On Finding a Cycle Basis with a Shortest Maximal Cycle

David M. Chickering Dan Geiger * David Heckerman

Microsoft Corporation
One Microsoft Way
Redmond 98052, WA

Abstract

The Shortest Maximal Cycle Basis (SMCB) problem is that of finding a cycle basis B of a given graph G such that the length of the longest cycle included in B is the smallest among all bases of G. We show that any cycle basis B' of G such that the sum of the lengths of the cycles included in B' is the smallest among all cycle bases of G constitutes a solution to the SMCB problem. Finding a basis with the latter property requires at most $O(m^3 n)$ steps using Horton's algorithm where m is the number of edges and n is the number of vertices.

Key words: algorithms, analysis of algorithms, combinatorical problems, coding theory.

*Author's primary affiliation: Computer Science Department, Technion, Haifa 32000, Israel.

1 Introduction

The Shortest Cycle Basis (SCB) problem is that of finding a cycle basis B of a given graph G with the property that the sum of the lengths of the cycles included in B is the smallest among all bases of G. This problem and some variations of it were dealt in several articles (e.g., [St64, Zy69, HS75, De79, Sy79, Ko80, CRS81]). The latest reference is Horton (87) who establishes a polynomial algorithm having a time complexity of $O(m^3n)$ where m is the number of edges and n is the number of vertices in G. Horton states that his algorithm is the first polynomial algorithm that actually solves this problem and brings citations that contain counter-examples to the attempts made prior to his.

The Shortest Maximal Cycle Basis (SMCB) problem is that of finding a cycle basis B of a given graph G with the property that the length of the longest cycle included in B is the smallest among all bases of G. We show that any basis that constitutes a solution to the SCB problem is also a solution to the SMCB problem. This correspondence between the two problems is shown to hold in any vector space over GF_2, not necessarily those induced by cycles of a graph. Consequently, Horton's polynomial algorithm for the SCB problem solves the SMCB problem as well.

2 Definitions and Basic Properties

By a *graph* $G = (V(G), E(G))$ we mean an undirected graph with no self-loops or parallel edges where $V(G)$ is the set of vertices and $E(G)$ is the set of edges. A connected subgraph C is called a *simple cycle* if each vertex is incident to two edges in C. A subgraph C is called a *cycle* if each vertex has an even degree in C. Note that a cycle need not be a connected subgraph and that a simple cycle is a cycle. Each cycle C can be written as a vector of length $|E(G)|$ having 1 in each location that corresponds to an edge in $E(C)$ and having 0 otherwise. The *sum* $C_1 \oplus C_2$ of two cycles C_1 and C_2 is the subgraph induced by the edges $(E(C_1) \cup E(C_2)) \setminus (E(C_1) \cap E(C_2))$. Equivalently, the vector corresponding to $C_1 \oplus C_2$ is the sum mod 2 of the vectors corresponding to C_1 and C_2. The set of vectors corresponding to all cycles of a graph form a linear vector space over GF_2. (GF_2 is the field with constants $\{0, 1\}$ and addition taken mod 2.) We will call this linear vector space the *cycle space* and a basis for this vector space will be called a *cycle basis*. The dimension of the cycle space is $|E(G)| - |V(G)| + 1$ [TS92].

We use the following basic properties of vector spaces.

Proposition 1 *Let B be a basis of a vector space $\mathcal{V}$. If any vector v in B is replaced by the sum of v and a linear combination of the vectors in $B \setminus \{v\}$, then the resulting set of*

vectors is a basis of $\mathcal{V}$.

Proposition 2 *Let* $\{v_1, \ldots, v_k\}$ *be a basis of a vector space* $\mathcal{V}$ *over* GF_2 *and let* $\{u_1, \ldots, u_k\}$ *be another basis of* $\mathcal{V}$. *Then, there exists a permutation* θ *of* $\{1, \ldots, k\}$ *such that for* $i = 1, \ldots, k$ *each* $u_{\theta(i)}$ *can be written as the sum of* v_i *and a linear combination of* $\{v_1, \ldots, v_k\} \setminus \{v_i\}$.

