The Parameterized Complexity of Regular Subgraph Problems and Generalizations

Luke Mathieson[1] **Stefan Szeider**[1]

[1]Department of Computer Science
University of Durham
South Road
Durham
DH1 3LE, UK
Email: {luke.mathieson,stefan.szeider}@durham.ac.uk

Abstract

We study variants and generalizations of the problem of finding an r-regular subgraph (where $r \geq 3$) in a given graph by deleting at most k vertices. Moser and Thilikos (2006) have shown that the problem is fixed-parameter tractable (FPT) if parameterized by (k, r). They asked whether the problem remains fixed-parameter tractable if parameterized by k alone. We answer this question negatively: we show that if parameterized by k alone the problem is $W[1]$-hard and therefore very unlikely fixed-parameter tractable. We also give $W[1]$-hardness results for variants of the problem where the parameter is the number of vertex and edge deletions allowed, and for a new generalized form of the problem where the obtained subgraph is not necessarily regular but its vertices have certain prescribed degrees. Following this we demonstrate fixed-parameter tractability for the considered problems if the parameter includes the regularity r or an upper bound on the prescribed degrees in the generalized form of the problem. These FPT results are obtained via kernelization, so also provide a practical approach to the problems presented.

Keywords: Parameterized Complexity, Regular Subgraphs

1 Introduction

The problem of deciding whether a graph contains a non-trivial (i.e., degree at least three) regular subgraph has a long history in the field of complexity theory. Chvátal et al. (1979) give an NP-completeness result for the Cubic Subgraph problem (i.e., the problem of deciding whether a given graph has a 3-regular subgraph). Plesník (1984) shows that the Cubic Subgraph problem remains NP-complete even when restricted to a planar bipartite graph with maximum degree 4, and that the r-Regular Subgraph problem with $r \geq 3$ is NP-complete even for bipartite graphs of degree at most $r + 1$. Cheah and Corneil (1990) extend this and show that the same result holds for general graphs. Stewart (1994, 1996, 1997) gives a series of results for further constraints.

From a parameterized complexity perspective (see Section 4 for a basic introduction) there are a few natural parameterizations, by either the size of the subgraph, by the number of vertices or edges to remove to obtain a regular subgraph, or by the regu-

larity desired. Moser and Thilikos (2006) show that the problem of finding an r-regular induced subgraph on k vertices, parameterized by k is $W[1]$-hard. They also show that the Vertex Deletion to Regular Subgraph problem (which they call k-Almost r-Regular Graph) where the goal is to delete at most k vertices leaving an r-regular graph, is fixed-parameter tractable when parameterized by (k, r), with a problem kernel with $O(kr(r + k)^2)$ vertices. Stewart (2007) points out how the fixed-parameter tractability of Vertex Deletion to Regular Subgraph parameterized by (k, r) can be established by means of general logical methods. They also state that the complexity of Vertex Deletion to Regular Subgraph parameterized by k alone is an open problem.

In this paper we answer Moser and Thilikos's question, showing that Vertex Deletion to Regular Subgraph is $W[1]$-hard. We also explore several other variations of the problem, resulting in further hardness and tractability results.

The problems that we cover in this paper come in a few basic forms, centred around two general themes, whether the problem is parameterized by the number of deletion operations (deletion operations are explained in Section 2.2), k, or the number of deletion operations *and* the regularity of the graph, (k, r). This results in the following basic definition:

Deletion to Regular Subgraph
Instance: A graph $G = (V, E)$, two nonnegative integers k and r.
Question: Is there an r-regular subgraph of G obtainable by at most k deletions?

It is interesting to alter what deletion operations are available. If we restrict the operations to vertex deletion only, then we have Vertex Deletion to Regular Subgraph.

We can also further impose that we require *exactly* k operations be performed, giving Exact Deletion to Regular Subgraph.

It is also of interest to generalize both the desired degree and the cost of a deletion. To this end instead of aiming to have each remaining vertex be of degree r we introduce a degree function δ. The contribution of each edge to this total, and the cost of deleting an edge or vertex is described by a weight function ρ. This results in the following generalization:

Weighted Deletion to Chosen Degree Subgraph
Instance: A graph $G = (V, E)$, nonnegative integers k and r, a weight function $\rho : V \cup E \rightarrow \mathbb{N}^+$ and a degree function $\delta : V \rightarrow \{0, \ldots, r\}$.
Question: Is there a subgraph H of G obtainable by deletions of total cost at most k where for each

vertex v in $V(H)$, $\sum_{e \in E(v)} \rho(e) = \delta(v)$?

Of course we may also demand here that the cost be exact as well.

In this paper we show that in all examined cases parameterization by k alone gives $W[1]$-hardness, but parameterization by (k, r) gives fixed-parameter tractability. In fact if $r = 0$ the problem is equivalent to VERTEX COVER, and is thus fixed-parameter tractable. We also give several hardness results for other problems that prove useful in completion of the result. Hardness is shown by reduction from MULTI-COLOURED CLIQUE, a very useful problem, introduced by Fellows et al. (2007). Fixed parameter tractability is shown via kernelization. This is a particularly useful technique as it provides a polynomial time preprocessing algorithm (in the form of polynomial time reduction rules). This leaves a problem kernel which may then be solved by any chosen means, whether that be an exact method, approximation algorithm, or heuristic such as a genetic algorithm or simulated annealing. For a fuller treatment of kernelization in the context of parameterized complexity and preprocessing see the survey of Guo and Niedermeier (2007).

2 Preliminaries

2.1 Graph Theory and Notation

Throughout this paper we will refer only to simple, undirected graphs. Given a graph G the vertex (edge) set of G will be denoted $V(G)$ $(E(G))$ except where specific labels are given. The edge between two vertices u and v will be denoted uv (or equivalently vu). The degree of a vertex u will be denoted $d(u)$.

