

# Parameterizing MAX SNP problems above Guaranteed Values

Meena Mahajan, Venkatesh Raman, Somnath Sikdar

The Institute of Mathematical Sciences,  
C.I.T Campus, Taramani, Chennai 600113.  
{meena,vraman,somnath}@imsc.res.in

**Abstract.** We show that every problem in MAX SNP has a lower bound on the optimum solution size and that the above guarantee question with respect to that lower bound is fixed parameter tractable. We next introduce the notion of ‘tight’ upper and lower bounds for the optimum solution and show that the parameterized version of a variant of the above guarantee question with respect to the tight lower bound cannot be fixed parameter tractable unless  $P = NP$ , for a number of NP-optimization problems.

## 1 Introduction

In this paper, we consider the parameterized complexity of NP-maximization problems  $Q$  with the following property: for each instance  $I$  of  $Q$ , the optimum  $\text{opt}(I)$ , is lower-bounded by a function of the input size. That is, for all instances  $I$ ,  $\text{opt}(I) \geq f(|I|)$  for some function  $f$ . For such an optimization problem  $Q$ , the standard parameterized version  $\tilde{Q}$  defined by

$$\tilde{Q} = \{(I, k) : I \text{ is an instance of } Q \text{ and } \text{opt}(I) \geq k\}$$

is easily seen to be fixed parameter tractable. For if  $k \leq f(|I|)$ , we answer ‘yes’; else,  $f(|I|) < k$  and so  $|I| < f^{-1}(k)$ <sup>1</sup> and we have a kernel.

Thus for such an optimization problem it makes sense to define an “above guarantee” parameterized version  $\bar{Q}$  as

$$\bar{Q} = \{(I, k) : I \text{ is an instance of } Q \text{ and } \text{opt}(I) \geq f(|I|) + k\}.$$

Such above guarantee parameterized problems were first considered by Mahajan and Raman in [5]. The problems dealt with by them are MAX SAT and MAX CUT. An instance of the MAX SAT problem is a boolean formula  $\phi$  in conjunctive normal form and the standard parameterized version asks whether  $\phi$  has at least  $k$  satisfiable clauses,  $k$  being the parameter. Since any boolean formula  $\phi$  with  $m$  clauses has at least  $\lceil m/2 \rceil$  satisfiable clauses (see Motwani and Raghavan [6]), by the above argument, this problem is fixed parameter tractable. The above guarantee MAX SAT question considered in [5] asks whether a given formula  $\phi$  has at least  $\lceil m/2 \rceil + k$  satisfiable clauses, with  $k$  as parameter. This was shown to be fixed parameter tractable.

The standard parameterized version of the MAX CUT problem asks whether an input graph  $G$  has a cut of size at least  $k$ , where  $k$  is the parameter. This problem is also fixed parameter tractable since any graph  $G$  with  $m$  edges has a cut of size  $\lceil m/2 \rceil$ . The above guarantee MAX CUT question considered in [5] asks whether an input graph  $G$  on  $m$  edges has a cut of size at least  $\lceil m/2 \rceil + k$ , where  $k$  is the parameter. This problem was shown to be fixed parameter tractable too.

To date, very few results have been published on the parameterized complexity of above guarantee problems. See for instance [2, 9]. In this paper, we consider above guarantee questions for

---

<sup>1</sup> Assuming  $f$  to be invertible; the functions considered in this paper are.

problems in the class MAX SNP. This paper is structured as follows. In Section 2, we introduce the necessary ideas about parameterized complexity and state some basic definitions needed in the rest of the paper. In Section 3, we show that every problem in the class MAX SNP has a guaranteed lower bound and that the above guarantee problem with respect to this lower bound is fixed parameter tractable. In Section 4, we define a notion of *tight lower bound* and show that a variant of the above guarantee question with respect to tight lower bounds is hard (unless  $P = NP$ ) for a number of NP-maximization problems. Finally in Section 5, we end with a few concluding remarks.

## 2 Preliminaries

We briefly introduce the necessary concepts concerning optimization problems and parameterized complexity.

To begin with, a parameterized problem is a subset of  $\Sigma^* \times \mathbb{N}$ , where  $\Sigma$  is a finite alphabet and  $\mathbb{N}$  is the set of natural numbers. An instance of a parameterized problem is therefore a pair  $(I, k)$ , where  $k$  is the parameter. In the framework of parameterized complexity, the run time of an algorithm is viewed as a function of two quantities: the size of the problem instance *and* the parameter. A parameterized problem is said to be *fixed parameter tractable* (fpt) if there exists an algorithm for the problem with time complexity  $O(f(k) \cdot |I|^{O(1)})$ , where  $f$  is a recursive function of  $k$  alone. The class FPT consists of all fixed parameter tractable problems.

A parameterized problem  $\pi_1$  is *fixed-parameter-reducible* to a parameterized problem  $\pi_2$  if there exist functions  $f, g : \mathbb{N} \rightarrow \mathbb{N}$ ,  $\Phi : \Sigma^* \times \mathbb{N} \rightarrow \Sigma^*$  and a polynomial  $p(\cdot)$  such that for any instance  $(I, k)$  of  $\pi_1$ ,  $(\Phi(I, k), g(k))$  is an instance of  $\pi_2$  such that  $\Phi(I, k)$  is computable in time  $f(k) \cdot p(|I|)$  and  $(I, k) \in \pi_1$  if and only if  $(\Phi(I, k), g(k)) \in \pi_2$ .

An NP-optimization problem  $Q$  is a 4-tuple  $Q = \{\mathcal{I}, S, V, \text{opt}\}$ , where

1.  $\mathcal{I}$  is the set of input instances. It is assumed that  $\mathcal{I}$  can be recognized in polynomial time.
2.  $S(x)$  is the set of feasible solutions for the input  $x \in \mathcal{I}$ .
3.  $V$  is a polynomial-time computable function called the *objective function* and for each  $x \in \mathcal{I}$  and  $y \in S(x)$ ,  $V(x, y) \in \mathbb{N}$ .
4.  $\text{opt} \in \{\max, \min\}$ .
5. The following decision problem (called the *underlying decision problem*) is in NP: Given  $x \in \mathcal{I}$  and an integer  $k$ , does there exist a feasible solution  $y \in S(x)$  such that  $V(x, y) \geq k$ , when  $Q$  is a maximization problem (or,  $V(x, y) \leq k$ , when  $Q$  is a minimization problem).

The class MAX SNP was defined by Papadimitriou and Yannakakis [7] using logical expressiveness. They showed that a number of interesting optimization problems such as MAX 3-SAT, INDEPENDENT SET-B, MAX CUT, MAX  $k$ -COLORABLE SUBGRAPH etc. lie in this class. They also introduced the notion of MAX SNP-completeness by using a reduction known as the *L-reduction*. We define this next.

