Parameterized (in)approximability of subset problems

Édouard Bonnet[a,*], Vangelis Th. Paschos[a,1]

[a] *PSL* Research University, Université Paris-Dauphine, LAMSADE, CNRS UMR 7243*

Abstract

We discuss approximability and inapproximability in FPT-time for a large class of *subset problems* where a feasible solution S is a subset of the input data. We introduce the notion of *intersective* approximability that generalizes the one of *safe* approximability introduced in (J. Guo, I. Kanj and S. Kratsch, *Safe approximation and its relation to kernelization*, IPEC 2011) and show strong parameterized inapproximability results for many of the subset problems handled.

Keywords: Approximation, Complexity, Graph, Parameterized algoririthm.
2010 MSC: 68Q15, 68Q17.

1. Introduction

Parameterized approximation aims at bringing together two very active fields of theoretical computer science, polynomial approximation and parameterized computation. We say that a minimization (resp., maximization) problem Π, together with a parameter k, is *parameterized r-approximable*, if there exists an FPT-time algorithm which computes a solution of size at most (resp., at least) rk whenever the input instance has a solution of size at most (resp., at least) k, otherwise, it outputs an arbitrary solution. This line of research was initiated by three independent works [5, 3, 4]. For a very interesting overview of older results, see [7].

*Corresponding author
Email addresses: `edouard.bonnet@lamsade.dauphine.fr` (Édouard Bonnet), `paschos@lamsade.dauphine.fr` (Vangelis Th. Paschos)
[1] Also, Institut Universitaire de France

Our goal in this paper is to study parameterized approximability of subset problems (whose formal definition is given in Section 2), via the introduction of a new approximability framework called *intersective approximation*, which is quite natural when handling such problems.

Intersective approximability generalizes the model of *safe* approximability, introduced earlier by [6]. An approximation is said to be safe, if it produces solutions containing an optimal solution. Safe approximation only captures minimization problems and can be used in order to get strong inapproximability results. For instance, it is shown in [6] that a safe $c \log n$-approximation, for any $c > 0$, for GENERALIZED MIN DOMINATING SET, can be turned into an exact FPT algorithm, contradicting $\mathbf{FPT} \neq \mathbf{W[2]}$.

Intersective approximability relaxes the requirement of (complete) inclusion of an optimal solution into the approximate solution computed, by just asking these two solutions to have a non-empty intersection. This relaxation allows the new model to apply non-trivially to maximization subset problems, too. We use intersective approximability, in order to establish meta-theorems producing as corollaries strong negative results for subset problems.

2. Subset problems

Subset problems can be defined as follows.

Definition 1. *A problem* Π *is called a* subset problem*, if the following conditions hold:*

- *feasible solutions for* Π *are subsets of elements encoding its instances that verify some specific property;*
- Π *is* decomposable*, i.e., for any instance* I *and element* e *of the encoding of* I*, there exists an instance* $I(e)$ *such that if* S *is a solution for* $I(e)$*, then* $S \cup \{e\}$ *is a solution for* I*.*

The existence of one instance encoding for Π satisfying Definition 1 is sufficient for Π to be a subset problem; thus, Definition 1 does not depend on the encoding.

In what follows, we give several examples of graph, set and satisfiability subset problems in order to clarify the notion of decomposability of the second item in Definition 1. We use the standard notation on graphs: $G[S]$ is the subgraph of G induced by S, $N(v)$ is the set of neighbors of v, $N[v] = N(v) \cup \{v\}$, V usually denotes the set of vertices of a graph, E the set of its edges, $n = |V|$ and $m = |E|$.

