Foundations of Software Technology and Theoretical Computer Science (Bangalore) 2008.
Editors: R. Hariharan, M. Mukund, V. Vinay; pp 1-12

Implicit Branching and Parameterized Partial Cover Problems (Extended Abstarct)

Omid Amini[1], Fedor V. Fomin[2] and Saket Saurabh[2]

[1] Max-Planck-Institut für Informatik
Omid.Amini@mpi-inf.mpg.de

[2] Department of Informatics, University of Bergen,
N-5020 Bergen, Norway.
{fedor.fomin|saket.saurabh}@ii.uib.no

ABSTRACT. Covering problems are fundamental classical problems in optimization, computer science and complexity theory. Typically an input to these problems is a family of sets over a finite universe and the goal is to cover the elements of the universe with as few sets of the family as possible. The variations of covering problems include well known problems like SET COVER, VERTEX COVER, DOMINATING SET and FACILITY LOCATION to name a few. Recently there has been a lot of study on partial covering problems, a natural generalization of covering problems. Here, the goal is not to cover all the elements but to cover the specified number of elements with the minimum number of sets.
In this paper we study partial covering problems in graphs in the realm of parameterized complexity. Classical (non-partial) version of all these problems have been intensively studied in planar graphs and in graphs excluding a fixed graph H as a minor. However, the techniques developed for parameterized version of non-partial covering problems cannot be applied directly to their partial counterparts. The approach we use, to show that various partial covering problems are fixed parameter tractable on planar graphs, graphs of bounded local treewidth and graph excluding some graph as a minor, is quite different from previously known techniques. The main idea behind our approach is the concept of *implicit branching*. We find implicit branching technique to be interesting on its own and believe that it can be used for some other problems.

1 Introduction

Covering problems are basic, fundamental and widely studied problems in algorithms and combinatorial optimizations. In general these problems ask for selecting a least sized family of sets to cover all the elements. One of the prominent covering problem is the classical SET COVER problem. SET COVER problem consists of a family $\mathscr{F}$ of sets over a universe $\mathscr{U}$ and the goal is to cover this universe $\mathscr{U}$ with the least number of sets from $\mathscr{F}$. Other classical problems in the framework of covering include well known problems like VERTEX COVER, DOMINATING SET, FACILITY LOCATION, k-MEDIAN, k-CENTER problems, on which hundreds of papers have been written.

In this paper we study the generalization of these problems to the *partial covering problems*, where the objective is not to cover all the elements but to cover the pre-specified number of elements with minimum number of objects. More precisely, in the partial covering

FSTTCS 2008
IARCS Annual Conference on
Foundations of Software Technology and Theoretical Computer Science
http://drops.dagstuhl.de/opus/volltexte/2008/1736

problem, for a given integer $t \geq 0$, we want to cover at least t elements rather than covering all the elements. For an example, in PARTIAL VERTEX COVER (PVC), the goal is to cover at least t edges with minimum number of vertices not all the edges while in PARTIAL SET COVER (PSC) the goal is to cover at least t elements of $\mathscr{U}$ with minimum number of sets from $\mathscr{F}$. Other problems are defined similarly. Partial covering problems are studied intensively not only because they generalize classical covering problems, but also because of many real life applications. They have received a lot of attention recently, see, for example [4, 5, 7, 18].

While different variations of PSC were studied intensively and many approximation algorithm and non-approximability results exist in the literature, only few things are known on their parameterized complexity. In this paper we fill this gap by initiating parameterized algorithmic study of these problems on structural graphs like planar graphs, graphs of bounded genus and graphs of bounded local treewidth. In parameterized algorithms, for decision problems with input size n, and a parameter k, the goal is to design an algorithm with runtime $\tau(k) \cdot n^{O(1)}$, where τ is a function of k alone. Problems having such an algorithm are said to be fixed parameter tractable (FPT). There is also a theory of hardness using which one can identify parameterized problems that are not amenable to such algorithms. This hardness hierarchy is represented by $W[i]$ for $i \geq 1$. For an introduction and more recent developments see the books [15, 17, 21]. In this paper, we always parameterize a problem by the size of the partial set cover, i.e. all our algorithms for finding a partial set cover of size k that cover at least t sets with input of size n are of running time $\tau(k) \cdot n^{O(1)}$.

Our Approach and Results. The main ideas behind our approach can be illustrated by planar instances of PARTIAL VERTEX COVER and PARTIAL DOMINATING SET. Let a planar graph $G = (V, E)$ on n vertices, and integers k, t, be an instance of PARTIAL VERTEX COVER. Let S be the set vertices in G of degree at least t/k. If S is sufficiently big, say, its size is at least $4k$, then (by the Four color theorem), the subgraph of G induced on S contains an independent set of size at least k. This yields that there are k vertices of S that are pairwise non-adjacent in G, and since each of these vertices covers at least t/k edges, we have that in total they cover at least t edges. If the size of S is less than $4k$, we apply *explicit branching*. The crucial observation here is that if G has a partial vertex cover of size at most k, then this cover must contain at least one vertex of S. Thus by making a guess on the vertices $x \in S$, whether x is in a partial vertex cover of size at most k, we can guarantee, that if the problem has a solution, then at least one of our guesses is correct. For each of the guesses x, we create a new subproblem for PARTIAL VERTEX COVER, where the input is the subgraph of G induced on $V \setminus \{x\}$ and we are asked to cover $t - deg(x)$ edges by $k - 1$ vertices, where $deg(x)$ is the number of edges adjacent to x. The number of subproblems we generate in this way is at most $4k$, and we call the procedure recursively on each subproblem. The depth of the recursion is at most k, and the number of recursive calls at each steps is at most $4k$, resulting in total running time $(4k)^k \cdot n^{O(1)}$. Actually, in our arguments we used planarity only to conclude that a graph has large independent set. Definitely, this approach is valid for many other graph classes with large independent sets, like bipartite graphs, degenerate graphs and graphs excluding some graph as a minor. (We provide detailed consequences of this approach in Section 5.)