Let  $\pi_1$  and  $\pi_2$  be two optimization (maximization or minimization) problems. We say that  $\pi_1$  *L-reduces* to  $\pi_2$  if there are two polynomial-time algorithms  $f, g$ , and constants  $\alpha, \beta > 0$  such that for each instance  $I_1$  of  $\pi_1$ :

1. Algorithm  $f$  produces an instance  $I_2 = f(I_1)$  of  $\pi_2$ , such that the optima of  $I_1$  and  $I_2$ ,  $\text{opt}(I_1)$  and  $\text{opt}(I_2)$ , respectively, satisfy  $\text{opt}(I_2) \leq \alpha \cdot \text{opt}(I_1)$ .
2. Given any solution of  $I_2$  with cost  $c_2$ , algorithm  $g$  produces a solution of  $I_1$  with cost  $c_1$  such that  $|c_1 - \text{opt}(I_1)| \leq \beta \cdot |c_2 - \text{opt}(I_2)|$

We call such an  $L$ -reduction from  $\pi_1$  to  $\pi_2$  as an  $\langle f, g, \alpha, \beta \rangle$  reduction.

A problem  $\pi$  is MAX SNP-hard if every problem in the class MAX SNP  $L$ -reduces to  $\pi$ . A problem  $\pi$  is MAX SNP-complete, if  $\pi$  is in MAX SNP and is MAX SNP-hard.

### 3 Parameterizing above Guaranteed Values

Consider the problem MAX 3-SAT which is complete for the class MAX SNP. An instance of MAX 3-SAT is a boolean formula  $f$  in conjunctive normal form with at most three literals per clause. As already stated, any boolean formula with  $m$  clauses has at least  $\lceil m/2 \rceil$  satisfiable clauses and thus the following above guarantee parameterized problem is fixed parameter tractable.

$$L = \{ \langle f, k \rangle : f \text{ is a MAX 3-SAT instance and } \exists \text{ an assignment satisfying} \\ \text{at least } k + \lceil m/2 \rceil \text{ clauses of the formula } f \}.$$

Since MAX 3-SAT is MAX SNP-complete and has a guaranteed lower bound, we have

**Proposition 1** *Every maximization problem in MAX SNP has a polynomial-time computable lower bound. That is, if  $\pi \in$  MAX SNP, then for each instance  $x$  of  $\pi$  there exists a positive number  $\gamma_x$  such that  $\gamma_x \leq \text{opt}(x)$ .*

*Proof.* Let  $\pi$  be a problem in MAX SNP and let  $\langle f, g, \alpha, \beta \rangle$  be an  $L$ -reduction from  $\pi$  to MAX 3-SAT. Then for an instance  $x$  of  $\pi$ ,  $f(x)$  is an instance of MAX 3-SAT such that  $\text{opt}(f(x)) \leq \alpha \cdot \text{opt}(x)$ . If  $f(x)$  is a formula with  $m$  clauses, then  $\lceil m/2 \rceil \leq \text{opt}(f(x))$  and therefore  $\text{opt}(x)$  is bounded below by  $\lceil m/2 \rceil / \alpha$ . This proves that each instance  $x$  of  $\pi$  has a lower bound.

We can express this lower bound in terms of the parameters of the  $L$ -reduction. Since  $f(x)$  is an instance of MAX 3-SAT, we can take the size of  $f(x)$  to be  $m$ . Then  $\gamma_x = \lceil m/2 \rceil / (2 \cdot \alpha)$ . ■

Note that this lower bound  $\gamma_x$  depends on the complete problem to which we reduce  $\pi$ . By changing the complete problem, we might construct different lower bounds for the problem at hand. It is also conceivable that there exist more than one  $L$ -reduction between two optimization problems. Different  $L$ -reductions should give different lower bounds. Thus the polynomial-time computable lower bound that we exhibit in Proposition 1 is a special lower bound obtained from a specific  $L$ -reduction to a specific complete problem (MAX 3-SAT) for the class MAX SNP. Call the lower bound of Proposition 1 a MAX 3-SAT-lower bound for the problem  $\pi$ .

Since the above guarantee parameterized version  $L$  of MAX 3-SAT is known to be FPT, we immediately have the following.

**Theorem 1.** *If  $\pi$  is a maximization problem in MAX SNP and if  $\gamma_x$  represents the MAX 3-SAT-lower bound for an instance  $x$  of  $\pi$  then the following parameterized problem is FPT:*

$$L_\pi = \{ \langle x, k \rangle : x \text{ is an instance of } \pi \text{ and } \text{opt}(x) \geq \gamma_x + k \}$$

*Proof.* We will make use of the fact that there exists a fixed parameter tractable algorithm  $\mathcal{A}$  for MAX 3-SAT which takes as input, a pair of the form  $\langle \psi, k \rangle$ , and in time  $O(|\psi| + h(k))$ , returns YES if there exists an assignment to the variables of  $\psi$  that satisfies at least  $\lceil m/2 \rceil + k$  clauses, and NO otherwise. See [5, 10] for such algorithms.

Consider an instance  $\langle x, k \rangle$  of  $L_\pi$ . Let  $\langle f, g, \alpha, \beta \rangle$  be an  $L$ -reduction from  $\pi$  to MAX 3-SAT. Then  $f(x)$  is an instance of MAX 3-SAT. Let  $f(x)$  have  $m$  clauses. Then the guaranteed lower bound for the instance  $x$  of  $\pi$ ,  $\gamma_x = \frac{m}{2\alpha}$ , and  $\text{opt}(f(x)) \leq \alpha \cdot \text{opt}(x)$ . Apply algorithm  $\mathcal{A}$  with input  $\langle f(x), k\alpha \rangle$ . If  $\mathcal{A}$  outputs YES, then  $\text{opt}(f(x)) \geq m/2 + k \cdot \alpha$ , implying  $\text{opt}(x) \geq \frac{m}{2\alpha} + k = \gamma_x + k$ . Thus  $\langle x, k \rangle \in L_\pi$ .

If  $\mathcal{A}$  answers NO, then  $\lceil \frac{m}{2} \rceil \leq \text{opt}(f(x)) < \lceil \frac{m}{2} \rceil + k\alpha$ . Apply algorithm  $\mathcal{A}$   $k\alpha$  times on inputs  $(f(x), 1), (f(x), 2), \dots, (f(x), k\alpha)$  to obtain  $\text{opt}(f(x))$ . Let  $c' = \text{opt}(f(x))$ . Then use algorithm  $g$  of the  $L$ -reduction to obtain a solution to  $x$  with cost  $c$ . By the definition of  $L$ -reduction, we have  $|c - \text{opt}(x)| \leq \beta \cdot |c' - \text{opt}(f(x))|$ . But since  $c' = \text{opt}(f(x))$ , it must be that  $c = \text{opt}(x)$ . Therefore we simply need to compare  $c$  with  $\gamma_x + k$  to check whether  $\langle x, k \rangle \in L_\pi$ .

