Clique Problem, Cutting Plane Proofs and Communication Complexity☆

Stasys Jukna

Institute of Computer Science, University of Frankfurt, Frankfurt a.M., Germany
Institute of Mathematics and Informatics, Vilnius University, Vilnius, Lithuania.

Abstract

Motivated by its relation to the length of cutting plane proofs for the Maximum Biclique problem, we consider the following communication game on a given graph G with maximum bipartite clique size K. Two parties separately receive disjoint subsets A, B of vertices such that $|A| + |B| > K$. The goal is to identify a nonedge between A and B. We prove that $O(\log n)$ bits of communication are enough for any n-vertex graph.

Key words: Computational complexity, communication complexity, clique problem, cutting plane proof

1. Introduction

A *clique* in a graph is a set of pairwise adjacent vertices. A clique is *maximal*, if it cannot be extended to a larger clique by adding a new vertex. A *biclique* is a pair of disjoint subsets of vertices such that every vertex in one set is adjacent with all vertices in the other set. Thus, the edges between these sets form a complete bipartite subgraph (which is not necessarily an *induced* subgraph if the graph is not bipartite). A *nonedge* in a graph is a pair of its nonadjacent vertices.

The *size* of a clique (or biclique) is the number of its vertices. The *clique number* of G, denoted by $\omega(G)$, is the maximum size of a clique in G, and the *biclique number* of G, denoted by $\omega_{\rm b}(G)$, is the maximum size of a biclique in G.

If A and B are disjoint subset of vertices of size $|A| + |B| > \omega_{\rm b}(G)$, then there must be at least one nonedge lying between A and B. If both A and B are cliques, then the same holds under a weaker condition $|A| + |B| > \omega(G)$. How difficult is to find such a nonedge? In particular, what is the communication complexity of this task?

To be more specific, consider the following communication game between two players, Alice and Bob. To avoid trivialities, we will assume (without mentioning this) that our graphs have no complete stars, that is, vertices adjacent to all remaining vertices—such vertices can be ignored.

Clique Game on $G = (V, E)$:
Alice gets a clique $A \subseteq V$, Bob gets a clique $B \subseteq V$ such that $A \cap B = \emptyset$ and $|A| + |B| > \omega(G)$. The goal is to find a nonedge of G lying between A and B.

☆Research supported by the DFG grant SCHN 503/5-1.

URL: `http://www.thi.informatik.uni-frankfurt.de/ jukna/` (Stasys Jukna)

The communication complexity of this game is the minimum, over all (deterministic) communication protocols, of the number of bits communicated on a worst-case input (A, B). We stress that the graph G in this game is *fixed* and is known to both players. The players are not adversaries—they help and trust each other. The difficulty, however, is that Alice cannot see Bob's set B, and Bob cannot see Alice's set A.

The restriction that the players get cliques (not arbitrary sets of vertices) is not crucial. Indeed, if one of the sets, say, A is not a clique, then it contains a nonedge. Alice can then send both endpoints of this nonedge to Bob using at most $2\lceil \log_2 n \rceil$ bits, and the game is over. So, the only non-trivial inputs in the clique game are cliques.

> **Biclique Game on $G = (V, E)$:**
> Alice gets a set $A \subseteq V$ of vertices, Bob gets a set $B \subseteq V$ of vertices such that $A \cap B = \emptyset$ and $|A| + |B| > \omega_{\mathrm{b}}(G)$. The goal is to find a nonedge of G lying between A and B.

The main difference from the clique game is that we have a stronger promise $|A|+|B| > \omega_{\mathrm{b}}(G)$. If the underlying graph G is bipartite with a bipartition $V = V_1 \cup V_2$, then we additionally require that $A \subseteq V_1$ and $B \subseteq V_2$.

