Finding minimum clique capacity

Maria Chudnovsky[1]
Columbia University, New York, NY 10027

Sang-Il Oum[2]
KAIST, Daejeon, 305-701 Republic of Korea

Paul Seymour[3]
Princeton University, Princeton, NJ 08544

July 21, 2009; revised July 25, 2009

[1]Supported by NSF grant DMS-0758364.
[2]Supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. 2009-0063183)
[3]Supported by ONR grant N00014-01-1-0608.

Abstract

Let C be a clique of a graph G. The *capacity* of C is defined to be $(|V(G) \setminus C| + |D|)/2$, where D is the set of vertices in $V(G) \setminus C$ that have both a neighbour and a non-neighbour in C. We give a polynomial-time algorithm to find the minimum clique capacity in a graph G. This problem arose as an open question in a study [1] of packing vertex-disjoint induced three-vertex paths in a graph with no stable set of size three, which in turn was motivated by Hadwiger's conjecture.

1 Introduction

In this paper, all graphs are finite and have no loops or multiple edges. If G is a graph, a subset X of $V(G)$ is a *clique* if all members of X are pairwise adjacent, and is *stable* if all members of X are pairwise nonadjacent. If $C \subseteq V(G)$, a vertex $v \in V(G) \setminus C$ is *complete* to C if v is adjacent to every member of C, and *anticomplete* to C if it has no neighbour in C.

Let C be a clique of a graph G. Define $A, B, D \subseteq V(G) \setminus C$ as follows:

- A is the set of all $v \in V(G) \setminus C$ that are complete to C;
- B is the set of all $v \in V(G) \setminus C$ that are anticomplete to C; and
- D is the set of all $v \in V(G) \setminus C$ that have both a neighbour and a non-neighbour in C.

Thus $A \cup B \cup D = V(G) \setminus C$, and if $C \neq \emptyset$ then A, B, D are pairwise disjoint.

The problem of choosing a clique C with $|C|$ maximum is NP-hard. On the other hand, it is easy to find a clique C with $|C| + |A|/2$ maximum in polynomial time. To see this, take two copies $p(v), q(v)$ of each vertex v of G, and let H be the bipartite graph, with bipartition

$$(\{p(v) : \ v \in V(G)\}, \{q(v) : \ v \in V(G)\}),$$

in which for all $v \in V(G)$, we make $p(v), q(v)$ nonadjacent, and for all distinct $u, v \in V(G)$, we make $p(u), q(v)$ adjacent if and only if u, v are nonadjacent in G. Find the maximum stable set X in H, and let C be the set of all $v \in V(G)$ such that $p(v), q(v)$ are both in X. It is easy to check that C is the clique of G with $|C| + |A|/2$ maximum.

We define the *capacity* $cap(C)$ of the clique C to be $(|A| + |B|)/2 + |D|$, and in this paper we study finding a clique C with minimum capacity (that is, with $|C| + (|A| + |B|)/2$ maximum). It turns out that we can modify the simple algorithm just given to solve the capacity problem.

A *seagull* in G is an induced three-vertex path in G. In [1] the problem of packing vertex-disjoint seagulls was studied, in connection with Hadwiger's conjecture, and the following min-max formula was given for the maximum seagull packing in graphs with no three-vertex stable set (the *five-wheel* is the graph with six vertices in which one vertex is complete to the vertex set of a cycle of length five):

1.1 *Let G be a graph with no three-vertex stable set, and let $k \geq 0$ be an integer, such that if $k = 2$ then G is not a five-wheel. Then G has k pairwise disjoint seagulls if and only if*

- $|V(G)| \geq 3k$,
- *G is k-connected,*
- *every clique of G has capacity at least k, and*
- *the complement graph of G has a matching of cardinality k.*

This did not directly yield a polynomial-time algorithm to compute the size of the optimum seagull packing, however, because the authors did not know how to compute in polynomial time whether every clique has capacity at least k, and had to resort to the ellipsoid method. In this paper we give a polynomial-time algorithm for the missing step.

2 The algorithm

Our result is:

2.1 *There is an algorithm, with running time $O(n^{3.5})$, which with input an n-vertex graph G, finds a clique C in G with minimum capacity.*

We begin with the following; then 2.1 follows by running 2.2 for every vertex c in turn.

