Parameterized Complexity of Finding Regular Induced Subgraphs

HANNES MOSER[1] AND DIMITRIOS M. THILIKOS[2]

ABSTRACT. The r-REGULAR INDUCED SUBGRAPH problem asks, given a graph G and a non-negative integer k, whether G contains an r-regular induced subgraph of size at least k, that is, an induced subgraph in which every vertex has degree exactly r. In this paper we examine its parameterization k-SIZE r-REGULAR INDUCED SUBGRAPH with k as parameter and prove that it is $W[1]$-hard. We also examine the parameterized complexity of the dual parameterized problem, namely, the k-ALMOST r-REGULAR GRAPH problem, which asks for a given graph G and a non-negative integer k whether G can be made r-regular by deleting at most k vertices. We show that this problem is in FPT by proving the existence of a problem kernel of size $O(kr(r+k)^2)$.

1 Introduction

Regular graphs as well as regular subgraphs have been intensively studied from a structural point of view (e.g., [1]). An interesting problem related to regular graphs is to decide whether a given graph contains an regular subgraph. One of the first problems of this kind was stated by Garey and Johnson: CUBIC SUBGRAPH, i.e., the problem of deciding whether a given graph contains a 3-regular subgraph, is NP-complete [9]. This result was later expanded in [17], where it was shown that CUBIC SUBGRAPH is NP-complete on planar graphs with maximum degree 7. Moreover, it was shown by Stewart that CUBIC SUBGRAPH is also NP-complete on bipartite graphs with maximum degree 4 [19]. The same author showed that the more general problem of deciding whether a given graph contains an r-regular subgraph for some fixed degree $r > 3$ is NP-complete on general graphs as well as on planar graphs [18] (where in the latter case only $r = 4$ and $r = 5$ were considered, since any planar graph contains a vertex of degree at most 5). Note that this problem is polynomial-time solvable for $r \leq 2$ [4].

[1]Supported by the EC Research Training Network HPRN-CT-2002-00278 (COMBSTRU) and by the Deutsche Forschungsgemeinschaft, Project ITKO, NI 369/5.

[2]Supported by the spanish CICYT project TIN-2004-07925 (GRAMMARS).

We consider a variant of this problem, where we ask whether a given graph G contains an *induced* subgraph of at least k vertices that is r-regular, and we call it r-REGULAR INDUCED SUBGRAPH for any $r \geq 0$. The *exact* version of this problem is obtained if we ask for an induced subgraph of size *exactly* k (for the difference between the original and the exact version of the problem, see Figure 1). In this paper, we examine the following dual parameterizations of the problem.

k-ALMOST r-REGULAR GRAPH:
Input: A graph $G = (V, E)$ and an integer k.
Question: Is there a vertex subset $S \subseteq V$ of size at most k such that $G[V \setminus S]$ is r-regular?

k-SIZE r-REGULAR INDUCED SUBGRAPH
Input: A graph $G = (V, E)$ and an integer k.
Question: Is there a vertex subset $S \subseteq V$ of size at least k such that $G[S]$ is r-regular?

Note that for $r = 0$, r-REGULAR INDUCED SUBGRAPH is equivalent to INDEPENDENT SET. In this paper, we show that r-REGULAR INDUCED SUBGRAPH is NP-complete for $r \geq 1$ even if we restrict the input of the problem to planar graphs or triangle free planar graphs (for planar graphs, we consider $r \leq 5$ and for triangle free planar graphs we consider[1] $r \leq 3$). Our reduction also implies that k-SIZE r-REGULAR INDUCED SUBGRAPH is $W[1]$-hard for any $r \geq 0$, suggesting that no FPT algorithm exists for this parameterized problem. However, the dual problem, k-ALMOST r-REGULAR GRAPH, is easier. We design an FPT algorithm running in $O(n(k+r) + kr^2(k+r)^2 \cdot (r+2)^k)$ steps, where n is the number of vertices of the input graph. Our algorithm is based on the existence of a reduction of k-ALMOST r-REGULAR GRAPH to a problem kernel of size $O(kr(k+r)^2)$ (in the special case where $r = 1$, this size can be improved to $O(k^2)$).