1.1. Motivation

Our motivation to consider clique and biclique games cames from their connection to the length of so-called "tree like" cutting plane proofs for the Maximum Clique problem on a fixed n-vertex graph $G = (V, E)$. The goal in this problem is to find a clique in G of size $\omega(G)$. If we assign a variable x_v to each vertex $v \in V$, then cliques in G are exactly the 0-1 solutions of the system consisting of linear inequalities $x_u + x_v \leq 1$ for all nonedges $\{u, v\} \notin E$, and $x_v \geq 0$ for all vertices $v \in V$. Cutting plane proofs are aiming to prove that, when augmented by the inequality $\sum_{v \in V} x_v \geq \omega(G) + 1$, this system has no 0-1 solutions. When doing this, one applies so-called Gomory–Chvátal rules to derive new inequalities, until a contradiction expressed as inequality $0 \leq -1$ is derived. The proof is *tree-like*, if every derived inequality can be used at most once: if one needs it later again, then it must be re-derived. The length of a proof is the number of produced inequalities. It is known [6] that $n^{\omega(G)}$ inequalities are always sufficient, but no graph requiring super-polynomially many inequalities is known.

In the "find a hurt axiom" game, we (the adversary) first color the variables in red and blue. Then, given a 0-1 assignment α to the variables such that $\sum_{v \in V} \alpha_v \geq \omega(G) + 1$, we split the bits of α between Alice and Bob so that Alice can only see red bits, and Bob can see only blue bits. Their goal is to find a red vertex u and a blue vertex v such that $\alpha_u = \alpha_v = 1$ and $\{u, v\}$ is a nonedge in G. If the graph is bipartite with bipartition $V = V_1 \cup V_2$, then we only have inequalities $x_u + x_v \leq 1$ for all nonedges $\{u, v\}$ with $u \in V_1$ and $v \in V_2$. In this case, the goal is to find such a nonedge under a stronger promise $\sum_{v \in V} \alpha_v \geq \omega_{\mathrm{b}}(G) + 1$.

Suppose now that for some coloring of variables, the "find a hurt axiom" game for a graph G requires K bits of communication. Impagliazzo et al. [9] have proved that then every tree-like cutting planes proof of the 0-1 unsatisfiability of the system for G must either use super-polynomially large coefficients, or must produce at least $2^{\Omega(K/\log n)}$ inequalities; see [10, Section 19.3] for details. It was therefore a hope that n-vertex graphs G requiring more than $\log^2 n$ bits of communication in the biclique or at least in the clique game exits.

1.2. Our results

Our main result destroys the first hope: in the biclique game a logarithmic number of bits of communication is enough regardless of the given graph.

Theorem 1. *In the biclique game,* $7.3 \log_2 n + O(1)$ *bits of communication are enough for every n-vertex graph.*

Our second result destroys the second hope for many graphs. It shows that also in the clique game, a logarithmic number of bits is enough for many graphs. Let $\kappa(G)$ denote the number of maximal cliques in G.

Theorem 2. *In the clique game,* $\log_2 \kappa(G) + 7.3 \log_2 n + O(1)$ *bits of communication are enough for every n-vertex graph.*

There are many n-vertex graphs $G = (V, E)$ for which $\kappa(G)$ is polynomial in n. In particular, $\kappa(G) \leq n(d/2)^{p-2}$ holds for every K_p-free graph of maximal degree $d \geq 2$ [14]; $\kappa(G) \leq n^p$, where p is the chromatic number of G [12]; $\kappa(G) \leq (|E|/p + 1)^p + |E|$, where p is the maximum number of edges in an induced matching in the complement of G [4, 2]. If $p = O(\log n)$ then Theorem 2 implies that $O(\log^2 n)$ bits of communication are enough in the clique game on all such graphs. Consequently, the communication complexity arguments fails for such graphs, even for the Maximum Clique problem (not just for the Maximum Biclique problem).

The rest is devoted to the proofs of Theorems 1 and 2. But first we shortly recall two powerful tools we will use in the proofs.

2. Results we use

Recall that a *threshold-k* function Th_k^n accepts a 0-1 vector of length n if and only if it contains at least k ones. By a *monotone circuit* we will mean a circuit consisting of fanin-2 AND and OR gates; no negated variables are allowed as inputs. The *depth* of a circuit is the length of a longest path from an input to the output gate.

