International Conference on Automation, Control, Engineering and Computer Science (ACECS'14)
Proceedings - Copyright IPCO-2014, pp.131-137
ISSN 2356-5608

A three-phase approach for computing a fundamental cycle basis of minimal length in a graph

Mohamed Khalil Labidi [1], Zaher Mahjoub [2]

University of Tunis El Manar – Faculty of Sciences of Tunis

University Campus 2092 El Manar Tunis, Tunisia

[1] Mohamed.Khalil.Labidi@gmail.com

[2] Zaher.Mahjoub@fst.rnu.tn

Abstract — **We address the combinatorial optimisation problem (COP) of computing a fundamental cycle basis (FCB) of minimum length in a connected graph, the length being the sum of the number of arcs of the cycles of the FCB. This COP being NP-hard, we propose a three-phase heuristic for computing an approximate solution. The first phase is constructive and consists in computing an initial FCB. The second is an improving iterative procedure permitting to reduce the length of the former FCB and leading, in general, to a non fundamental cycle basis (NFCB), whereas the third phase consists in transforming the NFCB into a final FCB better than the initial one. In order to validate our theoretical contribution, we present an experimental study achieved on a series of random, benchmark and real graphs.**

Keywords — **COP, cotree, fundamental cycle basis, graph, heuristic, NP-hard, performance, spanning tree.**

I. INTRODUCTION – DEFINITIONS AND BRIEF SURVEY

A. Definitions

We begin by introducing some useful definitions. Let $G = (U, E)$ be a simple connected graph, involving n vertices (nodes) and m edges (arcs). Let μ be a cycle of G and $\vec{\mu}$ its representing vector i.e. a vector of m elements where element i $(i = 1 \dots m)$ is equal to 1 (resp. 0) if arc i of G belongs to μ. A cycle basis (CB) is a minimum number of independant cycles, denoted $\{\mu^1, \mu^2, \dots, \mu^c\}$ [10], c being the graph cyclomatic number i.e. $c = m - n + 1$, such that any cycle representing vector μ can be expressed as follows :

$$\vec{\mu} = \lambda_1 \overrightarrow{\mu^1} + \cdots + \lambda_c \overrightarrow{\mu^c}$$

A spanning tree (ST) of a given graph G, denoted $G_{ST,}$ is a spanning subgraph of G, tree-structured and involving the n nodes of G hence has $n - 1$ arcs. The remaining c arcs of G are called chords and constitute the corresponding cotree denoted G_{CT}. A fundamental cycle (FC) according to G_{ST} is a cycle obtained by adding a chord to G_{ST}. A fundamental cycle basis (FCB) associated to G_{ST} is constituted by all the c corresponding FC's. Remark that when G is connected, by successively introducing the c chords of G_{CT} to G_{ST}, we create the c cycles of the associated FCB.

As a consequence, a cycle basis $\mathcal{C} = \{\mu^1, \mu^2, \dots, \mu^c\}$ is called fundamental **if and only if** there exists no cycle of the basis whose arcs belong both to other cycles of the basis [19]. This theorem may expressed by the following formulae.

$$\forall\, \mu^i \in \mathcal{C} : \mu^i \setminus \bigcup_{\mu^j \in \mathcal{C} \setminus \{\mu^i\}} \mu^j \neq \emptyset$$

An illustrative example is depicted below for a graph where $n = 6$, $m = 9$, $c = 4$ (straight (resp. dashed) line arcs constitute a spanning tree (resp. a cotree)).