Let a graph be encoded by the set of its vertices and the set of its edges. An *independent set* is a vertex-subset verifying the stability property (the induced subgraph contains no edge). MAX INDEPENDENT SET is decomposable with $G(v) = G[V \setminus N[v]]$. Indeed, any independent set in G containing v, is an independent set in $G[V \setminus N[v]]$ combined with the vertex v. A *clique* is a vertex-subset verifying the property of containing pairwise adjacent vertices. MAX CLIQUE is decomposable with $G(v) = G[N(v)]$. Indeed, any clique containing v has all its other vertices in $N(v)$. A *vertex cover* is a vertex-subset which covers all the edges. MIN VERTEX COVER is decomposable with $G(v) = G[V \setminus \{v\}]$. A *generalized dominating set* is a vertex-subset whose members dominate an imposed subset of vertices $V' \subseteq V$. GENERALIZED MIN DOMINATING SET aims at finding a minimum-size generalized dominating set, given a graph G and a subset $V' \subseteq V$. MIN DOMINATING SET is a special case of GENERALIZED MIN DOMINATING SET with $V' = V$. GENERALIZED MIN DOMINATING SET is decomposable with $(G, V')(v) = (G[V \setminus \{v\}], V' \setminus N[v])$. A *feedback vertex set* is a vertex-subset S such that $G[V \setminus S]$ is a forest. MIN FEEDBACK VERTEX SET is decomposable with $G(v) = G[V \setminus \{v\}]$.

Let a set system be encoded by a ground set X and a collection $\mathcal{S}$ of its subsets. A *set cover* is a subcollection of $\mathcal{S}$ that covers C. MIN SET COVER is decomposable with $(\mathcal{S}, X)(S) = ((\mathcal{S} \setminus \{S\})_{|X \setminus S}, X \setminus S)$, where $A_{|B}$ is the projection to B of all the subsets in A.

Let a CNF formula ϕ be encoded by its variables X and its clauses C. SAT is decomposable with $\phi(x) = C[x \leftarrow \top]$. Analogous formulations make that MAX SAT or MIN SAT problems are subset problems. For the same reason, SAT-k, asking for determining a truth assignment setting at most k variables to *true* and its optimization counterpart MAX SAT-k are subset problems.

On the other hand, all problems are not subset problems. For instance, partition problems such as MIN VERTEX COLORING, MIN DOMATIC NUMBER, MAX ACHROMATIC NUMBER, or BIN PACKING are *not* subset problems.

In what follows, we will focus mainly on *optimization* subset problems.

3. Intersective approximability of subset problems

As we have already mentioned in Section 1, intersective approximability extends the notion of safe approximability of [6], by allowing the approximate solutions computed not to thoroughly contain an optimal solution (for the case of minimization problems) but only to have a non-empty intersection with some optimal solution.

Definition 2. *A ρ-approximation algorithm $\mathtt{A}$ is said to be* intersective *for a problem Π if, when running on any instance I of Π, it computes a ρ-approximate solution $\mathtt{A}(I)$ and there exists an optimal solution S_0 of I such that $\mathtt{A}(I) \cap S_0 \neq \emptyset$.*

Note that a safe approximation is a special case of an intersective approximation. From Definition 2 and since the intersective model does not require that the approximate solution contains an optimal solution, intersective approximation can also fit maximization problems. Therefore, the model applies to any optimization subset problem.

In what follows, we prove that intersective approximation in FPT time is very unlikely for $\mathbf{W}[\cdot]$-hard subset problems, since such an approximation can be transformed into an exact FPT algorithm. However, as we will see, there is an important difference between minimization and maximization problems since, for the former, this transformation can be done only if intersective FPT approximation ratio is under a certain approximation level, while, for the latter, such transformation is independent on the level of the ratio.

We first prove the following more general theorem where, given an instance I of a subset problem Π, we denote by k the optimal value of I.