As this paper focuses on graph modification problems, we also define the following operation for a graph G and a set S of vertices: $G - S = G[V(G) \setminus S]$, where $G[X]$ is the subgraph of G induced by vertex set X.

2.2 Graph Modification

There are two basic operations to modify a graph to obtain a subgraph, vertex deletion and edge deletion. These operations alter a graph $G = (V, E)$ into a new graph $G' = (V', E')$. Deleting an edge uv simply removes that edge from the graph (i.e., $E' = E \setminus \{uv\}$). Deleting a vertex u removes that vertex, and any incident edges (i.e., $V' = V \setminus \{u\}$, $E' = E \setminus \{uv \mid v \in V\}$).

In this paper we also use weighted versions of these operations, which are defined in the natural fashion. Given a weighted edge or vertex, the cost of deletion is simply that weight. Note particularly that when a vertex is deleted the cost is simply the weight of the vertex alone, not the weight of the vertex plus the weights of the incident edges, even though they are also removed (this is completely equivalent to the normal definition for unweighted graphs, where the cost of deleting a vertex is one operation, regardless of any incident edges).

2.3 Some Parameterized Complexity Theory

Here we will briefly introduce some relevant, key concepts of parameterized complexity. For a more in-depth introduction and study see the books of Downey & Fellows (1997), Flum & Grohe (2006) and Niedermeier (2006). For the sake of clarity any problem is understood to be a decision problem unless explicitly stated otherwise (and the parameterized complexity classes that are referenced are defined for decision problems).

Traditionally problems have been analyzed in one dimension, that of the size n of the input. The difficulty of solution of a problem with respect to this measure forms the fundamental basis of traditional complexity theory, and in particular the classes P and NP. Parameterized complexity adds a second measure, that of a parameter k, which is given as a special part of the input. Then, analogously to the definitions of P and NP, a series of complexity classes are defined with respect to their apparent difficulty of solution with respect to this two-dimensional measure. If a problem has an algorithm that runs in time $O(f(k)p(n))$, where p is a polynomial and f is any computable function of k, then the problem is *fixed-parameter tractable*, or in the class FPT. Naturally there are problems that are suspected not to be in FPT. These problems are members of various parameterized complexity classes, most commonly $W[t]$ for some fixed $t \geq 1$. Hardness (or completeness) in regards to such a class gives an analogous intuition to a problem being NP-hard in the classical structure. That is, it is not likely to be in FPT (i.e., not likely to have an algorithm that runs in time $O(f(k)p(n))$ as above).

Supporting this theory are many techniques for proof either of membership of FPT or of $W[t]$-hardness. Here we give a brief introduction to those techniques salient to this paper.

FPT Reductions

An FPT reduction is the parameterized complexity equivalent of a P-time many-one reduction in classical complexity theory. It is the primary method of demonstrating that two problems are of equivalent complexity, and that a particular problem is $W[t]$-hard. Given two parameterized problems Π_1 and Π_2, an FPT reduction $\Pi_1 \leq_{FPT} \Pi_2$ is a mapping from Π_1 to Π_2 that maps an instance (I, k) of Π_1 to an instance (I', k') of Π_2 such that

1. $k' = h(k)$ for some computable function h,

2. (I, k) is a YES-instance of Π_1 if and only if (I', k') is a YES-instance of Π_2 and

3. the mapping can be computed in time $O(f(k)p(|I|))$, where f is some computable function of the parameter k alone and p is a polynomial.

Then if Π_2 is in FPT, Π_1 is also in FPT and if Π_1 is $W[t]$-hard, Π_2 is also $W[t]$-hard. If two such mappings exist, one from Π_1 to Π_2 and another from Π_2 to Π_1, then the two problems are equivalent (with respect to FPT reductions) .

The classes $W[t]$, $t = 1, 2, \ldots$, are defined as equivalence classes of certain parameterized problems under FPT reductions. The classes form the chain FPT $\subseteq W[1] \subseteq W[2] \subseteq \ldots$, where all inclusions are believed to be strict.

Reduction Rules and Kernelization

One of the key techniques of parameterized complexity is that of reduction to problem kernel (kernelization). A problem is *kernelizable* if and only if given an instance (I, k) of the problem, where I is the (classical) input and k is the parameter, it is possible to produce in polynomial time an instance (I', k') where $|I'| \leq g(k')$ and $k' = h(k)$ for computable functions g and h, and (I, k) is a YES-instance if and only if (I', k') is a YES-instance. It can be shown that if a

problem is kernelizable in this sense, then it is fixed-parameter tractable (and vice versa). Kernelization is normally accomplished by the application of *reduction rules* to the instance. Estivill-Castro et al. (2005) give a recent example of the application of kernelization, along with more explanation of the theory.

2.4 A Useful Construction: The Fixing Gadget

Throughout the paper it will be useful to have a gadget that allows us to regularize any given graph. The following construction produces an almost r-regular graph, where all vertices have degree r except two with degree $r - 1$. The first part of the construction consists of a vertex c, a set $L = \{l_1, \ldots, l_r\}$ of vertices, r edges cl_i, r further vertices $M = \{m_1, \ldots, m_r\}$, and edges such that each vertex $m_i \in M$ has an edge to each vertex $l_j \in L$ except when $i = j$. Then c has degree r, as does each vertex in L. Each vertex in M has degree $r - 1$. Let C be the graph constructed so far, we then make a copy C', and add an edge between each vertex in $M \subseteq V(C)$ to its corresponding vertex in $M' \subseteq V(C')$, *except* between for m_r and m'_r. Thus each vertex now has degree r, except m_r and m'_r, which have degree $r - 1$, and will be used as attachment points. We will refer to an instance of this construction as a *fixing gadget*. See Figure 1 for an example.

