Finding Odd Cycle Transversals

Bruce Reed, Kaleigh Smith, and Adrian Vetta*

McGill University
School of Computer Science

September 12, 2003

Abstract

We present an $O(mn)$ algorithm to determine whether a graph G with m edges and n vertices has an odd cycle cover of order at most k, for any fixed k. We also obtain an algorithm that determines, in the same time, whether a graph has a half integral packing of odd cycles of weight k.

1 The Algorithm

Finding a minimum *transversal* (or *cover*) for the set of odd cycles in a graph is a fundamental problem in combinatorial optimization. Determining a minimum edge transversal is equivalent to the classic NP-complete problem, *maximum cut* [1]. Moreover, an easy reduction shows that the minimum vertex transversal problem contains the maximum cut problem as a special case. The vertex version also has direct applications in combinatorial biology (see, for example, [5]). In these applications, the transversals are typically small.

We can, trivially, determine whether a graph has an odd cycle transversal containing k vertices via brute force enumeration in $O(mn^k)$ time. In this note, we present, for each fixed k, an algorithm with the following specifications:

Algorithm OddCycleCover(k)
Input: A graph G.
Output: Either a set Y of at most k vertices of G which intersects every odd cycle, or the information that no such set exists.
Running time: $O(4^k kmn)$.

Our algorithm is recursive, we choose an arbitrary vertex v and then call OddCycleCover(k) on the graph $G - v$. If $G - v$ contains no odd cycle transversal of magnitude at most k then neither does G, and we return this fact. Otherwise, we let X' be the odd cycle transversal returned by the recursive call and note that $X = X' + v$ is an odd cycle transversal of G of size at most $k+1$. If $|X| \leq k$ then we simply return X. Otherwise the algorithm implements a subroutine which, when given an odd cycle transversal with $k+1$ vertices, determines whether the graph has an odd cycle transversal with k or fewer

*Email: {reed,kaleigh,vetta}@cs.mcgill.ca

vertices. The subroutine outputs such a transversal if one exists and runs in $O(4^k km)$ time. Hence the overall running time of the algorithm is as claimed. It remains only to describe the subroutine.

2 Transversals and Flows

Given a graph G, an odd cycle transversal X of G, and a partition of $G-X$ into two stable sets S_1 and S_2, we construct an auxiliary bipartite graph G' as follows. The vertex set of the auxiliary graph is $V(G') = V(G) - X + \{x_1, x_2 : x \in X\}$. We maintain a one-to-one correspondence between the edges of G and the edges of the auxiliary graph by the following scheme:

(i) For an edge e of $G-X$ there is a corresponding edge in G' with the same endpoints.

(ii) For an edge $e \in G$ joining a vertex y in S_i to a vertex x in X the corresponding edge of G' joins y to x_{3-i}.

(iii) For an edge $e \in G$ joining two vertices x and y in X then the corresponding edge of G' either (arbitrarily) joins x_1 to y_2 or joins y_1 to x_2.

Observe that G' has a 2-colouring using the stable sets $S'_1 = S_1 \cup \{x_1 : x \in X\}$ and $S'_2 = S_2 \cup \{x_2 : x \in X\}$. Given a subset Y of X we say that a partition of $Y' = \{y_1, y_2 : y \in Y\}$ into two sets Y_A and Y_B is *valid* if, $\forall y \in Y$, exactly one of $\{y_1, y_2\}$ is in Y_A. Our interest in the auxiliary graph is then motivated by the following lemma.

Lemma 1 *An odd cycle transversal X is of minimum size if and only if for any valid partition (Y_A, Y_B) of any subset Y of X there are $|Y|$ vertex disjoint paths from Y_A to Y_B in $G' - \{x_1, x_2 : x \in X - Y\}$.*

Proof.
($\Leftarrow$) Take a transversal X and suppose that Z is a smaller transversal whose removal produces two stable sets $\bar{S}_1$ and $\bar{S}_2$. Set $Y = X - Z$, with valid partition (Y_A, Y_B) defined by:

$$\begin{aligned} Y_A &= \{y_1 : y \in Y \cap \bar{S}_1\} \cup \{y_2 : y \in Y \cap \bar{S}_2\} \\ Y_B &= \{y_2 : y \in Y \cap \bar{S}_1\} \cup \{y_1 : y \in Y \cap \bar{S}_2\} \end{aligned}$$