r-REGULAR INDUCED SUBGRAPH belongs to the general category of subgraph problems, i.e., problems that ask for the existence of an (induced) subgraph with a certain property (e.g., being r-regular). Most of the problems of this type concern hereditary properties (a property is *hereditary* if it holds for any induced subgraph of G whenever it holds for G) and can be classified as NP-hard [13, 20]. Basically all such problems, when parameterized by the number of vertices that need to be removed in order to obtain the desired property, admit fixed-parameter algorithms provided that the corresponding property is hereditary [2, 12]. For an example of related re-

[1] This restriction follows from Euler's formula.

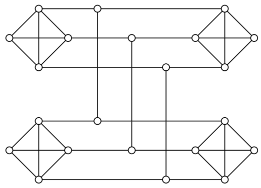

Figure 1. For 3-Regular Induced Subgraph, the instance (G, k) is a yes-instance iff $k \in \{0, 1, \ldots, 17, 18\}$. However, for its "exact version", the same instance is a yes-instance iff $k \in \{4, 8, 12, 16, 18\}$.

sults, we refer the reader to the results in [14], [8, 16], and [5, 11] where the imposed property is chordality, 2-colorability, and acyclicity, respectively.

Notice that r-regularity is not a hereditary property as subgraphs of r-regular graphs are generally not r-regular, for $r \geq 1$. This suggests that the existing results do not help to prove hardness or fixed-parameter intractability for the problems we consider in this paper. Moreover, the lack of heredity makes it harder to design parameterized algorithms. Another vertex removal problem with a non-hereditary property was examined in [6] where it is shown that the problem of converting a given graph into a grid by vertex/edge removals and additions is fixed-parameter tractable.

The fact that r-regularity is not a hereditary property makes it possible to define different versions for both problems above if, in their questions, we ask for a vertex set S of size *exactly* k (instead of one of size at most k). As the present "$\leq$"-versions of the problems are weaker than (can be reduced to) their "=" counterparts, we will prove all of our hardness results for the above weaker setting (see Figure 1). However, in order to obtain a stronger result, all the algorithms in this paper are designed for the "exact" versions of our problems (modifications for solving the original versions are easy and have the same running times).

The remaining document is organized as follows: First, we shortly introduce necessary definitions from parameterized complexity theory and graph theory in Section 2. Section 3 is dedicated to hardness results. In Section 4 we present the problem kernel and the exact algorithm for k-Almost r-Regular Graph.

2 Preliminaries

In this paper we deal with fixed-parameter algorithms that emerge from the field of parameterized complexity analysis [7, 15]. An instance of a parameterized problem consists of a problem instance I and a parameter k. A parameterized problem is *fixed-parameter tractable* if it can be solved in $f(k) \cdot |I|^{O(1)}$ time, where f is a computable function depending only on the parameter k, not on the input size $|I|$. One of the common methods to prove that a problem is fixed-parameter tractable is to provide *data reduction rules* that lead to a *problem kernel*: Given a problem instance (I, k), a data reduction rule replaces that instance by a another instance (I', k') in polynomial time, such that (I, k) is a yes-instance iff (I', k') is a yes-instance. An instance to which none of a given set of reduction rules applies is called *reduced* with respect to the rules. A parameterized problem is said to have a problem kernel if, after the application of the reduction rules, the resulting reduced instance has size $f(k)$ for a function f depending only on k. Analogously to classical complexity theory, Downey and Fellows [7] developed a framework providing a reducibility and completeness program. A *parameterized reduction* from a parameterized language L to another parameterized language L' is a function that, given an instance (x, k), computes in time $f(k) \cdot n^{O(1)}$ an instance (x', k') (with k' depending only on k) such that $(x, k) \in L \Leftrightarrow (x', k') \in L'$. The basic complexity class for fixed-parameter intractability is $W[1]$ as there is good reason to believe that $W[1]$-hard problems are not fixed-parameter tractable [7].