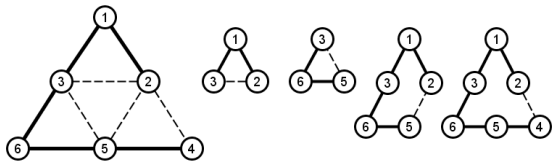

Fig. 1 FCB 1, length=17

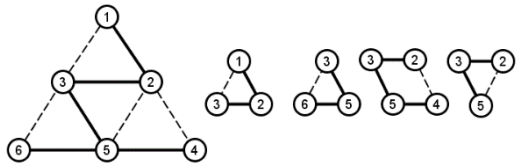

Fig. 2 FCB 2, length = 13

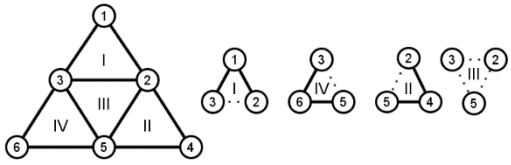

Fig. 3 NFCB, length = 12

B. Brief survey

A cycle basis may in fact be seen as a compact description of all the cycles in a graph. That is why the NP-hard problem [7] of the determination of an FCB of minimal length (FCBM) has many practical applications e.g. in network verification (since Kirchhoff in the 19th century), minimal and perfect hashing functions generation (used in compiler design), periodic scheduling, planification of complex synthesis in organic chemistry, graph drawing, surface construction… [14][21].

It has to be noticed that the FCBM problem has interested many researchers since the sixties of the previous century. In this context, four main research axes may be considered:

- Constructive heuristics

- Improving iterative heuristics
- Formulation of the FCBM as an integer programming problem
- Computing lower bounds for the length of an FCBM.

The general approach mostly adopted for solving the problem we address consists in constructing a particular spanning tree minimising the length of the corresponding FCB.

The basic idea is a judicious node-arc sorting during the ST construction. For this purpose, several search strategies are known in the literature i.e. largest first [17][13][5], depth first [20] or hybrid [9]. One or more sorting criteria are then used e.g. only one such as in SDS, DDS, UE and MBFS [7], two successive criteria when two or more candidates are equal according to the first criterion such as in ELAP [8] and UV [4]. We also find specific rankings such as UV2 and NT [6] or the C-Order [2].

Another idea consists, once a FCB of a given length is constructed, in introducing successive modifications in this FCB in order to obtain another FCB of smaller length. In 2004, the first iterative improving heuristic (IIH), called *Local Search* was designed [1]. As a matter of fact, several previous works are known in the literature, however the cycle bases they obtain, even of reduced lengths when compared to the initial FCB, are non fundamental in general [12].

Therefore, a simple and logical idea consists in starting from a non fundamental cycle basis (NFCB) of reduced length, then "*fundamentalising*" it i.e. transforming it into an FCB, even if it requires increasing the length.

The remainder of the paper is organised as follows. In section 2, we describe our three-phase approach for constructing an FCB of reduced length. Section 3 is devoted to an experimental study validating our contribution and achieved on a three graph sets : random, benchmark and real. Finally, we conclude our paper in section 4 and present some further perspectives.

II. THREE-PHASE FCB CONSTRUCTING APPROACH

A. Basic idea

The idea we adopted for constructing an FCB of reduced length consists in a three-phase procedure i.e. (i) using a constructive heuristic (CH) for the determination of an initial FCB ; (ii) using an iterative improving heuristic (IIH) for the determination of a cycle basis (non fundamental in general) of smaller length ; and (iii) using a "*fundamentalising*" heuristic (ARTF) transforming the latter into an FCB. This approach is depicted below.

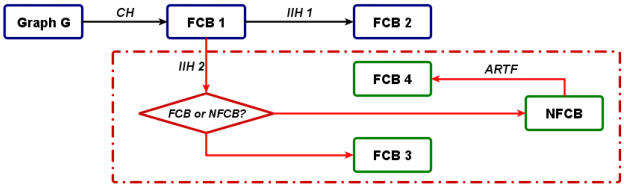

Fig. 4 Three-phase approach

B. Phase 1 : Initial FCB construction

The constructive heuristics known in the literature generally involve two steps i.e. constructing a spanning tree then the associated FCB.