Theorem 3. *Let* Π *be an optimization subset problem. Then:*

- *if* Π *is a minimization problem and admits an intersective* r*-approximation computed in time* $O(f(n,k))$*, for some* $r > 1$ *and some positive increasing function* f*, then* Π *can be optimally solved in time* $O((rk)^k f(n,k))$*;*
- *if* Π *is a maximization problem, any intersective approximation computed in time* $O(f(n,k))$*, for some positive increasing function* f *can be transformed into an exact algorithm running in time* $O(k^k f(n,k))$*.*

Proof. Consider some minimization problem Π, an intersective FPT approximation algorithm $\mathtt{A}$ for Π achieving approximation ratio r and let I be any instance of Π. Compute an intersective approximation $S = \mathtt{A}(I) = \{e_1, \ldots, e_{|S|}\}$ for I. If $|S| > rk$, then answer that I is a NO-instance. Otherwise, branch on the at most rk instances $I(e_1), \ldots, I(e_{|S|})$ (since Π is decomposable, all these instances are well-defined). For all these instances, compute an r-approximation and keep the recursion on. When k elements have been taken in the solution, stop the recursion.

We claim that the best solution found at a leaf of the branching tree is an optimal solution. Indeed, starting from the root one can, by definition of intersective approximation, move to a child which has taken an element e contained in an optimal solution.

The branching tree has depth k since, at each step, one element is added in the solution and arity bounded by rk. Hence, the number of its nodes is bounded by $2(rk)^k$. On each node, some $O(f(n,k))$ computation is done. So, the overall complexity is $O((rk)^k f(n,k))$.

We now handle maximization problems. Consider some maximization problem Π, an intersective approximation algorithm $\mathtt{A}$ for Π (achieving any approximation ratio) and let I be any instance of Π. Compute an intersective approximation $S = \mathtt{A}(I)$ for I. If $|S| \geqslant k$, answer YES and output this solution. Otherwise $|S| < k$ and the exact branching algorithm of the previous paragraph runs in time $O(k^k f(n,k))$. If one of the leaves of the branching tree contains a feasible solution, answer YES and output this solution. Otherwise, anwser NO.

Theorem 3 has the following almost immediate but important corollary.

Theorem 4. *Let Π be a subset optimization problem. Then:*

- *if Π is a minimization problem and admits an FPT intersective $(g(k)\log n)$-approximation for some function g, then Π admits an exact FPT algorithm;*
- *if Π is a maximization problem and admits an FPT intersective approximation, then Π admits an exact FPT algorithm.*

Proof. For minimization problems just observe that, when $r = g(k)\log n$, the number of nodes in the branching tree is bounded by $2(kg(k))^k(\log n)^k$ and, on each node, some FPT computation is done, bounded by, say, $f(k)p(n)$. So, the overall complexity is $2(kg(k))^k f(k)(\log n)^k p(n)$, which is FPT, considering that $(\log n)^k$ is FPT with respect to k [10].

For maximization problems, the proof comes directly from Theorem 3 and it can be easily seen that no specific approximation guarantee is required for them.

Remark 1. The result of Theorem 4 for the case of minimization problems works, in fact, even if we consider approximation ratios $O(g(k)(\log n)^{h(k)})$ for *any* (increasing) functions h and g. Indeed, $O((g(k)(\log n)^{h(k)}k)^k f(k))$ (where f is the complexity of the FPT intersective algorithm) is $O(F(k)p(n))$, for some function F and polynomial p [10].

Based upon Theorem 4, the following holds for the intersective FPT approximability of $\mathbf{W}[\cdot]$-hard problems.

Corollary 5. *Unless the $\boldsymbol{W}$-hierarchy collapses at some level:*

- *no FPT intersective $(g(k)\log n)$-approximation exists for $\mathbf{W}[\cdot]$-hard minimization problems, for any positive increasing function g;*

- *no FPT intersective r-approximation (for any r) exists for* $\mathbf{W}[\cdot]$*-hard maximization problems.*

In particular:

- *unless* $\boldsymbol{FPT} = \boldsymbol{W[2]}$*, no FPT intersective* $(g(k)\log n)$*-approximation exists for either* MIN SET COVER*, or* GENERALIZED MIN DOMINATING SET*, for any positive increasing function g;*
- *unless* $\boldsymbol{FPT} = \boldsymbol{W[1]}$*, no FPT intersective approximation exists either for* MAX INDEPENDENT SET*, or for* MAX CLIQUE*.*

Note that the negative result for MIN SET COVER above, transfers also to MIN DOMINATING SET thanks to the classical approximability-preserving reduction from MIN SET COVER to MIN DOMINATING SET [9] that is also parameter-preserving.