Theorem 3 (Valiant [18])**.** *Every threshold function* Th_k^n *can be computed by a monotone circuit of depth at most* $5.3 \log_2 n + O(1)$*.*

The second result we use is the tight relation between the depth of circuits computing a given boolean function f and the communication complexity in the following communication game for f. Alice gets a vector $x \in f^{-1}(1)$, Bob gets a vector $y \in f^{-1}(0)$, and their goal is to find a position i such that $x_i \neq y_i$. In the *monotone* game, the goal is to find a position i such that $x_i = 1$ and $y_i = 0$. In general, such a position may not exists, but it always exists, if the function f is monotone. These games were introduced by Karchmer and Wigderson in [11], where they prove that the communication complexity of the (monotone) game on a (monotone) boolean function f is *exactly* the minimum depth of a (monotone) circuit for f. We will only use the easier direction: small depth gives efficient communication protocols.

Lemma 1 (Karchmer–Wigderson [11])**.** *If a monotone boolean function f can be computed by a monotone circuit of depth d, then the monotone game for f can be solved using d bits of communication.*

To recall how do small-depth circuits actually lead to efficient communication protocols, we include a simple proof of this lemma.

Proof. We may assume that Alice and Bob have agreed on a monotone circuit g of smallest depth computing f. Now suppose Alice gets an input x such that $g(x) = 1$, and Bob gets an input y such that $g(y) = 0$. In order to find an i such that $x_i = 1$ and $y_i = 0$, the players traverse the circuit g backwards starting at the output gate by keeping the invariant: $g'(x) = 1$ and $g'(y) = 0$ for every reached subcircuit g'.

Namely, suppose the output gate of g is an AND gate, that is, we can write $g = g_0 \wedge g_1$. Then Bob sends a bit i corresponding to a function g_i such that $g_i(y) = 0$; if both $g_0(y)$ and $g_1(y)$ output 0, then Bob sends 0. Since $g(x) = 1$, we know that $g_i(x) = 1$. If $g = g_0 \vee g_1$, then it is Alice who sends a bit i corresponding to a function g_i such that $g_i(x) = 1$; again, if both $g_0(x)$ and $g_1(x)$ output 1, then Alice sends 0. Since $g(y) = 0$, we know that $g_i(y) = 0$.

Alice and Bob repeat this process until they reach a leaf of the circuit. This leaf is labeled by some variable (the circuit is monotone). If this is the i-th variable, then $x_i = 1$ and $y_i = 0$ implying that i is a correct answer. □

3. The biclique game: proof of Theorem 1

The strategy of the proof is to recast the biclique game as a Karchmer–Wigderson game on a small-depth monotone circuit.

Let $G = (V, E)$ be a graph on n vertices, and let $F := \binom{V}{2} \setminus E$ be the set of all nonedges of G. Say that a nonedge e is *incident* with a subset $A \subseteq V$, if $e \cap A \neq \emptyset$. Associate with every subset $A \subseteq V$ two vectors p_A and q_A in $\{0,1\}^{|F|}$ whose coordinates correspond to nonedges $e \in F$:

- $p_A(e) = 1$ if and only if $e \cap A \neq \emptyset$;
- $q_A(e) = 0$ if and only if $e \cap A \neq \emptyset$.

Thus, p_A is the characteristic vector of all nonedges incident with A, and q_A is the complement of p_A. Given an input (A, B), the goal in the biclique game is to find a position (a nonedge) e such that $p_A(e) = 1$ (e is incident with A) and $q_B(e) = 0$ (e is incident with B). By Lemma 1, this task can be solved by providing a small-depth monotone circuit separating the vectors p_A and q_B.

To construct such a circuit, take a set $X = \{x_e \colon e \in F\}$ of boolean variables, one for each nonedge of G. Associate with each vertex $v \in V$ the monomial $m_v(X)$, which is the AND of all variables x_e corresponding to the nonedges e incident with v:

$$m_v(X) := \bigwedge_{e \in F \colon v \in e} x_e \, .$$

Thus, m_v accepts a given set of nonedges if and only if this set contains all nonedges incident with v. Let $f_k(X)$ be the threshold-k function applied to the outputs of these monomials:

$$f_k(X) := \mathrm{Th}_k^n(m_v(X) \colon v \in V) \, .$$

Since each monomial m_v has a monotone fanin-2 circuit of depth at most $\log_2 n + 1$, Theorem 3 implies that the function f_k can be computed by a monotone circuit of depth at most $5.3 \log_2 n + O(1) + \log_2 n + 1 = 6.3 \log_2 n + O(1)$.

