Algorithmica (2010) 57: 398–412
DOI 10.1007/s00453-008-9223-x

Fixed-Parameter Approximation: Conceptual Framework and Approximability Results

Liming Cai · Xiuzhen Huang

Received: 9 December 2007 / Accepted: 28 July 2008 / Published online: 15 August 2008

Abstract The notion of fixed-parameter approximation is introduced to investigate the approximability of optimization problems within the framework of fixed-parameter computation. This work partially aims at enhancing the world of fixed-parameter computation in parallel with the conventional theory of computation that includes both exact and approximate computations. In particular, it is proved that fixed-parameter approximability is closely related to the approximation of small-cost solutions in polynomial time. It is also demonstrated that many fixed-parameter intractable problems are not fixed-parameter approximable. On the other hand, fixed-parameter approximation appears to be a viable approach to solving some inapproximable yet important optimization problems. For instance, all problems in the class MAX SNP admit fixed-parameter approximation schemes in time $O(2^{O((1-\epsilon/O(1))k)}p(n))$ for any small $\epsilon > 0$.

Keywords Fixed parameter computation · Fixed-parameter approximation · Fixed-parameter tractability · Approximation algorithm · Approximation scheme

1 Introduction

The theory of fixed-parameter complexity was initiated by Downey and Fellows [17, 19] to study the exact computation of important computational problems whose

A preliminary version of this paper was presented at the Second International Workshop on Parameterized and Exact Computation (IWPEC'06), Lecture Notes in Computer Science 4169, pp. 96–108, Zurich, Switzerland, September 13–15, 2006.

L. Cai
Department of Computer Science, The University of Georgia, Athens, GA 30605, USA
e-mail: cai@cs.uga.edu

X. Huang (✉)
Department of Computer Science, Arkansas State University, State University, AR 72467, USA
e-mail: xzhuang@csm.astate.edu

input contains a significant numerical parameter. The complexity of such fixed-parameter problems is measured in the value of the parameter, as well as the size of the input. For example, the problem of determining if a given graph G has a vertex cover of size k can be accomplished in time $O(1.2738^k + kn)$ [14]. Thus the problem is called fixed-parameter tractable since it can be solved feasibly for small or fixed parameters. In contrast, other problems, such as to determine if a given graph has a dominating set of size k, seem not to behave well enough from the fixed-parameter tractability perspective. All known algorithms for such problems run in time $\Omega(n^{k+1})$, formidable even for small values of k. Studies [19] have shown that the fixed-parameter tractability of many computational problems hinges upon the answers to some long-standing open questions in conventional complexity theory [17, 18]. In particular, problems, such as determining the existence of a dominating set of size k, are not fixed-parameter tractable unless SAT $\in DTIME(2^{o(n)})$. Given such problems, it is natural to ask whether fixed-parameter tractable algorithms may exist that can give "approximate" answers to decision questions involving these computational problems [23, 24].

Another issue motivating this research is the "asymmetrical" phenomenon of fixed-parameter algorithms when they are used to find the optimal solution of optimization problems. For instance, consider the algorithm of running time $O(1.2738^k + kn)$ that can determine if a given graph has a vertex cover of (the given) size k. It can be used to determine the size k_0 of the minimum vertex cover and find the cover when $k_0 \leq k$. However, it does not guarantee, with the same running time in k, to determine the size k_0 of the minimum vertex cover for $k_0 > k$. Therefore, one fundamental issue is how much time is needed in estimating value of k_0 with respect to the value of k. Such issue has been addressed for a few individual fixed-parameter problems in the past. A typical example is the algorithm developed for the problem of graph tree decomposition (refer to [19] and [5]). In a fixed-parameter tractable time, it can either determine if a given graph G has a tree width larger than the parameter value k or produces a tree decomposition of width $\leq k$. The yielded tree decomposition is an approximate solution with a "guaranteed ratio" of 1 with respect to the given parameter k but not to the optimum. Like conventional approximation, this new type of approximation produces a range of values as an estimation of the optimum.

We develop this concept into the notion of fixed-parameter (FP-)approximation to study the existence of fixed-parameter tractable algorithms that may give approximate answers for fixed-parameter problems. Intuitively speaking, an FP-approximation algorithm for a minimization problem either gives a negative answer to the question of "$OPT(I) \leq k$?" or produces a solution with value bounded by $g(k)$ for some fixed function g. In addition, the algorithm runs in time $O(f(k)n^c)$ for some fixed function f and a constant $c > 0$. In Sect. 3 we give the formal definition for FP-approximation. FP-approximation is well defined because we can show that either fixed-parameter tractability or polynomial-time approximation would imply FP-approximation. We believe the new notion enhances the parameterization framework to be in parallel with the conventional theory of computing that includes both exact and approximation computation.

We show that fixed-parameter approximability is equivalent to approximating solutions of a small (but unbounded) cost in polynomial time. This complements the known results that fixed-parameter tractability is equivalent to determining solutions of a small cost in polynomial time [8, 9]. In general, we prove that an optimization problem is not fixed-parameter approximable to ck unless the problem of finding solutions of cost bounded by $s(n)$ is approximable to the ratio c in polynomial time, for some unbounded and nondecreasing function $s(n)$. According to the results in [11, 22], for many optimization problems, approximating solutions of a small cost cannot be done in polynomial time. Therefore, our research essentially shows the fixed-parameter inapproximability for many optimization problems. In Sect. 4, we show that a number of fixed-parameter problems, including finding the minimum dominating set, are likely not FP-approximable.

