LOCAL SEARCH FOR THE MINIMUM FUNDAMENTAL CYCLE BASIS PROBLEM

E.Amaldi, L.Liberti, N.Maculan[§], F.Maffioli

DEI, Politecnico di Milano, I-20133 Milano
{amaldi,liberti,maculan,maffioli}@elet.polimi.it

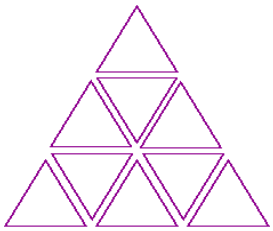
Abstract

Let G=(V,E) be a bi-connected graph with non-negative weights on its m edges. Let n be the number of vertices of G. With respect to any spanning tree T of G we have a fundamental cycle basis of G formed by the m-n+1 fundamental cycles, each one corresponding to one of the co-tree edges. The problem of determining one such basis minimising the sum of the length of its cycles (MinFCB) is known to be NP-hard. Applications exist in organic chemistry, periodic scheduling, and electrical networks. A recent result proves that no polynomial time approximation scheme can exist for MinFCB unless P=NP. The problem of determining a tree with respect to which the longest fundamental cycle is minimum is also NP-hard. When fundamentality is not required, that is when one looks for a basis of the cycle space of the graph, both problems are solvable in polynomial time. Constructive heuristics have been proposed in the literature, which are fast and an obvious first choice for very large instances. However they tend to produce solutions which are far from optimal. We report about on-going research on Local Search approaches for MinFCB. A formulation as Mixed Integer Programming problem is presented and proved effective for obtaining bounds. Preliminary computational results are reported. The final part of this work outlines some future research directions.

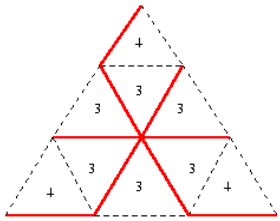
Keywords: fundamental cycle basis, local search, integer programming, bounds, timetable planning.

1. Introduction

Let $G = (V,E)$ be a simple, undirected, bi-connected graph with n nodes and m edges. A set of cycles in the graph is a *cycle basis* if it is a basis in the cycle vector space of the graph. When each edge e of G is given a cost $c(e)$, the cost of a cycle is the sum of the costs of its edges. The problem of finding a minimum cost *cycle basis* of a graph can be solved in $O(m^3 + mn^2 \log n)$ time [6].
Given any spanning tree T of G, the edges in $G \setminus T$ (the co-tree) are called *chords* of G w.r.t. T. Any chord uniquely identifies a cycle consisting of the chord itself and the unique path in T joining the two vertices incident on the chord. These $m - n + 1$ cycles are called *fundamental cycles*; they form a cycle basis which is



Cycle basis which is not fundamental



Fundamental cycle basis with cost = 30

called *fundamental cycle basis* (FCB) of G with respect to T. The set of fundamental cycles always constitutes a basis of the cycle space, whereas the opposite is not true, as shown in the figure above. A *minimum fundamental cycle basis* is a FCB having minimum cost. Finding a minimum FCB is referred to as problem **MinFCB**. Minimum FCBs arise in a variety of application fields, such as VLSI design [1], and periodic timetable planning [7], as well as being an interesting combinatorial optimisation problem in itself. Other applications are listed in [3].

MinFCB is NP-hard [2] even when the graph is uniformly weighted. Moreover, it is proved in [5] that the weighted version cannot have a PTAS unless $\mathcal{P}=\mathcal{NP}$. In the same work, a $2^{O(\sqrt{(\log n \log\log n)})}$ approximation algorithm is presented. Several heuristics have been proposed for MinFCB [2,3], all based on a "spanning tree growth" strategy: a spanning tree is constructed iteratively adding vertices and edges of G in the order

[§] On leave from COPPE, Federal University of Rio de Janeiro, Brasil.

that is likely to give a FCB having least cost. Such approaches are fast (and thus the solution methods of choice for extremely large graphs), but tend to provide solutions far more costly than the optimum one. Looking instead for a FCB which minimises the cost of its most costly cycle we have the **Min-max FCB** problem: this problem is also NP-hard as proved in [4].

