BOLYAI SOCIETY
MATHEMATICAL STUDIES, X

Bollobás kötet
Budapest (Hungary), 1998
Budapest, 2000, pp. 1–62.

Better Bounds for Max Cut

B. BOLLOBÁS and A. D. SCOTT

For a multigraph G, let $f(G)$ be the size of a largest cut of G. We define $f(m)$ to be the minimum of $f(G)$ over graphs of size m, and $f_w(m)$ to be the minimum over multigraphs of size m. For n sufficiently large, and $0 \leq \binom{k}{2} \leq n$, we determine $f(m)$ for $m = \binom{n}{2} + \binom{k}{2}$ and give the extremal graphs. Furthermore, by considering the weighted problem, we determine $f(m)$ and $f_w(m)$ to within an additive constant for every m, and find the extremal graphs for many values of m. This extends independent work of Alon and Halperin.

In the second part of the paper, we turn to the problem of finding efficient algorithms for obtaining large bipartite subgraphs. We give a linear time algorithm that, for a multigraph G with m edges and n vertices, finds a bipartite subgraph with $f_w(m)$ edges. We give an algorithm running in time $O(2^{ck^4} + m + n)$ that finds a bipartite subgraph with at least $m/2 + \sqrt{m/8} + k$ edges if one exists and otherwise provides an optimal partition. We also provide a linear time weak approximation algorithm for $f(m) - m/2 - \sqrt{m/8}$.

In the final part of the paper, we generalize our results to the related problems Max k-Cut and Max Directed Cut.

1. Introduction

The well-known Max Cut problem asks for a largest cut of a graph G. A cut of maximal size clearly corresponds to a bipartite subgraph of maximal size, and we shall use both formulations. Max Cut is NP-hard and has been the focus of extensive study, both from the algorithmic perspective in computer science and the extremal perspective in combinatorics. The extremal problem asks how small a largest bipartite subgraph of a graph with m edges can be, and which graphs achieve this bound. The algorithmic problem asks for efficient algorithms that determine or approximate the maximal size of a bipartite subgraph and that provide large bipartite subgraphs. An important survey of the Max Cut problem is given by Poljak and Tuza [32]; an excellent bibliography from the perspective of combinatorial optimization is given by Laurent [26].

This article is a combination of survey and research paper. We shall indicate some recent progress on the Max Cut problem, from both combinatorial and algorithmic perspectives, and prove a number of new results. The article was originally written in the autumn of 1997 for the Erdős Workshop in Budapest in the summer of 1998. Early in 1998, we became aware of the work of Alon and Halperin [2], who also addressed the extremal Max Cut problem. They determined the recurrence (5) for the extremal function $f_w(m)$; however, they did not determine the extremal graphs or consider the algorithmic aspects of the problem. Although our approach is similar to Alon and Halperin's, we give the details for clarity of exposition, and also so that we can obtain the extremal graphs for (4) and for some cases of (5). In addition, the arguments are used in later sections on algorithms.

For a graph G, let $f(G)$ be the maximal number of edges in a bipartite subgraph of G. For $m > 0$, we define $f(m)$ to be the minimum value of $f(G)$ for graphs G with m edges. As observed by Erdős, $f(m) \geq m/2$. This can be seen by noting that a random bipartition of a graph G gives a cut with expected weight $e(G)/2$. In 1973, answering a question of Erdős, Edwards ([9], see also [10]) proved that

$$f(m) \geq \left\lceil \frac{m}{2} + \sqrt{\frac{m}{8} + \frac{1}{64}} - \frac{1}{8} \right\rceil . \tag{1}$$

We remark that, in fact, we can demand significantly more from a bipartition: it was shown in [6] that every graph G with m edges has a bipartition $V(G) = V_1 \cup V_2$ such that (1) is satisfied and, in addition, for $i = 1, 2$,

$$e(V_i) \leq \frac{m}{4} + \sqrt{\frac{m}{32} + \frac{1}{256}} - \frac{1}{16}. \tag{2}$$

This is an example of a *judicious partitioning result,* in which we demand that *every* class in a vertex partition satisfies some inequality. We remark that the result is best possible when $m = \binom{2l+1}{2}$, in which case K_{2l+1} is the unique extremal graph (for $l \geq 2$). We shall not consider judicious partitions further in this paper; however, a discussion of related results and problems can be found in [7].

The Edwards bound (1) is exact for complete graphs. Thus if $m = \binom{n}{2}$ then $f(m) = f(K_n) = \lfloor n^2/4 \rfloor$. In fact, if $n \neq 4$ then K_n is the unique extremal graph (for $n = 3$, two additional extremal graphs are obtained by taking an edge-disjoint union of two copies of K_3). For other values of m, however, the situation is less simple. Indeed, Erdős [11] conjectured that the difference between $f(m)$ and (1) can be arbitrarily large. This was proved by Alon [1], who showed that there exist $c, c' > 0$ such that, if n is sufficiently large and $m = 2n^2$,

$$f(m) \geq n^2 + \frac{n}{2} + c\sqrt{n} \geq \frac{m}{2} + \sqrt{\frac{m}{8}} + c' m^{1/4}. \tag{3}$$

Alon also showed that, for some $c'' > 0$ and every $m > 0$,

$$f(m) \leq \frac{m}{2} + \sqrt{\frac{m}{8}} + c'' m^{1/4}.$$

Thus the Edwards formula is exact for $m = \binom{n}{2}$ and out by $O(m^{1/4})$ when m is about halfway between $\binom{n}{2}$ and $\binom{n+1}{2}$.

Our first aim in this paper is to determine $f(m)$ exactly for a range of values between $\binom{n}{2}$ and $\binom{n+1}{2}$. Indeed, suppose $m = \binom{n}{2} + \binom{k}{2}$, where n and k are non-negative integers with $0 \leq \binom{k}{2} \leq n$. The (two) graphs obtained by taking an edge-disjoint union of K_n and K_k show that

$$f(m) \leq \left\lfloor \frac{n^2}{4} \right\rfloor + \left\lfloor \frac{k^2}{4} \right\rfloor$$

while any graph obtained by deleting $\binom{n+1}{2} - \binom{n}{2} - \binom{k}{2}$ edges from K_{m+1} shows that

$$f(m) \leq \left\lfloor \frac{(n+1)^2}{4} \right\rfloor.$$

In § 2 we shall prove that, provided m is sufficiently large,

$$f(m) = \min\left\{ \left\lfloor \frac{n^2}{4} \right\rfloor + \left\lfloor \frac{k^2}{4} \right\rfloor, \left\lfloor \frac{(n+1)^2}{4} \right\rfloor \right\}. \tag{4}$$

Furthermore, the graphs we have described include all the extremal graphs unless $k = 4$, when the graphs that can be obtained from an edge-disjoint union of K_n and two copies of K_3 are also extremal.

Note that $\lfloor (n+1)^2/4 \rfloor$ is smaller than $\lfloor n^2/4 \rfloor + \lfloor k^2/4 \rfloor$ if

$$n - \sqrt{\frac{n}{2}} + \varepsilon(n,k) \le \binom{k}{2} \le n$$

where $|\varepsilon(n,k)|$ is $O(1)$. If there is some integer k with $\binom{k}{2}$ in this range, then since $f(m)$ is monotonic increasing and $f\left(\binom{n+1}{2}\right) = \lfloor (n+1)^2/4 \rfloor$, it follows that, for $\binom{n}{2} + \binom{k}{2} \le m \le \binom{n+1}{2}$,

$$f(m) = \left\lfloor \frac{(n+1)^2}{4} \right\rfloor.$$

We obtain the surprising consequence that $f(m)$ is constant on intervals of length up to about $\sqrt{n/2} \approx (m/2)^{1/4}$.

In §3 we turn to the problem of determining $f(m)$ for arbitrary values of m. Our arguments are similar to those of §2, but we are faced with some additional technical difficulties, which make it necessary to consider graphs in which the edges are weighted. For a graph G with edge-weighting w, let $f(G)$ be the maximal weight of a cut of G. Let $f_w(m)$ be the minimum of $f(G)$ over graphs whose edges are weighted with nonnegative integers and have total weight m (or, equivalently, over multigraphs with m edges). It is easily seen that $f_w(m) \le f(m)$. We prove that, for sufficiently large m,

$$f_w(m) = \min\left\{ \left\lfloor \frac{(n+1)^2}{4} \right\rfloor, \left\lfloor \frac{n^2}{4} \right\rfloor + f_w\left(m - \binom{n}{2}\right) \right\}, \tag{5}$$

where the integer n is defined by $\binom{n}{2} \le m < \binom{n+1}{2}$. This provides a recursive formula for $f_w(m)$. (This recurrence was found independently by Alon and

Halperin [2] and implies (4).) To prove a recursion, note that if we were to know $f_w(m)$ for every $m \le m_0$, where m_0 is sufficiently large, then (5) determines $f_w(m)$ for all m. In any case, (5) determines $f_w(m)$ to within an additive constant. Furthermore, by considering K_{n+1} and graphs of form $K_n \cup H$ with $e(H) = m - \binom{n}{2}$, it is easy to see that

$$f(m) \le \min\left\{ \left\lfloor \frac{(n+1)^2}{4} \right\rfloor, \left\lfloor \frac{n^2}{4} \right\rfloor + f\left(m - \binom{n}{2}\right) \right\}.$$

Setting $C = \max_{m<m_0} \left| f(m) - f_w(m) \right|$, it follows that for every $m > 0$,

$$\left| f(m) - f_w(m) \right| \le C.$$

It seems likely that $f(m) = f_w(m)$ for every $m > 0$.

We use the results of §3 in §4, where we return to the problem of finding extremal graphs for Max Cut. Writing $m = \binom{n_1}{2} + \cdots + \binom{n_k}{2}$, where the n_i are nonnegative integers with $\binom{n_i}{2} < n_{i+1}$ for $i < k$, we determine $f(m)$ provided n_{k-1} is sufficiently large: if $\lfloor n_i^2/4 \rfloor + \cdots + \lfloor n_k^2/4 \rfloor < \lfloor (n_i+1)^2/4 \rfloor$ for each i, then $f(m) = \sum_{i=1}^{k} \lfloor n_i^2/4 \rfloor$; we also give the extremal graphs.

In the second part of the paper we concentrate on algorithmic results. Max Cut is NP-hard (see [24], [16]), even for quite restricted classes of graphs ([21], [5]), and it is therefore of interest to find polynomial time algorithms that give large bipartite subgraphs. A number of authors have given algorithms that yield a bipartite subgraph at least as large as that guaranteed by the Edwards bound (1) (see remarks in Section 2, where several algorithms are also discussed; see also [35]). In Section 5 we give a linear time algorithm that is optimal in terms of edge-weight: for any graph with integer edge-weights and total weight m, the algorithm yields a bipartite graph of weight at least $f_w(m)$.

Much recent progress on the Max Cut problem has concerned the existence of good approximation algorithms. Building on results in the theory of probabilistically checkable proofs, Håstad [18] has shown that, for any $\varepsilon > 0$, it is NP-hard to approximate Max Cut within a factor $17/16 - \varepsilon$. On the positive side, Goemans and Williamson [17] have given a 1.1383-approximation algorithm. (For the Max k-Cut problem Kann, Khanna, Lagergren and Panconesi [23] have shown that it is NP-hard to approximate within a factor $1 + 1/34k$; while Frieze and Jerrum [15] have given an algorithm that approximates within a factor $(1 - 1/k + 2\ln k/k^2)^{-1}$.) Furthermore, it is known that good approximation algorithms exist for dense graphs (see [4], [25], [13]). Note that the difficulty for these algorithms lies in recognising and partitioning graphs G for which $f(G)$ is large. Graphs for which $f(G)$ is small can be partitioned by the trivial greedy algorithm that yields a cut of weight at least $w(G)/2$.

In Sections 6 and 7 we concentrate on graphs of weight m for which $f(G)$ is close to $f_w(m)$. Mahajan and Raman [29] have shown that there are algorithms running in time $O(n^3 + m2^{4k})$ and $O(2^{ck^2} + m + n)$ that find a cut of size at least $\lceil m/2 \rceil + k$ in a graph with m edges and n vertices, if such a cut exists. In Section 6, we show that, for any fixed integer k, there is an algorithm running in time $O(2^{ck^4} + m + e)$ th at finds a cut of weight at least $m/2 + \sqrt{m/8} + k$ in a graph with integer edge-weights, e edges and total weight m if such a cut exists and otherwise finds an optimal cut. In Section 7 we concentrate on the quantity $f(G) - m/2 - \sqrt{m/8}$: we note that it is NP-hard to approximate this quantity within a factor $(9/8 - \varepsilon)$, but provide a linear time algorithm that approximates its logarithm.

In the final part of the paper we consider two related problems. In Section 8, we consider the Max k-Cut problem for $k > 2$. We prove versions of our results on bipartitions for the k-partite case. Finally, in Section 9, we consider the problem of finding large bipartite subgraphs of a directed graph, and give some extremal results.

Throughout the paper, we use w for an integer-valued edge-weighting. For disjoint sets of vertices we write $E(X,Y)$ for the set of edges between X and Y, $e(X,Y) = \big|E(X,Y)\big|$ and $w(X,Y) = \sum_{e\in E(X,Y)} w(e)$. We will also sometimes write $e(x,Y)$ and $w(x,Y)$ for $e\big(\{x\},Y\big)$ and $w\big(\{x\},Y\big)$.

Part I: The Extremal Problem

2. Max Cut for graphs

Our main aim in this section is to find the exact value of $f(m)$ for every sufficiently large m of form $\binom{n}{2} \le m = \binom{n}{2} + \binom{k}{2} \le \binom{n+1}{2}$, and determine the extremal graphs. The value of $f(m)$ can also be obtained from the results of the next section and from Alon and Halperin [2]. However, our aim in this section is also to determine the extremal graphs, which turn out to be surprisingly varied.

Note that, for any m, we can obtain an upper bound for $f(m)$ by writing $m = \binom{n_1}{2} + \binom{n_2}{2} + \cdots \binom{n_r}{2}$, where each n_i in turn is chosen to be as large as possible; then by considering $K_{n_1} \cup \cdots \cup K_{n_r}$, it is clear that $f(m) \le \lfloor n_1^2/4 \rfloor + \cdots + \lfloor n_r^2/4 \rfloor$. A straightforward calculation shows that, for every m,

$$f(m) \le \frac{m}{2} + \sqrt{\frac{m}{8}} + (8m)^{1/4} + O(m^{1/8}), \tag{6}$$

while taking $k \approx \sqrt{2n} - 1$ in the theorem below shows that

$$f(m) \ge \frac{m}{2} + \sqrt{\frac{m}{8}} + \big(1 + o(1)\big)\,(8m)^{1/4}$$

for infinitely many values of m.

Theorem 1. *For $n > 5 \times 10^8$, every graph G with*

$$e(G) = \binom{n}{2} + \binom{k}{2}$$

where

$$0 \le \binom{k}{2} \le n - 1$$

satisfies

$$f(G) \ge \min\left\{ \left\lfloor \frac{n^2}{4} \right\rfloor + \left\lfloor \frac{k^2}{4} \right\rfloor, \left\lfloor \frac{(n+1)^2}{4} \right\rfloor \right\}. \tag{7}$$

Furthermore, the extremal graphs are the two graphs obtained by taking an edge disjoint union of K_n and K_k if $\lfloor n^2/4 \rfloor + \lfloor k^2/4 \rfloor \le \lfloor (n+1)^2/4 \rfloor$ and $k \ne 4$; and all graphs obtained by deleting $\binom{n+1}{2} - \binom{n}{2} - \binom{k}{2}$ edges from K_{n+1} if $\lfloor n^2/4 \rfloor + \lfloor k^2/4 \rfloor \ge \lfloor (n+1)^2/4 \rfloor$. If $k = 4$ then the extremal graphs *are obtained by taking an edge-disjoint union of K_n and K_4 (two graphs) or K_n and two copies of K_3 (seven graphs).*

We will make use of several lemmas in our proof of Theorem 1. Lemma 2 is due to Edwards [9]; recently a short proof was given by Erdős, Gyárfás and Kohayakawa [12]. Poljak and Turzík [33] gave an $O(n^3)$ algorithm for finding a bipartite subgraph of the type guaranteed in the lemma; Ngoc and Tuza [30] improved upon this by giving an algorithm running in time $O(m)$. The proof that we give is similar to the proofs of Erdős, Gyárfás and Kohayakawa and of Ngoc and Tuza, but is slightly simpler and also yields an $O(m)$ algorithm.

