



A kernelization algorithm for d -Hitting Set ^{☆, ☆☆}

Faisal N. Abu-Khzam

Department of Computer Science and Mathematics, Lebanese American University, 475 Riverside Drive, Suite 1846, New York, NY 10115-0065, USA

ARTICLE INFO

Article history:

Received 13 April 2009

Received in revised form 19 August 2009

Available online 23 September 2009

Keywords:

Fixed-parameter algorithms

Hitting Set

Kernelization

Crown decomposition

Vertex cover

Hypergraphs

ABSTRACT

For a given parameterized problem, π , a kernelization algorithm is a polynomial-time pre-processing procedure that transforms an arbitrary instance of π into an equivalent one whose size depends only on the input parameter(s). The resulting instance is called a problem kernel. In this paper, a kernelization algorithm for the 3-Hitting Set problem is presented along with a general kernelization for d -Hitting Set. For 3-Hitting Set, an arbitrary instance is reduced into an equivalent one that contains at most $5k^2 + k$ elements. This kernelization is an improvement over previously known methods that guarantee cubic-order kernels. Our method is used also to obtain quadratic kernels for several other problems. For a constant $d \geq 3$, a kernelization of d -Hitting Set is achieved by a non-trivial generalization of the 3-Hitting Set method, and guarantees a kernel whose order does not exceed $(2d - 1)k^{d-1} + k$.

© 2009 Elsevier Inc. All rights reserved.

1. Introduction

Let C be a collection of subsets of a finite set S . A hitting set of C is a subset of S that has a non-empty intersection with each element of C . Identifying small hitting sets has a wide range of applications in a variety of domains that include Bioinformatics, Computer Networks as well as Software Testing [13,14,19]. Formally, the decision version of the Hitting Set problem can be defined as follows:

Given: A collection C of subsets of a set S and a parameter k .

Question: Does C have a hitting set of size k or less?

Hitting Set is \mathcal{NP} -complete, even if the cardinality of every element of C is bounded by two [11]. From the viewpoint of parameterized complexity, Hitting Set is $\mathcal{W}[2]$ -hard in general [8]. However, the problem becomes fixed-parameter tractable (\mathcal{FPT}) when the cardinality of every element of C is upper-bounded by a fixed number d . In this latter case, the problem is called d -Hitting Set (henceforth d HHS). Many fixed-parameter algorithms appeared that address d HHS problems. 3HHS, in particular, has been a focal point of attention in recent Hitting Set algorithms [9,17]. This paper is concerned mainly with the parameterized search version of d HHS.

On the other hand, kernelization has become a major pre-processing tool for solving problems that fall in the class \mathcal{FPT} . For an arbitrary instance of an \mathcal{FPT} problem, a kernelization procedure is a polynomial-time pre-processing algorithm that could result in one of the following outputs:

- An early detection of failure;

[☆] This research has been supported in part by the Lebanese American University under grant URC-c-2004-63.

^{☆☆} A preliminary version of this paper appeared in Proceedings of the 10th International Workshop on Algorithms and Data Structures (WADS 2007).

E-mail address: faisal.abukhzam@lau.edu.lb.

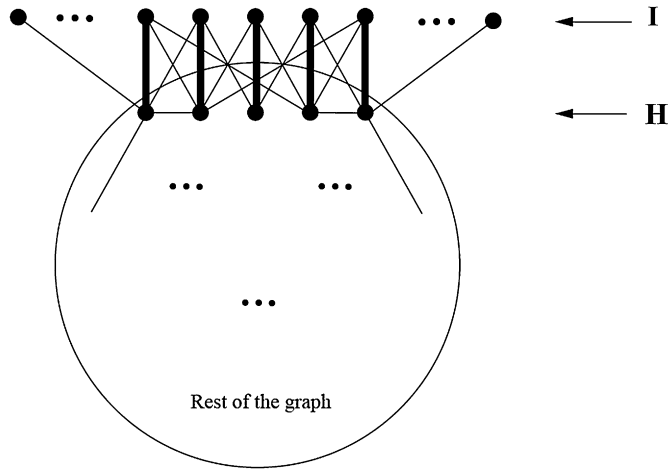


Fig. 1. A crown decomposition of width 5. Bold edges are elements of M .