In this paper we assume that all graphs are simple and undirected. For a graph $G = (V, E)$ we write $V(G)$ to denote its vertex set and $E(G)$ to denote its edge set. For a subset $V' \subseteq V$, by $G[V']$ we mean the subgraph of G induced by V'. We write $G \setminus V'$ to denote the graph $G[V \setminus V']$. If $v \in V$ we also write $G - v$ instead of $G \setminus \{v\}$. The *(open) neighborhood* $N(V')$ of a given set $V' \subseteq V$ is the set of all vertices in $V \setminus V'$ adjacent to some vertex in V'. We sometimes write $N_G(V')$ to emphasize that we refer to the open neighborhood of V' within the graph G. We write K_r to denote the complete graph with r vertices.

In the next section, we show that k-Almost r-Regular Graph as well as k-Size r-Regular Induced Subgraph are NP-complete. Moreover, we show that the latter problem is also $W[1]$-hard.

3 Hardness and Completeness Results

In this section we first show that r-Regular Induced Subgraph is NP-complete by giving a polynomial-time reduction from Vertex Cover. A similar but independently obtained result has recently appeared in [3] proving the NP-hardness of finding an induced r-regular bipartite graph.

THEOREM 1. *The r-Regular Induced Subgraph problem is NP-compl. for any $r \geq 0$. It also remains NP-complete when restricted to planar graphs (for $r \leq 5$) or to triangle-free planar graphs (for $r \leq 3$).*

Proof. We first prove the theorem in its general statement and then we explain how to modify the proof for its planar versions. For the proof we use the dual problem of r-Regular Induced Subgraph, that is, we search for a vertex subset S of size at most k such that $G \setminus S$ is r-regular. The dual problem is polynomially equivalent to r-Regular Induced Subgraph. For $r = 0$ the dual problem is identical to Vertex Cover, which is known to be NP-complete. For all remaining $r > 0$ we give a reduction from the dual problem with $r = 0$.

Let (G, k) be an instance of the dual problem with $r = 0$. We construct an instance (G', k') of the dual problem with $r > 0$ as follows: First, we set $G' := G$ and $k' := k \cdot (r+1)$. For each vertex $v \in G$ we add a copy of K_{r+1} to G'. Let R_v be the copy corresponding to vertex v. For all vertices $v \in G$ we identify v with an arbitrary vertex in R_v, i.e., we set $v = w$ for some arbitrary $w \in R_v$.

We have to show that (G, k) with $r = 0$ is a yes-instance iff (G', k') with $r > 0$ is a yes-instance.

($\Rightarrow$): Suppose that there is a size-k solution S for (G, k), that is, $G \setminus S$ consists of isolated vertices. We define a new solution set $S' := \bigcup_{v \in S} R_v$ of size $k \cdot (r + 1)$ for G'. Clearly, $G' \setminus S'$ is a graph in which every connected component is a R_v, i.e., $G' \setminus S'$ is an r-regular graph, thus S' is a solution for (G', k').

($\Leftarrow$): Suppose that there is a size-k' solution S' for (G', k'). We say that S' is *clustered* if

$$\forall v \in V(G) : R_v \cap S' \neq \emptyset \Rightarrow R_v \subseteq S',$$

and notice that if S' is clustered then $S = \{v \in V(G) \mid R_v \cap S' \neq \emptyset\}$ is a solution for the instance (G, k) of Vertex Cover. In case the solution S' is not clustered, we can turn it into a clustered one according to the following claim, which completes the proof of correctness of the reduction.

Claim: *Given a solution S' for (G', k') where $|S'| \leq k'$, we can always construct a clustered solution S'' for the same problem instance.*

Proof of claim: Omitted.