Lemma 2. *For a connected graph G,*

$$f(G) \geq \frac{e(G)}{2} + \frac{|G|-1}{4}.$$

Proof. Given an ordering of the vertices of G, we can partition $V(G)$ by using the greedy algorithm: at each step a vertex is added to whichever class contains fewer of its predecessors (or to either class if both classes contain the same number). If we write $e(v)$ for the number of predecessors of v that are adjacent to v then $\sum_{v\in V(G)} e(v) = e(G)$, and the size of the bipartite graph between the two vertex classes is at least

$$\sum_{v\in V(G)} \lceil e(v)/2 \rceil = \frac{e(G)}{2} + \frac{k}{2}$$

where k is the number of vertices with an odd number of predecessors.

It is therefore enough to find an ordering of $V(G)$ in which at least $(n-1)/2$ vertices have an odd number of predecessors. For $|G| \leq 1$ this is trivial. If $|G| = n > 1$, then we first find a set of vertices S such that $G[S]$ is a star and $G \setminus S$ is connected. Let T be a spanning tree of G. If two endvertices of T are adjacent, say v and w, then let $S = \{v, w\}$. Otherwise, let T' be the tree obtained by removing all endvertices of T, let v be any endvertex of T' and let S contain v, together with all endvertices of T adjacent to v. In either case, S contains a vertex v_s together with an independent set of neighbours of v_s and $G \setminus S$ is connected. We repeat the process with $G \setminus S$, continuing until at most one vertex remains.

We order the vertices of G one star at a time. Given a star S, let R be the set of vertices we have already ordered. Let S^+ be those vertices of $S \setminus \{v_s\}$ with an odd number of neighbours in R and let $S^- = S \setminus \left(S^+ \cup \{v_s\}\right)$. After R, take S^+ (in any order), followed by v_s, followed by S^- (in any order). Note that the vertices in $S^+ \cup S^-$ all have an odd number of predecessors in this ordering, and $|S^+ \cup S^-| \geq |S|/2$. Thus in the complete ordering, since the stars together contain at least $n-1$ vertices, at least $(n-1)/2$ vertices have an odd number of predecessors.

It is easy to see that this proof gives an algorithm that runs in time $O(m)$. Note that the tree T can be updated efficiently between the removal of successive stars. ■

In the proof above, we can avoid the need to generate a star partition by constructing more directly an ordering of the vertices. Begin with any ordering $v_1 < \cdots < v_n$ of $V(G)$ in which every vertex except v_1 has at least one predecessor. For each vertex calculate the number of predecessors, and let X be the set of vertices with an even number of predecessors. For each $x \in X \setminus v_1$ find the largest predecessor of x. This can clearly all be done in time $O(m)$. Now suppose that two vertices have the same largest predecessor, say v. Reorder $V(G)$ by moving x and y to immediately before v, leaving x and y in the same order. The parity of v does not change, since it has gained two predecessors, while x and y now have an odd number of predecessors. No other vertex has changed its set of predecessors. However, some vertices that previously had x or y as largest predecessor may now have v: we can check this by examining the neighbours of x and y. By a similar argument, if any $x \in X$ has largest predecessor $y \in X$, then moving x to a position immediately before y gives an ordering in which x and y have an odd number of predecessors and only neighbours of x (in the original ordering) can have a new largest predecessor. Repeating this stage of the algorithm, we note that each vertex is moved at most once (when its parity changes), and that we examine the neighbours of a vertex only when we move it (note also that we need only look at successors of a vertex, and no vertex gains successors before being moved). Thus this part of the algorithm runs in time $O(m+n) = O(m)$. Finally, suppose that no two vertices in X have the same largest predecessor and no vertex in X has largest predecessor in X. Now if $x \in X$ has no predecessors then either $x = v_1$, or (since vertices are only ever moved downwards) x must have been moved at some point in the algorithm, which implies $x \notin X$. Thus every $x \in X \setminus v_1$ has a largest predecessor in $V(G) \setminus X$. It follows that we have an injection $X \setminus v_1 \to V(G) \setminus X$, so $|X| \le (n-1)/2 + 1$, and thus at least $n - |X| \ge (n-1)/2$ vertices have an odd number of predecessors. The algorithm is now completed greedily as before.

Note that Lemma 2 furnishes a quick proof of the Edwards formula (1). Indeed, given a graph G, we may assume G is connected or else identify one vertex from each component. Let $n = |G|$, so $e(G) \le \binom{n}{2}$: it follows from the lemma that $f(G) \ge e(G)/2 + (n-1)/4$, and (1) follows by a simple calculation.

Lemma 3 was noted by several authors (see [3], [27], [28], [1]). We prove it here for completeness.

Lemma 3. *For a nonempty graph G,*

$$f(G) \ge \left(\frac{1}{2} + \frac{1}{2\chi}\right) e(G).$$

Proof. Fix a colouring of G with $t = \chi(G)$ colours, and let the colour classes be $V_1, \ldots, V_t$. Let $S \cup T$ be a random partition of $[t]$ into a set of size $\lfloor t/2 \rfloor$ and a set of size $\lceil t/2 \rceil$. Then, writing $m = e(G)$, the expected number of edges between $\bigcup_{i \in S} V_i$ and $\bigcup_{i \in T} V_i$ is

$$\left(\left\lfloor \frac{t^2}{4} \right\rfloor / \binom{t}{2}\right) m \ge \left(\frac{t^2-1}{4} / \binom{t}{2}\right) m = \left(\frac{1}{2} + \frac{1}{2t}\right) m.$$

Therefore some partition satisfies this inequality. ■

Lemma 3 also gives a fast proof of the Edwards formula, as observed independently by Alon [1] and Hofmeister and Lefmann [20].

We will also need a lemma concerning partitions of graphs whose edges are weighted with (positive or negative) integers. Note that as a consequence of this lemma, for $m = \binom{n}{2}$ we obtain the extremal graphs for the formula of Edwards.

Lemma 4. *Let H be a graph whose edges have integer weights. If n is an integer with*

$$w(H) \ge \binom{n}{2}$$

then there is a partition of H into two sets such that the total weight of edges between the sets is at least $\lfloor n^2/4 \rfloor$. For $n \neq 4$, the unique extremal graph is K_n with all edges of weight 1. For $n = 4$ the extremal graphs are K_4 with all edges of weight 1 and the graphs obtained by taking the edge sum of two copies of K_3 with all edges of weight 1.

Proof. The proof that such a partition exists is straightforward, since we may consider H as a weighted complete graph, by adding an edge of weight 0 between every pair of nonadjacent vertices. If H contains an edge with weight at most 0 then contracting that edge does not decrease the weight of the graph. Therefore we may assume that H is a complete graph and all edges have weight at least 1, so $|H| \leq n$. A random partition of $V(H)$ into sets of size $\lfloor |H|/2 \rfloor$ and $\lceil |H|/2 \rceil$ yields a bipartite subgraph of expected weight at least $\lfloor n^2/4 \rfloor$.

To derive the extremal graphs, note first that we can assume that all edges have nonnegative weight, since contracting an edge with negative weight increases the total weight, and we can then do better than $\lfloor n^2/4 \rfloor$ in the argument above (this remark also applies if $w(H) > \binom{n}{2}$). If H is not K_n with all edges of weight 1, then we can contract to a complete graph with at least one edge of weight greater than 1. Thus $|H| < n$ and all edges have weight at least 1. Writing $h = |H|$, a random bipartition into sets of size $\lceil h/2 \rceil$ and $\lfloor h/2 \rfloor$ yields a bipartite subgraph of expected weight at least

$$\frac{\lfloor h^2/4 \rfloor}{\binom{h}{2}} w(H) = \left\lfloor \frac{h^2}{4} \right\rfloor \binom{n}{2} \Big/ \binom{h}{2}. \tag{8}$$

If $h < n-1$ or $h = n-1$ and n is odd then (8) is strictly larger than $\lfloor n^2/4 \rfloor$. Otherwise, $h = n-1$ is odd and, since all edges have weight at least 1, H is the edge sum of K_{n-1} with all weights 1 and a graph H^* with $V(H^*) = V(H)$ and weight $\binom{n}{2} - \binom{n-1}{2} = n-1$. Furthermore, all bipartitions of $V(H)$ into two sets of size $\lfloor h/2 \rfloor$ and $\lceil h/2 \rceil$ yield a bipartite subgraph of size exactly $\lfloor n^2/4 \rfloor$, since otherwise some partition would exceed the expectation (8). Since every bipartition of K_{n-1} into two sets of size $\lfloor (n-1)/2 \rfloor$ and $\lceil (n-1)/2 \rceil$ gives a bipartite subgraph of size $\lfloor (n-1)^2/4 \rfloor = n^2/4 - n/2$, it follows that every bipartition of H^* into sets of size $\lceil (n-1)/2 \rceil$ and $\lfloor (n-1)/2 \rfloor$ gives a bipartite subgraph of size $n/2$.

If H^* is not complete, let v be a vertex that is not adjacent to every other vertex of H^*. We can partition $V(H^*) \setminus v$ into two sets V_1 and V_2

with $|V_1| = |V_2| = (n-2)/2$, such that $\left|\Gamma(v) \cap V_1\right| > \left|\Gamma(v) \cap V_2\right|$. Then a dding v to V_1 or V_2 gives two partitions into sets of size $\lfloor (n-1)/2 \rfloor$ and $\lceil (n-1)/2 \rceil$ that yield bipartite subgraphs with different sizes. It follows that H^* is complete: the only possibility is $n = 4$, $H^* = K_3$ with all edges of weight 1. Thus for $n > 4$, H must be K_n, with all edges of weight 1. For $n = 4$, a simple case check shows that H can also be any graph obtained by taking the edge sum of two copies of K_3 with all edges of weight 1. ∎

Another bound on $f(G)$ was given by Poljak and Turzik [31], who showed that every connected graph G with edge-weighting w has a bipartite subgraph of weight at least

$$\frac{1}{2}w(G) + \frac{1}{4}\min_T w(T)$$

where the minimum is taken over spanning trees T of G. Poljak and Turzik show that there is an algorithm running in time $O(n^3)$ that finds the required subgraph; Poljak and Tuza [32] show that the algorithm runs in time $O(mn)$. We show that there is an $O(m)$ algorithm: note that for unweighted graphs, we obtain a cut of weight at least $e(G)/2 + \left(|G| - 1\right)/4$, thus giving another proof of Lemma 2.

Theorem 5. *There is an algorithm running in time $O(m)$ that finds in every connected graph G with m edges and edge-weighting w a cut of weight at least*

$$\frac{1}{2}w(G) + \frac{1}{4}\min_T w(T).$$

Before proving this theorem we need a lemma. We say that a collection of induced stars or single vertices $S_1, \ldots, S_t$ in a graph G is a *tree-like star-covering* if every vertex in G belongs to some S_i and the graph with vertices $S_1, \ldots, S_t$ and edges between S_i and S_j iff $S_i \cap S_j \neq \emptyset$ is a tree.

Lemma 6. *There is an algorithm running in time $O(m)$ that finds a tree-like star-covering of any connected graph G with m edges.*

Proof. Recall that a rooted spanning tree T of a graph G is a *depth-first search (DFS) tree* if, for every $uv \in E(G)$, either the path from the root r

to u in T contains v or the path from r to v contains u. In other words, if we delete $v \in V(T)$, then the components of $T \setminus v$ containing each of the children of v are not joined by any edge. A DFS tree can be found in time $O(m)$ (see, for instance, [30]).

Let T be a DFS tree in G with root x. For every $v \in V(T)$ that is not an endvertex of T let T_v be the induced star containing v and its children. If v is an endvertex of T then let $T_v = \{v\}$. Then $\{T_v : v \in V(T)\}$ is a tree-like star covering of G. ■

Proof of Theorem 5. Keeping the notation of the proof of Lemma 6, let

$$\mathcal{T}_0 = \{T_v : d_T(x,v) \equiv 0 \bmod 2\}$$

and

$$\mathcal{T}_1 = \{T_v : d_T(x,v) \equiv 1 \bmod 2\}.$$

Each of $\mathcal{T}_0$ and $\mathcal{T}_1$ is a collection of disjoint induced stars and single vertices, and every edge of T is contained in some member of $\mathcal{T}_0$ or $\mathcal{T}_1$. We partition whichever of $\mathcal{T}_0$ and $\mathcal{T}_1$ has the greater weight, say $\mathcal{T}_i$ one star at a time. Suppose we have a partial partition $V_1 \cup V_2$ and wish to partition a star T_v. We greedily assign v to one class and $T_v \setminus v$ to the other so that the weight of the partial partition is increased by at least $w(T_v, V_1 \cup V_2)/2 + w(T_v)/2$. Repeating for all stars, we obtain a cut of weight at least

$$\frac{1}{2}w(G) + \frac{1}{2}w(\mathcal{T}_i) \geq \frac{1}{2}w(G) + \frac{1}{4}w(T).$$

The algorithm clearly runs in time $O(m)$. ■

Finally, it will be useful to have the following remark.

Lemma 7. *If $W \subset V(G)$ and $H = G[W]$ then*

$$f(G) \geq f(H) + \frac{1}{2}\big(e(G) - e(H)\big).$$

Proof. Partition H, then add the remaining vertices from G one at a time to whichever class has fewer neighbours. The resulting partition clearly satsfies the inequality. ■

With these lemmas in hand, we turn to the proof of Theorem 1.

Proof of Theorem 1. Let $n_0 = 5 \times 10^8$. Suppose that $n > n_0$ and G is a graph with

$$m = \binom{n}{2} + \binom{k}{2} < \binom{n+1}{2} \tag{9}$$

edges and

$$f(G) \le \min\left\{ \left\lfloor \frac{n^2}{4} \right\rfloor + \left\lfloor \frac{k^2}{4} \right\rfloor, \left\lfloor \frac{(n+1)^2}{4} \right\rfloor \right\}. \tag{10}$$

We shall prove that G is one of the extremal graphs given in the statement of the theorem. Note that we may assume that G is connected by identifying one vertex from each component.

We begin by showing that G consists of a very large complete graph on $n - O\big(\sqrt{n}\big)$ vertices, together with $O\big(\sqrt{n}\big)$ 'exceptional vertices'.

Clearly $|G| \ge n$ and $k < \sqrt{2n} + \frac{1}{2}$. Note also that $n > 140(n^{3/4} + k + 1)$. If $k < 2$, we are done by Lemma 4; so we may assume $k \ge 2$. Let $\chi = \chi(G)$. Now by (9),

$$\lfloor n^2/4 \rfloor + \lfloor k^2/4 \rfloor \le \frac{m}{2} + \frac{n+k}{4}, \tag{11}$$

and so it follows from (10) that

$$f(G) \le \frac{m}{2} + \frac{n+k}{4}. \tag{12}$$

Therefore, by Lemma 2,

$$|G| \le n + k + 1. \tag{13}$$

Furthermore, Lemma 3 and (12) imply that

$$\frac{m}{2\chi} \leq \frac{n+k}{4}$$

and so by (9)

$$\chi \geq \frac{2m}{n+k} > n - k - 1.$$

Now if $\overline{G}$ contains $2k+2$ independent edges, we can cover $\overline{G}$ by $|\overline{G}| - (2k+2) \leq n-k-1$ edges and vertices, which is equivalent to colouring G with at most $n-k-1$ colours. It follows that $\overline{G}$ contains at most $2k+1$ independent edges and therefore that some set Y of at most $4k+2$ vertices meets all edges of $\overline{G}$. Let $X = V(G) \setminus Y$; then $G[X]$ is complete, and

$$|X| = |G| - |Y| \geq n - 4k - 2. \tag{14}$$

We have partitioned G into a large complete graph $G[X]$ and a small set Y which we shall regard as a set of exceptional vertices. Note that it follows from (14) that any partition of X into two sets of equal size (or sizes differing by 1) corresponds to a bipartite subgraph of $G[X]$ of size

$$\left\lfloor \frac{|X|^2}{4} \right\rfloor \geq \frac{e(X)}{2} + \frac{|X|-1}{4} \geq \frac{e(X)}{2} + \frac{n-4k-3}{4} \tag{15}$$

Throughout the proof, we shall consider partial partitions of $V(G)$ in which we partition Y and some vertices from X, and then extend these to partitions of $V(G)$ in which X is split as evenly as possible.