1) Spanning tree construction: Due to its interesting complexity/solution quality ratio, we adopted here the choice of [1] which adapted the approach described in [17].

The constructing ST heuristic scans the nodes of the input graph in a precise order. Indeed, for a given node u, its neighbours are first examinated. Then follows a discussion on the different cases that may occur i.e. either they belong or not to the list of already visited nodes. This is done in order to know either a considered arc (u,v) creates or not a cycle when adding it to the already constructed partial spanning tree.

We can distinguish here four possible cases:

(i) None of the two nodes has been visited.
(ii) Only one node has been visited.
(iii) The two nodes have been visited but both do not belong to the same connected component (of the partial spanning tree constructed so far).

The two nodes have been visited and both belong to the same connected component (of the partial spanning tree constructed so far). In such case, (u, v) is a chord.

Since, for any graph, we have $\sum_{v \in V} \deg(v) = 2m$ where $\deg(v)$ is he degree of node v, we can deduce that such an algorithm has an $\mathcal{O}(m)$ complexity.

For the node visiting order, we used in fact both the node degree decreasing order (DEC) and ELAP (*One step at a time lookahead principle*) heuristic proposed in [8] which defines a second criterion to choose between nodes having the same degree.

2) FCB construction: Constructing an FCB, once an ST is determined, consists as previously precised in successively introducing the chords and determining the cycle induced by each chord. Here, for each chord, one has to scan all arcs incident to one node of the chord, then starting a depth first search restricted on these arcs until getting a positive response i.e. a cycle terminating at the other node of the chord.

We may see that in the worst case the search procedure may scan all the ST thus the n nodes. Having to look at every arc incident to each node of the chord, we deduce that the overall complexity is $\mathcal{O}(m)$ in the worst case.

C. Phase 2 : Improving iterative heuristic (IIH)

Concerning the improving iterative heuristic (IIH), we used the one proposed in [12][16] and based on successive logical combinations of the cycles. It permits to reduce the length of the cycle basis but does not guarantee that it is an FCB. This procedure is described below.

Let $\mathcal{C}_i$ and $\mathcal{C}_j$ be two cycles whose lengths are denoted θ_i, θ_j. We define their logical combination, denoted $\mathcal{C}_{ij}$, as follows :

$$\mathcal{C}_{ij} = \mathcal{C}_i \cup \mathcal{C}_j \setminus \mathcal{C}_i \cap \mathcal{C}_j \text{ whose length is } \theta_{ij}$$

We then eliminate among $\mathcal{C}_i$, $\mathcal{C}_j$, and $\mathcal{C}_{ij}$ the cycle with the largest length as follows :

$$if \begin{cases} \theta_i \leq \theta_j \\ and \\ \theta_{ij} < \theta_j \end{cases} then \begin{cases} \mathcal{C}_j \leftarrow \mathcal{C}_{ij} \\ and \\ \theta_j \leftarrow \theta_{ij} \end{cases}$$

$$if \begin{cases} \theta_i > \theta_j \\ and \\ \theta_{ij} < \theta_i \end{cases} then \begin{cases} \mathcal{C}_i \leftarrow \mathcal{C}_{ij} \\ and \\ \theta_i \leftarrow \theta_{ij} \end{cases}$$

This procedure is iterated as long as it permits to reduce the length of the cycle basis. However, the combination process does not guarantee that the new obtained cycle ($\mathcal{C}_{ij}$) is an elementary one i.e. it may be constituted by two disjoint elementary cycles. But, if the resulting cycle is not an elementary one, its length will necessarily be larger than those of the two input cycles. Therefore it will be rejected. Hence it is not use checking the *elementarity* property.

As to the complexity of the whole procedure, we can see that combining two cycles costs $\mathcal{O}(1)$ since cycle lengths are negligible relatively to m. Hence we get an $\mathcal{O}(c^2)$ complexity for IIH.

