A better polynomial-time algorithm for finding minimum cycle bases in undirected graphs

E. Amaldi[1], C. Iuliano[1] and R. Rizzi[2]

[1] Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy
amaldi@elet.polimi.it, iuliano@gmail.com
[2] Dipartimento di Matematica ed Informatica, Università di Udine, Italy
rrizzi@dimi.uniud.it

Abstract

Let $G = (V, E)$ be an undirected graph with $n = |V|$ nodes and $m = |E|$ edges. Assume without loss of generality that G is simple, that is without loops and multiple edges. An *elementary cycle* is a connected subset of edges such that all incident nodes have degree 2. A *cycle* is a subset of edges such that every node of V is incident to an even number of edges of the cycle. Cycles can be viewed as the (possibly empty) union of edge-disjoint elementary cycles. The composition of two cycles is defined as the symmetric difference of the corresponding edge sets. Each cycle C can be represented by an edge incidence vector χ_C in $\{0,1\}^m$, where $\chi_C(e) = 1$ precisely when $e \in C$. Associated with any undirected graph G there is a vector space over $GF(2)$, called the *cycle space*, consisting of the edge incidence vectors of all cycles, including the null cycle. If G has p connected components, the dimension of this space is $\nu(G) = m - n + p$. A maximal set of linearly independent cycles is called a *cycle basis*.

We consider the following combinatorial optimization problem related to cycle bases, known as the *minimum cycle basis* problem.

> MIN CB: Given a connected graph G with a nonnegative weight $w(e)$ assigned to each edge $e \in E$, find a cycle basis B of minimum total weight, i.e., which minimizes $w(B) = \sum_{i=1}^{m-n+1} w(b_i)$, where $w(b_i)$ is the sum of the weights of all edges in cycle b_i.

Short cycle bases are of interest in a variety of fields including, for instance, electric networks, structural engineering, chemistry and biochemistry, and surface reconstruction from point clouds.

MIN CB has been attracting considerable attention. The first polynomial-time algorithm was proposed by Horton [1] and has a $O(m^3 n)$ time complexity. The approach proceeds into two steps. First Horton notices that a subset of $O(mn)$ candidate cycles contains a minimum cycle basis. This set of candidate cycles, denoted by $\mathcal{H}$, is obtained by considering for each node v and each edge e of G the cycle consisting of the edge e and of the two shortest paths from v to the two endpoints of e. Since $\mathcal{H}$ turns out to be a matroid, a minimum cycle basis can then be found by selecting the $\nu = m - n + 1$ shortest linearly independent candidates cycles, by using Gaussian elimination. Exploiting fast matrix multiplication, the overall algorithm complexity can be reduced to $O(m^\omega n)$ [2], where ω is the exponent of fast matrix multiplication.

A different type of approach was proposed by de Pina [3] and improved in [5]. In these algorithms the cycles to be included in a minimum cycle basis are determined sequentially. Consider any spanning tree T of G, let $e_1, \ldots, e_N$ be the edges of $E \backslash T$ and $e_{N+1}, \ldots, e_m$ the co-tree edges, both in some arbitrary but fixed order.

Any cycle of G can be viewed as a restricted incidence vector in $\{0,1\}^N$ and it is easy to verify that linear independence of the restricted vectors is equivalent to linear independence of the full incidence vectors. Suppose that the cycles $C_1, \dots, C_i$ of a minimum cycle basis have already be determined. To determine cycle C_{i+1}, we first compute a non-zero support vector $S_{i+1} \in \{0,1\}^N$ that is orthogonal to the linear subspace generated by the cycles computed so far, namely such that $< C_j, S_{i+1} >= 0$ for all $1 \leq j \leq i$. Then cycle C_{i+1} is the shortest cycle C in the graph G such that $< C, S_{i+1} >= 1$. Since C_{i+1} is not orthogonal to S_{i+1}, C_{i+1} is linear independent w.r.t. $C_1, \dots, C_i$. The optimality of the resulting cycle basis is guaranteed by the fact that such a shortest cycle C is selected at each step. The overall complexity of the variant presented in [5] is $O(m^2 n + mn^2 \log n)$.

In a third type of algorithm described in [7], only the Horton cycles that contain a node of a close-to-minimum feedback vertex set are considered. Although this problem of finding a minimum set of nodes whose deletion makes a graph acyclic is NP-hard, it can be approximated in polynomial time within a factor 2. A simple way to extract a minimum cycle basis from the resulting set of candidate cycles then leads to an overall $O(m^2 n + mn^2)$ complexity, which can be further reduced to $O(m^2 n/\log(n) + mn^2)$ by using a bit-packing trick.

Surprisingly very little attention has been devoted so far to evaluate the actual performance of these algorithms and we are not aware of any computational comparison carried out on a set of benchmark instances.

In this work we revisit Horton's and de Pina's approaches and propose a hybrid algorithm which has $O(m^2 n/\log(n))$ worst-case complexity and outperforms in practice previous algorithms on a variety of instances. On the one hand, we show that the size of Horton's set of candidate cycles $\mathcal{H}$ can be substantially reduced. More precisely, we describe simple and efficient procedures to identify candidate cycles in $\mathcal{H}$ that can be deleted since they can be obtained by composing two shorter Horton cycles or k shorter Horton cycles according to a wheel structure, for some integer $k \geq 3$. Interestingly, although the resulting reduced set $\mathcal{H}'$ is still of cardinality $O(mn)$, the incidence vectors of these cycles are very sparse: they contain at most mn ones. Similarly to Horton's algorithm, the candidate cycles in $\mathcal{H}'$ are then considered in nondecreasing order of their weight. To test whether a candidate cycle at hand is linearly independent from those that have been previously selected, we propose an improved linear independence test à la de Pina. The idea is to reduce the overall computational load by progressively and appropriately building the spanning tree whose co-tree edges determine a basis of the linear subspace that is orthogonal to the linear subspace generated by the cycles selected so far.

References

1. J. D. Horton (1985). A polynomial-time algorithm to find the shortest cycle basis of a graph. *SIAM J. on Computing*, 16(2), 358-366.
2. A. Golynski, J. D. Horton (2002). A polynomial time algorithm to find the minimum cycle basis of a regular matroid. Proceedings of SWAT 2002, *Lecture Notes in Computer Science*, Vol. 2368, 200-209.
3. J. C. de Pina (1995). Applications of shortest path methods. PhD thesis, University of Amsterdam, The Netherlands.
4. F. Berger, P. Gritzmann, S. de Vries (2004). Minimum cycle basis for network graphs. *Algoritmica*, 40(1), 51-62.
5. T. Kavitha, K. Mehlhorn, D. Michail, K. Paluch (2004), A faster algorithm for minimum cycle basis of graphs. Proceedings of 31st ICALP, 846-857.
6. T. Kavitha, K. Mehlhorn, D. Michail (2007), New approximation algorithms algorithm for minimum cycle bases in graphs. Proceedings of STACS, *Lecture Notes in Computer Science*, Vol. 4393, 512-523.
7. K. Mehlhorn, D. Michail (2007), Minimum cycle bases: faster and simpler. Manuscript.