Nordic Journal of Computing 1(1994), 493–515.

HARD GRAPHS FOR THE RANDOMIZED BOPPANA-HALLDÓRSSON ALGORITHM FOR MAXCLIQUE

MARCUS PEINADO
Department of Computer Science, Boston University
Boston, MA 02215, U.S.A.

Abstract. A randomized version of the MAXCLIQUE approximation algorithm by Boppana and Halldórsson is analyzed. The Boppana Halldórsson algorithm has the best performance guarantee currently known for the MAXCLIQUE problem. This paper presents a class of graphs on which the performance ratio of the randomized version of the algorithm is not better than $\Omega(\sqrt{n})$ with probability greater than $1 - 1/n^{\omega(1)}$.

CR Classification: F.2.2, G.2.1, G.2.2, G.3

Key words: approximation algorithms, MAXCLIQUE, randomization

1. Introduction

Unlike many other NP-hard problems, the MAXCLIQUE problem has resisted attempts to find efficient approximation algorithms. Indeed, the well known result of Arora *et al.* [1992] proves that no deterministic polynomial time algorithm can approximate the maximum clique in a graph to within a factor of n^c for some (very small) $c > 0$ unless $P = NP$. The performance of an approximation algorithm A for MAXCLIQUE on an input graph G is generally measured by the ratio of the size of the largest clique in G and the size of the clique A finds when run on G. The *performance guarantee* of A is the maximum such ratio over all inputs. Recently, Boppana and Halldórsson [1992] have found a subgraph exclusion algorithm to approximate MAXCLIQUE with a performance guarantee of $O(n/\log^2 n)$. This is currently the best performance guarantee known for the MAXCLIQUE problem. Much effort has been devoted to narrowing the gap between the positive and the negative approximation result. Currently, the sharpest negative bound is due to Bellare and Sudan [1994]. Assuming $N\tilde{P} \neq$ co-$R\tilde{P}$, it asserts that MAXCLIQUE cannot be approximated within n^c where $c = 0.25 - o(1)$.

Boppana and Halldórsson [1992] also show, non-constructively, that graphs on which the performance of their algorithm is not better than $\Theta(n/\log^2 n)$ have to exist. Indeed, it is not too difficult to construct graphs explicitly on which the performance of the algorithm is bad. These simple constructions rely on the fact that the algorithm selects the vertices in one particular

Received May 1994. Accepted September 1994.

ordering (e.g. lexicographic order). Thus, one might reasonably expect that if the algorithm selects the vertices *at random*, its performance might with high probability improve. A similar idea is the basis of most successful randomized algorithms: bad worst-case performance of the deterministic algorithm on few inputs is traded off against a very small probability of bad performance of the randomized algorithm on many inputs. Kučera [1991] investigates this idea in the context of the GRAPH COLORING problem. He derives a lower bound for the randomized greedy heuristic for GRAPH COLORING.

This paper shows that randomization can only have limited success when applied to the Boppana-Halldórsson algorithm. It displays a class of graphs which contain cliques of size n^α (given any constant $\alpha < 1/2$) and proves that the size of the clique found by the randomized version of the Boppana-Halldórsson algorithm is smaller than n^δ (for all $\delta > 0$) with probability greater than $1 - n^{-\omega(1)}$. Consequently, even with polynomial amplification, the probability of finding a larger clique is less than $n^{-\omega(1)}$. As an intermediate step in the proof, it is shown that the central subprocedure of the algorithm – which in itself is a generalization of the randomized greedy method – performs worse than n^α for all $\alpha < 1$.

Unlike the results mentioned above which are based on probabilistically checkable proof techniques, the negative results in this paper apply only to the algorithm considered here – not to algorithms for the MAXCLIQUE problem in general. This allows us to give a tighter negative bound. In addition, our result does not depend on any complexity theoretic assumption like $N\tilde{P} \neq \text{co-}R\tilde{P}$ which – although generally believed to be true – is open and clearly stronger than $P \neq NP$.

The critical component of the graphs discussed in this paper are random graphs with a forced clique of size n^α $(0 < \alpha < 1)$. These graphs have been used by Jerrum [1992] to show that the Metropolis process cannot approximate MAXCLIQUE to within a factor n^α for any $\alpha < 1/2$.

The analysis of the algorithm's performance on random graphs is complicated by the fact that during its execution, the algorithm destroys (excludes) certain parts of the graph. As parts of the originally random graph are removed, it can no longer be assumed that the remaining edges are independent. Therefore, the well known techniques for the analysis of random graphs cannot be applied. For this reason, the graphs presented here are somewhat more complicated. They retain their basic random graph structure even after a limited number of subgraph exclusions.

The remaining parts of the introduction contain definitions, a description of the algorithm and an outline of the paper.

1.1 Definitions and Notation

Let $G = (V, E)$ be a graph. If G is not clear from the context, $V(G)$ is used to denote the vertex set of G. For $v \in V$ let the neighborhood of v in G be $\mathcal{N}_G(v) = \{u \in V | \{v, u\} \in E\}$ and let the neighborhood of a set

$S \subseteq V$ of vertices be $\mathcal{N}_G(S) = \bigcap_{v \in S} \mathcal{N}_G(v)$. Similarly define $\bar{\mathcal{N}}_G(v) = \{u \in V | \{v, u\} \notin E \text{ and } u \neq v\}$ and $\bar{\mathcal{N}}_G(S) = \bigcap_{v \in S} \bar{\mathcal{N}}_G(v)$. Thus $\mathcal{N}_G(S)$ is the set of all vertices $v \in V$ which are adjacent to *all* vertices in S, and $\bar{\mathcal{N}}_G(S)$ is the set of vertices which are adjacent to *no* vertex in S. If the graph G can be inferred from the context we will simply write $\mathcal{N}(v)$, $\bar{\mathcal{N}}(v)$, $\mathcal{N}(S)$, $\bar{\mathcal{N}}(S)$. $\mathcal{G}(n, p)$ denotes the distribution of random graphs with n vertices and edge probability p. In general, given a distribution $\mathcal{D}$, the expression '$X \in \mathcal{D}$' is used as a shorthand for 'generate X according to distribution $\mathcal{D}$' or, equivalently, 'let X be a random variable with distribution $\mathcal{D}$'. $\mathcal{P}(S)$ denotes the power set of the set S. Throughout this paper, n denotes the number of vertices in the given graph. All logarithms have base 2.

The standard o and ω notation is used in two contexts: $g(n) < n^{o(1)}$ expresses that the function $g(n)$ grows asymptotically slower than n^ϵ for all constant $\epsilon > 0$. Similarly, $n^{-\omega(1)}$ denotes a function that approaches zero faster than $1/p(n)$ for any polynomial $p(n)$.

1.2 The Algorithm

We give a brief summary of the presentation of Boppana and Halldórsson [1992]. The algorithm consists of a subgraph exclusion procedure and a recursive subprocedure (RAMSEY) which is motivated by Ramsey theory and which, given an input graph, returns a clique and an independent set. The subgraph exclusion procedure calls RAMSEY, stores the clique returned, and removes the independent set from the graph. This is repeated until the graph has become empty.

The RAMSEY subprocedure is a generalization of the greedy method:

```
greedy(G):
        IF G is empty THEN return ∅
        ELSE choose a vertex v
                return {v}∪ greedy(N(v))
```

The selected vertex is called a **pivot vertex**. The vertex set returned is a clique because when a vertex is selected, its non-neighborhood is no longer considered. Ignoring the non-neighborhood can lead to arbitrarily bad results because it might contain much larger cliques than the neighborhood.

RAMSEY improves and generalizes the greedy method by making an additional call to search the non-neighborhood of the pivot vertex. Thus, each recursive call has two cliques to choose from: the clique found in the neighborhood of the pivot together with the pivot and the clique found in the non-neighborhood. RAMSEY returns the larger one.

Clearly, the same idea can be used to find an independent set by interchanging the terms neighborhood and non-neighborhood. RAMSEY returns both an independent set and a clique in the input graph.

RAMSEY$((V, E))$:
1 IF (V, E) is empty THEN return $(\emptyset, \emptyset)$
2 ELSE choose a vertex $v \in V$
3 $(C_1, I_1) :=$ RAMSEY$(\mathcal{N}(v))$
4 $(C_2, I_2) :=$ RAMSEY$(\bar{\mathcal{N}}(v))$
5 return (larger of $(C_1 \cup \{v\}, C_2)$, larger of$(I_1, I_2 \cup \{v\})$)

Using Ramsey theory, Boppana and Halldórsson [1992] show for the clique C and independent set I returned by RAMSEY(G) that $|C| \cdot |I| \geq \log^2 n/4$. This bound in itself does not guarantee a minimum size of C since $|I|$ can be large.