The proof of Theorem 4 gives also some hints for obtaining FPT algorithms for intersectively approximable problems. For instance:

- for MIN VERTEX COVER, the classical polynomial 2-approximation maximal matching algorithm is an intersective approximation algorithm (while not always a safe one) which, by Theorem 4 derives an FPT algorithm running in time $O^*((2k)^k)$;
- for MAX MINIMAL VERTEX COVER (it consists of determining a maximum-cardinality vertex cover that is minimal for inclusion; an optimal solution for this problem is the complement of a minimum independent dominating set), all the (polynomial) algorithms proposed in [1], compute intersective approximations (in fact, any algorithm for this problem is either optimal or intersective); so, the application of the algorithm of Theorem 3 derives an FPT algorithm running in time $O^*(k^k)$;
- there exists a 2-approximation for MAX CUT which produces solutions of size greater than $m/2$; hence, this approximation is obviously intersective and application of Theorem 3 again derives an FPT algorithm running in time $O^*(k^k)$.

Of course, the inclusion of all of the three problems above in **FPT** is already known and through faster FPT algorithms, but Theorem 4 could be helpful for other problems of still unknown status.

Although very interesting as approximation concept, safe approximation is quite rare and very restrictive, since inclusion of some optimum in an approximate solution is a very strong requirement. Indeed, no safe algorithm is known for problems other than MIN VERTEX COVER (this algorithm is the immediate algorithm for MIN VERTEX COVER derived by application of Nemhauser-Trotter's Theorem [8] for MIN VERTEX COVER) and a few restrictive versions of some other minimization subset problems. Another drawback of the safe model, which intersective approximation handles, is that it cannot extend to maximization problems. For such problems, safe approximation (the computed solution is contained in an optimal solution) can be used k times as a polynomial time oracle to guess one by one each element of an optimal solution; so safe approximation is not likely to happen for hard maximization problems. Furthermore, intersective approximation seems to be more realistic because it captures the behavior of approximations algorithms that in general build solutions with both elements inside and elements outside an optimal solution.

Let us note that the so-called *fixed-cardinality graph problems* are subset problems, too. Such problems are defined on some graph $G(V, E)$ with two integers k and p. Feasible solutions are subsets $V' \subseteq V$ of size exactly k. The value of their solutions is a linear combination of sizes of edge subsets and the objective is to determine whether there exists a solution of value at least, or at most p. Notable representatives of such problems are MAX k-VERTEX COVER (where one looks for a set of k vertices that covers a maximum number of edges), GENERALIZED k-DENSEST SUBGRAPH (given a subset V' of V, one looks for a superset of V' with k vertices inducing a subgraph of G with a maximum number of edges) and its minimization version GENERALIZED k-LIGHTEST SUBGRAPH, GENERALIZED MAX $(k, n-k)$-CUT (given a set V' of vertices, one looks for a superset S of V' with k vertices with a maximum number of edges between S and $V \setminus S$) and its minimization version GENERALIZED MIN $(k, n-k)$-CUT, MAX

k-SET COVER (here one looks for a family of k subsets that covers a maximum number of elements), etc. These problems are known to be **W[1]**-hard with respect to k [2].

An intersective approximation for fixed-cardinality problems implies that k' elements from the solution, $0 < k' \leqslant k$, are common to both the optimum and the approximate solution. Then, Theorem 4 derives the following.

Corollary 6. *Unless* $\mathbf{W}[\mathbf{1}] = \boldsymbol{FPT}$*, no intersective approximation algorithm can exist for any of the problems* MAX k-VERTEX COVER*,* GENERALIZED k-DENSEST SUBGRAPH*,* GENERALIZED k-LIGHTEST SUBGRAPH*,* GENERALIZED MAX $(k, n-k)$-CUT*,* GENERALIZED MIN $(k, n-k)$-CUT*,* MAX k-SET COVER*. Also, unless* $\mathbf{W}[\mathbf{2}] = \boldsymbol{FPT}$*, no intersective FPT approximation algorithm can exist for* MAX k-SET COVER*.*

4. Final remarks

Intersective approximability, importantly relaxes and generalizes the safe approximability of [6] since (i) it is more natural and reflects the realistic behavior of an approximation algorithm and (ii) it encompasses maximization problems. Also, while producing strong negative results, intersective approximability may also produce positive approximation results.

