Journal of Computer and System Sciences 75 (2009) 137–153

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Parameterizing above or below guaranteed values ☆

Meena Mahajan, Venkatesh Raman *, Somnath Sikdar

The Institute of Mathematical Sciences, C.I.T. Campus, Taramani, Chennai 600113, India

ARTICLE INFO

Article history:
Received 16 February 2008
Received in revised form 21 July 2008
Available online 22 August 2008

Keywords:
Parameterized complexity
NP-optimization problems
Fixed-parameter tractability
Above guarantee parameterizations

ABSTRACT

We consider new parameterizations of NP-optimization problems that have nontrivial lower and/or upper bounds on their optimum solution size. The natural parameter, we argue, is the quantity above the lower bound or below the upper bound. We show that for every problem in MAX SNP, the optimum value is bounded below by an unbounded function of the input-size, and that the above-guarantee parameterization with respect to this lower bound is fixed-parameter tractable. We also observe that approximation algorithms give nontrivial lower or upper bounds on the solution size and that the above or below guarantee question with respect to these bounds is fixed-parameter tractable for a subclass of NP-optimization problems.

We then introduce the notion of 'tight' lower and upper bounds and exhibit a number of problems for which the above-guarantee and below-guarantee parameterizations with respect to a tight bound is fixed-parameter tractable or W-hard. We show that if we parameterize "sufficiently" above or below the tight bounds, then these parameterized versions are not fixed-parameter tractable unless P = NP, for a subclass of NP-optimization problems. We also list several directions to explore in this paradigm.

1. Introduction and motivation

Parameterized complexity is an approach developed by Downey and Fellows for dealing with computationally hard problems where small parameter values cover many practical applications. Consider, for instance, the NP-complete Vertex Cover and Dominating Set problems. These problems are defined as follows: Given a graph G and a positive integer parameter k, decide whether G has a vertex cover (respectively, dominating set) of size at most k. Both problems can be solved in time $O(n^{k+2})$, where n is the number of vertices of G. What is interesting is that for the Vertex Cover problem there exists an algorithm with run-time $O(c^k \cdot n)$, where c is a constant, whereas for Dominating Set there is reason to believe that no such algorithm exists.

Parameterized complexity is mainly concerned with obtaining algorithms for parameterized problems with run-time $O(f(k) \cdot n^{O(1)})$, where f is a computable function of k alone, as against a run-time of $O(n^{O(k)})$. Here k is the parameter for the problem and n is the input size. A parameterized problem which admits an algorithm with run time $O(f(k) \cdot n^{O(1)})$ is called *fixed-parameter tractable* (FPT). We also use the term *FPT time* to describe running times of the form $O(f(k) \cdot n^{O(1)})$, where f, k and n are as specified above. The parameter k is not unique, that is, it is possible to parameterize a problem

☆ A preliminary version of this paper appeared in the Proceedings of the 2nd International Workshop on Parameterized and Exact Computation (IWPEC), 2006 [M. Mahajan, V. Raman, S. Sikdar, Parameterizing MAX SNP problems above-guaranteed values, in: Proceedings of the 2nd International Workshop on Parameterized and Exact Computation, in: Lecture Notes in Comput. Sci., vol. 4169, Springer, 2006, pp. 38–49. [29]].

* Corresponding author.
E-mail addresses: meena@imsc.res.in (M. Mahajan), vraman@imsc.res.in (V. Raman), somnath@imsc.res.in (S. Sikdar).

0022-0000/$ – see front matter
doi:10.1016/j.jcss.2008.08.004

in more than one way and using more than one parameter. For a comprehensive introduction to parameterized complexity see the classic monograph by Downey and Fellows [11] or the recent texts by Niedermeier [35] and Flum and Grohe [16].

For an NP-optimization problem Q, the *standard parameterized version* Q_{par} is the following decision problem.

Input: A tuple (I,k), where I is an instance of Q and k, the parameter, is a nonnegative integer.
Question: Is the optimum solution size of I at least k (if Q is a maximization problem) or at most k (if Q is a minimization problem)?

Although the standard parameterized version is the most popularly studied parameterization of NP-optimization problems, there are many problems for which the standard version is *trivially* fixed-parameter tractable. Consider for instance the problem Max c-Sat. An instance of this problem consists of a Boolean CNF formula with at most c literals per clause and the objective is to find an assignment which satisfies the maximum number of clauses. This problem is known to be NP-complete for $c \geqslant 2$. It is well known that if ϕ is a Boolean CNF formula with m clauses then there exists an assignment that satisfies at least $\lceil m/2 \rceil$ clauses and that such an assignment can be found in time $O(|\phi|)$ (see [33]).

Now consider what this means for the standard parameterized version of Max c-Sat, for $c \geqslant 2$. An instance of the parameterized version consists of a tuple of the form (ϕ, k), where ϕ is a Boolean CNF formula with m clauses and at most c literals per clause and k is a nonnegative integer. The question is whether there exists an assignment that satisfies at least k clauses of ϕ. If $k \leqslant \lceil m/2 \rceil$, then there exists an assignment which satisfies at least k clauses (because there is one that satisfies $\lceil m/2 \rceil$ clauses) and such an assignment can be obtained in linear time. We therefore answer yes and output the assignment. Otherwise, $k > \lceil m/2 \rceil$ and since every clause has at most c literals, we have $|\phi| \leqslant 2kc$. We now use a brute-force algorithm that looks at all possible assignments to the variables of ϕ (which are at most $2kc$ in number) and check whether any of these assignments satisfies at least k clauses of ϕ. This brute-force algorithm runs in time $O(2^{2kc} \cdot |\phi|)$ and is, by definition, an FPT-algorithm for Max c-Sat. Note, however, that when the brute-force algorithm is applied, k, and hence the running time is large for all practical purposes.

A similar state-of-affairs exists for several other problems: Max Cut, Planar Independent Set and Max Acyclic Subgraph to name a few. All these problems have some nontrivial lower bound for the optimum value which is exploited to give a trivial fixed-parameter algorithm. When the parameter value is below the default lower bound, the answer is no; otherwise, the standard brute-force algorithm itself gives a fixed-parameter algorithm since in this case, the parameter value is considerably large compared to the input size. However, as we observed before, these algorithms are not necessarily practical.

To deal with this problem, Mahajan and Raman [28] considered a different parameterization of Max Sat and Max Cut, where the parameter is the *difference* between the optimum value and the guaranteed lower bound. They showed that both these problems are FPT under this parameterization as well. More recently, the "above-guarantee" versions of Linear Arrangement [20] and Minimum Profile [21,23] have been shown to be in FPT. Our first observation in this paper is that not just these problems, but all optimization problems in MAX SNP have a nontrivial lower bound for the optimum value. We prove that the above-guarantee parameterization with respect to this lower bound is fixed-parameter tractable. We also observe that approximation algorithms give nontrivial lower or upper bounds on the solution size. We show that the above or below-guarantee question with respect to these bounds is fixed-parameter tractable whenever the standard parameterized versions of these problems is fixed-parameter tractable. This is dealt with in Section 3 after some definitions in Section 2 related to optimization problems and parameterized complexity.

We next show that parameterizing above any nontrivial lower bound may not be interesting because, for some nontrivial lower bounds, the above-guarantee question may still be trivially FPT. This motivates us to define what are known as *tight lower bounds*.

Almost all the problems we discussed also have nontrivial upper bounds for the optimum value, and another natural parameterization is to parameterize *below* the upper bound. For example, Max Sat has the number of clauses in the input formula, m, as an upper bound; the upper bound for Max Cut is the number of edges m in the graph. The natural *below-guarantee* parameterized questions are: can you satisfy all but k clauses and is there a cut of size at least $m-k$? The first is FPT for 2-Cnf Sat [40], and hard (not in FPT unless $\mathsf{P} = \mathsf{NP}$) for c-Cnf Sat for $c \geqslant 3$ [28]. The second problem is the well-known Odd Cycle Transversal problem which was shown to be FPT in [41]. Several vertex/edge-deletion problems, for example, König Vertex/Edge Deletion [32], Feedback Vertex Set [39], Planar Vertex Deletion [31], fit in the below-guarantee framework. In Section 4, we list a number of problems having tight upper and lower bounds.

In Section 5 we identify some problems for which the above-guarantee or below-guarantee version is unlikely to be in FPT. In Section 6 we show that if we parameterize "sufficiently above" tight lower bounds then the above-guarantee question becomes hard (unless $\mathsf{P} = \mathsf{NP}$) for a number of NP-optimization problems. We also show a similar result for the below-guarantee question with respect to tight upper bounds. Finally in Section 7 we list a number of interesting research directions.

2. Preliminaries

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

To begin with, a parameterized problem is a subset of $\Sigma^* \times \mathbb{Z}^{\geqslant 0}$, where Σ is a finite alphabet and $\mathbb{Z}^{\geqslant 0}$ is the set of nonnegative integers. 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 computable function of k alone. The class FPT consists of all fixed-parameter tractable problems. An FPT-algorithm for a parameterized problem is an algorithm that solves that problem in time $O(f(k) \cdot p(|I|))$ for some computable function f and polynomial p. We assume that an FPT-algorithm for a parameterized problem produces a witness whenever it outputs YES.