- An early solution;
- An equivalent instance whose size is bounded by a function of the input parameter.

In the third case, the function is often polynomial in the input parameter(s), and the resulting reduced instance is called a problem kernel.

3HS is known to have a cubic-size kernel obtained by simple counting arguments [17]. In this paper, we show how to obtain a quadratic-order kernel. In addition to counting techniques, our method extends the notion of crown reduction to hypergraphs. Crown reduction was introduced by the authors of [6] and developed further in [1,2] in the context of kernelization for Vertex Cover. We show how our crown reduction method extends to a general d -Hitting Set kernelization.

2. Background

We shall treat the input, (S, C) , of a hitting set instance as a hypergraph whose vertices and edges are the elements of S and C respectively. This allows us to use a few known graph-theoretic terminologies. We use $N_H(A)$ to denote the set of neighbors of A in H , where A is a set of vertices and H is a graph or subgraph. We shall use the term “order” to denote the number of vertices (or elements of S). An independent set of a hypergraph is a set of vertices that are pairwise non-adjacent, in the sense that no two of them belong to an element of C . Moreover, a hitting set of (S, C) is a vertex cover of the corresponding hypergraph. We shall say that a subset A of S **hits** a subset C' of C if every element of C' has a non-empty intersection with A . For each vertex $x \in S$, we denote by $E(x)$ the set of all edges containing x . For $E \subset C$, $V(E)$ denotes the set of all vertices that are elements of elements of E . Finally, our algorithmic analysis assumes that a hypergraph is represented by an incidence matrix, associated with a linear array keeping the degrees of all vertices.

The following two pre-processing rules, dubbed **domination rules**, are used by previously known d HS kernelization algorithms. They first appeared in [21].

- **Vertex Domination Rule:** If $x, y \in S$ are such that $E(x) \subset E(y)$, then delete x .
- **Edge Domination Rule:** If $e_1, e_2 \in C$ are such that $V(\{e_1\}) \subset V(\{e_2\})$ then delete e_2 .

Our kernelization will be based on the notion of **crown decomposition**: a recent FPT technique that was introduced by Fellows et al. in [6] and was further studied in conjunction with the Vertex Cover problem in [1,2]. In short, a crown decomposition (or just crown) of a simple undirected graph G is a triple (H, I, M) such that:

- I is an independent set of G .
- H is the neighborhood of I in G .
- M is a matching in G that satisfies:
 - (i) Every edge of M joins a vertex from H to a vertex of I .
 - (ii) Every vertex of H is matched, under M , to a vertex of I .

The set H is called the head of a crown (H, I, M) and $|H|$ is the crown width. Fig. 1 illustrates the notion of a crown decomposition.

Deciding whether a graph has a crown decomposition is solvable constructively in polynomial time [2]. The following lemma provides a necessary and sufficient condition for a graph to have a crown. It can be deduced from [6, p. 7].

Lemma 1. A graph G has a crown decomposition of positive width if and only if G has an independent set I that satisfies $|N_G(I)| \leq |I|$.

Once an independent set I that satisfies the condition of Lemma 1 is found in $G = (V, E)$, a crown can be constructed in time $O(|V|^{2.5})$ by a procedure provided also in [6]. We shall refer to this procedure by *Construct_Crown* in this paper.

Throughout this paper, the main focus is on how to obtain an upper bound on the order of a 3HS kernel. Therefore, we shall omit the complete details of efficiency analysis whenever it is clear that the run time is polynomial in the input size, which is in $\theta(|S| + |C|)$.