Lemma 2. *Let A be a set of $|A| = k$ vertices, and B a set of $|B| > \omega_{\mathrm{b}}(G) - k$ vertices. Then $f_k(p_A) = 1$ and $f_k(q_B) = 0$.*

Proof. The function f_k accepts a given a set of nonedges of G if and only if this set contains all nonedges incident with at least k distinct vertices. Thus, we can write f_k as the OR of monomials

$$m_S(X) := \bigwedge_{v \in S} m_v = \bigwedge_{e\colon e \cap S \neq \emptyset} x_e$$

over all k-elements subset $S \subseteq V$. Since vector p_A sets to 1 all variables x_e with $e \cap A \neq \emptyset$, this vector is accepted by the monomial m_A, and hence, by the function f_k.

Now let B be an arbitrary set of $|B| > \omega_{\mathrm{b}}(G) - k$ vertices. We have to show that the vector q_B is rejected by every monomial m_S corresponding to a subset S of $|S| = k$ vertices. Since vector q_B sets to 0 all variables x_e with $e \cap B \neq \emptyset$, this is equivalent to showing that some nonedge must be incident with both sets S and B. Since we assumed that G contains no complete stars, this holds if the sets S and B share a common vertex. So, we can assume that $S \cap B = \emptyset$. But then the condition $|S| + |B| > \omega_{\mathrm{b}}(G)$ implies that there must be a nondege lying between S and B. □

We can now describe a communication protocol for the biclique game on a given n-vertex graph G. Recall that inputs to this game are pairs (A, B) of disjoint subsets of vertices such that $|A| + |B| > \omega_{\mathrm{b}}(G)$. The goal is to find a nonedge lying between A and B.

1. Alice first uses at most $\log_2 n + 1$ bits to communicate to Bob the size $k = |A|$ of her set A.
2. Knowing k, the players take a monotone circuit of depth $d = 6.3 \log_2 n + O(1)$ computing the k-th function f_k. By Lemma 2, the players know that this function separates the associated vectors p_A and q_B: $f_k(p_A) = 1$ and $f_k(q_B) = 0$.
3. The players then use Lemma 1 to transform the circuit for f_k to a protocol. The protocol finds a position e such that $p_A(e) = 1$ and $q_B(e) = 0$.
4. By the definition of vectors p_A and q_B, the found nonedge e must be incident with both sets A and B. Since $A \cap B = \emptyset$, the nonedge e can lie neither within A (then we would have $q_B(e) = 1$) nor within B (then we would have $p_A(e) = 0$). So, e must be a desired nonedge lying between A and B.
5. The total number of communicated bits is $d + \log_2 + 1 = 7.3 \log_2 n + O(1)$.

This completes the proof of Theorem 1. □

Remark 1. One could presume that the main reason, why the biclique game has small communication complexity, is just the fact that the biclique problem *is* solvable in polynomial time via, say, the maximum matching algorithm. In the biclique problem, we are given a graph G and a positive integer r; the goal is to decide whether G contains a biclique $A \times B$ of size $|A| + |B| \geq r$. However, it is known [13] that a similar *maximum edge biclique* problem is already **NP**-complete, even for bipartite graphs. In this latter problem, the goal is to decide whether G contains a biclique $A \times B$ with $|A \times B| \geq r$ edges. If G is a graph, in which every biclique has at most r edges, then the corresponding to this problem game is, given two disjoint sets A, B of vertices such that $|A \times B| > r$, to find a nonedge between A and B. It is easy to see that $O(\log n)$ bits of communication are enough also in this game. For this, it is enough just to replace the condition $|A| + |B| > \omega_{\mathrm{b}}(G)$ in Lemma 2 by the condition $|A \times B| > r$. The rest of the proof is the same, by using condition $|B| > r/k$ instead of $|B| > \omega_{\mathrm{b}}(G) - k$ in Lemma 2.