A parameterized problem π_1 is *fixed-parameter-reducible* to a parameterized problem π_2 if there exist functions $f, g : \mathbb{Z}^{\geqslant 0} \to \mathbb{Z}^{\geqslant 0}$, $\Phi : \Sigma^* \times \mathbb{Z}^{\geqslant 0} \to \Sigma^*$ and a polynomial $p(\cdot)$ such that for any instance (I, k) of π_1, $(\Phi(I, k), g(k))$ is an instance of π_2 computable in time $f(k) \cdot p(|I|)$ and $(I, k) \in \pi_1$ if and only if $(\Phi(I, k), g(k)) \in \pi_2$. If π_1 is fixed-parameter reducible to π_2 and π_2 if FPT, then so is π_1. A parameterized problem π_1 is *fixed-parameter equivalent* to a parameterized problem π_2 if π_1 is fixed-parameter reducible to π_2 and vice versa. Note that if π_1 and π_2 are fixed-parameter equivalent then π_1 is fixed-parameter tractable if and only if π_2 is.

We sometimes use the $O^*(\cdot)$ notation to describe the running times of algorithms that take exponential time. The O^* notation suppresses polynomial factors in the running time expression. For instance, we write $O^*(T(n))$ for a time-complexity of the form $O(T(n) \cdot \text{poly}(n))$ where $T(n)$ grows exponentially with n, the input size. A time-complexity of the form $O(f(k) \cdot \text{poly}(n))$ for an FPT-algorithm is sometimes written as $O^*(f(k))$.

An NP-optimization (NPO) problem Q is a four-tuple $Q = \{\mathscr{I}, S, V, \text{opt}\}$, where

1. $\mathscr{I}$ is the set of input instances. (Without loss of generality, $\mathscr{I}$ can be recognized in polynomial time.)
2. $S(x)$ is the set of feasible solutions for the input $x \in \mathscr{I}$.
3. V is a polynomial-time computable function called the *cost function* and for each $x \in \mathscr{I}$ 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 of* Q) is in NP: Given $x \in \mathscr{I}$ and an integer k, does there exist a feasible solution $y \in S(x)$ such that $V(x, y) \geqslant k$, when Q is a maximization problem (or, $V(x, y) \leqslant k$, when Q is a minimization problem).

From the above definition, it follows that the optimum value of every NP-optimization problem is bounded below by 1. We call a lower bound *trivial* if it is at most a constant. Given an NP-optimization problem Q, a *nontrivial lower bound* f for Q is an unbounded function $f : \mathbb{N} \to \mathbb{N}$ such that for all instances x of Q, we have $\text{opt}(x) \geqslant f(|x|)$. A *nontrivial upper bound* g is an unbounded function $g : \mathbb{N} \to \mathbb{N}$ such that for all instances x of Q, we have $\text{opt}(x) \leqslant g(|x|)$.

The class MAX SNP was defined by Papadimitriou and Yannakakis [37] 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 lie in this class. They also introduced the notion of completeness for MAX SNP by a reduction known as the *L-reduction*. We define this next.

Let Q_1 and Q_2 be two optimization (maximization or minimization) problems. We say that Q_1 L-reduces to Q_2 if there exist polynomial-time computable functions f, g, and constants $\alpha, \beta > 0$ such that for each instance I_1 of Q_1:

1. $f(I_1) = I_2$ is an instance of Q_2, such that $\text{opt}(I_2) \leqslant \alpha \cdot \text{opt}(I_1)$.
2. Given any solution y_2 of I_2, g maps (I_2, y_2) to a solution y_1 of I_1 such that $|V(I_1, y_1) - \text{opt}(I_1)| \leqslant \beta \cdot |V(I_2, y_2) - \text{opt}(I_2)|$.

We call such an L-reduction from Q_1 to Q_2 an (f, g, α, β)-reduction.

A problem Q is MAX SNP-*hard* if every problem in the class MAX SNP L-reduces to Q. A problem Q is MAX SNP-*complete*, if Q is in MAX SNP and is MAX SNP-hard. Some example MAX SNP-complete problems are MAX c-SAT for any constant c, INDEPENDENT SET-B, VERTEX COVER-B, DOMINATING SET-B, MAX CUT, MAX DIRECTED CUT, MAX k-COLORABLE SUBGRAPH. Cai and Chen [4] established that the standard parameterized version of all maximization problems in the class MAX SNP are fixed-parameter tractable. In the next section, we show that for all problems in MAX SNP, a certain above-guarantee question is also fixed-parameter tractable.

3. Parameterizing above or below-guaranteed values

The main objective of this section is to show that there exist broad classes of NP-optimization problems in which every problem has a nontrivial lower or upper bound on the optimum solution size and the above or below guarantee question with respect to these bounds is FPT. Recall that a lower or upper bound is *nontrivial* if it is an unbounded function of the input size.

We first show that every problem in the class MAX SNP has a nontrivial lower bound on the optimal solution size and that the parameterized above- or below-guarantee question with respect to this lower bound is FPT.

3.1. Parameterizing above the max 3-sat lower bound

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. Using this, we show the following generalization.

Proposition 1. *If Q is in MAX SNP, then for each instance I of Q there exists a positive number γ_x such that $\gamma_x \leqslant \mathrm{opt}(x)$. Further, if Q is NP-hard, then the function $\gamma : x \to \gamma_x$ is unbounded, assuming P $\neq$ NP.*

Proof. Let Q be a problem in MAX SNP and let (f, g, α, β) be an L-reduction from Q to MAX 3-SAT. Then for an instance I of Q, $f(x)$ is an instance of MAX 3-SAT such that $\mathrm{opt}(f(x)) \leqslant \alpha \cdot \mathrm{opt}(x)$. If $f(x)$ is a formula with m clauses, then $\lceil m/2 \rceil \leqslant \mathrm{opt}(f(x))$ and therefore $\mathrm{opt}(x)$ is bounded below by $\lceil m/2 \rceil/\alpha$. This proves that each instance x of Q 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 $\gamma_x = |f(x)|/2\alpha = m/2\alpha$. Further, note that if m is not unbounded, then we can solve Q in polynomial time via this reduction. $\square$

Note that this lower bound γ_x depends on the complete problem to which we reduce Q. 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 could 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 Q.

Consider the above-guarantee parameterized version L of MAX 3-SAT:

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

This problem is fixed-parameter tractable with respect to parameter k [28] and using this we prove the following.

Theorem 1. *Let Q be a maximization problem in MAX SNP and (f, g, α, β) an L-reduction from Q to MAX 3-SAT. For an instance x of Q, let γ_x represent the MAX 3-SAT-lower bound of x. Then the following problem is in FPT:*

$$L_Q = \{(x, k)\colon\ x \text{ is an instance of } Q \text{ and } \mathrm{opt}(x) \geqslant \gamma_x + k\}.$$

Proof. We make use of the fact that there exists a fixed-parameter tractable algorithm $\mathscr{A}$ for MAX 3-SAT which takes as input, a pair of the form (ψ, k), and in time $O(|\psi| + h(k))$, returns YES if there exists an assignment to the variables of ψ that satisfies at least $\lceil m/2 \rceil + k$ clauses, and NO otherwise. See [28,36] for such algorithms.

Consider an instance (x, k) of L_Q. 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 Q, $\gamma_x = m/2\alpha$, and $\mathrm{opt}(f(x)) \leqslant \alpha \cdot \mathrm{opt}(x)$. Apply algorithm $\mathscr{A}$ on input $(f(x), k\alpha)$. If $\mathscr{A}$ outputs YES, then $\mathrm{opt}(f(x)) \geqslant m/2 + k\alpha$, implying $\mathrm{opt}(x) \geqslant m/2\alpha + k = \gamma_x + k$. Thus $(x, k) \in L_Q$.

If $\mathscr{A}$ answers NO, then $\lceil m/2 \rceil \leqslant \mathrm{opt}(f(x)) < \lceil m/2 \rceil + k\alpha$. Apply algorithm $\mathscr{A}$ on the inputs $(f(x), 1), (f(x), 2), \ldots, (f(x), k\alpha)$, one by one, to obtain $\mathrm{opt}(f(x))$. Let $c' = \mathrm{opt}(f(x))$. 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 $|c - \mathrm{opt}(x)| \leqslant \beta \cdot |c' - \mathrm{opt}(f(x))|$. But since $c' = \mathrm{opt}(f(x))$, it must be that $c = \mathrm{opt}(x)$. Therefore we simply need to compare c with $\gamma_x + k$ to check whether $(x, k) \in L_Q$.

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 Q 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_Q. $\square$

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 bounded below 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, $\mathrm{opt}(x) \leqslant \gamma_x + k$, with k as parameter.