The total time complexity of the above algorithm is  $O(k\alpha \cdot (|f(x)| + h(k\alpha)) + p_1(|x|) + p_2(|f(x)|))$ , where  $p_1(\cdot)$  is the time taken by algorithm  $f$  to transform an instance of  $\pi$  to an instance of MAX 3-SAT, and  $p_2(\cdot)$  is the time taken by  $g$  to output its answer. Thus the algorithm that we outlined is indeed an FPT algorithm for  $L_\pi$ . ■

Note that the proof of Proposition 1 also shows that every minimization problem in MAX SNP has a MAX 3-SAT-lower bound. For minimization problems whose optimum is lower bounded by some function of the input, it makes sense to ask how far removed the optimum is with respect to the lower bound. The parameterized question asks whether for a given input  $x$ ,  $\text{opt}(x) \leq \gamma_x + k$ , with  $k$  as parameter. The following result can be proved similarly to Theorem 1.

**Theorem 2.** *If  $\pi$  is a minimization problem in MAX SNP and if  $\gamma_x$  represents the MAX 3-SAT-lower bound for an instance  $x$  of  $\pi$  then the following parameterized problem is FPT:*

$$L_\pi = \{\langle x, k \rangle : x \text{ is an instance of } \pi \text{ and } \text{opt}(x) \leq \gamma_x + k\}$$

Examples of minimization problems in MAX SNP include VERTEX COVER- $B$  and DOMINATING SET- $B$ , the restriction of the VERTEX COVER and the DOMINATING SET problems to graphs whose vertex degree is bounded by  $B$ .

## 4 Hardness Results

For an optimization problem, the question of whether the optimum is at least lower bound  $+k$ ,  $k$  being the parameter, is not always interesting because if the lower bound is “loose” then the problem is trivially fixed parameter tractable. For instance, for the MAX CUT problem, the question of whether an input graph has a cut of size at least  $\frac{m}{2} + k$  is fpt since any graph  $G$  with  $m$  edges,  $n$  vertices and  $c$  components has a cut of size at least  $\frac{m}{2} + \lceil \frac{n-c}{4} \rceil$ . Thus if  $k \leq \lceil \frac{n-c}{4} \rceil$ , we answer YES; else,  $\lceil \frac{n-c}{4} \rceil < k$  and we have a kernel.

We therefore examine the notion of a *tight lower bound* and the corresponding above guarantee question. A tight lower bound is essentially the best possible lower bound on the optimum solution size. For the MAX SAT problem, this lower bound is  $m/2$ : if  $\phi$  is an instance of MAX SAT, then  $\text{opt}(\phi) \geq m/2$ , and there are infinitely many instances for which the optimum is *exactly*  $m/2$ . This characteristic motivates the next definition.

**Definition 1 (Tight Lower Bound)** Let  $Q = \{\mathcal{A}, S, V, \text{opt}\}$  be an NPO problem and let  $f : \mathbb{N} \rightarrow \mathbb{N}$ . We say that  $f$  is a *tight lower bound* for  $Q$  if the following conditions hold:

1.  $f(|I|) \leq \text{opt}(I)$  for all  $I \in \mathcal{I}$ .
2. There exists an infinite family of instances  $\mathcal{I}' \subseteq \mathcal{I}$  such that  $\text{opt}(I) = f(|I|)$  for all  $I \in \mathcal{I}'$ .

Note that we define the lower bound to be a function of the *input size* rather than the input itself. This is in contrast to the lower bound of Proposition 1 which depends on the input instance. We can define the notion of a tight *upper* bound analogously.

**Definition 2 (Tight Upper Bound)** Let  $Q = \{\mathcal{I}, S, V, \text{opt}\}$  be an NPO problem and let  $g : \mathbb{N} \rightarrow \mathbb{N}$ . We say that  $g$  is a **tight upper bound** for  $Q$  if the following conditions hold:

1.  $\text{opt}(I) \leq g(|I|)$  for all  $I \in \mathcal{I}$ .
2. There exists an infinite family of instances  $\mathcal{I}' \subseteq \mathcal{I}$  such that  $\text{opt}(I) = g(|I|)$  for all  $I \in \mathcal{I}'$ .

Some example optimization problems which have tight lower and upper bounds are given below. The abbreviations TLB and TUB stand for tight lower bound and tight upper bound, respectively.

### 1. MAX EXACT $c$ -SAT

INSTANCE    A boolean formula  $F$  with  $n$  variables and  $m$  clauses with each clause having *exactly*  $c$  distinct literals.

QUESTION    Find the maximum number of simultaneously satisfiable clauses.

BOUNDS      TLB =  $(1 - \frac{1}{2^c})m$ ;            TUB =  $m$ .

The expected number of clauses satisfied by the random assignment algorithm is  $(1 - \frac{1}{2^c})m$ ; hence the lower bound. To see tightness, note that if  $\phi(x_1, \dots, x_c)$  denotes the EXACT  $c$ -SAT formula comprising of all possible combinations of  $c$  variables, then  $\phi$  has  $2^c$  clauses of which exactly  $2^c - 1$  clauses are satisfiable. By taking disjoint copies of this formula one can construct EXACT  $c$ -SAT instances of arbitrary size with exactly  $(1 - \frac{1}{2^c})m$  satisfiable clauses.

### 2. CONSTRAINT SATISFACTION PROBLEM (CSP)

INSTANCE    A system of  $m$  linear equations modulo 2 in  $n$  variables, together with positive weights  $w_i$ ,  $1 \leq i \leq m$ .

QUESTION    Find an assignment to the variables that maximizes the total weight of the satisfied equations.

BOUNDS      TLB =  $\frac{W}{2}$ , where  $W = \sum_{i=1}^m w_i$ ;            TUB =  $W$ .

If we use  $\{+1, -1\}$ -notation for boolean values with  $-1$  corresponding to true, we can write the  $i$ th equation of the system as  $\pi_{j \in \alpha_i} x_j = b_i$ , where each  $\alpha_i$  is a subset of  $[n]$  and  $b_i \in \{+1, -1\}$ . To see that we can satisfy at least half the equations in the weighted sense, we assign values to the variables sequentially and simplify the system as we go along. When we are about to give a value to  $x_j$ , we consider all equations reduced to the form  $x_j = b$ , for a constant  $b$ . We choose a value for  $x_j$  satisfying at least half (in the weighted sense) of these equations. This procedure of assigning values ensures that we satisfy at least half the equations in the weighted sense. A tight lower bound instance, in this case, is a system consisting of pairs  $x_j = b_i, x_j = \bar{b}_i$ , with each equation of the pair assigned the same weight. See [3] for more details.

### 3. MAX INDEPENDENT SET-B

INSTANCE    A graph  $G$  with  $n$  vertices such that the degree of each vertex is bounded by  $B$ .

QUESTION    Find a maximum independent set of  $G$ .

BOUNDS      TLB =  $\frac{n}{B+1}$ ;            TUB =  $n$ .

A graph whose vertex degree is bounded by  $B$  can be colored using  $B + 1$  colors and in any valid coloring of the graph, the vertices that get the same color, form an independent set. By the pigeon hole principle, there exists an independent set of size at least  $n/(B + 1)$ . The complete graph  $K_{B+1}$  on  $B + 1$  vertices has an independence number of  $\frac{n}{B+1}$ . By taking disjoint copies of  $K_{B+1}$  one can construct instances of arbitrary size with independence number exactly  $\frac{n}{B+1}$ .