4. The clique game: proof of Theorem 2

Consider the clique game for a given n-vertex graph $G = (V, E)$. Inputs to this game are pairs (A, B) of disjoint cliques such that $|A| + |B| > \omega(G)$, and the goal is to find a nonedge lying between A and B.

Let us first see why we cannot use the same separating functions f_k as in the biclique game. The reason if that vector q_B sets to 1 all variables x_e such that $e \cap B = \emptyset$. Hence, if $S \cap B = \emptyset$ and if there are no nonedges between S and B, then $q_B(e) = 1$ for all nonedges incident with S, implying that the monomial m_S wrongly accepts the vector q_B. To get rid of this problem, we use more complicated separating functions.

The *induced k-clique function* of an n-vertex graph G is a monotone boolean function of n variables, one for each vertex. Given a subset of vertices, the function outputs 1 if and only if some k of these vertices form a clique in G.

Define a modified version g_k of the function f_k by taking the induced k-clique function Cliq of G instead of the threshold function Th^n_k:

$$g_k(X) := \mathrm{Cliq}(m_v(X) \colon v \in V)\,.$$

Lemma 3. *Let A and B be two disjoint cliques in G such that $|A| + |B| > \omega(G)$. If $|A| = k$, then $g_k(p_A) = 1$ and $g_k(q_B) = 0$.*

Proof. The modified function g_k accepts a subset of nonedges if and only if this set contains all nonedges incident to some set S of $|S| = k$ vertices *forming a clique in G*. Thus, g_k is the OR of monomials $m_S(X)$ over all k-cliques $S \subseteq V$ (instead of all k-element subsets). Recall that m_S is the AND of all variables x_e corresponding to nonedges e incident with S.

Since the monomial m_A accepts p_A, the vector p_A is accepted by g_k. To show that every monomial m_S rejects the vector q_B, we now use the fact that both S and B are cliques. Since we assumed that G contains no complete stars, this holds if the sets S and B share a common vertex. So, we can assume that $S \cap B = \emptyset$. But then the condition $|S| + |B| > \omega(G)$ ensures that there must be a nonedge e lying between S and B. Since the vector q_B sets the variable x_e to 0, $m_S(q_B) = 0$ follows. □

The rest of the proof of Theorem 2 is the same as that of Theorem 1 by using Lemma 3 instead of Lemma 2. The only difference is that now we do not know how deep monotone circuits computing the functions g_k are. So, let $d(G)$ denote the maximum, over all integers $1 \leq k \leq n$, of the minimum depth of a monotone circuit computing the induced k-clique function of G.

Since each monomial $m_v(X)$ has a monotone circuit of depth at most $\log_2 n + 1$, all functions $g_1, \ldots, g_n$ can be computed by monotone circuits of depth at most $d(G) + \log_2 n$. Thus, arguing as in the proof of Theorem 1, and using functions g_k instead of f_k, we will obtain a protocol for the clique game on G which uses at most $d(G) + 2\log_2 n + O(1)$ bits of communication (Alice spends additional $\log_2 n + 1$ bits to send the size $|A| = k$ of here clique). So, it remains to upper-bound $d(G)$ is terms of the number $\kappa(G)$ of maximal cliques in G.

It is easy to see that the induced k-clique function of a complete graph K_n is the threshold-k function Th^n_k. Thus, in terms of graphs, Theorem 3 states that $d(K_n) \leq 5.3\log_2 n + O(1)$. The complete graph K_n has only one maximal clique—the graph itself; hence, $\kappa(K_n) = 1$. But Valiant's theorem can be easily extended to n-vertex graphs G with a larger number of maximal cliques.

Lemma 4. $d(G) \leq \log_2 \kappa(G) + 5.3 \log_2 n + O(1)$.