3. A quadratic kernel for 3-Hitting Set

Our kernelization algorithm consists of a number of reduction steps that are based on counting techniques, and the notion of crown reduction. We start by generalizing the notion of crown decomposition to hypergraphs.

A crown in a hypergraph (S, C) is defined as a tripe (H, I, M) such that:

- I is an independent set of (S, C) .
- $H = \{J \subset S : J \cup \{x\} \in C \text{ for some } x \in I\}$.
- M is a matching in the bipartite graph $(I \cup H, E)$ where $E = \{(x, J) : x \in I \text{ and } J \in H \text{ and } (\{x\} \cup J) \in C\}$.
- Every element of H is matched under M .

The crown reduction rule for 3HS consists of deleting all elements of I without making any further changes to C and k . The elements of H will automatically become edges of C . The soundness of this rule is based on the following lemma.

Lemma 2. Let (S, C, k) be an instance of 3HS and let (H, I, M) be a crown of (S, C) . Then (S, C, k) is equivalent to the instance $(S - I, (C - E(I)) \cup H, k)$.

Proof. Let A be a hitting set of size k of (S, C) and let B be the subset of H that consists of pairs that have non-empty intersection with A . Then every pair $\{y, z\}$ of $H - B$ is a subset of an edge $\{x, y, z\}$ of M , where $x \in A \cap I$ is used to hit $\{x, y, z\}$. Replacing every such x by an element of its matched pair $\{y, z\}$ produces another hitting set whose size is bounded above by $|A|$. (It could be smaller than A when the pairs of $H - B$ have common elements that belong to a solution of (S, C, k) .) \square

In addition to the use of hypergraph crowns, our kernelization algorithm makes use of the following new reduction rule, which generalizes the high-degree rule used in the Vertex Cover kernelization of [5].

High-Degree rule: If $x \in S$ has more than k edges whose pairwise intersection is $\{x\}$, then:

- add x to any potential solution of (S, C, k) ,
- delete all elements of $E(x)$, and
- decrease k by 1.

For $x \in S$, let G_x denote the simple graph obtained from $E(x)$ by deleting x . Applying the high-degree rule would be equivalent to finding a matching of size $k + 1$ or more in G_x . The corresponding run time is in $O(|N(x)|^{2.5})$ for each x in S . We can save this time by restricting the application of the rule to a special subset of C , as will be described in the next subsection.

We shall see that a crown can be constructed in any 3HS instance as long as its order exceeds $5k^2 + k$. In such cases, Lemma 2 is applied to produce the desired kernel.

3.1. Weakly related edges

Let (S, C, k) be a 3HS instance. Two elements of C are said to be weakly related if they don't share more than one common vertex. A subset W of C is a maximal collection of (pair-wise) weakly related edges if every edge of C is either in W or shares at least two vertices with at least one element of W . Weakly related edges play a central role in our algorithm, mainly due to the following lemma.

Lemma 3. Let (S, C, k) be a yes instance of 3HS and let W be a collection of weakly related edges of (S, C) , then $|W| \leq k^2$.

Proof. At most k vertices hit all elements of W . By the high-degree rule and the definition of W , each such vertex belongs to at most k edges of W . \square

A maximal set of weakly related edges can be constructed by starting with a singleton subset W of C and greedily filling it with elements of C as long as it satisfies the above condition. The corresponding run-time is in $O(|C||W|)$. The construction can be made more efficient if a lexicographical ordering of the edges is used, provided $|W|$ is kept sorted during its construction.

We shall assume that edges of size two are placed in W before any edge of size three. Moreover, we keep track of the number of edges in $E(x) \cap W$ for each element $x \in S$. If, at any point, $|E(x) \cap W|$ exceeds k , then we apply the high-degree rule. This action results in deleting all of $E(x)$, which may violate the maximality of W . Therefore, whenever the high-degree rule is applied, the elements of C must be re-checked. Since k is decremented along with the deletion of $E(x)$, this reduction operation adds at most a factor of k to the total run-time of W 's construction. Finally, we use the result of Lemma 3 by halting the construction process whenever $|W|$ exceeds k^2 . It follows that the overall run time of W 's construction is in $O(k^3|C|)$.