FP-approximation is also considered as an alternative to polynomial-time approximation. In general, we expect the approximability of certain optimization problems to be improved under the fixed-parameter setting. Indeed, in Sect. 5 we show that all optimization problems in the optimization class MAX SNP admit FP-approximation schemes, i.e., they all have algorithms that run in time $O(2^{O((1-\epsilon/O(1))k)}p(n))$ to produce a solution of value bounded by $(1+\epsilon)k$ for any $\epsilon > 0$ for a minimization problem (or, a solution of value bounded by $(1-\epsilon)k$ for any $\epsilon > 0$ for a maximization problem). This complements the known results that MAX SNP-complete problems do not admit polynomial time approximation schemes unless P = NP.

In Sect. 6, we discuss some further results in the approximability improvements with FP-approximation. In fact, Bodlaender and Fellows [6] are able to show fixed-parameter approximability for the problem of BANDWIDTH, which is W[1]-hard under a uniform reduction due to Bodlaender [4]. Moreover, we suspect that for some other polynomial-time approximable optimization problems, the approximation ratios can be improved when fixed-parameter tractability is the only concern regarding the running time of the algorithm. In particular, we show that problem BIN PACKING, which is MAX SNP-hard and fixed-parameter intractable unless P = NP [7], admits an (asymptotic) FP-approximation scheme. Our result improves the one devised by Karmarkar and Karp [28] in the sense that the ϵ we obtain in the ratio $1+\epsilon$ decreases much faster with respect to the value of the optimal solution $OPT(I)$. We conjecture that the techniques used in the proof may be applied to improving the approximation performance of other fixed-parameter problems.

2 Preliminaries

2.1 Polynomial-Time Approximation

We provide some basic terminologies for studying approximation algorithms. For a reference of the theory of approximation, the readers are referred to [2].

An *NP optimization problem* Q is a 4-tuple (I_Q, S_Q, f_Q, OPT), where:

1. I_Q is the set of input instances. It is recognizable in polynomial time.
2. For each instance $I \in I_Q$, $S_Q(I)$ is the set of feasible solutions for I, which is defined by a polynomial p and a polynomial time computable predicate π (p and π only depend on Q) as $S_Q(I) = \{y : |y| \le p(|I|) \text{ and } \pi(I, y)\}$.

3. $f_Q(I, y)$ is the objective function mapping a pair $I \in I_Q$ and $y \in S_Q(I)$ to a non-negative integer. The function f_Q is computable in polynomial time.
4. $OPT \in \{\max, \min\}$. Q is called a *maximization problem* if $OPT = \max$, and a *minimization problem* if $OPT = \min$.

An *optimal solution* y_0 for an instance $I \in I_Q$ is a feasible solution in $S_Q(I)$ such that $f_Q(I, y_0) = OPT\{f_Q(I, z) \mid z \in S_Q(I)\}$. We will denote by $OPT(I)$ the value $OPT\{f_Q(I, z) \mid z \in S_Q(I)\}$.

An algorithm A is called an *approximation algorithm* for an NP optimization problem $Q = (I_Q, S_Q, f_Q, OPT)$ if, for each input instance I in I_Q, A returns a feasible solution $y_A(I)$ in $S_Q(I)$. The solution $y_A(I)$ has an *approximation ratio* $r(n)$ if it satisfies the following condition:

$$OPT(I)/f_Q(I, y_A(I)) \le r(|I|) \quad \text{if } Q \text{ is a maximization problem,}$$

$$f_Q(I, y_A(I))/OPT(I) \le r(|I|) \quad \text{if } Q \text{ is a minimization problem.}$$

The approximation algorithm A has an *approximation ratio* $r(n)$ if for any instance I in I_Q, the solution $y_A(I)$ constructed by the algorithm A has an approximation ratio bounded by $r(|I|)$.

An NP optimization problem Q has a *polynomial-time approximation scheme* (PTAS) if there is an algorithm A_Q that takes a pair (I, ϵ) as input, where I is an instance of Q and $\epsilon > 0$ is a real number, and returns a feasible solution y for I such that the approximation ratio of the solution y is bounded by $1 + \epsilon$, and for each fixed $\epsilon > 0$, the running time of the algorithm A_Q is bounded by a polynomial of $|I|$.

For example, the time complexity of a PTAS algorithm may be of the form $O(2^{1/\epsilon}|I|^c)$ for a fixed constant c or of the form $O(|I|^{1/\epsilon})$. The latter type of computations with small ϵ values may turn out to be practically intractable. This leads to the following definition in [12].

An NP optimization problem Q has an *efficient polynomial-time approximation scheme* (EPTAS) if it admits a polynomial-time approximation scheme whose time complexity is bounded by $O(f(1/\epsilon)|I|^c)$, where f is a recursive function and c is a constant.

2.2 Parameterized Complexity

We briefly introduce some basic concepts in the theory of fixed-parameter complexity. We refer the reader to Downey and Fellows [19] for further details.

A fixed-parameter problem is defined over $\Sigma^* \times N$, where Σ is a finite alphabet and N is the set of natural numbers. Each instance of the fixed-parameter problem Π is a pair (I, k), where k is called the *parameter*. A fixed-parameter problem is *fixed-parameter tractable* if there is an algorithm to decide the membership of the problem in time $O(f(k)n^c)$, where $f(k)$ is a recursive function and c is a constant.

The fixed-parameter (decision) version of a given optimization problem Π is defined in [7]. It asks whether the problem has the optimal cost $OPT(I) \ge k$, or $OPT(I) \le k$, depending on whether Π is a maximization or minimization problem.

In this paper, whenever there is no confusion, we do not distinguish between an optimization problem and its fixed-parameter decision version. In our discussions, we always assume that the complexity functions are "nice" with both domain and range being non-negative integers and the values of the functions and their inverses can be easily computed.