Proof. Let $M = (m_{i,j})$ be the non-singular matrix that maps the first basis to the second one and let ϕ denote a permutation of $\{1, \ldots, k\}$. Since M is non-singular its determinant $\sum_\phi m_{1,\phi(1)} m_{2,\phi(2)}, \ldots, m_{k,\phi(k)}$ (sum taken mod 2) must contain at least one addend in which all factors are 1. Let θ be a permutation that corresponds to any such addend. θ satisfies the condition of this proposition. □

3 Main Result

Define the *length* of a vector v, denoted by $|v|$, to be the number of 1's that it contains. The *shortest basis* of a vector space is a basis B in which the sum of the lengths of all vectors in B is minimized. Consider the following exponential greedy algorithm for finding the shortest basis of a vector space.

Algorithm FindBasis(*Input:* **A basis** $v_1, \ldots, v_k$ **of a vector space** $\mathcal{V}$**;**
Output: **A shortest basis of** $\mathcal{V}$**);**

Until no changes occur;
  For $i = 1$ **to** k **do;**
    $v_i \leftarrow v_i \oplus \alpha(v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_k)$
      where $\alpha(v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_k)$ **is a linear combination**
      of $\{v_1, \ldots, v_k\} \setminus \{v_i\}$ **chosen such that the updated**
      vector v_i **will have the smallest possible length;**
  end;
end;
Output$(v_1, \ldots, v_k)$**;**

The algorithm always terminates because in each step the sum of the lengths of the vectors $v_1, \ldots, v_k$ is reduced by at least one. The algorithm always outputs a basis because, due to Proposition 1, after each step $\{v_1, \ldots, v_k\}$ remains a basis of $\mathcal{V}$. Furthermore, when the algorithm stops no v_i can be improved by adding to it a linear combination of the other $k - 1$ vectors. We now argue that the algorithm always outputs a shortest basis. Let

$u_1, \ldots, u_k$ be a shortest basis of $\mathcal{V}$ and $v_1, \ldots, v_k$ be the basis FindBasis generates. Then, by Proposition 2, there exists a permutation θ of $\{1, \ldots, k\}$ such that each $u_{\theta(i)}$ equals $v_i \oplus \alpha(v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_k)$. Consequently, for each i, $|v_i| \leq |u_{\theta(i)}|$ because otherwise FindBasis would have performed an additional iteration. However, equality must hold for all i because $u_1, \ldots, u_k$ is a shortest basis. Thus, FindBasis always terminates and outputs a shortest basis. In fact, we can prove that FindBasis modifies each vector only once at the first iteration of the external loop, however, the proof of this observation is not needed herein and is thus omitted.

This discussion implies the following property of every shortest basis.

Theorem 3 *Let $\{u_1, \ldots, u_k\}$ and $\{v_1, \ldots, v_k\}$ each be a shortest basis of a vector space $\mathcal{V}$ over* GF_2 *having lengths $|u_1| \leq |u_2| \leq \ldots |u_k|$, and $|v_1| \leq |v_2| \leq \ldots |v_k|$, respectively. Then, for $i = 1, \ldots, k$, $|u_i| = |v_i|$.*

Proof. Apply FindBasis to the basis $\{v_1, \ldots, v_k\}$. We first argue that FindBasis will not make any changes to the given basis. Assume it does. Consider the set $\{v_1, \ldots, v_k\}$ just after some v_i has been changed. The resulting new set of vectors is a basis whose length is shorter than the given basis which contradicts the minimality of the given basis. We have shown previously that the outcome of FindBasis satisfies $|v_i| \leq |u_{\theta(i)}|$ for $i = 1, \ldots, k$ where θ is a permutation on $\{1, \ldots, k\}$. Since $u_1, \ldots, u_k$ is a shortest basis, equality must hold for all i. □

It is worth noting that a shortest basis of a vector space of GF_2 contains as its smallest element a vector of $\mathcal{V}$ whose length is minimum across all vectors of $\mathcal{V}$. The existence of a polynomial algorithm for finding a vector whose length is minimal in a given vector space over GF_2 is a major open question in coding theory [BMT78].