We now show that every vertex in Y has either very many or very few neighbours in X. Indeed, suppose some $v \in Y$ has $\big|\Gamma(v) \cap X\big| \geq 5k/2 + 2$ and $\big|X \setminus \Gamma(v)\big| \geq 5k/2 + 2$. We partition G as follows. Since $n > n_0$ we have $n > 5k + 4$, so we can find a partition $X = X_1 \cup X_2$ with $|X_1| \leq |X_2| \leq |X_1| + 1$ and

$$\Big|\big|\Gamma(v) \cap X_1\big| - \big|\Gamma(v) \cap X_2\big|\Big| \geq \frac{5k}{2} + 2. \tag{16}$$

Adding v to whichever of X_1 and X_2 contains fewer of its neighbours, it follows from (14) that we obtain a bipartition of $H = G\big[X \cup \{v\}\big]$ with at least

$$\frac{e(H)}{2} + \frac{|X|-1}{4} + \frac{5k}{4} + 1 > \frac{e(H)}{2} + \frac{n+k}{4}$$

edges between the two classes. Thus, by Lemma 7,

$$f(G) \geq f(H) + \frac{1}{2}\big(e(G) - e(H)\big) > \frac{m}{2} + \frac{n+k}{4}, \tag{17}$$

which contradicts (12). We may therefore assume that for every $v \in Y$, either $\big|\Gamma(v) \cap X\big| < 5k/2 + 2$ or $\big|X \setminus \Gamma(v)\big| < 5k/2 + 2$. Let

$$Y^+ = \Big\{v \in Y \,:\, \big|\Gamma(v) \cap X\big| > |X| - 5k/2 - 2\Big\}$$

$$Y^- = \Big\{v \in Y \,:\, \big|\Gamma(v) \cap X\big| < 5k/2 + 2\Big\}.$$

Then $Y^+ \cup Y^-$ is a partition of Y.

Next we show that the subgraph induced by $X \cup Y^+$ is nearly complete. Indeed, we claim that $e\big(\overline{G}[X \cup Y^+]\big) < 5k/2 + 2$. If not, then let $W \subset Y^+$ be minimal such that

$$e\big(\overline{G}[X \cup W]\big) \geq \frac{5k}{2} + 2. \tag{18}$$

Since $\big|X \setminus \Gamma(v)\big| < 5k/2 + 2$ for every $v \in Y^+$, we have

$$5k/2 + 2 \leq e\big(\overline{G}[X \cup W]\big) \leq 5k + 4.$$

Since $n > n_0$, it follows that $n > 4(5k+4)$, so we can find a partition $V_1 \cup V_2$ of $X \cup W$ such that $|V_1| \leq |V_2| \leq |V_1| + 1$ and all the edges of $\overline{G}[X \cup W]$ are contained in V_1. Then since $\binom{|X\cup W|}{2} \geq e(X\cup W) + (5k/2) + 2$, it follows from (18) and (14) that

$$\begin{aligned} f\big(G[X\cup W]\big) &\geq |V_1|\,|V_2| \\ &\geq \left\lfloor \frac{|X\cup W|^2}{4} \right\rfloor \\ &\geq \frac{1}{2}\big(e(X\cup W) + 5k/2 + 2\big) + \frac{|X| + |W| - 1}{4} \\ &> \frac{1}{2}e(X\cup W) + \frac{n+k}{4}, \end{aligned}$$

and we are done, as in (17). Thus we may assume that

$$e\big(G[X\cup Y^+]\big) < \frac{5k}{2} + 2, \tag{19}$$

so $G[X\cup Y^+]$ is nearly a complete graph. In particular,

$$f\big(G[X\cup Y^+]\big) \geq \left\lfloor \frac{|X\cup Y^+|^2}{4} \right\rfloor \tag{20}$$

Now we show that there are not too many edges between Y^- and $X\cup Y^+$. Note first that every vertex $v \in Y^-$ has fewer than $5k/2 + 2$ neighbours in $X\cup Y^+$: otherwise, since $e\big(\overline{G}[X\cup Y^+]\big) < 5k/2 + 2$ and $\big|\Gamma(v)\cap X\big| < 5k/2 + 2$ (and since $|X| > 6(5k/2+2) = 15k + 12$ which, as $n > n_0$, follows from (14)) we can find a partition of $X\cup Y^+$ into sets W_1 and W_2 with $|W_1| \leq |W_2| \leq |W_1| + 1$ such that all edges of $\overline{G}[X\cup Y^+]$ are contained in W_1 and $\big|\,|\Gamma(v)\cap W_1| - |\Gamma(v)\cap W_2|\,\big| \geq 5k/2 + 2$. Arguing in the same way as from (16) we arrive at a contradiction. Thus

$$\big|\Gamma(v)\cap(X\cup Y^+)\big| < \frac{5k+4}{2} \tag{21}$$

for every $v \in Y^-$.

Now suppose that

$$e(Y^-, X \cup Y^+) > 70n^{3/4}.$$

Let $Y_0 \subset Y^-$ be minimal such that $e(Y_0, X \cup Y^+) > 70n^{3/4}$, and let $U = \Gamma(Y_0) \cap (X \cup Y^+)$. Note that the minimality of Y_0 and (21) imply that $|U| < 70n^{3/4} + 5k/2 + 2$. Since $n > n_0$ it follows that $|U| < |X|/2$. Let $Y_0 = Y_1 \cup Y_2$ be a random partition, where each vertex of Y_0 is in Y_1 or Y_2 independently with probability 1/2. Let

$$U_1 = \left\{ u \in U \,:\, \left|\Gamma(u) \cap Y_1\right| > \left|\Gamma(u) \cap Y_2\right| \right\}$$

and let $U_2 = U \setminus U_1$. For $u \in U$, let $d_u = \left|\Gamma(u) \cap Y_0\right|$ and define $\Delta(u) = \left|\left|\Gamma(u) \cap Y_1\right| - \left|\Gamma(u) \cap Y_2\right|\right|$. Then

$$\mathbb{E}\big(\,\Delta(u)\big) = \mathbb{E}\big(\left|\,S(d_u)\right|\big),$$

where we write $S(d_u)$ for the position after d_u steps of a simple symmetric random walk on $\mathbb{Z}$ starting from 0. It is easily checked that $\mathbb{E}\left|\,S(d)\right| \geq \sqrt{d}/2$, and so

$$\begin{aligned}
\mathbb{E}\big(\,e(Y_1, U_2) + e(Y_2, U_1)\big) &= \frac{1}{2}e(U, Y_0) + \frac{1}{2}\sum_{u \in U} \mathbb{E}\big(\,\Delta(u)\big) \\
&\geq \frac{1}{2}e(U, Y_0) + \frac{1}{2}\sum_{u \in U} d_u^{1/2}/2 \\
&\geq \frac{1}{2}e(U, Y_0) + e(U, Y_0)/12n^{1/4},
\end{aligned}$$

since $d_u \leq |Y_0| \leq |Y| \leq 4k + 2 < 9\sqrt{n}$ and so $d_u^{1/2} > d_u/3n^{1/4}$; also

$$\mathbb{E}\big(\,e(Y_1, Y_2)\big) = \frac{1}{2}e(Y_0).$$

Since $|U| < |X|/2$, we can extend the partition $U_1 \cup U_2$ of U to a partition $T_1 \cup T_2$ of $X \cup Y^+$ with $|T_1| \le |T_2| \le |T_1| + 1$. Then, by (19) and (14),

$$\begin{aligned} e(T_1, T_2) &\ge \left\lfloor \frac{|X \cup Y^+|^2}{4} \right\rfloor - e\big(\overline{G}[X \cup Y^+]\big) \\ &\ge \frac{1}{2} e(X \cup Y^+) + \frac{|X \cup Y^+| - 1}{4} - \frac{5}{2}k - 2 \\ &> \frac{1}{2} e(X \cup Y^+) + \frac{n - 14k - 11}{4}. \end{aligned}$$

Thus, partitioning $X \cup Y^+ \cup Y_0$ into $T_1 \cup Y_2$ and $T_2 \cup Y_1$, we see that

$$\begin{aligned} f\big(G[X \cup Y^+ \cup Y_0]\big) &\ge \mathbb{E}\big(e(T_1 \cup Y_2, T_2 \cup Y_1)\big) \\ &= \mathbb{E}\big(e(T_1, T_2) + e(Y_1, Y_2) + e(Y_1, U_2) + e(Y_2, U_1)\big) \\ &> \frac{1}{2} e(X \cup Y^+ \cup Y_0) + \frac{n - 14k - 11}{4} + \frac{e(U, Y_0)}{12 n^{1/4}} \\ &> \frac{1}{2} e(X \cup Y^+ \cup Y_0) + \frac{n + k}{4}, \end{aligned}$$

provided $e(U, Y_0) > (45k + 33) n^{1/4}$, which follows from $e(U, Y_0) > 70 n^{3/4}$, since $n > n_0$. Thus it follows from Lemma 7 and (12) that we may assume

$$e(Y^-, X \cup Y^+) \le 70 n^{3/4}. \qquad (22)$$

Next we prove that $|X \cup Y^+| = n$ or $n+1$. Now since $|Y^-| \le |Y| \le 4k+2$, we have

$$e(Y^-) \le \binom{4k+2}{2} = 8k^2 + 6k + 1.$$

So, since $n > n_0$, it follows that

$$\begin{aligned} e(X \cup Y^+) &\geq e(G) - e(Y^-) - e(Y^-, X \cup Y^+) \\ &\geq \binom{n}{2} + \binom{k}{2} - (8k^2 + 6k + 1) - 70n^{3/4} \\ &\geq \binom{n}{2} - \frac{15}{2}k^2 - \frac{13}{2}k - 1 - 70n^{3/4} \\ &> \binom{n}{2} - 16n. \end{aligned}$$

So in order to have enough vertices for the edges, we must have $|X \cup Y^+| \geq n - 16$, and thus by (12),

$$|Y^-| \leq k + 17. \tag{23}$$

Now if $|X \cup Y^+| \leq n - 1$, then $e(X \cup Y^+) \leq \binom{n-1}{2}$ and so, since $n > n_0$,

$$\begin{aligned} e(Y^-) &\geq \binom{n}{2} + \binom{k}{2} - e(X \cup Y^+) - e(Y^-, X \cup Y^+) \\ &\geq \binom{k}{2} + n - 1 - 70n^{3/4} \\ &> \binom{k}{2} + \frac{n}{2}, \end{aligned}$$

which contradicts (23). So $|X \cup Y^+| \geq n$. Similarly, we have $|X \cup Y^+| \leq n+1$, since, by (19),

$$\begin{aligned} \binom{n+2}{2} - e\big(\overline{G}[X \cup Y^+]\big) &> \binom{n+2}{2} - \frac{5k}{2} - 2 \\ &> \binom{n}{2} + \binom{k}{2}. \end{aligned}$$

We have shown that $|X \cup Y^+| = n$ or $|X \cup Y^+| = n+1$. If $|X \cup Y^+| = n+1$ then, by (20),

$$f(G) \geq \left\lfloor \frac{(n+1)^2}{4} \right\rfloor$$

with equality iff $Y^- = \emptyset$, in which case G consists of K_{n+1} with $\binom{n+1}{2} - m$ edges deleted.

Otherwise $|X \cup Y^+| = n$. Now let H be the weighted graph consisting of all edges of $E(Y^-) \cup E(Y^-, X \cup Y^+)$ with weight $+1$ and all edges of $E\big(\overline{G}[X \cup Y^+]\big)$ with weight -1, so H has total weight $\binom{k}{2}$. It follows from Lemma 4 that H has a cut of weight at least $\lfloor k^2/4 \rfloor$.

Note that since $n > n_0$ it follows from (14), (19) and (22) that $|H| \leq |Y| + e(Y^-, X \cup Y^+) \leq 4k + 2 + 70n^{3/4} + 5k/2 + 2 < |X|/2$. We can therefore extend a partition of H to a partition of G in which $X \cup Y^+$ is evenly partitioned, so

$$f(G) \geq \left\lfloor \frac{n^2}{4} \right\rfloor + f(H) \geq \left\lfloor \frac{n^2}{4} \right\rfloor + \left\lfloor \frac{k^2}{4} \right\rfloor$$

with equality iff $H \cong K_k$, with all edges of weight 1 (or $H \cong K_4$ or $H \cong 2K_3$ when $k = 4$). It follows immediately that $G[X \cup Y^+]$ is complete, and the extremal graphs are as described in the statement of the theorem. ■

What prevents us from extending the argument in the proof of Theorem 1 to graphs with $\binom{n}{2} + \binom{k}{2} + \binom{l}{2}$ edges? The problem is that when we remove the copy of K_n in the argument above, we are left with a graph with weighted edges. If we have $\binom{k}{2}$ edges then Lemma 4 gives us the unique extremal graph, whereas to deal with $\binom{k}{2} + \binom{l}{2}$ edges we would need a version of Theorem 1 for weighted graphs. Our aim in the next section is to prove such a theorem. In particular, it will enable us to determine $f(m)$ exactly for a much wider range of m, and to within an additive constant for every value of m. It will not, however, yield all the extremal graphs.

3. Max cut for multigraphs

Our aim in this section is to determine $f(m)$ to within an additive constant for every integer m, and to determine $f(m)$ exactly for a larger range of values of m. In order to do this, we consider graphs with integer edge-weightings: note that, if all weights are positive, we can think of these as multigraphs, where the weight of an edge indicates its multiplicity. As in the unweighted case, for a graph G with edge-weighting w, we write $f(G)$ for the maximal weight of a bipartite subgraph of G. We define $f_w(m)$ to be the minimum of $f(G)$ over graphs G with $w(G) = \sum_{e\in E(G)} w(e) = m$ and all weights non-negative integers.

Note that the restriction to positive integers means that $f_w(m)$ is the minimum of $f(G)$ over a finite set of multigraphs. Thus there is a (very slow) algorithm to determine $f(m)$. However, we do not lose anything by allowing negative weights: if $w(G) = m$ and G has an edge xy with negative weight, then consider the graph $H = G/xy$ obtained by contracting the edge xy to a single vertex z and defining $w(vz) = w(vx) + w(vy)$ for $v \neq x, y$. Repeating the process until we obtain a graph H with no edges of negative weight, it is clear that $w(H) \geq w(G)$. Since $f_w(m)$ is monotone increasing, it follows that $f(G) \geq f(H) \geq f_w(m)$.

What can we say about $f_w(m)$? Clearly it is subadditive: since $f(G \cup H) = f(G) + f(H)$ for any graphs G and H, it follows that $f_w(m + r) \leq f_w(m) + f_w(r)$ and, similarly, $f(m + r) \leq f(m) + f(r)$. Furthermore, it follows from Lemma 4 that, for $m = \binom{n}{2}$, we have $f_w(m) = f(m) = \lfloor n^2/4 \rfloor$. All the work in this section will go into proving a lower bound for $f_w(m)$ for other values of m.

We begin with Theorem 8, which provides a recursive lower bound on $f_w(m)$ and hence $f(m)$ (as noted in the introduction, this was proved independently by Alon and Halperin [2]). The approach we use in proving the theorem is similar to that used in the proof of Theorem 1. However since we are dealing with weighted graphs the details are rather different.

Theorem 8. *Let G be a graph with integer-valued edge-weighting w. Suppose $w(G) = m$. Then, provided $m > m_0$,*

$$f(G) \geq \min_{n\geq 1} \left\{ \left\lfloor \frac{n^2}{4} \right\rfloor + f_w\left(m - \binom{n}{2}\right) \right\}, \tag{24}$$

where we define $f_w(r) = 0$ *for* $r < 0$.