3 Definition of FP-Approximation

We define the fixed-parameter approximation in the following. This notion was originally conceived by Downey and Fellows [23, 24]. Independently, Chen et al. [16] and Downey et al. [20] introduced similar frameworks of parameterized approximation from slightly different perspectives. Recently Marx has a review paper [29] "Parameterized Complexity and Approximation Algorithms", which discussed the different ways of parameterized approximation, provided a review of the results from these works, and proposed future research directions on parameterized approximation.

Definition Let Π be the fixed-parameter version of a minimization problem. Let f be a (recursive) function, p be a polynomial independent of f, and k be a constant. Then Π is *fixed-parameter approximable to $g(k)$ in time $O(f(k)p(n))$* for some fixed function g if there is an algorithm such that given any instance I with parameter k, and question $OPT(I) \leq k$, the algorithm which runs in $O(f(k)p(n))$ steps, where $n = |I|$, (1) either outputs "no" (asserting the optimal cost is larger than k), or (2) outputs "yes" and produces a solution of cost at most $g(k)$.

Fixed-parameter approximation for a maximization problem can be defined similarly.

The problem Π is called *FP-approximable to $g(k)$* or simply *FP-approximable* when g is linear in k. The algorithm is called an *FP approximation algorithm* for the problem. We define *FPA* to be the class of fixed-parameter optimization problems that can be FP-approximable to $g(k)$ for some recursive function g.

Definition A minimization (or, maximization) problem Π is said to have a *fixed-parameter approximation scheme*, if the minimization problem Π can be FP-approximable to $(1+\epsilon)k$ for any given small constant $\epsilon > 0$ (or, if the maximization problem Π can be FP-approximable to $(1-\epsilon)k$ for any given small constant $\epsilon > 0$).

In the above definition, the time function of the approximation algorithm may depend on the given value of ϵ. The fixed-parameter approximation scheme is *efficient* if the polynomial p in the time function $O(f(k)p(n))$ does not depend on ϵ. In addition, when Π is fixed-parameter approximable under the situation for sufficiently large k, it is called *asymptotic* fixed-parameter approximable.

We show that the fixed-parameter approximability is well-defined under the parameterization framework. Essentially, we prove that fixed-parameter tractability implies fixed-parameter approximability. As shown in the following, this relationship holds for many optimization problems that are *solution-constructible*, a property first introduced by Cai and Chen [7]. (Note that in [7] the term *fixed-parameter tractability with witness* is used instead of solution constructible. Interested readers are referred to [32] for another related notion of self-reducibility.)

Definition Assume the existence of a fixed-parameter algorithm with running time $T(|I|, k)$ that can determine $OPT(I) \leq k$ (or, $OPT(I) \geq k$) for each input (I, k). A minimization (or, maximization) problem is *solution-constructible* if for any input (I, k), a solution of cost at most (or, at least) k (if it does exist) can be constructed in time polynomial in both $|I|$ and $T(|I|, k)$.

Almost all NP-hard optimization problems studied in the literature [2, 25, 27] are solution-constructible. In fact, solution-constructibility is one of the necessities that the decision problem formulation characterizes the difficulty of the original optimization problem.

Theorem 3.1 *Let* Π *be a fixed-parameter version of some solution-constructible optimization problem. If* Π *is fixed-parameter tractable*, *then* Π *is fixed-parameter approximable to* k.

Proof We briefly verify this for minimization problems. The proof for maximization problems is similar. We describe a fixed-parameter tractable approximation algorithm for the problem Π as follows.

Let B be an algorithm that solves the decision problem Π. For each input of size n and parameter k, it runs in time $O(h(k)n^c)$ for some (recursive) function $h(k)$ and some constant $c > 0$. Given an input (I, k), B is called to determine $OPT(I) \leq k$. If $k < OPT(I)$, output "no" and halt. Otherwise, because Π is solution-constructible, there is an algorithm that constructs a solution of cost at most k. The time used for the construction is $O(T^d p(n))$ for some constant d and polynomial p, where T is the running time of B. So the total running time is bounded by $O(h(k)^d q(n))$ for some polynomial q. Therefore, Π is fixed-parameter approximable to k in time $O(h(k)^d q(n))$. □

We establish a close relationship between polynomial-time approximation and fixed-parameter approximability. We show the simple fact that if an optimization problem admits a polynomial-time approximation algorithm then the problem is fixed-parameter approximable. We consider minimization problems only. Similar results hold for maximization problems as well.

Theorem 3.2 *Let* Π *be a minimization problem that is approximable in polynomial time to the ratio* $r \geq 1$ *for some constant* r. *Then* Π *is fixed-parameter approximable to* rk.

Proof Let A be a polynomial-time approximation algorithm for Π achieving the ratio r. Let $A(I)$ be the cost of a solution S obtained by the algorithm A such that $A(I)/OPT(I) \leq r$. Given an input I and a value for the parameter k, one can approximately answer the question "is $OPT(I) \leq k$?" by utilizing the algorithm A in the following way. First, answer "no" if $k < A(I)/r$. According to the definition of fixed-parameter approximability, this is the only correct answer since $k < A(I)/r \leq OPT(I)$. Second, if $k > A(I)$, or $A(I)/r \leq k \leq A(I)$, then output the solution S. Note that in the second case, the algorithm returns a solution with cost $A(I) \leq rk$. □

4 Fixed-Parameter Inapproximability

Many optimization problems including CLIQUE, DOMINATING SET, and LONGEST COMMON SUBSEQUENCE are provably fixed-parameter intractable, i.e., they are hard for various levels of the W-hierarchy. Since the set of all fixed-parameter tractable problems forms the lowest level of the W-hierarchy, the above problems are not fixed-parameter tractable unless the W-hierarchy collapses to the lowest level. It is desirable to know whether there are fixed-parameter tractable algorithms that give approximate answers. In this section, we show that for many optimization problems such approximation algorithms do not exist either. First, we relate fixed-parameter approximability to the approximation of solutions with a small cost in polynomial time.