The purpose of the subgraph exclusion algorithm is to modify the graph such that, eventually, $|I|$ will be small. This is achieved by repeatedly calling RAMSEY and excluding (removing) the returned independent sets:

IS Removal(G):
$i := 1$
$(C_i, I_i) :=$ RAMSEY(G)
WHILE $G \neq \emptyset$
$G := G \setminus I_i$
$i := i + 1$
$(C_i, I_i) :=$ RAMSEY(G)
return $\max_{j \leq i} C_j$

A clique in G can lose at most one vertex per iteration because a clique and an independent set can share at most one vertex. If the graph has a large enough clique, a constant fraction of the graph will be left even if all independent sets of a certain minimum size k are excluded. If RAMSEY is run on the resulting graph, the size of I can be at most k. This implies a lower bound on $|C| \geq \log^2 n/(4k)$. If the largest clique is small, the performance of the algorithm on the graph is trivially good. The result of this analysis is a performance guarantee of $O(n/\log^2 n)$ (cf. Boppana and Halldórsson [1992] for details). Furthermore, Boppana and Halldórsson [1992] show non-constructively that this performance guarantee is tight, i.e. that graphs have to exist on which the performance of the algorithm is not better than $\Theta(n/\log^2 n)$.

An important concept in the analysis of RAMSEY which was used in Boppana and Halldórsson [1992] and which will be used frequently in this paper is the tree of recursive calls made by RAMSEY. We call this tree the *computation tree*. Each node in the computation tree corresponds to a recursive call made by RAMSEY. If the input graph to the recursive call is empty, RAMSEY returns in line 1 and the corresponding node has no children. Otherwise, the node has two children corresponding to the two recursive calls of lines 3 and 4. We adopt the convention of identifying the recursive call of line 3 which searches the neighborhood with the left child and of identifying the call in line 4 (non-neighborhood) with the right child. We will ignore all nodes corresponding to recursive calls with an empty input graph. Each

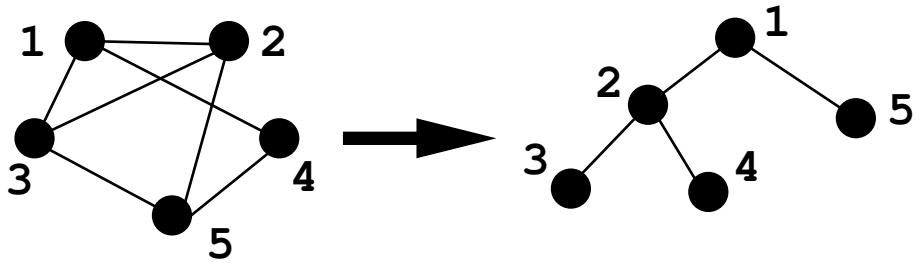

Fig. 1: The computation tree (labeled by the pivots) that results from running RAMSEY on the graph on the left-hand side and choosing the pivots according to the lexicographic order.

node can be labeled with the pivot vertex chosen in line 2 or with the input graph to the corresponding recursive call. We will use both kinds of labels to identify nodes. Fig. 1 shows a five vertex graph and the computation tree which results if the pivots are chosen in lexicographic order. The cliques and independent sets found by RAMSEY are closely related to the paths in this tree. Given any path from the root to a leaf, the leaf and the parents of all left edges form a clique. Similarly, the leaf and the parents of all right edges form an independent set. Thus, the size of the largest clique plus the size of the largest independent set limits the maximal possible path length.

The algorithm as described by Boppana and Halldórsson is deterministic. The pivots are selected according to some predefined ordering (e.g. lexicographic). It is relatively easy to construct graphs (together with an ordering of the vertices) on which the algorithm performs badly. This simple construction which depends on the fact that the algorithm chooses the vertices in the order given, breaks down if the pivot vertices are chosen at random. In this paper, we analyze a randomized version of the algorithm in which RAMSEY chooses the pivots at random, i.e. in each recursive call the pivot is chosen uniformly at random from the vertex set of the input graph to the recursive call. We call the randomized RAMSEY subprocedure R-RAMSEY. Furthermore, we allow polynomial amplification, i.e. we analyze a procedure PAR-RAMSEY, which calls R-RAMSEY $n^{O(1)}$ times and returns the largest clique and the largest independent set found in all runs. Thus, if R-RAMSEY finds a clique of a certain size with probability at least n^{-k} for some $k \in \mathbb{N}$, then PAR-RAMSEY will return a clique of that size with probability arbitrarily close to one. Finally, let PAR-IS-EXCLUSION denote the subgraph exclusion procedure which calls PAR-RAMSEY instead of RAMSEY. The main result of this paper is:

THEOREM 1. *There is a function $h(n) < n^{o(1)}$ such that for all $\alpha \in (0, 1/2)$ and infinitely many $n \in \mathbb{N}$ there are graphs G of size $|V(G)| = n$ with cliques of size n^{α} such that*

$$\mathbf{P}(\text{PAR-IS-EXCLUSION}(G) \text{ finds a clique larger than } h(n)) < n^{-\omega(1)}$$

Furthermore, for infinitely many n there is a polynomial-time computable distribution $\mathcal{D}_n$ on graphs of size n such that the statement remains true even if the input graph G is a random variable with distribution $\mathcal{D}_n$.

Thus, we show that there are graphs on which PAR-IS-EXCLUSION does not approximate the maximum clique better than $\Theta(\sqrt{n})$. Furthermore, we construct efficiently computable distributions such that almost every graph sampled from these distributions has this property.

1.3 Roadmap

Most of this paper is dedicated to constructing hard graphs for the randomized Ramsey subprocedure. The last section shows that these graphs remain hard even if the subgraph exclusion procedure is added to the algorithm. The proof for R-RAMSEY is divided into a part which establishesa graph property that guarantees graphs to be hard for the algorithm (sections 3 and 4) and a second part (section 5) which contains the construction of a class of graphs which has this property.

Section 2 is an informal description of the main ideas of the construction. In section 3, we define a Markov chain whose state space was chosen to model the size of the clique found by the algorithm as it progresses. The transition probabilities are closely related to the probability that the algorithm finds a useful vertex in the next step. We go on to show that it is extremely unlikely that the Markov chain will, within the given number of steps, reach a state which would correspond to a non-negligible clique size.

In section 4, we define a graph property and show that if the input graph possesses this property, RAMSEY will, with very high probability, find only a negligibly small clique. The central part of the corresponding proof shows the correspondence between the behavior of the algorithm and the Markov chain and uses the result of section 3.

What remains to be done is to construct graphs which contain large cliques and which have the property defined in section 4. Section 5 defines a class of graphs and shows that it has these properties. This completes the analysis for the randomized Ramsey subprocedure. Finally, section 6 shows that the constructed graphs continue to be hard even if the subgraph exclusion procedure is added.

2. An Example

The purpose of this section is to give some intuition about the construction. We describe how the algorithm is likely to behave on an input graph which is somewhat simpler than those described in this paper; yet shares most of the properties which are needed to make the construction work. We focus on these properties.

The graph class considered in this section are random graphs $G = (V, E)$ with large embedded cliques. Generate G as follows: Given the vertex set V of size n and the size l of the clique to be embedded, randomly select a

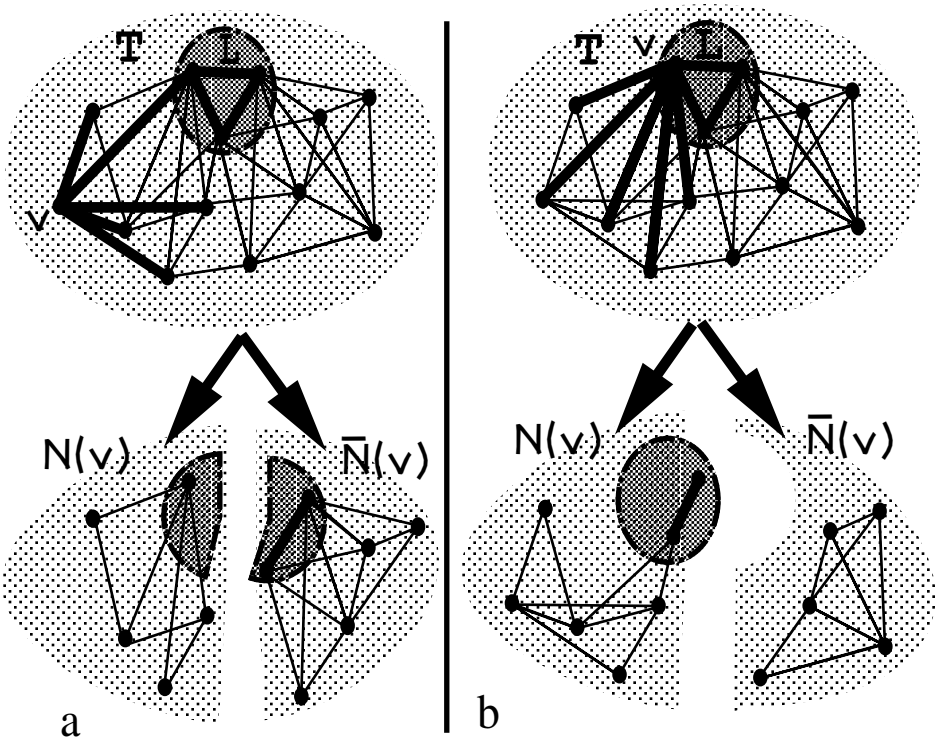

Fig. 2: The behavior of R-RAMSEY when run on a random graph with a large embedded clique: top: the input; bottom: the inputs to the two recursive calls made by R-RAMSEY ($\mathcal{N}(v)$ and $\bar{\mathcal{N}}(v)$); a) the pivot is in T. b) the pivot is in L.

subset L of V of size $l = |L|$ and force it to be a clique by putting all edges in $\{\{v_i, v_j\} : v_i, v_j \in L, v_i \neq v_j\}$ into E. Determine all other edges by random independent coin flips. Let $T = V \setminus L$ be the vertices not belonging to the embedded clique. We note that graphs of this kind form the 'skeleton' of the graphs constructed in this paper and that they share most of the relevant properties. We fix l to be $l = n^\alpha$ for constant α such that $0 < \alpha < 1$ and go on to examine these properties.