Proof. Let G be a graph with $w(G) = m$ and $f(G) = f_w(m)$ that does not satisfy (24). We can consider G as a weighted complete graph by defining $w(xy) = 0$ if x and y are nonadjacent. If G contains an edge xy with nonpositive weight then replace G with G/xy. Clearly $w(G/xy) \geq w(G)$ and $f(G/xy) \leq f(G)$. Repeating this process, we may assume that G is a weighted complete graph with all edges of positive weight. Let $m = w(G)$ and define the integer n by

$$\binom{n}{2} \leq m < \binom{n+1}{2}. \tag{25}$$

Since G is complete and every edge has weight at least 1, we have $|G| \leq n$. We will use the fact that, as in (6), for some $c, c_0 > 0$ and every integer m,

$$f(m) \leq \frac{m}{2} + \sqrt{\frac{m}{8}} + cm^{1/4} < \frac{m}{2} + \frac{n}{4} + c_0\sqrt{n}. \tag{26}$$

We will use $c_1, c_2, \ldots$ to refer to constants in the proof below; suitable constants can easily be determined. In several places, we shall assume that n is larger than some fixed constant.

Note that the proof of Lemma 3 carries over straightforwardly to the weighted case (see also Section 5). In particular, since G is complete, it follows from the weighted version of Lemma 3 that

$$f(G) \geq \left(\frac{1}{2} + \frac{1}{2|G|}\right) m$$

and so by (26),

$$\frac{m}{2|G|} \leq \frac{n}{4} + c_0\sqrt{n}.$$

Since $m \geq \binom{n}{2}$, we have

$$|G| \geq \frac{n(n-1)}{n + 4c_0\sqrt{n}} \tag{27}$$

$$= \frac{n-1}{1+4c_0/\sqrt{n}}$$

$$> n - c_1\sqrt{n}.$$

We now find a small set of vertices that meets all edges with weight greater than 1. Let M be a matching of maximal weight in G. Consider the random partition $V(G) = V_1 \cup V_2$, where for each edge $xy \in M$ we independently assign $x \in V_1$ and $y \in V_2$ or $x \in V_2$ and $y \in V_1$ with equal probability; if there is a vertex not covered by M, we assign it to V_1 or V_2 with equal probability. The expected weight of edges between V_1 and V_2 is

$$w(M) + \frac{1}{2}\big(w(G) - w(M)\big) = \frac{1}{2}w(G) + \frac{1}{2}w(M).$$

It follows from (26) that any matching in G has weight at most

$$\frac{n}{2} + 2c_0\sqrt{n}. \tag{28}$$

Now let $e_1, \ldots, e_k$ be a maximal set of independent edges of G with $w(e_i) > 1$ for $i = 1, \ldots, k$. Extend this arbitrarily to a maximal matching M. Then $|M| \geq \big(|G| - 1\big)/2$, so by (27),

$$w(M) \geq \frac{|G|-1}{2} + k > \frac{n - c_1\sqrt{n} - 1}{2} + k$$

and so by (28) we have $k \leq \frac{1}{2}\big(c_1\sqrt{n} + 1\big) + 2c_0\sqrt{n} \leq c_2\sqrt{n}$. Let Y be the set of vertices spanned by $e_1, \ldots, e_k$. Then

$$|Y| = 2k \leq 2c_2\sqrt{n} \tag{29}$$

and any edge of weight greater than 1 is incident with Y. Let $X = V(G)\backslash Y$: then by (27) and (29)

$$|X| \geq |G| - |Y| \geq n - (c_1 + 2c_2)\sqrt{n}. \tag{30}$$

Note that X induces a complete graph with all edges of weight 1.

Now for $y \in Y$, consider the edges between y and X, and order them in increasing order of weight (order edges of the same weight arbitrarily). Let Z_1 be the vertices in X incident with the first $\lfloor |X|/2 \rfloor$ edges and let $Z_2 = X \setminus Z_1$. Consider the partition of $X \cup \{y\}$ into $Z_1 \cup \{y\}$ and Z_2: since this partitions X into sets of size $\lfloor |X|/2 \rfloor$ and $\lceil |X|/2 \rceil$, we see that

$$\begin{aligned} f\big(X \cup \{y\}\big) &\geq \left\lfloor \frac{|X|^2}{4} \right\rfloor + w(y, Z_2) \\ &\geq \frac{1}{2}w(X) + \frac{|X|-1}{4} + \frac{1}{2}w(y,X) + \frac{1}{2}\left(w(y,Z_2) - w(y,Z_1)\right) \\ &= \frac{1}{2}w\big(X \cup \{y\}\big) + \frac{|X|-1}{4} + \frac{w(y,Z_2) - w(y,Z_2)}{2}. \end{aligned}$$

Hence, by Lemma 7 and (30),

$$f(G) \geq \frac{1}{2}w(G) + \frac{n - (c_1 + 2c_2)\sqrt{n} - 1}{4} + \frac{1}{2}\big(w(y,Z_2) - w(y,Z_1)\big)$$

and so, by (26),

$$w(y,Z_2) - w(y,Z_1) \leq (2c_0 + c_2 + c_1/2)\sqrt{n} + 1 < c_3\sqrt{n}. \tag{31}$$

It follows, in particular, that for each $y \in Y$ there is an integer $t(y)$ such that all but at most $c_3\sqrt{n}$ of the edges between y and X have the same weight $t(y)$.

We have defined $t(v)$ for $v \in Y$; set $t(v) = 1$ for $v \in X$. Then $t(v)$ denotes the "typical" weight of edges incident with a vertex v. We could obtain a graph H with the same vertex weights as G from a complete graph on $\sum_{v \in V(G)} t(v)$ vertices by partitioning its vertices into sets $\big\{T_v : v \in V(G)\big\}$, where $|T_v| = t(v)$: contracting each set T_v to a single vertex v^* gives a graph in which all but $O\big(\sqrt{n}\big)$ edges from each vertex v^* have weight $t(v)$ (since all but $O\big(\sqrt{n}\big)$ vertices v^* in H correspond to vertices v in X for which $t(v) = 1$). We shall show that in fact G is not too far from H. Note that the edge in H between vertices v^* and w^* has weight $t(v)t(w)$. With this in mind, for an edge xy in G we define

$$u(xy) = t(x)t(y) - w(xy).$$

Thus u denotes the weight we have to add to each edge of G in order to obtain the graph H.

We know that $u(e) = 0$ for every edge e in $G[X]$. Suppose that for some $y \in Y$ we have

$$\sum_{x \in X} \left| u(xy) \right| \geq c_4 \sqrt{n}$$

where we define $c_4 = 2c_0 + c_2 + c_1/2 + 1$. Then since $w(xy) = t(y)$ for all but at most $c_3\sqrt{n}$ edges between y and X, and $w(y, X) = \sum_{x \in X} \big(t(y) + u(xy)\big)$, we can partition X into $Z_1 \cup Z_2$ as before (except that we order edges between y and X with increasing u-weight). The total weight (with weighting w) of edges between $Z_1 \cup \{y\}$ and Z_2 is then at least

$$\left\lfloor \frac{|X|^2}{4} \right\rfloor + \frac{w(y,X)}{2} + \frac{1}{2} \sum_{x \in X} \left| u(yx) \right| > \frac{w\big(X \cup \{y\}\big)}{2} + \frac{|X|-1}{4} + \frac{c_4 n^{1/4}}{2}, \tag{32}$$

and hence by Lemma 7 and (30)

$$\begin{aligned} f(G) &\geq \frac{1}{2} w(G) + \frac{1}{4}\Big(n - (c_1 + 2c_2)\sqrt{n} - 1\Big) + \frac{1}{2} c_4 \sqrt{n} \\ &> \frac{m}{2} + \frac{n}{4} + c_0 \sqrt{n}, \end{aligned}$$

which contradicts (26).

Thus we may assume that, for every $y \in Y$,

$$\sum_{x \in X} \left| u(xy) \right| < c_4 \sqrt{n}. \tag{33}$$

Suppose that

$$\sum_{x\in X}\sum_{y\in Y}\left|u(xy)\right| > c_5 n^{3/4}$$

where $c_5 = 4c_4^{3/4}$. It follows from (33) that we can pick a subset Y' of Y such that

$$c_5 n^{3/4} < \sum_{x\in X}\sum_{y\in Y'}\left|u(xy)\right| < c_5 n^{3/4} + c_4\sqrt{n}. \tag{34}$$

Then the number of vertices $x \in X$ such that $u(xy) \neq 0$ for some $y \in Y'$ is at most $c_5 n^{3/4} + c_4\sqrt{n} < n/4$, provided n is sufficiently large. We construct a partition of $Y' \cup X$ as follows. Let $Y' = Y_1 \cup Y_2$ be a random partition of Y', where each $y \in Y'$ is in Y_1 or Y_2 independently with probability 1/2. Let X' be the set of vertices $x \in X$ such that $w(xy) \neq 0$ for some $y \in Y'$. Let

$$Z_1 = Y_1 \cup \left\{x \in X' \,:\, u(x, Y_1) < u(x, Y_2)\right\}$$

and

$$Z_2 = Y_2 \cup \left\{x \in X' \,:\, u(x, Y_2) \leq u(x, Y_1)\right\}.$$

Finally, extend the partial partition $Z_1 \cup Z_2$ to a partition of $X \cup Y'$ by adding the remaining members of X arbitrarily so that the final partition $W_1 \cup W_2$ satisfies

$$\sum_{v\in W_1} t(v) \leq \sum_{v\in W_2} t(v) \leq \sum_{v\in W_1} t(v) + 1. \tag{35}$$

(Note that this is possible since $Z_1 \cup Z_2$ contains at most $n/2$ elements of X, provided n is sufficiently large.) Now by Lemma 9 below, for $x \in X'$ we have $\mathbb{E}\left|\, u(x, Y_1) - u(x, Y_2)\right| \geq \mathbb{E}\left(\sum_{i=1}^{U} \pm 1\right) \geq \frac{1}{2}\sqrt{U}$, where $U = \sum_{y\in Y'}\left|u(xy)\right|$.

So by (33), (34) and (35), the expected weight of edges joining W_1 and W_2 is at least

$$\mathbb{E}\Big(\big\lfloor t(X \cup Y')^2/4 \big\rfloor + u(Y')/2 + \sum_{x \in X} \max\big(u(x,Y_1), u(x,Y_2)\big) \Big)$$

$$\geq \frac{1}{2} w(X \cup Y') + \frac{|X|-1}{4} + \mathbb{E} \sum_{x \in X} \frac{\big|u(x,Y_1) - u(x,Y_2)\big|}{2}$$

$$\geq \frac{1}{2} w(X \cup Y') + \frac{|X|-1}{4} + \frac{1}{4} \sum_{x \in X} \Big(\sum_{y \in Y'} \big|u(xy)\big| \Big)^{1/2}$$

$$\geq \frac{1}{2} w(X \cup Y') + \frac{|X|-1}{4} + \frac{c_5 n^{3/4}}{4\sqrt{c_4} n^{1/4}}$$

$$\geq \frac{1}{2} w(X \cup Y') + \frac{|X|-1}{4} + c_4 \sqrt{n}.$$

As in (32), this yields a contradiction, so we may assume that

$$\sum_{x \in X} \sum_{y \in Y} \big|u(xy)\big| < c_5 n^{3/4}. \tag{36}$$

It follows that there are at most $c_5 n^{3/4}$ vertices of X which are incident to an edge e with $u(e) \neq 0$.

Let G' be the graph with edges $\{e \in E(G) : u(e) \neq 0\}$ with edge-weighting u and vertices $\{v \in V(G) : u(vw) \neq 0 \text{ for some } w \in V(G)\}$. Then, by (29) and (36),

$$|G'| \leq c_5 n^{3/4} + |Y| \leq c_6 n^{3/4}$$

which by (27) is smaller than $n/4$ for sufficiently large n.

Finally, let $W_1 \cup W_2$ be a partition of G' such that the total weight of edges between W_1 and W_2 is at least $f(G')$. Since $|G'| < n/4$ provided n

is sufficiently large, it follows from (30) that we can extend $W_1 \cup W_2$ to a partition $V_1 \cup V_2$ of $V(G)$ such that

$$\big|t(V_1) - t(V_2)\big| \le 1.$$

Let $t = \sum_{v \in V(G)} t(v)$. Then

$$\begin{aligned} w(G) &= \sum_{vw \in E(G)} t(v)t(w) + \sum_{vw \in E(G)} u(vw) \\ &\le \binom{t}{2} + u(G'), \end{aligned}$$

while the weight of edges between V_1 and V_2 is at least

$$t(V_1)t(V_2) + u(V_1, V_2) \ge \left\lfloor \frac{t^2}{4} \right\rfloor + f(G').$$

Therefore

$$\begin{aligned} f(G) &\ge \left\lfloor \frac{t^2}{4} \right\rfloor + f(G') \\ &\ge \left\lfloor \frac{t^2}{4} \right\rfloor + f_w\left(w(G) - \binom{t}{2}\right). \qquad \blacksquare \end{aligned}$$

We have used an estimate in the proof above that is an immediate consequence of the following trivial lemma. We are interested in random sums $\sum \varepsilon_i a_i$, where the a_i are independent Bernoulli random variables taking values $+1$ and -1 with probability $1/2$. We shall write $\pm$ instead of ε_i.

Lemma 9. *Let $s_1 + \cdots + s_k$ be a partition of n and $t_1 + \cdots + t_l$ a refinement of $s_1 + \cdots + s_k$. Then*

$$\mathbb{E}\left|\sum_{i=1}^{k} \pm s_i\right| \ge \mathbb{E}\left|\sum_{i=1}^{l} \pm t_i\right|.$$

Proof. It is enough to consider the simple refinement when $l = k + 1$, $s_i = t_i$ for $i < k$ and $s_k = t_k + t_{k+1}$. We may couple the sums $\sum \pm s_i$ and $\sum \pm t_i$ so that $\pm s_i$ and $\pm t_i$ have the same sign for $i < k$, while $\pm s_k$, $\pm t_k$ and $\pm t_{k+1}$ are independent. Let $S = \sum_{i=1}^{k-1} \pm s_i = \sum_{i=1}^{k-1} \pm t_i$. We must show

$$\mathbb{E}|S \pm s_k| \geq \mathbb{E}|S \pm t_k \pm t_{k+1}|.$$

Now for real numbers $\alpha \geq \beta \geq 0$ and $L \geq 0$,

$$\begin{aligned} |L+\alpha| + |L-\alpha| &\geq L + \alpha + |L-\alpha| \\ &\geq L + \alpha + |L-\beta| + \beta - \alpha \\ &= |L+\beta| + |L-\beta|, \end{aligned}$$

so in general for $|\alpha| \geq |\beta|$ and any L,

$$|L+\alpha| + |L-\alpha| \geq |L+\beta| + |L-\beta|.$$

Conditioning on the value of S, we see that since $|t_k - t_{k+1}| \leq |t_k + t_{k+1}|$, we have

$$\big|S + (t_k - t_{k+1})\big| + \big|S - (t_k - t_{k+1})\big| \leq |S + t_k + t_{k+1}| + |S - t_k - t_{k+1}|$$

and so

$$\mathbb{E}|S \pm t_k \pm t_{k+1}| \leq \frac{1}{2}|S + t_k + t_{k+1}| + \frac{1}{2}|S - t_k - t_{k+1}| = \mathbb{E}|S \pm s_k|.$$

The result follows immediately. ■

For what value of n is the quantity in Theorem 8 minimized? Suppose $\binom{n_0}{2} \leq m < \binom{n_0+1}{2}$, say $m = \binom{n_0}{2} + r$. Clearly we must have $n \leq n_0 + 1$, and it follows from (30) that we may assume $n > n_0 - c\sqrt{n_0}$. We claim that (24) is minimized with $n = n_0$ or $n = n_0 + 1$. Now since the argument of Lemma 3 applies to multigraphs as well as graphs, we can deduce the Edwards formula for multigraphs:

$$f_w(m) \geq \frac{m}{2} + \sqrt{\frac{m}{8}} + O(1).$$

Since $f_w(m) \leq f(m)$, it follows from (6) that

$$f_w(m) = \frac{m}{2} + \sqrt{\frac{m}{8}} + O(m^{1/4}). \tag{37}$$

If $n = n_0 - t$, with $0 \leq t \leq c\sqrt{n}$, then

$$m - \binom{n}{2} = t(n_0 - t) + \binom{t}{2} + r$$

so by (37),

$$\left\lfloor \frac{n^2}{4} \right\rfloor + f_w\left(m - \binom{n}{2}\right)$$

$$= \frac{m}{2} + \frac{n}{4} + \left(\frac{t(n_0 - t) + \binom{t}{2} + r}{8}\right)^{1/2} + O(n_0 t + r + 1)^{1/4}.$$

Provided n is sufficiently large, and $0 \leq t < c\sqrt{n_0}$, this is minimal when $t = 0$. We conclude the following.