Vertex Cover remains NP-complete when restricted to triangle free planar graphs [10]. Therefore, the above proof also implies that r-Regular Induced Subgraph remains NP-complete even when we restrict it to planar graphs for $r \leq 5$. The only modification is that, for the cases where $r = 4$ or $r = 5$, we attach to G an octahedron or an icosahedron instead of K_5 or K_6, respectively. Moreover, the same reduction implies also the NP-completeness for triangle free planar graphs for $r \leq 3$ when, in the case $r = 3$, we replace K_4 by the cube. ∎

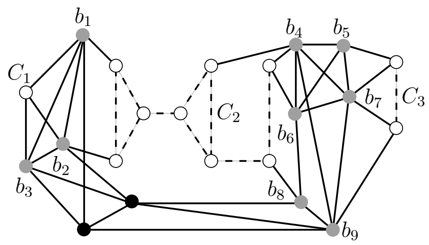

Figure 2. Example of a graph with clean regions C_1, C_2, and C_3 (white vertices, $r = 3$). The dotted edges denote the connected subgraph each clean region induces. Dirty vertices are gray or black, boundary vertices are gray and all other dirty vertices are black. The boundary for C_1 is $B_1 = \{b_1, b_2, b_3\}$, the boundary for C_2 is $B_2 = \{b_1, b_2, b_4, b_6, b_8\}$, and the boundary for C_3 is $B_3 = \{b_5, b_7, b_9\}$. Note that boundaries can have vertices in common, for instance, $B_1 \cap B_2 \neq \emptyset$.

It is known that the parameterized version of INDEPENDENT SET (the dual problem of VERTEX COVER) where the parameter is the size of the independent set is $W[1]$-hard. As the reduction in the above proof is a parametric reduction, we also have the following:

THEOREM 2. *k-SIZE r-REGULAR INDUCED SUBGRAPH is $W[1]$-hard.*

4 A Problem Kernel

A central ingredient for presenting the results of this section is the notion of a clean region. We call a vertex of G *clean* if it has degree r, and *dirty* otherwise. We define a *clean region* in G as a maximal subset of clean vertices that induces a connected subgraph in G. Let $\{C_i : i \in I\}$ be the set of all clean regions. The open neighborhood of each clean region C_i is called its *boundary* B_i (notice that two different boundaries may share common vertices). A clean region C_i is called *isolated* if $B_i = \emptyset$. Observe that the neighborhood of a non-isolated clean region consists entirely of dirty vertices. See Figure 2 for examples of clean regions and their boundaries. The detection of all clean regions in G can be done in $O(nr)$ steps.

The main result of this section is the following.

THEOREM 3. *The exact version (demanding a solution of size exactly k) of the k-ALMOST r-REGULAR GRAPH problem, for $r \geq 1$, has a kernel of size $O(kr(k+r)^2)$, which can be constructed in $O(n \cdot (k+r))$ time.*

Proof. The idea of our kernelization is to apply a series of reduction steps to the input instance that either give a negative answer or produce a new equivalent instance satisfying a bigger subset of the following properties.

(1) All vertices in G have degree at least r and at most $k+r$,
(2) each vertex of a boundary B_i has at most r clean neighbors in C_i,
(3) the isolated clean regions of G contain in total at most $(3k^2+k)/2$ vertices,
(4) for every clean region C_i with boundary B_i, $|C_i| \leq 1+(r+1)\cdot(1+\max\{\lceil\frac{k+1}{r+1}\rceil, |B_i|\})$.

Then we will prove that if all the above properties hold for an instance of k-Almost r-Regular Graph, then the size of the instance is the claimed one. For our presentation, we will use $(K_r, 1)$ as the no-instance of k-Almost r-Regular Graph for $r \geq 0$. We proceed with the first reduction step.

Step 1:

1. *While* $k \geq 0$ *and* $\exists_{v\in V(G)}$ $\deg_G(v) < r \vee \deg_G(v) > k+r$*:*
   $(G,k) \leftarrow (G \setminus \{v\}, k-1)$.
2. *If* $k \geq 0$*, then return* (G,k)*, otherwise, return* $(K_r, 1)$.

Consider an instance (G,k) of k-Almost r-Regular Graph. Vertices v in G with $\deg(v) < r$ obviously must be contained in the solution S. Likewise, graph vertices v with degree $\deg(v) > k+r$ must be in S, as we would have to put more than k of its neighbors into S to achieve degree r for v. We conclude that **Step 1** produces an equivalent instance satisfying property (1).