#### 4. MAX PLANAR INDEPENDENT SET

INSTANCE A planar graph  $G$  with  $n$  vertices and  $m$  edges.

QUESTION Find a maximum independent set of  $G$ .

BOUNDS TLB =  $\frac{n}{4}$ ; TUB =  $n$ .

A planar graph is 4-colorable and in any valid 4-coloring of the graph, the vertices that get the same color form an independent set. By the pigeon hole principle, there exists an independent set of size at least  $\frac{n}{4}$ . A disjoint set of  $K_4$ 's can be use to construct arbitrary sized instances with independence number exactly  $\frac{n}{4}$ .

#### 5. MAX ACYCLIC DIGRAPH

INSTANCE A directed graph  $G$  with  $n$  vertices and  $m$  edges.

QUESTION Find a maximum acyclic subgraph of  $G$ .

BOUNDS TLB =  $\frac{m}{2}$ ; TUB =  $m$ .

To see that any digraph with  $m$  arcs has an acyclic subgraph of size  $\frac{m}{2}$ , place the vertices  $v_1, \dots, v_n$  of  $G$  on a line in that order with arcs  $(v_i, v_j)$ ,  $i < j$ , drawn above the line and arcs  $(v_i, v_j)$ ,  $i > j$ , drawn below the line. Clearly, by deleting all arcs either above or below the line we obtain an acyclic digraph. By the pigeonhole principle, one of these two sets must have size at least  $\frac{m}{2}$ . To see that this bound is tight, consider the digraph  $D$  on  $n$  vertices:  $v_1 \leftrightarrow v_2 \leftrightarrow v_3 \leftrightarrow \dots \leftrightarrow v_n$  which has a maximum acyclic digraph of size exactly  $\frac{m}{2}$ . Since  $n$  is arbitrary, we have an infinite set of instances for which the optimum matches the lower bound exactly.

#### 6. MAX PLANAR SUBGRAPH

INSTANCE A connected graph  $G$  with  $n$  vertices and  $m$  edges.

QUESTION Find an edge-subset  $E'$  of maximum size such that  $G[E']$  is planar.

BOUNDS TLB =  $n - 1$ ; TUB =  $3n - 6$ .

Any spanning tree of  $G$  has  $n - 1$  edges; hence any maximum planar subgraph of  $G$  has at least  $n - 1$  edges. This bound is tight as the family of all trees achieve this lower bound. An upper bound is  $3n - 6$  which is tight as for each  $n$ , a maximal planar graph on  $n$  vertices has exactly  $3n - 6$  edges.

#### 7. MAX CUT

INSTANCE A graph  $G$  with  $n$  vertices,  $m$  edges and  $c$  components.

QUESTION Find a maximum cut of  $G$ .

BOUNDS TLB =  $\frac{m}{2} + \lceil \frac{n-c}{4} \rceil$ ; TUB =  $m$ .

The lower bound for the cut size was proved by Poljak and Turzik [8]. This bound is tight for complete graphs. The upper bound is tight for bipartite graphs.

A natural question to ask in the above-guarantee framework is whether the language

$$L = \{\langle I, k \rangle : \text{opt}(I) \geq \text{TLB}(I) + k\}$$

is in FPT. The parameterized complexity of such a question is not known for most problems. To the best of our knowledge, this question has been resolved only for the MAX SAT and MAX  $c$ -SAT problems [5] and, very recently, for the LINEAR ARRANGEMENT problem [2].

In this section, we study a somewhat different, but related, parameterized question: Given an NP-maximization problem  $Q$  which has a tight lower bound (TLB) a function of the input size, what is the parameterized complexity of the following question?

$$Q(\epsilon) = \{\langle I, k \rangle : \text{opt}(I) \geq \text{TLB}(I) + \epsilon \cdot |I| + k\}$$

Here  $|I|$  denotes the input size,  $\epsilon$  is some fixed positive rational and  $k$  is the parameter. We show that this question is not fixed parameter tractable for a number of problems, unless  $P = NP$ .

**Theorem 3.** *The  $Q(\epsilon)$  problem is not fixed parameter tractable for the following problems  $Q$  unless  $P = NP$ :*

Problem	$\text{TLB}(I) + \epsilon \cdot  I  + k$	Range of $\epsilon$
1. MAX SAT	$(\frac{1}{2} + \epsilon)m + k$	$0 < \epsilon < \frac{1}{2}$
2. MAX $c$ -SAT	$(\frac{1}{2} + \epsilon)m + k$	$0 < \epsilon < \frac{1}{2}$
3. MAX EXACT $c$ -SAT	$(1 - \frac{1}{2^c} + \epsilon)m + k$	$0 < \epsilon < \frac{1}{2^c}$
4. CSP	$(\frac{1}{2} + \epsilon)m + k$	$0 < \epsilon < \frac{1}{2}$
5. PLANAR INDEPENDENT SET	$(\frac{1}{4} + \epsilon)n + k$	$0 < \epsilon < \frac{3}{4}$
6. INDEPENDENT SET- $B$	$(\frac{1}{B+1} + \epsilon)n + k$	$0 < \epsilon < \frac{B}{B+1}$
7. MAX ACYCLIC SUBGRAPH	$(\frac{1}{2} + \epsilon)m + k$	$0 < \epsilon < \frac{1}{2}$
8. MAX PLANAR SUBGRAPH	$(1 + \epsilon)n - 1 + k$	$0 < \epsilon < 2$
9. MAX CUT	$\frac{m}{2} + \lceil \frac{n-c}{4} \rceil + \epsilon n + k$	$0 < \epsilon < \frac{1}{4}$
10. MAX DICUT	$\frac{m}{4} + \sqrt{\frac{m}{32} + \frac{1}{256}} - \frac{1}{16} + \epsilon m + k$	$0 < \epsilon < \frac{3}{4}$

The proof, in each case, follows this outline: Assume that for some  $\epsilon$  in the specified range,  $Q(\epsilon)$  is indeed in FPT. Now consider an instance  $\langle I, s \rangle$  of the underlying decision version of  $Q$ . Here is a P-time procedure for deciding it. If  $s \leq \text{TLB}$ , then the answer is trivially YES. If  $s$  lies between  $\text{TLB}$  and  $\text{TLB} + \epsilon|I|$ , then “add” a gadget of suitable size corresponding to the TUB, to obtain an equivalent instance  $\langle I', s' \rangle$ . This increases the input size, but since we are adding a gadget whose optimum value matches the upper bound, the increase in the optimum value of  $I'$  is more than proportional, so that now  $s'$  exceeds  $\text{TLB} + \epsilon|I'|$ . If  $s$  already exceeds  $\text{TLB} + \epsilon|I|$ , then “add” a gadget of suitable size corresponding to the TLB, to obtain an equivalent instance  $\langle I', s' \rangle$ . This increases the input size faster than it boosts the optimum value of  $I'$ , so that now  $s'$  exceeds  $\text{TLB} + \epsilon|I'|$  by only a constant, say  $c$ . Use the hypothesised fpt algorithm for  $Q(\epsilon)$  with input  $\langle I', c \rangle$  to correctly decide the original question.