Theorem 10. *For every sufficiently large positive integer* m,

$$f_w(m) = \min\left\{\left\lfloor \frac{(n+1)^2}{4} \right\rfloor, \left\lfloor \frac{n^2}{4} \right\rfloor + f_w\left(m - \binom{n}{2}\right)\right\}, \tag{38}$$

where n *is defined by*

$$\binom{n}{2} \leq m < \binom{n+1}{2}.$$

As remarked in the introduction, probably $f_w(m) = f(m)$ for every m. Even if this is not true, it seems likely that (38) holds with $f(m)$ in place of $f_w(m)$ when m is sufficiently large.

4. Extremal graphs for Max Cut

We can apply Theorem 10 to obtain extremal graphs and multigraphs in more cases than Theorem 1 and Lemma 4. Let us note first that any integer m can be written in the form

$$m = \binom{n_1}{2} + \binom{n_2}{2} + \cdots + \binom{n_k}{2}$$

for some $k > 0$, where $n_1 > \cdots > n_k \geq 2$ and each n_i in turn is chosen to be as large as possible. For $1 \leq i < k$, let

$$M_i = \left\lfloor \frac{n_1^2}{4} \right\rfloor + \cdots + \left\lfloor \frac{n_{i-1}^2}{4} \right\rfloor + \left\lfloor \frac{(n_i+1)^2}{4} \right\rfloor$$

and define

$$M = \left\lfloor \frac{n_1^2}{4} \right\rfloor + \cdots + \left\lfloor \frac{n_k^2}{4} \right\rfloor$$

then it follows by repeated application of Theorem 10 that, provided n_{k-1} is sufficiently large,

$$f_w(m) = \min\{M_1, \ldots, M_{k-1}, M\}.$$

For $1 \leq i < k$, we can obtain a graph G with m edges and $f(G) = M_i$ by deleting $\binom{n_i+1}{2} - \binom{n_i}{2} - \cdots - \binom{n_k}{2}$ edges from the graph

$$K_{n_1} \cup \cdots \cup K_{m_{i-1}} \cup K_{n_i+1}, \qquad (39)$$

while the graph

$$K_{n_1} \cup \cdots \cup K_{n_k} \tag{40}$$

has m edges and no bipartite subgraph with more than M edges. Note that in both (39) and (40), we could instead take any edge-disjoint union of the complete graphs. Thus there may be many possible extremal graphs.

Recall that in the case $k = 2$, Theorem 1 asserts that for sufficiently large m, the extremal graphs are precisely the graphs (39) and (40) and their variants obtained by taking different edge-disjoint unions (note that the case $n_k = 4$ is special, since we can take two copies of K_3 instead of K_4).

Keeping the notation of the last few paragraphs, we can extend Theorem 10 for graphs as follows.

Theorem 11. *Let m be a positive integer and define k, $n_1, \ldots, n_k$, and $M_1, \ldots, M_{k-1}, M$ as above. Suppose that*

$$M < \min\{M_1, \ldots, M_{k-1}\}. \tag{41}$$

Then, provided n_{k-1} is sufficiently large,

$$f(m) = M$$

and the extremal graphs are obtained by taking an edge-disjoint union of $K_{n_1}, \ldots, K_{n_k}$, unless $n_k = 4$, in which case there is an additional set of extremal graphs obtained by taking an edge-disjoint union of $K_{n_1}, \ldots, K_{n_{k-1}}, K_3, K_3$.

Proof. We argue by induction on k. For $k = 1$, the result follows immediately from Lemma 4. For $k \geq 2$, we know from Theorem 10 and example (40) that $f(m) = f_w(m) = M$. Let G be a graph with m edges and $f(G) = f(m)$. As in the proof of Theorem 1, we can decompose G as the edge-sum of K_n and H, where H is a weighted graph in which all edges are weighted ± 1. Furthermore, any partition of H can be extended to an optimal partition of K_n, so $f(G) = f(K_n) + f(H)$ and we must therefore have $n = n_1$ and so $w(H) = \binom{n_2}{2} + \cdots + \binom{n_k}{2}$. If H has an edge xy with negative weight, then contracting xy gives a weighted graph H' with

$$w(H') > \binom{n_2}{2} + \cdots + \binom{n_k}{2}.$$

It follows from Theorem 10 and the inductive hypothesis that

$$f(H') \geq \left\lfloor \frac{n_1^2}{4} \right\rfloor + \cdots + \left\lfloor \frac{n_k^2}{4} \right\rfloor + 1$$

and so, since $f(H) \geq f(H')$, we obtain $f(G) \geq f(K_{n_1}) + f(H) > M$, which is a contradiction.

Thus all edges of H must have weight $+1$, so we can consider H as an unweighted graph, with $\binom{n_2}{2} + \cdots + \binom{n_k}{2}$ edges. For $2 \leq i < k$, let

$$M_i' = \left\lfloor \frac{n_2^2}{4} \right\rfloor + \cdots + \left\lfloor \frac{n_{i-1}^2}{4} \right\rfloor + \left\lfloor \frac{(n_i+1)^2}{4} \right\rfloor$$

and let

$$M' = \left\lfloor \frac{n_2^2}{4} \right\rfloor + \cdots + \left\lfloor \frac{n_k^2}{4} \right\rfloor$$

it follows from (41) that

$$M' < \min\{M_2', \ldots, M_{k-1}'\}.$$

Thus we may apply the inductive hypothesis to H: the result follows immediately. ■

A similar argument gives the following result for weighted graphs.

Theorem 12. *Under the conditions of Theorem* 11,

$$f_w(m) = M$$

and the extremal weighted graphs are the edge sums of $K_{n_1}, \ldots, K_{n_k}$*, unless* $n_k = 4$*, in which case the edge sums of* $K_{n_1}, \ldots, K_{n_{k-1}}, K_3, K_3$ *are also extremal.*

Proof. We argue as in the proof of Theorem 11, except that we use the decomposition of Theorem 21 below. Note that in the decomposition $K_t^* \oplus H$ if $f(K_t^*) = \lfloor t^2/4 \rfloor$ then K_t^* must not have been contracted ■

What happens when $M \geq \min\{M_1, \ldots, M_k\}$? We conjecture that the natural extension of Theorem 1 should hold: the extremal grahs are obtained by deleting edges in (39). The weighted case seems more complicated.

Part II: Algorithms for Max Cut

5. An extremal algorithm for Max Cut

In this second part of the paper, we turn from extremal questions to the problem of finding polynomial time algorithms that give large bipartite subgraphs of a graph or edge-weighted graph. In this section we describe a linear time algorithm that, given a graph with total edge weight m, gives a bipartite subgraph of weight at least $f(m)$. In subsequent sections, we give a linear time algorithm that, for graphs G of weight m, finds a cut of weight at least $m/2 + \sqrt{m/8} + k$ if such a cut exists a nd otherwise finds an optimal cut, and an algorithm that approximates the order of magnitude of $f(G) - m/2 - \sqrt{m/8}$.

We remark that it often appears to be easier to find efficient algorithms for partitioning unweighted graphs than it is for partitioning weighted graphs. We shall assume below that we are dealing with graphs that have integer edge-weightings, where we al low both positive and negative weights. We may also assume that our graphs are connected: given a graph G with n vertices and e edges, we can identify a vertex from each component in time $O(e+n)$ to obtain a graph H with $f(G) = f(H)$; any biparti tion of H yields an equivalent bipartition of G in time $O(n)$. (We should also note that we have assumed that all arithmetical operations can be performed in unit time, regardless of the magnitude of edge-weights.)

The main result of this section is the following linear time algorithm.

Theorem 13. *There is an algorithm that, given a graph G with e edges, integer-valued edge-weighting w and total weight m finds a cut of weight at least $f_w(m)$ in time $O\big(e+|G|\big)$.*

By taking $w \equiv 1$, we obtain the following immediate corollary for unweighted graphs.

Corollary 14. *There is an algorithm that, given a multigraph G with m edges and n vertices, finds in time $O(m+n)$ a bipartite subgraph of G with at least $f_w(m)$ edges.*

Many of the results from Sections 2 and 3 have efficient corresponding algorithms. Let us note first that Theorem 5 has the following immediate corollary.

Lemma 15. *There is an algorithm that, given a connected graph G with n vertices and e edges, and an edge-weighting w with positive integers and total weight m, finds in time $O(e+n)$ a bipartite subgraph of G with weight at least*

$$\frac{w(G)}{2}+\frac{|G|-1}{4}.$$

We shall find it useful to have the following lemma.

Lemma 16. *There is an algorithm that, given a graph G with e edges and edge-weighting w, finds in time $O(e+n)$ a contraction of G to a complete graph in which all edges have positive weight.*

Proof. Begin by deleting all edges with weight 0. We then take a greedy colouring of G (which takes time $O(n+e)$) and contract each colour class to a single vertex to obtain a weighted complete graph H with $|H| = O\big(\sqrt{e}\big)$. We now repeated ly contract edges of nonpositive weight until we obtain a graph with all edge weights positive. Since each contraction takes time $O\big(\sqrt{e}\big)$ and there are at most $O\big(\sqrt{e}\big)$ contractions, the algorithm terminates in time $O(e)$ (note that we can deal with all edges with nonpositive weight in time $O(e)$ by processing one vertex at a time). ∎

We shall also need algorithmic versions of Lemmas 3, 4 and 7, and a result concerning weighted matchings. Most of the following lemma can be found in Hofmeister and Lefmann [19].

Lemma 17. *We consider graphs G with n vertices, e edges and edge-weighting w.*

(i) *There is an algorithm running in time $O(e+n)$ that, given a matching M in G, finds a cut with weight at least*

$$\frac{1}{2}w(G) + \frac{1}{2}w(M).$$

(ii) *There is an algorithm running in time $O(e+n)$ that, given a proper k-colouring of G, finds a cut with weight at least*

$$\left(\frac{1}{2} + \frac{1}{2k}\right) w(G).$$

In particular, there is an $O(e+n)$ algorithm that finds a cut with weight at least

$$\left(\frac{1}{2} + \frac{1}{2|G|}\right) w(G).$$

(iii) *Given a weighted graph G and a partial partition $V_1 \cup V_2$ of $V(G)$, we can find in time $O(e+n)$ a cut of weight at least*

$$w(V_1, V_2) + \frac{1}{2}\big(w(G) - w(V_1 \cup V_2)\big).$$

(iv) *There is an algorithm running in time $O(e+n)$ that finds a matching of weight at least $w(G)/n$.*

Proof. Parts (i), (ii) and (iv) are obtained by applying algorithms from [19]. Note that if $|G|$ is odd, we may add in an isolated vertex.

Part (iii) follows by using the greedy algorithm: add each vertex of $V(G) \setminus (V_1 \cup V_2)$ in turn to whichever side of the partition gives the heavier cut. ■

We now prove the main result of this section.

Proof of Theorem 13. Let m_0 be large enough for (38) to apply: that is, for $m \geq m_0$, and $\binom{n}{2} \leq m < \binom{n+1}{2}$, $f_w(m) = \min\left\{ \lfloor (n+1)^2/4 \rfloor, \lfloor n^2/4 \rfloor + f_w\left(m - \binom{m}{2}\right) \right\}$. By (1) (for weighted graphs) and (6), there is m_1 such that, provided $m \geq m_1$,

$$\frac{m}{2} + \sqrt{\frac{m}{8}} - 1 \leq f_w(m) \leq \frac{m}{2} + \sqrt{\frac{m}{8}} + 2m^{1/4}. \tag{42}$$

The main part of our algorithm will apply to edge-weighted graphs with weight at least $M = \max\{m_0, m_1, K\}$, where K is a large fixed constant; we deal separately with graphs of smaller weight.

We begin by contracting G to a complete weighted graph with the algorithm from Lemma 16. We may thus assume that G is a complete weighted graph with e edges, all of positive weight, and that $w(G) = m$. We shall show that we can find a cut of weight at least $f_w(m)$ in time $O(e)$.

If $w(H) \leq M$, then since there are only finitely many graphs with positive edge weights and total weight at most M (that is, multigraphs with at most M edges), we can examine all partitions of H in fixed time, or else store all optimal partitions as a look-up table. Note that this may introduce a large constant into the time or space complexity of the algorithm: we return to this point after the proof.

We may therefore assume that G has weight $m \geq M$. Our algorithm follows parallel to the proof of Theorem 8. Note that if at any time we find a cut with weight at least $m/2 + \sqrt{m/8} + 2m^{1/4}$ then we can halt the algorithm immediately.

Define the integer n by

$$\binom{n}{2} \leq m < \binom{n+1}{2}. \tag{43}$$

Then, by (42),

$$f_w(m) \leq \frac{m}{2} + \frac{n}{4} + 3\sqrt{n}. \tag{44}$$

Since G is complete we have

$$|G| \leq n.$$

On the other hand, by Lemma 17(ii) we can find in time $O\big(e + |G|\big)$ a bipartite subgraph of G with weight at least $\left(\frac{1}{2} + \frac{1}{2|G|}\right)m$. Thus we can halt the algorithm if

$$\left(\frac{1}{2} + \frac{1}{2|G|}\right) m > \frac{m}{2} + \sqrt{\frac{m}{8}} + 2m^{1/4}, \tag{45}$$

which is true for sufficiently large m unless

$$|G| > n - 8\sqrt{n} \tag{46}$$

Note that this implies m is $O(e)$.

Now we find a small set of edges that meets all edges in G of weight more than 1. We can find a maximal matching M in G in time $O(e)$, by choosing greedily edges of weight more than 1 and then filling out with edges of weight 1. By Lemma 17(i) we can find in time $O(e)$ a bipartite subgraph of G with weight at least $\big(m + w(M)\big)/2$. Thus we are done if $w(M) > \sqrt{m/2} + 4m^{1/4}$. Otherwise, provided m is sufficiently large, using (43) we see that M contains at most

$$\sqrt{\frac{m}{2}} + 4m^{1/4} - \left\lfloor \frac{|G|}{2} \right\rfloor < 7\sqrt{n} \tag{47}$$

edges of weight greater than 1. We obtain either a bipartite subgraph with weight at least $f_w(m)$, in which case we halt the algorithm, or else a set of at most $7\sqrt{n}$ edges, and hence a set Y of at most $14\sqrt{n}$ vertices of G, meeting all edges with weight more than 1.

Let $X = V(G) \setminus Y$, so by (46) and (47),

$$|X| = |G| - |Y| \geq n - 22\sqrt{n}.$$

Note that $G[X]$ is a complete graph in which all edges have weight 1. For $y \in Y$, let us consider the edges between y and X. As in the proof of

Theorem 8, we order the edges into increasing order of weight, which takes time $O(n \log n)$, which is $O(e)$, and partition $X \cup \{y\}$ into $Z_1 \cup \{y\}$ and Z_2, where $\{yz : z \in Z_1\}$ are the lightest $\lfloor |X|/2 \rfloor$ edges and $\{yz : z \in Z_2\}$ are the heaviest $\lceil |X|/2 \rceil$ edges. Extending to a partition of $V(G)$ with the algorithm of Lemma 17(iii), we obtain a bipartite subgraph of weight at least $w(G)/2 + \big(|X| - 1\big)/4 + \big(w(y, Z_2) - w(y, Z_1)\big)/2$, which is at least

$$\frac{w(G)}{2} + \frac{n - 15\sqrt{n} - 1}{4} + \frac{w(y, Z_2) - w(y, Z_1)}{2}. \tag{48}$$

Either this is a bipartite subgraph of weight at least $f(m)$ or, by (44),

$$w(y, Z_2) - w(y, Z_1) < 15\sqrt{n}. \tag{49}$$

It follows that all but $15\sqrt{n}$ of the edges between y and X have the same weight, say $t(y)$. Note that it is easy to check that (49) is satisfied and to determine $t(y)$ for each $y \in Y$ in total time $O\big(|Y| n \log n\big)$ which is $O(e)$.