Note the main difference between T and L: G restricted to T is a random graph and, thus, does not contain any cliques larger than approximately $2 \log n$.[1] G restricted to L on the other hand is a clique of size $n^\alpha \gg 2 \log n$. Any cliques the algorithm might find in T are small and practically irrelevant. The goal is to ensure that the algorithm does not find any large subcliques of L. For this, it is necessary that the probability of selecting a vertex from L is small whenever the algorithm randomly picks a vertex in line 2 of R-RAMSEY.

[1] We refer to several properties which are satisfied with very high probability. In order to simplify the exposition in this section, we will only name these properties and omit the statement 'with high probability'.

Initially, L is much smaller than T. In particular, $|L| < |V|/n^{\epsilon}$ ($\epsilon > 0$) so that the probability that the randomly chosen vertex is in L is at most $n^{-\epsilon}$. Much of the analysis in this paper is needed to show that this probability is not likely to become larger than $n^{-\delta}$ for some $\delta > 0$ as recursive calls are made and the input graphs to these recursive calls are considered. For now, let us consider the behavior of R-Ramsey on an input graph as in Fig. 2.

Assume that the graph in the top half of the figure is the input to R-Ramsey. The round, dark shaded region represents the clique L. Since the graph is not empty, R-Ramsey randomly picks a pivot vertex v. Fig. 2a illustrates the case $v \in T$ which is far more likely since T is much larger than L. The edges between v and any other vertex in the graph were determined independently at random with probability 1/2. Therefore, v is adjacent to approximately half the vertices in T and half the vertices in L. In other words, the graphs induced by $\mathcal{N}(v)$ and $\bar{\mathcal{N}}(v)$ have about the same size and ratio of vertices from L and vertices from T. These two graphs (Fig. 2a bottom) are the inputs to the recursive calls made in lines 3 and 4. As the recursive depth increases by one, it is important to note that (a) the input graphs to the next recursive call have the same structure: they are random graphs with embedded cliques; (b) the ratio between vertices from L and T remains essentially unchanged and, most importantly, (c) the largest clique in the input to either of the two recursive calls (lines 3 and 4) has been cut in half. Thus, $\Theta(\log n)$ steps of this kind would be sufficient to completely destroy the large clique L.

Fig. 2b illustrates the far less likely case $v \in L$. Since L is a clique, $L \setminus \{v\}$ is in the neighborhood of L. T, on the other hand, is split about evenly between $\mathcal{N}(v)$ and $\bar{\mathcal{N}}(v)$ because the edges between v and T were chosen independently at random. Thus L has been isolated from about half of T. This is illustrated in the bottom part of Fig. 2b. In the recursive call in line 3 which searches $\mathcal{N}(v)$, the probability of selecting a vertex from L has doubled. If this were to happen too often ($\Omega(\log n)$ times), finding a large subset of L would become easy. However, this event is extremely unlikely.

Note that independently of the choice of the pivot v, the graphs induced by $\mathcal{N}(v)$ and $\bar{\mathcal{N}}(v)$ do not change in their structure: they are random graphs with (possibly empty) embedded cliques. Therefore, the analysis can be applied recursively. It should now be clear that at any given recursive call with input graph (V', E'):

$$\frac{|L \cap V'|}{|V'|} = \frac{|L \cap V'|}{|(L \cap V') \cup (T \cap V')|} \approx \frac{|L|/2^{i-k}}{|L|/2^{i-k} + |T|/2^{i}} = \left(1 + \frac{n^{\epsilon}}{2^{k}}\right)^{-1} \quad (1)$$

where i is the depth of the recursive call and k is the number of vertices from L on the path from the root of the computation tree to the current recursive call. We write '$\approx$' instead of '$=$' in the second step of Eq. (1) because of small deviations likely to occur in random graphs. The expression $|L \cap V'|/|V'|$ represents the probability of finding a vertex from L when the algorithm randomly chooses a vertex from V'.

So far, we have relied on random graph properties which hold with very high probability as long as the graphs are sufficiently large. These properties control the split ratios when a pivot v is chosen: If $v \in L$ then (with high probability p) T is split approximately in half between $\mathcal{N}(v)$ and $\bar{\mathcal{N}}(v)$. If $v \in T$ then, in addition, L is also split in half (with high probability p). The probability p depends on the size m of the input graph to the recursive call. At the deeper recursive levels, the input graphs to the recursive calls become small (size $m \leq \log^{O(1)} n$). The probability that such a graph (of size m) does not have the required properties increases to $1/q(n)$ where q is polynomial in n. Thus the analysis cannot be applied to recursive calls whose input graphs are too small. However, this is not a problem because these graphs are so small that they cannot contain any significant number of vertices from L.

3. The Markov chain

Assume R-RAMSEY is run on an input graph $G = (V, E)$ and let L and T be two sets such that $L \cup T = V$ and $L \cap T = \emptyset$. Consider any path in the computation tree. The path is identified by a sequence of pivot vertices. In the spirit of the previous section, we want to limit the number of vertices from L in the path. We begin by modeling the event '$v \in L$' as a random variable.

DEFINITION 1. *Given $\epsilon > 0$ and a constant $f > 1$, let $(X_i)_{i \in \mathbb{N}}$ be $\{0,1\}$-random variables whose distribution is bounded by*

$$\mathbf{P}(X_i = 1 | \sum_{j=1}^{i-1} X_j = k) \leq q_k = \begin{cases} (1 + n^{\epsilon}/f^k)^{-1} & \text{if } k \leq \log^{1-\epsilon/4} n \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

Let $\mathcal{SP}_f$ denote this class of stochastic processes.

For the rest of this paper, the constant f can be assumed to be $f = 2 + \epsilon$ for some arbitrarily small constant ϵ. Let v_i be the i-th vertex in the path. Interpret $X_i = 1$ as the event '$v_i \in L$' and $X_i = 0$ as '$v_i \in T$', for i smaller than the path length. Then $\mathbf{P}(X_i = 1 | \sum_{j=1}^{i-1} X_j = k)$ represents the probability of $v_i \in L$ given that there are k vertices from L before i in the path. The q_k have been chosen so as to be manageable upper bounds on this probability. Notice that for $k \leq \log^{1-\epsilon/4} n$, q_k is essentially the expression in Eq. (1).

DEFINITION 2. *For $i, j \in \mathbb{N}$ let*

$$p_{ij} = \begin{cases} q_i & \text{if } j = i + 1 \\ 1 - q_i & \text{if } j = i \\ 0 & \text{otherwise} \end{cases}$$

The state space $\mathbb{N}$ together with the transition probabilities $(p_{ij})_{i,j \in \mathbb{N}}$ define a Markov chain $\mathcal{MC}_f$. Let the initial distribution be concentrated on state 0

and let the random variables $(Y_i)_{i\in\mathbb{N}}$ *denote the state of the Markov chain after* i *transitions.*

We simplify the analysis by approximating the sums of X_i by the Markov chain just defined. The intuition is the same. The state of the Markov chain corresponds to the number of vertices from L in the path so far. It can be shown by induction on i that for all $x \in \mathbb{R}$ and $i \in \mathbb{N}$

$$\mathbf{P}(\sum_{j=1}^{i} X_j > x) \leq \mathbf{P}(Y_i > x) \tag{3}$$

The following Lemma is the key in the proof that the number of vertices from L in any path is likely to remain small.