Theorem 2. *Let Q be a minimization problem in MAX SNP and (f, g, α, β) an L-reduction from Q to MAX 3-SAT. For an instance x of Q, let γ_x represent the MAX 3-SAT-lower bound of x. Then the following problem is in FPT:*

$$L_Q = \{(x, k)\colon\ x \text{ is an instance of } Q \text{ and } \mathrm{opt}(x) \leqslant \gamma_x + k\}.$$

Proof. As before, let $\mathscr{A}$ be an FPT-algorithm for MAX 3-SAT which takes as input, a pair of the form (ψ, k), and in time $O(|\psi| + h(k))$, returns YES if there exists an assignment to the variables of ψ that satisfies at least $\lceil m/2 \rceil + k$ clauses, and NO otherwise. Let (x, k) be an instance of L_Q and $f(x)$ the instance of MAX 3-SAT with m clauses. Then $\gamma_x = m/2\alpha$ and $\mathrm{opt}(f(x)) \leqslant \alpha \cdot \mathrm{opt}(x)$. Apply algorithm $\mathscr{A}$ on input $(f(x), (k + 1) \cdot \alpha)$. If $\mathscr{A}$ outputs YES, then $\mathrm{opt}(f(x)) \geqslant m/2 + (k + 1) \cdot \alpha$, implying $\mathrm{opt}(x) \geqslant m/2\alpha + k + 1 = \gamma_x + k + 1$. In this case, $(x,k) \notin L_Q$ and we return NO. If $\mathscr{A}$ answers NO, then $\lceil m/2 \rceil \leqslant \mathrm{opt}(f(x)) < \lceil m/2 \rceil + (k+1)\cdot\alpha$. Apply algorithm $\mathscr{A}$ $(k+1)\cdot\alpha - 1$ times on inputs $(f(x),1),(f(x),2),\ldots,(f(x),(k+1)\cdot\alpha-1)$ to obtain $\mathrm{opt}(f(x))$. Obtain $\mathrm{opt}(x)$ as described in the proof of Theorem 1 and decide the instance (x,k) appropriately. □

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

3.2. When the guarantee is defined by an approximation algorithm

If an NP-maximization problem admits an α-approximation algorithm, $0 < \alpha < 1$, then $\alpha \cdot \textsc{opt}$ is a nontrivial polynomial-time computable lower-bound on the solution size, where OPT denotes the optimal solution size. For minimization problems that admit an α-approximation algorithm we have $\alpha > 1$ and $\alpha \cdot \textsc{opt}$ is a polynomial-time computable upper-bound on the solution size. One can parameterize above or below these nontrivial guaranteed values. We show that if (1) the NP-optimization problem is polynomially bounded, (2) its standard parameterized version is in FPT, and (3) α is a constant, then the $\alpha \cdot \textsc{opt} + k$ question (for maximization problems) and the $\alpha \cdot \textsc{opt} - k$ question (for minimization problems) are in FPT.

Theorem 3. *Let Q be an* NPO *problem which admits a constant-factor α-approximation algorithm such that its standard parameterized version is in* FPT. *Then the following problems are in* FPT:

$$L_1 = \{(I,k):\ \max(I) \geqslant \alpha \cdot \textsc{opt} + k\} \quad \textit{if } Q \textit{ is a maximization problem};$$

$$L_2 = \{(I,k):\ \min(I) \leqslant \alpha \cdot \textsc{opt} - k\} \quad \textit{if } Q \textit{ is a minimization problem}.$$

Proof. Suppose Q is a maximization problem and let (I,k) be an instance of L_1. Then $\max(I) \geqslant \alpha \cdot \textsc{opt} + k$ if and only if $k \leqslant (1-\alpha)\cdot \textsc{opt}$, that is, if and only if $k/(1-\alpha) \leqslant \textsc{opt}$. Since the standard parameterized version of Q is in FPT, deciding whether $k/(1-\alpha) \leqslant \textsc{opt}$ is in FPT. The proof when Q is a minimization problem is similar. □

4. Tight lower and upper bounds

For an optimization problem, the question of whether the optimum is at least lower bound $+\,k$, for some lower bound and k as parameter, is not always interesting because if the lower bound is "loose" then the problem is trivially fixed-parameter tractable. For instance, the following "above-guarantee" version of MAX CUT is trivially in FPT. Given a connected graph G on m edges, does G have a cut of size at least $m/2 + k$? Since any connected graph G with m edges and n vertices has a cut of size at least $m/2 + \lceil (n-1)/4 \rceil$ [38], if $k \leqslant \lceil (n-1)/4 \rceil$, we answer YES. Otherwise $n \leqslant 4k$ and a brute-force algorithm that considers all possible vertex partitions is an FPT-algorithm.

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 ϕ is an instance of MAX SAT, then $\mathrm{opt}(\phi) \geqslant m/2$, 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 $Q = \{\mathscr{I}, S, V, \mathrm{opt}\}$ be an NP-optimization problem and let $f : \mathbb{N} \to \mathbb{N}$. We say that f is a ***tight lower bound*** for Q if the following conditions hold:

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

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 $Q = \{\mathscr{I}, S, V, \mathrm{opt}\}$ be an NP-optimization problem and let $g : \mathbb{N} \to \mathbb{N}$. We say that g is a ***tight upper bound*** for Q if the following conditions hold:

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

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 $= (1 - 1/2^c)m$; TUB $= m$.
The expected number of clauses satisfied by the random assignment algorithm is $(1 - 1/2^c)m$; hence the lower bound. To see tightness, note that if $\phi(x_1, \ldots, x_c)$ denotes the Exact c-Sat formula comprising of all possible combinations of c variables, then ϕ 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 $(1 - 1/2^c)m$ satisfiable clauses.

2. Max Lin-2
INSTANCE A system of m linear equations modulo 2 in n variables, together with positive weights w_i, $1 \leqslant i \leqslant m$.
QUESTION Find an assignment to the variables that maximizes the total weight of the satisfied equations.
BOUNDS TLB $= W/2$, where $W = \sum_{i=1}^{m} w_i$; TUB $= W$.
If we use $\{+1, -1\}$-notation for boolean values with -1 corresponding to true then we can write the ith equation of the system as $\prod_{j \in \alpha_i} x_j = b_i$, where each α_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 [22] 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 $= n/(B + 1)$; 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 pigeonhole 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 $n/(B + 1)$. By taking disjoint copies of K_{B+1} one can construct instances of arbitrary size with independence number exactly $n/(B + 1)$.

4. Min Dominating Set-B
INSTANCE A graph G with n vertices such that the degree of each vertex is bounded by B.
QUESTION Find a minimum dominating set of G.
BOUNDS TLB $= n/(B + 1)$; TUB $= n$.
Observe that any vertex can dominate at most $B + 1$ vertices (including itself) and thus any dominating set has size at least $n/(B + 1)$. A set of r disjoint copies of K_{B+1} has a minimum dominating set of size exactly $r = n/(B + 1)$. The upper bound of n is met by the class of empty graphs.

5. Min Vertex Cover-B
INSTANCE A graph G with n vertices and m edges such that the degree of each vertex is at least one and at most B.
QUESTION Find a minimum vertex cover of G.
BOUNDS TLB $= m/B$; TUB $= nB/(B + 1)$.
Each vertex can cover at most B edges and so any vertex cover has size at least m/B. This bound is tight for a set of disjoint copies of $K_{1,B}$'s. The upper bound follows from the fact that an independent set in such a graph has size at least $n/(B + 1)$. The upper bound is met by a disjoint collection of K_{B+1}'s.

6. 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 $= n/4$; 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 pigeonhole principle, there exists an independent set of size at least $n/4$. A disjoint set of K_4's can be use to construct arbitrary sized instances with independence number exactly $n/4$.

7. Max Acyclic Digraph
INSTANCE A directed graph G with n vertices and m arcs.
QUESTION Find a maximum arc-induced acyclic subgraph of G.
BOUNDS TLB $= m/2$; TUB $= m$.
To see that any digraph with m arcs has an acyclic subgraph of size $m/2$, place the vertices $v_1, \ldots, v_n$ 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 $m/2$. To see that this bound is tight, consider the digraph D on n vertices: $v_1 \leftrightarrows v_2 \leftrightarrows v_3 \leftrightarrows \cdots \leftrightarrows v_n$ which has a maximum acyclic digraph of size exactly $m/2$. Since n is arbitrary, we have an infinite set of instances for which the optimum matches the lower bound exactly.

8. 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 achieves this lower bound. An upper bound is $3n-6$ which is tight since for each n, a maximal planar graph on n vertices has exactly $3n-6$ edges.

9. Max Cut
INSTANCE A graph G with n vertices, m edges and c components.
QUESTION Find a maximum cut of G.
BOUNDS TLB $= m/2 + \lceil (n-c)/4 \rceil$; TUB $= m$.
The lower bound for the cut size was proved by Poljak and Turzík [38]. This bound is tight for complete graphs. The upper bound is tight for bipartite graphs.

10. Min Linear Arrangement
INSTANCE An undirected graph $G = (V, E)$ with n vertices and m edges.
QUESTION Find a one-to-one mapping $\sigma : V \to \{1, 2, \ldots, |V|\}$ such that

$$\sum_{\{u,v\}\in E} \left|\sigma(u) - \sigma(v)\right|$$

is a minimum. The minimum value of the objective function over all one-to-one maps is denoted by ola(G).
BOUNDS TLB $= m$; TUB $= n(n-1)(n+1)/6 = \binom{n+1}{3}$.
Clearly $|\sigma(u) - \sigma(v)| \geqslant 1$ for any mapping σ and hence $\sum_{\{u,v\}\in E} |\sigma(u) - \sigma(v)| \geqslant m$. This lower bound is tight for the set of paths. For any n-vertex graph G, ola$(G) \leqslant$ ola(K_n), where K_n denotes the complete graph on n vertices. It can be easily seen that for the case of K_n, the value of the objective function remains the same for all one-to-one maps σ and that ola$(K_n) = 1 \cdot (n-1) + 2 \cdot (n-2) + \cdots + (n-1) \cdot 1 = n(n-1)(n+1)/6$.

11. Min Profile
INSTANCE An undirected graph $G = (V, E)$ with n vertices and m edges.
QUESTION Find a one-to-one mapping $\sigma : V \to \{1, 2, \ldots, |V|\}$ such that $\sum_{v\in V} \mathrm{prf}_\sigma(v)$ is a minimum, where $\mathrm{prf}_\sigma(v) = \sigma(v) - \min\{\sigma(u)\colon u \in N[v]\}$.
The minimum value of the objective function over all one-to-one maps σ is denoted by prf(G).
BOUNDS TLB $= m$; TUB $= \binom{n}{2}$.
This problem is equivalent [3] to the well-known Interval Graph Completion problem [17,23] in that if the minimum number of edges required to be added to a graph to make it interval is k then its profile is $m+k$, where m denotes the number of edges in the graph. Consequently, for any graph $G = (V, E)$,

$$m \leqslant \mathrm{prf}(G) \leqslant \binom{n}{2}.$$

Since for each positive integer m there exists an interval graph with m edges, this is a tight lower bound on the profile of a graph. Since complete graphs are interval graphs too, this result also shows that $\binom{n}{2}$ is a tight upper bound. Interestingly, there is a different lower bound if we restrict the input graph to be connected. If G is a connected graph on n vertices then prf$(G) \geqslant n-1$ [26] and this bound is tight since the profile of a path on n vertices is $n-1$.