The situation summarised above motivated the interest in studying the application of Local Search methods to MinFCB. The Local Search approaches experimented so far are presented in Section 2. A lower bounding method is described in Section 3. Preliminary computational results are reported in Section 4, and Section 5 outlines current lines of research.

2. Local Search approaches

Let T be a spanning tree of G. Removing any edge e of T naturally partitions T in two connected components T_1, T_2. The cut t_e consisting of e and all the edges of G having one vertex in $V(T_1)$ and the other in $V(T_2)$ is called the *fundamental cut* of e with respect to T in G. Let e,f be edges of G such that e is in T and f is in $E\backslash T$, and let $\pi = (e,f)$ be an edge exchange, called *edge swap*. We can define the action of π on the set $\mathcal{T}$ of all spanning trees of G by setting $\pi T = T'$ where T' is the spanning tree derived from T where e has been replaced by f. Note that π is well-defined on $\mathcal{T}$ if and only if f is in the fundamental cut t_e. Since any spanning tree of G gives rise to an FCB, we can define a mapping m between $\mathcal{T}$ and the set $\mathcal{F}$ of all FCBs of G. It turns out [8] that this mapping is surjective but not injective: if $t_e = \{e,f\}$ and $\pi = (e,f)$ then $m(T) = m(\pi T)$. In other words, edge swaps in fundamental cuts of cardinality 2 induce different spanning trees but the same FCB. Let T be an initial spanning tree, and $P = \{(e,f) \mid e \text{ in } T \text{ and } f \text{ in } t_e \text{ s.t. } |t_e| > 2, f \neq e\}$. For all π in P let d_π be the cost of $m(\pi T)$. Choose π in P such that d_π is minimum and replace T with πT. This *move* is inserted first into a simple local search algorithm that terminates when π is the identity. For the choice of the initial spanning tree T any of the existing fast heuristics in [2, 3] can be used.

The move can be implemented in three steps: calculating FCB costs, choosing the minimum FCB cost over a finite set, and finding fundamental cuts. The complexity of a straightforward implementation of the whole move is $O(m^2 n^2)$. This is too heavy in settings which require repeated applications of the move like in any Local Search framework. It is easy to note that at each repeated application we end up with a tree πT which differs from the initial tree T only by an edge swap. This suggests looking for a differential calculation of the set of fundamental cuts and FCB costs with respect to the chosen exchange π. In order to do so we use some results from graph theory (presented without proofs due to the size limitation of this paper) which allow us to calculate fundamental cuts and FCB costs for each π more efficiently.

Let $\pi = (e,f)$ with e in T and f in t_e. The following properties hold:

1. for any edge h in T, π changes t_h if and only if f is in t_h as well;
2. $\pi(t_e) = t_f$;
3. for any edge h in T, $\pi(t_h)$ is the symmetric difference of the edge sets t_h and t_e.

For all chords k of G with respect to T let c_k be the fundamental cycle in G defined by k. The following hold:

4. for any edge h which is not in t_e, π fixes c_h;
5. for any chord h in t_e, $\pi(c_h)$ is the symmetric difference of the edge sets c_h and c_f.

By using the above properties on fundamental cuts and cycles, we can efficiently update the set of fundamental cuts and compute the FCB cost.

We have so far experimented with: (a) a simple Local Search (LS) algorithm incorporating the move described above, (b) a Variable Neighbourhood Search (VNS) algorithm, and (c) a Tabu Search (TS) approach in which a VNS diversification strategy is adopted when the search is not giving good enough results. LS has been in some case incorporated into a multi-start (MS) approach.

3. Lower bounds

Obtaining lower bounds for MinFCB is a difficult task. When the graph exhibits some regularities and is uniformly weighted it is often the case that a fairly good lower bound can be obtained by some algebraic formula: an example is the bound $4(n-1)^2$ for the n-square mesh graph, derived from the number of "small squares" in the mesh. For weighted cases one needs formulating the problem as an Integer Programming (IP) or Mixed IP problem. Several such formulations have been tested for MinFCB. From the point of view of getting good bounds using powerful yet standard codes like CPLEX, the following has given the best results so far.