For the next step, observe that taking a vertex of a clean region into the solution S causes its clean neighbors to have a degree less than r in $G \setminus S$, forcing them into the solution as well. By applying the same argument inductively on the clean neighbors we can see that either no vertex of a clean region is a part of the solution S, or the entire clean region is contained in S.

Step 2:

1. *While* $k \geq 0$ *and* G *contains a clean region* C_i *whose boundary* B_i *contains a vertex* v *with more than* r *clean neighbors in* C_i*:*
   $(G,k) \leftarrow (G \setminus C_i, |k| - |C_i|)$.
2. *If* $k \geq 0$ *then return* (G,k)*, otherwise, return* $(K_r, 1)$.

To justify **Step 2**, assume that a solution S of size exactly k exists, i.e., $G \setminus S$ is r-regular. Notice that all vertices in $N_G(S)$ are dirty. Therefore, a clean region will be a subgraph of either $G[S]$ or of $G \setminus (S \cup N_G(S))$.

Let C_i be a clean region and let $v \in B_i$ be a vertex of its boundary with more than r neighbors in C_i. If C_i is a subgraph of $G \setminus (S \cup N_G(S))$ then v can have at most r neighbors in C_i because $G \setminus S$ is an r-regular graph. Therefore $C_i \subseteq S$ and thus **Step 2** produces an equivalent instance for k-Almost r-Regular Graph, satisfying properties (1) and (2).

Our observation that either no vertex of a clean region or the entire clean region is contained in the solution implies that isolated clean regions that contain more than k vertices cannot be part of the solution. This leads to the next reduction step.

Step 3:

1. *While G contains an isolated clean region C_i where $|C_i| \geq k+1$:* $(G,k) \leftarrow (G \setminus C_i, k)$.
2. *for $i = 1, \ldots, k$ do:* *If G contains s isolated clean regions of i vertices, then modify G by removing* $\max\{0, s - \lceil k/i \rceil\}$ *of them.*

Concerning the second part of this step, observe that if there are more than k isolated clean regions of equal size i, then we can remove all but $\lceil k/i \rceil$ of them. Considering all possible sizes (at most k), we can conclude that there are at most $\sum_{i=1,\ldots,k} \lceil \frac{k}{i} \rceil \cdot i \leq \sum_{i=1,\ldots,k} (\frac{k}{i}+1) \cdot i = (3k^2+k)/2$ vertices in isolated clean regions. Therefore, **Step 3** produces an equivalent instance for k-Almost r-Regular Graph, satisfying properties (1)–(3).

Let E_i be the set of edges connecting vertices in B_i with vertices in C_i. The idea for the next reduction step is to search for all clean regions C_i of size greater than $(x+\delta_i) \cdot (r+1) + \delta_i$ where $x = \max\{|B_i|, \lceil \frac{k+1}{r+1} \rceil\}$ and $\delta_i = |E_i| \bmod 2$, and to replace each one by a clean region of size $(x+\delta_i) \cdot (r+1) + \delta_i$ without affecting the neighborhoods of each vertex in the corresponding boundary B_i (notice that $(x+\delta_i) \cdot (r+1) + \delta_i \geq x \cdot (r+1) \geq k+1$, which is important as this prevents such a new clean region from being part of the solution). To do this, we first define an $(r+1)$-regular graph $R_{r,x}$ as follows: Take a cycle of $2 \cdot (x+\delta_i)$ edges, remove every second edge $\{y,z\}$ and replace it by a graph $G_{y,z}$ consisting of a $(r-1)$-clique whose vertices are all connected with v and u (notice that $G_{y,z}$ is K_{r+1} with an edge removed). The resulting graph is r-regular. As any single $G_{y,z}$ contains a matching of size $\lfloor \frac{r-1}{2} \rfloor + 1$ (see Figure 3), $R_{r,x}$ contains a matching M of size $(x+\delta_i)(\lfloor \frac{r-1}{2} \rfloor + 1)$. Observe that $R_{r,x}$ remains connected if we remove from it all (or some part) of the edges of M. Finally, $R_{r,x}$ contains $(x+\delta_i) \cdot (r+1)$ vertices. The next step applies these observations.

