Smaller kernels for hitting set problems of constant arity
Nishimura, N. and Ragde, P. and Thilikos, D.M.
Research Report LSI-04-29-R

Departament de Llenguatges i Sistemes Informàtics

UNIVERSITAT POLITÈCNICA DE CATALUNYA

Smaller kernels for hitting set problems of constant arity*

Naomi Nishimura[1], Prabhakar Ragde[1], and Dimitrios M. Thilikos[2]

[1] School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, {nishi,plragde}@uwaterloo.ca.

[2] Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, Campus Nord – Mòdul C5, c/Jordi Girona Salgado 1-3, 08034 Barcelona, Spain, sedthilk@lsi.upc.es.

Abstract. We demonstrate a kernel of size $O(k^2)$ for 3-Hitting Set (Hitting Set when all subsets in the collection to be hit are of size at most three), giving a partial answer to an open question of Niedermeier by improving on the $O(k^3)$ kernel of Niedermeier and Rossmanith. Our technique uses the Nemhauser-Trotter linear-size kernel for Vertex Cover, and generalizes to demonstrating a kernel of size $O(k^{r-1})$ for r-Hitting Set (for fixed r).

1 Introduction

Kernelization is a central technique in the development of fixed-parameter tractable algorithms. Intuitively, a kernelization for a parameterized problem $\mathcal{P}$ is a polynomial-time algorithm that, given any instance $\mathcal{I}$, either determines that it is a "no" instance, or finds another instance $\mathcal{I}'$ (with perhaps a modified parameter value) such that $\mathcal{I}'$ is a "yes" instance if and only if $\mathcal{I}$ is, and the size of $\mathcal{I}'$ (which is the kernel) is bounded by some function of k. Since many problems have exponential-time brute-force algorithms, applying one to $\mathcal{I}'$ yields a fixed-parameter tractable algorithm for $\mathcal{P}$. Although technically any problem

* The two first authors were supported by the Natural Sciences and Engineering Research Council of Canada (NSERC). The third author was partially supported by the EU within the 6th Framework Programme under contract 001907 (DELIS) and by the Spanish CICYT project TIC-2002-04498-C05-03 (TRACER).

admitting a fixed-parameter tractable algorithm has a kernel, it is more useful to discover kernelization techniques as an aid to finding fixed-parameter tractable algorithms than to extract kernels from such algorithms.

This paper describes an algorithm for finding kernels for variations of the Hitting Set problem, one of the original NP-complete problems from Karp's 1972 paper [5]. In order to describe these variations, we need some terminology. Given two sets S and T, we say S *hits* T (or T *is hit by* S) if $S \cap T = \emptyset$. We apologize for the violent metaphor, which we inherited; our preference would be to use *touches* or *meets*.

r-Hitting Set

Input: A collection $\mathcal{C}$ of subsets of a finite set S such that for every $T \in \mathcal{C}$, $|T| \leq r$.

Parameter: A non-negative integer k.

Question: Is there a subset H of S such that $|H| \leq k$ and for every $T \in \mathcal{C}$, H hits T?

We describe an instance of r-Hitting Set by a tuple $\mathcal{I} = (S, \mathcal{C}, k)$. For any $r \geq 2$, r-Hitting Set is NP-complete, since for $r = 2$ it is simply the Vertex Cover problem (S is the set of vertices and $\mathcal{C}$ the set of edges).

Because of the set-theoretic nature of r-Hitting Set, kernels for it have a slightly stronger property than described in the general introduction above. Given a collection $\mathcal{C} \subseteq 2^S$ and a subset S' of S, we denote by $\mathcal{C}|_{S'}$ the collection of sets in $\mathcal{C}$ restricted to S', that is, the set $\{T \cap S' \mid T \in \mathcal{C}, T \cap S' \neq \emptyset\}$. A kernel for an instance of r-Hitting Set is a subset K of S such that any solution to the instance $\mathcal{I}' = (K, \mathcal{C}|_K, k)$ of r-Hitting Set is a solution to $\mathcal{I}$.

If we can find a kernel of size $f(k)$ for r-Hitting Set, we can solve any instance in time $O(f(k)^k n)$ by simply trying all subsets of the kernel to see if they are solutions to the restricted problem. As we mentioned above, this approach works for other kernelizable problems, though in the case of r-Hitting Set an approach based on bounded search trees is faster [8], and kernelization serves mainly to bring the running time of the search algorithm down from $O(c^k n)$ to $O(c^k f(k) + n)$ (for some constant c). The ultimate goal is to find a linear-sized

kernel for problems, because this brings down the cost of the brute-force search, and implies the existence of a constant-factor approximation algorithm.