Rather than proving the details for each item separately, we use this proof sketch to establish a more general theorem (Theorem 4 below) which automatically implies items 1 through 8 above. It does not imply items 9 and 10, since MAX CUT and MAX DICUT do not meet the technical conditions of the theorem. However, the above proof idea works for MAX CUT and MAX DICUT as well; see the appendix for details.

The conditions in Theorem 4 are rather technical. So we first demonstrate the result for a specific problem (MAX 3-SAT). Recall that an instance of MAX 3-SAT is a boolean formula in

conjunctive normal form with  $m$  clauses and at most 3 literals per clause. One can easily verify that  $\lceil m/2 \rceil$  is a tight lower bound for this problem.

**Proposition 2** *Let  $\phi$  be an instance of the MAX 3-SAT problem with  $n$  variables and  $m$  clauses. If the following parameterized problem*

$$\text{MAX 3-SAT}(\epsilon) = \{ \langle \phi, k \rangle : \exists \text{ an assignment satisfying } \lfloor (1/2 + \epsilon) \cdot m + k \rfloor \text{ clauses of } \phi \}$$

*has a fixed parameter tractable algorithm  $\mathcal{A}$  for any  $0 < \epsilon < 1/2$ , then  $\text{P} = \text{NP}$ .*

*Proof.* Suppose, for the purpose of contradiction, that there exists an fpt algorithm  $\mathcal{A}$  for the problem MAX 3-SAT( $\epsilon$ ) with time complexity  $O(f(k) \cdot m^{O(1)})$ , for some  $0 < \epsilon < 1/2$ . We will use  $\mathcal{A}$  to decide the decision version of MAX 3-SAT, which is NP-complete, in polynomial time. An instance of the decision version of MAX 3-SAT is a pair  $\langle \phi, s \rangle$ , where  $\phi$  is a boolean formula on  $m$  clauses and  $n$  variables and  $s$  is an integer. The question is to decide whether there exists an assignment which satisfies  $s$  clauses of  $\phi$ .

*Case 1:*  $s \leq \lceil m/2 \rceil$ . Return YES.

*Case 2:*  $\lceil m/2 \rceil < s < \lfloor (\frac{1}{2} + \epsilon) \cdot m \rfloor$ .

We claim that in this case, we can transform the instance  $\langle \phi, s \rangle$  into an ‘equivalent’ instance  $\langle \phi', s' \rangle$  where  $\phi'$  is a 3-SAT instance on  $m'$  clauses and  $n'$  variables such that

$$\left\lfloor \left( \frac{1}{2} + \epsilon \right) \cdot m' \right\rfloor \leq s'. \quad (1)$$

Let  $y_1, \dots, y_t$  be  $t$  new variables, where  $t$  will be determined later. Define  $\phi'$  as

$$\phi' = \phi \wedge y_1 \wedge y_2 \wedge \dots \wedge y_t.$$

Set  $m' = m + t$ ,  $n' = n + t$  and  $s' = s + t$ . Clearly there exists an assignment that satisfies  $s$  clauses of  $\phi$  if and only if there exists an assignment that satisfies  $s + t$  clauses of  $\phi'$ . Choose  $t$  so that inequality (1) holds. We will show that such a  $t$  actually exists. Inequality (1) holds if we choose  $t$  so that

$$\left( \frac{1}{2} + \epsilon \right) \cdot m - s \leq \left( \frac{1}{2} - \epsilon \right) t.$$

Note that  $0 < (\frac{1}{2} + \epsilon) \cdot m - s \leq \epsilon m$  and  $(\frac{1}{2} - \epsilon) > 0$ , and so such a  $t$  is actually linear in  $m$ . Thus without loss of generality we can assume the input instance satisfies Case 3 below.

*Case 3:*  $\lfloor (\frac{1}{2} + \epsilon) \cdot m \rfloor \leq s$ .

In this case, we transform  $\langle \phi, s \rangle$  into an ‘equivalent’ instance  $\langle \psi, s' \rangle$ , where  $\psi$  is a 3-SAT instance with  $m'$  clauses and  $n'$  variables such that

1.  $s' = \lfloor (\frac{1}{2} + \epsilon) \cdot m' \rfloor$ .
2. There exists an assignment satisfying  $s$  clauses of  $\phi$  iff there exists an assignment satisfying  $s'$  clauses of  $\psi$ .

Let  $z_1, \dots, z_t$  be  $t$  new variables, where  $t$  will be determined later. Define  $\psi$  as follows:

$$\psi = \phi \wedge z_1 \wedge \bar{z}_1 \wedge \dots \wedge z_t \wedge \bar{z}_t.$$



$\psi$  is a 3-SAT instance on  $m + 2t$  clauses and  $n + t$  variables. Clearly there exists an assignment satisfying  $s$  clauses of  $\phi$  iff there exists an assignment satisfying  $s + t$  clauses of  $\psi$ . Choose  $t$  to be the largest non-negative integer satisfying

$$\left\lfloor \left( \frac{1}{2} + \epsilon \right) \cdot (m + 2t) \right\rfloor \leq s + t. \quad (2)$$

(Inequality 2 holds if and only if  $\lfloor (\frac{1}{2} + \epsilon) m + 2t\epsilon \rfloor \leq s$ .) Then, since  $2\epsilon < 1$ , it is easy to see that in fact equality is achieved in (2).

Now call algorithm  $\mathcal{A}$  with input  $\langle \psi, 0 \rangle$ . If  $\mathcal{A}$  answers YES, then there exists an assignment satisfying  $s + t$  clauses of  $\psi$ , and so there exists an assignment satisfying  $s$  clauses of  $\phi$ . Therefore, we answer YES. If  $\mathcal{A}$  answers NO, then there is no assignment that satisfies  $s + t$  clauses of  $\psi$  and hence no assignment satisfying  $s$  clauses of  $\phi$ . We answer NO. Therefore in time  $O(f(0) \cdot m^{O(1)})$  we can decide MAX 3-SAT implying  $P = NP$ . ■

To generalize the above argument, we first need some definitions.

**Definition 3 (Dense Set)** Let  $Q = \{\mathcal{I}, S, V, \text{opt}\}$  be an NPO problem. A set of instances  $\mathcal{I}' \subseteq \mathcal{I}$  is said to be **dense with respect to a set of conditions**  $\mathcal{C}$  if there exists a constant  $c \in \mathbb{N}$  such that for all closed intervals  $[a, b] \subseteq \mathbb{R}^+$  of length  $|b - a| \geq c$ , there exists an instance  $I \in \mathcal{I}'$  with  $|I| \in [a, b]$  such that  $I$  satisfies all the conditions in  $\mathcal{C}$ . Further, if such an  $I$  can be found in polynomial time (polynomial in  $b$ ), then  $\mathcal{I}'$  is said to be **dense poly-time uniform with respect to**  $\mathcal{C}$ .

For example, for the MAXIMUM ACYCLIC SUBGRAPH problem, the set of all oriented digraphs is dense (poly-time uniform) with respect to the condition:  $\text{opt}(G) = |E(G)|$ .

We also need the notion of a partially additive NP-optimization problem.