The main drawback of explicit branching is that we cannot use it for many partial covering problems, in particular for PARTIAL DOMINATING SET. Even for planar graphs, the existence of a large independent set of vertices of degree at least t/k does not imply that k vertices can dominate at least t vertices. To overcome this obstacle, we do the following. We start as in the case of PARTIAL VERTEX COVER, by selecting the set S consisting of vertices of degree at least t/k. If there are more than k vertices in S which are at distance at least three from each other, we have the solution. Otherwise, we know that at least one vertex from S should be in a partial dominating set but we cannot use explicit branching by trying all vertices of S because the size of S can be too large. However, we show in this case that the graph formed by S and their neighbors is of small diameter, and thus, by well known properties of planar graphs, has small treewidth. (Very loosely small here means bounded by some function of k.) In this case we apply *implicit branching*, which means that we do not create a new subproblem for every vertex of S, but instead for every i, $1 \leq i \leq k$, we make a guess that exactly i vertices of S are in a partial dominating set. Thus we branch on k cases and try to solve the problem recursively. We formulate these ideas in details in Sections 3.1 and 3.2 and show how it is sufficient to just know the size of an intersection of an optimal partial dominating set with S rather than the actual intersection itself to solve the problem.

Again, the only property of planar graphs we mentioned here was the property that non-existence of a large set of pairwise remote vertices in a graphs yields a small treewidth. But this property can be shown not only for planar graphs, but more generally for graphs of bounded local treewidth, the class of graphs containing planar graphs, graphs of bounded genus, graphs of bounded vertex degree, and graphs excluding an apex graph as a minor. With more additional work we show that similar ideas can be used to prove that much more general problem, namely a weighted version of the PARTIAL (k, r, t)-CENTER problem, where the goal is to cover at least t elements by balls of radius r centered around at most k vertices, is FPT on graphs of bounded local treewidth. This result can be found in Section 3.2. This is mainly of theoretical interest because the running time of the algorithm is $2^{k^{O(k)}} \cdot n^{O(1)}$. Such a huge running time is due to the bounds on the treewidth of a graph, which is used in implicit branching. Due to the generality of the result for graphs with bounded local treewidth, we do not see any reasonable way of overcoming this problem. But because of numerous application, we find it is worth to search for faster practical algorithms on subclasses of graphs of bounded local treewidth, in particular on planar graphs. As a step in this direction, we obtain much better combinatorial bounds on the treewidth of planar graphs in implicit branching, which results in algorithms of running time $2^{O(k)} \cdot n^{O(1)}$ on planar graphs. The combinatorial arguments used for the exponential speedup (Section 3.3) are interesting on their own. In Section 4, we show that the PARTIAL (k, r, t)-CENTER problem is FPT on graphs excluding a fixed graph as a minor. The proof of this result is based on the decompositions theorem of Robertson and Seymour from Graph Minors [24]. The algorithm is quite involved, it uses two levels of dynamic programming and two levels of implicit branching, and can be seen as a non-trivial extension of the algorithm of Demaine et al. [10] for classical covering problems to partial covering problems.

Finally, let us remark that while DOMINATING SET is FPT on d-degenerated graphs [3], there are strong arguments that our results cannot be extended to this class of sparse graphs. This is because Golovach and Villanger [19] showed that PARTIAL DOMINATING SET is W[1]-hard on d-degenerated graphs.

2 Preliminaries

Let $G = (V, E)$ be an undirected graph where V (or $V(G)$) is the set of vertices and E (or $E(G)$) is the set of edges. We denote the number of vertices by n and number of edges by m. For a subset $V' \subseteq V$, by $G[V']$ we mean the subgraph of G induced by V'. By $N(u)$ we denote (open) neighborhood of u that is set of all vertices adjacent to u and by $N[u] = N(u) \cup \{u\}$. Similarly, for a subset $D \subseteq V$, we define $N[D] = \cup_{v \in D} N[v]$. The *distance* $d_G(u, v)$ between two vertices u and v of G is the length of the shortest path in G from u to v. The *diameter* of a graph G, denoted by $diam(G)$, is defined to be the maximum length of a shortest path between any pair of vertices of $V(G)$. By an abuse of notation, we define diameter of a graph as the maximum of the diameters of its connected components. For $r \geq 0$, the *r-neighborhood* of a vertex $v \in V$ is defined as $N^r_G[v] = \{u \mid d_G(v, u) \leq r\}$. We also let $B_r(v) = N^r_G[v]$ and call it a ball of radius r around v. Similarly $B_r(A) = \cup_{v \in A} N^r_G[v]$ for $A \subseteq V(G)$. Given a weight function $w : V \rightarrow \mathbb{R}$ and $A \subseteq V(G)$, $w(B_r(A)) = \sum_{u \in B_r(A)} w(u)$.

Given an edge $e = (u, v)$ of a graph G, the graph G/e is obtained by contracting the edge (u, v) that is we get G/e by identifying the vertices u and v and removing all the loops and duplicate edges. A *minor* of a graph G is a graph H that can be obtained from a subgraph of G by contracting edges. A graph class $\mathcal{C}$ is *minor closed* if any minor of any graph in $\mathcal{C}$ is also an element of $\mathcal{C}$. A minor closed graph class $\mathcal{C}$ is *H-minor-free* or simply *H-free* if $H \notin \mathcal{C}$.