3.2. A crown kernelization of 3HS

Again, we assume that (S, C, k) is a yes instance of 3HS. Let W be any maximal set of weakly related edges, $I = \{x \in S: x \notin V(W)\}$, and $H = \{\{x, y\}: \{x, y\} \subset e \text{ for some } e \in W\}$. Then we observe the following:

- All elements of C whose cardinality is less than three belong to W (by construction).
- I is an independent set.

To see this, note that any edge e that contains two elements of I qualifies for membership in W . This would violate the assumed maximality of W .

- $S \setminus I$ has at most $2k^2 + k$ elements.

This is guaranteed by the existence of a solution A of size $\leq k$ that hits all the elements of W . Each element x of A belongs to at most k elements of W . So the restriction of $E(x)$ to W comprises at most $2k$ vertices. In addition to the elements of A , there are at most $2k^2$ vertices that belong to $\bigcup_{x \in A} E(x)$.

- Every edge e of size three that contains an element x of I consists of x and a pair from H .

Otherwise, e could be added to W , violating W 's maximality.

- $|H| \leq 3k^2$.

Each edge of size three in W gives rise to three (unique) pairs of H .

Consider the simple bipartite graph $G_{I,H} = (H \cup I, E_I)$ where edges of E_I are treated as simple edges connecting an element x of I to an element $\{y, z\}$ of H whenever $\{x, y, z\}$ is in E (or just E_I).

Lemma 4. *If $|I| \geq |H|$, then the graph $G_{I,H}$ has a crown (H', I', M) such that: $H' \subset H$ and $I' \subset I$.*

Proof. If $|I| \geq |H|$, then I satisfies the condition of Lemma 1 and the promised crown (H', I', M) can be constructed using *Construct_Crown* (the procedure described in [6]). \square

By the crown reduction rule, if the original instance (S, C, k) of 3HS has a solution, then some hitting set contains at least one element of every pair of H' and excludes all elements of I' . Therefore, we can safely delete all elements of I' , but we do not delete their incident edges, which are elements of H' .

To summarize, our 3HS kernelization algorithm is the following:

Algorithm 3HS-Kernel

Input: Instance (S, C, k) of 3HS.

Output: Either NO if (S, C) has no hitting set of size $\leq k$, or an instance (S', C', k') and a partial solution A such that $k' = k - |A|$ and $|S'| \leq 5k^2 + k$.

Begin

```

While  $C$  contains a singleton set  $\{x\}$ 
    add  $x$  to  $A$ , delete  $\{x\}$  from  $C$  and decrement  $k$ .
Apply the high-degree rule
Construct  $W$ 
If  $|W| > k^2$  return NO
Construct the simple bipartite graph  $G_{I,H}$ 
    
```

```

If  $|I| \geq |H|$ 
   $(H', I', M) \leftarrow \text{Construct\_Crown}(G_{I,H})$ 
  Remove all elements of  $I'$ 
Return the (possibly) new instance

```

End

Finally, we can state our kernelization theorem.

Theorem 1. *There is a polynomial-time algorithm that, for an arbitrary input instance (S, C, k) of 3HS, either determines that (S, C, k) is a no instance or computes a kernel instance whose order is bounded above by $5k^2 + k$.*

Proof. Based on the above discussion, the number of elements that remain in I cannot exceed the number of pairs that belong to $H \setminus H'$. The theorem follows from the facts that H has at most $3k^2$ pairs and the vertex set of W has at most $2k^2 + k$ elements. \square

Note that our 3HS kernelization focused only on reducing the order of the input hypergraph. This is a common approach in graph-theoretic problems. The Vertex Cover problem is known to have a linear problem kernel because of a bound of $2k$ on the number of vertices, yet the number of edges is bounded by a quadratic function of k . This is believed to be the best possible kernelization. In fact, during the writing of this paper, we learned that Holger Dell proved the following [7]:

Proposition 1. *Let $d > 1$ and $\epsilon > 0$. Unless the PH collapses to the third level, dHS does not have a polynomial-time reduction mapping instances on n vertices to instances of size $n^{d-\epsilon}$.*

It follows that dHS does not have a kernel of size $O(k^{d-\epsilon})$ unless the PH collapses. In particular, it is hard to have a 3HS kernel with $O(k^2)$ edges. In our kernelization, however, if every pair of vertices appears together (i.e., co-occur) in at most c edges, for some constant c , then the number of edges is bounded by $3ck^2$. This is seen easily from the fact that $|H| \leq 3k^2$. If the number of edges is not reduced below some large multiple of k^2 , then we obtain an instance with high-degree vertices and/or high co-occurrences of pairs of vertices. This is a favorable condition for most (subsequent) parameterized search algorithms, as illustrated in [9] where the algorithm starts by branching at vertices of high-degree. The hard instances for such search-tree based algorithms are those with small vertex degrees (almost three), which are guaranteed to have linear size after applying our kernelization.

Remark 1. One of the applications of the above algorithm is a quadratic kernel for the Triangle Vertex Deletion Problem (TVD), which takes a simple undirected graph G and a positive integer k as input and asks whether G has a set of k or less vertices whose complement induces a triangle-free subgraph. A TVD instance is transformed into a 3HS one easily by treating S and C as the sets of vertices and triangles. Moreover, we must cease the use of the edge domination rule, as we did already in the above pseudocode, to make sure that no edge (or triangle) is deleted without deleting at least one of its vertices. Such kernel is termed “vertex-induced”. Vertex Inducedness of kernels was studied in [3] where an algorithm producing a cubic-order vertex-induced kernel for 3HS was obtained.

4. d-HS kernelization

The dHS problem has a kernel of size $O(k^d)$ that can be obtained using the Sunflower Lemma [10]. An improved kernelization algorithm can be obtained by a careful generalization of our 3HS approach. We discuss this generalization in this section, keeping our main focus on how to obtain an upper bound on the order of the resulting kernel. We do not try to study the most efficient techniques for designing a kernelization algorithm. However, the run time of our algorithm will be shown to be (asymptotically bounded by) a polynomial function of the input size. The reader should keep in mind that d is a small constant, since the parameterized complexity of the problem gets higher as the value of d increases.

Again, we consider an instance (S, C, k) of dHS. The concept of a *subedge* plays a central role in this section. As the name suggests, a subedge of (S, C) is a non-empty subset of S that is contained in an element (edge) of C . A *j-subedge* is a subedge of cardinality j . For an edge $e \in C$, we shall use the expression “ e' occurs in e ” when e' is a subedge of e .

4.1. The high-degree rule for dHS

If $x \in S$ is the unique pair-wise intersection of more than k edges of C , then we can include x in any solution and delete it. However, this intuitive generalization of the high-degree rule is hard to implement in polynomial time. In fact, checking whether $x \in S$ satisfies this property requires solving instances of the Maximum Matching problem in hypergraphs, which is NP-hard. However, we can still have a high-degree rule for dHS that generalizes its 3HS homonym, as well as the first reduction rule used in [17] (discussed in Section 2). The rule is stated as follows:

If a subedge $e \subset S$ satisfies:

- (i) $|e| = d - 2$, and
- (ii) e is the pair-wise intersection of more than k edges.

Then add e to C and delete all elements of C that contain e (as a subset).

The soundness of this rule is obvious. If a subedge e satisfies (i) and (ii) and does not contain any element of some solution, A , then each of the edges that intersect at e has a distinct element of A . This is not possible, unless (S, C, k) is a no instance. Note the implicit application of the edge domination rule in the edge deletion part (after adding e). Therefore, the resulting instance is not vertex-induced. This did not happen in the 3HS case since e was a singleton, which allowed us to delete e by placing its unique element in a potential solution.