LEMMA 1. *Let* $f > 1$ *(*f *constant),* $(X_i)_{i\in\mathbb{N}} \in \mathcal{SP}_f$ *and* $\epsilon \in (0,1)$*. For all* $h \in \mathbb{N}$ *there exists an* $N_0 \in \mathbb{N}$ *such that for all* $n \geq N_0$*:*

$$\mathbf{P}(\sum_{i=1}^{Z} X_i > \log^{1-\epsilon/4} n) < \frac{1}{n^h} \tag{4}$$

provided $Z \leq n^{\epsilon/3}$*.*

PROOF. Because of Eq. (3), it is sufficient to consider Y_Z instead of $\sum_{i=1}^{Z} X_i$. Define the random variables T_i ($i \in \mathbb{N}$) as the number of steps the Markov chain spends in state i provided that it reaches that state. Note that for $m > 0$, $\mathbf{P}(T_i < m) = 1 - p_{ii}^m$ because

$$\mathbf{P}(T_i < m) = \sum_{j=0}^{m-1} \mathbf{P}(T_i = j) = \sum_{j=0}^{m-1} p_{ii}^j (1 - p_{ii}) = 1 - p_{ii}^m.$$

Now, for $k = 1 + \log^{1-\epsilon/4} n$

$$\begin{aligned}
\mathbf{P}(Y_Z \geq k) &= \mathbf{P}(\sum_{i=1}^{k-1} T_i < Z) \leq \mathbf{P}(\bigcap_{i=1}^{k-1} \{T_i < Z\}) = \prod_{i=1}^{k-1} \mathbf{P}(T_i < Z) \\
&= \prod_{i=1}^{k-1} (1 - p_{ii}^Z) \leq \prod_{i=1}^{k-1} \left(1 - \frac{1}{(1 + \frac{f^i}{n^\epsilon})^Z}\right) \leq \prod_{i=1}^{k-1} \left(1 - \mathrm{e}^{-\frac{Z f^i}{n^\epsilon}}\right) \\
&\leq \left(1 - \mathrm{e}^{-\frac{Z f^k}{n^\epsilon}}\right)^{k-1} \leq \left(1 - \mathrm{e}^{-\frac{1}{n^{\epsilon/2}}}\right)^{\log^{1-\epsilon/4} n} \\
&< \left(\frac{2}{n^{\epsilon/2}}\right)^{\log^{1-\epsilon/4} n} \leq n^{-\epsilon/4 \log^{1-\epsilon/4} n}
\end{aligned}$$

As n grows, the exponent goes toward minus infinity. The last three steps are valid for sufficiently large n. The step from $1 - \mathrm{e}^{-\frac{1}{n^{\epsilon/2}}}$ to $\frac{2}{n^{\epsilon/2}}$ follows by considering the limit as $n \to \infty$ of the quotient of the two functions and applying l'Hospital's rule. □

4. The Graph Property

We return now to a level of detail which includes the properties of the PAR-RAMSEY algorithm. Consider any graph $G = (V, E)$ and $L \subseteq V$. For $C, D \subseteq V$, $C \cap D = \emptyset$, let $\mathcal{N}_{CD}^G = \mathcal{N}(C) \cap \bar{\mathcal{N}}(D)$. We call the induced subgraph of G whose vertex set is $\mathcal{N}_{CD}^G$ the (C, D)-*induced subgraph of* G. Assume RAMSEY is run on G. Consider any node x in the computation tree and the vertices on the path that leads to node x. Let C be the set of those vertices in the path at which it turns to the left (neighborhood) and let D be those vertices at which the path turns to the right (non-neighborhood). The input graph to the recursive call corresponding to node x consists of the vertices which are adjacent to all vertices in C and non-adjacent to all vertices in D. This graph is exactly the (C, D)-induced subgraph of G.

Furthermore, let

$$\mathcal{C}_G = \{(C, D) \subseteq V^2 : C \cap D = \emptyset \text{ and } |L \cap (C \cup D)| < \log^{1-\epsilon/4} n\} \tag{5}$$

$\mathcal{C}_G$ identifies the class of pairs $(C, D) \in V^2$ whose (C, D)-induced subgraphs must have the property which will be defined in this section. A pair (C, D) corresponds to a set of potential paths. Since no vertex can appear more than once in a path, $\mathcal{C}_G$ can be restricted to (C, D) such that $C \cap D = \emptyset$. Furthermore, we will see that (C, D) which contain many ($\geq \log^{1-\epsilon/4} n$) vertices from L do not have to be considered because any such path is extremely unlikely. In fact, $\mathcal{C}_G$ could be restricted even further – requiring C to be a clique and D to be an independent set – since this has to be the case for any path leading to $\mathcal{N}_{CD}^G$. We ignore this restriction because it would not simplify the analysis. If G and L are clear from the context, we write $\mathcal{C}$ and $\mathcal{N}_{CD}$ instead of $\mathcal{C}_G$ and $\mathcal{N}_{CD}^G$.

Now, we define hardness in terms of a set of criteria on a graph. These hardness criteria are a more precise formulation of the intuition given in section 2. It will be proved using Lemma 1 that these conditions are sufficient to guarantee that PAR-RAMSEY is very likely to find only very small cliques.

DEFINITION 3. *Given $\epsilon > 0$, $f > 1$ and g, a pair (G, L), where $G = (V, E)$ is a graph and $L \subseteq V$, is called* **hard** *if*

(1) the size of the largest clique plus the size of the largest independent set in G restricted to $V \setminus L$ is less than $n^{\epsilon/3} - 2\log^{1-\epsilon/4} n$, and

(2) for all $(C, D) \in \mathcal{C}_G$

$$|L \cap \mathcal{N}_{CD}| < g \quad \text{or}$$
$$|L \cap \mathcal{N}_{CD}| n^{\epsilon}/f^k < |\mathcal{N}_{CD} \setminus L|$$

where $k = |L \cap (C \cup D)|$

Let $\mathcal{D}_n$ be a probability distribution on pairs (G, L) with $|V(G)| = n$. Given an infinite index set $I \subseteq \mathbb{N}$, the (infinite) sequence of distributions $\mathcal{D}_I =$

$(\mathcal{D}_n)_{n\in I}$ is called **hard** *if given some constant $\epsilon \in (0,1)$, $f > 1$, $g(n) < n^{o(1)}$, for all $n \in I$:*

$$\mathbf{P}(\ (G,L) \text{ is hard}) > 1 - n^{-\omega(1)} \tag{6}$$

where (G,L) is a random variable with distribution $\mathcal{D}_n$.

In the pairs (G,L) we construct, L will contain the only large cliques in the graph. The term *hard* refers to the difficulty for RAMSEY of finding a large subset of L. Intuitively, condition 1 in the definition implies that the graph becomes empty after relatively few steps of the algorithm. Condition 2 states that either there are negligibly few (less than g) vertices from L or there are many more vertices not from L than there are from L in the (C,D)-induced subgraph. Furthermore, k denotes the number of vertices from L in the path and f is the constant factor by which the ratio of T-vertices to L-vertices decreases each time an L-vertex is found.

LEMMA 2. *Given constant $\epsilon \in (0,1)$, $f > 1$, $g(n) < n^{o(1)}$ and infinite $I \subseteq \mathbb{N}$, let $\mathcal{S} = \{(G_n, L_n) : n \in I\}$ be a sequence of hard pairs such that $|V(G_n)| = n$. Then, there exists a function $h(n) < n^{o(1)}$ such that for all $(G_n, L_n) \in \mathcal{S}$*

$$\mathbf{P}(\text{PAR-RAMSEY}(G_n) \textit{ finds more than } h(n) \textit{ vertices } v \in L_n) < n^{-\omega(1)} \tag{7}$$

PROOF. Consider $(G,L) \in \mathcal{S}$ ($n = |V(G)|$) and any path in any computation tree of R-RAMSEY(G). Let T_0 be the first node in the path at which $|L \cap V_T| \leq g(n)$, where (V_T, E_T) is the graph associated with T_0. The path can contain at most $g(n)$ vertices $v \in L$ after T_0 and we will show now that with high probability, it does not contain more than $\log^{1-\epsilon/4} n$ vertices $v \in L$ before T_0.

We do so by interpreting the vertices on the path as a stochastic process and applying Lemma 1. Let X_i be the indicator random variable of the event $v_i \in L$, where v_i is the i-th vertex in the path, $i \in \{1, \ldots, T_0\}$. Let V_i be the vertex set of the graph associated with position i and let k_i be the number of $v \in L$ in the path before position i. Clearly, $V_i = \mathcal{N}_{CD}$, where C and D are the sets of neighboring and non-neighboring vertices in the path before position i as defined at the beginning of this section. If $k_i < \log^{1-\epsilon/4} n$ then $(C,D) \in \mathcal{C}$ and since T_0 is the first point in the path at which $|V_i \cap L| < g(n)$, the inequality $|L \cap V_i| n^\epsilon / f^{k_i} < |V_i \setminus L|$ has to hold for all path positions i before T_0 by point 2 of Definition 3. Hence, we have for all induced subgraphs (V_i, E_i) at path positions i before T_0 with $k_i \leq \log^{1-\epsilon/4} n$:

$$\mathbf{P}(X_i = 1 | \sum_{j=1}^{i-1} X_j = k_i) = \mathbf{P}(v \in L | k_i \text{ vertices } v \in L \text{ before } i) = \frac{|L_i|}{|V_i|}$$

$$= \frac{|L_i|}{|L_i| + |V_i \setminus L|} \leq \frac{|L_i|}{|L_i| + |L_i| n^\epsilon / f^k} = \frac{1}{1 + n^\epsilon / f^{k_i}} = q_{k_i}$$

where $L_i = L \cap V_i$ and q_{k_i} is defined in Eq. (2). If $k_i > \log^{1-\epsilon/4} n$, the probability is trivially bounded above by $q_{k_i} = 1$. Thus, the X_i are as in Definition 1 and Lemma 1 can be applied to bound the number of C-vertices before T_0. [2] For all $k \in \mathbb{N}$ and sufficiently large $n \in \mathbb{N}$:

$$\begin{aligned}
\mathbf{P}(\sum_{i=1}^{T_0} X_i > \log^{1-\epsilon/4} n) &= \mathbf{P}(\sum_{i=1}^{T_0} X_i > \log^{1-\epsilon/4} n | T_0 > n^{\epsilon/3}) \mathbf{P}(T_0 > n^{\epsilon/3}) \\
&\quad + \mathbf{P}(\sum_{i=1}^{T_0} X_i > \log^{1-\epsilon/4} n | T_0 \leq n^{\epsilon/3}) \mathbf{P}(T_0 \leq n^{\epsilon/3}) \\
&\leq \mathbf{P}(\sum_{i=1}^{n^{\epsilon/3}} X_i > \log^{1-\epsilon/4} n) + \mathbf{P}(T_0 > n^{\epsilon/3}) \\
&\leq 2\mathbf{P}(\sum_{i=1}^{n^{\epsilon/3}} X_i > \log^{1-\epsilon/4} n) < n^{-k}
\end{aligned}$$