Lemma 4.1 *Let* Π *be an optimization problem and* r *be a constant. Then* Π *is not fixed-parameter approximable to* rk *unless* Π *is approximable in time* $t(OPT(I))n^c$ *to the ratio* r *for some (recursive) function* t *and some constant* c.

Proof We give the proof for minimization problems. The proof for maximization problems is similar.

Let Π be a minimization problem. Assume that Π is fixed-parameter approximable to rk. Then there is an algorithm A for Π that runs in time $O(f(k)n^c)$, for some nondecreasing function f and constant c. Furthermore, given an input (I, k), the algorithm gives an approximate answer to the question whether $OPT(I) \leq k$, for each value of the parameter k. Fix an input I. Then for the parameter $k = 0, 1, \ldots, k_0$, where $k_0 = (OPT(I) - 1)/r$, the algorithm will output "no" because producing a solution of cost $A(I) \leq rk \leq r(OPT(I)-1)/r)$ implies that $A(I) \leq OPT(I)-1$. Let k_1 be the largest value of the parameter k that allows the algorithm to output "no". Then $k_1 + 1$ allows the algorithm to produce a solution with cost $\leq r(k_1 + 1) \leq rOPT(I)$ because the answer of the algorithm on the parameter $k = OPT(I)$ cannot be "no".

We construct an algorithm B that calls the fixed-parameter approximation algorithm A for $k = 0, 1, \ldots,$ until A produces a solution. According to the analysis above, the solution has the cost $rk \leq r(k_1+1) \leq rOPT(I)$. The solution achieves the ratio $rOPT(I)/OPT(I) = r$. The total running time for algorithm B is bounded by $(k_1 + 1)f(k_1 + 1)n^c \leq OPT(I)f(OPT(I))n^c = t(OPT(I))n^c$, where $t(OPT(I)) = OPT(I)f(OPT(I))$. □

Lemma 4.1 gives some surprisingly nice implications.

Theorem 4.1 *For any constant* $r > 1$, *a minimization problem* Π *is not fixed-parameter approximable to* rk, *unless the problem of finding solutions of cost bounded by* $s(n)$ *can be approximated to the ratio* r *in polynomial time, for some unbounded non-decreasing function* $s(n)$.

Proof Assume that Π is fixed-parameter approximable to rk. By Lemma 4.1 and its proof, Π can be approximated to the ratio r by an algorithm A of time $t(OPT(I))n^d$ for some recursive function t and some constant d. When the input instance I is subject to the restriction that $OPT(I) \leq s(n)$, where s is the inverse function of t, the algorithm A runs in time $t(s(n))n^d = n^{d+1}$ to approximate solutions of cost at most $s(n)$ (see [7, 8] for techniques for constructing and computing inverse functions). □

Theorem 4.1 has a direct impact on the fixed-parameter approximability of the problem DOMINATING SET. We have the following result.

Corollary 4.1 *For any $r \geq 1$,* DOMINATING SET *is not fixed-parameter approximable to rk in time $O(2^k n^d)$, where d is a constant, unless* LOG DOMINATING SET *is approximable to the ratio r in polynomial time, where* LOG DOMINATING SET *is the problem of finding dominating sets of size $\leq \log n$ in a given graph of n vertices.*

We then relate the fixed-parameter approximability of the problem DOMINATING SET to another problem TOURNAMENT DOMINATING SET, which is complete under the polynomial time reduction for the complexity classes LOGSNP introduced by Papadimitriou and Yannakakis [31].

A tournament graph is a directed graph $G = (V, E)$, where for any two vertices $u, v \in V$, $u \neq v$, exactly one of the directed edge (u, v) or (v, u) is in E.

> TOURNAMENT DOMINATING SET-DECISION: given a tournament graph G, and a parameter k, is there is there a dominating set of size at most k for the graph G?

Note that [31] when $k > \log n$, the answer to TOURNAMENT DOMINATING SET-DECISION is always positive. Therefore, the problem instances become non-trivial only when $k \leq \log n$. The problem can be solved in time $O(n^{\log n})$. Hence, the problem is unlikely to be NP-hard. It is unknown whether the problem is solvable in polynomial time.

The following is the optimization problem.

> TOURNAMENT DOMINATING SET: given a tournament graph G, try to find a minimum dominating set for the graph G.

From Theorem 4.1, we have

Corollary 4.2 *For any $r \geq 1$,* DOMINATING SET *is not fixed-parameter approximable to rk in time $O(2^k n^d)$, where d is a constant, unless* TOURNAMENT DOMINATING SET *is approximable to the ratio r in polynomial time.*

Interested readers are referred to [13, 26] for a recent complementary result which shows the impracticability of approximation algorithms with small approximation ratio for LOG DOMINATING SET and TOURNAMENT DOMINATING SET. Specifically, it was shown in [13, 26] that LOG DOMINATING SET and TOURNAMENT DOMINATING SET are unlikely to have polynomial time approximation schemes of running time $f(1/\epsilon)n^{o(1/\epsilon)}$, for any recursive function f.

Theorem 4.1 shows the close relationship between fixed-parameter approximability and the ability to approximate small-value instances in polynomial time. The approximability of optimization problems with small value constrained objective functions have been studied in the literature (see for example, [11, 22]). In partciular, it has been shown that, for a number of important problems, including CLIQUE, DOMINATING SET that are hard for the W-hierarchy, the approximation for small value instances, such as LOG CLIQUE, LOG DOMINATING SET, is as hard as subexponential-time (randomized) simulation of NP-computation. Note that subexponential-time (randomized) simulation of NP-computation is a seemingly weaker assumption than P $=$ NP but still highly unlikely. Since the collapsing of the W-hierarchy would also imply subexponential-time simulation of NP-computation, the fixed-parameter inapproximability results developed in this section complement nicely the fixed-parameter intractability results developed in [7, 9, 19].