Let $G°$ be the directed graph obtained from G by substituting each edge with two arcs in opposite directions between the same pair of nodes, and weight the arcs of $G°$ setting $D_{ij} = D_{ji} = d_{ij}$ for all edges $\{i,j\}$ of E ($i < j$). Let A be the set of arcs of $G°$. We need two sets of variables: w_{ij} is a binary variable associated to edge $\{i,j\}$ and equal to 1 whenever the edge is in the spanning tree; x_{ij}^{kl} is a non-negative variable giving the amount of flow from node k to node l in arc (i,j) of A. Let $F(k)$ and $B(k)$ be, respectively, the set of arcs leaving and entering node k in $G°$. We impose a flow of 1 from node k to node l for each edge $\{k,l\}$ of G by the following constraints:

$$\Sigma_{j \in F(k)} x_{kj}^{kl} - \Sigma_{j \in B(k)} x_{jk}^{kl} = 1 \qquad \forall \{k,l\} \in E$$

$$\Sigma_{j \in F(k)} x_{ij}^{kl} - \Sigma_{j \in B(k)} x_{ji}^{kl} = 0 \qquad \forall i \neq k,l$$

The link between the two sets of variables is ensured by imposing that

$$x_{ij}^{kl} \leq w_{ij} \text{ and } x_{ji}^{kl} \leq w_{ij} \qquad \forall \{k,l\},\{i,j\} \in E$$

These together with the following simple constraint determine a connected graph with n vertices and $n-1$ edges, that is a spanning tree:

$$\Sigma_{\{i,j\} \in E} w_{ij} = n - 1$$

MinFCB then corresponds to the problem of minimising with all the constraints above the following objective function:

$$z = \Sigma_{\{k,l\} \in E} \Sigma_{(i,j) \in A} D_{ij} x_{ij}^{kl} - \Sigma_{\{i,j\} \in E} w_{ij} d_{ij} + \Sigma_{\{i,j\} \in E} (1 - w_{ij}) d_{ij}$$

or equivalently

$$\Sigma_{\{k,l\} \in E} \Sigma_{(i,j) \in A} D_{ij} x_{ij}^{kl} - 2\Sigma_{\{i,j\} \in E} w_{ij} d_{ij}$$

This formulation was solved using CPLEX stopped before the first branching iteration. The results were unsatisfactory (as expected) for mesh graphs with unit edge costs (between 30% and 150% lower than the best result obtained by Local Search). The situation is much better for randomly generated weighted graphs, for which on average we have a value of the bound off by about 8%. This however is still unsatisfactory for use in a Branch-and-Bound approach.

4. Some computational results

We have carried out extensive computational experiments with a square mesh graphs with side N. Each such graph has N^2 vertices and $2N(N-1)$ edges. We have found these graphs to be a hard challenge for MinFCB. Table 1 lists N, the FCB cost found by the simple Local Search for the corresponding N-square mesh graph, and the computational time required to perform the computation. By comparison, we have also listed the computational result of our own implementation of the NT heuristic for MinFCB described in [3], as well as the results obtained with the metaheuristics VNS and TS. All the tests have been carried out on a Pentium 4

2.66 GHz machine with 1GB RAM running Linux. The source code, in C++, has been compiled with gcc v. 3.0. The results reported in Table 1 show that LS obtains much better solutions than its tree-growth based counterpart, albeit at the price of a much higher computation time.

In order to test the heuristics with weighted graphs, we have generated some simple, bi-connected graphs where the edge costs are given by the Euclidean distance between the vertices, positioned randomly on the plane. Each edge is randomly generated with probability p. For each n in $\{10+10k \mid k=0,...,4\}$ we have tested a random graph of size n for each probability p from 0.2 to 0.8 in increasing steps of 0.2. Table 2 reports the results of the edge-swapping heuristic on these random graphs. For each n and p we can find cost and CPU time for the LS heuristic, for computing a lower bound by partially solving the MILP formulation of section 3, for the VNS heuristic, and for the TS heuristic. For some of the smaller instances, the MILP solution procedure has been allowed to terminate, thus obtaining the optimal solution (these are the bound values marked with a † in the table). Entries marked with * indicate that this is the best value found. Obvious observations are: the bound is heavy to compute and not as tight as hoped; VNS gives good results, but with high computing times; LS is fast, but not so good; TS strikes the best compromise between time and solution value.