Note that it is also possible to use the following as an alternative in some cases: Take the complete graph K_{r+1} on $r + 1$ vertices (so all vertices have degree r), then compute a matching (of size at most $(r + 1)/2$). Each edge of the matching can then be broken as needed to provide two edges to join the clique to the rest of the graph. This second construction cannot be used in the hardness proofs however, as it introduces (non-trivial) cliques into the graph.
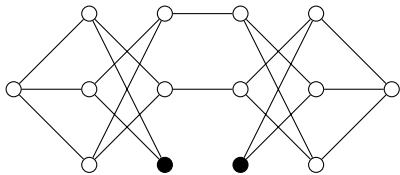

Figure 1: Fixing gadget for $r = 3$.

3 Hardness Results

The reduction for our hardness results will be from the STRONGLY REGULAR MULTI-COLOURED CLIQUE problem, a variant of the MULTI-COLOURED CLIQUE problem which was shown to be $W[1]$-hard by Fellows et al. (2007). The problem is defined as follows:

MULTI-COLOURED CLIQUE
Instance: A graph $G = (V, E)$, vertex-coloured with k colours.
Parameter: k.
Question: Does G contain a properly coloured k-clique?

This problem may be alternately defined with the original graph being properly vertex-coloured, without changing its complexity. The STRONGLY REGULAR MULTI-COLOURED CLIQUE problem is defined similarly, but with each vertex in the input graph having degree d to each colour class (so each vertex has degree kd), where d is an arbitrary integer.

Recall that the CLIQUE problem asks if a given graph has a k-clique. CLIQUE is $W[1]$-complete when parameterized by k. We then define the following special case of CLIQUE:

REGULAR CLIQUE
Instance: A regular graph $G = (V, E)$, an integer k.
Parameter: k.
Question: Does G contain a k-clique?

Before we proceed to the main result, we need first to prove some preliminary lemmas.

Lemma 3.1. REGULAR CLIQUE *is* $W[1]$-*complete.*

Proof. Membership in $W[1]$ follows immediately as the problem is a special case of CLIQUE. To prove hardness we reduce from CLIQUE. Let (G, k) be an instance of CLIQUE. We construct an instance (G', k) of REGULAR CLIQUE by first taking G and modifying it. Let Δ be the maximum degree of G, then choose r to be Δ if Δ is even, or $\Delta + 1$ otherwise (i.e., $r = \Delta + (\Delta \mod 2)$). We will now demonstrate how to make the graph r-regular. We can now use the fixing gadget construction presented in Section 2.4 to increase the degree of each vertex as necessary by attaching as many fixing gadgets as necessary by the two attachment vertices. This attachment is made between a vertex v and an instance of the fixing gadget by adding the edges between each attachment vertex and v (or perhaps only one of these edges, as below). If the degree of the vertex is initially even, then this is an integral number of fixing gadgets. In the case where the degree of the vertex is initially odd, the vertex will reach degree $r - 1$ by this method, and we will have to take another degree $r - 1$ vertex and attach one fixing gadget attachment vertex to the first, and the other attachment vertex to the second. Note that there is an even number of vertices of odd degree in G (and G' initially, an immediate corollary of the basic theorem $\sum_{v \in V} d(v) = 2|E|$), and thus there is an even number of vertices requiring an odd increase of degree (i.e., where $r - d(v)$ is odd), as we have chosen r to be even. Thus there is always some pairing of such vertices as necessary. Let G' denote the constructed graph.

Now if there were a k-clique in G, there will certainly be a clique in G' on k vertices, since G is an induced subgraph of G'. Further note that the fixing gadgets added to create G' contain no cliques, and can introduce no non-trivial cliques (as the two attachment vertices in a fixing gadget are not adjacent), thus if there is a clique on k' vertices in G', it must be contained within the vertices that correspond to the vertices of G, thus G has a k-clique. Clearly the construction of the new instance can be done in polynomial time (and thus is a polynomial-time reduction, and subsequently an FPT reduction). $\square$

Lemma 3.2. STRONGLY REGULAR MULTI-COLOURED CLIQUE *is* $W[1]$-*complete.*

Proof. Again $W[1]$ membership follows as the problem is a special case of CLIQUE.

It is useful to sketch the reduction from CLIQUE to MULTI-COLOURED CLIQUE as given by Fellows et al. (2007). Given an instance (G, k) of CLIQUE, construct an instance of MULTI-COLOURED CLIQUE (G', k') by taking k vertex disjoint copies $G_1, \ldots, G_k$ of G, assigning each G_i a different colour. Then for every pair of vertices u, v in G, if uv is an edge, add the edges $u_i v_j$, for all i, j, where a_i is the vertex in G_i corresponding to vertex a in G. Let $k' = k$. Then if there were a k-clique in the original instance, there

will be a properly coloured clique in the new instance, and vice versa.

We may use the same construction to reduce REGULAR CLIQUE to STRONGLY REGULAR MULTI-COLOURED CLIQUE. The result follows immediately. □

Theorem 3.3. VERTEX DELETION TO REGULAR SUBGRAPH *and* DELETION TO REGULAR SUBGRAPH *are* $W[1]$*-hard for parameter* k.

Proof. Consider an instance (G, k), with $G = (V, E)$, of STRONGLY REGULAR MULTI-COLOURED CLIQUE. Note that G is kd-regular and each vertex has exactly d neighbours in each colour class. We denote the set of vertices of colour i by V_i ($1 \leq i \leq k$). Then $V = \bigcup_{i=1}^{k} V_i$ forms a partition of V. Observe also that each colour class is of the same size, denote this size as s (i.e., $|V_i| = s$ for all $1 \leq i \leq k$).

We construct an instance (G', k'), with $G' = (V', E')$, of DELETION TO REGULAR SUBGRAPH by first defining k sets V_i' ($1 \leq i \leq k$) such that for each vertex $v \in V_i$ we add a vertex v' to V_i'. We add all possible edges between pairs of vertices in the same set V_i'. We will call each of these subgraphs a *colour class gadget* or *class gadget* for short.