5 FP-Approximation Scheme for Problems in MAX SNP

In this section, we show that the notion of FP-approximation provides a viable alternative to approximation. In particular, we prove that FP-approximation schemes exist for all problems in the class MAX SNP. This is somewhat surprising since it is a well-known result that MAX SNP-complete problems do not admit polynomial-time approximation scheme unless P $=$ NP.

The class MAX SNP was introduced by Papadimitriou and Yannakakis [30] to capture a collection of optimization problems. For the purpose of investigating the approximability of these problems, the following approximation-preserving reduction was introduced.

Definition (see [30]) Let Π_1 and Π_2 be two optimization problems with cost functions f_1 and f_2. Π_1 *L-reduces* to Π_2 if there are two polynomial time algorithms A and B and two constants $\alpha, \beta > 0$ such that for each instance I_1 of Π_1, (1) the algorithm A produces an instance $I_2 = A(I_1)$ such that $OPT_{\Pi_2}(I_2) \leq \alpha OPT_{\Pi_1}(I_1)$, and (2) given any solution S_2 for I_2 with cost $f_2(I_2, S_2)$, algorithm B produces a solution S_1 for I_1 with cost $f_1(I_1, S_1)$ such that $|OPT_{\Pi_1}(I_1) - f_1(I_1, S_1)| \leq \beta |OPT_{\Pi_2}(I_2) - f_2(I_2, S_2)|$.

It is known from the work of Cai and Chen [7] that the standard parameterized versions of all maximization problems in the class MAX SNP are fixed-parameterized tractable. The proof of this earlier result was later refined by Cai and Juedes [10] to show that L-reductions actually preserve subexponential-time computability. We show in the following that L-reduction preserves FP-approximation scheme as well.

Lemma 5.1 *Let* Π_1 *and* Π_2 *are two minimization problems with* Π_1 *L-reducible to* Π_2. *Then* Π_1 *has an FP-approximation scheme if* Π_2 *has one*.

Proof Let algorithms A and B be the two algorithms associated with the L-reduction of running time $q(n)$ and $r(n)$ respectively. Also let α, β be the two associated constants. We assume that Π_2 admits a FP-approximation scheme M with running time $f(k,\epsilon)p(n,\epsilon)$. We give in the following an FP-approximation scheme for problem Π_1.

FP-APPROXIMATION SCHEME FOR Π_1

1. Given an input instance I_1 for Π_1, parameter k, and $\epsilon > 0$, call algorithm A to produce an instance $I_2 = A(I_1)$ for Π_2.
2. For $h = 0, 1, 2, \ldots, \alpha k$, and $\epsilon' = \epsilon/\alpha\beta$, run M on the instance I_2 to decide if $OPT_{\Pi_2}(I_2) \le h$. Then there must exist an h_0, $0 \le h_0 \le \alpha k$, such that M answers "no" on all $h = 0, 1, \ldots, h_0$ but yields on $h_0 + 1$ an solution S_2 with value $f_2(I_2, S_2) \le (1+\epsilon')(h_0+1)$.
3. If $h_0 = \alpha k$, it is clear that $OPT_{\Pi_2}(I_2) > \alpha \cdot k$. By the L-reduction, $OPT_{\Pi_2}(I_2) \le \alpha OPT_{\Pi_1}(I_1)$, it can be concluded that $OPT_{\Pi_1}(I_1) > k$. So output "no" and stop.
4. Otherwise, the solution S_2 satisfies $f_2(I_2, S_2) \le (1+\epsilon')(h_0+1) \le (1+\epsilon')\alpha k$. Because $OPT_{\Pi_2}(I_2) > h_0$, $OPT_{\Pi_2}(I_2) \ge h_0 + 1$. This means the ratio $f_2(I_2, S_2)/OPT_{\Pi_2}(I_2) \le 1 + \epsilon'$.
5. Call algorithm B on S_2 and I_2 to produce a solution S_1 of value $f_1(I_1, S_1)$. This solution S_1 satisfies ratio $f_1(I_1, S_1)/OPT_{\Pi_1}(I_1) \le 1 + \alpha\beta\epsilon' = 1 + \epsilon$ (refer to [30], Proposition 2).
6. If $f_1(I_1, S_1) \le (1+\epsilon)k$, output solution S_1 and stop.
7. Otherwise, $f_1(I_1, S_1) > (1+\epsilon)k$. Because $f_1(I_1, S_1) \le (1+\epsilon)OPT_{\Pi_1}(I_1)$, this implies $OPT_{\Pi_1}(I_1) > k$. Output "no" and stop.

On the question "$OPT_{\Pi_1}(I_1) \le k$?", the above algorithm outputs either "no" or a solution S_1 of value $f_1(I_1, S_1) \le (1+\epsilon)k$. The total time is $q(n) + r(n) + \alpha k f(k, \epsilon')p(n, \epsilon')$, which is bounded by $O(f'(k,\epsilon)p'(n,\epsilon))$ for some polynomial p' and recursive function f'. □

By carefully examining the time complexity analysis given in the proof of Lemma 5.1, we have the following technical lemma

Lemma 5.2 *Let Π_1 and Π_2 are two minimization problems with Π_1 L-reducible to Π_2 associated with two constants α and β. If Π_2 has an FP-approximation scheme of time $O(2^{O((1-\epsilon)k)}p(n))$ for some polynomial p, then Π_1 has an FP-approximation scheme of time $O(2^{O((1-\epsilon/\alpha\beta)k)}q(n))$ for some polynomial q.*

Before we establish the main result of this section, we prove the following

Lemma 5.3 *The MAX SNP-complete problem* VERTEX COVER *with bounded degree* 3 (*abbreviated as VC*-3) *admits an FP-approximation scheme of running time $O(2^{O((1-\epsilon/O(1))k)}p(n))$, where p is a polynomial.*

Proof Let $G = (V, E)$ be a connected graph with bounded degree 3. Suppose $n = |V| > 4$, k is the parameter, and $0 < \epsilon < 1$. We first partition the vertex set V into two subsets V_1 and V_2 such that $|V_1| = \delta n$ and $|V_2| = (1-\delta)n$, where $\delta = \epsilon/4$. Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be the two subgraphs of G induced by V_1 and V_2 respectively. Then we apply the fixed-parameter algorithm in [14] to the subgraph G_2.