We use the standard definitions of *treewidth* and *tree decomposition*. We use $\mathbf{tw}(G)$ to denote the treewidth of a graph G. The definition of treewidth can be generalized to take into account the local properties of G and is called *local treewidth* [16, 20]. The *local treewidth* of a graph G is the function $\mathbf{ltw}^G : \mathbb{N} \rightarrow \mathbb{N}$ that associates with every integer $r \in \mathbb{N}$ the maximum treewidth of an r-neighborhood of vertices of G, i.e., $\mathbf{ltw}^G(r) = \max_{v \in V(G)} \{\mathbf{tw}(G[N^r_G[v]])\}$. A graph class $\mathscr{G}$ has *bounded local treewidth*, if there exists a function $f : \mathbb{N} \rightarrow \mathbb{N}$ such that for each graph $G \in \mathscr{G}$, and for each integer $r \in \mathbb{N}$, we have $\mathbf{ltw}^G(r) \leq f(r)$. The class $\mathscr{G}$ has *linear local treewidth*, if in addition the function f can be chosen to be linear, that is $f(r) = cr$ where $c \in \mathbb{R}$ is a constant. For a given function $f : \mathbb{N} \rightarrow \mathbb{N}$, $\mathscr{G}_f$ is the class of all graphs G of local tree-width at most f, that is, $\mathbf{ltw}^G(r) \leq f(r)$ for every $r \in \mathbb{N}$. A well known graph classes which are known to have bounded local treewidth are planar graphs, graphs of bounded genus, and graphs of bounded maximum degree. By a result of Robertson and Seymour [22], $f(r)$ can be chosen as $3r$ for planar graphs. Similarly Eppstein [16] showed that $f(r)$ can be chosen as $c_g g(\Sigma) r$ for graphs embeddable in a surface Σ, where $g(\Sigma)$ is the genus of the surface Σ and c_g is a constant depending only on the genus of the surface. Demaine and Hajiaghayi [11] extended this result and showed that the concept of bounded local treewidth and linear local treewidth are the same for minor closed families of graphs.

3 FPT Algorithms for Weighted Partial-(k, r, t)-Center Problem

3.1 Developing a Step by Step Procedure

In this section we give a template of a generic algorithm for partial covering problems arising on graphs. We use this later to show that partial covering problems arising on graphs are fixed parameter tractable in graphs of bounded local treewidth. We formulate the template through the following problem.

WEIGHTED PARTIAL-(k,r,t)-CENTER (WP-(k,r,t)-C): Given an undirected graph $G = (V, E)$, with weight function $w : V \rightarrow \{0,1\}$ and integers k, r and t. The problems asks whether there exists a $C \subseteq V$ of size at most k (k centers), such that $w(B_r(C)) \geq t$. Here k and r are the parameters.

When all the vertices have weight 1 this is a PARTIAL-(k,r,t)-CENTER (P-(k,r,t)-C) problem, and for $r = 1$ and $w(v) = 1$ for all $v \in V$ this is PARTIAL DOMINATING SET problem. To formulate PSC problem as WP-(k,r,t)-C problem, we consider the incidence bipartite graph associated with the instance of PSC problem and give weights 1 to the vertices associated with elements and 0 to the vertices associated with sets. Since PVC can be transformed to PSC problem, WP-(k,r,t)-C also generalizes PVC. One defines PARTIAL HITTING SET similarly.

Unlike the non-partial and non-weighted version of WP-(k,r,t)-C problem, the first major challenge in partial covering problems is: which t elements we choose to cover? To find an answer to this we define the following set S and the corresponding graph $\mathcal{G}$, which forms the first step of the algorithm:

(T1) Define $S = \{v \mid v \in V,\ w(B_r(v)) \geq t/k\}$ and $\mathcal{G} = \bigcup_{v \in S} G[B_r(v)]$.

The basic observation is that if there exists a subset $C \subseteq V$ of size at most k such that $w(B_r(C,r)) \geq t$ then $C \cap S \neq \emptyset$. Given the graph $\mathcal{G}$ our second idea is to:

(T2) Check the diameter of $\mathcal{G}$, and if $diam(\mathcal{G})$ is large then we argue that this is a YES instance by providing a subset C of size at most k and $w(B_r(C)) \geq t$.

Now when the $diam(\mathcal{G})$ is small, the treewidth of the graph $\mathcal{G}$ is bounded and hence dynamic programming over graphs with bounded treewidth can be used. But we still do not know whether we can find the desired C among the vertices of $\mathcal{G}$. Hence even if we find out that there is no $X \subseteq S$ such that $|X| \leq k$ and $w(B_r(X)) \geq t$, we can not guarantee that this is a NO instance of the problem. So to overcome this difficulty we resort to an implicit branching by using the earlier observation that there is no desired C whose intersection with S is empty. Before we go further, given a vertex set S and $\mathcal{G}$ (as defined above), we define $\mu(S,i) = \max_{A \subseteq S, |A|=i}\{w(B_r(A))\}$.

(T3) Using dynamic programming over graphs with bounded treewidth, compute $\mu(S,i)$ for $\mathcal{G}$ for $1 \leq i \leq k$ as well as a subset $A_i \subseteq S$ such that $w(B_r(A_i)) = \mu(S,i)$.

(T4) Now we make k recursive calls to reduce the size of k on the fact that if there exists a C then its intersection with S is between $1 \leq i \leq k$. Now we reduce the parameters t to $t - \mu(S,i)$ and k to $k - i$ and try to solve the problem recursively.

In the recursive steps, we follow the above steps and either we move forward to a larger $\mathcal{G}$ or we get a desired solution for the problem. More precisely, suppose we are at the i^{th} step of recursion then we do as follows:

(T5) Enlarge $\mathcal{G}$ by adding some new vertices to S. Let S_i be the set of new vertices added to S that is those set of vertices which are not in S and $w(B_r(v)) \geq t/k$ where t and k are the current parameters obtained after reductions done in previous recursive calls.