Proof. Let $G = (V, E)$ be a graph, and $\mathrm{Cliq}(x)$ be its induced k-clique function. The variables x_v here correspond to the vertices $v \in V$. Given an assignment $x \in \{0,1\}^n$ to these variables, the function accepts x if and only if the set $S_x = \{v\colon x_v = 1\}$ contains a k-clique of G. Since every clique is contained in some maximal clique, we have that $\mathrm{Cliq}(x) = 1$ if and only if $|S_x \cap C| \geq k$ holds for at least one maximal clique C of G. This latter condition can be tested by threshold-k function $\mathrm{Th}_k^n(\delta_C \wedge x)$, where $\delta_C \in \{0,1\}^n$ is the characteristic vector of C, and $\delta_C \wedge x$ is a component-wise AND. By taking the OR, over all $\kappa(G)$ maximal cliques C, of monotone circuits computing the threshold functions $\mathrm{Th}_k^n(\delta_C \wedge x)$, and using Theorem 3, we obtain a monotone circuit of depth at most $\log_2 \kappa(G) + 5.3 \log_2 n + O(1)$ computing $\mathrm{Cliq}(x)$. □

This completes the proof of Theorem 2.

5. Conclusion and open problems

Note that our communication protocols are not explicit because the construction of a small-depth monotone circuits for the majority function in [18] is probabilistic. To get an explicit protocol, one can use the construction of a circuit of depth $K \log_2 n$ for the majority function given in [1]. But the constant K resulting from this construction is huge, it is about 5000.

The main message of Theorem 1 is that communication complexity arguments cannot yield any non-trivial lower bounds on the length of cutting plane proofs for systems corresponding to the Maximum Biclique problem, because $O(\log n)$ bits of communication are enough in the biclique game on *all* n-vertex graphs G.

However, the case of the Maximum Clique problem remains unclear. Theorem 2, together with known upper bounds on the number of maximal cliques, implies that $O(\log n)$ bits are enough for a lot of graphs. Still, we could not exclude that graphs requiring more bits exist. Do n-vertex graphs G requiring more than $\log^2 n$ bits of communication in the clique game exist?

The clique and biclique games on a given graph G are special cases of a *monotone* Karchmer–Wigderson game: given a pair (A, B) of two intersecting subsets of a fixed n-element set, find an element in their intersection $A \cap B$. (In the biclique game, elements are nonedges.) In the *non-monotone* game, inputs are pairs of distinct sets, and the goal is to find an element in the symmetric difference $A \oplus B := (A \setminus B) \cup (B \setminus A)$.

It is usually much easier to find an element in the symmetric difference than in the intersection. Say, if the players know that $|A| \neq |B|$, then $O(\log n)$ bits are already enough to find an element in $A \oplus B$ [7]. However, monotone games (with the goal to find an element in the intersection) usually require much more bits of communication. For example, if A is a set of $m = n/3$ vertex-disjoint edges in K_n, and B is the set of edges in the complement of a clique on $m-1$ vertices in K_n, then $A \cap B \neq \emptyset$. Since the sizes $|A| = m$ and $|B| = \binom{n}{2} - \binom{m-1}{2}$ are clearly different, [7] implies that $O(\log n)$ bits of communication are enough to find an element (an edge) in $A \oplus B$. But it is known [16] that even $\Omega(n)$ bits of communication are necessary to find an element in $A \cap B$. It is therefore interesting that, in the biclique game, a logarithmic number of communicated bits is enough even to find an element in the intersection $A \cap B$, not just in $A \oplus B$.

A next open problem is to understand the (monotone) complexity of the induced k-clique functions. To our knowledge, these functions have not been investigated earlier. Recall that such a function takes subsets of vertices of a given (fixed) graph G as inputs, and accepts such a subset if and only if the induced subgraph of G on these vertices contains a k-clique. This function reminds us the well-known **NP**-complete Clique function CLIQUE(n, k) restricted to only *spanning* subgraphs of one *fixed* graph G. Recall that inputs for CLIQUE(n, k) are graphs on the same (fixed) set of vertices; thus, variables in this case correspond to edges, not vertices. The function accepts a graph if and only if it contains a k-clique.

