Algorithms for finding minimum fundamental cycle bases in graphs

Edoardo Amaldi, Leo Liberti, Francesco Maffioli

DEI, Politecnico di Milano, Piazza L. da Vinci 32, 20133 Milano, Italy

Nelson Maculan

COPPE, Universidade Federal do Rio de Janeiro, P.O. Box 68511, 21941-972, Rio de Janeiro, Brazil

27 February 2004

Abstract

We describe new heuristics for solving the problem of finding the fundamental cycle bases of minimum cost in a simple, undirected, biconnected graph G. Since each spanning tree of G is associated to a fundamental cycle basis, edge swaps are iteratively performed on the current spanning tree so as to improve the cost of the corresponding fundamental cycle basis. Furthermore, we establish graph-theoretical structural results that allow an efficient implementation of our algorithms.

1 Introduction

Let $G = (V, E)$ be a simple, undirected and biconnected graph with n nodes and m edges, weighted by a non-negative cost function $w : E \to \mathbb{R}^+$, which is extended to sets of edges in the natural way (if $F \subseteq E$, $w(F) = \sum_{e\in F} w(e)$). A set of cycles in the graph is a *cycle basis* if it is a basis of the cycle vector space. The cost of a set of cycles is the sum of the costs of all cycles in the set. Given any spanning tree of G with edge set $T \subseteq E$, the edges in T are called *branches* of the tree, and those in $E\backslash T$ are called the *chords* of G with respect to T. Any chord uniquely identifies a cycle consisting of the chord

Email addresses: `amaldi@elet.polimi.it` (Edoardo Amaldi), `liberti@elet.polimi.it` (Leo Liberti), `maffioli@elet.polimi.it` (Francesco Maffioli), `maculan@cos.ufrj.br` (Nelson Maculan).

itself and the unique path in T connecting the two nodes incident on the chord. These $m - n + 1$ cycles are called *fundamental cycles* and they form a *Fundamental Cycle Basis* (FCB) of G with respect to T. It was shown that a cycle basis is fundamental if and only if each cycle in the basis contains at least one edge which is not contained in any other cycle in the basis [9]. Finding the Minimum Fundamental Cycle Basis (MIN FCB) of a graph is an $\mathcal{NP}$-hard problem [2]. Furthermore, it does not admit a polynomial-time approximation scheme unless $\mathcal{P} = \mathcal{NP}$; a $(4+\varepsilon)$-approximation algorithm was found for complete graphs, and a $2^{O(\sqrt{\log n \log\log n})}$-approximation algorithm for arbitrary graphs [7].

Interest in minimum FCBs arises in a variety of application fields, such as electrical circuit testing [1], periodic timetable planning [6] and generating minimal perfect hash functions [3].

2 Edge-swapping local search and metaheuristics

Our local search for the MIN FCB problem is based on an iterative improvement of a current spanning tree, obtained by performing edge swaps. We start from an initial spanning tree grown by adding nodes to the tree in such a way that short fundamental cycles are completed early in the process (based on [8]). At each iteration, we identify the edge swap between branch and chord that leads to the largest decrease in FCB cost. This edge-swapping operation is inserted in a local search procedure.

Consider any given spanning tree T of G. For each branch $e \in T$, the *fundamental cut* of G induced by e is the edge set $\delta^e_T = \{\{u,v\} \in E \mid u \in S^e_T, v \in \bar{S}^e_T\}$, where S^e_T, $\bar{S}^e_T$ is the node partition induced by the removal of e from T. For any chord $f \in \delta^e_T$, let $\pi = (e, f)$ be the edge swap which consists in removing e while adding f to T. Denote by πT the resulting spanning tree. Now for each such edge swap π we calculate the cost difference Δ_π between the FCB of T and that of πT. Let Δ_{opt} be the largest such difference, and π_{opt} be the correspoding edge swap. The local search iteratively identifies π_{opt} and updates the current T with $\pi_{\text{opt}} T$ while π_{opt} is not the identity.

Applying an edge swap to a spanning tree may change the fundamental cycles and cut structure considerably. Hence, efficient procedures are needed to determine the cuts $\delta^e_{\pi T}$ for all $e \in \pi T$, and to compute Δ_π from the data at the previous iteration, namely from T, π and the cuts δ^e_T, for $e \in T$.