Deciding whether a $(d - 2)$ -subedge e is contained in more than k edges whose pair-wise intersection is e is to find a matching of size $> k$ in the simple graph consisting of the set $C'(e) = \{e' : e \cup e' \in C\}$. This is an instance of the Maximum Matching problem in simple graphs, which is solvable in polynomial time ($O(|C'(e)|^{2.5})$). As in the 3HS case, we may choose to delay the application of this rule and make it part of the subsequent construction of a maximal set of weakly related edges. In either case, the run time will be a low-order polynomial of the input size.

If the high-degree rule does not apply to a dHS instance (S, C, k) , then we shall say that (S, C, k) is **reduced**. This could be the result of applying the rule iteratively until it cannot be applied.

4.2. Weakly related edges

Two edges are weakly related if their intersection contains at most $d - 2$ elements and neither of them is a subset of the other. Note the difference between this notion and its 3HS counterpart. This particular generalization is needed because of the difficulty of applying the intuitive generalization of the high-degree rule.

Again, our kernelization algorithm proceeds by constructing a maximal set, W , of weakly related edges. The construction starts by placing all edges of size $d - 2$ or less in W . At each subsequent stage, an element e of C is selected and checked against the current elements of W . Then e is added to W if the elements of $W \cup \{e\}$ are weakly related. Each edge addition takes $O(d|W|)$ steps. The total construction time is in $O(d|W||C|)$, which is at most quadratic in the input size.

Let A be a solution of a reduced instance (S, C, k) of dHS, and let W be as described above. Denote by W_e the set of edges of W that contain a subedge e properly. If e is a $(d - 2)$ -subedge then, by the high-degree rule, $|W_e| \leq k$. To achieve our sought upper bound, we apply the following reduction algorithm.

The High Occurrence Rule:

```

For  $i = d - 2$  downto 1 do
  For each  $i$ -subedge  $e$  of  $W$  do
    if  $|W_e| > k^{d-1-i}$ , then
      Add  $e$  to  $W$ 
      Delete (from  $C$  and  $W$ ) all edges containing  $e$ 
    
```

Knowing that every $(d - 2)$ -subedge occurs in at most k elements of W , we prove the soundness of each iteration of the above rule as follows.

Assume that the i th iteration is correct and every i -subedge occurs in at most k^{d-1-i} edges of W . Let e be an $(i - 1)$ -subedge occurring in more than k^{d-i} such edges. If $e \cap A = \emptyset$, then each element of W_e must have (in addition to elements of e) at least one element from A . Since $|A| \leq k$, some element x of A must appear in more than k^{d-1-i} edges of W_e . It follows that $e \cup \{x\}$ is an i -subedge occurring in more than k^{d-1-i} edges of W . This is impossible due to the action performed in the previous iteration. Therefore, it is safe to add e to W (and C) and delete all elements of W_e .

Lemma 5. Let (S, C, k) be a reduced yes instance of dHS and let W be a maximal set of weakly related edges that result from applying the high-occurrence rule. Then $|W| \leq k^{d-1}$.

Proof. Follows immediately from the high-occurrence rule since any 1-subedge is contained in at most k^{d-2} edges and the elements of A form at most k singleton subedges that occur in every element of W . \square

As for the run time of the above reduction rule, we note that $|W|$ has $O(2^d|W|)$ subedges in the worst-case. So the total run time is in $O(d2^d|W|^2)$. A better run time can be achieved if, at each iteration, a sorted list L_i of all i -subedges of W is formed. L_i is sorted according to a lexicographical ordering of the i -subedges, and it contains the (initially zero) number of occurrences for each subedge. Constructing L_i takes $O(d^i|W|\log|W|)$ time. Then, a single pass through edges of W would

suffice to determine the number of occurrences of all i -subedges and making a list of those that occur in more than k^{d-1-i} edges. The total run time is in $O(2^d |W| \log |W|)$.¹

Every edge of W has at most d elements. Thus $|V(W)| \leq dk^{d-1}$. However, one can prove a tighter bound just as in the 3HS case. In fact, at most k vertices hit all edges of W , and each belongs to at most k^{d-2} such edges. Excluding the k elements of a solution, each edge has at most $d-1$ (non-hitting) vertices. It follows that there are $k + (d-1)k^{d-1}$ vertices in $V(W)$.