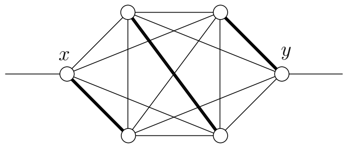

Figure 3. The graph $G_{y,z}$ for $r = 5$ and a matching in it.

Step 4:
Apply the following replacement procedure for each non-isolated clean region C_i of G with $|C_i| > (x + \delta_i) \cdot (r + 1) + \delta_i$:

1. *From property (2), $|E_i| \leq r \cdot |B_i|$. Subdivide in G all the edges in E_i and then remove all vertices in C_i from G. Notice that the set L of subdivision vertices survives after this removal and $|L| = |E_i|$.*

2. *Add $R_{r,x}$ to G, i.e., $G \leftarrow G \cup R_{r,x}$. Choose, arbitrarily, a subset $M' \subseteq M$ of the matching M where $|M'| = |L|/2 + \delta_i(r-2)/2$ (notice that $\delta_i = 1$ implies that r is odd). Notice that this is possible as $|M| = (x+\delta_i)\cdot(\lfloor\frac{r-1}{2}\rfloor+1) \geq (x+\delta_i)\cdot r/2 \geq (|B_i|+\delta_i)\cdot r/2 \geq |L|/2+\delta_i r/2 \geq |M'|$ (recall that $\lfloor\frac{r-1}{2}\rfloor + 1 \geq r/2$ holds when r is an integer). If $\delta_i = 0$ then remove the edges of M' from $R_{r,x}$ and identify, arbitrarily, their endpoints with the vertices in L. If $\delta_i = 1$, then remove the edges of M' from $R_{r,x}$, identify, arbitrarily, $|L| - 1$ of their endpoints with all vertices in L except one (say w), and then make w adjacent with the remaining $2|M'| - (|L| - 1) = |L| + \delta_i(r-2) - (|L| - 1) = r - 1$ endpoints of the edges in M' (this makes w end up with degree r).*

Step 4 replaces a clean region C_i of size more than $(x + \delta_i) \cdot (r + 1) + \delta_i$ by one of size $(x + \delta_i) \cdot (r + 1) + \delta_i \geq k + 1$. Moreover, the new clean region C_i' is connected, has the same boundary B_i as C_i, and all vertices in B_i have the same number of neighbors in C_i' as they had in C_i. We will now prove that **Step 4** produces an equivalent instance for k-Almost r-Regular Graph. For this, let S be a size-k vertex set such that $G \setminus S$ is r-regular. A solution S for G cannot contain a vertex of any C_i that changed, as C_i contains more than k vertices. We maintained the vertex degree of all vertices in B_i, C_i' is also a clean region in G', and we did not alter the subgraph $G \setminus C_i = G' \setminus C_i'$, thus $G' \setminus S$ must also be r-regular. Therefore, S is also a solution for G'. The same argument holds for the other direction: A solution S' for G' cannot contain any vertex of any C_i', as C_i' contains more than k vertices, and since $G \setminus C_i = G' \setminus C_i'$ we know that S' is also a solution for G. Finally, it is easy to verify that the new instance satisfies properties (1) – (4).

The last reduction step is the following:

Step 5:
If G contains more than $rk(k+r)(k+3r+5)+k+k(k+r)+(3k^2+k)/2$ vertices then return $(K_r,1)$, otherwise return (G,k).