Some of the following structural properties are straightforward, others can be proved by careful case enumeration.

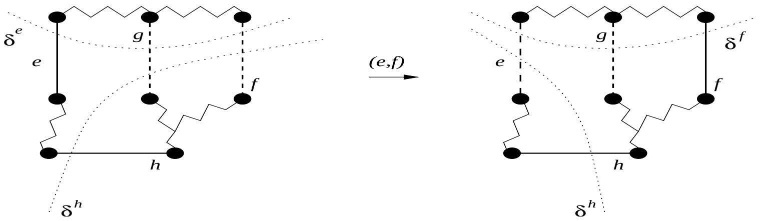

Fig. 1. Let $g \in \delta^h \cap \delta^e$. Then $g \notin \pi(\delta^h)$.

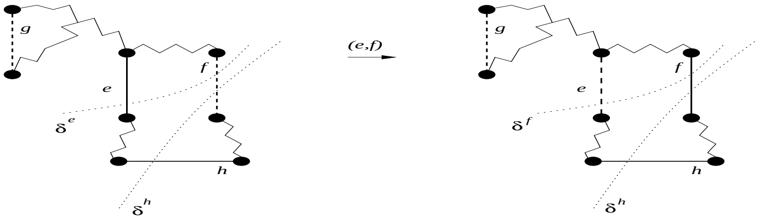

Fig. 2. Let $g \notin \delta^h \cup \delta^e$. Then $g \notin \pi(\delta^h)$.

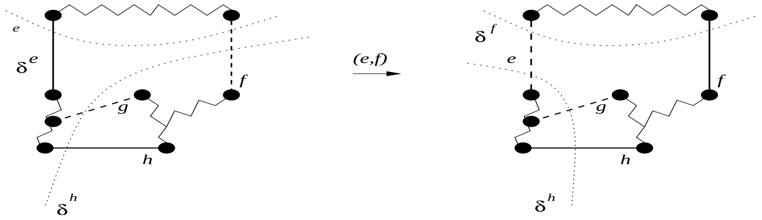

Fig. 3. Let $g \in \delta^h$ and $g \notin \delta^e$. Then $g \in \pi(\delta^h)$.

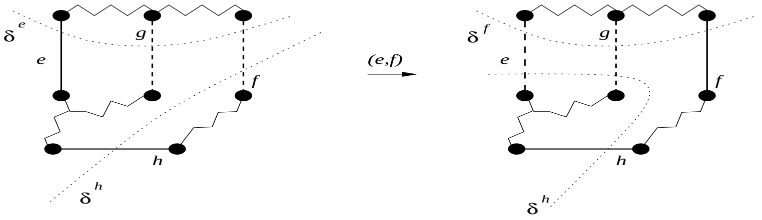

Fig. 4. Let $g \notin \delta^h$ and $g \in \delta^e$. Then $g \in \pi(\delta^h)$.

Efficient cut structure update:

- any edge swap $\pi = (e, f)$ applied to a spanning tree T, where $e \in T$ and $f \in \delta^e_T$, changes a cut δ^h_T if and only if $f \in \delta^h_T$;
- $\delta^h_{\pi T}$ can be determined by taking the symmetric difference $\delta^h_T \triangle \delta^e_T$ (see Figures 1-4 for a graphical sketch of the proof).

Efficient cycle structure update (notation: γ^h_T is the unique fundamental cycle of G w.r.t. the chord h):

- if $h \notin \delta^e_T$, then γ^h_T is unchanged by π;
- if $h \in \delta^e_T$, then $\gamma^h_{\pi T}$ can be determined by taking the symmetric difference $\gamma^h_T \triangle \gamma^f_T$.

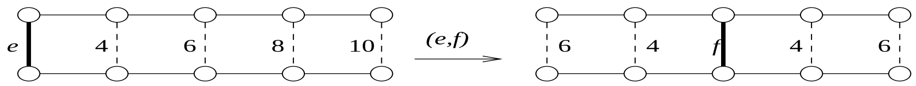

Fig. 5. All edge weights are equal to 1 and the numbers indicated on the chords correspond to the costs of the corresponding fundamental cycles. The cut on the left has a difference between cheapest and most expensive cycle of $10 - 4 = 6$; after the edge swap the difference is $6 - 4 = 2$.