For each edge uv in G where $u \in V_i$ and $v \in V_j$ with $i \neq j$, we add to G' two vertices $u'_{v'}$ and $v'_{u'}$, with the edges $u'u'_{v'}$, $u'_{v'}v'_{u'}$ and $v'_{u'}v'$. For each pair V_i' and V_j' (where $i \neq j$) of class gadgets, denote the set of these new vertices and edges as P_{ij}. We denote by P_{ij}^i the set of all vertices $u'_{v'} \in P_{ij}$ where $u' \in V_i'$. Furthermore, for each pair of vertices u_v and $u'_{v'}$ in the same P_{ij}^i we add the edge $u_v u'_{v'}$ to P_{ij}^i if u and u' belong to the same class gadget and $u \neq u'$. We call each such P_{ij} a *connection gadget*, and each P_{ij}^i a *side* of the connection gadget. There are $\binom{k}{2}$ connection gadgets in total. Figure 2 gives a sketch of the structure of a connection gadget.
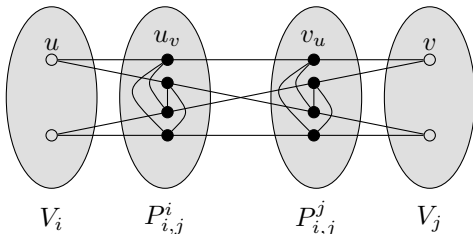

Figure 2: A sketch of illustrating the arrangement of the connection gadgets.

At this point we have k gadgets corresponding to the k colour classes in the original graph, each with s vertices of degree $(s - 1) + d(k - 1)$, and $\binom{k}{2}$ gadgets corresponding to the "inter-colour-class" edges, each with $2sd$ vertices of degree $2 + (s - 1)d$ (sd vertices in each half). Now we choose r for the instance such that $r \geq \max((s - 1) + d(k - 1), 2 + (s - 1)d)$, and $r \equiv s + 1$ modulo 2 (i.e., r is of opposite parity to s). In particular we may choose the smallest r such that this is true.

Now we add for each class gadget V_i' a gadget V_i'' that contains $r + 1 - ((s - 1) + d(k - 1))$ vertices with s edges per vertex, such that each vertex in V_i'' is adjacent to every vertex in the class gadget V_i'. We refer to V_i'' as a *degree gadget*. We then add a further set of fixing gadgets as before to complete the degree of each vertex in the degree gadget to $r + 1$. Note

that by choosing r to have opposite parity to s, we guarantee that this is possible (if s is odd, r will be even and each vertex will require $r + 1 - s$ additional edges, which is even, and thus achievable; if s is even, r will be odd, then $r + 1 - s$ is again even, and we can complete the construction). Thus each vertex in each class gadget and degree gadget has degree one too many, but the fixing gadgets attached to each degree gadget have the correct degree.

We similarly adjust the connection gadgets by adding two degree gadgets, each with $r + 1 - 2 + (s - 1)d$ vertices, one for each side of the connection gadget. Every vertex in the degree gadget is connected to every vertex in its associated side of the connection gadget. Again we complete the degree of vertices in the degree gadgets to $r + 1$ by adding fixing gadgets, and as before, by the choice of r we can guarantee that this can be done (if s is even, r is odd and $r + 1 - sd$ is even, if s is odd, r is even and $r + 1 - sd$ is even). Thus each vertex in the connection gadgets has degree $r + 1$, as does each vertex in the degree gadgets. Each vertex in each fixing gadget has degree r.

Now we set $k' = k + 2\binom{k}{2}$.

Claim 3.1. The following statements are equivalent:

1. (G, k) is a YES-instance of STRONGLY REGULAR MULTI-COLOURED CLIQUE.

2. (G', k') is a YES-instance of VERTEX DELETION TO REGULAR SUBGRAPH.

3. (G', k') is a YES-instance of DELETION TO REGULAR SUBGRAPH.

$(1 \Rightarrow 2)$ Assume that (G, k) is a YES-instance of STRONGLY REGULAR MULTI-COLOURED CLIQUE. Then there exist k vertices $v_1, \ldots, v_k$, one from each colour class, that form a properly coloured clique. Assume without loss of generality that $v_i \in V_i$. Then we can delete from G' the corresponding vertices v_i' from V_i', and the pairs of vertices $(v_i')_{v_j'}$ and $(v_j')_{v_i'}$ from P_{ij} that correspond to the edges in the clique. Then each remaining vertex in each class gadget has had precisely one incident edge removed from it, as have the vertices in each degree gadget associated with the class gadget. So the components corresponding to the colour classes and their immediate extension are now r-regular. Similarly each vertex in every connection gadget and their associated connection gadgets has had exactly one incident edge removed, either by the vertex removed from the connection gadget, or from the parent vertex in the class gadget (but never both). Now each vertex in these gadgets has degree precisely r. We have chosen one vertex from each V_i', and two vertices from each P_{ij}, giving a total of $k' = k + 2\binom{k}{2}$ vertices, thus (G', k') is also a YES-instance of VERTEX DELETION TO REGULAR SUBGRAPH.

$(2 \Rightarrow 3)$ Assume that (G', k') is a YES-instance of VERTEX DELETION TO REGULAR SUBGRAPH. Then clearly it is also a YES-instance of DELETION TO REGULAR SUBGRAPH.