The third step follows because $T_0 > n^{\epsilon/3}$ implies $\sum_{i=1}^{n^{\epsilon/3}} X_i > \log^{1-\epsilon/4} n$. To see this, note that the first condition in Definition 3 implies that the graph is exhausted after at most $k = n^{\epsilon/3} - 2\log^{1-\epsilon/4} n$ vertices $v \notin L$ have been selected, because all vertices in the path at which it turns left form a clique and all those where it turns right form an independent set. If $T_0 > n^{\epsilon/3}$ there have to be more than $n^{\epsilon/3}$ vertices in the path and, since at most $k = n^{\epsilon/3} - 2\log^{1-\epsilon/4} n$ of them can be in $V_i \setminus L$, at least $2\log^{1-\epsilon/4} n$ of the $n^{\epsilon/3}$ vertices must be in L.

The last step follows directly from Lemma 1. Therefore, with high probability, the total number of $v \in L$ in the path is bounded by $h(n) = g(n) + \log^{1-\epsilon/4} n < n^{o(1)}$. Considering all $O(n)$ paths and using polynomial amplification can increase the probability only by a polynomial factor. □

Lemma 2 decouples the construction of a hard graph from the algorithm itself. We can now construct a hard graph for RAMSEY only in terms of the conditions stated in Definition 3.

5. A Class of Hard Graphs

We will now describe a class of graphs which contain large (n^α, $0 < \alpha < 1$) cliques which PAR-RAMSEY will not find with high probability. More formally, we describe an infinite sequence of distributions $\mathcal{MG}_\alpha$ such that if (G, L) is generated according to one of these distributions, L contains

[2] Formally, there is a slight problem, because (X_i) has only been defined for $i \in \{1, \ldots, T_0\}$ while $\mathcal{SP}_f$ is a sequence of X_i for $i \in \mathbb{N}$. However, this is irrelevant since Lemma 1 depends only on X_i for $i \in \{1, \ldots, T_0\}$. To make the statement formally correct one can append any sequence (X_j) $(j > T_0)$ which satisfies Eq. (2) to the (X_i) $(i \leq T_0)$ defined above.

cliques of size n^α. With high probability, the largest clique PAR-RAMSEY finds is smaller than n^δ for all $\delta > 0$. It can be shown that the distribution $\mathcal{G}(n, p, n^\alpha)$ of random graphs with an embedded clique of size n^α (as described in Jerrum [1992] and in section 2 has this property. Here, we describe a somewhat more complicated distribution which in addition can be shown to preserve these properties even if a limited number of independent sets is excluded from the graph (as is done by PAR-IS-EXCLUSION). Still, random graphs with large embedded cliques are the 'skeleton' of these graphs.

DEFINITION 4. *Given an even m, generate (G_s, L_s) as follows, where $G_s = (V_s, E_s)$ is a graph. Let $V_s = \{1, \ldots, m\}$ and let $L_s \subseteq V_s$ be a randomly chosen subset of V_s of size $|L_s| = m/2$. Furthermore, let $\{u, w\} \in E_s$ if $u, w \in L_s$ and $u \neq w$. Determine all other edges of G_s by independent random coin flips with probability* 0.5.

G_s is essentially a random graph of size m with a built-in clique of size $m/2$. We will expand each vertex of G_s into a subgraph by means of a generalized version of the graph product (or graph composition [Garey and Johnson 1979]). Our definition of the graph product $\otimes$ is similar to the one used in Chang *et al.* [1994]. However, we replace vertices from L_s and vertices from $V_s \setminus L_s$ by different graphs.

DEFINITION 5. *Given a graph $G_1 = (V_1, E_1)$, $L_1 \subseteq V_1$ and two graphs G_2, G_3, define $(H, L) = (G_1, L_1) \otimes (G_2, G_3)$ as follows: The graph H is constructed by replacing each vertex $v \in L_1$ of G_1 with a copy of G_2 and by replacing each vertex $v \notin L_1$ of G_1 with a copy of G_3. Furthermore, for each edge $\{u, v\}$ in G_1, each vertex in the copy of G_2 or G_3 replacing u is connected to every vertex in the copy of G_2 or G_3 replacing v. The vertices in L are the vertices of H which are generated by replacing L_1 by copies of G_2.*

DEFINITION 6. *Given an even $m \in \mathbb{N}$ and $\alpha \in (0, 1)$, generate (G_s, L_s) of size m according to Definition 4. Let G_2 be an independent set of size $\lfloor 5\frac{1-\alpha}{\alpha} \log^2 m \rfloor$ and let G_3 be a random graph of size $\lceil (m/2)^{(1-\alpha)/\alpha} \rceil$ with edge probability 1/2. Let $H = (G_s, L_s) \otimes (G_2, G_3)$. The probability distributions of the components of H determine the distribution $\mathcal{MG}_{n,\alpha}$ of the graphs H, where n, the size of H, is uniquely determined by m. For fixed α let $\mathcal{MG}_\alpha$ denote the sequence $(\mathcal{MG}_{n,\alpha})_n$ of distributions.*

Fig. 3 illustrates the construction of H. We call the copies of G_2 and G_3 which make up H, the **segments** of H. Given any enumeration of segments which are copies of G_2, let L_i denote the i-th segment. Similarly, let T_i be the i-th segment corresponding to a copy of G_3. L is the union of the L_i. Denote the union of all T_i by T. If each segment were collapsed into one vertex, the result would be the skeletal graph G_s. Let n be the number of vertices in H. It is easy to see that $n = n(m) = (m/2)^{1/\alpha}(1 + o(1))$ and thus

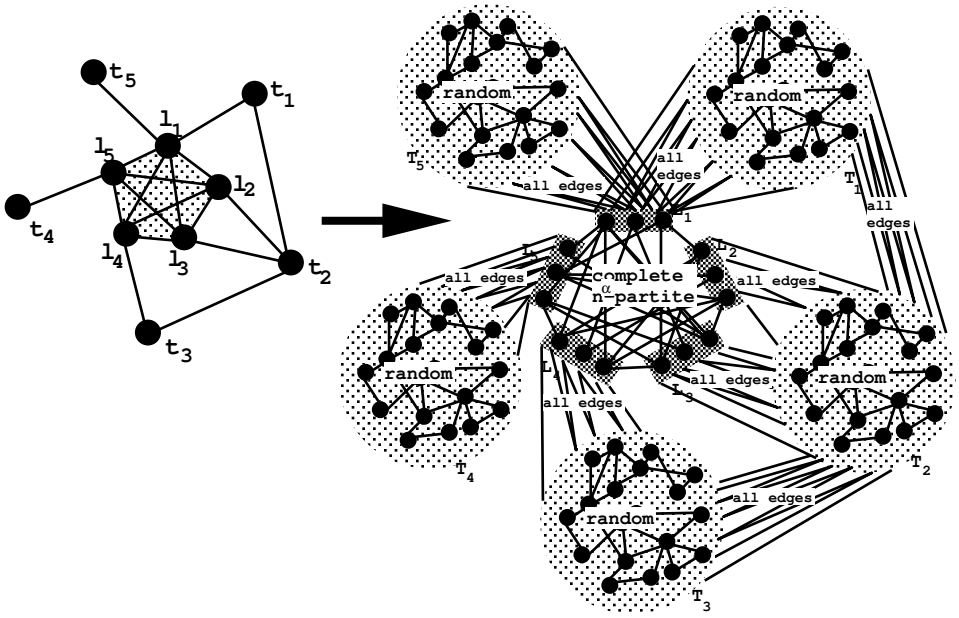

Fig. 3: The construction of H. Left: the skeletal graph G_s; Right: the graph H

$m/2 \leq n^\alpha \leq m$. For each $\alpha < 1$ there is an $\epsilon > 0$ such that $|L|/|V| < n^{-\epsilon}$. Let $I = \{n(k) : 2k \in \mathbb{N}\}$; i.e. I is the set of graph sizes n for which $\mathcal{MG}_{n,\alpha}$ is defined.

