Simpler Parameterized Algorithm for OCT

Daniel Lokshtanov* Saket Saurabh* Somnath Sikdar†

Abstract

We give a simple and intuitive fixed parameter tractable algorithm for the Odd Cycle Transversal problem, running in time $O(3^k \cdot k \cdot |E| \cdot |V|)$. Our algorithm is best viewed as a reinterpretation of the classical Iterative Compression algorithm for Odd Cycle Transversal by Reed, Smith and Vetta [8].

1 Introduction

Iterative Compression is a tool that has recently been used successfully to solve a number of problems in Parameterized Complexity. This technique was first introduced by Reed, Smith and Vetta in order to solve the Odd Cycle Transversal problem. In this problem we are given a graph G together with an integer k. The objective is to find a set S of at most k vertices whose deletion makes the graph bipartite [8], and a set S such that $G \setminus S$ is bipartite is called an odd cycle transversal of G. The method of Iterative Compression was used in obtaining faster fixed parameter tractable (FPT) algorithms for Feedback Vertex Set, Edge Bipartization, Chordal Deletion and Cluster Vertex Deletion on undirected graphs [2, 3, 6, 4]. The technique was also used by Chen et al. [1] to show that the Directed Feedback Vertex Set problem is FPT, resolving a long standing open problem in Parameterized Complexity.

While the algorithm of Reed, Smith and Vetta for Odd Cycle Transversal was a breakthrough for parameterized algorithms, the algorithm and correctness proof is quite hard to understand. In an attempt to remedy this, Hüffner [5] provided an alternative algorithm for the problem. In this paper we give yet another algorithm for OCT. We believe that our algorithm is simpler and more intuitive than the previous versions.

*The University of Bergen, Norway. {daniello|saket.saurabh}@ii.uib.no
†The Institute of Mathematical Sciences, India. somnath@imsc.res.in

2 The Method of Iterative Compression

The method of Iterative Compression was introduced by Reed et al. [8] in order to solve the Odd Cycle Transversal (OCT) problem. The idea is to reduce the problem in question to a modified version, where we are also given as input a solution that is almost good enough, but not quite. For the case of Odd Cycle Transversal, we are given an odd cycle transversal S' of G of size $k+1$. We call this problem the *compression* version of Odd Cycle Transversal. The crux of the Iterative Compression method is that often the compression version of a problem is easier to solve than the original one.

Suppose we could solve the compression version of the problem in $O(f(k)n^c)$ time. We show how to solve the original problem in $O(f(k)n^{c+1})$ time. Order the vertices of $V(G)$ into $v_1 v_2 \ldots v_n$ and define $V_i = \{v_1 \ldots v_i\}$ for every i. Notice that if G has an odd cycle transversal S of size k then $S \cap V_i$ is an odd cycle transversal of $G[V_i]$ for every $i \leq n$. Furthermore, if S is an odd cycle transversal of $G[V_i]$ then $S \cup \{v_{i+1}\}$ is an odd cycle transversal of $G[V_{i+1}]$. Finally, V_k is an odd cycle transversal of size k of $G[V_k]$. These three facts together with the $f(k)n^c$ algorithm for the compression version of OCT give a $f(k)n^{c+1}$ time algorithm for OCT as follows. Call the algorithm for the compression version with input $(G[V_{k+1}], V_{k+1}, k)$. The algorithm will either report that $(G[V_{k+1}, k])$ has no odd cycle transversal of size k or return such an odd cycle transversal, call it S_{k+1}. In the first case G has no k-sized odd cycle transversal. In the second, call the algorithm for the compression version with input $(G[V_{k+2}], S_{k+1} \cup \{v_{k+2}\}, k)$. Again we either receive a "no" answer or a k-sized odd cycle transversal S_{k+2} of $G[V_{k+2}]$ and again, if the answer is negative then G has no k-sized odd cycle transversal. Otherwise we call the compression algorithm with input $(G, S_{k+2} \cup \{v_{k+3}\}, k)$ and keep going on in a similar manner. If we receive a negative answer at some step we answer that G has no k-sized odd cycle transversal. If we do not receive a negative answer at any step, then after $n-k$ calls to the compression algorithm we have a k-sized odd cycle transversal of $G[V_n] = G$. Thus we have resolved the input instance in time $O(f(k)n^{c+1})$. We refer to [7] for a more thorough introduction to Iterative Compression.