We claim that there are not $|Y|$ vertex disjoint paths from Y_A to Y_B in $G' - (X - Y)$ (more formally, we mean, $G' - \{x_1, x_2 : x \in X - Y\}$). To prove this we will show that $Z - X$ is a cutset of $G' - (X_Y)$ separating Y_A from Y_B. Since $|Z - X| < |Y|$, this gives the claim. Take a minimal path P from Y_A to Y_B in

$$G' - (X - Y) - (Z - X) = G' - (X \cap Z) - (Z - X) = G' - Z$$

We may assume that the endpoints of P are u and v respectively. Moreover, by minimality, none of the internal vertices of P are in Y'. By symmetry, we may also assume either that $\{u, v\} \in Y \cap \bar{S}_1$ or that $u \in Y \cap \bar{S}_1$ and $v \in Y \cap \bar{S}_2$. Consider now the former case. We have that $u = y_1$ for some $y \in Y$ and $v = w_2$ for some $w \in Y$. Therefore, the path P has odd length. However, by definition, P is a path in $G' - Z$ and hence, must have even length as $u, v \in \bar{S}_1$. We obtain a similar contradiction in the latter case.

($\Rightarrow$) Take a minimum transversal X whose removal produces two stable sets S_1 and S_2. Let $Y \subseteq X$ have a valid partition (Y_A, Y_B) with less than $|Y|$ vertex disjoint paths from Y_A to Y_B in $G' - (X - Y)$. Then take a cutset W separating Y_A from Y_B of magnitude less than $|Y|$ in $G' - (X - Y)$.

We claim that $G - (X - Y + W)$ is bipartite. Clearly it is enough to show that there is a two colouring of $G' - W$ such that for each $y \in Y$, y_1 and y_2 receive the same colour. Because no component of $G - (X - Y + W)$ contains both a vertex of Y_A and a vertex of Y_B, we can obtain a two colouring as follows. For each component U, if U intersects Y_B then the vertices of U coloured $i \in \{1, 2\}$ are those in S_i, otherwise they are those in S_{3-i}. By our choice of Y_A and Y_B, y_1 and y_2 have the same colour in this colouring and hence $G - (X - Y + Z)$ is bipartite. Thus $X - Y + Z$ is an odd cycle transversal of smaller magnitude than X. $\square$

Using this lemma, we obtain our aforementioned subroutine. Given an odd cycle transversal X of magnitude $k + 1$, we can test for a smaller transversal by solving a maximum flow problem for each choice of sets Y_A and Y_B as in the statement of the lemma. There are $2^{|X|} = 2^{k+1}$ choices for Y and, given Y, there are $2^{|Y|} \leq 2^{k+1}$ choices for Y_A and Y_B. Since we can solve a maximum flow problem with demand d on a graph with m edges in $O(dm)$ time, our subroutine can indeed be implemented in $O(4^k km)$ time.

3 Concluding Remarks

It is shown in [3] that, for all k, there is an n_k such that every graph either has a set of $2k$ odd cycles such that each vertex is in at most two of these cycles, or it has an odd cycle transversal with at most n_k vertices. Furthermore, as described in the last section of [3], given an odd cycle transversal of order n_k, we can test for the existence of such a set of $2k$ odd cycles in $O(nm)$ time. Using this and OddCycleCover(n_k) gives an $O(nm)$ algorithm for testing for the existence of a half integral packing of $2k$ odd cycles.

Using results of Robertson and Seymour [4] about routing in graphs with large clique minors and the techniques of [3], we can actually preprocess our input to the algorithm OddCycleCover(k) so that it has $O(n)$ edges. This preprocessing takes $O(n^2)$ time and improves the running time of the algorithm to match this. We omit the messy argument.

References

[1] R. Karp, "Reducibility among Combinatorial Problems", in R. Miller and J. Thatcher eds., *Complexity of Computer Computations*, Plenum Press, pp85-103, 1972.

[2] M. Garey, D. Johnson and L. Stockmeyer, "Some Simplified NP-complete Graph Problems", *Theoretical Computer Science*, **1**, pp237-267, 1976.

[3] B. Reed, "Mangoes and Blueberries", *Combinatorica*, **19(2)**, pp267-296, 1999.

[4] N. Robertson and P. Seymour, "Graph Minors. XIII. The Disjoint Paths Problem", *Journal of Combinatorial Theory Series B*, **63(1)**, pp65-110, 1995.

[5] R. Rizzi, V. Bafna, S. Istrail and G. Lancia, "Practical Algorithms and Fixed-Parameter Tractability for the Single Individual SNP Haplotyping Problem", *Algorithms in Bioinformatics: Second International Workshop*, LNCS **2452**, pp29-43, 2002.