(T6) Either we bound the diameter and hence the treewidth of $\mathcal{G}$. Else, we select a set C of at most k vertices such that $w(B_r(C)) \geq t$ and C respects the guesses made on the number of vertices we need to select from S_j, $1 \leq j \leq i-1$. That is, the possible number of vertices in $C \cap S_j$.

This completes the framework in which we will be working. In the next Section we prove that WP-(k,r,t)-C Problem is FPT in graphs with bounded local treewidth by proving the necessary technical lemmas needed for this generic algorithm to work.

3.2 An Algorithm for WP-(k,r,t)-C in Graphs of Bounded Local Treewidth

We first give an upper bound on the treewidth of $\mathcal{G}$, the graphs we obtained in the recursive calls which is crucial for analysis of the algorithm.

LEMMA 1. *Let G be a graph on n vertices and m edges and H be an induced subgraph of G such that the diameter of each of the connected components of H is at most ℓ. Let C be a subset of $V(H)$ of size at most k and A be a subset of $V(G)$. Then there exists a function $g(k,r,\ell)$ such that if $diam(G[B_r(A) \cup H]) > g(k,r,\ell)$, then there is a subset $T \subseteq A$ such that* **(a)** *$|T| \geq k$;* **(b)** *for all $u,v \in T, d_G(u,v) \geq 2r+1$; and* **(c)** *for all $u \in T$ and for all $v \in C$, $d_G(u,v) \geq 2r+1$. In particular, one can take $g(k,r,\ell) = (6r+2)2k\ell$ and find the desired set T in $O(m+n)$ time.*

PROOF. Since C is a subset of size at most k, we have that it intersects at most k connected components of H. Let these connected components be $H_1, \ldots, H_r$, where $r \leq k$. We contract each of these connected components to a vertex and obtain a new graph G'. Let the contractions of $H_1, \ldots, H_r$ correspond to vertices $v_{H_1}, \cdots, v_{H_r}$ in our new graph G' and this set of vertices be called X. For a vertex $v \in V(G)$, we define its image, $im(v)$, in G' as v_{H_i} if it is in H_i for $1 \leq i \leq r$ and v otherwise. For a subset $W \subseteq V$, its image $im(W)$ in $V(G')$, is defined as the set $\{im(v) \mid v \in W\}$.

For any subset $W \subseteq V(G)$, we claim that $diam(G'[im(W) \cup X]) \geq diam(G[W \cup H])/\ell$ (let us remind that we define the diameter of the graph as the maximum diameter of its connected components).

To prove the claim we observe that a path P' in $G'[im(W) \cup X]$ can be lifted to a path P in $G[W \cup H]$ by replacing every vertex in X on path P' by local paths in each connected component H_j of H. As the diameter of each H_j is bounded by ℓ, in this way, the length of a path can only be increased by at most a constant multiplicative factor ℓ. This gives us $diam(G[W \cup H]) \leq \ell \cdot\ diam(G'[im(W) \cup X])$, which completes the proof of the claim.

To finish the proof of the lemma we proceed as follows: We apply the above claim to the subset $W = B_r(A)$. Since $diam(G[B_r(A) \cup H]) > g(k,r,\ell) = (6r+2)2k\ell$, we have that

$$diam(G'[im(B_r(A)) \cup X]) \geq \frac{diam(G[B_r(A) \cup H])}{\ell} > \frac{g(k,r,\ell)}{\ell} = 2(6r+2)k.$$

Thus there is a connected component $\mathfrak{C}$ of $G'[im(B_r(A)) \cup X]$ of diameter more than $2(6r+2)k$. Let $im(v_1), \ldots, im(v_\kappa)$, $\kappa \leq k$, be the image of vertices of C in this component. Observe that $im(A) \cup \{im(v_1), \ldots, im(v_\kappa)\}$ form an r-center in $\mathfrak{C}$. Since the diameter of this component is at least $2(6r+2)k$, we can find a subset $Y \subseteq im(A) \cup \{im(v_1), \ldots, im(v_\kappa)\}$ of size at least $2k$ such that for any two vertices $u,v \in Y$, $d_{G'}(u,v) \geq 4r+1$. To see this, let us assume that $P = u_0u_1u_2 \cdots u_q$, $q \geq 2(6r+2)k$, is a path which realizes this diameter. Let $V_i \subseteq V(\mathfrak{C})$ be the subset of vertices of distance exactly i from u_0. Since $im(A) \cup \{im(v_1), \ldots, im(v_\kappa)\}$ forms an r-center, its intersection with $\bigcup_{j=i}^{i+2r} V_i$, $1 \leq i \leq q-2r$, is non-empty. Now one can form Y by selecting a vertex of $im(A) \cup \{im(v_1), \ldots, im(v_\kappa)\}$ from $\cup_{i=0}^{2r} V_i$ and then alternately not selecting any vertex from next $4r+1$ V_i's and then selecting a vertex of $im(A) \cup \{im(v_1), \ldots, im(v_\kappa)\}$ from one of the next $2r+1$ blocks of V_i's, and so on.

We put $Z = Y \cap \{im(v_1), \ldots, im(v_\kappa)\}$. Let us remark that, for each vertex v in $\{im(v_1), \ldots, im(v_\kappa)\} \setminus Z$ there is at most one vertex v in $Y \setminus Z$ such that $d_{G'}(u,v) \leq 2r$. Otherwise it will violate the condition that the distance between any two vertices from Y is at least $4r+1$ in G'. We construct the set T' by removing all vertices from $Y \setminus Z$ which are at distance at most $2r$ from $\{im(v_1), \ldots, im(v_\kappa)\} \setminus Z$. The subset $T' \subseteq im(A)$ satisfies the following conditions: **(a)** $|T'| \geq k$; **(b)** for all $u, v \in T'$, $d_{G'}(u,v) \geq 2r+1$; and **(c)** for all $u \in T'$ and for all $im(v_j)$, $1 \leq j \leq k$, $d_{G'}(u, im(v_j)) \geq 2r+1$.