D. 'Fundamentality' verifying

We based the procedure that verifies whether the cycle basis (CB) obtained after applying IIH is

fundamental or not on the theorem seen above [19] i.e. a CB is fundamental if and only if there exists no cycle of the basis whose arcs belong to other cycles of the basis. In other words, we have to find for each cycle of the CB an exclusive arc i.e. that belongs to no other cycle. This exclusive arc will correspond to a chord. If this is not possible, the CB will be considered non fundamental.

Therefore the *fundamentality* verifying algorithm (FVA) we designed is as follows :

- **Step 1**. Arc extraction : extract from each cycle its exclusive arcs (each exclusive arc may play the role of a chord)
- **Step 2**. Chord choice : choose for each cycle one exclusive arc from the set of its own exclusive arcs.

By this way we'll have a chord for every fundamental cycle belonging to the CB. Hence, the CB will be an FCB if and only if the number of candidate chords is equal to $c = m - n + 1$ i.e. the cyclomatic number of the graph if is connected.

As to the complexity of FVA, choosing a particular data structure for the input graph permits to execute step 1 in $\mathcal{O}(m)$ time in the worst case (see section 3 below). As to step 2, it costs $\mathcal{O}(m^2)$. Therefore the overall complexity is $\mathcal{O}(m^2)$ i.e. $\mathcal{O}(n^4)$ (resp. $\mathcal{O}(n^2)$) for a dense (resp. sparse) graph for which $m = \mathcal{O}(n^2)$ (resp. $m = \mathcal{O}(n)$).

Remark that an alternative FVA (AFVA) involving only one step (by merging steps 1 and 2) may be designed. In this case, its complexity will be $\mathcal{O}(cm)$ i.e. $\mathcal{O}(n^4)$ (resp. $\mathcal{O}(n^2)$) for a dense (resp. sparse) graph for which $m = \mathcal{O}(n^2)$ and $c = \mathcal{O}(n^2)$ (resp. $m = \mathcal{O}(n)$ and $c = \mathcal{O}(n)$). We preferred proceeding in two steps for practical implementation reasons.

E. Phase 3 : Transforming a CB into an FCB

Disposing of both a spanning tree (ST) and its associated FCB on which we applied the IIH leading to a cycle basis that has been proved non fundamental. The following phase consists in transforming the latter into an FCB. The transformation procedure (TP) is depicted below.

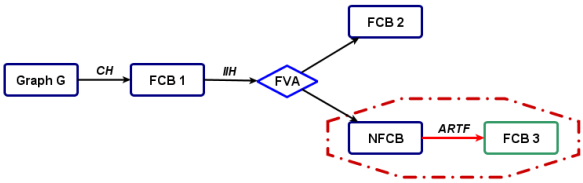

Fig. 5 ARTF Scheme

After applying the previous FVA, the cycles of the basis may be partitioned into two sets where the first (resp. second) involves any cycle that has one (resp. no) exclusive arc. A cycle of the first set will be called fundamental.

In order to illustrate our procedure, let us revisit the example of figure 3 and depict the cycle partitioning (see figures 6 and 7, where straight lines correspond to green coloured arcs, dashed lines to blue coloured arcs and dotted lines to black coloured arcs).

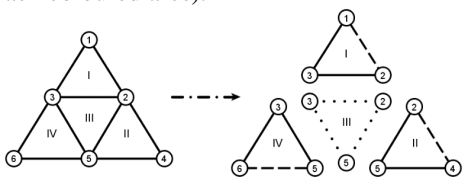

Fig. 6 Non FCB Partitioning

We can remark that the fundamental cycles I, II and IV are each constituted by branch arcs (in green) and an exclusive chord (in blue). On the other hand, any arc of the non fundamental cycle III belongs to another cycle. Hence we decide to keep portions (arcs) that will be used to construct the complete spanning tree. This latter will involve a subset of the branch arcs belonging to the fundamental cycles. This subset will be augmented by introducing other arcs in order to connect its previous ones and hence construct a complete spanning tree.