Setting $t(v) = 1$ for $v \in X$, we may assume that $t(v)$ has been defined for all $v \in V(G)$. For $x, y \in V(G)$, define $u(xy) = w(xy) - t(x)t(y)$. It follows from (49) that, for $y \in Y$,

$$\sum_{x \in X} \big| u(xy) \big| < 15\sqrt{n}.$$

Now it is straightforward to calculate $U := \sum_{x \in X, y \in Y} \big| u(x, y) \big|$ in time $O(e)$. If $U > 30n^{3/4}$ then we can find a set $Y' \subset Y$ with

$$30n^{3/4} \le \sum_{x \in X, y \in Y} \big| u(xy) \big| < 30n^{3/4} + 15\sqrt{n}$$

by choosing vertices one at a time from Y until both inequalities are satisfied. Let $Z = \big\{ x \in X : u(xy) \neq 0 \text{ for some } y \in Y' \big\}$. Then $|Z| \le 30n^{3/4} + 15\sqrt{n}$. In the proof of Theorem 10, we used a random bipartition $Y' = Y_1 \cup Y_2$, and defined $Z_1 = \big\{ x \in Z : u(x, Y_1) \ge u(x, Y_2) \big\}$ and $Z_2 = Z \setminus Z_1$. Extending such a partition here to a partition $W_1 \cup W_2$ of $X \cup Y'$ satisfying (35), we get

$$\text{(50)}\quad \mathbb{E}\big(\,w(W_1,W_2)\big) \geq \frac{1}{2}w(X\cup Y') + \frac{|X|-1}{4} + \sum_{x\in X, y\in Y'} \big|\,u(x,y)\big|\,/3n^{1/4}$$

$$> \frac{1}{2}w(X\cup Y') + \frac{|X|}{4} + 10\sqrt{n}.$$

It follows that as in the proof of Theorem 8 after (34) we can extend the partition as in (35) and obtain a cut of G with weight at least

$$\text{(51)}\qquad \frac{1}{2}w(G) + 10\sqrt{n} + \frac{1}{4}|X| > \frac{1}{2}m + \frac{n}{4} + 3\sqrt{n}$$

which suffices by (44). Thus it is enough to find a bipartition of $X \cup Y'$ that does at least as well as (50). It follows from Lemma 18 below that we can do this in time $O\big(|Y'|^2|U| + e\big)$ which by assumption is at most $c\big(\sqrt{n}\,\big)^2 n^{3/4} + e$, which is $O(e)$. (Note that we apply Lemma 18 with vertex sets $V_1 = Y'$, $V_2 = X$ and edge set $E(Y') \cup E(Y,X)$.)

Otherwise, $U \leq 30n^{3/4}$, so $\big\{x \in X \,:\, u(xy) = 0 \text{ for all } y \in Y\big\}$ contains at least $n/2$ vertices, provided U is sufficiently large. Thus we may consider G as the edge sum of a graph H_1 with edge weights given by u and a graph H_2 with edge weights given by t. Since $\big|X \setminus V(H_1)\big| > n/2$, any partition of H_1 can be extended to a partition of $V(G)$ in which the two halves have t-weight differing by at most 1. This corresponds to a bipartite subgraph of H_2 with weight at least $\big\lfloor t(G)^2/4\big\rfloor$.

It is therefore enough to find a bipartite subgraph of H_1 with u-weight at least $f\big(u(H_1)\big)$, which we do recursively with the algorithm above. ∎

In the proof above, we have used a lemma which we now give.

Lemma 18. *There is a polynomial time algorithm that, given a graph G with e edges, integer-valued edge-weighting w and total weight m, and* *a bipartition $V(G) = V_1 \cup V_2$, where V_2 is independent, finds a bipartite subgraph with weight at least*

$$\frac{1}{2}w(G) + \sum_{v \in V_2} d_v^{1/2}/2 \tag{52}$$

where $d_v = \sum_{u \in V_1} \big|w(vu)\big|$. If all edges have nonzero weight and G contains no isolated vertices then the algorithm runs in time $O\big(|V_1|^2 D + e\big)$, where $D = \sum_{v \in V_2} d_v$.

Proof. We obtain a bipartition of $V(G)$ by first partitioning V_1 as $W_1 \cup W_2$ and then adding each vertex v of V_2 to W_1 if $w(v, W_1) \leq w(v, W_2)$ and to W_2 otherwise. If we take a random partition of V_1, then as in the proof of Theorem 8, we see that the expected weight of edges in the bipartite subgraph given by the resulting partition of $V(G)$ is at least (52).

We derandomize this as follows. First delete all edges with weight 0: this takes time $O(e)$. Now partition V_1 one vertex at a time. Suppose we have a partition $V_1 = W_1' \cup W_2' \cup W_3$, where W_3 is the set of vertices we have not yet assigned, and we wish to add the vertex $v \in W_3$ to W_1' or W_2'. Consider a random bipartition $W_1 \cup W_2$ of V_1, chosen uniformly among all partitions that extend the partial partition $W_1' \cup W_2'$: define a random variable $X = w(W_1, W_2)$ and, for each $x \in V_2$, define $Y_1(x) = w(x, W_1)$ and $Y_2(x) = w(x, W_2)$. Then the bipartite subgraph obtained by greedily adding vertices in V_2 to W_1 or W_2 has weight

$$Z = X + \sum_{x \in V_2} \max\big\{Y_1(x), Y_2(x)\big\}.$$

We compare $\mathbb{E}(Z \mid v \in W_1)$ and $\mathbb{E}(Z \mid v \in W_2)$ explicitly. First of all, $\mathbb{E}(X \mid v \in W_1) - \mathbb{E}(X \mid v \in W_2) = \frac{1}{2}\big(w(v, W_2') - w(v, W_1')\big)$, which is easily calculated in time $O\big(|V_1|\big)$. For $x \in V_2$, the distribution of $w(x, W_1)$ is the result of a random walk with initial value $w(x, W_1')$ and (independent, equiprobable) increments $\big\{0, w(xy)\big\}$ for each edge xy with $y \in V_1 \setminus (W_1' \cup W_2')$. Since the walk has integer values in $[-d_v, d_v]$ and there are at most $|V_1|$ steps, we can update the distribution after each step in time $O(d_v)$ and thus obtain the distribution of $w(x, W_1)$ in time $O(|V_1| d_v)$. Since $w(x, W_1) + w(x, W_2) = w(x, V_1)$, we obtain the joint distribution of $Y_1(x)$ and $Y_2(x)$ and hence the distribution of $\max\big\{Y_1(x), Y_2(x)\big\}$ in time $O\big(|V_1| d_v\big)$. Similar comments apply to the distributions conditioned on $v \in W_1$ or $v \in W_2$, so we can determine the conditional distribution of $\sum_{x \in V_2} \max\{Y_1(x), Y_2(x)\}$ in time $O(|V_1|D)$. We can therefore decide to assign each vertex to W_1 or W_2 so as to maximize the expectation of Z in time $O(|V_1|D)$ and hence obtain a partition satisfying (52) in time $O(|V_1|^2 D)$. ■

In Theorem 13 we may have an extremely large look-up table. This can be avoided by slightly weakening the results we expect from our algorithm. For graphs with large total weight we run the algorithm as before. However, when we reach the po int where we want to bipartition a small subgraph we use an approximation algorithm instead. Thus we lose at most some fixed additive constant on the bound given in the theorem.

6. An algorithm for graphs without large cuts

A parametrized problem with parameter k is said to be *fixed parameter tractable* (see [8]) if there is an algorithm running in time $O(f(k)N^t)$, where N is the input size, t is a constant and f is any (not necessarily polynomial) fun ction. Mahajan and Raman [29] show that the problem of finding a cut of size at least $\lceil m/2 \rceil + k$ if one exists in a graph with m edges and n vertices is fixed parameter tractable and give algorithms running in time $O(m2^{4k} + n^3)$ and $O(2^{ck^2} + m + n)$, and deduce that if $k \in O(\log mn)$ then there is a polynomial time algorithm. We remark that the first algorithm relies on finding a cut of weight at least $m/2 + (n-1)/4$; since Ngoc and Tuza [30] (see also Lemma 2 above) have found an $O(m+n)$ algorithm, the first algorithm of Mahajan and Raman can be improved to $O(m2^{4k} + n)$.

Note that by the bound (1) of Edwards, every graph of weight m satisfies $f(G) \geq m/2 + \sqrt{m/8} + O(1)$, while we know that $f_w(m) \leq m/2 + \sqrt{m/8} + O(m^{1/4})$. The algorithms of Mahajan and Raman for finding cuts of size at least $\lceil m/2 \rceil + k$ exploit the fact that $f(m) - \lceil m/2 \rceil$ grows as fast as $c\sqrt{m}$: for instance, if $k < (n-1)/4 \leq \sqrt{m/8} + O(1)$ we can use the Edwards result (1), whereas if $k \geq (n-1)/4$ we can examine all partitions explicitly. Our aim i n this section is to show that the problem of finding a cut of size at least $m/2 + \sqrt{m/8} + k$, if such a cut exists, is fixed-parameter tractable, and give an algorithm runing in time $O(2^{ck^4} + e + n)$.

We begin with an extension of Lemma 15 for graphs with arbitrary non-zero integer edge-weightings.

Lemma 19. *There is an algorithm running in time $O\big(e+|H|\big)$ that given a connected graph H with e edges and an edge-weighting w with non-zero integers, finds a cut of weight at least*

$$\frac{w(H)}{2}+\frac{|H|-1}{4}.$$

Proof. As in the proof of Theorem 5 we can find a collection $\mathcal{S}$ of vertex-disjoint induced stars containing at least $\big(|H|-1\big)/2$ edges. We partition the edges of $\mathcal{S}$ into $\mathcal{S}^+$ and $\mathcal{S}^-$, the edges of positive and negative weight respectively. Contracting every edge in $\mathcal{S}^-$ gives a graph of weight $w(H)-w(\mathcal{S}^-)$, in which the edges of $\mathcal{S}^+$ form a collection of induced stars. We take each star S in turn: if $|S|>2$ then we identify its endvertices to obtain an edge of weight $w(S)$. We end up with a matching of weight $w(\mathcal{S}^+)$, and the algorithm of Lemma 17(i) gives a cut of weight $\frac{1}{2}\big(w(H)-w(\mathcal{S}^-)\big)+\frac{1}{2}w(\mathcal{S}^+)\geq w(H)/2+\big(|H|-1\big)/4$. ■

We will also need an extension of the Edwards bound (1) to weighted graphs.

Theorem 20. *There is an algorithm running in time $O\big(e+|H|\big)$ that given a graph H with e edges and integer-valued edge-weighting w, finds a cut of weight at least*

$$\frac{w(H)}{2}+\sqrt{\frac{U}{8}+\frac{1}{64}}-\frac{1}{8}, \tag{53}$$

where $U=\sum_{e\in E(H)}\big|w(e)\big|$.

Proof. Let $h=\sqrt{U/8+1/64}-1/8$. We begin by deleting edges of weight 0. We may assume H is connected or else identify a vertex from each component. If $|H|-1>4h$ then Lemma 19 provides a cut of weight at least $w(H)/2+\big(|H|-1\big)/4\geq w(H)/2+h$ as required. Otherwise, $|H|-1\leq 4h$. Let u be the weighting on $E(H)$ defined by $u(e)=\big|w(e)\big|$, so $u(H)=U$. Then the algorithm of Lemma 17(iv) gives a matching M

with $u(M) \geq U/|H|$. Let M^+ be the edges of M with positive weight and M^- the edges with negative weight. As in the proof of Lemma 19, we contract the edges of M^- and apply the algorithm of Lemma 17(i) to M^+ (with edge-weighting w) to obtain a cut of weight at least $\frac{1}{2}\big(w(H) - w(M^-)\big) + \frac{1}{2}w(M^+) \geq w(H)/2 + U/2|H| \geq w(H)/2 + U/(8h+2)$, and a simple calculation shows $U/(8h+2) = h$. ■

Our main tool will be the following decomposition result.

Theorem 21. *Let $c > 0$ be a constant. There is an algorithm running in time $O\big(e + |G|\big)$ that takes as input a graph G with e edges, integer-valued edge-weighting w and total weight m and gives as output either a cut of weight at least $m/2 + \sqrt{m/8} + cm^{1/4}$ or a decomposition of G as an edge sum*

$$G \equiv K_t^* \oplus H$$

where K_t^* is obtained from a complete graph K_t of order $t = n + O(1)$ with all edges of weight 1 by contracting at most $O(m^{1/4})$ edges and H is a graph such that $\big|V(H) \cap V(K_t^*)\big| = O(n^{3/4})$. Here the integer n is defined by $\binom{n}{2} \leq m < \binom{n+1}{2}$.

Proof. The algorithm in the proof of Theorem 13 began by contracting G to a complete graph with positive edge-weights. This worked since the total weight was not reduced, and we were looking for a cut of weight at least $f_w(m)$. Here, however, we must be more careful: we may be looking for a cut of weight more than $f_w(m)$, and such a cut could be destroyed by any compression of G. For instance, the path P_4 with four vertices and all edges with weight 1 has $f(P_4) = 3$, while its contraction to C_3, all edges with weight 1, has $f(C_3) = 2$.

We may clearly delete all edges of weight 0. Identifying one vertex from each component, we may assume G is connected. Define, as usual, the integer n by $\binom{n}{2} \leq m < \binom{n+1}{2}$. Let M be large enough so that, if $m \geq M$, (38) holds, and

$$(54)\qquad \frac{m}{2} + \sqrt{\frac{m}{8}} - 1 \leq f_w(m) \leq \frac{m}{2} + \sqrt{\frac{m}{8}} + 2m^{1/4} \leq \frac{m}{2} + \frac{n}{4} + 3\sqrt{n}.$$

Suppose first that $m \le M$. Let $U = \sum_{e\in E(G)} \left|w(e)\right|$. If $U > 4(8c^2+1)M$ then Theorem 21 provides the required partition. Thus we may assume that $U \le 4(8c^2+1)M$, and since G has at most U edges we can examine all partitions explicitly in constant time.

Otherwise, we have $w(G) = m > M$. Note first that we may assume

$$|G| > n - 12c\sqrt{n}. \tag{55}$$

Otherwise, applying the algorithm of Lemma 17(ii), as in (45) we obtain a cut of weight at least $m/2+\sqrt{m/8}+cm^{1/4}$, provided m is sufficiently large. For a similar reason, we may assume m is $O(e)$: otherwise, we may apply Lemma 16 to obtain a contraction of G that fails (55).