Let $L(B)$ denote the length of the longest vector in a basis B. Define the *shortest maximal basis* of $\mathcal{V}$ to be a basis B such that $L(B)$ is minimized over all bases of $\mathcal{V}$.

Theorem 4 *Let B be a shortest basis of a vector space $\mathcal{V}$ over* GF_2 *and let B' be a shortest maximal basis of $\mathcal{V}$. Then, $L(B) = L(B')$.*

Proof. Let $B = \{u_1, \ldots, u_k\}$ and $B' = \{w_1, \ldots, w_k\}$. Apply FindBasis to B' and suppose the algorithm outputs $B'' = \{v_1, \ldots, v_k\}$. The algorithm never increases the length of an updated vector. Thus $L(B'') \leq L(B')$. Since FindBasis generates a shortest basis, by Theorem 3, $L(B'') = L(B)$. Hence, $L(B) \leq L(B')$. Since B' is a shortest maximal basis, equality is implied. □

Consequently, every algorithm that finds a shortest basis also finds a shortest maximal basis. Furthermore, in the case of cycle spaces, finding the shortest (cycle) basis requires only polynomial number of steps using Horton's algorithm. The algorithm can be roughly

described as follows. First, for each pair of a vertex x and an edge (a, b) find a shortest path $p(a, x)$ between a and x and a shortest path $p(b, x)$ between b and x. Then form a cycle using $p(a, x)$, $p(b, x)$ and (a, b) unless $p(a, x)$ and $p(b, x)$ share any vertex other than x. In the latter case the degenerated cycle is ignored. Consequently, the number of cycles generated, r, is bounded by mn where m is the number of edges and n is the number of nodes. Finally a greedy algorithm is used which selects in each step the shortest cycle among the r cycles generated in the previous phase such that the newly selected cycle is independent of the previously selected ones. The last step is implemented by applying Gaussian elimination to a 0-1 matrix whose rows are the vectors corresponding to the r cycles generated in the first phase of the algorithm.

Theorem 4 shows that this algorithm also solves the SMCB problem.

Acknowledgment

We thank Jack Breese and Seffi Naor for their comments.

References

[BMT78] E.R. Berlekamp, R.J. McEliece, and H.C.A. van Tilborg, *On the inherent intractability of certain coding problems*, IEEE Trans. Inform. Theory, IT-24 (1978), 384–386.

[CRS81] D. W. Cribb, R. D. Ringeisen and D. R. Shier, *On cycle bases of a graph*, Congr. Numer., 32 (1981), pp. 221–229.

[De79] N. Deo, *Minimum length fundamental cycle set*, IEEE Trans. Circuits and Systems, 26 (1979), pp. 894–895.

[DPK82] N. Deo, G. M. Prabhu and M. S. Krishnamoorthy, *Algorithms for generating fundamental cycles in a graph*, ACM Trans. Math. Software, 8 (1982), pp. 26–42.

[HS75] E. Hubicka and M. M. Syslo, *Minimal bases of cycles of a graph*, in Recent Advances in Graph Theory, Proceeding of the symposium held in Prague, June 1974, M. Fiedler, ed., Academia Praha, Prague, 1975, pp. 283–293.

[Ho87] J. D. Horton, *A polynomial-time algorithm to find the shortest cycle basis of a graph*, SIAM J. Comput., 16 (1987), pp. 358–366.

[Ko80] E. Kolasinska, *On a minimum cycle basis of a graph*, Zastos. Mat., 16 (1980), pp. 631–639.

[St64] G. F. Stepanec, *Basis systems of vector cycles with extremal properties in graphs*, Uspekhi Mat. Nauk, 19 (1964), pp. 171–175. (In Russian.)

[Sy79] M. M. Syslo, *On cycle bases of a graph*, Networks, 9 (1979), pp. 123–132.

[TS92] K. Thulasiraman and M. N. S. Swamy, *Graphs: theory and algorithms*, Wiley, 1992.

[Zy69] A. A. Zykov, *Theory of Finite Graphs*, Nauka, Novosibirsk, 1969. (In Russian.)