To be more precise, we proceed as follows.

(a) Given a CB decomposed into fundamental and non fundamental cycles, we first collect the informations retrieved on all different cycles i.e. arc types: either branch, chord or *non fondamental* arc (i.e. arc belonging to a non fundamental cycle). Note that we do not consider as a branch any green coloured arc belonging to both a fundamental cycle and a non fundamental one. The subset of all nfa's will constitute the subgraph to be processed. Figure 7 describes the decomposition procedure.

International Conference on Automation, Control, Engineering and Computer Science (ACECS'14)
Proceedings - Copyright IPCO-2014, pp.131-137
ISSN 2356-5608

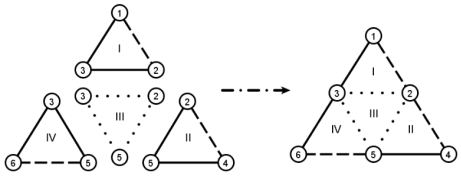

Fig. 7 Arc colouring Procedure (branchs in green, chords in blue, nfa's in black)

(b) We check whether our graph is constituted by different subgraphs among which we can distinguish those who have not yet been processed (i.e. those involving nfa's). A local constructive heuristics is used here. Indeed, in not considering the parts composed of chords or branchs, we obtain a subgraph (that may be non connected) for which we are constructing a spanning tree.

The whole fundamentalisation heuristic, called ARTF, is detailed below.

- Step 1. Decompose the CB into fundamental and non fundamental cycles
- Step 2. Colour all the arcs of the graph in red
- Step 3. Colour the arcs belonging to fundamental cycles : green for branchs, blue for chords
- Step 4. Colour the nfa's in black (regardless of their previous colour)
- Step 5. Constitute the partial spanning tree (PST) involving green arcs
- Step 6. Add to the PST arc bridges (i.e. arcs belonging to no cycle) red coloured. Update the PST connected components
- Step 7. Extract the nodes that will be processed by CH. Sort them according to a determined key corresponding to their relative degrees computed in terms of the not yet processed arcs (black coloured).
- Step 8. Apply the local CH with its two variants.

It is easy to see that the overall complexity is $\mathcal{O}(m+c) = \mathcal{O}(m)$ since $c = \mathcal{O}(m)$. We have to add that once this procedure achieved, will have to construct the new FCB. We recapitulate in the following our three-phase approach complexity:

Table 1 Complexity of the three-phase approach

CH		**IIH**	**FVA**	***Fondamentalisation***	
ST	**Calculate_cycles**			**ARTF**	**Calculate_cycles**
$\mathcal{O}(m)$	$\mathcal{O}(cm)$	$\mathcal{O}(c^2)$	$\mathcal{O}(m^2)$	$\mathcal{O}(m)$	$\mathcal{O}(cm)$

III. Experimental Study

In order to validate our theoretical contribution, we present in this section an experimental study achieved on a series of random, benchmark and real graphs. We precise that we coded our algorithms in C++ under Linux, using STL and *Boost* libraries [18]. The target machine is a Lenovo Thinkpad (i5, 2.40 GHz clock, 4 GB RAM). We have to mention that we used the data structure *Adjacency_list*, for graph representation and two particular others for cycle representation as follows :

- *cycles* : each cycle is represented by a dynamic list of its arcs
- *e_cycles* : each arc of the graph is represented by the list of the cycles it belongs to.

We'll mention in the following the lengths of the CB's obtained by the two approaches (the first using decreasing sort (DEC), the second is Edyhazy's special sort (ELAP)) We'll also give the lengths after each phase (CH, IIH, ARTF).

