T. Saitoh), masuda@kobe-u.ac.jp (S. Masuda).

http://dx.doi.org/10.1016/j.dam.2017.01.026
0166-218X/

Please cite this article in press as: S. Shimizu, et al., Fast maximum weight clique extraction algorithm: Optimal tables for branch-and-bound, Discrete Applied Mathematics (2017), http://dx.doi.org/10.1016/j.dam.2017.01.026

use some upper bounds shown above [13,15,28]. Other approaches have been proposed by previous studies [4,2,8,18–20,24,25,35]. The computation time of algorithms including branch-and-bound procedures strongly depends on tightness and computation time of upper bound calculation. Controlling their balance is very important for branch-and-bound algorithms.

In this paper, we propose a new exact branch-and-bound algorithm for MWCP. Our algorithm consists of two phases, a *precomputation phase* and a *branch-and-bound phase*. In the precomputation, the weights of maximum weight cliques in many small subgraphs are calculated and stored in *optimal tables*. In the branch-and-bound phase, each problem is divided into smaller subproblems and solved in a recursive manner. The branch-and-bound phase is nearly the same as other branch-and-bound algorithms, i.e., the upper bound of each subproblem is calculated using the optimal tables, and the subproblem is pruned if it is unnecessary.

The remainder of this paper is organized as follows. An outline of the proposed algorithm, OTClique, is described in Section 2. Experimental results are shown in Section 4. We conclude the paper in Section 5.

2. Proposed algorithm *OTClique*

The proposed *OTClique* algorithm is outlined as follows.

- Precomputation phase: determines branching order and generates the optimal tables
- Branch-and-bound phase: solves the problem via a branch-and-bound procedure by pruning unnecessary subproblems by their upper bounds.

Before explaining the proposed algorithm, we define some notations and analyze some properties of our upper bound function $UB(\cdot,\cdot)$. We then describe the phases of the proposed algorithm in detail.

2.1. Notation

For an undirected graph $G = (V, E)$ and a set of vertices $V' \subseteq V$, let $G(V')$ and $w_{opt}(V')$ denote the subgraph of G induced by V' and the weight of the maximum weight clique in $G(V')$, respectively. For any vertex $v \in V$, $N(v)$ denotes the set of vertices adjacent to v in G. For any integer $k \geq 2$, a k-tuple $\Pi = (P_1, P_2, \ldots, P_k)$ is a partition of V if $P_1, P_2, \ldots, P_k$ are mutually disjoint and $\bigcup_{i=1}^{k} P_i = V$.

2.2. Upper bound function $UB(\cdot,\cdot)$

Here, we present an analysis of the following function for a subset of vertices $V' \subseteq V$ and a partition $\Pi = (P_1, P_2, \ldots, P_k)$ of V:

$$UB(\Pi, V') = \sum_{i=1}^{k} w_{opt}(V' \cap P_i). \tag{1}$$

The following lemma shows that $UB(\Pi, V')$ is an upper bound of the weight of the maximum weight clique in $G(V')$.

Lemma 1. *Let $G = (V, E)$ be a vertex-weighted graph and $\Pi = (P_1, P_2, \ldots, P_k)$ be a partition of V. Then, the following inequality holds for any $V' \subseteq V$:*

$$w_{opt}(V') \leq UB(\Pi, V'). \tag{2}$$

Proof. The following inequality is immediately obtained, where C is the maximum weight clique in $G(V')$:

$$w_{opt}(V') = w(C) \tag{3}$$

$$= \sum_{i=1}^{k} w(C \cap P_i) \tag{4}$$

$$\leq \sum_{i=1}^{k} w_{opt}(V' \cap P_i) \tag{5}$$

$$= UB(\Pi, V'). \quad \square \tag{6}$$

2.2.1. Example

Let $G = (V, E)$ be a graph shown in Fig. 1 and $\Pi = (P_1, P_2, P_3)$ be a partition of V, where P_1, P_2 and P_3 are $\{v_1, v_2\}$, $\{v_3, v_4, v_5\}$ and $\{v_6, v_7, v_8\}$, respectively. The weights of the vertices are shown in Fig. 1. For example, the value