3 An algorithm for Odd Cycle Transversal

We now show how to solve the compression version of Odd Cycle Transversal in time $O(3^k \cdot k \cdot |E|)$. From the discussion in Section 2 it will follow that OCT can be solved in time $O(3^k \cdot k \cdot |E| \cdot |V|)$. For two vertex subsets V_1 and V_2 of $V(G)$ a walk from V_1 to V_2 is a walk with one endpoint in V_1 and the other in V_2, or a single vertex in $V_1 \cap V_2$. The following is a simple fact about bipartite graphs.

Fact 3.1 *Let $G = (V_1 \uplus V_2, E)$ be a bipartite graph with vertex bipartition $V_1 \uplus V_2$. Then*

1. *For $i \in \{1,2\}$, no walk from V_i to V_i has odd length.*

2. *No walk from V_1 to V_2 has even length.*

A *walk* in a graph is an alternating sequence of vertices and edges $v_0, e_1, v_1, e_2, \ldots, v_n$, such that $e_i = (v_{i-1}, v_i)$ is an edge for $i \in \{1, \ldots, n\}$. The length l of a walk is the number of edges used in the sequence.

Recall that we are given a graph G and an odd cycle transversal S' of G of size $k+1$ and we have to decide whether G has an odd cycle transversal of size at most k. If such an odd cycle transversal S exists then there exists a partition of S' into $L \uplus R \uplus T$, where $T = S' \cap S$ and L and R are subsets of the left and right bipartitions of the resulting graph. The algorithm iterates over all 3^k partitions of S into $L \uplus R \uplus T$. For each partition we run an algorithm that takes as input a partition of S' into $L \uplus R \uplus T$, runs in $O(k \cdot |E|)$ time and decides whether there exists a set of vertices T' of size at most $k - |T|$ in $G \setminus S'$ such that $G \setminus (T \cup T')$ is bipartite with bipartitions V_L and V_R such that $L \subseteq V_L$ and $R \subseteq V_R$. In the remainder of this section we give such an algorithm. This algorithm together with the outer loop over all partitions of S' yields the $O(3^k \cdot k \cdot |E|)$ time algorithm for the compression step.

Before proceeding we do a simple "sanity check". If there is an edge in $G[L]$ or $G[R]$ it is clear that X can not exist since then either V_L or V_R can not be an independent set. Hence if there is an edge in $G[L]$ or $G[R]$ we can immediately skip to the next partition of S'. Now, since $G \setminus S'$ is bipartite, let $A \uplus B$ be a bipartition of $G \setminus S'$. Let A_l and B_l be the neighbors of L in A and B respectively. Similarly let A_r and B_r be the neighbours of R in A and B respectively.

Lemma 3.2 *Let (G, S', k) be an instance of the compression version of* Odd Cycle Transversal *and let $S' = L \uplus R \uplus T$. If $X \subseteq (V(G) \setminus S')$ is a set of vertices such that $G \setminus (T \cup X)$ is bipartite with bipartitions V_L and V_R such that $L \subseteq V_L$ and $R \subseteq V_R$, then in $G \setminus (S' \cup X)$, there are no paths between A_l and B_l; B_l and B_r; B_r and A_r; and, A_r and A_l.*

Proof. Any path from A_l to B_l in $G \setminus (S' \cup X)$ has odd length and can be extended to a walk from L to L of odd length in $G' \setminus (T \cup X)$, contradicting Fact 3.1. A symmetric argument shows that there are no paths between B_r and A_r in $G \setminus (S' \cup X)$. Any path from B_l to B_r in $G \setminus (S' \cup X)$ must be of even length and can be extended to a walk in $G \setminus (T \cup X)$ from L to R of even length, again contradicting Fact 3.1. A symmetric argument yields that there are no paths between A_r and A_l. ∎

Lemma 3.3 *Let (G, S', k) be an instance of the compression version of* Odd Cycle Transversal *and let $S = L \uplus R \uplus T$ such that $G[L]$ and $G[R]$ are independent sets. Let X be a set of vertices in $V(G) \setminus S'$ such that in $G \setminus (S' \cup X)$, there are no paths between A_l and B_l; B_l and B_r; B_r and A_r; and, A_r and A_l. Then $G \setminus (T \cup X)$ is bipartite with bipartitions V_L and V_R such that $L \subseteq V_L$ and $R \subseteq V_R$.*

Proof. Notice that every path from a vertex in L to another vertex in L with inner vertices only in $V(G) \setminus (S' \cup X)$ must have even length. Similarly every path from a vertex in R to another vertex in R with inner vertices only in $V(G) \setminus (S' \cup X)$ must have even length and every path from a vertex in L to a vertex in R with inner vertices only in $V(G) \setminus (S' \cup X)$ must have odd length. Since $G[L]$ and $G[R]$ are independent sets it follows that if $G \setminus (T \cup X)$ is bipartite then it has bipartitions V_L and V_R such that $L \subseteq V_L$ and $R \subseteq V_R$. We now prove that $G \setminus (T \cup X)$ is bipartite.