Lifting the subset T' to G one gets a T (by taking inverse image of vertices in T') of the desired kind. ∎

Another essential part of our algorithm is dynamic programming on graphs with bounded treewidth which will be used in **(T6)**. To do so we use a variation of the Theorem 4.1 of [9].

THEOREM 2. [⋆]* *Let G be a graph on n vertices, given with (a) a weight function $w : V \to \{0,1\}$, (b) a tree decomposition of width $\leq \mathbf{b}$, and (c) positive integers k, r and t. Furthermore let $S_1, \cdots, S_p$ be disjoint subsets of $V(G)$ with an associated positive integer a_i for $1 \leq i \leq p$ and $\sum_{i=1}^{p} a_i = a$. Then we can check the existence of a weighted partial-(k,r,t)-center such that it contains a_i elements from S_i, $1 \leq i \leq p$, in $O((2r+1)^{\frac{3\mathbf{b}}{2}} 2^{\frac{a}{2}} \cdot nt)$ time and, in case of a positive answer, construct a weighted partial-(k,r,t)-center of G in the same time.*

The rest of the section is devoted to the proof of the following theorem.

THEOREM 3. *Let $f : \mathbb{N} \to \mathbb{N}$ be a given function. Then WP-(k,r,t)-C problem can be solved in time $O(\tau(k,r) \cdot t \cdot (m+n))$ for graphs in $\mathcal{G}_f$, where τ is a function of k and r. In particular, WP-(k,r,t)-C problem is FPT for planar graphs, graphs of bounded genus and graphs of bounded maximum degree.*

Let us remark that for fixed k, r and t, our algorithm runs in linear time.

PROOF. The proof of the theorem is divided into three parts: Algorithm, correctness and the time complexity. We first describe the algorithm.

Algorithm: First we set up notations used in the algorithm. By $\mathcal{S}$ we mean a family of pairs (X, i) where X is a subset of $V(G)$, i is a positive integer, and for any two elements $(X_1, i_1), (X_2, i_2) \in \mathcal{S}$, $X_1 \cap X_2 = \emptyset$. Given a family $\mathcal{S}$, we define $\rho(\mathcal{S}) = \sum_{(X,i) \in \mathcal{S}} i$ and

$$\mu(w, \mathcal{S}) = \max \left\{ w(B_r(D)) \;\middle|\; D \subseteq V(G), |D| = \rho(\mathcal{S}), \forall (X,i) \in \mathcal{S}\ |D \cap X| = i \right\},$$

that is a subset $D \subseteq \bigcup_{(X,i) \in \mathcal{S}} X$ of size $\rho(\mathcal{S})$, under the additional constraint that for each element (X,i) of $\mathcal{S}$ we pick exactly i elements in X. A subset D realizing $\mu(w, \mathcal{S})$ will be called an *$\mathcal{S}$-center*. Our detailed algorithm is given in Figure 1.

Correctness: The correctness of the algorithm follows (almost directly) from its detailed descriptions in the earlier sections and hence we remark on the necessary points of the proof. Whenever we answer YES, we output a set C which has weight at least t that is $w(B_r(C)) \geq t$ and C is of size at most k and hence these steps do not require any justification.

*Results marked with [⋆] will appear in the long version of the paper.

Algorithm PCentre(G, r, k, t, w, $\mathcal{S}$, C, S, $\mu(w,\mathcal{S})$)
(The algorithms takes as an input (a) a graph $G = (V,E) \in \mathscr{G}_f$, ($b$) positive integers k, r and t, (c) a weight function $w : V \to \{0,1\}$, (d) a family $\mathcal{S}$ of pairs (X,i), (e) an $\mathcal{S}$-center C, (f) a set S which is equal to $\cup_{(X,i)\in\mathcal{S}} X$ and (g) the value of $\mu(w,\mathcal{S})$. It returns either a set C such that $w(B_r(C)) \geq t$ or returns NO, if no such set exists. The algorithm is initialized with **PCentre**$(G,r,k,t,w,\emptyset,\emptyset,\emptyset,0)$).

Step 0 : If $\mu(w,\mathcal{S}) \geq t$, then answer YES and return C.

Step 1: If $k = 0$ and $\mu(w,\mathcal{S}) < t$, then **return** NO and EXIT.

Step 2: First define A as follows: $A = \{v \mid v \in V, v \notin S, w(B_r(v)) \geq t/k\}$. If A is empty **return** NO and EXIT. Else let $S = S \cup A$ and define $\mathcal{G} = \bigcup_{v\in S} G[B_r(v)]$.

Step 3: Compute the diameter, *diam*, of $\mathcal{G}$.

Step 4: If $diam > ((12r+4)(k+\rho(\mathcal{S})))^{|\mathcal{S}|+1}$ then apply Lemma 1 to find the subset $T \subseteq A$ of size k such that: **(a)** for all $u, v \in T$, $d_G(u,v) \geq 2r+1$; and **(b)** for all $u \in T$ and for all $v \in C$, $d_G(u,v) \geq 2r+1$ and **return** $C = C \cup T$ and EXIT.

Step 5: Else, the graph $\mathcal{G}$ has bounded local treewidth, compute a tree decomposition of width $f(diam)$ of $\mathcal{G}$.

Step 6: For every $1 \leq p \leq k$, using the dynamic programming of Theorem 2, compute a $\mathcal{S} \cup \{(A,p)\}$-center D_p of weight $\mu(w, \mathcal{S} \cup \{(A,p)\})$. If for some ***recursive calls***, $1 \leq p \leq k$, **PCentre**$(G, r, k-p, t-\mu(w,\mathcal{S}\cup\{(A,p)\}), w, \mathcal{S}\cup\{(A,p)\}, D_p, S, \mu(w,\mathcal{S}\cup\{(A,p)\}))$ returns a set C then answer YES and **return** C else answer NO and EXIT.