Assume that a solution S of size at most k exists, i.e., $G \setminus S$ is r-regular. We define $D = N_G(S)$ and $F = V(G) \setminus (S \cup D)$ and observe that S, D, F is a 3-partition of $V(G)$. From property (1) every vertex in G has degree at most $r+k$. Therefore, the number of vertices in the neighborhood of S cannot exceed $k(r+k)$ and thus $|D| \leq k(r+k)$. We also observe that all vertices in F are clean, otherwise $G \setminus S$ would contain a vertex not having degree r, which contradicts S being a solution. It remains to bound the size of F. Recall that a clean region C_i is either completely contained in S or no vertex in C_i is member of S, thus $C_i \subseteq F$. Therefore, as all vertices in F are clean, F is a union of clean regions. Suppose that F consists of a set $\mathcal{C} = \{C_i \mid 0 \leq i \leq q\}$ of q non-isolated clean regions. As there are no edges in G with endpoints in both S and F (i.e., D separates S and F) and all vertices in F are clean, we obtain that all boundary vertices of the clean regions in $\mathcal{C}$ must be in D, i.e., $\bigcup_{i=1,\ldots,q} B_i \subseteq D$. Also, since $G \setminus S$ is r-regular, each vertex of D belongs to at most r sets in $\mathcal{B} = \{B_1, \ldots, B_q\}$ and this implies that $q \leq r \cdot |D|$ and that $\sum_{i=1,\ldots,q} |B_i| \leq r \cdot |\bigcup_{i=1,\ldots,q} B_i| \leq r \cdot |D| \leq rk(k+r)$. From property (4), $|C_i| \leq (\max\{\lceil \frac{k+1}{r+1} \rceil, |B_i|\} + \delta_i) \cdot (r+1) + \delta_i \leq \max\{|B_i| \cdot (r+1), k+r+2\} + r + 2$. Recall that F contains at most $\sum_{1,\ldots,q} |C_i|$ vertices from non-isolated clean regions. From property (3), no more than $(3k^2+k)/2$ vertices are contained in isolated regions. Therefore, $|F| \leq (3k^2+k)/2 + \sum_{1,\ldots,q} (\max\{|B_i| \cdot (r+1), k+r+2\} + r + 2) \leq (3k^2+k)/2 + \sum_{1,\ldots,q} |B_i| \cdot (r+1) + k + 2r + 4 \leq (3k^2+k)/2 + \sum_{1,\ldots,q} |B_i| \cdot (r+1) + \sum_{1,\ldots,q} (k+2r+4) = (3k^2+k)/2 + rk(k+r)(r+1) + rk(k+r)(k+2r+4) = (3k^2+k)/2 + rk(k+r)(k+3r+5)$. Since $|S| \leq k$ and $|D| \leq k(k+r)$ we can conclude that **Step 5** returns an equivalent instance of size $O(kr(k+r)^2)$ and the claimed kernel size is correct. To complete the proof, observe that the first step requires $O((k+r)n)$ steps and all the rest runs in $O(rn)$ steps. ■

Notice that Theorem 3 holds also for the non-exact version (demanding a solution of size at most k) of k-Almost r-Regular Graph. The only modification is that we have to replace the second part of **Step 3** by a deletion of all isolated clean regions.

The above kernelization applies also for $r = 1$. In this case, every non-isolated clean region contains a single vertex and **Step 4** does not apply at all. This permits us to make a better counting of the vertices in F that, apart from those belonging to isolated clean regions, are at most as many as the vertices in D. As any isolated clean region contains exactly 2 vertices when $r = 1$, the second part of **Step 3** should be applied only for $i = 2$ leaving at most $k + 1$ vertices in isolated clean regions. Therefore, in the case $r = 1$, the kernel has size at most $|S| + 2|D| \leq k + 2k(k+1) + k + 1 = O(k^2)$.

Using a bounded search tree technique, it is possible to prove that solution for k-Almost r-Regular Graph can be found in $O(nr(r+2)^k)$ time, if it exists (the proof is easy and omitted). We conclude to the following.

THEOREM 4. *For any $r \geq 0$, there exists an algorithm for k-*Almost *r-*Regular Graph *with parameter k that runs in $O(n(k+r) + kr^2(k+r)^2 \cdot (r+2)^k)$ steps.*

5 Conclusion

Notice that our results do not prove that k-Almost r-Regular Graph is in FPT when r is part of the input problem. We can prove that this version has the same parameterized complexity as the problem asking whether it is possible to delete at most k vertices such that the resulting graph is regular (without knowing its degree in advance). It remains open whether they are fixed-parameter tractable.