Consider a pair $(G, L) \in \mathcal{MG}_{n,\alpha}$. Each segment L_i consists of at most $5\frac{1-\alpha}{\alpha}\log^2(4n^\alpha) \leq 2\log^2 n$ vertices from L and each segment T_i consists of significantly more ($\Theta(n^\epsilon)$) vertices from T. L forms a complete $\geq n^\alpha$-partite graph. Hence, every $v \in L$ is a member of several cliques of size apprxomately n^α. All vertices of the same segment are connected to the same T_j's and L_j's. Thus, whenever a vertex is selected, the graph is split along segment lines. If the pivot is in L_i then the set of T-segments is split between the neighborhood and non-neighborhood whereas all L-segments will be in the neighborhood. It is crucial to ensure that there are enough T-segments in the neighborhood. This is achieved by the random skeletal graph.

The proof that $\mathcal{MG}_\alpha$ is hard for all $\alpha < 1$ is contained in the next three subsections. In section 5.1 we bound the sizes of neighborhoods of vertex sets in the skeletal graph G_s. In section 5.2, we extend this to a bound on the number of segments in the neighborhood of a vertex set in the graphs in $\mathcal{MG}_{n,\alpha}$. Finally, in section 5.3 we use this bound to prove that $\mathcal{MG}_\alpha$ is hard.

5.1 Neighborhood Sizes in the Skeletal Graph

For the rest of this section, let β, γ be constants such that $0 < \beta < 1 < \gamma$. Let $r = r(n) = (1 - 3\log\log n/\log n)\log n$. Given $G_s = (V_s, E_s)$ with n vertices and L_s as in Definition 4, let

$$\mathcal{C}_{r(n)} = \{(C, D) \subseteq V_s^2 : C \cap D = \emptyset \text{ and } |C \cup D| < r(n)\} \tag{8}$$

and, for $C, D \subseteq V_s$ let

$$Y_{CD}^s = |\mathcal{N}_{CD}^{G_s} \cap T_s| \quad \text{and} \quad Z_{CD}^s = |\mathcal{N}_{CD}^{G_s} \cap L_s| \tag{9}$$

where $T_s = V_s \setminus L_s$, Y_{CD}^s is the number of $v \in T_s$ in the (C, D)-induced subgraph of G_s and Z_{CD}^s is the number of $v \in L_s$ in $\mathcal{N}_{CD}$. The result of this subsection is a bound on Y_{CD}^s and Z_{CD}^s for all $(C, D) \in \mathcal{C}_{r(n)}$. We have to limit the size of $C \cup D$ by $r(n)$ because Lemma 3 is not true for larger sets.

LEMMA 3. *Let (V_s, E_s), $|V_s| = n$, be a graph generated as in Definition 4. Then, as $n \to \infty$,*

$$\mathbf{P}\left(\forall (C, D) \in \mathcal{C}_{r(n)} : Y_{CD}^s > \beta\frac{n}{2}2^{-|C\cup D|}\right) > 1 - 2^{-\Theta(\log^3 n)} \quad \textit{and} \tag{10}$$

$$\mathbf{P}\left(\forall (C, D) \in \mathcal{C}_{r(n)} : Z_{CD}^s < \gamma\frac{n}{2}2^{-|(C\cup D)\setminus L_s|}\right) > 1 - 2^{-\Theta(\log^3 n)} \tag{11}$$

PROOF. Consider the first statement. Let $A_{CD}^{(G)}$ be the event $\{G : Y_{CD}^s \leq \beta(n/2)2^{-|C\cup D|}\}$. Then

$$\mathbf{P}(\exists (C, D) \in \mathcal{C}_{r(n)} : Y_{CD}^s \leq \beta\frac{n}{2}2^{-|C\cup D|}) = \mathbf{P}(\bigcup_{(C,D)\in\mathcal{C}_{r(n)}} A_{CD}^{(G)})$$

$$\leq \sum_{(C,D)\in\mathcal{C}_{r(n)}} \mathbf{P}(A_{CD}^{(G)}) \leq |\mathcal{C}_{r(n)}| \cdot \max_{(C,D)\in\mathcal{C}_{r(n)}} \mathbf{P}(A_{CD}^{(G)})$$

It is easy to see that $|\mathcal{C}_{r(n)}| < 2^{O(\log^2 n)}$ since $|C \cup D| < r(n)$ for all $(C, D) \in \mathcal{C}_{r(n)}$.

For the bound on the probability of $A_{CD}^{(G)}$, consider any $(C, D) \in \mathcal{C}_{r(n)}$. For $v \in T_s$, let X_v be the indicator random variable of the event $v \in \mathcal{N}_{CD}$. Note that $\mathbf{P}(X_v = 1) = 2^{-(|C\cup D|)}$ because all edges between v and $C \cup D$ exist independently with probability 0.5. Furthermore, $Y_{CD}^s = \sum_{v\in T_s\setminus(C\cup D)} X_v$, i.e. Y_{CD}^s is the number of ones in a Bernoulli trial of length $|T_s \setminus (C \cup D)| = n/2 - x$ with $p = 2^{-|C\cup D|}$, where $x = |T_s \cap (C \cup D)|$. Note that $x < r(n)$. We can use Chernoff bounds [Hagerup and Rüb 1989] to estimate Y_{CD}^s.

$$\max_{(C,D)\in\mathcal{C}_{r(n)}} \mathbf{P}\left(Y_{CD}^s \leq \beta\frac{n}{2}2^{-|C\cup D|}\right)$$

$$\leq \max_{(C,D)\in\mathcal{C}_{r(n)}} \left(\frac{(n/2 - x)p}{\beta\frac{n}{2}p}\right)^{\beta(n/2)p} \mathrm{e}^{\beta(n/2)p - (n/2-x)p}$$

$$\begin{aligned} &\leq \max_{(C,D)\in\mathcal{C}_{r(n)}} \left(\frac{1}{\beta}\right)^{\beta(n/2)p} \mathrm{e}^{\beta(n/2)p-(n/2)p} n \\ &< n\left(\frac{1}{\beta^{\beta}\mathrm{e}^{1-\beta}}\right)^{(n/2)2^{-r}} \leq 2^{-\Theta(\log^3 n)} \end{aligned}$$

This statement, when combined with the bound on $|\mathcal{C}_{r(n)}|$ proves that

$$\mathbf{P}(\exists (C,D)\in\mathcal{C}_{r(n)} : Y^s_{CD} \leq \beta\frac{n}{2}2^{-|C\cup D|}) < 2^{-\Theta(\log^3 n)}$$

We can show in a similar way that $\mathbf{P}(\exists C,D : (C,D)\in\mathcal{C}_r(n) : Z^s_{CD} \geq \gamma\frac{n}{2}2^{-|(C\cup D)\setminus L|}) < 2^{-\Theta(\log^3 n)}$. This proves the lemma. □

In the graphs in $\mathcal{MG}_\alpha$, the connections between different segments are determined by the edges of the skeletal graph G_s. Thus, $|\mathcal{N}_{G_s}(v)|$ corresponds to the number of segments which are adjacent to segment v. The next lemma maps the random graph property just proved to the graphs in $\mathcal{MG}_\alpha$ (our target graphs).

5.2 Mapping the Target Graph to G_s

Given $(G=(V,E),L)$ generated according to $\mathcal{MG}_{n,\alpha}$ and G's skeletal graph $G_s=(V_s,E_s)$, define the function $f:\mathcal{P}(V)\to\mathcal{P}(V_s)$ as follows:

$$f(S) = \{t_i\in V_s : S\cap T_i\neq\emptyset\}\cup\{l_i\in V_s : S\cap L_i\neq\emptyset\} \tag{12}$$

where t_i (l_i) is the vertex replaced by the segment T_i (L_i respectively). The function f maps a set S of vertices of G to the segments intersected by S. Note the following facts about f:

$$f(A\cup B) = f(A)\cup f(B) \tag{13}$$
$$f(A\setminus L) = f(A)\setminus f(L) \tag{14}$$
$$|f(A)| \leq |A| \tag{15}$$
$$|f(A\cap L)| + |f(A\setminus L)| \geq |f(A)| \tag{16}$$

where A and B are arbitrary subsets of V_s. We need the following notation for the rest of this section: For $C,D\subseteq V$ let

$$Y_{CD} = |\{i : T_i\subseteq\mathcal{N}_{CD}\}| \quad \text{and} \quad Z_{CD} = |\{i : L_i\subseteq\mathcal{N}_{CD}\}|$$

be the number of segments T_i (L_i respectively) contained in $\mathcal{N}_{CD}$. Let

$$\mathcal{C}_f = \{(C,D)\subseteq V^2 : f(C)\cap f(D)=\emptyset \text{ and } |f(C\cup D)| < r(n^\alpha)\} \tag{17}$$

$\mathcal{C}_f$ is the class of pairs (C,D) which intersect less than $r(n^\alpha)$ segments. Furthermore, the set of segments intersected by C is disjoint from the one intersected by D. The following fact is an immediate consequence of the construction of $\mathcal{MG}_\alpha$.