**Definition 4 (Partially Additive Problems)** An NPO problem  $Q = \{\mathcal{I}, S, V, \text{opt}\}$  is said to be **partially additive** if there exists an operator  $+$  which maps a pair of instances  $I_1$  and  $I_2$  to an instance  $I_1 + I_2$  such that

1.  $|I_1 + I_2| = |I_1| + |I_2|$ , and
2.  $\text{opt}(I_1 + I_2) = \text{opt}(I_1) + \text{opt}(I_2)$ .

A partially additive NPO problem that also satisfies the following condition is said to be additive in the framework of Khanna, Motwani et al [4]: there exists a polynomial-time computable function  $f$  that maps any solution  $s$  of  $I_1 + I_2$  to a pair of solutions  $s_1$  and  $s_2$  of  $I_1$  and  $I_2$ , respectively, such that  $V(I_1 + I_2, s) = V(I_1, s_1) + V(I_2, s_2)$ .

For graph-theoretic optimization problems, the operator  $+$  can be interpreted as disjoint union. Then the problems MAX CUT, MAX INDEPENDENT SET-B, MINIMUM VERTEX COVER, MINIMUM DOMINATING SET, MAXIMUM DIRECTED ACYCLIC SUBGRAPH, MAXIMUM DIRECTED CUT are partially additive. For other graph-theoretic problems, one may choose to interpret  $+$  as follows: given graphs  $G$  and  $H$ ,  $G + H$  refers to a graph obtained by placing an edge between a vertex of  $G$  and a vertex of  $H$ . The MAX PLANAR SUBGRAPH problem is partially additive with respect to both these interpretations of  $+$ . For boolean formulae  $\phi$  and  $\psi$  in conjunctive normal form with disjoint sets of variables, define  $+$  as the conjunction  $\phi \wedge \psi$ . Then the MAX SAT problem is easily seen to be partially additive.

**Theorem 4.** Let  $Q = \{\mathcal{I}, S, V, \text{opt}\}$  be a polynomially bounded NP-maximization problem such that the following conditions hold.

1.  $Q$  is partially additive.
2.  $Q$  has a tight lower bound (TLB) of the form  $\rho \cdot |I| + \alpha_1$ , where  $\rho$  is a positive rational and  $\alpha_1 \in \mathbb{Q}$ . The infinite family of instances  $\mathcal{I}'$  witnessing the tight lower bound is dense poly-time uniform with respect to the condition  $\text{opt}(I) = \rho \cdot |I| + \alpha_1$ .
3.  $Q$  has a tight upper bound  $g(|I|) = \sigma |I| + \alpha_2$ , where  $\sigma > \rho$  and  $\alpha_2 \in \mathbb{Q}$ . The infinite family of instances  $\mathcal{I}'$  witnessing the tight upper bound is dense poly-time uniform with respect to the condition  $\text{opt}(I) = g(|I|)$ .
4. The underlying decision problem  $\tilde{Q}$  of  $Q$  is NP-hard.

Define  $Q(\epsilon)$  to be the following parameterized problem

$$Q(\epsilon) = \{\langle I, k \rangle : \text{opt}(I) \geq \text{TLB}(I) + \epsilon \cdot |I| + k\}$$

where  $0 < \epsilon < \sigma - \rho$ . If  $Q(\epsilon)$  is FPT for any  $0 < \epsilon < \sigma - \rho$ , then  $\text{P} = \text{NP}$ .

*Proof.* Suppose, for the purpose of contradiction,  $Q(\epsilon)$  is in FPT for some  $0 < \epsilon < \sigma - \rho$ . Let  $\mathcal{A}$  be the fpt algorithm for  $Q(\epsilon)$  with running time  $O(f(k) \cdot \text{poly}(|I|))$ .

We will describe an algorithm  $\mathcal{A}'$  that makes use of  $\mathcal{A}$  to solve the decision version  $\tilde{Q}$  of  $Q$  in polynomial time. Let  $\langle I, s \rangle$  be an instance of  $\tilde{Q}$ .  $\langle I, s \rangle \in \tilde{Q}$  iff  $\text{opt}(I) \geq s$ . The algorithm  $\mathcal{A}'$  considers four cases and works as described below.

*Case 1:*  $s \leq \rho \cdot |I| + \alpha_1$ .

Since  $\text{opt}(I) \geq \rho \cdot |I| + \alpha_1$ ,  $\mathcal{A}'$  answers YES.

*Case 2:*  $\rho \cdot |I| + \alpha_1 < s < \rho \cdot |I| + \alpha_1 + \epsilon \cdot |I|$

In this case,  $\mathcal{A}'$  transforms  $\langle I, s \rangle$  into an instance  $\langle I', s' \rangle$  such that

1.  $(\rho + \epsilon) \cdot |I'| + \alpha_1 \leq s'$ .
2.  $|I'| = \text{poly}(|I|)$ .
3.  $\text{opt}(I) \geq s$  if and only if  $\text{opt}(I') \geq s'$ .

This will show that we can, without loss of generality, go to Cases 3 or 4 below. Let  $I_1$  be a TUB instance of  $Q$  (i.e.  $\text{opt}(I_1) = g(|I_1|)$ ), and let  $I' = I + I_1$ . Define  $s' = s + g(|I_1|)$ . From the definition of a partially additive NPO problem, it follows that  $\text{opt}(I') \geq s'$  if and only if  $\text{opt}(I) \geq s$ . Choose  $I_1$  so that

$$(\rho + \epsilon) \cdot |I'| + \alpha_1 \leq s' \tag{3}$$

Inequality (3) holds iff

$$(\rho + \epsilon) \cdot |I| + \alpha_1 \leq s + g(|I_1|) - (\rho + \epsilon) \cdot |I_1|. \tag{4}$$

Note that  $h(n) = g(n) - (\rho + \epsilon)n = (\sigma - \rho - \epsilon)n + \alpha_2 > 0$  for sufficiently large  $n \in \mathbb{N}$  and  $h$  is increasing. Thus by choosing a sufficiently large instance  $I_1$ , we can satisfy inequality (4). It is also important to note that we can choose  $I_1$  so that  $|I_1| \leq \text{poly}(|I|)$ . This follows because of the range in which  $s$  lies and the polynomial-time uniform density of the TUB.

*Case 3:*  $(\rho + \epsilon) \cdot |I| + \alpha_1 + p_0 < s$ , where  $p_0 = \lceil c\epsilon \rceil$  and  $c$  is the constant appearing in the definition of density.

In this case,  $\mathcal{A}'$  transforms the instance  $\langle I, s \rangle$  into an instance  $\langle I', s' \rangle$  such that

1.  $(\rho + \epsilon) \cdot |I'| + \alpha_1 \leq s' \leq (\rho + \epsilon) \cdot |I'| + \alpha_1 + p_0$ .
2.  $\text{opt}(I') \geq s'$  if and only if  $\text{opt}(I) \geq s$ .