As already mentioned in the introduction, one of the applications of the FCB problem is *periodic timetabling* of transportation systems. In particular, Liebchen [7] has worked with Berlin Underground to use optimisation techniques in designing the timetables of the underground system. The model is based on the Periodic Event Scheduling Problem (PESP). It has been shown that the number of integer variables in the model can be minimized by identifying an FCB of a suitable graph G. Furthermore, the number of discrete values that each integer variable can take is proportional to the total FCB cost. In short, good models for the PESP problem can be obtained by solving the MINFCB problem.

We have been given a test instance by C. Liebchen (called the liebchen1 instance henceforth). The liebchen1 instance has 88 vertices and 316 edges. The costs on the edges are rather uniformly distributed: the values are: 4, 5, 6, 51, 52, 53, 54, 57, 58, 59 spread in the following fashion.

cost	4	5	6	51	52	53	54	57	58	59
occurrences	61	4	10	11	15	14	60	8	37	129

The results have been encouraging (see table 3). However, there was a supplementary condition on this instance that imposed that certain edges necessarily had to be in the spanning tree solution (call this modified liebchen1 instance, with a fixed initial partial solution, liebchen3 henceforth). There were 55 fixed edges in the partial solution, out of a possible total of 87 tree branches. This severely restricted the scope of application of the edge-swapping heuristic.

Table 3 reports FCB costs and CPU times obtained using the edge-swapping local descent on the liebchen1 and liebchen3 instances, as well as the FCB costs obtained using Deo's NT heuristic, VNS and TS. The last column contains the lower bound obtained with the MINFCB formulation of section 3.

5. Future work

The Local Search approaches tested so far represent only some out of many possibilities. Our very basic implementations of VNS and TS improve substantially on the performances of the pure Local Search approach, but other variants are possible and will be tested. Another direction of research suggests to experiment with other methods for obtaining bounds. Indeed one of the IP formulations of MinFCB looks quite promising for the application of Lagrangean relaxation and related subgradient techniques, and is currently being implemented. The possibility of exploiting polynomial methods for solving the non-fundamental version of the problem in order to get bounds for the fundamental case also deserves to be given more consideration. Finally little is known about the (quite natural) matroid generalisation of MinFCB.

	Local Search		NT [3]		VNS		TS	
N	FCB cost	CPU time	FCB cost	CPU time	FCB cost	CPU time	FCB cost	CPU time
5	72	0: 00: 00	78	0: 00: 00	72	0: 00: 00	72	0: 00: 00
10	474	0: 00: 00	518	0: 00: 00	466*	0: 00: 03	468*	0: 00: 02
15	1318	0 :00: 00	1588	0: 00: 00	1296*	0: 00: 34	1296*	0: 00: 12
20	2608	0: 00: 03	3636	0: 00: 00	2576*	0: 02: 11	2598*	0: 00: 37
25	4592	0: 00: 16	6452	0: 00: 00	4466*	0: 14: 59	4534*	0: 02: 07
30	6956	0: 00: 47	11638	0: 00: 00	6846*	0: 27: 18	6950*	0: 03: 50
35	10012	0: 02: 19	16776	0: 00: 00	9936*	1: 36: 00	10004*	0: 08: 36
40	13548	0: 06: 34	28100	0: 00: 01	13366*	3: 31: 26	13530*	0: 16: 00
45	18100	0: 14: 22	35744	0: 00: 01	17890*	8: 01: 49	18092*	0: 35: 58
50	23026	0: 31: 04	48254	0: 00: 03	22790*	22: 17: 27	23008*	1: 11: 37

Table 1: Computational results (FCB cost and CPU times) on *N*x*N* mesh graphs having unit edge costs, with Local Search, Deo's NT heuristic, VNS, and TS performances (FCB costs and times (h:mm:ss)).

6. References

[1] Brambilla, A. and Premoli, A. (2001) "Rigorous Event-Driven (RED) Analysis of Large-Scale Nonlinear *RC* Circuits", *IEEE Transactions on Circuits and Systems-I*, 48(8):938-947.