$(3 \Rightarrow 1)$ Assume that (G', k') is a YES-instance of DELETION TO REGULAR SUBGRAPH. Then there are $k + 2\binom{k}{2}$ deletions that can be made to make G' r-regular. Obviously we cannot delete any vertices from the fixing gadgets in the graph. Further we cannot delete any vertices from the degree gadgets, as this would reduce the degree of their attached fixing gadgets. Thus the deleted vertices must come from class and connection gadgets. Again there must be precisely one vertex from each such component, if there is less than one in such a component, the degree of at least some of the vertices in that component will

remain $r + 1$, if there are more than one, the degree of some vertices in the component will drop below r. Also note that for each vertex u_v deleted from one side of a connection gadget, the vertex deleted in the other side must be the vertex v_u. If it were not, then at least one vertex in each side would have degree $r - 1$. Also, the vertex deleted from each side of each connection gadget must be attached to the vertex deleted from the adjacent class gadget, otherwise the vertices attached to vertices deleted from the class gadget will have degree at most $r - 1$. Thus we can see that if (G', k') is a YES-instance, the set of vertices to be deleted is very precise and restricted. In fact, if we are to use only the allotted budget of $k + 2\binom{k}{2}$, we must choose precisely one vertex from each class gadget, and two vertices from each connection gadget, where the vertices from the connection gadget component are connected to the vertices deleted from the two class gadgets it is associated with. Similarly, assume that some edge deletion is used, but then each edge deletion can only reduce the degree of two vertices, leaving us with too many edges to delete, or vertices of degree less than r. So clearly the only operation that can be used in this case is vertex deletion. Thus we may more precisely claim that if (G', k') is a YES-instance for DELETION TO REGULAR SUBGRAPH, it must be via vertex deletion alone. Thus it is clear that if (G', k') is a YES-instance for DELETION TO REGULAR SUBGRAPH, then (G, k) must be a YES-instance of STRONGLY REGULAR MULTI-COLOURED CLIQUE. The solution for (G, k) is the set of vertices $\{v_1, \ldots, v_k\}$, one from each colour class, corresponding to the k vertices chosen from the class gadgets. The edges of the clique correspond to the $2\binom{k}{2}$ vertices chosen from the connection gadgets.

We can construct G' from G in polynomial time, as we are adding only $(4r + 3)(2r + 2s - s - sd - dk + 1)$ vertices, where $r, s, d \leq n$, thus it is also an FPT reduction, and we have the desired result. $\qquad\square$

We also note that the above proof suffices if we also include the operation of edge addition, giving the following:

Corollary 3.4. EDIT TO REGULAR SUBGRAPH *parameterized by the number of* edit *operations k is $W[1]$-hard.*

We may also consider the similar problem of finding a regular subgraph of an unknown regularity (i.e., when r is not given):

Corollary 3.5. DELETION TO SOME REGULAR SUBGRAPH *parameterized by the number of edit operations k is $W[1]$-hard.*

Proof. Given an instance (G, k) of DELETION TO REGULAR SUBGRAPH, we construct an instance (G', k) of DELETION TO SOME REGULAR SUBGRAPH as follows:

We simply add one r-regular connected component with more than k vertices. This can be done by taking, for example, k fixing gadgets and connecting them in a ring. We clearly cannot alter this component within the budget, thus the only possible solution is the same as that for (G, k). Thus if (G', k) is a YES-instance of DELETION TO SOME REGULAR SUBGRAPH, (G, k) must be a YES instance of DELETION TO REGULAR SUBGRAPH. Naturally if (G, k) is a YES-instance of DELETION TO REGULAR SUBGRAPH, the same solution will result in a regular graph in (G', k), so (G', k) is a YES-instance of DELETION TO SOME REGULAR SUBGRAPH. $\qquad\square$

Of course the same proof again suffices for the edit version of the problem.

We also obtain the following result.

Corollary 3.6. WEIGHTED DELETION TO CHOSEN DEGREE SUBGRAPH *parameterized by the number k of edit operations is $W[1]$-hard.*

Proof. Clearly DELETION TO REGULAR SUBGRAPH, is a restriction of WEIGHTED DELETION TO CHOSEN DEGREE SUBGRAPH, with $\rho(e) = 1$, $\rho(v) = 1$ and $\delta(v) = r$ for each edge e and vertex v. $\qquad\square$

Once again we may make a similar claim for the edit version of the problem, WEIGHTED EDIT TO CHOSEN DEGREE SUBGRAPH.

4   Fixed Parameter Tractability

Moser and Thilikos (2006) give several tractability results for regular induced subgraph problems, and in doing so contribute several significant and natural ideas that are of use in the more general setting of this paper. Several of the reduction rules that we develop have direct analogs in their paper, and in particular we use their notion of a "clean region". We however exploit the structure available more fully, using annotation. In this case annotation proves a powerful tool for generalizing, and thus simplifying the problem. We are thus able to get more general results that include their results as special cases. In particular we avoid the complex clean region replacement that they undertake as the annotation allows a simpler representative replacement. Abu-Khzam & Fernau (2006) give a further examination of annotation with respect to kernelization.

4.1   Definitions

First we will define various terms that allow a more elegant treatment of the result.

Given a graph G, a function $\delta : V \to \{0, \ldots, r\}$ and a function $\rho : V \cup E \to \mathbb{N}^+$. We say a vertex $v \in V$ is *clean* if $\sum_{e \in E(v)} \rho(e) = \delta(v)$. Then a *clean region* is a set of *clean* vertices that form a connected subgraph. Note that not all edges incident on the vertices of the clean region need have both endpoints in the clean region. We can greedily calculate the collection of maximal clean regions in a graph in polynomial time. Note that these maximal clean regions are disjoint. In general when we refer to a clean region, we will mean a maximal clean region, though strictly the results are unaffected.

A clean region is *independent* if there are no edges from the clean region to any vertex outside the clean region.

Given a clean region C we call the set of vertices not in C adjacent to a vertex in C as the *boundary* of C.

It is also notationally convenient to define the degree of a vertex v restricted to a set X of vertices as $d_X(v)$. So $d_X(v)$ is the number of neighbours of v that are in the set X, and we extend this notation to sets of vertices, for example the degree of a boundary B restricted to its clean region C is $d_C(B) = \sum_{b \in B} d_C(b)$.