Consider a cycle in $G \setminus (T \cup X)$. If C does not contain any vertices of $(L \cup R)$ then $|E(C)|$ is even since $G \setminus S'$ is bipartite. Let $v_1, v_2, \ldots v_t$ be the vertices of $(L \cup R) \cap C$ in their order of appearance along C. Let $v_0 = v_t$, then we have that $|E(C)| = \sum_{i=0}^{t-1} d_c(v_i, v_{i+1})$. But then $E(C)$ must be even since the number of indices i such that $v_i \in L$ and $v_{i+1} \in R$ is equal to the number of indices j such that $v_j \in R$ and $v_{j+1} \in L$. This concludes the proof. ∎

To check whether $G \setminus T$ has an odd cycle transversal X such that $G \setminus (T \cup X)$ is bipartite with bipartitions V_L and V_R such that $L \subseteq V_L$ and $R \subseteq V_R$ we proceed as follows. Construct an auxiliary graph $\widetilde{G}$ from $G \setminus S'$ by introducing two special vertices s, t and connecting s to each vertex in $A_l \cup B_r$ and t to each vertex in $A_r \cup B_l$. Lemmas 3.2 and 3.3 show that it is sufficient to check whether there is an st-separator in $\widetilde{G}$ of size at most $k - |T|$. This can be done using max flow in time $O(k \cdot |E|)$. These discussions together with Lemmata 3.2 and 3.3 bring us to the following theorem.

Theorem 3.4 *There is an algorithm that given a graph $G = (V, E)$ and integer k decides whether G has an* OCT *of size at most k in time $O(3^k \cdot k \cdot |E| \cdot |V|)$.*

4 Concluding Remarks

In this paper we gave an alternate algorithm for Odd Cycle Transversal based on the Iterative Compression technique. Traditionally, algorithms that use Iterative Compression partition the given $k+1$-sized solution into two parts. Our algorithm is the first to partition this set in three parts. This is a key element in deriving our algorithm. We believe that partitioning the given $k+1$-sized solution into more than two parts will be useful in designing Iterative Compression based algorithms.

Acknowledgments

The authors would like to thank Michael Fellows, Fedor V. Fomin, Rolf Niedermeier, Venkatesh Raman and Frances Rosamond for helpful discussions and for suggesting to put this article in print.

References

[1] J. Chen, Y. Liu, S. Lu, I. Razgon, B. O'Sullivan, A Fixed-Parameter Algorithm for the Directed Feedback Vertex Set Problem, Journal of the ACM, 55(5) (2008).

[2] F. Dehne, M. Fellows, M. Langston, F. Rosamond, and K. Stevens, An $O(2^{O(k)}n^3)$ FPT algorithm for the undirected feedback vertex set problem, Theory of Comput. Syst., 41(3), 479–492 (2007).

[3] J. Guo, J. Gramm, F. Hüffner, R. Niedermeier, and S. Wernicke. Compression-based fixed-parameter algorithms for feedback vertex set and edge bipartization, J. Comput. Syst. Sci. 72(8), 1386–1396 (2006).

[4] F. Hüffner, C. Komusiewicz, H. Moser, and R. Niedermeier, Fixed-parameter algorithms for cluster vertex deletion, in the proceedings of LATIN'08, LNCS 4957, 711-722 (2008).

[5] F. Hüffner, Algorithm engineering for optimal graph bipartization, Journal of Graph Algorithms and Applications, 13(2), 77–98 (2009).

[6] D. Marx, Chordal deletion is fixed-parameter tractable, in the proceedings of WG'06, LNCS 4271, 37–48 (2006).

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

[8] B. Reed, A. Vetta and K. Smith, Finding Odd Cycle Transversals, Operations Research Letters, 32, 229–301 (2004).