4.3. Crown reduction

Let I be the complement of $V(W)$ in S , and let H be the set of all $(d-1)$ -subedges of W . Then we observe the following:

- Elements of C whose cardinality is less than d are placed in W .
- I is an independent set.
- Every edge e of size d that contains an element x of I consists of x and a $(d-1)$ -subedge H . Otherwise, e could be added to W , violating W 's maximality.
- $|V(H)| \leq dk^{d-1}$: each edge of size d of W gives rise to d $(d-1)$ -subedges.

As in the 3HS case, we proceed by constructing the bipartite graph $G_{I,H}$. The rest of the work is identical to the crown construction and reduction that was used in the 3HS case. All these steps, and the ones that preceded them, are performed in polynomial-time. The number of vertices in the resulting instance is bounded above by $|V(H)| + |V(W)| \leq dk^{d-1} + (d-1)k^{d-1} + k$. To conclude:

Theorem 2. *There is a polynomial-time algorithm that, for an arbitrary input instance (S, C, k) of dHS, either determines that (S, C, k) is a no instance or computes a kernel instance whose order is bounded above by $(2d-1)k^{d-1} + k$.*

5. Concluding remarks

We presented an algorithm that produces quadratic-order kernels for 3HS. Previously known kernelization strategies could guarantee cubic-order kernels only. An attempt for improving the kernel size appeared also in [18] by local usage of the NT kernelization of Vertex Cover [16]. However, it was observed that such technique does not always produce correct kernels [20].

Our kernelization method extends the notion of crown reduction to hypergraphs. An important feature of this method is its isolation of a subgraph that is almost simple in the input hypergraph: the subgraph formed by the set W of weakly related edges. We believe that this strategy could be employed in other hypergraph problems.

Many known graph-theoretic problems reduce to Hitting Set. In addition to the quadratic kernel for *Triangle Vertex Deletion*, our 3HS algorithm has been used to deduce quadratic-order kernels for *Cluster Vertex Deletion* and *Feedback Vertex Set* in tournaments, which were studied in [12]. Moreover, the approach introduced in this paper has been used in kernelization algorithms for some maximization problems such as *Triangle Packing*, *3-Set Packing* and *3D Matching* [4,15].

We generalized our kernelization approach to work for any dHS problem. For $d > 3$, our dHS kernel is not necessarily vertex-induced, in the sense that the resulting hyper-subgraph could have vertices that share edges that are deleted from the original instance. Any vertex-induced version has to cease the use of the edge domination rule, which is also part of the general high-degree rule (when $d > 3$). It would thus be interesting to find a general vertex-induced kernel.

It is now believed that, for $r < d$, no dHS kernelization can guarantee a bound of $O(k^r)$ on the number of edges. It remains to know whether a bound that is asymptotically smaller than $O(k^{d-1})$ exists on the number of vertices. We note that, as the ratio of edges to vertices ($|C|/|S|$) gets higher, the degrees of vertices and the number of co-occurrences of pairs of vertices gets higher as well. Eventually, this increases the chances of applying reduction rules. With this in mind, we believe that an asymptotic bound that is significantly better than $O(k^{d-1})$ is also hard to obtain.

Acknowledgment

We would like to thank Daniel Raible and Drew Mellor for their valuable comments.