It can be verified that the complexity of identifying the best edge swap π_{opt} and applying it to T to obtain πT is $O(m^2 n^2)$.

The implementation of the local search algorithm described above is computationally intensive. For large-scale problems, we would like to test the edge swap only for a small subset of pairs e, f while minimizing the chances of missing pairs which yield large cost decreases. A good strategy is to focus on branches inducing fundamental cuts whose edges define fundamental cycles with "unbalanced" costs, i.e., with a large difference between the cheapest and the most expensive of those fundamental cycles. See Fig. 5 for a simple example.

To try to escape from local minima, we have included the above edge-swap move within two well-known metaheuristics: variable neighbourhood search (VNS) [4] and tabu search (TS) [5]. We used a basic implementation of VNS. Our implementation of the Tabu search, on the other hand, is a blend of classic TS and VNS. If π_{opt} is the identity, an edge swap that worsens the FCB cost is applied to the current solution and inserted in a tabu list. If all possible edge swaps are tabu or a pre-determined number of successive non-improving moves is exceeded, t random edge swaps are applied to the current spanning tree. The number t increases until a pre-determined limit is reached, and is then re-set to 1. The procedure runs until a given termination condition is met.

3 Some computational results

We ran extensive tests over three classes of graphs.

(1) *Square mesh graphs with unit edge costs.* These are $n \times n$ square meshes with nodes positioned at (p, q) where $p, q \in \mathbb{Z}$ and $0 \leq p, q < n$ (n^2 vertices and $2n(n-1)$ edges). Because of the high degree of symmetry of the graph topology and the uniform edge costs, these are considered hard instances where previous constructive heuristics [2,3] performed badly, with FCB costs being on average three times as large as those of the solutions produced by our algorithms.

(2) *Random simple Euclidean weighted graphs.* The nodes are positioned randomly on a 20×20 square centered at the origin. Each edge between pair of nodes is randomly generated with probability p and cost equal to the Euclidean distance between its adjacent nodes. Our solutions were on average 50% better than those obtained with previous constructive methods [2,3]. For small instances (10-15 nodes) our local search actually found the optimal solutions.

(3) *Application to periodic timetabling.* This application is described in [6]. Finding minimum FCBs of appropriate graphs leads to a more compact MIP formulation of a certain type of Periodic Event Scheduling Problem (PESP). We were able to find solutions between 5% to 20% better than those found by C. Liebchen using a purpose-built modification of Deo's methods.

References

[1] A. Brambilla, A. Premoli, Rigorous event-driven (red) analysis of large-scale nonlinear rc circuits, IEEE Transactions on Circuits and Systems–I: Fundamental Theory and Applications 48 (8) (2001) 938–946.

[2] N. Deo, G. Prabhu, M. Krishnamoorthy, Algorithms for generating fundamental cycles in a graph ACM Transactions on Mathematical Software (8) (1982) 26–42

[3] N. Deo, N. Kumar, J. Parsons, Minimum-length fundamental-cycle set problem: New heuristics and an empirical investigation, Congressus Numerantium 107 (1995) 141–154.

[4] P. Hansen, N. Mladenović, Variable neighbourhood search, in: F. Glover, G. Kochenberger (Eds.), Handbook of Metaheuristics, Kluwer, Dordrecht, 2003.

[5] A. Hertz, E. Taillard, D. de Werra, Tabu search, in: E. Aarts, J. Lenstra (Eds.), Local Search in Combinatorial Optimization, Wiley, Chichester, 1997, pp. 121–136.

[6] C. Liebchen, R. H. Möhring, A case study in periodic timetabling, in: D. Wagner (Ed.), Electronic Notes in Theoretical Computer Science, Vol. 66, Elsevier, 2002.

[7] G. Galbiati, E. Amaldi, On the approximability of the minimum fundamental cycle basis problem, Workshop on Approximation and Online Algorithms (ALGO-WAOA03), Budapest (in press in the Lecture Notes in Computer Science series) .

[8] K. Paton, An algorithm for finding a fundamental set of cycles of a graph, Communications of the ACM 12 (9) (1969) 514–518.

[9] M. Sysło, On some problems related to fundamental cycle sets of a graph, in: R. Chartrand (Ed.), Theory of Applications of Graphs, Wiley, New York, 1981, pp. 577–588.