It is also useful to define a weighted degree function $d^\rho : V \to \mathbb{N}^+$ such that for each vertex v, $d^\rho(v) = \sum_{e \in E(v)} \rho(e)$. As above we denote the weighted degree of a vertex v restricted to a set of vertices X as $d_X^\rho(v)$ and extend it as before to sets.

4.2 Weighted Deletion to Chosen Degree Subgraph

In this section we consider the WEIGHTED DELE-TION TO CHOSEN DEGREE SUBGRAPH problem as defined earlier, but parameterized by both the number of deletions k and the maximum desired degree r.

4.2.1 Reduction Rules

Let $(G, (k, r))$, with $G = (V, E)$, be an instance of WEIGHTED DELETION TO CHOSEN DEGREE SUBGRAPH. The following reduction rules produce from $(G, (k, r))$ an equivalent instance $(G', (k', r'))$ of DELETION TO CHOSEN DEGREE SUBGRAPH. For all reduction rules $r' = r$.

Reduction Rule 1: If there exists a vertex $v \in V$ with $d^\rho(v) < \delta(v)$, then $G' = G - \{v\}$, $k' = k - \rho(v)$.

Reduction Rule 2: If there exists a vertex $v \in V$ with $d(v) > k + r$, then $G' = G - \{v\}$, $k' = k - \rho(v)$.

Reduction Rule 3: If there exists an independent clean region $C \subseteq V$, then $G' = G - C$, $k' = k$.

Reduction Rule 4: If there is a clean region C with a vertex b in its boundary where $d_C^\rho(b) > \delta(b)$, then $G' = G - C$, $k' = k - \sum_{v \in C} \rho(v)$.

This also gives an algorithmically useful corollary.

Corollary 4.1. *If there exists a clean region C with a vertex b in its boundary such that $d_C^\rho(b) > \delta(b)$ and $\sum_{v \in C} \rho(v) > k$, then $(G, (k, r))$ is a NO-instance.*

Reduction Rule 5: If there exists a clean region C with boundary B such that $\sum_{v \in C} \rho(v) > k$ and for each boundary vertex b we have $d_C^\rho(b) \leq \delta(b)$, then for each $b \in B$, set $\rho(b) = k + 1$ and set $\delta(b) = \delta(b) - d_C^\rho(b)$, and $G' = G - C$, $k' = k$.

Reduction Rule 6: If there exists a clean region C, with boundary B such that Reduction Rules 4 and 5 do not apply, (i.e., there are no boundary vertices with excessive weighted degree into the clean region, and the weight of the clean region is not larger than k), then modify the instance as follows:

1. Add a new vertex v such that $\rho(v) = \sum_{c \in C} \rho(c)$, and $\delta(v) = d_B^\rho(C)$.

2. For each boundary vertex $b \in B$, add an edge bv such that $\rho(bv) = d_C^\rho(b)$.

3. Delete C.

4. Set $k' = k$.

Note that the vertex v added in Reduction Rule 6 is a special vertex in that we allow it to have $\rho(v) > r$. This does not affect the existence of a solution, and if desired, a less elegant, alternate reduction rule can be substituted where the region is replaced by a series of vertices each with ρ at most r. In the kernelization this increases the size of X (only) by a factor of at most k.

Lemma 4.2. *Reduction Rules 1–6 are sound. That is, each reduction rule takes an instance $(G, (k, r))$ of WEIGHTED DELETION TO CHOSEN DEGREE SUBGRAPH and produces an instance $(G', (k', r'))$ of WEIGHTED DELETION TO CHOSEN DEGREE SUBGRAPH such that $(G, (k, r))$ is a YES instance if and only if $(G', (k', r'))$ is a YES instance.*

Proof. Rule 1: Clearly v cannot remain in the final graph unmodified, but as we cannot add edges, there is no way of increasing the degree of v. Thus v must be deleted as part of any solution.

Rule 2: If v were to remain in the final graph, we must either delete more than k edges or neighbouring vertices, each with weight at least 1, which we cannot do. Thus the only possibility is to delete v.

Rule 3: Clearly an independent clean region needs no changing, thus we can safely ignore it, as it will play no role in the solution.

Rule 4: If there were such a b, then at least one of the edges from b into the clean region must be deleted, but then a vertex v of the clean region would now have weighted degree less than $\delta(v)$, and would have to be deleted (as per Reduction Rule 1). This would obviously cascade, resulting in the entire clean region being deleted. Thus the only possible option is to delete the clean region.

Rule 5: As with Reduction Rule 4, deletion of any vertex or edge in the clean region or between the clean region and the boundary would require the clean region to be deleted entirely. As the clean region is of total weight greater than k, it obviously cannot be deleted within a cost k solution. Thus it suffices to increase the weight of each vertex in the boundary, as these cannot be deleted either, and reduce their degree funtion appropriately.

Rule 6: As C is a clean region, deletion of any vertex or edge in the clean region or boundary will result in the entire clean region being deleted, thus it is sufficient to represent the clean region as one appropriately weighted clean vertex. $\qquad\square$

4.2.2 Kernel Lemma

Lemma 4.3 (Kernel Lemma). *If $(G = (V, E), (k, r))$ is reduced under Reduction Rules 1–6 and $|V| > k + k(k + r) + kr(k + r)$, then $(G, (k, r))$ is a NO-instance for WEIGHTED DELETION TO CHOSEN DEGREE SUBGRAPH.*

Proof. Assume that $(G = (V, E), (k, r))$ is a YES-instance for k-DELETION r-REGULAR SUBGRAPH. Further assume that the instance is reduced under Reduction Rules 1–6. Let S be the set of edges and vertices deleted as part of the solution, $|S| \leq k$ (more particularly $\rho(S) \leq k$). As any edge in the solution is adjacent to only two vertices, our worst case occurs when the solution is all vertices, so it suffices to only consider S. Further let H be the set of vertices consisting of the endpoints of any edges in S and the neighbours of any vertices in S, and $X = V \setminus \{H \cup S\}$. Note that H is a cut-set seperating S and X. Figure 3 gives an example of this partitioning for an example graph with $r = 3$.