Natural questions for maximization problems in the above or below-guarantee framework are whether the languages

$$L_{a,\max} = \{(I,k)\colon \max(I) \geqslant \mathrm{TLB}(I) + k\},$$
$$L_{b,\max} = \{(I,k)\colon \max(I) \geqslant \mathrm{TUB}(I) - k\}$$

are in FPT. For minimization problems, one can ask whether the following are in FPT.

$$L_{a,\min} = \{(I,k)\colon \min(I) \leqslant \mathrm{TLB}(I) + k\},$$
$$L_{b,\min} = \{(I,k)\colon \min(I) \leqslant \mathrm{TUB}(I) - k\}.$$

The parameterized complexity of such questions are not known for most problems which are known to have tight bounds. To the best of our knowledge, the above-guarantee question has been shown to be in FPT only for the Max Sat and Max c-Sat problems [28] and, more recently, for Linear Arrangement [20] and both versions of Minimum Profile [21,23].

Since most vertex/edge deletion problems can be cast as below-guarantee parameterized questions, comparatively more results are known about such problems. For instance, for the Independent Set problem a trivial upper bound on the solution size is the number of vertices in the graph. This bound is tight as the family of trivial graphs meets this bound. Given a graph G on n vertices, the question of whether there exists k vertices whose deletion leaves a trivial graph (or equivalently, does G have an independent set on $n-k$ vertices?) is fixed-parameter tractable, being equivalent to the well-known Vertex Cover problem [35]. Other examples include Feedback Vertex Set [39], Directed Feedback Vertex Set [7], Odd Cycle Transversal [41] and Chordal Vertex Deletion [30] which are all of the form: "is there an acyclic (undirected/directed), bipartite and chordal graph, respectively, on $n-k$ vertices?"

An interesting example of a non-graph-theoretic problem parameterized below a guaranteed upper bound is the Min 2-Sat Deletion problem [28].

Min 2-Sat Deletion
Input: A Boolean 2-CNF formula ϕ with m clauses.
Parameter: A positive integer k.
Question: Does there exist an assignment that satisfies $m-k$ clauses of ϕ? In other words, can k clauses be deleted from ϕ to make it satisfiable?

This problem has been shown to be fixed-parameter tractable by Razgon et al. [40]. Note that the problem of deciding whether k clauses can be deleted from a c-CNF formula to make it satisfiable is NP-hard for $c \geqslant 3$ [28].

In the next section we exhibit problems whose above or below-guarantee parameterized versions are hard.

5. Hard above or below-guarantee problems

We first exhibit two problems whose above-guarantee parameterized versions are hard (not in FPT unless $\mathsf{P}=\mathsf{NP}$). To the best of our knowledge, these are the only ones in this category. Consider the problem Min Weight t-connected Spanning Subgraph defined as follows [8]:

Input: A connected graph G with n vertices and nonnegative integers t and k.
Parameter: The integer k.
Question: Does there exists a t-vertex-connected spanning subgraph of G with at most k edges?

This problem is NP-complete for undirected graphs for $t \geqslant 2$ [8] and is easily fixed-parameter tractable as shown below.

Lemma 1. *Let $G=(V,E)$ be a simple, connected graph on n vertices and let k, t be nonnegative integers. Then deciding whether G has a t-vertex-connected spanning subgraph with at most k edges can be done in time polynomial in n, for every fixed k.*

Proof. For $t \geqslant 2$, a t-vertex-connected spanning subgraph of G must have at least n edges. Therefore if $k<n$, answer no; else, $n \leqslant k$ and any brute-force algorithm that solves the problem is fixed-parameter tractable. □

As discussed in the proof of Lemma 1 above, n is a trivial lower bound for the problem and is tight for the case $t=2$. The above-guarantee version, however, is not fixed-parameter tractable, unless $\mathsf{P}=\mathsf{NP}$.

Theorem 4. *Let $G=(V,E)$ be a simple, connected graph on n vertices and let k,t be nonnegative integers. Deciding whether G has a t-vertex connected spanning subgraph with at most $n+k$ edges is not fixed-parameter tractable with respect to parameter k, unless* $\mathsf{P}=\mathsf{NP}$.

Proof. Note that for $t=2$ and $k=0$ this problem is equivalent to asking whether G has a Hamiltonian cycle and hence if the above-guarantee question can be answered in time $O(f(k)\cdot n^c)$, the Hamiltonian Cycle problem can be solved in polynomial time implying $\mathsf{P}=\mathsf{NP}$. □

Next consider the Bounded Degree Min Spanning Tree [18] problem.

Input: A connected graph $G=(V,E)$ with edge costs $w: E \to \mathbb{Z}^+$ and nonnegative integers c and k.
Parameter: The integer k.
Question: Does there exist a spanning subgraph T of total edge-weight at most k such that each vertex in T has degree at most c?

Since the total weight of any spanning tree is at least $n-1$, we have the following result.

Lemma 2. *The* Bounded Degree Min Spanning Tree *problem is fixed-parameter tractable with respect to parameter k.*

The above-guarantee version (Does G have a spanning tree with total weight at most $n-1+k$ and vertex degrees bounded by c?) is again not fixed-parameter tractable unless $\mathsf{P}=\mathsf{NP}$ as the case $c=2$ and $k=0$ reduces to solving the Hamiltonian Path problem.

Theorem 5. *Given a connected graph $G=(V,E)$ with edge costs $w: E \to \mathbb{Z}^+$ and positive integers c and k, deciding whether G has a spanning tree with weight at most $n-1+k$ and vertex degrees bounded by c is not fixed-parameter tractable with respect to k unless* $\mathsf{P}=\mathsf{NP}$.

We next consider a problem parameterized below a tight upper bound, called Wheel-free Deletion, that is W[2]-hard [27]. A vertex v in a graph G is said to be *universal* if v is adjacent to all vertices of G. A *wheel* is a graph W that has a universal vertex v such that $W-v$ is a cycle. A graph is wheel-free if no subgraph of G is a wheel. The class of wheel-free graphs is hereditary [25] and poly-time recognizable [27]. The Wheel-free Vertex/Edge Deletion problem is defined below:

Input: A graph $G=(V,E)$ with n vertices and m edges and a nonnegative integer k.
Parameter: The integer k.
Question: Does G have an induced subgraph on $n-k$ vertices/$m-k$ edges that is wheel-free?

Note that for all n and m there exist wheel-free graphs on n vertices or m edges. The tight upper bound is witnessed by the class of paths, for instance.

In [27], the Wheel-free Vertex/Edge Deletion problems were shown to be W[2]-hard by a reduction from Hitting Set. This is one of the few graph-modification problems known to be hard.

6. Parameterizing sufficiently above or below guaranteed values

In this section, we study somewhat different, but related, parameterized questions: Given an NP-maximization problem Q with a tight lower and a tight upper bound, denoted by TLB and TUB, respectively, what is the parameterized complexity of the following questions?

$$Q_{a,\max}(\epsilon)=\{(I,k):\ \max(I) \geqslant \textsc{tlb}(I)+\epsilon\cdot|I|+k\}, \tag{1}$$

$$Q_{b,\max}(\epsilon)=\{(I,k):\ \max(I) \geqslant \textsc{tub}(I)-\epsilon\cdot|I|-k\}. \tag{2}$$

Here $|I|$ denotes the input size, ϵ is some fixed positive rational, k is the parameter and a and b denote, respectively, the above and below-guarantee version of the problem. For NP-minimization problems, the corresponding questions are:

$$Q_{a,\min}(\epsilon)=\{(I,k):\ \min(I) \leqslant \textsc{tlb}(I)+\epsilon\cdot|I|+k\}, \tag{3}$$

$$Q_{b,\min}(\epsilon)=\{(I,k):\ \min(I) \leqslant \textsc{tub}(I)-\epsilon\cdot|I|-k\}. \tag{4}$$

In Theorem 6, we show that Problems 1 and 3 are not fixed-parameter tractable for a certain class of problems, unless $\mathsf{P}=\mathsf{NP}$. Theorem 7 establishes this result for Problems 2 and 4. To define the class of optimization problems for which we establish the hardness result in Theorem 6, we need some definitions. To motivate these, we start with an overview of the proof for maximization problems (Problem 1 above).

Assume that for some ϵ in the specified range, $Q_{a,\max}(\epsilon)$ is indeed in FPT. Now consider an instance (I,s) of the underlying decision version of Q. Here is a P-time procedure for deciding it. If $s \leqslant \textsc{tlb}$, then the answer is trivially YES. If s lies between TLB and $\textsc{tlb}+\epsilon|I|$, then “add” a gadget of suitable size corresponding to the TUB, to obtain an equivalent instance (I',s'). 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 $\textsc{tlb}+\epsilon|I'|$ and we handle this case next. If s already exceeds $\textsc{tlb}+\epsilon|I|$, then “add” a gadget of suitable size corresponding to the TLB, to obtain an equivalent instance (I',s'). This increases the input size faster than it boosts the optimum value of I', so that now s' exceeds $\textsc{tlb}+\epsilon|I'|$ by only a constant, say c_1. Use the hypothesized FPT algorithm for $Q_{a,\max}(\epsilon)$ with input (I',c_1) to correctly decide the original question.