Figure 1: Algorithm for Weighted Partial Center Problem

Our observation is that if there exists a subset C such that $w(B_r(C)) \geq t$ and $|C| \leq k$, then C and $A = \{v \mid v \in V,\ w(B_r(v)) \geq t/k\}$ have non empty intersection. Hence we recursively solve the problem with an assumption that $|C \cap A| = p$, $p \in \{1,2,\cdots,k\}$. In recursive steps we have a family $\mathcal{S}$ of pairs (X,i) such that we want to compute C with additional constraints that for all $(X,i) \in \mathcal{S}$, $|C \cap X| = i$. At this stage the only way we can have solution is when there exists a non-empty set A such that

$$C \cap A \neq \emptyset \text{ where } A = \left\{ v \;\middle|\; v \in V, v \notin (\cup_{(X,i)\in\mathcal{S}} X), w(B_r(v)) \geq \tfrac{t-\mu(w,\mathcal{S})}{k-\rho(\mathcal{S})} \right\} \neq \emptyset.$$

Now based on the diameter of the graph $\mathcal{G} = \bigcup_{v\in S} G[B_r(v)]$, where $S = A \cup_{(X,i)\in\mathcal{S}} X$, we either apply Lemma 1 or make further recursive calls.

(1.) When we apply Lemma 1, the diameter of the graph is more than $((12r+4)k)^{|\mathcal{S}|+1}$, and hence we obtain a set $T \subseteq A$ such that T is of cardinality $k - \rho(\mathcal{S})$ and the distance between any two vertices in T and distance between vertices of T and C, C a $\mathcal{S}$-center, is at least $2r+1$. In $|C \cup T| = |C| + |T| \leq \rho(\mathcal{S}) + k - \rho(\mathcal{S}) \leq k$, and

$$w(B_r(C \cup T)) = w(B_r(C)) + w(B_r(T)) \geq \mu(w,\mathcal{S}) + (k-\rho(\mathcal{S})) \times \tfrac{t-\mu(w,\mathcal{S})}{k-\rho(\mathcal{S})} \geq t.$$

(2.) Else the diameter and hence the treewidth of the graph $\mathcal{G}$ is at most $f(((12r+4)k)^{|\mathcal{S}|+1})$. Hence in this case there is a solution to the problem precisely when there exists p, $1 \leq p \leq k - \rho(\mathcal{S})$, for which recursive call to **PCentre** returns a solution in Step 6 of the algorithm. This completes the correctness of the algorithm.

Time Complexity: The running time depends on the number of recursive calls we make and the upper bound on the treewidth of the graphs $\mathcal{G}$ which we obtain during the execution of the algorithm. First we bound the number of recursive calls. An easy bound is k^k since the number of recursive calls made at any step is at most k and the depth of the recursion tree is also at most k. This bound can be improved as follows. Let $N(k)$ be the number of recursive calls. Then $N(k)$ satisfies the recurrence $N(k) \leq \sum_{i=1}^{k} N(k-i)$, which solves to 2^k.

At every recursive call we perform a dynamic programming algorithm and since the size of the family $\mathcal{S}$ is at most $k-1$, the diameter of the graph does not exceed $((12r+4)k)^k$ at any step of the algorithm. Let $h(r,k) = 3 \cdot f(((12r+4)k)^k)/2$. Then the dynamic programming algorithm can be performed in $O((2r+1)^{h(r,k)} 2^{\frac{k}{2}} \cdot (n+m)t)$ time in any recursive step of the algorithm. Hence the total time complexity of the algorithm is upper bounded by $O((2r+1)^{h(r,k)} 2^{\frac{3k}{2}} \cdot (n+m)t)$. This completes the proof. ∎

3.3 Improved Algorithm for Planar Graphs

In the last section we gave an algorithm for WP- (k,r)-C problem in graphs of bounded local treewidth. The time complexity of the algorithm was dominated by the upper bound on the treewidth of the graph $\mathcal{G}$, which were considered in the recursive steps of the algorithm. If the input to the algorithm **Algorithm PCentre** is planar, then a direct application of Lemma 1 gives us that the treewidth of the graph $\mathcal{G}$, obtained in the recursive steps of the algorithm, is bounded by $O((rk)^{O(rk)})$. In this section we reduce this upper bound to $O(rk)$ using grid arguments. We also need to slightly modify **Algorithm PCenter** by replacing the diameter arguments with treewidth based arguments. We give the modified steps here:

Modified Step 3: Compute the treewidth of $\mathcal{G}$.

Modified Step 4: If $\mathbf{tw}(\mathcal{G}) > g(r,k)$ (to be specified later) find a subset $T \subseteq A$ of size k such that: **(a)** for all $u, v \in T$, $d_G(u,v) \geq 2r+1$; and **(b)** for all $u \in T$ and for all $v \in C$, $d_G(u,v) \geq 2r+1$ and **return** $C = C \cup T$ and EXIT.

Modified Step 5: Else, the graph $\mathcal{G}$ has bounded treewidth, compute a tree decomposition of width at most $g(r,k)$ of $\mathcal{G}$.

To give the combinatorial bound on the treewidth of the graph $\mathcal{G}$, we need the following relation between the size of grids and the treewidth of the planar graph.

LEMMA 4.[23] *Let $s \geq 1$ be an integer. The treewidth of every planar graph G with no $(s \times s)$-grid as a minor is upper bounded by $6s-4$.*

The notations used in the next lemma is the same as in **Algorithm PCentre**.