Like safe approximability and despite the narrowness of both notions, intersective approximability has the merit to give new insights in the field of parameterized approximation that is in its beginnings and needs several precisions and hypotheses for stabilizing its formal framework.

Finally, let us note that intersective approximability can be extended to several problems that are not subset problems *per se*. We just sketch such an extension to coloring problems. A solution for a k-coloring can be seen as k sets $S_1, \ldots, S_k$ where S_i is the set of vertices (or edges) receiving color i. A ρ-intersective approximation to a k-coloring problem can be defined as an h-coloring $S'_1, \ldots, S'_h$ such that there exists an optimal solution $S_1, \ldots, S_k$ with $k \geqslant h/\rho$ and two integers i, j satisfying $S_i = S'_j$. Under this extended definition of intersective approximability, the following can be proved in a way similar to that of Theorem 4.

Corollary 7. *If a k-coloring problem* Π *has an FPT intersective* $(c \log n)$*-approximation (as extended just above) for some constant* $c > 0$*, then* Π *admits an exact FPT algorithm.*

Acknowledgement. The very useful and pertinent comments and suggestions of an anonymous referee are gratefully acknowledged.

References

[1] N. Boria, F. Della Croce, V. Paschos, On the MAX MIN VERTEX COVER problem, in: Proc. Workshop on Approximation and Online Algorithms, WAOA'13, Lecture Notes in Computer Science, Springer-Verlag, 2013.

[2] L. Cai, Parameter complexity of cardinality constrained optimization problems, The Computer Journal 51 (2008) 102–121.

[3] L. Cai, X. Huang, Fixed-parameter approximation: conceptual framework and approximability results, in: H. L. Bodlaender, M. A. Langston (Eds.), Proc. International Workshop on Parameterized and Exact Computation, IWPEC'06, Vol. 4169 of Lecture Notes in Computer Science, Springer-Verlag, 2006, pp. 96–108.

[4] Y. Chen, M. Grohe, M. Grüber, On parameterized approximability, in: H. L. Bodlaender, M. A. Langston (Eds.), Proc. International Workshop on Parameterized and Exact Computation, IWPEC'06, Vol. 4169 of Lecture Notes in Computer Science, Springer-Verlag, 2006, pp. 109–120.

[5] R. G. Downey, M. R. Fellows, C. McCartin, Parameterized approximation problems, in: H. L. Bodlaender, M. A. Langston (Eds.), Proc. International Workshop on Parameterized and Exact Computation, IWPEC'06, Vol. 4169 of Lecture Notes in Computer Science, Springer-Verlag, 2006, pp. 121–129.

[6] J. Guo, I. Kanj, S. Kratsch, Safe approximation and its relation to kernelization, in: D. Marx, P. Rossmanith (Eds.), Proc. International Workshop on Parameterized and Exact Computation, IPEC'11, Vol. 7112 of Lecture Notes in Computer Science, Springer-Verlag, 2011, pp. 169–180.

[7] D. Marx, Parameterized complexity and approximation algorithms, The Computer Journal 51 (1) (2008) 60–78.

[8] G. L. Nemhauser, L. E. Trotter, Vertex packings: structural properties and algorithms, Math. Programming 8 (1975) 232–248.

[9] A. Paz, S. Moran, Non deterministic polynomial optimization problems and their approximations, Theoret. Comput. Sci. 15 (1981) 251–277.

[10] C. Sloper, J. A. Telle, An overview of techniques for designing parameterized algorithms, The Comput. J. 51 (1) (2008) 122–136.