We make the following claims:

Claim 4.1. $|H| \leq k(k + r)$.

No vertex has degree greater than $(k + r)$, otherwise the graph is not reduced under Reduction Rule 2. Thus if S were all vertices, they could have at most $(k + r)$ neighbours each. As H is the entire neighbourhood of S, $|H| \leq |S|(k + r) = k(k + r)$.

Claim 4.2. $|X| \leq kr(k + r)$.

X must consist only of clean regions, otherwise S is not a solution. Each vertex h in H can have at most r neighours in X, otherwise S is not a solution. If h were adjacent to a clean region with total weight greater than k, this region would have been removed under Reduction Rule 5, thus it can only be adjacent to small clean regions. As the graph is reduced, each of these clean regions contains precisely one vertex,
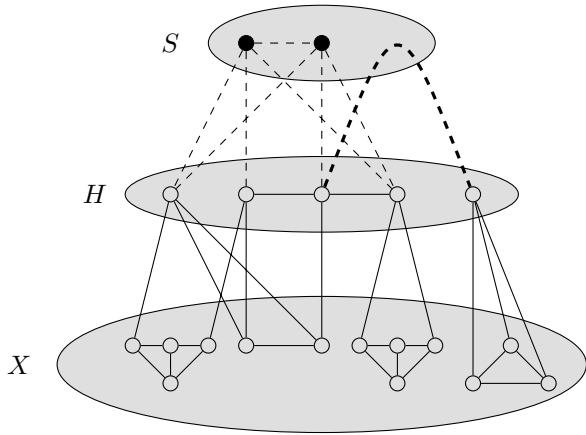
6

Figure 3: Example of the partitioning described in the Kernel Lemma. $r = 3$.

by Reduction Rule 6. Thus each h can have at most r neighbours. As there are $k(k + r)$ such vertices, $|X| \leq kr(k + r)$.

Claim 4.3. There are at most $k + k(k+r) + kr(k+r)$ vertices in G.

$|V| = |S| + |H| + |X|$. By Claims 4.2 and 4.1, $|H| \leq k(k + r)$, and $|X| \leq kr(k + r)$. There are no other vertices in the graph, otherwise the graph is not reduced under Reduction Rules 3 and 4. Thus as $|S| \leq k$, $|V| \leq k + k(k+r) + kr(k+r)$.

Then by Claim 4.3, if $(G = (V, E), (k, r))$ is a YES-instance for WEIGHTED DELETION TO CHOSEN DEGREE SUBGRAPH, then $|V| \leq k + k(k+r) + kr(k+r)$. Thus the Kernel Lemma holds. □

To complete the proof of FPT membership, we need to demonstrate that the Reduction Rules can be executed in polynomial time. Clearly Reduction Rules 1 and 2 can be carried out in linear time, and each can be applied at most k times. Thus Reduction Rules 1 and 2 contribute $O(kn)$ to the running time.

We can calculate the clean regions of the graph greedily in linear time, with the boundary calculated at that time. Thus an independent clean region can be identified in linear time, and deleted in linear time. Similarly any clean region with a vertex b in its boundary such that $d_C^\rho(b) > \delta(b)$, can be identified quickly. Similarly regions to which Reduction Rules 5 and 6 apply can be identified at this point, and replaced or removed as required. This can be done at most k times, thus Reduction Rules 3, 4, 5 and 6 contribute $O(kn)$ to the running time.

This leads immediately to the following theorem:

Theorem 4.4. WEIGHTED DELETION TO CHOSEN DEGREE SUBGRAPH *is fixed-parameter tractable for parameter (k,r).*

In particular an instance $(G, (k, r))$ of WEIGHTED DELETION TO CHOSEN DEGREE SUBGRAPH with n vertices and m edges can be solved in time $O(kn + f(k, r))$, where $f(k, r)$ is the running time of whatever algorithm or heuristic is applied to the kernel (whose size is bounded, so the running time is guaranteed to be a function of (k, r). A simple approach would be the application of a bounded search tree which branches on which problem vertex or edge to delete, which gives a running time of $O((k^3 + 2k^2r + kr^2)^k)$.

Note that if we have a graph where initially $\rho(v) = 1$, $\rho(e) = 1$ and $\delta(v) = r$ for each vertex v and edge e, then this is precisely the DELETION TO REGULAR SUBGRAPH problem, thus we also gain the following result:

Corollary 4.5. DELETION TO REGULAR SUBGRAPH *is fixed-parameter tractable for parameter (k,r).*

4.2.3 The Exact Case

The previous proof can be modified easily to demonstrate fixed-parameter tractability for the EXACT WEIGHTED DELETION TO CHOSEN DEGREE SUBGRAPH problem. In this case we are interested in deleting elements with a total weight of k. Thus we may be interested in deleting elements where the deletion does not fix the degree of some vertex, it simply adds to the total cost. However, the only areas where we may delete these from that are not already included in the kernel are independent clean regions of 'low' weight ($\leq k$).

Of course we need not retain all such independent clean regions on the chance that they may be needed. Obviously any independent clean region of weight greater than k can still be removed without consequence, it could never be part of any solution of cost k. So we need only concern ourselves with independent clean regions of weight less than or equal to k. Recall that a clean region is defined as a set of vertices, thus in particular, the weight of a clean region is the sum of weights of the vertices, not the edges.