Suppose the minimum vertex cover found by the fixed-parameter algorithm for the subgraph G_2 is C_2. It is clear that the vertices in C_2 plus the vertices in V_1 form a vertex cover of G. Let $OPT(G)$ be the size of the minimum vertex cover of G. We claim that $(|C_2| + |V_1|)/OPT(G) \leq 1 + \epsilon$.

Since $G_2 = (V_2, E_2)$ is a subgraph of the graph G, the size of its minimum vertex cover is not larger than the size of the minimum vertex cover of G. That is, $|C_2| \leq OPT(G)$. For the graph G with n vertices, the number of edges $|E| \geq n - 1$. Since G has bounded degree 3, the size of the minimum vertex cover $|OPT|$ satisfies: $|OPT| \geq (n-1)/3 \geq n/4$, since $n > 4$. Therefore,

$$\begin{aligned}(|C_2| + |V_1|)/OPT(G) &\leq (OPT(G) + |V_1|)/OPT(G) \\ &= 1 + \delta n/OPT(G) \\ &\leq 1 + \delta n/(n/4) \\ &\leq 1 + 4\delta \\ &= 1 + \epsilon.\end{aligned}$$

The running time of the FP-approximation algorithm is bounded by the time for finding the minimum vertex cover for G_2, which is $O(1.2738^{O((1-\delta)n)}p(n))$ [14]. Since $k \geq OPT(G) \geq n/4$, and $\delta = \epsilon/4$, we have the time complexity $O(1.2738^{O((1-\delta)k)}p(n)) = O(1.2738^{O((1-\epsilon/4)k)}p(n))$. □

By Lemma 5.2 and Lemma 5.3, we have

Theorem 5.1 *All minimization problems in the class MAX SNP admit FP-approximation schemes of time* $O(2^{O((1-\epsilon/O(1))k)}p(n))$, *where* p *is a polynomial.*

Note that for maximization problems in the class MAX SNP, it is proved by Cai and Chen (Theorem 3.4, [7]) that they are in FPT and can be solved in time $O(2^k p(n))$. These exact algorithms can determine if $OPT \geq (1-\epsilon)k$ in time $O(2^{O((1-\epsilon)k)}p(n))$. Therefore,

Theorem 5.2 *All maximization problems in the class MAX SNP admit FP-approximation schemes of time* $O(2^{O((1-\epsilon)k)}p(n))$, *where* p *is a polynomial.*

In conclusion,

Corollary 5.1 *All problems in the class MAX SNP admit FP-approximation schemes of time* $O(2^{O((1-\epsilon/O(1))k)}p(n))$, *where* p *is a polynomial.*

Theorem 4.1 shows that, with an amount of time less than what is needed by exact algorithms, the optimization problems in MAX SNP can be FP-approximated to an arbitrary accuracy. On the other hand, any improvement in the running time of the parameterized algorithms for the parameterized decision problems would imply more efficient FP-approximation schemes for the problems. For example, by applying the parameterized algorithm of time $O(1.194^k k^2 + n)$ for the problem VC-3 in [15], the time complexity of the FP-approximation scheme for VC-3 could be improved from the aforementioned $O(1.2738^{O((1-\epsilon/4)k)}p(n))$ to $O(1.194^{O((1-\epsilon/4)k)}q(n))$, where p and q are polynomials.

6 Improving Approximability for BIN PACKING

In this section, we show that the notion of FP-approximation can be used to improve the approximation of parameterized intractable problems as well.

Consider the BIN PACKING problem [25]: Given a finite set $U = \{u_1, \ldots, u_n\}$ of items and a rational size $s(u) \in [0, 1]$ for each item $u \in U$, find a partition of U into disjoint subsets $U_1, \ldots, U_k$ such that the sum of the sizes of the items in each U_i is no more than 1 and k is minimized. Note that the decision problem for $k = 3$ is NP-hard [25] and that it is not fixed-parameter tractable unless P = NP [7]. So, we cannot claim fixed-parameter approximability simply from the results shown in Sect. 2. However, based on Theorem 3.2 and the fact that this problem can be approximated to the ratio 3/2 [33], we obtain the following result. (Note that better asymptotic ratios for BIN PACKING are known [2].)

Corollary 6.1 BIN PACKING *is fixed-parameter approximable to* $(3/2)k$.

We show in the following that this approximation ratio can be significantly improved when fixed-parameter feasibility is the only concern regarding the running time of the algorithm.

Let $0 < \epsilon \le 1/2$ be a rational number such that $\epsilon \le 1/OPT(U)$, where U is the set of items for an instance of BIN PACKING. Let $V \subseteq U$ such that $u \in V$ if and only if $s(u) \ge \epsilon$. We have the following lemma.