To make this proof idea work, we require that the following conditions be met:

1. The NPO problem should be such that “addition” of problem instances is well defined and that the optimum of the sum is equal to the sum of the optima (see Definition 3).
2. There exist gadgets whose addition to a problem instance increases the instance size faster than it does the optimum value (see Property P1 below).
3. There exist gadgets whose addition to a problem instance increases the optimum value faster than it does the instance size (see Property P2 below).
4. The gadgets mentioned in points 2 and 3 must be easily constructible (see Definition 4).

Definition 3 *(Partially additive problems).* An NPO problem $Q = \{\mathscr{I}, S, V, \text{opt}\}$ 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. [24]: 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 $V(I_1 + I_2, s) = V(I_1, s_1) + V(I_1, s_2)$.

For many 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 some (possibly arbitrarily chosen) vertex of G and some (possibly arbitrarily chosen) vertex of H. The Max Planar Subgraph problem is partially additive with respect to both these interpretations of $+$. For boolean formulae ϕ and ψ 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.

Definition 4 *(Dense set).* Let $Q = \{\mathscr{I}, S, V, \text{opt}\}$ be an NPO problem. A set of instances $\mathscr{I}' \subseteq \mathscr{I}$ 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 $|b - a| \geqslant c$, there exists an instance $I \in \mathscr{I}'$ 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 $\mathscr{I}'$ is said to be ***dense poly-time uniform with respect to*** $\mathcal{C}$.

For example, for the Maximum Acyclic Digraph problem, the set of all oriented digraphs (digraphs without 2-cycles) is dense (poly-time uniform) with respect to the condition: $\text{opt}(G) = |E(G)|$.

Let $Q = \{\mathscr{I}, S, V, \text{opt}\}$ be an NP-optimization problem with a tight lower bound $f : \mathbb{N} \to \mathbb{N}$ and a tight upper bound $g : \mathbb{N} \to \mathbb{N}$. We assume that both f and g are increasing and satisfy the following conditions

- P1 For all $a, b \in \mathbb{N}$, $f(a + b) \leqslant f(a) + f(b) + c^*$, where c^* is a constant (positive or negative).
- P2 There exists $n_0 \in \mathbb{N}$ and $r \in \mathbb{Q}^+$ such that $g(n) - f(n) > rn$ for all $n \geqslant n_0$.

Property P1 is satisfied by linear functions ($f(n) = an + b$) and by some sub-linear functions such as $\sqrt{n}$, $\log n$, $1/n$. Note that a super-linear function cannot satisfy P1.

Now that we have formally defined all the required properties, we can state the theorem precisely.

Theorem 6. *Let $Q = \{\mathscr{I}, S, V, \text{opt}\}$ be a polynomially bounded* NP*-optimization problem such that the following conditions hold.*

1. *Q is partially additive.*
2. *Q has a tight lower bound* (TLB) *f, which is increasing and satisfies condition* P1. *The infinite family of instances $\mathscr{I}'$ witnessing the tight lower bound is dense poly-time uniform with respect to the condition* $\text{opt}(I) = f(|I|)$.
3. *Q has a tight upper bound* (TUB) *g, which with f satisfies condition* P2. *The infinite family of instances $\mathscr{I}'$ 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.*

Let $p := \sup\{r \in \mathbb{Q}^+ \colon g(n) - f(n) > rn$ for all $n \geqslant n_0\}$ and for $0 < \epsilon < p$, define $Q_a(\epsilon)$ to be the following parameterized problem

$$Q_a(\epsilon) = \{(I, k)\colon \text{opt}(I) \geqslant f(|I|) + \epsilon|I| + k\} \quad \textit{for maximization problems};$$

$$Q_a(\epsilon) = \{(I, k)\colon \text{opt}(I) \leqslant f(|I|) + \epsilon|I| + k\} \quad \textit{for minimization problems}.$$

If $Q_a(\epsilon)$ is FPT for any $0 < \epsilon < p$, then $\mathrm{P} = \mathrm{NP}$.

Proof. We present a proof for NP-maximization problems and towards the end we outline the necessary changes needed for this proof to work for minimization problems. Therefore let Q be an NP-maximization problem and suppose that for some $0 < \epsilon < p$, the parameterized problem $Q_a(\epsilon)$ is fixed-parameter tractable. Let $\mathscr{A}$ be an FPT-algorithm for it with run time $O(t(k)\text{poly}(|I|))$. We will use $\mathscr{A}$ to solve the underlying decision problem of Q in polynomial time. Note that for $0 < \epsilon < p$, $g(n) - f(n) - \epsilon n$ is strictly increasing and strictly positive for large enough values of n.

Let (I, s) be an instance of the decision version of Q. Then (I, s) is a YES-instance if and only if $\max(I) \geqslant s$. We consider three cases and proceed as described below.

Case 1. $s < f(|I|)$.

Since $\max(I) \geqslant f(|I|)$, we answer YES.

Case 2. $f(|I|) \leqslant s < f(|I|) + \epsilon|I|$.

In this case, we claim that we can transform the input instance (I, s) into an 'equivalent' instance (I', s') such that

1. $f(|I'|) + \epsilon|I'| \leqslant s'$.
2. $|I'| = \mathrm{poly}(|I|)$.
3. $\mathrm{opt}(I) \geqslant s$ if and only if $\mathrm{opt}(I') \geqslant s'$.

This will show that we can, without loss of generality, go to Case 3 below directly.

To achieve the transformation, add a TUB instance I_1 to I. Define $I' = I + I_1$ and $s' = s + g(|I_1|)$. Then it is easy to see that $\max(I) \geqslant s$ if and only if $\max(I') \geqslant s'$. We want to choose I_1 such that $f(|I'|) + \epsilon|I'| \leqslant s'$. Since $|I'| = |I| + |I_1|$ and $s' = s + g(I_1)$, and since $f(|I|) < s$, it suffices to choose I_1 satisfying

$$f\big(|I| + |I_1|\big) + \epsilon|I| + \epsilon|I_1| \leqslant f\big(|I|\big) + g\big(|I_1|\big).$$

By Property P1, we have $f(|I| + |I_1|) \leqslant f(|I|) + f(|I_1|) + c^*$, so it suffices to satisfy

$$f\big(|I_1|\big) + c^* + \epsilon|I| + \epsilon|I_1| \leqslant g\big(|I_1|\big).$$

By Property P2 we have $g(|I_1|) > f(|I_1|) + p|I_1|$, so it suffices to satisfy

$$c^* + \epsilon|I| \leqslant (p - \epsilon)|I_1|.$$

Such an instance I_1 (of size polynomial in $|I|$) can be chosen because $0 < \epsilon < p$, and because the tight upper bound is polynomial-time uniform dense.

Case 3. $f(|I|) + \epsilon|I| \leqslant s$.

In this case, we transform the instance (I, s) into an instance (I', s') such that

1. $f(|I'|) + \epsilon|I'| + c_1 = s'$, where $0 \leqslant c_1 \leqslant c_0$ and c_0 is a fixed constant.
2. $|I'| = \mathrm{poly}(|I|)$.
3. $\max(I') \geqslant s'$ if and only if $\max(I) \geqslant s$.

We then run algorithm $\mathscr{A}$ with input (I', c_1). Algorithm $\mathscr{A}$ answers YES if and only if $\max(I') \geqslant s'$. By condition 3 above, this happens if and only if $\max(I) \geqslant s$. This takes time $O(t(c_1) \cdot \mathrm{poly}(|I'|))$.

We obtain I' by adding a TLB instance I_1 to I. What if addition of any TLB instance yields an I' with $s' < f(I') + \epsilon|I'|$? In this case, s must already be very close to $f(|I|) + \epsilon|I|$; the difference $k \triangleq s - f(|I|) - \epsilon|I|$ must be at most $\epsilon d + c^*$, where d is the size of the smallest TLB instance I_0. (Why? Add I_0 to I to get $s + f(d) < f(|I| + d) + \epsilon(|I| + d)$; applying property P1, we get $s + f(d) < f(|I|) + f(d) + c^* + \epsilon|I| + \epsilon d$, and so $k < c^* + \epsilon d$.) In such a case, we can use the FPT algorithm $\mathscr{A}$ with input (I, k) directly to answer the question "Is $\max(I) \geqslant s$?" in time $O(t(\epsilon d + c^*) \cdot \mathrm{poly}(|I|))$.

So now assume that $k \geqslant c^* + \epsilon d$, and it is possible to add TLB instances to $|I|$. Since f is an increasing function, there is a *largest* TLB instance I_1 we can add to I to get I' while still satisfying $s' \geqslant f(I') + \epsilon|I'|$. The smallest TLB instance bigger than I_1 has size at most $|I_1| + c$, where c is the constant that appears in the definition of density. We therefore have the following inequalities

$$f\big(|I'|\big) + \epsilon|I'| \leqslant s' < f\big(|I'| + c\big) + \epsilon\big(|I'| + c\big).$$

Since f is increasing and satisfies property P1, we have

$$\big[f\big(|I'| + c\big) + \epsilon\big(|I'| + c\big)\big] - \big[f\big(|I'|\big) + \epsilon|I'|\big] \leqslant f(c) + c^* + \epsilon c \triangleq c_0,$$

and hence $s' = f(|I'|) + \epsilon|I'| + c_1$, where $0 \leqslant c_1 \leqslant c_0$. Note that c_0 is a constant independent of the input instance (I, s). Also, since Q is a polynomially bounded problem, $|I_1|$ is polynomially bounded in $|I|$.