This ensures that we need only consider Case 4 below. To do this, define  $k = s - (\rho + \epsilon) \cdot |I| - \alpha_1$ . By assumption,  $k > p_0$ . Choose a TLB instance  $I_1$  in the range  $\lceil (k + \alpha_1 - p_0)/\epsilon \rceil \leq |I_1| \leq \lfloor (k + \alpha_1)/\epsilon \rfloor$ . Such an instance exists by definition of density and our choice of  $p_0$ . Let  $I' = I + I_1$ . Then  $\text{opt}(I') \geq s + \rho \cdot |I_1| + \alpha_1$  if and only if  $\text{opt}(I) \geq s$ . So set  $s' = s + \rho \cdot |I_1| + \alpha_1$ . Note that  $|I'| = |I| + |I_1|$ , and so

$$\begin{aligned} s' - (\rho + \epsilon) \cdot |I'| - \alpha_1 &= s + \rho \cdot |I_1| - (\rho + \epsilon) \cdot |I'| \\ &= s - (\rho + \epsilon) \cdot |I| - \epsilon \cdot |I_1| \\ &= k + \alpha_1 - \epsilon \cdot |I_1| \geq 0 \end{aligned}$$

$$\begin{aligned} (\rho + \epsilon) \cdot |I'| + \alpha_1 + p_0 - s' &= (\rho + \epsilon) \cdot |I'| + \alpha_1 + p_0 - (s + \rho \cdot |I_1| + \alpha_1) \\ &= (\rho + \epsilon) \cdot |I| + \epsilon \cdot |I_1| + p_0 - s \\ &\geq (\rho + \epsilon) \cdot |I| + (k + \alpha_1 - p_0) + p_0 - s = 0 \end{aligned}$$

*Case 4:*  $(\rho + \epsilon) \cdot |I| + \alpha_1 \leq s \leq (\rho + \epsilon) \cdot |I| + \alpha_1 + p_0$ .

Now  $\langle I, s \rangle \in \tilde{Q}$  if and only if  $\langle I, k \rangle \in Q(\epsilon)$ , where  $k = s - (\rho + \epsilon) \cdot |I| - \alpha_1$ . Also,  $k \leq p_0$ . So  $\mathcal{A}'$  calls  $\mathcal{A}$  with input  $\langle I', k \rangle$  and answers accordingly. This takes time  $O(f(p_0) \cdot \text{poly}(|I|))$ .

Thus the total time taken to decide any instance  $\langle I, s \rangle$  of the decision version  $\tilde{Q}$  is polynomial in  $|I|$ , implying  $P = NP$ . ■

*Remark 1.* The result stated in Theorem 4 goes through if we assume the tight upper bound  $g$  to be a strictly increasing, polynomial-time computable, super-linear function. For such a function  $g$ , given any  $\epsilon > 0$ , there exists  $n_0 \in \mathbb{N}$  such that

$$h(n) = g(n) - (\rho + \epsilon) \cdot n - \alpha_1 > 0 \quad \forall n \geq n_0.$$

Moreover, the function  $h$  is strictly increasing for all  $n \geq n_0$ . We could therefore choose an instance  $I_1$  to satisfy inequality (4). The rest of the proof goes through as stated.

*Remark 2.* In order to apply Theorem 4 to prove items 1 through 8 of Theorem 3 one needs to interpret the  $+$  operator appropriately. Except for the MAX PLANAR SUBGRAPH, the operator  $+$  is to be interpreted throughout as disjoint union. For the MAX PLANAR SUBGRAPH problem,  $G + H$  is a graph consisting of  $G$  and  $H$  with an arbitrary edge connecting them. Thus if  $G$  and  $H$  are connected, so is  $G + H$ .

## 5 Conclusion

We have shown that every problem in MAX SNP has a lower bound on the optimal solution size and that the above guarantee question with respect to that lower bound is in FPT. We have also shown that the  $\text{TLB}(I) + \epsilon \cdot |I| + k$  question is hard for a general class that includes a number of NP-maximization problems. However we do not know the parameterized complexity of tight lower bound  $+ k$  questions for most NPO problems. In particular, apart from MAX SAT, MAX  $c$ -SAT and LINEAR ARRANGEMENT, this question is open for the rest of the problems stated in Theorem 3. It would be interesting to explore the parameterized complexity of these problems and above guarantee problems in general.

## References

1. N. ALON, B. BOLLABÁS, A. GYÁRFÁS, J. LEHEL AND A. SCOTT. *Maximum directed cuts in acyclic digraphs*. Available at: <http://www.math.tau.ac.il/~nogaa/PDFS/publications.html>
2. G. GUTIN, A. RAFIEY, S. SZEIDER AND A. YEO. *The Linear Arrangement Problem Parameterized Above Guaranteed Value*. Available at: <http://arxiv.org/abs/cs.DS/0511030>
3. J. HÅSTAD AND S. VENKATESH. *On the Advantage Over a Random Assignment*. Proceedings of the 34th Annual ACM Symposium on Theory of Computing, 2002, pages 43-52.
4. S. KHANNA, R. MOTWANI, M. SUDAN AND U. VAZIRANI. *On Syntactic Versus Computational Views of Approximability*. SIAM Jour. Computing. Vol. 28, No. 1, pp 164-191.
5. M. MAHAJAN AND V. RAMAN. *Parameterizing above Guaranteed Values: MaxSat and MaxCut*, Journal of Algorithms **31**, 335-354 (1999).
6. R. MOTWANI AND P. RAGHAVAN. *Randomized Algorithms*. Cambridge University Press, 1995.
7. C. H. PAPADIMITRIOU AND M. YANNAKAKIS. *Optimization, Approximation, and Complexity Classes*, JCSS **43**, 425-440 (1991).
8. S. POLJAK AND D. TURZIK. *A Polynomial Algorithm for Constructing a Large Bipartite Subgraph with an Application to a Satisfiability Problem*. Canadian Jour. Math., Vol. 34, No. 3, 1982, pp. 519-524.
9. R. NIEDERMEIER. *Invitation to Fixed Parameter Algorithms*. Oxford University Press, 2006.
10. P. ROSSMANITH AND R. NIEDERMEIER. *New Upper Bounds for Maximum Satisfiability*. Journal of Algorithms, 36: 63-88, 2000.

## 6 Appendix

The proof of hardness of the MAX CUT and MAX DICUT problems (items 9 and 10 of Theorem 3) do not follow from Theorem 4 because the tight lower bounds for cut size are, respectively,  $\frac{m}{2} + \lceil \frac{n-c}{4} \rceil$  and  $\frac{m}{4} + \sqrt{\frac{m}{32} + \frac{1}{256}} - \frac{1}{16}$  which are not of the form  $\rho \cdot |I| + \alpha_1$ .

We first prove the hardness result for MAX CUT. In what follows, we let  $\text{maxcut}(G)$  denote the size of the maximum cut of an undirected graph  $G$ .