2 Reducing kernel size

The first instance of kernelization most students of parameterized complexity [4] see is the $O(k^2)$ kernel for Vertex Cover due to S. Buss [2, cited in [1]]. But Vertex Cover is one of the few problems known to have a linear-sized kernel, provided by the following theorem due to Nemhauser and Trotter [6] (with improved running time due to Chen et al. [3]).

Theorem 1. *There is an algorithm running in time $O(kn + k^3)$ that, for any instance $\mathcal{I} = (S, \mathcal{C}, k)$ of* Vertex Cover *with input size n, either computes a kernel $K \subseteq S$, $|K| \leq 2k$, or proves that $\mathcal{I}$ is a "no" instance.*

The proof of Theorem 1 makes elegant use of the linear programming relaxation of the integer program for Vertex Cover, but we do not need to know anything about the proof; we will use the algorithm as a black-box subroutine in computing our kernel for r-Hitting Set. To illustrate the method, we first demonstrate a kernel of size $6k^2$ for 3-Hitting Set. Theorem 2 below improves the $O(k^3)$ kernel for 3-Hitting Set given by Niedermayer and Rossmanith [8], which is basically a generalization of Buss's $O(k^2)$ kernel for Vertex Cover. We save a factor of k by using the fact that Nemhauser and Trotter give a kernel of size $O(k)$, not $O(k^2)$, for r-Hitting Set in the case $r = 2$, though an entirely different technique is employed to make use of this fact. Niedermeier [7, p. 31] lists as an open problem the use of Nemhauser-Trotter techniques to improve the size of the kernel for r-Hitting Set.

Theorem 2. *There is an algorithm running in time $O(kn + k^4)$ that, for any instance $\mathcal{I} = (S, \mathcal{C}, k)$ of* 3-Hitting Set *with input size n, either computes a kernel $K \subseteq S$, $|K| \leq 6k^2$), or proves that $\mathcal{I}$ is a "no" instance.*

Proof: Our algorithm actually does something slightly stronger; it computes a set $F \subseteq S$ of elements that must be in any solution to $\mathcal{I}$, and a set $M \subseteq S$ of

elements that may be in the solution, with $|F \cup M| \leq 6k^2$. (Many kernelization algorithms do this, including that of Theorem 1, though we do not use this fact.)

We start by forming a collection $\mathcal{G} \subseteq \mathcal{C}$ using the following greedy algorithm: Start with $\mathcal{G}$ empty, and repeatedly choose an arbitrary set $C \in \mathcal{C}$. Add C to $\mathcal{G}$, and delete from $\mathcal{C}$ any set with a nonempty intersection with C. Repeat until $\mathcal{C}$ is empty. This takes $O(n)$ time.

If $|\mathcal{G}|$ contains more than k sets, $\mathcal{I}$ is a "no" instance, because any two sets in $\mathcal{G}$ are disjoint, and so more than k elements are required to hit them all. If $|\mathcal{G}| \leq k$, we proceed to construct the kernel. Let E be the set of elements appearing in the sets in $\mathcal{G}$, that is, $E = \bigcup_{G \in \mathcal{G}} G$. E must hit every set in $\mathcal{C}$ (if it did not hit some set, the set would not have been deleted in the algorithm that created $\mathcal{G}$, which is a contradiction), and $|E| \leq 3k$. Thus E is a hitting set, but if a hitting set of size k exists, it may not be contained within E. We will use the elements of E to construct our kernel.

For each $e \in E$, we define $\mathcal{C}_e$ to be the collection of sets in $\mathcal{C}$ containing the element e, that is, $\mathcal{C}_e = \{T \mid e \in T, T \in \mathcal{C}\}$. Think of $\mathcal{C}_e$ as a subproblem induced by the element e. We also define $\mathcal{C}'_e$ to be the sets in $\mathcal{C}_e$ but with the element e removed, that is, $\mathcal{C}'_e = \{T \setminus \{e\} \mid T \in \mathcal{C}_e\}$. Note that $\mathcal{I}_e = (S \setminus \{e\}, \mathcal{C}'_e, k)$ is an instance of Vertex Cover, since every set in $\mathcal{C}'_e$ has size at most two. Since every $T \in \mathcal{C}_e$ must be hit, we will either choose e to be in the hitting set, or we find a hitting set for $\mathcal{I}_e$. The former can be achieved by adding e to F, and the latter by adding to M a kernel for the instance $\mathcal{I}_e$ of Vertex Cover. Doing this for every element $e \in E$ will give us the sets F and M that we seek.