References

- [1] F.N. Abu-Khzam, R.L. Collins, M.R. Fellows, M.A. Langston, W.H. Suters, C.T. Symons, Kernelization algorithms for the vertex cover problem: Theory and experiments, in: Workshop on Algorithm Engineering and Experiments (ALENEX), 2004, pp. 62–69.
- [2] F.N. Abu-Khzam, M.R. Fellows, M.A. Langston, W.H. Suters, Crown structures for vertex cover kernelization, Theory Comput. Syst. 41 (3) (2007) 411–430.

¹ Recall that d is a small constant.

- [3] F.N. Abu-Khzam, H. Fernau, Kernels: Annotated, proper and induced, in: International Workshop on Parameterized and Exact Computation (IWPEC), 2006, pp. 264–275.
- [4] F.N. Abu-Khzam, A quadratic kernel for 3-set packing, in: Proceedings of the 6th Annual Conference on Theory and Applications of Models of Computation, TAMC 2009, in: Lecture Notes in Comput. Sci., vol. 5532, 2009, pp. 81–87.
- [5] J.F. Buss, J. Goldsmith, Nondeterminism within \mathcal{P} , *SIAM J. Comput.* 22 (1993) 560–572.
- [6] B. Chor, M.R. Fellows, D. Juedes, Linear kernels in linear time, or how to save k colors in $O(n^2)$ steps, in: International Workshop on Graph Theoretic Concepts in Computer Science (WG), 2004, pp. 257–269.
- [7] H. Dell, Private communication, 2009.
- [8] R.G. Downey, M.R. Fellows, Parameterized Complexity, Springer, 1999.
- [9] H. Fernau, A top-down approach to search-trees: Improved algorithms for 3-Hitting Set, in: Electronic Colloquium on Computational Complexity (ECCC), 2004, p. 073.
- [10] J. Flum, M. Grohe, Parameterized Complexity Theory, Springer, 2006.
- [11] M.R. Garey, D.S. Johnson, Computers and Intractability, W.H. Freeman, New York, 1979.
- [12] Falk Hüffner, Algorithms and experiments for parameterized approaches to hard graph problems, PhD thesis, Institut für Informatik, Friedrich-Schiller-Universität Jena, 2007.
- [13] James A. Jones, Mary Jean Harrold, Test-suite reduction and prioritization for modified condition/decision coverage, *IEEE Trans. Software Eng.* 29 (3) (2003) 195–209.
- [14] Fabian Kuhn, Pascal von Rickenbach, Roger Wattenhofer, Emo Welzl, Aaron Zollinger, Interference in cellular networks: The minimum membership set cover problem, in: 11th International Computing and Combinatorics Conference (COCOON), Kunming, Yunnan, China, August 2005.
- [15] Hannes Mozer, A problem kernelization for graph packing, in: 35th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM), in: Lecture Notes in Comput. Sci., vol. 5404, 2009, pp. 401–412.
- [16] G.L. Nemhauser, L.E. Trotter, Vertex packings: Structural properties and algorithms, *Math. Program.* 8 (1975) 232–248.
- [17] R. Niedermeier, P. Rossmanith, An efficient fixed-parameter algorithm for 3-Hitting Set, *J. Discrete Algorithms* 1 (2003) 89–102.
- [18] N. Nishimura, P. Ragde, D.M. Thilikos, Smaller kernels for hitting set problems of constant arity, in: International Workshop on Parameterized and Exact Computation (IWPEC), in: Lecture Notes in Comput. Sci., vol. 3162, 2004, pp. 121–126.
- [19] D. Ruchkys, S. Song, A parallel approximation hitting set algorithm for gene expression analysis, in: SBAC-PAD '02: Proceedings of the 14th Symposium on Computer Architecture and High Performance Computing, IEEE Computer Society, Washington, DC, USA, 2002, p. 75.
- [20] D.M. Thilikos, Private communication, 2005.
- [21] K. Weihe, Covering trains by stations or the power of data reduction, in: R. Battiti, A.A. Bertossi (Eds.), International Conference on Algorithms and Experiments, 1998, pp. 1–8.