**Theorem 5.** *For  $0 < \epsilon < \frac{1}{4}$  define MAX CUT( $\epsilon$ ) as follows:*

$$\text{MAX CUT}(\epsilon) = \left\{ (G, k) : \text{maxcut}(G) \geq \frac{m}{2} + \left\lceil \frac{n-c}{4} \right\rceil + \epsilon \cdot n + k \right\}.$$

*If MAX CUT( $\epsilon$ ) is fixed parameter tractable for any  $0 < \epsilon < \frac{1}{4}$  then P = NP.*

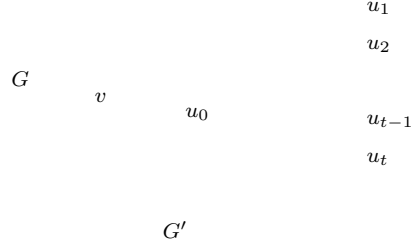
*Proof.* Suppose for the purpose of contradiction that there exists an  $0 < \epsilon < \frac{1}{4}$  for which the language MAX CUT( $\epsilon$ ) is fixed parameter tractable. Let  $\mathcal{A}$  be an fpt algorithm for MAX CUT( $\epsilon$ ) with running time  $O(f(k) \cdot \text{poly}(n+m))$ . We will use  $\mathcal{A}$  to solve the decision version of MAX CUT, which is NP-complete, in polynomial time.

Let  $(G, s)$  be an instance of the decision version of MAX CUT. Then  $(G, s)$  is a ‘yes’-instance if and only if  $\text{maxcut}(G) \geq s$ . Assume that  $G$  has  $n$  vertices,  $m$  edges and  $c$  components. If  $s \leq \frac{m}{2} + \lceil \frac{n-c}{4} \rceil$  then  $(G, s)$  is already a ‘yes’-instance. Otherwise there are two cases to handle.

*Case 1:*  $\frac{m}{2} + \lceil \frac{n-c}{4} \rceil < s < \frac{m}{2} + \lceil \frac{n-c}{4} \rceil + \epsilon n$

In this case, we transform the input instance  $(G, s)$  into an instance  $(G', s')$  of the same problem such that  $G'$  has  $m'$  edges,  $n'$  vertices,  $c$  components and the following hold

1.  $\text{maxcut}(G) \geq s$  if and only if  $\text{maxcut}(G') \geq s'$ ,
2.  $\frac{m'}{2} + \lceil \frac{n'-c}{4} \rceil + \epsilon n' \leq s'$ .



**Fig. 1.** Constructing the instance  $G'$ .

This will show that we can, without loss of generality, assume that  $s \geq \frac{m}{2} + \lceil \frac{n-c}{4} \rceil + \epsilon n$ . Let  $v$  be an arbitrary vertex of  $G$  and let  $u_0, u_1, \dots, u_t$  be  $t+1$  new vertices. Construct a star graph using the vertices  $u_0, u_1, \dots, u_t$  with  $u_0$  as the center of the star and add an edge between  $u_0$  and  $v$  as shown in Figure 1. We will choose the value of  $t$  later. Call the new graph  $G'$ . Then  $G'$  has  $m' = m + t + 1$  edges,  $n' = n + t + 1$  vertices and  $c$  components. Define  $s' = s + t + 1$ . It is easy to verify that  $\text{maxcut}(G) \geq s$  if and only if  $\text{maxcut}(G') \geq s'$ . Choose  $t$  so that

$$s' \geq \frac{m'}{2} + \left\lceil \frac{n' - c}{4} \right\rceil + \epsilon \cdot n. \quad (5)$$

The above inequality holds if and only if  $\frac{m}{2} + \lceil \frac{n-c}{4} \rceil + t + 1 \geq \frac{m'}{2} + \lceil \frac{n'-c}{4} \rceil + \epsilon \cdot n$ . One can verify that by choosing  $t$  so that  $t + 1 > \frac{\epsilon \cdot n}{1/4 - \epsilon}$  one can satisfy inequality 5.

*Case 2:*  $s \geq \frac{m}{2} + \lceil \frac{n-c}{4} \rceil + \epsilon n$

In this case, we transform the instance  $(G, s)$  into  $(G', s')$  such that  $G'$  has  $m'$  edges,  $n'$  vertices,  $c$  components and the following hold

1.  $\text{maxcut}(G) \geq s$  if and only if  $\text{maxcut}(G') \geq s'$ ,
2.  $s' = \frac{m'}{2} + \lceil \frac{n'-c}{4} \rceil + \epsilon n' + d$ , where  $d$  is a constant independent of the input instance  $(G, s)$ .

We then apply the fpt algorithm  $\mathcal{A}$  with input  $(G', d)$ . Clearly,  $\mathcal{A}$  answers ‘yes’ if and only if  $\text{maxcut}(G) \geq s$  if and only if  $\text{maxcut}(G') \geq s'$ . The time taken by  $\mathcal{A}$  is  $O(f(d) \cdot \text{poly}(m' + n'))$ .

Let  $v$  be an arbitrary vertex of  $G$ . Attach the complete graph on  $t$  vertices,  $K_t$ , to  $v$ . The value of  $t$  will be chosen later. Call the resulting graph  $G'$ .  $G'$  has  $n' = n + t$  vertices,  $m' = m + \binom{t}{2} + 1$  edges and  $c$  components. Define  $k = s - (\frac{m}{2} + \lceil \frac{n-c}{4} \rceil + \epsilon n)$  and  $s' = s + \frac{1}{2} \binom{t}{2} + \lceil \frac{t-1}{4} \rceil + 1$ . It is easy to see that  $\text{maxcut}(G) \geq s$  if and only if  $\text{maxcut}(G') \geq s'$ . One can verify that  $t$  can be suitably chosen so that

$$s' = \frac{m'}{2} + \left\lceil \frac{n' - c}{4} \right\rceil + \epsilon n' + d,$$

where  $d$  is a fixed constant independent of the input instance  $(G, s)$ . ■

The MAX DICUT problem is defined below.

MAX DICUT

**INSTANCE**    A directed graph  $D = (V, A)$  with  $m$  arcs.  
**QUESTION**    Find a vertex partition  $V = X \uplus Y$  such that the number of arcs with starting point in  $X$  and with end point in  $Y$  is maximized.  
**BOUNDS**      TLB =  $\frac{m}{4} + \sqrt{\frac{m}{32} + \frac{1}{256}} - \frac{1}{16}$ ;      TUB =  $m$ .

The tight lower bound is achieved in regular orientations of complete graphs of odd order [1]. The tight upper bound is achieved in bipartite directed graphs  $D = (X \uplus Y, A)$  with all arcs starting at  $X$  and ending at  $Y$ . We let  $\max \text{dicut}(D)$  denote the size of the maximum directed cut of  $D$ .

**Theorem 6.** For  $0 < \epsilon < \frac{3}{4}$  define  $\text{MAX DICUT}(\epsilon)$  as follows:

$$\text{MAX DICUT}(\epsilon) = \left\{ (D, k) : \max \text{dicut}(D) \geq \frac{m}{4} + \sqrt{\frac{m}{32} + \frac{1}{256}} - \frac{1}{16} + \epsilon \cdot m + k \right\}.$$

If  $\text{MAX DICUT}(\epsilon)$  is fixed parameter tractable for any  $0 < \epsilon < \frac{3}{4}$  then  $\text{P} = \text{NP}$ .

The proof of this is similar to that of Theorem 5 and is omitted.