Applying Theorem 1, either $\mathcal{I}_e$ is a "no" instance, or we can find a kernel K_e of size at most $2k$ for $\mathcal{I}_e$ in time $O(kn_e + k^3)$ (where n_e is the size of the instance $\mathcal{I}_e$). If $\mathcal{I}_e$ is a "no" instance, e must be in any solution H for $\mathcal{I}$. Suppose it is not. Then in order for H to hit all the sets in $\mathcal{C}_e$, H would have to be a solution to $\mathcal{I}_e$. But since $\mathcal{I}_e$ is a "no" instance, this would make the size of H greater than k, which is a contradiction to H being a solution for $\mathcal{I}$. Thus if $\mathcal{I}_e$ is a "no" instance, we add e to F. If instead it is a "yes" instance, we add K_e to M.

Since, for each of the at most $3k$ elements of E, we either added one element to F or at most $2k$ elements to M, $|F \cup M| \leq 6k^2$, as required.

We claim that the set $K = F \cup M$ is a kernel for $\mathcal{I}$. To see this, suppose that $\mathcal{I}'$ is the problem defined by the candidate kernel K, that is, $\mathcal{I}' = (K, \mathcal{C}|_K, k)$. If $\mathcal{I}'$ is a "no" instance, then $\mathcal{I}$ must be as well, since if H is a solution for $\mathcal{I}$, $H \cap K$ will be a solution for $\mathcal{I}'$. If $\mathcal{I}'$ is a "yes" instance, then let $H' \subseteq K$ be a solution. Since every element added to F must be in any solution, we know that $F \subseteq H'$. We need to show that H' is a solution for $\mathcal{I}$. To do this, we take an arbitrary set T in $\mathcal{C}$, and show that H' hits it.

If F hits T, we are done, so suppose it does not. Since E hits T, there must be some element e in $E \cap T$; if e is in the hitting set H', then T is hit by H', since it is hit by e. If e is not in H', then since F does not hit T, e is not in F. It follows that $\mathcal{I}_e$ is a "yes" instance, and $K_e \subseteq M$.

Since K_e is a kernel for $\mathcal{I}_e$, K_e must hit $T \setminus \{e\}$ (because K_e contains a solution for $\mathcal{I}_e$, which is a "yes" instance). Since $T \cap K_e$ is nonempty, it is a set in $\mathcal{C}|_K$. Since H' is a hitting set for $\mathcal{C}|_K$, it must hit $T \cap K_e$, and therefore it hits T, as required.

The running time of the procedure is $O(n)$ to find E plus $\sum_{e \in E} O(kn_e + k^3)$ to find F and M, where n_e is the number of sets in $\mathcal{C}$ that contain e. Since every set in $\mathcal{C}$ contains at most three elements, $\sum_e n_e \leq 3n$, and since $|E| \leq 3k$, the total running time is $O(kn + k^4)$. □

In proving the above theorem, we used a subroutine for kernelizing Vertex Cover to create an algorithm for kernelizing 3-Hitting Set. We can continue this process, using a subroutine for kernelizing $(r-1)$-Hitting Set to create an algorithm for kernelizing r-Hitting Set.

Theorem 3. *For fixed $r \geq 2$, if there is an algorithm running in time $O(kn+k^r)$ that finds an $O(k^{r-2})$-size kernel for "yes" instances of $(r-1)$-Hitting Set, then there is an algorithm running in time $O(kn + k^{r+1})$ that finds an $O(k^{r-1})$-size kernel for "yes" instances of r-Hitting Set.*

Proof: The proof of this theorem is a generalization of the proof of Theorem 2, which we can describe more succinctly now. Let $\mathcal{I} = (S, \mathcal{C}, k)$ be an instance of r-Hitting Set. As before, we form a set F of elements that must be in any solution to $\mathcal{I}$, and a set M of elements that might be, with $F \cup M$ being our kernel of the desired size. Let $\mathcal{G}$ be a maximal pairwise disjoint collection of sets

from $\mathcal{C}$ chosen using a greedy algorithm running in time $O(n)$. If $|\mathcal{G}| > k$, there is no solution to $\mathcal{I}$. If $|\mathcal{G}| \leq k$, we let $E = \bigcup_{C \in \mathcal{G}} C$. E must hit every set in $\mathcal{C}$, and $|E| \leq rk$.