It is known that, for particular choices of $k = k(n)$, the function CLIQUE(n, k) requires monotone circuits of depth $\Omega(\sqrt{k} \log n)$ [17, 3], and even of depth $\Omega(n^{1/3})$ [8]. Can the arguments of [17, 3, 8] be adapted induced k-clique functions (at least for random, not explicitly given graphs)? Actually, it is even not clear whether there exist a sequence $(G_n : n = 1, 2, \ldots)$ of n-vertex graphs G_n for which the induced k-clique functions form an **NP**-complete problem.

It also remains unclear how crucially the communication complexity of the clique game depends on the monotone circuit depth of induced k-clique functions. We have only shown that the latter is always an *upper* bound for the former (up to an additive logarithmic factor). Does some reasonable converse (up to an additive $\log^2 n$ factor) of this inequality hold?

Finally, let us mention that a different type of (adversarial) games, introduced in [15], was recently used in [5] to derive strong lower bounds for tree like *resolution* proofs for the Maximum Clique problem. Is there some analogue of these games in the case of cutting plane proofs?

Acknowledgments

I am thankful to Mario Szegedy for interesting initial discussions on the biclique game, and to Jacobo Torán for detecting an error in a previous protocol for the clique game.

References

[1] M. Ajtai, J. Komlós, and E. Szemerédi, Sorting in $c \log n$ parallel steps, Combinatorica 3(1) (1983) 1–19.

[2] V. E. Alekseev, An upper bound for the number of maximal independent sets in a graph, Discrete Math. and Appl. 17(4) (2007) 355–359.

[3] N. Alon and R. Boppana, The monotone circuit complexity of boolean functions, Combinatorica 7(1) (1987) 1–22.

[4] E. Balas and C. S. Yu, On graphs with polynomially solvable maximal-weight clique problem, Networks 19(2) (1989) 247–253.

[5] O. Beyersdorff, N. Galesi, and M. Lauria, Parameterized complexity of DPLL search procedures, Springer Lect. Notes in Comput. Sci. 6695 (2011) 19–22.

[6] V. Chvátal, W. Cook, and M. Hartmann, On cutting-plane proofs in combinatorial optimization, Lin. Algebra and Appl. 114/115 (1989), 455–499.

[7] G. S. Brodal and T. Husfeldt, A communication complexity proof that symmetric functions have logarithmic depth. Technical Report RS-96-1, BRICS, 1996.

[8] M. Goldmann and J. Håstad, A simple lower bound for monotone clique using a communication game, Inf. Process. Lett. 41(4) (1992) 221–226.

[9] R. Impagliazzo, T. Pitassi, and A. Urquhart, Upper and lower bounds for tree-like cutting planes proofs, in: Proc. of 9th Ann. IEEE Symp. on Logics in Computer Science (1994) 220–228.

[10] S. Jukna, Boolean Function Complexity: Advances and Frontiers, Springer 2012.

[11] M. Karchmer and A. Wigderson, Monotone circuits for connectivity require super-logarithmic depth, SIAM J. Discrete Math. 3 (1990) 255–265.

[12] J. W. Moon and L. Moser, On cliques in graphs, Israel J. Math. (1965) 23–28.

[13] R. Peeters, The maximum edge biclique problem is NP-complete, Discrete Appl. Math 131 (2003) 651–654.

[14] E. Prisner, Graphs with few cliques, in: Graph Theory, Combinatorics, and Applications, Proc. of 7th Quadrennial Int. Conf. on the Theory and Applications of Graphs (Y. Alavi, A. Schwenk ed.) John Wiley and Sons, (1995) 945–956.
[15] P. Pudlák and R. Impagliazzo, A lower bound for DLL algorithms for SAT, in: Proc. of 11th Symp. on Discrete Algorithms (2000) 128–136.
[16] R. Raz and A. Wigderson, Monotone circuits for matching require linear depth, J. ACM 39(3) (1992), 736–744.
[17] A. A. Razborov, Lower bounds for the monotone complexity of some boolean functions, Soviet Math. Dokl. 31 (1985) 354–357.
[18] L. G. Valiant, Short monotone formulae for the majority function, J. Algorithms 5 (1984) 363–366.