[2] Deo, N., Prabhu, G.M., and Krishnamoorthy, M.S. (1982) "Algorithms for generating Fundamental Cycles in a Graph", *ACM Transactions on Mathematical Software*, 8(1):26-42.

[3] Deo, N., Kumar, N., and Parsons, J. (1995) "Minimum-length Fundamental-cycle Set Problem: New Heuristics and an Empirical Investigations", *Congressus Numerantium*, 107:141-154.

[4] Galbiati G. (2001) "On Min-max Cycle Bases", in *Proceedings ISAAC 2001* (Eades and Takaoka eds.), *LNCS 2223*, Springer-Verlag.

[5] Galbiati, G., and Amaldi, E. (2003) "On the Approximability of the Minimum Fundamental Cycle Basis Problem", *Workshop on Approximation and Online Algorithms (WAOA03), Budapest, Sept. 2003*.

[6] Gerards A.M.H., De Pina J.C., and Schrijver A. (2002) "Shortest Circuit Bases of Graphs", *to appear*.

[7] Liebchen, C. (2003) "Finding Short Integral Cycle Bases for Cyclic Timetabling", TU Berlin, Institut für Mathematik, *Internal report 2003/12*.

[8] Sysło, M. (1979) "On Cycle Bases of a Graph", *Networks* 9:123-132.

	Local search		NT [3]	VNS	TS	Lower bound
Instance	FCB cost	CPU time	FCB cost	FCB cost	FCB cost	FCB cost
liebchen1	40407	0: 06	50259	39851*	39867*	31220.534
liebchen3	56887	0: 01	-	-	-	39907.96

Table 3: Computational results on Liebchen's instances. Missing values are due to a missing implementation of the corresponding algorithm in presence of a partial initial solution.

	p=0.2							
n	LS	Time	Bound	Time	VNS	Time	TS	Time
10	216.698†	0	216.698†	0	216.698†	0	216.698†	0
20	1052.38†	0	1052.38†	0: 56	1052.38†	0	1052.38†	0
30	3315.89	0	2750.92	0: 28	3111.71*	0: 14	3315.89	0
40	4634.04	0	4065.187	16: 58	4504.84*	0: 22	4633.45*	0
50	7007.34	0: 01	6448.711	2: 38: 51	6991.53*	1: 11	7007.34	0: 02
	p=0.4							
n	LS	Time	Bound	Time	VNS	Time	TS	Time
10	472.599	0	459.305†	0: 02	459.305†	0	472.599	0
20	2021.82	0	1894.747	0: 08	2021.37*	0: 04	2021.37*	0
30	4467.13	0	4265.6	22: 56	4455.2*	0: 29	4455.2*	0: 01
40	7685.97	0: 01	-	-	7648*	1: 46	7684.53*	0: 02
50	11096.8	0: 05	-	-	11022.8*	9: 32	11073.4*	0: 12
	p=0.6							
n	LS	Time	Bound	Time	VNS	Time	TS	Time
10	581.525	0	547.406†	0: 08	547.406†	0	547.406†	0
20	2776.22	0	2627.558	0: 59	2756.6*	0: 08	2756.6*	0
30	7031.2	0	6445.83	39: 32	6979.15*	1: 13	7031.2	0: 03
40	11686.0	0: 02	-	-	11513*	6: 40	11683.4*	0: 04
50	19387.3	0: 10	-	-	19174.1*	7: 06	19174.1*	1: 06
	p=0.8							
n	LS	Time	Bound	Time	VNS	Time	TS	Time
10	992.866	0	775.838†	0: 26	775.838†	0	775.838†	0
20	3478.11	0	3164.9	2: 31	3383.45*	0: 13	3383.45*	0: 02
30	8971.78	0: 01	7823.848	1: 43: 05	8384.32*	2: 42	8930.17*	0: 02
40	14946.4	0: 07	-	-	14870.7*	5: 30	14902.2*	0: 16
50	25349.9	0: 12	-	-	25061.2*	31: 55	25245.5*	0: 53

Table 2: Computational results on weighted Euclidean random graphs: Local Search, Lower Bound, VNS and TS performances (FCB cost and times (mm:ss or h:mm:ss)).