Lemma 6.1 *Let B be an optimal packing for the set V. There is an algorithm A that uses B to construct an approximate packing for U such that $A(U)/OPT(U) \le 1 + 2\epsilon + 2/OPT(U)$.*

Proof Assume $B = U_1, \ldots, U_p$ is an optimal packing for V. Then $B(V) = OPT(V) = p$. The algorithm constructs another packing A for U as follows. Pick items from $U - V$ and pack them into bins $U_1, \ldots, U_p$ using the well-known FFD (first fit decreasing) heuristic [25]. Notice that FFD may need to use some extra bins $U_{p+1}, \ldots, U_q$ to complete the packing.

The size of the space left in each bin U_i, $i = p + 1, \ldots, q - 1$ must be smaller than ϵ since each element in $U - V$ has size $< \epsilon$. Moreover, observe that (1) if $q > p$, the size of the space left in each U_i, $i = 1, \ldots, p$ is also smaller than ϵ; and (2) if $p = q$, $A(U) = OPT(U)$. It is easy to see that the observation (1) is true. We justify observation (2). When $p = q$, no new bins are needed to pack the items from the set $U - V$. If $A(U) > OPT(U)$, there must be an optimal packing for U that uses a smaller number of bins than p. If we take out all items in $U - V$ from this packing, then this results in a packing for the set V that uses fewer than $OPT(V)$ bins. This is a contradiction.

Based on these observations, $A(U)/OPT(U) = 1$ in the case when $p = q$. When $q > p$, we have two cases. When $\sum_{u\in U_q} s(u) \geq 1-\epsilon$, we have

$$\begin{aligned} OPT(U) \geq \sum_{u\in U} s(u) = \sum_{i=1}^{q} \sum_{s\in U_i} s(u) \\ \geq q(1-\epsilon) \\ \geq A(U)(1-\epsilon). \end{aligned}$$

Since $\epsilon \leq 1/2$, we have that $A(U)/OPT(U) \leq 1+2\epsilon$. When $\sum_{u\in U_q} s(u) < 1-\epsilon$, we have

$$\begin{aligned} OPT(U) \geq \sum_{u\in U} s(u) = \sum_{i=1}^{q} \sum_{s\in U_i} s(u) \\ \geq (q-1)(1-\epsilon) + \sum_{u\in U_q} s(u) \\ \geq A(U)(1-\epsilon) - \alpha, \end{aligned}$$

where $\alpha = -(\epsilon - 1 + \sum_{u\in U_q} s(u))$. It is clear that $\alpha \leq 1$. Since $(1-\epsilon) \geq 1/2$, $A(U)/OPT(U) \leq 1 + 2\epsilon + 2\alpha/OPT(U) \leq 1 + 2\epsilon + 2/OPT(U)$. □

Now we are ready to prove the main theorem of this section.

Theorem 6.1 *For any $\delta > 0$, problem* BIN PACKING *is (asymptotic) fixed-parameter approximable to* $(1+\delta)k$.

Proof We describe a fixed-parameter tractable process for BIN PACKING in the following. For a given δ, let $\epsilon = \delta/4$.

Given a set U of items and an integer k for the problem, first run the approximation algorithm FFD on U. Let the number of bins in the packing obtained by the algorithm be m and the approximation ratio be r. Output "no" and halt if $k < m/r$. Otherwise, let $V \subseteq U$ such that $u \in V$ if and only if $s(u) \geq \epsilon$. Then find an optimal packing B for set V by a brute force approach. According to Lemma 6.1, there is a packing algorithm A for the original set U of items achieving the ratio bounded by $1 + 4\epsilon = 1 + \delta$, because the ratio $1 + 2\epsilon + 2/OPT(U)$ is asymptotically bounded by $(1 + 4\epsilon)$ for a sufficiently large number of bins for the problem BIN PACKING.

Now output "no" and halt if $k < A(U)/(1+\delta)$. If $k \geq A(U)/(1+\delta)$, output the packing with cost $A(U)$ bounded by $(1+\delta)k$. The running time of FFD is quadratic in n. The time for A to construct the packing is also bounded by a polynomial in n. The time for constructing the packing B is $O(|V||V|)$. Note that $|V| < OPT(V)/\epsilon$ because each item in set V has the size at least ϵ and it needs at least $\epsilon|V|$ number of bins for the packing V alone. It is clear that $OPT(V) \leq OPT(U) \leq m \leq kr$, where m is the cost of the packing obtained by FFD. So $|V| \leq ck$ for some constant c and the time for constructing packing B is bounded by $O((ck)^{ck})$. It is not hard to see that the total time for the above process is $O((ck)^{ck} + n^d)$ for some constant d. □

According to the proof of Theorem 6.1, the total time of $O((ck)^{ck} + n^d)$ is very impractical when k is a large number. We obtain the following corollary regarding approximation of a small number of bins for the problem BIN PACKING.

Corollary 6.2 *There is an asymptotic polynomial-time approximation scheme for* BIN PACKING *when the number of bins is bounded by* $\log n / \log\log n$.

Note that BIN PACKING cannot be approximated to the ratio $3/2 - \epsilon$ in polynomial time for any $\epsilon > 0$ unless P = NP [25]. Therefore, it is not possible for the problem to have a polynomial-time approximation scheme. However, Karmarkar and Karp [28] devised an efficient approximation scheme for the problem with asymptotic ratio $1 + \beta$, where $\beta = O(\log^2 OPT(U)/OPT(U))$. Our proof in Lemma 6.1 shows an improvement in that the ratio $1 + \beta$ is determined by $\beta = O(1/OPT(U))$.