Note that the proof for Cases 2 and 3 do not make explicit use of the fact that Q is a maximization problem; the proof here goes through for minimization problems as well. In fact, the only change necessary for minimization problems is in Case 1 where if $s <$ TLB, we return NO. □

Remark 1. Note that there are some problems, notably MAX 3-SAT, for which the constant c_0 in Case 3 of the proof above, is 0. For such problems, the proof of Theorem 6 actually proves that the problem $Q' = \{(I, k)\colon\ \max(I) \geqslant f(|I|) + \epsilon|I|\}$ is NP-hard. But in general, the constant $c_0 \geqslant 1$ and so this observation cannot be generalized.

The constraints imposed in Theorem 6 seem to be rather strict, but they are satisfied by a large number of NP-optimization problems.

Corollary 1. *For any* NP*-optimization problem Q in the following list, the $Q_a(\epsilon)$ problem is not fixed-parameter tractable unless* P = NP:

	Problem	$\textsc{tlb}(I) + \epsilon \cdot \lvert I \rvert + k$	*Range of* ϵ
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.	Max Lin-2	$(\frac{1}{2} + \epsilon)m + k$	$0 < \epsilon < \frac{1}{2}$
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.	Dominating Set-B	$(\frac{1}{B+1} + \epsilon)n + k$	$0 < \epsilon < \frac{B}{B+1}$
8.	Vertex Cover-B	$\frac{m}{B} + \epsilon n + k$	$0 < \epsilon < \frac{(B-1)(2B+1)}{2B(B+1)}$
9.	Max Acyclic Subgraph	$(\frac{1}{2} + \epsilon)m + k$	$0 < \epsilon < \frac{1}{2}$
10.	Max Planar Subgraph	$(1 + \epsilon)n - 1 + k$	$0 < \epsilon < 2$
11.	Max Cut	$\frac{m}{2} + \lceil \frac{n-c}{4} \rceil + \epsilon n + k$	$0 < \epsilon < \frac{1}{4}$
12.	Max Dicut	$\frac{m}{4} + \sqrt{\frac{m}{32} + \frac{1}{256}} - \frac{1}{16} + \epsilon m + k$	$0 < \epsilon < \frac{3}{4}$

Remark 2. Note that the results for Vertex Cover-B and Max Cut do not follow directly from Theorem 6 (as the bounds involve both n, the number of vertices, and m, the number of edges) but they can proved independently using the same proof-technique.

We now extend Theorem 6 to the corresponding variant of the below-guarantee question. Let $Q = \{\mathscr{I}, S, V, \mathrm{opt}\}$ be an NP-optimization problem with a tight lower bound $f : \mathbb{N} \to \mathbb{N}$ and a tight upper bound $g : \mathbb{N} \to \mathbb{N}$ which are increasing functions and satisfy the following conditions

P3 For all $a, b \in \mathbb{N}$, $g(a + b) \leqslant g(a) + g(b) + c^*$, where c^* is a constant.
P4 There exists $r \in \mathbb{Q}^+$ such that $g(n) - f(n) > rn$ for all $n \geqslant n_0$ for some $n_0 \in \mathbb{N}$.

Theorem 7. *Let $Q = \{\mathscr{I}, S, V, \mathrm{opt}\}$ be a polynomially bounded* NP*-optimization problem such that the following conditions hold.*

1. *Q is partially additive.*
2. *Q has a tight lower bound* (TLB) *f such that the infinite family of instances $\mathscr{I}'$ witnessing the tight lower bound is dense poly-time uniform with respect to the condition* $\mathrm{opt}(I) = f(|I|)$.
3. *Q has a tight upper bound* (TUB) *g which is increasing, satisfies condition* P3*, and with f satisfies* P4*. The infinite family of instances $\mathscr{I}'$ witnessing the tight upper bound is dense poly-time uniform with respect to the condition* $\mathrm{opt}(I) = g(|I|)$.
4. *The underlying decision problem $\tilde{Q}$ of Q is* NP*-hard.*

Let $p := \sup\{r \in \mathbb{Q}^+ :\ g(n) - f(n) > rn$ for all $n \geqslant n_0\}$ and for $0 < \epsilon < p$, define $Q_b(\epsilon)$ to be the following parameterized problem

$$Q_b(\epsilon) = \{(I, k):\ \max(I) \geqslant \textsc{tub}(I) - \epsilon \cdot |I| - k\} \quad (\textit{maximization problems});$$

$$Q_b(\epsilon) = \{(I, k):\ \min(I) \leqslant \textsc{tub}(I) - \epsilon \cdot |I| - k\} \quad (\textit{minimization problems}).$$

If $Q_b(\epsilon)$ is FPT for any $0 < \epsilon < p$, then P = NP.

Proof Sketch. We sketch a proof for NP-minimization problems. Assume that for some ϵ in the specified range, $Q_b(\epsilon)$ is indeed in FPT. Consider an instance (I, s) of the underlying decision version of Q. Here is a P-time procedure for deciding it. If $s > \textsc{tlb}$, then the answer is trivially YES. If s lies between TUB and $\textsc{tub} - \epsilon|I|$, then "add" a gadget of suitable size corresponding to the TLB to obtain an equivalent instance (I', s'). This increases the input size, but since we are adding a gadget whose optimum value matches the lower bound, the increase in the optimum value of I' is less than proportional, so that now s' is less than $\textsc{tub} - \epsilon|I'|$. If s were already less than $\textsc{tub} - \epsilon|I|$, then "add" a gadget of suitable size corresponding to the TUB to obtain an equivalent instance (I', s'). This increases the optimum value faster than it does the instance size, so that now s' is less than $\textsc{tub} - \epsilon|I'|$ by only a constant, say c_1. Use the hypothesized FPT algorithm for $Q_b(\epsilon)$ with input (I', c_1) to correctly decide the original question. $\square$

The next result shows that for a number of NP-optimization problems, the below-guarantee parameterized variant is unlikely to be in FPT.

Corollary 2. *For any* NP*-optimization problem* Q *in the following list, the* $Q_b(\epsilon)$ *problem is not fixed-parameter tractable unless* $\mathsf{P} = \mathsf{NP}$:

	Problem	$\text{TLB}(I) + \epsilon \cdot \lvert I \rvert + k$	*Range of* ϵ
1.	Max Sat	$(1-\epsilon)m - k$	$0 < \epsilon < \frac{1}{2}$
2.	Max c-Sat	$(1-\epsilon)m - k$	$0 < \epsilon < \frac{1}{2}$
3.	Max Exact c-Sat	$(1-\epsilon)m - k$	$0 < \epsilon < \frac{1}{2^c}$
4.	Max Lin-2	$(1-\epsilon)m - k$	$0 < \epsilon < \frac{1}{2}$
5.	Planar Independent Set	$(1-\epsilon)n - k$	$0 < \epsilon < \frac{3}{4}$
6.	Independent Set-B	$(1-\epsilon)n - k$	$0 < \epsilon < \frac{B}{B+1}$
7.	Dominating Set-B	$(1-\epsilon)n - k$	$0 < \epsilon < \frac{B}{B+1}$
8.	Vertex Cover-B	$(\frac{B}{B+1} - \epsilon)n - k$	$0 < \epsilon < \frac{(B-1)(2B+1)}{2B(B+1)}$
9.	Max Acyclic Subgraph	$(1-\epsilon)m - k$	$0 < \epsilon < \frac{1}{2}$
10.	Max Planar Subgraph	$(3-\epsilon)n - 6 - k$	$0 < \epsilon < 2$
11.	Max Cut	$m - \epsilon n - k$	$0 < \epsilon < \frac{1}{4}$
12.	Max Dicut	$(1-\epsilon)m - k$	$0 < \epsilon < \frac{3}{4}$

7. Conclusion and further research

We have argued that for several optimization problems including all those in MAX SNP, the above or below-guarantee parameterization is the natural and more practical direction to pursue. In Section 5 we exhibited two problems for which the above-guarantee parameterization is hard and there are problems such as Max Sat [28], Min Linear Arrangement [20] and Min Profile [21,23] for which this question is fixed-parameter tractable. The main problem left open is:

Open Problem 1. Is there a characterization for the class of problems for which the above or below-guarantee question with respect to a tight lower or upper bound is in FPT (or W[1]-hard)?

We believe that there are several natural directions to pursue both from an algorithmic as well as from a practical point of view. As stated before, not many results are known on parameterized above or below-guarantee problems. In fact, the complexity of problems (1) through (9) stated in Section 4, when parameterized above their guaranteed values, is open. Some of the more interesting above-guarantee problems are:

Open Problem 2. Planar Independent Set: Given an n-vertex planar graph and an integer parameter k, does G have an independent set of size at least $\lceil n/4 \rceil + k$?

Open Problem 3. Max Exact c-Sat: Given a Boolean CNF formula F with m clauses such that each clause has exactly c distinct literals and an integer parameter k, does there exist an assignment that satisfies at least $(1 - 2^{-c})m + k$ clauses.

Here are some interesting below-guarantee problems which are open:

Open Problem 4. (See [32].) König Edge Deletion Set: Given a graph G on n vertices, m edges and an integer k, does there exist an edge-induced subgraph of G on $m - k$ edges that is König (a graph in which a minimum vertex cover and a maximum matching have the same size)? In other words, can k edges be deleted from G to make it König?