FACT 1. Let $(G = (V, E), L) \in \mathcal{MG}_\alpha$ and let Y^s_{CD} and Z^s_{CD} be the neighborhood sizes of the skeletal graph G_s of G as defined above. Then:

$$\forall (C, D) \in \mathcal{C}_f : Y_{CD} = Y^s_{f(C)f(D)}$$
$$\forall (C, D) \in \mathcal{C}_f : Z_{CD} = Z^s_{f(C)f(D)}$$

PROOF. $T_i \subseteq \mathcal{N}_G(C) \cap \bar{\mathcal{N}}_G(D) \iff t_i \in \mathcal{N}_{G_s}(f(C)) \cap \bar{\mathcal{N}}_{G_s}(f(D))$ Therefore

$$\begin{aligned} Y_{CD} &= |\{i : T_i \subseteq \mathcal{N}_G(C) \cap \bar{\mathcal{N}}_G(D)\}| \\ &= |T_s \cap \mathcal{N}_{G_s}(f(C)) \cap \bar{\mathcal{N}}_{G_s}(f(D))| = |T_s \cap \mathcal{N}^{G_s}_{f(C)f(D)}| \end{aligned}$$

The proof for the second statement proceeds as before. □

It is easy to see that

$$\{(f(C), f(D)) : (C, D) \in \mathcal{C}_f\} = \mathcal{C}_{r(n^\alpha)} \tag{18}$$

LEMMA 4. *For $(G, L) \in \mathcal{MG}_\alpha$:*

$$\mathbf{P}\left(\forall (C, D) \in \mathcal{C}_f : Y_{CD} > \beta \frac{n^\alpha}{2} 2^{-|f(C \cup D)|}\right) < 2^{-\Theta(\log^3 n)} \qquad \textit{and} \tag{19}$$

$$\mathbf{P}\left(\forall (C, D) \in \mathcal{C}_f : Z_{CD} < \gamma \frac{n^\alpha}{2} 2^{-|f((C \cup D) \setminus L)|}\right) < 2^{-\Theta(\log^3 n)} \tag{20}$$

PROOF. Consider the first event. By fact 1 and Eq. (18), it is equivalent to

$$\forall (C, D) \in \mathcal{C}_{r(n^\alpha)} : Y^s_{CD} > \beta \frac{n^\alpha}{2} 2^{-|C \cup D|}$$

Lemma 3 shows that the probability of this event is at least $1 - 2^{-\Theta(\log^3 n)}$.

The proof for the second statement proceeds as before, noting that

$$f((C \cup D) \setminus L) = (f(C) \cup f(D)) \setminus f(L) = (f(C) \cup f(D)) \setminus L_s. \square$$

5.3 The Target Graphs are Hard

LEMMA 5. *For all $0 < \alpha < 1$, the sequence of distributions $\mathcal{MG}_\alpha$ is hard and the L of each pair contains a clique of size n^α.*

PROOF. Consider (G, L) generated according to $\mathcal{MG}_{n,\alpha}$ for $n \in I$. It is easy to see that L contains a clique of size n^α. Pick one vertex from each L_i. There are $m/2 \geq n^\alpha$ segments L_i. Thus, one obtains at least n^α vertices and they form a clique.

Now, we show that with high probability except for L the graph has only small cliques and independent sets (first condition in Definition 3). It is elementary to see that in a random graph $G \in \mathcal{G}(n, 0.5)$ for fixed $\epsilon > 0$, the

size of the largest clique/independent set is less than $(2+\epsilon)\log n$ with probability $> 1 - 2^{-\Theta(\log^2 n)}$. Hence, the size of the largest clique/independent set in each T_i is at most $(2 + o(1))(1 - \alpha)/\alpha \log m$. Furthermore, with probability $> 1 - 2^{-\Theta(\log^2 n)}$, at most $(2 + o(1)) \log m$ segments T_i are totally connected or totally disconnected. Hence, ignoring L, the size of the largest clique plus the size of the largest independent set is at most $8(1 - \alpha)/\alpha \log^2 m + o(\log^2 m) < n^{o(1)}$ with probability $> 1 - 2^{-\Theta(\log^2 n)}$.

It remains to show that (G, L) also satisfies the second condition of Definition 3 with high probability. Lemma 4 guarantees that (G, L) has the following properties with probability $> 1 - 2^{-\Theta(\log^3 n)}$:

$$\forall (C, D) \in \mathcal{C}_f : Y_{CD} > \beta \frac{n^\alpha}{2} 2^{-|f(C \cup D)|} \tag{21}$$

$$\forall (C, D) \in \mathcal{C}_f : Z_{CD} < \gamma \frac{n^\alpha}{2} 2^{-|f((C \cup D) \setminus L)|} \tag{22}$$

Consider any $(C, D) \in \mathcal{C}$ (cf. Definition 3).

Case 1: $(C, D) \in \mathcal{C}_f$. In this case, we can refer directly to Eq. (21) and Eq. (22). The total number $|L \cap \mathcal{N}_{CD}|$ of vertices $v \in L$ in $\mathcal{N}_{CD}$ is less than $\gamma(n^\alpha/2)2^{-|f((C\cup D)\setminus L)|} 2 \log^2 n$. The total number $|\mathcal{N}_{CD} \setminus L|$ of vertices outside L is more than $\beta(n^\alpha/2)2^{-|f(C\cup D)|} n^\epsilon$. Therefore,

$$\frac{|\mathcal{N}_{CD} \setminus L|}{|\mathcal{N}_{CD} \cap L|} \geq \frac{\beta}{\gamma} \frac{n^\epsilon}{2 \log^2 n} 2^{-|(C\cup D)\cap L|}$$

This means that the (C, D)-induced subgraph of (G, L) satisfies the hardness requirement $|L \cap \mathcal{N}_{CD}| n^{\epsilon'}/2^{|(C\cup D)\cap L|} < |\mathcal{N}_{CD} \setminus L|$ where the constant $\epsilon' < \epsilon$ can be chosen arbitrarily close to ϵ.

Case 2: $(C, D) \notin \mathcal{C}_f$ and $f(C) \cap f(D) = \emptyset$. Therefore, $|f(C \cup D)| \geq r(n^\alpha)$. Remember that $(C, D) \in \mathcal{C}$ implies that $|(C \cup D) \cap L| < \log^{1-\epsilon/4} n$, i.e. Definition 3 requires us only to consider pairs (C, D) with less than $\log^{1-\epsilon/4} n$ vertices from L. There exist subsets $C' \subseteq C$ and $D' \subseteq D$ such that $|f(C' \cup D')| = r(n^\alpha) - 1$. Clearly, $Z_{CD} \leq Z_{C'D'}$ and $|(C' \cup D') \cap L| < \log^{1-\epsilon/4} n$. By Eq. (22), we have

$$\begin{aligned} Z_{CD} &\leq Z_{C'D'} < \gamma \frac{n^\alpha}{2} 2^{-|f((C'\cup D')\setminus L)|} \\ &\leq \gamma \frac{n^\alpha}{2} 2^{-(r(n^\alpha) - \log^{1-\epsilon/4} n)} = \frac{\gamma \alpha^3}{2} n^{1/\log^{\epsilon/4} n} \log^3 n \end{aligned}$$

Therefore, the (C, D)-induced subgraph of (G, L) satisfies the requirement $|L \cap \mathcal{N}_{CD}| < g(n)$ for some $g(n) < n^\delta$ (for all $\delta > 0$).

Case 3: $(C, D) \notin \mathcal{C}_f$ and $f(C) \cap f(D) \neq \emptyset$. There exists a vertex $v \in V_s$ such that $v \in f(C)$ and $v \in f(D)$.

If $v \in T_s$, then

$$\mathcal{N}^{G_s}_{f(C)f(D)} \subseteq \mathcal{N}_{G_s}(v) \cap \bar{\mathcal{N}}_{G_s}(v) = \emptyset$$

Therefore, $\mathcal{N}_{CD}^{G} \subseteq T_i$ and, hence $|L \cap \mathcal{N}_{CD}^{G}| = 0 < n^{o(1)}$.

If, on the other hand, $v \in L_s$, then

$$\mathcal{N}_{f(C)f(D)}^{G_s} \cap L_s \subseteq \bar{\mathcal{N}}_{G_s}(v) \cap L_s = \emptyset$$

Therefore, $L \cap \mathcal{N}_{CD}^{G} \subseteq L_i$ and, hence $|L \cap \mathcal{N}_{CD}^{G}| \leq 3\log^2 n < n^{o(1)}$.

In each case, one of the two conditions of Definition 3 (part 2) is satisfied. Thus, $(G, L) \in \mathcal{MG}_{n,\alpha}$ satisfies hardness condition 1 with probability $> 1 - 2^{-\Theta(\log^2 n)}$ and condition 2 with probability $> 1 - 2^{-\Theta(\log^3 n)}$, i.e. (G, L) is hard with probability $> 1 - n^{-\omega(1)}$. □

This concludes the proof that $\mathcal{MG}_\alpha$ is *hard* for PAR-RAMSEY. It remains to show that $\mathcal{MG}_\alpha$ is hard even for PAR-IS-EXCLUSION.