In this paper, we show that the parameterized problem asking whether we can make a graph r-regular by removing k vertices, with k as parameter, is fixed-parameter tractable by giving a (polynomial size) problem kernel. In the construction of the kernel we used the fact that big "clean regions" can be safely replaced by smaller ones (but not too small). Because r-regularity is not a hereditary property, we had to take care that such a replacement locally maintains r-regularity. Similar ideas were employed in [6] for a distinct, non-hereditary, property. It is an interesting problem to characterize the properties for which the vertex removal problem is fixed-parameter tractable. That way, one might extend the general result in [2] for non-hereditary properties as well.

Acknowledgements

We wish to thank an anonymous referee for detecting, in a previous version of this paper, a flaw in **Step 4** of our kernelization. We thank Jiong Guo and Sebastian Wernicke for giving helpful advice and inspiring ideas. We also thank Josep Diaz for encouraging us to work on this problem.

BIBLIOGRAPHY

[1] N. Biggs. *Algebraic Graph Theory, Second Edition.* Cambridge University Press, February 1994.

[2] L. Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Information Processing Letters*, 58(4):171–176, 1996.

[3] D. M. Cardoso, M. Kamiński, and L. Vadin. Maximum k-regular induced subgraphs. Technical Report RRR 3 – March 2006, Rutgers Center for Operations Research Rutgers University, 2006.

[4] S. A. Cook and P. McKenzie. Problems complete for deterministic logspace. *Journal of Algorithms*, 8(3):385–394, 1987.

[5] F. K. H. A. Dehne, M. R. Fellows, M. A. Langston, F. A. Rosamond, and K. Stevens. An $O(2^{o(k)}n^3)$ FPT-algorithm for the undirected feedback vertex set problem. In *Proceedings of COCOON'05*, volume 3595 of *LNCS*, pages 859–869. Springer, 2005.

[6] J. Díaz and D. M. Thilikos. Fast FPT-algorithms for cleaning grids. In *Proceedings of STACS'06*, volume 3884 of *LNCS*, pages 361–371. Springer, 2006.

[7] R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Springer, 1999.

[8] S. Fiorini, N. Hardy, B. Reed, and A. Vetta. Planar graph bipartization in linear time. In *Proceedings of GRACO'05*, volume 19 of *Electronic Notes in Discrete Mathematics*, pages 265–271 (electronic). Elsevier, 2005.

[9] M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., 1990.

[10] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(3):237–267, 1976.

[11] J. Guo, J. Gramm, F. Hüffner, R. Niedermeier, and S. Wernicke. Improved fixed-parameter algorithms for two feedback set problems. In *Proceedings of WADS'05*, volume 3608 of *LNCS*, pages 158–168. Springer, 2005. Extended version to appear under the title "Compression-based fixed-parameter algorithms for feedback vertex set and edge bipartization" in *Journal of Computer and System Sciences.*

[12] S. Khot and V. Raman. Parameterized complexity of finding subgraphs with hereditary properties. *Theoretical Computer Science*, 289(2):997-1008, 2002

[13] J. M. Lewis and M. Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980.

[14] D. Marx. Chordal deletion is fixed-parameter tractable. Manuscript, 2006.

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

[16] B. Reed, K. Smith, and A. Vetta. Finding odd cycle transversals. *Operations Research Letters*, 32(4):299–301, 2004.

[17] I. A. Stewart. Deciding whether a planar graph has a cubic subgraph is NP-complete. *Discrete Mathematics*, 126(1-3):349–357, 1994.

[18] I. A. Stewart. Finding regular subgraphs in both arbitrary and planar graphs. *Discrete Applied Mathematics*, 68(3):223–235, 1996.

[19] I. A. Stewart. On locating cubic subgraphs in bounded-degree connected bipartite graphs. *Discrete Mathematics*, 163(1-3):319–324, 1997.

[20] M. Yannakakis. Node- and edge-deletion NP-complete problems. In *Proceedings of STOC'78*, pages 253–264. ACM, New York, 1978.

Hannes Moser

Institut für Informatik, Friedrich-Schiller-Universität Jena, Ernst-Abbe-Platz 2, D-07743 Jena, Germany, moser@minet.uni-jena.de

Dimitrios M. Thilikos

Department of Mathematics, National and Capodistrian University of Athens Panepistimioupolis, GR-15784, Athens, Greece, sedthilk@math.uoa.gr