Open Problem 5. (See [9,34].) Perfect Vertex Deletion: Given a graph G on n vertices and m edges and an integer k, does there exist a vertex-induced subgraph on $n-k$ vertices that is perfect? A similar question can be framed for the edge version.

7.1. Deciding whether the optimum equals the guarantee

If an above-guarantee problem is FPT, we can test in polynomial time whether the optimum solution size for a given instance I equals the lower bound TLB(I) by simply running the algorithm on the input with $k = 1$. In fact, one way to prove that an above-guarantee problem *does not* have an FPT-algorithm is by showing that there is no polynomial time algorithm (assuming $\mathsf{P} \neq \mathsf{NP}$) that decides whether the optimum equals the guaranteed lower bound. The question of whether there exists such a polynomial time algorithm is open for problems (1) through (7) stated in Section 4. For many of these problems, such as Planar Independent Set and Max Cut, this may be an interesting independent problem in extremal graph theory.

Open Problem 6. Is there a polynomial time algorithm that decides whether

1. A given planar graph G on n vertices has a maximum independent set of size exactly $\lceil n/4 \rceil$?
2. A given connected graph G has a maximum cut of size exactly $\lceil m/2 \rceil + \lceil (n-1)/2 \rceil / 2$?

7.2. When the guarantee is a structural parameter

In this subsection we consider problems where the lower bound is a function of the problem instance rather than the input size.

7.2.1. Above-guarantee vertex cover

Consider the Vertex Cover problem. Given a graph $G = (V, E)$, a vertex cover of G is a subset $V' \subseteq V$, such that every edge of G has at least one endpoint in V'. The Vertex Cover problem is the problem of deciding whether, given a graph G and a positive integer k, G has a vertex cover of size at most k. This problem, when parameterized by k, is known to be in FPT by a number of FPT-algorithms (see [35]). Note that if M is a maximum matching of G of size μ, then any vertex cover has to include at least one vertex of each matched edge and therefore the size of a minimum vertex cover is at least that of a maximum matching. The set of all vertices of a maximum matching is clearly a vertex cover of the graph. Thus the size of a minimum vertex cover of $G = \mu + k$, for some $0 \leqslant k \leqslant 2\mu$. The class of graphs for which the size of a maximum matching equals that of a minimum vertex cover are called König graphs and it includes the set of all bipartite graphs.

One can now consider an above-guarantee version of the Vertex Cover problem.

Above Guar Vertex Cover
Input: A graph G with a maximum matching of size μ.
Parameter: A positive integer k.
Question: Does G have a vertex cover of size at most $\mu + k$?

The lower bound guarantee for this problem is different in the sense that it is a function of the *input instance* and not the *input size*. The Above Guar Vertex Cover is fixed-parameter reducible [32] (in fact, fixed-parameter equivalent [15]) to Min 2-Sat Deletion.

The parameterized complexity of these problems was open for quite some time until recently Razgon et al. [40] showed the latter problem to be fixed-parameter tractable. This also shows that Above Guar Vertex Cover is fixed-parameter tractable. Interestingly, it was already known that one can check in polynomial time whether the size of a minimum vertex cover equals that of a maximum matching [10]. This is in contrast to the problems considered before (problems in Section 4) where we do not know whether there exists a polynomial time algorithm to decide whether a given instance has an optimum value equal to the tight lower bound.

7.2.2. The Kemeny score problem

The Kemeny Score problem is a rank-aggregation problem that arises in social choice theory [2,13]. Informally, the goal of this problem is to combine a number of different rank orderings on the same set of candidates to obtain a "best" ordering. An election $(\mathcal{V}, C)$ consists of a set $\mathcal{V}$ of votes and set C of candidates. A *vote* is simply a preference list of candidates (a permutation of C). A "Kemeny consensus" is a preference list of candidates that is "closest" to the given set of votes. Given a pair of votes π_1, π_2, the *Kendall-Tau-distance* (*KT-distance* for short) between π_1 and π_2 is defined as

$$\operatorname{dist}(\pi_1, \pi_2) = \sum_{\{c,d\} \subseteq C} d_{\pi_1,\pi_2}(c, d),$$

where $d_{\pi_1,\pi_2}(c, d) = 0$ if π_1 and π_2 rank c and d in the same order, and 1 otherwise. The *score* of a preference list π with respect to an election $(\mathcal{V}, C)$ is defined as

$$\operatorname{scr}(\pi) = \sum_{\pi_i \in \mathcal{V}} \operatorname{dist}(\pi, \pi_i).$$

A preference list π with minimum score is called a *Kemeny consensus* of $(\mathcal{V}, C)$ and its score $\operatorname{scr}(\pi)$ is called the *Kemeny score* of the election $(\mathcal{V}, C)$.

The Kemeny Score problem is the following.

Kemeny Score
Input: An election $(\mathcal{V}, C)$ and an integer k.
Parameter: The integer k.
Question: Is the Kemeny score of $(\mathcal{V}, C)$ at most k?

This problem is NP-complete even for the case when the number of votes is four, whereas the complexity of the case $|\mathcal{V}| = 3$ is still open [13]. The case $|\mathcal{V}| = 2$ can be solved trivially as the Kemeny score is simply the KT-distance of the two votes.

In [2], Kemeny Score was shown to be in FPT. Also note that there are at most $|C|!$ distinct votes and therefore for constant $|C|$, the problem is polynomial-time solvable irrespective of the number of votes.

Observe that given an election $(\mathcal{V}, C)$ and a preference list π, the score of π with respect to $(\mathcal{V}, C)$ is lower bounded by

$$L_{\mathcal{V},C} = \sum_{\{a,b\}\in C} \min\{\nu(a,b), \nu(b,a)\},$$

where $\nu(a,b)$ is the number of preference lists in $\mathcal{V}$ that rank a higher than b. Thus the Kemeny score of $(\mathcal{V}, C)$ is lower bounded by $L_{\mathcal{V},C}$. Observe that this lower bound is tight since in the case where $\mathcal{V}$ contains $r-1$ identical preference lists along with a single copy of the reverse of these lists, the Kemeny score of $(\mathcal{V}, C)$ equals $L_{\mathcal{V},C} = \binom{|C|}{2}$.

Hence a more natural question is:

Above-Guarantee Kemeny Score
Input: An election $(\mathcal{V}, C)$ and an integer k.
Parameter: The integer k.
Question: Is the Kemeny score of $(\mathcal{V}, C)$ at most $L_{\mathcal{V},C} + k$?

We note that this problem is fixed-parameter tractable by a parameter-preserving reduction to a weighted variant of Directed Feedback Vertex Set, where the vertices have weights and one has to decide whether there exists a feedback vertex set of weight at most k, with k as parameter. The following reduction appears in [13]. Given an election $(\mathcal{V}, C)$, construct an arc-weighted directed graph G on $|C|$ vertices as follows: for each pair of vertices u and v, there exists an arc from u to v if and only if the majority of the votes in $\mathcal{V}$ rank u higher than v; the weight of the arc from u to v is the difference between the number of votes that rank u higher than v and vice versa. Note that in case of tie, there are no arcs between u and v.

Claim. *The election $(\mathcal{V}, C)$ has a Kemeny score of at most $L_{\mathcal{V},C} + k$ if and only if G has a feedback arc set of weight at most k.*

The problem of deciding whether an arc-weighted directed graph has a feedback arc set of size at most k fixed-parameter reduces to the problem of deciding whether a vertex-weighted directed graph has a feedback vertex set of size at most k [14]. This latter problem is in FPT since the algorithm for Directed Feedback Vertex Set presented in [7] actually works for this vertex-weighted variant of the problem. Consequently, Above-Guarantee Kemeny Score is in FPT.

Open Problem 7. Are there other "natural" problems where the lower bound is a function of the input instance rather than the input size? If so, are their above guaranteed versions fixed-parameter tractable?

7.3. Above-guarantee approximation

We examine the notion of an above-guarantee approximation algorithm. The idea, as in recent attempts to study parameterized approximation [5,6,12], is to try and approximate the parameter k by an efficient algorithm. To motivate this discussion, we consider the Vertex Cover problem once more. As we observed already, a graph G with a maximum matching of size μ has a minimum vertex cover of size $\beta(G) = \mu + k$ for some $0 \leqslant k \leqslant 2\mu$. A 2-approximate algorithm for this problem simply includes all vertices of a maximum matching. What is interesting is that no polynomial time algorithm is known for this problem which has an approximation factor a constant strictly less than 2. In fact, it is now an outstanding open problem whether there indeed exists such a polynomial time approximation algorithm.

An above-guarantee approximation algorithm for the Vertex Cover problem tries to approximate k instead of the "entire" vertex cover. For example, an algorithm which outputs a solution of size at most $\mu + \alpha(\beta(G) - \mu)$, where $\alpha > 1$, performs better than the 2-approximate algorithm (the one which includes all vertices of a maximum matching) whenever $\mu + \alpha(\beta(G) - \mu) < 2\mu$, that is, whenever $\beta(G) - \mu < \mu/\alpha$. Since $\mu = O(n)$, this means that whenever $\beta(G) - \mu < O(n/\alpha)$, the additive approximation algorithm beats the 2-approximate algorithm. Here n is the number of vertices in the input graph.

Formally, one can define an above-guarantee α-approximate algorithm as follows. If Q is an NP-maximization problem with a tight lower bound (TLB) on its optimal solution size, then an above-guarantee α-approximate algorithm for Q takes as input an instance I of Q and outputs a solution of size at least $\textsc{tlb}(I) + \alpha(\mathrm{opt}(I) - \textsc{tlb}(I))$ in time polynomial in $|I|$. For an NP-minimization problem, an α-approximate algorithm outputs a solution of size at most $\textsc{tlb}(I) + \alpha(\mathrm{opt}(I) - \textsc{tlb}(I))$ in time polynomial in $|I|$. Note that for an NP-maximization problem $\alpha < 1$; for a minimization problem $\alpha > 1$. One can now think of developing above-guarantee approximate algorithms for other problems which have a guaranteed lower bound on their solution size.