Notice also that given a sufficient quantity of independent clean regions of a given weight (say i), we could never use all of them, and thus need only retain a small number. Thus if we replace Reduction Rule 3, we can adjust our kernel size appropriately:

Reduction Rule 3a: If there exist more than $\lfloor k/i \rfloor$ independent clean regions of weight $i \leq k$, delete all but $\lfloor k/i \rfloor$ of them, $k' = k$.

Then we may modify the Kernel Lemma as follows:

Lemma 4.6 (Exact Kernel Lemma)**.** *If $(G = (V, E), (k, r))$ is reduced under Reduction Rules 1–6 (with Rule 3a replacing Rule 3) and $|V| > k + k(k + r) + kr(k+r) + k^2$, then $(G, (k, r))$ is a NO-instance for* EXACT WEIGHTED DELETION TO CHOSEN DEGREE SUBGRAPH.

Proof. We begin with the following claims:

Claim 4.4. There are no independent clean regions of size greater than k.

As each vertex has weight at least 1, a clean region of size greater than k must have weight greater than k, thus if one remained, the graph would not be reduced under Reduction Rule 3a.

Claim 4.5. There are at most k^2 vertices in independent clean regions.

By Reduction Rule 3a, there are at most $\lfloor k/i \rfloor$ independent clean regions of weight i. The largest size of any such region is k (by claim 4.4). Thus the total size is $\sum_{i=1}^{k} \lfloor k/i \rfloor \, i \leq \sum_{i=1}^{k} (k/i)i = \sum_{i=1}^{k} k = k^2$.

The proof of the Kernel Lemma now follows as before, simply with the new claims taken into account. □

The new reduction rule can clearly be enacted in polynomial time, we need only greedily keep the first few independent clean regions we find of each weight, which can be accomplished at the start of the algorithm in linear time.

Theorem 4.7. EXACT WEIGHTED DELETION TO CHOSEN DEGREE SUBGRAPH *is fixed-parameter tractable for parameter (k, r).*

As before, we also gain the following result as a special case:

Corollary 4.8. EXACT DELETION TO REGULAR SUBGRAPH *is fixed-parameter tractable for parameter* (k, r).

5 Conclusion

We have answered Moser and Thilikos's open question, and shown that VERTEX DELETION TO REGULAR SUBGRAPH is $W[1]$-hard when parameterized by the number k of vertex deletions. The problem remains hard when we extend the operations available to include edge deletion and/or edge addition. The generalized version of the problem WEIGHTED DELETION (EDIT) TO CHOSEN DEGREE SUBGRAPH is also $W[1]$-hard when parameterized by the number k of deletions.

If we include r as an additional parameter however, all the problems examined are fixed-parameter tractable, most notable being that WEIGHTED DELETION TO CHOSEN DEGREE SUBGRAPH is fixed-parameter tractable under such a parameterization with a problem kernel of at most $k+k(k+r)+kr(k+r)$ vertices. DELETION TO REGULAR SUBGRAPH is also fixed-parameter tractable with the same kernel, but we also demonstrate a method that allows avoidance of clean region contraction, which may be useful in practice. Similarly the exact versions of the problems remain tractable with the same parameterization.

As our FPT results derive from kernelization, this paper also provides several useful polynomial-time preprocessing algorithms producing bounded problem kernels which can then be solved by any method of choice, such as heuristics or approximations.

References

Abu-Khzam, F. & Fernau, H. (2006), Kernels: Annotated, Proper and Induced, *in* 'International Workshop on Parameterized and Exact Computation 2006 (IWPEC'06)', Lecture Notes in Computer Science, Springer, pp. 264–275.

Cheah, F. & Corneil, D. G. (1990), The Complexity of Regular Subgraph Recognition, 'Discrete Applied Mathematics', **27**, pp. 59–68.

Chvátal, V., Fleischner, H., Sheehan, J. & Thomassen, C. (1979), Three-regular Subgraphs of Four Regular Graphs, 'Journal of Graph Theory', **3**, pp. 371–386.

Downey, R. & Fellows, M. (1997), *Parameterized Complexity*, Springer.

Estivill-Castro, V., Fellows, M., Langston, M. & Rosamond, F. (2005), FPT is P-Time Extremal Structure I, *in* 'Algorithms and Complexity in Durham 2005 (ACiD'05)', Texts in Algorithmics, College Publications, pp. 1–41.

Flum, J. & Grohe, M. (2006), *Parameterized Complexity Theory*, Springer.

Guo, J. & Niedermeier, R. (2007), Invitation to Data Reduction and Problem Kernelization, 'SIGACT News', **38**(1), pp. 31–45.

Fellows, M., Hermelin, D. & Rosamond, F. (2007), On the Fixed-Parameter Intractability and Tractability of Multiple-Interval Graph Problems, Unpublished Result.

Moser, H. & Thilikos, D. (2006), Parameterized Complexity of Finding Regular Induced Subgraphs, *in* 'Algorithms and Complexity in Durham 2006 (ACiD'06)', Texts in Algorithmics, College Publications, pp. 107–118.

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

Plesník, J. (1984), A Note on the Complexity of Finding Regular Subgraphs, 'Discrete Mathematics', **49**, pp. 161–167.

Stewart, I. A. (1994), Deciding Whether a Planar Graph has a Cubic Subgraph is NP-Complete, 'Discrete Mathematics', **126**(1-3), pp. 349-357.

Stewart, I. A. (1996), Finding Regular Subgraphs in Both Arbitrary and Planar Graph, 'Discrete Applied Mathematics', **68**(3), pp. 223-235.

Stewart, I. A. (1997), On Locating Cubic Subgraphs in Bounded-degree Connected Bipartite Graphs, 'Discrete Mathematics', **163**(1-3), pp. 319-324.

Stewart, I. A. (2007), On the Fixed-Parameter Tractability of Parameterized Model-Checking Problems, 'Information Processing Letters', *to appear*.