6. The Subgraph Exclusion Algorithm

Let G^i be what remains of G in the i-th iteration of the while-loop of PAR-IS-EXCLUSION(G), i.e. G^i is the input to PAR-RAMSEY after the first $i-1$ independent sets found by PAR-RAMSEY have been removed from G. Let $L^i = L \cap V(G^i)$.

LEMMA 6. *Given $\alpha < 1/2$ and $n \in I$, let $(G, L) \in \mathcal{MG}_{n,\alpha}$, i.e. let (G, L) be generated according to Definition 6. If (G, L) is hard then (G^i, L^i) is hard for all $i \leq n^\alpha$. Furthermore, $L \cap V(G^{n^\alpha}) = \emptyset$.*

PROOF. For the first part, we note that the only properties of the T_i used in the proof of Lemma 5 are $|T_i| > n^\epsilon$ for some constant $\epsilon > 0$ and the fact that the T_i have only small independent sets. Certainly, the size of independent sets in T_i will not increase if vertices are removed. Furthermore, since $\alpha < 1/2$, there exists a constant $\epsilon > 0$ such that initially, $|T_i| = \lceil (m/2)^{(1-\alpha)/\alpha} \rceil = \Theta(n^{1-\alpha}) > \Theta(n^{1/2+\epsilon})$. The largest independent set in each T_i has less than $3\log n$ vertices (for almost every G). Therefore, even after n^α exclusions there will be more than $\Theta(n^{1/2+\epsilon}) - 3n^\alpha \log n = \Theta(n^{1/2+\epsilon})$ vertices left in each T_i. This means that as long as not more than n^α exclusions have occurred, we can prove hardness like we did in Lemma 5.

The proof of the second part of the lemma is based on the following fact: In each of the first n^α exclusions, exactly one entire L_i (and no vertex of any other L_j) will be removed from the graph. Therefore, after the first n^α independent sets have been excluded, L will have disappeared completely from the graph.

To see this, note first that each L_i is an independent set which is larger than any independent set in L^c. In almost every G the largest independent set in the subgraph induced by T will be at most $(4(1-\alpha)/\alpha + o(1))\log^2 m$ (cf. proof of Lemma 5). Since each L_i is an independent set larger than $5(1-\alpha)/\alpha \log^2 m$, any independent set containing an L_i is larger than any independent set not containing an L_i.

Second, note that if there is an L_i in the graph, RAMSEY will find all $v \in L_i$ as an independent set. This is the case simply because all the vertices of each L_i are in the neighborhood of exactly the same set of vertices. Therefore, in the computation tree of RAMSEY(G), there will be one path for each L_i which contains all $v \in L_i$ as parents of right (non-neighbor) edges.

Third, each independent set contains vertices from at most one L_i because all vertices from different L_i are connected by an edge. This completes the proof of the fact.

Points two and three show that the independent set returned by RAMSEY contains either all vertices of an L_i or none of its vertices. Point one shows that as long as there are L_i's in the graph, one of them will be in the independent set returned by RAMSEY. Thus, after n^α exclusions, L has been removed completely and the graph does not contain any clique larger than $O(\log^2 n)$. □

PROOF. **of Theorem 1:** Consider $\mathcal{MG}_\alpha = (\mathcal{MG}_{n,\alpha})_{n\in I}$ for $\alpha < 1/2$. Let $((G_n, L_n))_{n\in I}$ be a sequence of random variables such that (G_n, L_n) has distribution $\mathcal{MG}_{n,\alpha}$. Note that $|V(G_n)| = n$. Let $C_{ex}(G_n)$ be the size of the clique returned by PAR-IS-EXCLUSION(G_n). It has to be shown that there is a function $h(n) < n^{o(1)}$ such that as n grows

$$\mathbf{P}(C_{ex}(G_n) \geq h(n)) < n^{-\omega(1)}$$

Let $H(G_n)$ be the event that (G_n, L_n) is hard (for some $\epsilon > 0$ and $g(n) < n^{o(1)}$). Then

$$\begin{aligned} & \mathbf{P}(C_{ex}(G_n) \geq h(n)) \\ \leq\ & \mathbf{P}(C_{ex}(G_n) \geq h(n) \cap H(G_n)) + \mathbf{P}(C_{ex}(G_n) \geq h(n) \cap H^c(G_n)) \\ \leq\ & \mathbf{P}(C_{ex}(G_n) \geq h(n) | H(G_n)) + \mathbf{P}(H^c(G_n)) < n^{-\omega(1)} \end{aligned}$$

It remains to explain the last step. By Lemma 5, $\mathcal{MG}_\alpha$ is hard and therefore, by Definition 3, $\mathbf{P}(H^c(G_n)) < n^{-\omega(1)}$. As for the other term, observe that $C_{ex}(G_n) = \max_{i<n^\alpha} C_r^i(G_n^i)$ where C_r^i is the clique size returned by the i-th call to PAR-RAMSEY in the while-loop of the subgraph exclusion procedure. By Lemma 6, $H(G_n)$ implies $H(G_n^i)$ for all $i < n^\alpha$. Therefore

$$\begin{aligned} \mathbf{P}(C_{ex}(G_n) \geq h(n) | H(G)) &= \mathbf{P}\left(\bigcup_{i\leq n^\alpha} C_r^i(G_n^i) \geq h(n) | H(G_n) \right) \\ &\leq \sum_{i\leq n^\alpha} \mathbf{P}(C_r^i(G_n^i) \geq h(n) | H(G_n^i)) < n^{-\omega(1)} \end{aligned}$$

where the last step follows from Lemma 2. □

This implies that the *performance ratio* of PAR-IS-EXCLUSION is not better than $\Theta(\sqrt{n})$. The class of graphs which was described in Definition 6 is a particular example of graphs on which the algorithm shows this behavior.

7. Conclusions

We have constructed a class of graphs on which the Boppana-Halldórsson algorithm for MAXCLIQUE has a performance ratio of $\Omega(\sqrt{n})$. We have shown that not even randomizing the algorithm and allowing polynomial amplification can improve the performance ratio on this class of graphs.

Several open problems remain: What is the true performance guarantee of the randomized and polynomially amplified version of the algorithm? Is it better than the performance guarantee of $O(n/\log^2 n)$ of the original algorithm? The main obstacle to improving the lower bound of this paper seems to be the effect of the subgraph exclusions on the graph distribution. It appears to be quite difficult to analyze properties of a random graph after a number of independent sets found by PAR-RAMSEY have been excluded. These independent sets are only partially random and their exact nature is hard to predict. As they are excluded, dependencies arise in the graph and the standard analytical tools can no longer be applied. Ignoring the subgraph exclusion part, several partial results could easily be strengthened by changing parameters such as the size of the embedded clique or the probability of random edges.

Can the graphs be constructed deterministically, i.e. without having to rely on random graph properties? One of the two random graphs used in the construction (T_i) can easily be replaced by a deterministic graph due to Frankl and Wilson [1981] which contains only small cliques and independent sets ($\leq 2^{O(\sqrt{\log n \log\log n})}$). However, finding a deterministic version of the skeletal graph G_s would imply deterministically constructing a graph with no clique and no independent set larger than $O(\log n)$. Finding a polynomial time procedure for this problem has been a long standing open problem [Alon *et al.* 1992]. It appears that while this problem is not solved, major changes to our construction would be needed to make it fully deterministic (and polynomial time).

Acknowledgements

I would like to thank my advisor Steven Homer for his guidance and support.

References

ALON, N., SPENCER, J., AND ERDÖS, P. 1992. *The Probabilistic Method*. Wiley.

ARORA, S., LUND, C., MOTWANI, R., SUDAN, M., AND SZEGEDY, M. 1992. Proof verification and hardness of approximation problems. In *Proceedings 33rd IEEE Symposium on the Foundations of Computer Science*, 14–23.

BELLARE, M. AND SUDAN, M. 1994. Improved Non-Approximability Results. In *Proceedings of the 26th ACM Symposium on the Theory of Computing*, 184–193.

BOPPANA, R. AND HALLDÓRSSON, M. 1992. Approximating Maximum Independent Sets by excluding Subgraphs. *BIT 32*, 180–196.

CHANG, R., GASARCH, W. I., AND LUND, C. 1994. On bounded Queries and Approximation. Tech. Report TR CS-94-05, University of Maryland.

FRANKL, P. AND WILSON, R. M. 1981. Intersection Theorems with Geometric Consequences. *Combinatorica 1*, 4, 357–368.

GAREY, M. R. AND JOHNSON, D. S. 1979. *Computers and Intractability*. Freeman.

HAGERUP, T. AND RÜB, C. 1989. A Guided Tour of Chernoff Bounds. *Information Processing Letters 33*, 305 – 308.

JERRUM, M. 1992. Large Cliques Elude the Metropolis Process. *Random Structures and Algorithms 3*, 4, 347–360.

KUČERA, L. 1991. The greedy coloring is a bad probabilistic algorithm. *Journal of Algorithms 12*, 674–684.