LEMMA 5. [$\star$] *Let $G = (V,E)$ be a planar graph on n vertices and m edges. Let k, r and t be positive integers, and w be a weight function $w: V \to \{0,1\}$. Suppose that at some step in* **Algorithm PCentre** *we are given a family $\mathcal{S}$ of pairs (X,i), an $\mathcal{S}$-center C, a set $S = \cup_{(X,i)\in\mathcal{S}} X$ and the value of $\mu(\omega,\mathcal{S})$. Furthermore let $A = \{v \mid v \in V, v \notin S, w(B_r(v)) \geq t/k'\} \neq \emptyset$, $S^* = S \cup A$, where $k' = k - \sum_{(X,i)\in\mathcal{S}} i$. Finally, let $\mathcal{G} = \bigcup_{v\in S^*} G[B_r(v)]$. Then either there is a subset $T \subseteq A$ of size k' such that* **(a)** *for all $u, v \in T$, $d_G(u,v) \geq 2r+1$; and* **(b)** *for all $u \in T$ and for all $v \in C$, $d_G(u,v) \geq 2r+1$ or $\mathbf{tw}(\mathcal{G}) \leq O(rk)$.*

Let us set $g(r,k) = 6h(r,k)$. We can compute in $O(|\mathcal{G}|^4)$ time a tree decomposition of width ω of $\mathcal{G}$ such that $\mathbf{tw}(\mathcal{G}) \leq \omega \leq 1.5\mathbf{tw}(\mathcal{G})$ [25]. Moreover, given a graph $\mathcal{G}$, one can also construct a grid minor of size $(b/4) \times (b/4)$ where the largest grid minor possible in $\mathcal{G}$ is of order $b \times b$, in time $O(|\mathcal{G}|^2 \log |\mathcal{G}|)$ [6]. Hence if $\omega > g(r,k)$ then the $\mathbf{tw}(G) > 4h(r,k)$ and then by applying the polynomial time algorithm to compute grid minor, we can obtain a grid of size $\frac{4}{24}h(r,k)$. Let us finally observe that the proof of Lemma 5 is constructive, in a sense that given the grid $\mathcal{H}$, we can construct the desired set T in polynomial time. Hence by setting $h(r,k) = O(rk)$ in the time complexity analysis of Theorem 3 , we obtain the following theorem.

THEOREM 6. *WP-(k,r,t)-C problem can be solved in time $O(2^{O(kr)} \cdot n^{O(1)})$ on planar graphs.*

4 H-minor free graphs

The arguments of the previous sections were based on a specific graph class property, namely, that a graph with small diameter has bounded treewidth. Thus the natural limit of our framework is the class of graphs of bounded local treewidth. We overcome this limit and extend the framework on the class of graphs excluding a fixed graph H as minor. To do so we need to use the structural theorem of Robertson and Seymour [24] and an algorithmic version of this theorem by Demaine et al. [13]. The algorithm is quite involved, it uses two levels of dynamic programming and two levels of implicit branching, and can be seen as a non-trivial extension of the algorithm of Demaine et al. [10] for classical covering problems to partial covering problems.

THEOREM 7. $[\star]$ *PDS is fixed parameter tractable for the class of H-minor free graphs and the algorithm takes time $O(\tau(k) \cdot t \cdot n^{C_H})$, where τ is the function of k only and C_H is the constant depending only on the size of H.*

5 Partial Vertex Cover

While the results of the previous section can be used to prove that PVC is FPT on H-minor free graphs, we do not need that heavy machinery for this specific problem. In this section we show how implicit branching itself does the job, even for more general classes of graphs. We present a simple modification to our framework developed in the Section 3.1 and use it to show that PVC problem is FPT in triangle free graphs. Given a graph $G = (V, E)$ and a subset $S \subseteq V$, by $\partial S \subseteq E$ we denote the set of all edges having at least one end-point in S. Our modification in the generic algorithm is in step **(T2)**.
(T2′) Bound the size of S as a function of the parameter in every recursive step.

We call a graph class $\mathscr{G}$ *hereditary* if for any $G \in \mathscr{G}$, all the induced subgraphs of G also belong to $\mathscr{G}$. Let $\xi : \mathbb{N} \rightarrow \mathbb{N}$ be an increasing function. We say that a hereditary graph class $\mathscr{G}$ has the *ξ-bounded independent set property*, or simply the property IS_ξ, if for any $G \in \mathscr{G}$ there exists an independent set $X \subseteq V(G)$ such that $|V(G)| \leq \xi(|X|)$ and X can be found in time polynomial in the input size. There are various graph classes which have the property of IS_ξ. Every bipartite graph has an independent set of size at least $n/2$ and hence we can choose $\xi_b : \mathbb{N} \rightarrow \mathbb{N}$ as $\xi_b(k) = 2k$. A triangle free graph has an independent set of size at least $\max\{\Delta, n/(\Delta+1)\}$ where Δ is the maximum degree of the graph which implies that a triangle free graphs has an independent set of size at least $\sqrt{n}/2$. In this case we can choose the function $\xi_t : \mathbb{N} \rightarrow \mathbb{N}$ by $\xi_t(k) = 4k^2$. Every H-minor free graphs, and in particular for planar graphs and graphs of bounded genus have chromatic number at most $g(H)$ for some function depending on H alone. In this case G has an independent set of size at least $n/g(|H|)$ and we can take $\xi_H(n) = g(H)n$. For planar graphs $g(H)$ is 4.

We can show that if a graph class $\mathcal{G}$ has the property IS_ξ, then in the case of PVC for every $G \in \mathcal{G}$ either we can upper bound the size of S used in the implicit branching step by $\xi(k)$ or we can find a subset V' of size at most k such that $|\partial V'| \geq t$. The main theorem of this section is as follows.