On the other hand, the algorithm of Lemma 17(i) gives a bipartite subgraph of weight at least $w(G)/2 + \big(|G|-1\big)/4$ and so we can halt the algorithm unless

$$|G| < n + 14c\sqrt{n}. \tag{56}$$

Now we show that we may assume that G is nearly complete. We run the algorithm in Lemma 16 on G with edge-weighting u defined by $u(e) = \max\left\{ w(e), 0\right\}$ for $e \in E(G)$, and obtain a contraction to a weighted-complete graph H. If H satisfies

$$|H| \le n - 12c\sqrt{n} \tag{57}$$

then applying the same contraction to G with edge-weighting w gives a graph with weight at least $w(G)$ satisfying (55), which we partition with the algorithm of Lemma 17. Otherwise let W^- be the set of vertices of G which are identified with some other vertex in the contraction from G to H. Then Y^- meets all edges of G with weight at most 0, and

$$|Y^-| \le 2\big(|G| - |H|\big) \le 52c\sqrt{n}. \tag{58}$$

Let $Y^+ = V(G)\setminus Y^-$. Then $G[Y^+]$ is a complete graph in which all edges have positive weight, and it follows from (55) and (58) that

$$|Y^+| > n - 64c\sqrt{n}. \tag{59}$$

We claim that $G[Y^+]$ cannot contain too many edges with weight greater than 1. We choose greedily a maximal matching $e_1, \dots, e_t$ in $G[Y^+]$, where at each stage we choose an edge of weight more than 1 if possible, and otherwise any edge. Using the algorithm of Lemma 17(i) we obtain in time $O(e)$ a cut of weight at least $\frac{1}{2}w(Y^+) + \frac{1}{2}\sum_{i=1}^t w(e_i)$. By Lemma 17(iii), we can extend this to a cut of G of weight at least $\frac{1}{2}w(G) + \frac{1}{2}\sum_{i=1}^t w(e_i)$. It follows from (54 that either we can halt the algorithm or $\sum_{i=1}^t w(e_i) \le n/2 + 6\sqrt{n}$. By (59), $\{e_1, \dots, e_t\}$ contains fewer than $40c\sqrt{n}$ edges of weight more than 1, and since they were chosen greedily these edges meet every edge with weight greater than 1. Therefore there is a set Z of vertices that meets every edge in $G[Y^+]$ with weight greater than 1, where

$$|Z| \le 80c\sqrt{n}. \tag{60}$$

Let $Y = Y^- \cup Z$ and $X = V(G) \setminus Y$. Then by (58) and (60),

$$|Y| \le 144c\sqrt{n}$$

while by (55),

$$|X| \ge n - 156c\sqrt{n}. \tag{61}$$

The rest of the algorithm is broadly similar to the algorithm in the proof of Theorem 13. Note that $G[X]$ is a complete graph in which all edges have weight 1. As before, for each $y \in Y$ we partition X into sets Z_1 and Z_2 according to the weight of edges from Y, and then extend to a partition of G. As in (48), we obtain a cut of weight at least

$$\frac{w(G)}{2} + \frac{|X|-1}{4} + \frac{w(y, Z_2) - w(y, Z_1)}{2}$$

so by (61) we are done unless

$$w(y, Z_2) - w(y, Z_1) < 85c\sqrt{n}.$$

It follows that all but at most $85c\sqrt{n}$ edges between y and X have the same weight $t(y)$. As before, let $t(x) = 1$ for $x \in X$, and for $x, y \in V(G)$ define

$$u(xy) = w(xy) - t(x)t(y).$$

Once again, we calculate $U = \sum_{x\in X, y\in Y} \left|u(xy)\right|$. If $U/cn^{3/4}$ is sufficiently large then as in (50) and (51) we can find a bipartite subgraph of weight at least $m/2 + \sqrt{m/8} + cm^{1/4}$.

Otherwise, U is $O(n^{3/4})$, so provided m is sufficiently large we have $U < n/4$ and we may decompose G as the edge sum of a graph H with edge weights given by u and a graph K_t^* with edge weights given by t. Finally, note that $|K_t^*| = n + O\left(\sqrt{n}\right) = n + O(m^{1/4})$, while a simple calculation shows that if $f(G) \leq m/2 + \sqrt{m/8} + cm^{1/4}$ then $t = n + O(1)$; note that K_t^* is obtained from K_t by a sequence of at most $O(m^{1/4})$ contractions. ■

Note that in the decomposition $G = K_t^* \oplus H$ of Theorem 21 we have $f(G) = f(K_t^*) + f(H)$. Furthermore, it is straightforward to find an optimal partition of K_t^* by splitting into two classes with t-weight as equal as possible.

We can now prove the main theorem of this section.

Theorem 22. *There is an algorithm running in time $O(2^{ck^4} + e + n)$ that, given a weighted graph G with e edges, edge-weighting w and total weight m, and an integer k, finds a cut of G with weight $f(G)$ if $f(G) \leq m/2 + \sqrt{m/8} + k$ and otherwise a cut with weight at least $m/2 + \sqrt{m/8} + k$.*

Proof. If $\sum_{e\in E(G)} \left|w(e)\right| > 4(m + 8k^2)$ then we are done by Theorem 20. If $k > m^{1/4}$ then we examine all partitions explicitly. Otherwise, we apply Theorem 21 with $c = 1$: either we obtain a cut of weight at least $m/2 + \sqrt{m/8} + k$, or we obtain a decomposition $G = K_t^* \oplus H$ where $f(G) = f(K_t^*) + f(H)$ and K_t^*, H have edge-weightings w_1, w_2 respectively. We can calculate $f(K_t^*)$ exactly; so if $w_2(H) > 0$, define l by $m/2 + \sqrt{m/8} + k = f(K_t^*) + w_2(H)/2 + \sqrt{w_2(H)/8} + l$, and repeat the algorithm with H and l (note that $l \leq k$). If $w_2(H) \leq 0$ then define the edge-weighting u by $u(e) = \left|w_2(e)\right|$ for $e \in E(H)$. If $\left|u(H)\right| > 8k^2$ then Theorem 20 gives a cut of weight at least $w_2(H)/2 + k$, which gives a cut of G of weight at least $w_1(K_t^*)/2 + \sqrt{w_1(K_t^*)/8} + w_2(H)/2 + k \geq m/2 + \sqrt{m/8} + k$; otherwise, $\left|u(H)\right| < 8k^2$ and we can examine all partitions explicitly. ■

Theorem 22 gives a polynomial time algorithm for $k \le c(\log n)^{1/4})$. However, we cannot expect to go as high as n^ε for any $\varepsilon > 0$, since then considering graphs of form $K_N \cup G$, where $N > |G|^{2/\varepsilon}$, would give a polynomial time algorithm for Max Cut. In particular, we cannot expect to find optimal cuts for all graphs with $f(G) \le m/2 + \sqrt{m/8} + cm^{1/4}$. On the other hand, we have given an algorithm that finds a cut of weight at least $f_w(m)$, which can be as large as $m/2 + \sqrt{m/8} + cm^{1/4}$. We are led naturally to the following problem.

Problem 1. *Let $k > 0$ be a fixed integer. Is there a polynomial time algorithm that given a graph G of weight m finds either a cut of weight at least $f_w(m) + k$ or else an optimal cut?*

Note that $k = 0$ is Theorem 13. However, if m satisfies $m = \binom{n+1}{2} - n^\varepsilon$ and $f_w(m) = f(K_{n+1}) = \lfloor (n+1)^2/4 \rfloor$, for instance, then we may have $G = K_{n+1} \oplus H$, where H has weight $-n^\varepsilon$. Theorem 21 will find this decomposition, and H has a trivial cut of weight 0, but determining whether $f(H) \ge k$ is probably NP-hard.

7. A weak approximation algorithm for graphs that are close to extremal

In this section we concentrate on the existence of polynomial time algorithms for estimating the value of $f(G) - m/2 - \sqrt{m/8}$ for a graph of weight $m > 0$. Let us note first that we cannot expect an arbitrarily good approximation algorithm since this would immediately yield a good approximation algorithm for the Max Cut problem. For since $f(G) \ge m/2$, an algorithm approximating $f(G) - m/2 - \sqrt{m/8}$ within a factor $1 + \varepsilon$ gives $f(G)$ to within a factor $1 + \varepsilon/2$. Since Håstad [18] has shown that it is NP-hard to approximate $f(G)$ within any factor less than 17/16, it follows that it is NP-hard to approximate $f(G) - m/2 - \sqrt{m/8}$ to within any factor less than 9/8. However, we have the following weak approximation.

Theorem 23. *There is an algorithm running in time $O(e + n)$ that, given a graph G with e edges, edge-weighting w and total weight m, either finds* *an optimal partition or gives a real number α such that*

$$\frac{m}{2}+\sqrt{\frac{m}{8}}+m^{\alpha}\leq f(G)\leq\frac{m}{2}+\sqrt{\frac{m}{8}}+m^{4\alpha}.$$

Proof. Let $U=\sum_{e\in E(G)}\left|w(e)\right|$. We may assume $U>U_0$, for some large constant U_0, or else examine all partitions explicitly. If $U>4m$ then Theorem 20 gives a cut of weight at least $m/2+\sqrt{U/8}-1>m/2+\sqrt{m/8}+U^{1/4}$, so we can choose $\alpha=(\log_m U)/4$. Otherwise, applying the algorithm of Theorem 22 with $c=4$ gives either a cut of weight at least $m/2+\sqrt{m/8}+4m^{1/4}$, so we can choose α such that $m^{\alpha}=4m^{1/4}$ (note that $f(G)\leq U$), or else a decomposition of G as $K_t^*\oplus H$, where H has weighting w'.

If $w'(H)\leq 0$ then let $U'=\sum_{e\in E(H)}\left|w'(e)\right|$. If $U'<U_0$ we determine $f(H)$, and hence also $f(G)=f(K_t^*)+f(H)$, explicitly. Otherwise, since we have $f(G)\leq f(K_t^*)+U'$ and $f(G)=f(K_t^*)+f(H)\geq m/2+\sqrt{m/8}+\sqrt{U'/8}-2$, we may pick α such that $m/2+\sqrt{m/8}+m^{\alpha}=f(K_t^*)+\sqrt{U'/8}-2$.

Otherwise $w'(H)\geq 0$, so repeating the algorithm we can find β such that

$$\frac{w'(H)}{2}+\sqrt{\frac{w'(H)}{8}}+w'(H)^{\beta}\leq f(H)\leq\frac{w'(H)}{2}+\sqrt{\frac{w'(H)}{8}}+w'(H)^{4\beta}.$$

Then we may pick α such that

$$\frac{m}{2}+\sqrt{\frac{m}{8}}+m^{\alpha}=f(K_t^*)+\frac{w'(H)}{2}+\sqrt{\frac{w'(H)}{8}}+w'(H)^{\beta}.$$

■

Compare the problem of approximating $f(G)-m/2-\sqrt{m/8}$ with the problem of approximating $f(G)$. Since $f(G)\geq m/2$ for every graph G with weight m, there is a "cushion" of weight $m/2$ in measuring the effectiveness of approximation algorithms for $f(G)$. For instance, the trivial decision algorithm that always returns the value $m/2$ and the greedy algorithm that achieves this bound manage to approximate within a factor 2 (improving beyond this is a different matter, however, and the 1.1383-approximation algorithm of Goemans and Williamson [17] is a tour de force). This contrasts with other approximation problems, such as Chromatic Number or Independent Set, where no such trivial approximation algorithm exists.

The problem of approximating $f(G) - m/2 - \sqrt{m/8}$ therefore seems more difficult than that of approximating $f(G)$.

Problem 2. *Is there a polynomial time algorithm that approximates* $f(G) - m/2 - \sqrt{m/8}$ *within a constant factor for graphs of weight* m*? What about* $f(G) - f_w(m)$*?*

Slightly easier, perhaps, is the following.

Problem 3. *Is there a polynomial time algorithm that approximates* $f(G) - m/2$ *within a constant factor for graphs of weight* m*?*

Note that the graphs that concern us here are graphs with $f(G) \leq 0.63w(G)$. For graphs with $f(G) > 0.63w(G)$, the algorithm of Goemans and Williamson gives an approximation algorithm for $f(G) - m/2 - \sqrt{m/8}$.

Part III: Related problems

8. The Max k-Cut problem

In the first two parts of this paper we have concentrated entirely on the Max Cut problem. In this section, we give a brief account of related results for Max k-Cut. For an edge-weighted graph G we write $f_k(G)$ for the maximal weight of a k-cut; we consider an unweighted graph as having weight 1 on every edge, so $f_k(G)$ is the maximal size of a k-partite subgraph. We define $f_k(m)$ to be the minimum of $f_k(G)$ over graphs with integer edge-weightings and total weight m.

We begin with an analogue of the Edwards bound (1) for k-partite graphs. Note first that if $n = rk + s$, where $0 \leq s < k$, then writing

$m = \binom{n}{2}$,

$$(62) \qquad f_k(K_n) = \binom{n}{2} - s\binom{r+1}{2} - (k-s)\binom{r}{2}$$

$$= \left(1 - \frac{1}{k}\right)\binom{n}{2} + \frac{r(k-1)}{2} + \frac{s(s-1)}{2k}$$

$$= \left(1 - \frac{1}{k}\right)\binom{n}{2} + \frac{k-1}{2k}n - \frac{s(k-s)}{2k}$$

$$\geq \left(1 - \frac{1}{k}\right)m + \frac{k-1}{2k}n - \frac{k}{8}$$

$$= \left(1 - \frac{1}{k}\right)m + \frac{k-1}{2k}\sqrt{2m + \frac{1}{4}} + \frac{k^2 - 2k + 2}{8},$$

with equality if and only if k is even and $s = k/2$.

Theorem 24. *Let G be a weighted graph with total weight m. Then*

$$(63) \qquad f_k(G) \geq \left(1 - \frac{1}{k}\right)m + \frac{k-1}{2k}\sqrt{2m + \frac{1}{4}} + \frac{k^2 - 2k + 2}{8}$$

If $m \geq \binom{n}{2}$ then

$$f_k(G) \geq f_k(K_n)$$

where for $m > m_0$ we have equality if and only if G is a copy of K_n, where all edges have weight 1.

Proof. Note first that, writing $g(m)$ for the right hand side of (63), it is easily checked that $g(m)$ is monotonic increasing, while $g(m)/m$ is monotonic decreasing for $m > 0$. We consider G as a weighted complete graph. Contracting any edge of weight at most 0 we obtain a weighted complete graph H in which all edges have positive weight; furthermore, any partition of H corresponds to a partition of G with the same cut-weight. It is therefore enough to prove the theorem for weighted complete graphs in which all edges have positive weight.

Now let $r = |H|$ and $R = \binom{r}{2}$, so $w(H) \geq R$. A random partition of H into k pieces, of size as equal as possible, gives a subgraph with expected weight

$$\frac{f_k(K_r)}{R} w(H) \geq \frac{g(R)}{R} w(H) \geq g\big(w(H)\big)$$

since $g(R)/R \geq g\big(w(H)\big)/w(H)$. Thus there must be some k-partite subgraph with weight at least $g\big(w(H)\big) \geq g(m)$.

If $m \geq \binom{n}{2}$, $f_k(G) = f_k(K_n)$ and G either has an edge of negative weight or is not isomorphic to K_n with all edge-weights 1, then compressing all edges of negative weight (and possibly some further edges), we may assume that $|G|$ is a complete weighted graph on fewer than n vertices, with all edges of positive weight and $w(G) \geq \binom{n}{2}$. Now since $f(K_n)/\binom{n}{2}$ is monotonic decreasing and $f(K_{n-k})/\binom{n-k}{2} > f(K_n)/\binom{n}{2}$, considering random partitions into k sets of size $\big\lfloor |G|/k \big\rfloor$ and $\big\lceil |G|/k \big\rceil$. We see that $|G| > n - k$ and so G can be decomposed as the edge sum of $K_{|G|}$ and a weighted graph with at most $(k-1)n$ edges. Provided m is sufficiently large, it is now straightforward to construct two k-cuts (into sets of size $\big\lfloor |G|/k \big\rfloor$ and $\big\lceil |G|/k \big\rceil$) with different weights, and so some k-cut must have weight more than $w(G) f\big(K_{|G|}\big)/\binom{|G|}{2} \geq f(K_n)$. ■

Ngoc and Tuza [30] prove that, for $k \geq 3$, every connected graph G with order n, size m and $\chi(G) > k$ satisfies

$$f_k(G) \geq \left(1 - \frac{1}{k}\right) m + \frac{1}{k}(n-1) + \frac{k-3}{2}. \tag{64}$$

Furthermore, they give an $O(m)$ algorithm that achieves this quickly. They note that every graph obtained by attaching trees to the vertices of a copy of K_k is extremal.

Let us note that if G has a vertex of degree at most $k-1$ then

$$f_k(G) = f_k(G \setminus v) + d_G(v)$$

since deleting v from G and partitioning the rest of the graph optimally, we can then add v to a class in which it has no neighbours. Thus we can restrict our attention to graphs G with $\delta(G) \geq k$. If we allow the minimal degree to grow, then we can improve on (64).

Theorem 25. *Let G be a graph with $\delta(G) \geq \omega(n)$, where $\omega(n) \to \infty$ as $n \to \infty$. Then*

$$f_k(G) \geq \left(1 - \frac{1}{k}\right) m + \frac{k-1}{2k} n + o(n).$$

Proof. (Sketch) Take a random ordering and run the greedy algorithm. If x has $rk+s$ predecessors, where $0 \leq s \leq k-1$, then the most even partition of earlier vertices has s big classes and $k-s$ small classes. Thus we can pick a class in which x has at most rk neighbours, increasing the cutweight by at least $r(k-1)+s = \left(1-\frac{1}{k}\right)(rk+s)+s/k$. Since s is asymptotically uniformly distributed on $\{0,\ldots,k-1\}$, the expectation of s/k tends to $(k-1)/2$. Summing over all vertices gives the bound above. ■

The proof of Theorem 25 works in the same way as the proof of Lemma 2: we order the vertices and then partition greedily. One method for obtaining a good ordering was given immediately after Lemma 2, and it is natural to ask whether the same approach (taking an ordering in which every vertex except one has at least one predecessor and then successively modifying the order to obtain better orders) would work for larger k. In fact, the method does work for $k=3$, but for $k>3$ appears to give a poor bound. It would be interesting to know under what conditions the $o(n)$ term can be removed in Theorem 25. For instance, a connectivity condition may be enough.