2.2 *There is an algorithm, with running time $O(n^{2.5})$, which with input an n-vertex graph G and a vertex $c \in V(G)$, outputs a clique C containing c, with $cap(C)$ minimum over all cliques that contain c.*

Proof. Here is the algorithm. Let N be the set of neighbours of c and $M = V(G) \setminus (N \cup \{c\})$. Take two new vertices $p(v), q(v)$ of each vertex $v \in V(G)$, and make a graph H with vertex set

$$\{p(v) : v \in N\} \cup \{q(v) : v \in N \cup M\},$$

with edges as follows:

- $\{p(v) : v \in N\}$ and $\{q(v) : v \in N \cup M\}$ are stable sets
- for all distinct $u, v \in N$, $p(u)$ and $q(v)$ are adjacent if and only if u, v are nonadjacent in G
- for all $u \in N$ and $v \in M$, $p(u)$ and $q(v)$ are adjacent in H if and only if u, v are adjacent in G
- for all $u \in N$, $p(u)$ and $q(u)$ are nonadjacent in H.

Thus H is bipartite. Find a stable subset X of $V(H)$ with maximum cardinality. (This takes time $O(n^{2.5})$, using the algorithm of Hopcroft and Karp [2].) Then output

$$\{c\} \cup \{v \in N : p(v) \in X \text{ and } q(v) \in X\}.$$

That completes the description of the algorithm; now we discuss its correctness. Let X be the stable set of H chosen by the algorithm. Let k be minimum such that some clique containing c has capacity $k/2$.

(1) $|X| \geq 2n - k - 2$.

For let C be a clique of G with $c \in C$ and $cap(C) = k/2$. Let A, B, D be as usual. Thus $A, C \setminus \{c\} \subseteq N$ and $B \subseteq M$. The set

$$\{p(v) : v \in C \setminus \{c\}\} \cup \{q(v) : v \in A \cup B \cup (C \setminus \{c\})\}$$

is a stable set of H, with cardinality

$$|A| + |B| + 2|C| - 2 = 2(|A| + |B| + |C| + |D|) - 2cap(C) - 2 = 2n - k - 2.$$

Since X is a maximum stable set of H, it follows that $|X| \geq 2n - k - 2$. This proves (1).

Let $C = \{c\} \cup \{v \in N : \; p(v), q(v) \in X\}$. Thus C is the set returned by the algorithm, and $C \subseteq \{c\} \cup N$. Moreover, if $u, v \in C \setminus \{c\}$ are distinct then $p(u), q(v) \in X$, and since X is stable in H, we deduce that $p(u), q(v)$ are nonadjacent in H, and so u, v are adjacent in G. Consequently C is a clique of G.

(2) $cap(C) \le k/2$.

For let

$$\begin{aligned} A &= \{v \in N \setminus C : \; p(v) \in X \text{ or } q(v) \in X\} \\ B &= \{v \in M : \; q(v) \in X\} \\ D &= V(G) \setminus (A \cup B \cup C). \end{aligned}$$

Thus $|X| = 2(|C|-1) + |A| + |B|$, and since $|X| \ge 2n - k - 2$ it follows that $2(|C|-1) + |A| + |B| \ge 2n - k - 2$, that is,

$$2|C| + |A| + |B| \ge 2(|A| + |B| + |C| + |D|) - k.$$

Consequently $|A| + |B| + 2|D| \le k$. Now since X is stable in H, we deduce that for all $u \in C \setminus \{c\}$ and $v \in B$, $p(u), q(v)$ are nonadjacent in H, and so u, v are nonadjacent in G. Thus every vertex in B is anticomplete to $C \setminus \{c\}$, and since $B \subseteq M$, it follows that every vertex in B is anticomplete to C. We claim that every vertex in A is complete to C. For let $u \in C \setminus \{c\}$ and $v \in A$. Then $v \in N \setminus C$, and one of $p(v), q(v) \in X$; and so since $p(u), q(u) \in X$ and X is stable in H, it follows that either $p(u), q(v)$ are nonadjacent in H (if $q(v) \in X$) or $q(u), p(v)$ are nonadjacent in H (if $p(v) \in X$). In either case it follows that u, v are adjacent in G, and so v is complete to C, as claimed. Consequently $cap(C) \le |A| + |B| + 2|D| \le k/2$. This proves (2).

From (2) and the choice of k, it follows that $cap(C) = k/2$, and so the clique returned by the algorithm is indeed a clique containing c with minimum capacity. This proves 2.2. ∎

References

[1] Maria Chudnovsky and Paul Seymour, "Packing seagulls", submitted for publication.

[2] J. E. Hopcroft and R. M. Karp, "An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs", *SIAM Journal on Computing* 2 (1973), 225–231.