THEOREM 8. [$\star$] *Let $\mathscr{G}$ be a hereditary graph class with the property of IS_ξ for some integer function ξ. Then* PVC *can be solved in $O(\tau(k) \cdot n^{O(1)})$ time in $\mathscr{G}$ where $\tau(k) = \xi(k)^k$.*

6 Conclusion

In this paper we obtained a framework to give FPT algorithms for various partial covering problems in graphs with locally bounded treewidth and graphs excluding a fixed graph H as a minor. The main idea behind our approach was the concept of implicit branching which is of independent interest. We believe that it will be useful for other problems as well. We conclude with some open questions. For planar graphs (and even more generally, for H-minor free graphs), many non-partial versions of parameterized problems can be solved in subexponential time [12, 14]. We show that for planar graphs PARTIAL DOMINATING SET can be solved in time $2^{O(k)} \cdot n^{O(1)}$. Is this result tight, in a sense that up to some assumption in the complexity theory, there is no time $2^{o(k)} \cdot n^{O(1)}$ algorithm solving this problem on planar graphs?

Many non-partial parameterized problems on planar graphs can be solved by reducing to a kernel of linear size [2]. This does not seem to be the case for their partial counterparts and an interesting question here is, whether PARTIAL DOMINATING SET or PARTIAL VERTEX COVER can be reduced to polynomial sized kernels on planar graphs.

References

[1] J. Alber, H. Fan, M. R. Fellows, H. Fernau, R. Niedermeier, F. A. Rosamond, and U. Stege. *A refined search tree technique for Dominating Set on planar graphs.* Journal of Computer and System Sciences, 71(4), 385-405 (2005).

[2] J. Alber, M. R. Fellows and R. Niedermeier. *Polynomial-time Data Reduction for Dominating Set*. Journal of ACM 51(3): 363-384 (2004).

[3] N. Alon and S. Gutner. *Linear Time Algorithms for Finding a Dominating Set of Fixed Size in Degenerated Graphs.* In the Proc. of COCOON, LNCS 4598,394-405 (2007).

[4] S. Arora and G. Karakostas. *A $2+\epsilon$ Approximation Algorithm for the k-MST Problem*. In Proc. of Symposium on Discrete Algorithms (SODA), 754-759 (2000).

[5] Reuven Bar-Yehuda. *Using Homogenous Weights for Approximating the Partial Cover Problem*. In Proc. of Symposium on Discrete Algorithms (SODA), 71-75 (1999).

[6] H. L. Bodlaender, A. Grigoriev, A. M. C. A. Koster. *Treewidth Lower Bounds with Brambles*. In Proc. of European Symposium on Algorithms (ESA), LNCS 3669, 391-402, (2005).

[7] M. Charikar, S. Khuller, D. Mount and G. Narasimhan. *Algorithms for Facility Location Problems with Outliers*. In Proc. of Symposium on Discrete Algorithms (SODA), 642-651 (2001).

[8] J. Chen, I. A. Kanj and G. Xia. *Improved Parameterized Upper Bounds for Vertex Cover*. In Proc. of Mathematical Foundations of Computer Science (MFCS), LNCS 4162, 238-249 (2006).

[9] E. D. Demaine, F. V. Fomin, M.T. Hajiaghayi and D. M. Thilikos. *Fixed Parameter Algorithms for (k, r)-Center in Planar Graphs and Map Graphs*. ACM transactions on Algorithms, 1(1): 33-47 (2005).

[10] E. D. Demaine, F. V. Fomin, M.T. Hajiaghayi and D. M. Thilikos. *Subexponential Parameterized Algorithms on Bounded-genus Graphs and H-minor-free Graphs*. Journal of ACM 52(6): 866-893 (2005).

[11] E. D. Demaine and M.T. Hajiaghayi. *Equivalence of Local Treewidth and Linear Local Treewidth and its Algorithmic Applications*. In Proc. of Symposium on Discrete Algorithms (SODA), 840-849 (2004).

[12] E. DEMAINE AND M. HAJIAGHAYI, *The bidimensionality theory and its algorithmic applications*, The Computer Journal, to appear.

[13] E. D. Demaine, M. T. Hajiaghayi and K. C. Kawarabayashi. *Algorithmic Graph Minor Theory: Decomposition, Approximation and Coloring*. In Proc. of Foundations of Computer Science (FOCS), 637-646 (2005).

[14] F. Dorn, F. V. Fomin, and D. M. Thilikos. *Subexponential parameterized algorithms*, Computer Science Reviews, to appear.

[15] R.G. Downey and M.R. Fellows. *Parameterized Complexity*, Springer, (1999).

[16] D. Eppstein. *Diameter and Treewidth in Minor Closed Graph Families*, Algorithmica, 27 (3-4): 275-291 (2000).

[17] J. Flum and M. Grohe. *Parameterized Complexity Theory*, Springer, (2006).

[18] R. Gandhi, S. Khuller and A. Srinivasan. *Approximation Algorithms for Partial Covering Problems*. Journal of Algorithms 53(1): 55-84 (2004).

[19] P. Golovach and Y. Villanger. *Parameterized Complexity for Domination Problems on Degenerate Graphs*. To appear in the proceedings of WG, LNCS, (2008).

[20] M. Grohe. *Local Treewidth, Excluded Minors and Approximation Algorithms*. Combinatorica, 23(4): 613-632 (2003).

[21] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*, Oxford University Press, (2006).

[22] N. Robertson and P. Seymour. *Graph minors V, Excluding a Planar Graph*. Journal of Combinatorial Theory, Series B, 41(2): 92-114, (1986).

[23] N. Robertson, P. Seymour and R. Thomas. *Quickly Excluding a Planar Graph*. Journal of Combinatorial Theory, Series B, 62(2): 323-348 (1994)

[24] N. Robertson and P. Seymour. *Graph minors XVI, Excluding a Non-planar Graph*. Journal of Combinatorial Theory, Series B, 81(1): 43-76 (2003).

[25] P. D. Seymour and R. Thomas. *Call routing and the ratcatcher*. Combinatorica, 14(2):217241, (1994).