Problem 4. *Does every $(k-1)$-connected graph G satisfy*

$$f_k(G) \geq \frac{k-1}{k} m + \frac{k-1}{2k} n + O(1).$$

A best possible bound of this form would be $f_k(G) \geq \frac{k-1}{k} m + \frac{k-1}{2k} n - \frac{r(k-r)}{2k}$. A similar question arises if we assume $\delta(G) \geq k$. It would also be interesting to find efficient algorithms giving bounds of this form.

The results of earlier sections mostly carry over to the k-partite case.

Theorem 26. *Let G be a graph with edge-weighting w and total weight $w(G) = m$. Then provided m is sufficiently large*

$$f_k(G) \geq \min_{n \geq 0} \left\{ f_k(K_n) + f_k \left(m - \binom{n}{2} \right) \right\}.$$

As in the proof of Theorem 1, we need several lemmas. An analogue to Lemma 2 is given by Theorem 25, while using a greedy algorithm as in the proof of Lemma 7 we see that, for any graph G with edge-weighting w and any set $W \subset V(G)$, writing $H = G[W]$,

$$f_k(G) \geq f_k(H) + \frac{k-1}{k} \big(w(G) - w(H) \big) \tag{65}$$

A version of Lemma 4 is given by Theorem 24; this leaves Lemma 3. We have the following straightforward result.

Lemma 27. *For a nonempty edge-weighted graph G with total weight m,*

$$f_k(G) \geq \frac{k-1}{k} \left(1 + \frac{1}{\chi(G)} \right) m.$$

Proof. Fix a colouring of G with $c = \chi(G)$ colours and, as in Lemma 3, take a random partition of the colour classes into k sets of as equal size as possible. The expected size of the cut obtained is at least

$$\frac{f(K_c)}{\binom{c}{2}} m \geq \frac{k-1}{k} \left(1 + \frac{1}{c} \right) m.$$

■

Finally, we remark that as in the bipartite case there is $c_k > 0$ such that, for all m,

$$f_k(m) \leq \frac{k-1}{k} m + \frac{k-1}{2k} \sqrt{2m} + c_k m^{1/4}. \tag{66}$$

We can now proceed to the proof of the theorem.

Proof of Theorem 26 (Sketch). Let G be a counterexample with smallest weight. As in the proof of Theorem 8, we may contract edges of weight at most 0, and assume that G is a weighted complete graph with all edges of positive weight. Define n as in (25), so $|G| \leq n$. Now it follows from Lemma 27 that

$$f_k(G) \geq \frac{k-1}{k}\left(1+\frac{1}{|G|}\right)m.$$

Therefore, by (66),

$$\frac{k-1}{k}\frac{m}{|G|} \leq \frac{k-1}{2k}\sqrt{2m}+c_k m^{1/4}$$

and so

$$|G| \geq n - c_1\sqrt{n}. \tag{67}$$

Now let $C_1, \ldots, C_r$ be any collection of pairwise vertex-disjoint complete subgraphs of G, each with k vertices. We consider a random partition $V(G) = V_1 \cup \ldots \cup V_k$ as follows. Each C_i is randomly partitioned with one vertex in each vertex class V_j, and the remaining vertices are independently assigned to each vertex class with equal probability. Then the expected weight of the resulting k-partite subgraph is

$$\sum_{i=1}^{r} w(C_i) + \frac{k-1}{k}\left(w(G) - \sum_{i=1}^{r} w(C_i)\right) = \frac{k-1}{k}w(G) + \frac{1}{k}\sum_{i=1}^{r} w(C_i).$$

Thus it follows from (66) that

$$\sum_{i=1}^{r} w(C_i) \leq \frac{1}{2}(k-1)n + c_2\sqrt{n}. \tag{68}$$

Now if we greedily choose $C_1, C_2, \ldots$ to be vertex disjoint complete subgraphs with k vertices and maximal weight, it follows from (68) and (67) that at most $c_3\sqrt{n}$ of these subgraphs have weight more than $\binom{k}{2}$. Thus there is a set Y of $c_3k\sqrt{n}$ vertices meeting all edges with weight more than 1. Setting $X = V(G) \setminus Y$, we may assume that

$$|Y| \leq c_4\sqrt{n}$$

and

$$|X| \geq n - c_4\sqrt{n}.$$

Note that $G[X]$ is a complete graph in which all edges have weight 1.

Now for $y \in Y$, we can order the edges between y and X in increasing order of weight. Partitioning X into k pieces and using (65) we see that, as in (31), all but at most $c_5\sqrt{n}$ of the edges between y and X have the same weight, say $t(y)$.

As before, we set $t(y) = 1$ for $y \in X$, and define

$$u(xy) = t(x)t(y) - w(xy).$$

the remainder of the proof of Theorem 8 goes through essentially unchanged, except with k vertex classes instead of 2. ■

We note that it should be straightforward to generalize the algorithms of Sections 5, 6 and 7 to the k-partite case.

9. Maximum Cuts in directed graphs

In this final section, we make a few remarks concerning the analogue of Max Cut for directed graphs. For a directed graph H with edge-weighting w and a subset $S \subset V(H)$, we define

$$w(S, V \setminus S) = \sum_{x \in S} \sum_{y \in V \setminus S} w(xy)$$

where $w(xy) = 0$ if $xy \notin E(H)$. Then

$$g(H) = \max_{S \subset V(H)} w(S, V \setminus S).$$

For $m \geq 1$ we define $g(m)$ to be the maximum of $g(H)$ over all directed graphs whose edges are weighted with non-negative integers and have total weight m. It is easy to see that

$$g(m) \geq \lceil f(m)/2 \rceil .$$

Indeed, given a directed graph H, let G be the underlying (undirected) graph, where we define

$$w_G(xy) = w(xy) + w(yx).$$

Then G has total weight m, and therefore a cut of weight at least $f(m)$. Let $V(G) = V_1 \cup V_2$ be such a cut: then

$$w_H(V_1, V_2) + w_H(V_2, V_1) = w_G(V_1, V_2) \geq f(m)$$

so one of $w_H(V_1, V_2)$ and $w_H(V_2, V_1)$ is at least $\lceil f(m)/2 \rceil$.

Lemma 28. *If $m = \binom{2n+1}{2}$ then $g(m) = \binom{n+1}{2} = f(m)/2$.*

Proof. It is enough to construct a directed graph H with $w(H) = m$ and $g(H) \leq f(m)/2$. We define H by taking the vertices of $\mathbb{Z}_{2n+1}$, and adding edges from i to $i + j$, for any i and $1 \leq j \leq n$. Thus H is a regular oriented tournament, every vertex has indegree and outdegree n. Consider an arbitrary subset $S \subset V(H)$ with $|S| = h$. Clearly

$$g\big(S, V(H) \setminus S\big) = \sum_{v \in S} d^+(v) - e(S) \tag{69}$$

$$= nk - \binom{k}{2}$$

This is maximal when $k = n$ or $k = n + 1$, when

$$g(S, V(H) \setminus S) = n^2 - \binom{n}{2} = \binom{n+1}{2}. \qquad \blacksquare$$

Let us determine the extremal graphs for this result. Suppose H is a directed graph with $w(H) = m = \binom{2n+1}{2}$ and $g(H) = \binom{n+1}{2}$. Let G be the underlying graph: then $f(G) \leq n(n+1) = f(m)$. So by Lemma 4, G is a copy of K_{2n+1} in which all edges have weight 1. Therefore H is a directed tournament of order $2n+1$. If H is not regular, then pick n vertices with as large an outdegree as possible. Since their average outdegree must be greater than n it follows that we do better than (69). Since we must have equality in n, H must be a regular tournament. However, it follows from (69) that any regular tournament is extremal. Similar results follow when m is of form $\binom{2n+1}{2} + \binom{2k+1}{2}$, for $n > k$ sufficiently large, and so on, using Theorem 1 and Theorem 12.

What about if $m = \binom{2n}{2}$? The closest we have to a regular tournament in this case is a tournament with $2n$ vertices in which n vertices have outdegree $n - 1$ and n vertices have outdegree n. If we take a subset S of $k \leq n$ vertices, then

$$w(S, V \setminus S) = \sum_{x \in S} d^+(x) - \binom{k}{2} \leq kn - \binom{k}{2}.$$

This can be attained by taking the vertices with outdegree n and is then maximal when $k = n$, and

$$w(S, V \setminus S) = n^2 - \binom{n}{2} = \binom{n+1}{2}.$$

For $k \geq n + 1$, picking k vertices with outdegrees as large as possible, we have

$$w(S, V \setminus S) = n^2 + (k - n)(n - 1) - \binom{k}{2}.$$

This is maximal when $k = n + 1$, and

$$w(S, V \setminus S) = n^2 + n - 1 - \binom{n+1}{2} = \binom{n+1}{2} - 1.$$

Thus if H is a tournament of this form then

$$f(H) = \binom{n+1}{2}.$$

Clearly this is optimal among tournaments. However, if $m = \binom{2n}{2}$ then $f(m) = n^2$, so $g(H) = f(m)/2 + n/2$.

It would be interesting to determine the behaviour of $g(m) - f(m)/2$ for arbitrary m. It seems likely that, with more work, results similar to those proved above for Max Cut could be proved for the directed problem.

References

[1] N. Alon, Bipartite Subgraphs, *Combinatorica,* **16** (1996), 301–311.

[2] N. Alon and E. Halperin, Bipartite subgraphs of integer-weighted graphs, *Discrete Math.,* **181** (1998), 19–29.

[3] L. D. Anderson, D. D. Grant and N. Linial, Extremal k-colourable subgraphs, *Ars Combinatoria,* **16** (1983), 259–270.

[4] S. Arora, D. Karger and M. Karpinski, Polynomial time approximation schemes for dense instances of NP-hard problems, in: *Proc. 27th ACM STOC* (1995), 284–293.

[5] H. L. Bodlaender and K. Jansen, On the complexity of the Maximum Cut problem, in: *G. Goos and J. Hartmanis, eds, 11th Annual Symposium on Theoretical Aspects of Computer Science, LNCS 775,* 769–780.

[6] B. Bollobás and A. D. Scott, Exact bounds for judicious partitions, *Combinatorica,* to appear.

[7] B. Bollobás and A. D. Scott, Problems on judicious partitions, to appear.

[8] R. G. Downey and M. R. Fellows, Fixed parameter tractability and completeness I: Basic theory, *SIAM J. Comput.,* **24** (1995), 873–921.

[9] C. S. Edwards, Some extremal properties of bipartite subgraphs, *Canadian J. Math.,* **25** (1973), 475–485.

[10] C. S. Edwards, An improved lower bound for the number of edges in a largest bipartite subgraph, Proc 2nd Czech. Symposium on Graph Theory, Prague (1975), 167–181.

[11] P. Erdős, Some recent problems in Combinatorics and Graph Theory, Proc. 26th Southeastern International Conference on Graph Theory, Combinatorics and Computing, Boca Raton, 1995, Congressus Numerantium, to appear.

[12] P. Erdős, A. Gyárfás and Y. Kohayakawa, The size of the largest bipartite subgraphs, *Discrete Maths.,* **177** (1997), 267–271.

[13] W. Fernandez-de-la-Vega and M. Karpinski, Polynomial time approximability of dense weighted instances of Max Cut, Research report no 85171-CS, University of Bonn, 1997.

[14] W. Fernandez de la Vega, Max-cut has a randomized approximation scheme in dense graphs, *Random Structures and Algorithms,* **8** (1996), 187–198.

[15] A. Frieze and M. Jerrum, Improved approximation algorithms for MAX K-CUT and MAX BISECTION, *Algorithmica,* **18** (1997), 67–81.

[16] M. R. Garey, D. S. Johnson and L. Stockmeyer, Some simplified NP-complete graph problems, *Theoretical Computer Science,* **1** (1976), 237–267.

[17] M. X. Goemans and D. P. Williamson, 0.878 approximation algorithms for MAX CUT and MAX 2-SAT, Proc. 26th ACM Symp. Theory of Comput., 1994, 422–431; see also Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, *J. ACM,* **42** (1995), 1115–1145.

[18] J. Håstad, Some optimal inapproximability results, in: *Proc. 29th ACM Symp. on Theory of Comp.* (1995), 1–10.

[19] T. Hofmeister and H. Lefmann, A combinatorial design approach to Max Cut, *Random Structures and algorithms,* **9** (1996), 163–175.

[20] T. Hofmeister and H. Lefmann, On k-partite subgraphs, *Ars Combinatoria,* **50** (1998), 303–308.

[21] D. S. Johnson, The NP-completeness column: an ongoing guide, *J. Algorithms,* **6** (1985), 434–451.

[22] Y. Kajitani, J. D. Cho and M. Sarrafzadeh, New approximation results on graph matching and realted problems, Proc. 20th Workshop on Graph Theoretic Concepts in Computer Science, LNCS 903, 1994, 343–358.

[23] V. Kann, S. Khanna, J. Lagergren, A. Panconesi, On the Hardness of Approximating Max k-Cut and Its Dual, 1997, Article 2, *Chicago J. Theoretical Comp. Sci.*

[24] R. M. Karp, Reducibility among combinatorial problems, in: *R. E. Miller and J. W. Thatcher, eds, Complexity of Computer Computations,* Putnam Press (New York, 1972), 85–103.

[25] M. Karpinski, Polynomial time approximation schemes for some dense instances of NP-hard optimization problems, in: *Randomization and approximation techniques in computer science* (Bologna 1997), Lecture Notes in Computer Science **1269**, 1–14.

[26] M. Laurent, Max-Cut problem, in: *Annotated bibliographies in combinatorial optimization,* M. Dell'Amico, F. Maffioli and S. Martello, eds, Wiely, 1997, 241–259.

[27] J. Lehel and Zs. Tuza, Triangle-free partial graphs and edge-covering theorems. *Discrete Math.,* **30** (1982), 59–63.

[28] S. C. Locke, Maximum k-colourable subgraphs, *J. Graph Theory* **6** (1982), 123–132.

[29] M. Mahajan and V. Raman, Parametrizing above guaranteed values: MaxSat and MaxCut, *J. Algorithms* **31** (1999), 335–354.

[30] Nguyen Van Ngoc and Zs. Tuza, Linear time approximation algorithms for the Max Cut problem, *Combinatorics, Probability and Computing* **2** (1993), 201–210.

[31] S. Poljak and D. Turzik, A polynomial time heuristic for certain subgraph problems with guaranteed worst case bound, *Discrete Math.,* **58** (1986), 99–104.

[32] S. Poljak and Zs. Tuza, Maximum cuts and largest bipartite subgraphs, in: *Combinatorial optimization, DIMACS Ser. in Discrete Mathematics and Theoretical Computer Science,* volume 20, W. Cook, L. Lovász and P. Seymour, eds, 1995, xi+4 41pp., 181–244.

[33] S. Poljak and D. Turzik, A polynomial algorithm for constructing a large bipartite subgraph, with an application to a satisfiability problem, *Canadian J. Math.,* **34** (1982), 519–524.

[34] H. Saran and V. V. Vazirani, Finding k-cuts within twice the optimal, in: *Proc. 32nd Annual IEEE Symposium on Foundations of Computer Science* (1991).

[35] Zs. Tuza, Maximum cuts: improvements and local algorithmic analogues of the Edwards-Erdős inequality, *Discrete Math.,* **194** (1999), 39–58.

B. Bollobás

Trinity College
Cambridge CB2 1TQ

and

Department of Mathematical Sciences
University of Memphis
Memphis TN38152

`e-mail:`
`bollobas@ibex.msci.memphis.edu`

A. D. Scott

Department of Mathematics
University College London
Gower Street
London WC1E 6BT

`e-mail: scott@math.ucl.ac.uk`

and

Department of Mathematical Sciences
University of Memphis
Memphis TN38152