7 Further Research Work

This is a preliminary work for the new framework of fixed-parameter approximation. Further research may derive more positive and negative results. For example, the work on the fixed-parameter approximability of planar graph problems, such as PLANAR VERTEX COVER, PLANAR DOMINATING SET, and PLANAR INDEPENDENT SET, will be very interesting, compared with the results of Alber et al. [1], the $O(2^{\sqrt{k}} n^c)$-time parameterized algorithms, and of Baker [3], the $O(2^{1/\epsilon} n^c)$-time PTAS algorithm for these problems. Also we believe the inapproximability results derived in the present paper may shed light on the study of the FP-approximability of the problems in the newly-proposed parameterized class MINI[1] (refer to [21]).

Acknowledgements The authors would like to thank Rod Downey and Mike Fellows regarding the definition of the fixed-parameter approximability for the problem DOMINATING SET, and Jianer Chen and Mike Fellows for comments on the preliminary version of this work.

References

1. Alber, J., Fernau, H., Niedermeier, R.: Parameterized complexity: exponential speed-up for planar graph problems. J. Algorithms **52**, 26–56 (2004)
2. Ausiello, G., Crescenzi, P., Gambosi, G., Kann, V., Marchetti Spaccamela, A., Protasi, M.: Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties. Springer, Berlin (1999)
3. Baker, B.S.: Approximation algorithms for NP-complete problems on planar graphs. J. Assoc. Comput. Mach. **41**, 153–180 (1994)
4. Bodlaender, H.: Unpublished manuscipt
5. Bodlaender, H.L.: A linear-time algorithm for finding tree-decompositions of small treewidth. SIAM J. Comput. **25**, 1305–1317 (1996)
6. Bodlaender, H., Fellows, M.R.: Unpublished manuscipt
7. Cai, L., Chen, J.: On fixed-parameter tractability and approximability of NP optimization problems. J. Comput. Syst. Sci. **54**, 465–474 (1997)
8. Cai, L., Chen, J.: On the amount of nondeterminism and the power of verifying. SIAM J. Comput. **26**, 733–750 (1997)
9. Cai, L., Chen, J., Downey, R., Fellows, M.: On the structure of parameterized problems in NP. Inf. Comput. **123**, 38–49 (1995)

10. Cai, L., Juedes, D.W.: On the existence of sub-exponential time parameterized algorithms. J. Comput. Syst. Sci. **67**, 789–807 (2003)
11. Cai, L., Juedes, D., Kanj, I.: The inapproximability of non NP-hard optimization problems. Theor. Comput. Sci. **289**, 553–571 (2002)
12. Cesati, M., Trevisan, L.: On the efficiency of polynomial time approximation schemes. Inf. Process. Lett. **64**, 165–171 (1997)
13. Chen, J., Huang, X., Kanj, I., Xia, G.: W-hardness under linear FPT-reductions: structural properties and further applications. In: Proc. of COCOON. Lecture Notes in Computer Science, vol. 3595, pp. 975–984. Springer, Berlin (2005)
14. Chen, J., Kanj, I., Xia, G.: Improved parameterized upper bounds for vertex cover. In: Proc. of MFCS. Lecture Notes in Computer Science, vol. 4162, pp. 238–249. Springer, Berlin (2006)
15. Chen, J., Kanj, I., Xia, G.: Labeled search trees and amortized analysis: improved upper bounds for NP-hard problems. Algorithmica **43**, 245–273 (2005)
16. Chen, Y., Grohe, M., Gruber, M.: On parameterized approximability. In: Proc. of IWPEC. Lecture Notes in Computer Science, vol. 4169, pp. 109–120. Springer, Berlin (2006)
17. Downey, R.G., Fellows, M.R.: Fixed-parameter intractability. In: Proc. of the 7th Annual Conference on Structure in Complexity Theory, pp. 36–49 (1992)
18. Downey, R.G., Fellows, M.R.: Fixed-parameter tractability and completeness I: Basic results. SIAM J. Comput. **24**, 873–921 (1995)
19. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Monographs in Computer Science. Springer, New York (1999)
20. Downey, R., Fellows, M., McCartin, C.: Parameterized approximation algorithms. In: Proc. of IW-PEC. Lecture Notes in Computer Science, vol. 4169, pp. 121–129. Springer, Berlin (2006)
21. Downey, R.G., Estivill-Castro, V., Fellows, M.R., Prieto, E., Rosamond, F.A.: Cutting up is hard to do: the parameterized complexity of k-cut and related problems. Electr. Notes Theor. Comput. Sci. **78**, 209–222 (2003)
22. Feige, U., Kilian, J.: On limited versus polynomial nondeterminism, Chic. J. Theor. Comput. Sci., MIT Press (1997)
23. Fellows, M.R.: Personal communications (1999)
24. Fellows, M.R.: Personal communications (2005)
25. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-completeness. Freeman, San Francisco (1979)
26. Huang, X.: Parameterized complexity and polynomial-time approximation schemes. Ph.D. Dissertation, Texas A&M University (2004)
27. Kann, V.: On the approximability of NP-complete optimization problems. Ph.D. Thesis, Royal Institute of Technology, Sweden (1992)
28. Karmarkar, N., Karp, R.M.: An efficient approximation scheme for the one-dimensional bin packing problem. In: Proc. of FOCS, pp. 312–320 (1982)
29. Marx, D.: Parameterized complexity and approximation algorithms. Comput. J. **51**, 60–78 (2008)
30. Papadimitriou, C., Yannakakis, M.: Optimization, approximation, and complexity classes. J. Comput. Syst. Sci. **43**, 425–440 (1991)
31. Papadimitriou, C., Yannakakis, M.: On limited nondeterminism and the complexity of VC dimension. J. Comput. Syst. Sci. **53**, 161–170 (1996)
32. Schnorr, C.: Optimal algorithms for self-reducible problems. In: Proc. ICALP, pp. 322–337 (1976)
33. Simchi-Levi, D.: New worst-case results for the bin-packing problem. Nav. Res. Logist. **41**, 579–585 (1994)