For each $e \in E$, we define $\mathcal{C}_e = \{T \mid e \in T, T \in \mathcal{C}\}$, and $\mathcal{C}'_e = \{T \setminus \{e\} \mid T \in \mathcal{C}_e\}$. Then $\mathcal{I}_e = (S \setminus \{e\}, \mathcal{C}'_e, k)$ is an instance of $(r-1)$-Hitting Set. By the statement of the theorem, either $\mathcal{I}_e$ is a "no" instance, or we can find a kernel K_e of size $O(k^{r-2})$ for $\mathcal{I}_e$, where n_e is the size of $\mathcal{I}_e$. In the former case, e must be in any solution H for $\mathcal{I}$, because if it is not, H would have to be a solution to $\mathcal{I}_e$ in order for it to hit all the sets in $\mathcal{C}_e$; thus we can add e to F. In the latter case, we add K_e to M. Since we are adding at most rk sets of size $O(k^{r-2})$ to either F or M, $|F \cup M| = O(k^{r-1})$.

Then $K = F \cup M$ is a kernel for $\mathcal{I}$. To see this, we define $\mathcal{I}' = (K, \mathcal{C}|_K, k)$. If $\mathcal{I}'$ is a "no" instance, so is $\mathcal{I}$. If $\mathcal{I}'$ is a "yes" instance, let H' be a solution for it; $F \subseteq H'$. If T is any set in $\mathcal{C}$, we must show that H' hits T. Either F hits T, or, since E hits T, there exists $e \in E \cap T$; if $e \in H'$, then T is hit by H'. If $e \notin H'$, $\mathcal{I}_e$ is a "yes" instance. Since K_e is a kernel for $\mathcal{I}_e$, it hits $T \setminus \{e\}$, so $T \cap K_e$ is a set in $\mathcal{C}|_K$. Since H' is a hitting set for $\mathcal{C}|_K$, it must hit $T \cap K_e$ and thus T.

The running time of the procedure is $O(n) + \sum_e O(kn_e + k^r)$; since every set contains at most r elements, $\sum_e n_e \leq rn$, and the total running time is $O(kn + k^{r+1})$. □

The method of Niedermeier and Rossmanith also generalizes easily to provide a kernel of size $O(k^r)$ for r-Hitting Set (though this is not explicitly mentioned in their paper). Unfortunately, the constants hidden in the O-notation increase with r in our case, but not for the Niedermeier-Rossmanith kernel. Thus Theorem 3 is not of much practical interest for larger values of r; even in the case $r = 3$, the Niedermeier-Rossmanith kernel is smaller for $k < 6$. However, Niedermeier [7, p. 34] mentions that Nemhauser-Trotter kernelization seems to perform well in practice, and this suggests that our kernelization for 3-Hitting Set may also have practical merit.

3 Conclusion

Since HITTING SET is W[2]-complete when the size of sets in the collection is not bounded in size [4], it is unlikely that we will find a linear-sized kernel for this problem. However, the statement "r-HITTING SET has a kernel of size $f(r)k$" for some $f(r) = 2^{\Omega(r)}$ is not inconsistent with what we believe about separations among parameterized complexity classes. It would be interesting to either prove this statement, or to demonstrate it false by proving lower bounds on the size of kernels for this and other FPT problems.

4 Acknowledgements

Part of this work was done while the authors were attending the 2004 Workshop on Fixed-Parameter Tractability Methods in Geometry and Games at the Bellairs Research Institute of McGill University in Holetown, Barbados. We wish to thank Sue Whitesides for inviting us. The second author also wishes to thank Arju and Zuki for swimming safely so that he could pace on the beach and think.

References

1. Jonathan F. Buss and Judy Goldsmith. Nondeterminism within P. *SIAM Journal on Computing*, 22(3):560–572, 1993.
2. Sam Buss. private communication, 1989.
3. J. Chen, I.A. Kanj, and W. Jia. Vertex cover: further observations and further improvements. *Journal of Algorithms*, 41:280–301, 2001.
4. R.G. Downey and M.R. Fellows. *Parameterized Complexity*. Springer, 1999.
5. Richard M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.
6. G. L. Nemhauser and L. E. Trotter Jr. Vertex packings: Structural properties and algorithms. *Mathematical Programming*, 8:232–248, 1975.
7. Rolf Niedermeier. *Invitation to fixed-parameter algorithms*. PhD thesis, Universität Tübingen, 2002. Habilitation thesis.
8. Rolf Niedermeier and Peter Rossmanith. An efficient fixed parameter algorithm for 3-hitting set. *Journal of Discrete Algorithms*, 2(1):93–107, 2002.

Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya

Research Reports - 2004

- LSI-04-1-R : *Automatic Generation of Polynomial Loop Invariants: Algebraic Foundations*, Rodríguez, E. and Kapur, D.
- LSI-04-2-R : *Comparison of Methods to Predict Ozone Concentration* , Orozco, J.
- LSI-04-3-R : *Towards the definition of a taxonomy for the cots product´s market* , Ayala, Claudia P.
- LSI-04-4-R : *Modelling Coalition Formation over Time for Iterative Coalition Games*, Mérida-Campos, C. and Willmott, S.
- LSI-04-5-R : *Illegal Agents? Creating Wholly Independent Autonomous Entities in Online Worlds*, Willmott, S.
- LSI-04-6-R : *An Analysis Pattern for Electronic Marketplaces*, Queralt, A. and Teniente, E.
- LSI-04-7-R : *Exploring Dopamine-Mediated Reward Processing through the Analysis of EEG-Measured Gamma-Band Brain Oscillations*, Vellido, A. and El-Deredy, W.
- LSI-04-8-R : *Studying Embedded Human EEG Dynamics Using Generative Topographic Mapping*, Vellido, A. and El-Deredy, W. and Lisboa, P.J.G.
- LSI-04-9-R : *Similarity and Dissimilarity Concepts in Machine Learning*, Orozco, J.
- LSI-04-10-R : *A Framework for the Definition of Metrics for Actor-Dependency Models*, Quer, C. and Grau, G. and Franch, X.
- LSI-04-11-R : *QM: A Tool for Building Software Quality Models*, Carvallo, J.P. and Franch, X. and Grau, G. and Quer, C.
- LSI-04-12-R : *COSTUME: A Method for Building Quality Models for Composite COTS-based Software Systems*, Carvallo, J.P. and Franch, X. and Grau, G. and Quer, C.
- LSI-04-13-R : *Enabling Collaboration in Virtual Reality Navigators*, Theoktisto, V. and Fairén, M. and Navazo, I.
- LSI-04-14-R : *DesCOTS: A Software System for Selecting COTS Components*, Carvallo, J.P. and Franch, X. and Grau, G. and Quer, C.
- LSI-04-15-R : *Evaluation and symmetrisation of alignments obtained with the Giza++ software*, Lambert, P. and Castell, N.
- LSI-04-16-R : *A note on the use of topology extensions for provoking instability in communication networks*, Blesa, M.J.
- LSI-04-17-R : *An ISO/IEC-compliant Quality Model for ER Diagrams*, Costal, D. and Franch, X.
- LSI-04-18-R : *A Case Study on Pruning General Ontologies for the Development of Conceptual Schemas* , Conesa, J.
- LSI-04-19-R : *Adding Efficient and Reliable Access Paths to the JCF*, Marco, J. and Franch, X.

- LSI-04-20-R : *Exploiting Simple Corporate Memory in Iterative Coalition Games*, Mérida-Campos, C. and Willmott, S.
- LSI-04-21-R : *On the Semantics of Operation Contracts in Conceptual Modeling* , Queralt, A. and Teniente, E.
- LSI-04-22-R : *Complexity issues on bounded restrictive H-coloring*, Díaz, J. and Serna, M. and Thilikos, D.M.
- LSI-04-23-R : *Chromatic number in random scaled sector graphs*, Díaz, J. and Sanwalani, V. and Serna, M. and Spirakis, P.
- LSI-04-24-R : *Bounds on the bisection width for random d-regular graphs*, Díaz, J. and Serna, M. and Wormald, N.C.
- LSI-04-25-R : *Open Source environment to define constraints in route planning for GIS-T*, Pérez, L. and Silveira, A. da M.
- LSI-04-26-R : *A basic repository of operations for the refinement of general ontologies*, de Palol, X.
- LSI-04-27-R : *Tetrahedral mesh subdivision based on underlying volume data*, Rodríguez, L. and Navazo, I. and Vinacua, A.
- LSI-04-28-R : *The Price of Connectedness in Expansions*, Fomin, F.V. and Fraigniaud, P. and Thilikos, D.M.
- LSI-04-29-R : *Smaller kernels for hitting set problems of constant arity*, Nishimura, N. and Ragde, P. and Thilikos, D.M.

Hardcopies of reports can be ordered from:

Núria Sanchez
Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
Campus Nord, Mòdul C6
Jordi Girona Salgado, 1-3
03034 Barcelona,Spain
nurias@lsi.upc.es

See also the Departament WWW pages, http://www.lsi.upc.es/