Open Problem 8. Does there exist an above-guarantee approximation algorithm for the following problems?

1. Planar Independent Set.
2. Max Cut.

We note that the standard optimization versions of both these problems have good approximation algorithms. The Planar Independent Set problem has a PTAS due to Baker [1]. For Max Cut, there exists a 0.879-approximate algorithm due to Goemans and Williamson [19]. Baker's algorithm takes as input a planar graph G and a positive integer p and outputs an independent set of size at least $p/(p+1)$ times optimal in time $O(8^p p|V(G)|)$. Thus if a maximum independent set has size $n/4+k$, Baker's algorithm outputs a solution of size at least $p/(p+1) \cdot (n/4+k)$. Here again an above-guarantee approximate algorithm which outputs a solution of size at least $n/4+\alpha k$, α a constant, yields a better solution whenever

$$\frac{p}{p+1}\left(\frac{n}{4}+k\right) < \frac{n}{4}+\alpha k,$$

that is, whenever

$$k < \frac{n}{4\{p-\alpha(p+1)\}}.$$

Thus if the optimum is only a "small distance" away from the lower bound, an above-guarantee approximation performs better than the PTAS. A similar observation can be made for the Max Cut problem. This could be the motivation for developing such algorithms.

In summary, we believe that parameterizing above or below guaranteed bounds is a paradigm that enlarges the range of the parameter for which the parameterized algorithms remain practical. Further, as we have outlined, there are several interesting problems yet to be explored in this paradigm.

References

[1] B. Baker, Approximation algorithms for NP-complete problems on planar graphs, J. ACM 41 (1) (1994) 153–180.
[2] N. Betzler, M.R. Fellows, J. Guo, R. Niedermeier, F.A. Rosamond, Fixed-parameter algorithms for Kemeny scores, in: Proceedings of the 4th International Conference on Algorithmic Aspects in Information and Management (AAIM 2008), in: Lecture Notes in Comput. Sci., vol. 5034, Springer, 2008.
[3] A. Billionnet, On interval graphs and matrix profiles, RAIRO Oper. Res. 20 (1986) 245–256.
[4] L. Cai, J. Chen, On fixed-parameter tractability and approximability of NP-optimization problems, J. Comput. System Sci. 54 (3) (1997) 465–474.
[5] L. Cai, X. Huang, Fixed-parameter approximation: Conceptual framework and approximability results, in: Proceedings of the 2nd International Workshop on Parameterized and Exact Computation, in: Lecture Notes in Comput. Sci., vol. 4169, Springer, 2006, pp. 96–108.
[6] Y. Chen, M. Grohe, M. Grüber, On parameterized approximability, in: Proceedings of the 2nd International Workshop on Parameterized and Exact Computation, in: Lecture Notes in Comput. Sci., vol. 4169, Springer, 2006, pp. 109–120.
[7] J. Chen, Y. Liu, S. Lu, I. Razgon, B. O'Sullivan, Directed feedback vertex set is FPT, in: Proceedings of the 40th ACM Symposium on Theory of Computing (STOC 2008).
[8] J. Cherian, R. Thurimella, Approximating minimum size k-connected spanning subgraphs via matching, SIAM J. Comput. 30 (2) (2000) 528–560.
[9] M. Chudnovsky, G. Cornuéjols, X. Liu, P.D. Seymour, K. Vušković, Recognizing Berge graphs, Combinatorica 25 (2005) 143–186.
[10] R.W. Deming, Independence numbers of graphs – An extension of the König–Egerváry theorem, Discrete Math. 27 (1979) 23–33.
[11] R.G. Downey, M.R. Fellows, Parameterized Complexity, Springer, 1999.
[12] R.G. Downey, M.R. Fellows, C. McCartin, Parameterized approximation algorithms, in: Proceedings of the 2nd International Workshop on Parameterized and Exact Computation, in: Lecture Notes in Comput. Sci., vol. 4169, Springer, 2006, pp. 121–129.
[13] C. Dwork, R. Kumar, M. Naor, D. Sivakumar, Rank aggregation methods revisited, manuscript available at http://www.eecs.harvard.edu/~michaelm/CS222/rank2.pdf.
[14] G. Even, J. Naor, B. Schieber, M. Sudan, Approximating minimum feedback sets and multicuts in directed graphs, Algorithmica 20 (2) (1998) 151–174.
[15] M.R. Fellows, personal communication.
[16] J. Flum, M. Grohe, Parameterized Complexity Theory, Springer, 2006.
[17] M.R. Garey, D.S. Johnson, Computers and Intractability, Freeman, San Francisco, 1981.
[18] M.X. Goemans, Minimum bounded degree spanning trees, in: Proceedings of 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006).
[19] M.X. Goemans, D.P. Williamson, 0.879-Approximation algorithms for MAX CUT and MAX 2SAT, in: Proceedings of the 26th Annual ACM Symposium on Theory of Computing (STOC 1994), 1994, pp. 422–431.
[20] G. Gutin, A. Rafiey, S. Szeider, A. Yeo, The linear arrangement problem parameterized above-guaranteed value, in: Proceedings of the 6th Conference on Algorithms and Complexity (CIAC 2006), in: Lecture Notes in Comput. Sci., vol. 3998, Springer, 2006, pp. 356–367.
[21] G. Gutin, S. Szeider, A. Yeo, Fixed-parameter complexity of minimum profile problems, in: Proceedings of the 2nd International Workshop on Parameterized and Exact Computation (IWPEC 2006), in: Lecture Notes in Comput. Sci., vol. 4169, Springer, 2006, pp. 60–71.
[22] J. Håstad, S. Venkatesh, On the advantage over a random assignment, in: Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC 2002), 2002, pp. 43–52.
[23] P. Heggernes, C. Paul, J.A. Telle, Y. Villanger, Interval completion with few edges, in: Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC 2007), 2007, pp. 374–381.
[24] S. Khanna, R. Motwani, M. Sudan, U. Vazirani, On Syntactic versus computational views of approximability, SIAM J. Comput. 28 (1) (1998) 164–191.
[25] S. Khot, V. Raman, Parameterized complexity of finding subgraphs with hereditary properties, Theoret. Comput. Sci. 289 (2002) 997–1008.
[26] Y. Lin, J. Yuan, Profile minimization problem for matrices and graphs, Acta Math. Appl. Sin. Engl. Ser. 10 (1) (1994) 107–112.
[27] D. Lokshtanov, Wheel-free deletion is W[2]-hard, in: Proceedings of the International Workshop on Exact and Parameterized Computation (IWPEC 2008).
[28] M. Mahajan, V. Raman, Parameterizing above-guaranteed values: MaxSat and MaxCut, J. Algorithms 31 (2) (1999) 335–354.
[29] M. Mahajan, V. Raman, S. Sikdar, Parameterizing MAX SNP problems above-guaranteed values, in: Proceedings of the 2nd International Workshop on Parameterized and Exact Computation, in: Lecture Notes in Comput. Sci., vol. 4169, Springer, 2006, pp. 38–49.
[30] D. Marx, Chordal deletion is fixed-parameter tractable, in: Proceedings of the 32nd International Workshop (WG 2006), in: Lecture Notes in Comput. Sci., vol. 4271, Springer, 2006, pp. 37–48.
[31] D. Marx, I. Schlotter, Obtaining a planar graph by vertex deletion, in: Proceedings of the 33rd International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2007), 2007, pp. 292–303.

[32] S. Mishra, V. Raman, S. Saurabh, S. Sikdar, C.R. Subramanian, The complexity of finding subgraphs whose matching number equals the vertex cover number, in: Proceedings of the 18th International Symposium on Algorithms and Computation (ISAAC 2007), in: Lecture Notes in Comput. Sci., vol. 4835, Springer, 2007, pp. 268–279.
[33] R. Motwani, P. Raghavan, Randomized Algorithms, Cambridge University Press, 1995.
[34] A. Natanzon, R. Shamir, R. Sharan, Complexity classification of some edge-modification problems, in: Proceedings of the 25th International Workshop on Graph-Theoretic Concepts in Computer Science (WG 1999), in: Lecture Notes in Comput. Sci., vol. 1665, Springer, 1999, pp. 65–77.
[35] R. Niedermeier, Invitation to Fixed Parameter Algorithms, Oxford University Press, 2006.
[36] R. Niedermeier, P. Rossmanith, New upper bounds for maximum satisfiability, J. Algorithms 36 (1) (2000) 63–88.
[37] C.H. Papadimitriou, M. Yannakakis, Optimization, approximation, and complexity classes, J. Comput. System Sci. 43 (3) (1991) 425–440.
[38] S. Poljak, D. Turzík, A polynomial algorithm for constructing a large bipartite subgraph with an application to a satisfiability problem, Canad. J. Math. 34 (3) (1982) 519–524.
[39] V. Raman, S. Saurabh, C.R. Subramanian, Faster fixed-parameter tractable algorithms for finding feedback vertex sets, ACM Trans. Algorithms 2 (3) (2006) 403–415.
[40] I. Razgon, B. O'Sullivan, Almost 2-sat is fixed-parameter tractable, in: Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP 2008).
[41] B. Reed, K. Smith, A. Vetta, Finding odd cycle transversals, Oper. Res. Lett. 32 (4) (2004) 299–301.