A. Random graphs

A set of 16 random graphs was generated with Boost generator. This latter permits to generate graphs according to Erdös & Rényi model [11]. In this model, a graph is denoted $G = (n, p)$. Each of the $n(n-1)/2$ possible arcs has a probability p (in [0 1]) to exist. We precise that the generated graph is connected, non oriented and has no loops nor multiple arcs. Four values for n were chosen $(10, 20, 30, 40)$ and 4 for $(0.2, 0.4, 0.6, 0.8)$. The following table depicts for each graph the length of the constructed FCB obtained at each phase of the approach.

Table 2 FCB Lengths for Random Graphs

n	p	m	c	CH (DEC)	IIH	ARTF (DEC)	CH (ELAP)	IIH	ARTF (ELAP)
10	0.2	12	3	9	**9**	-	9	**9**	-
10	0.4	19	10	33	30	**31**	33	30	33
10	0.6	27	18	63	56	63	63	57	**62**
10	0.8	29	20	66	61	65	66	61	**64**
20	0.2	37	18	70	59	65	70	59	**62**
20	0.4	77	58	215	186	202	202	186	**197**
20	0.6	116	97	323	304	**323**	327	304	327
20	0.8	123	104	373	325	371	373	325	**363**
30	0.2	97	68	239	220	**239**	241	219	**239**
30	0.4	149	120	421	377	**413**	430	378	**413**
30	0.6	241	212	787	653	**779**	779	656	**779**
30	0.8	282	253	870	791	870	866	790	**866**
40	0.2	140	101	369	320	**352**	365	321	**352**
40	0.4	302	263	925	830	**904**	925	831	**904**
40	0.6	465	426	1466	1319	1466	1427	1315	**1425**
40	0.8	511	472	1732	1454	1732	1734	1449	**1728**

We remark that the approach using ELAP sorting outperforms the others. Indeed, it is the best 14 times out of 16 i.e. 87.5 % times and is exclusively the best 8 times, whereas the approach based on decreasing sorting (DEC) is the best only 8 times (50%). It shares the first rank with the former 5 times and is exclusively the best 2 times. On the other hand, ARTF-ELAP (resp. ARTF DEC) could improve the initial FCB length 11 times (resp. 9 times).

B. Benchmark graphs

We choosed here square grid graphs which are considered as the most rigorous benchmark graphs for testing FCBM solving heuristics [15]. This is due to their symmetric properties.

We precise that a $G(N,N)$ grid is the cartesian product of two chains, a vertical and a horizontal one, each of which involves N nodes and $N-1$ arcs. Hence, $G(N,N)$ has $n = N^2$ nodes, $m = 2N(N-1) = 2n - 2\sqrt{n}$ arcs and $c = (N-1)^2 = n - 2\sqrt{n} + 1$. On the other hand, we precise that the length of an optimal FCB is known as well as its structure, but no algorithm permitting to construct it is known so far [15]. Indeed, a $G(N,N)$ grid has an optimal FCB of

length $\theta^* = 6n - 20\sqrt{n} + 22$ for $N > 3$. Remark that an obvious optimal non fundamental cycle basis (NFCB) constituted by the $(N-1)^2$ square cycles is of length $\theta_{opt} = 4c = 4(N-1)^2 = 4n - 8\sqrt{n} + 4$.

We used for our experiments 4 values of N i.e. $5, 7, 10, 15$. We precise that NT corresponds to the so called heuristic 'NonTree Egdes' [6].

Table 3 FCB Lengths for Grid Graphs

N	n	m	c	θ					NT
				CH (DEC)	HC (ELAP)	HIA	ARTF (DEC)	ARTF (ELAP)	
5	25	40	16	76	72	64	**72***	**72***	78
7	49	84	36	224	184	144	188	**184**	196
10	100	180	81	716	508	324	572	**492**	518
15	225	420	196	2656	1848	784	2172	**1652**	1588

Table 4 FCB Ratios for Grid Graphs

N	n	m	c	θ^* (FCB)	θ_{opt} (NFCB)	Ratio (%) $\frac{\theta_{(ARTF)} - \theta^*}{\theta^*}$	Ratio (%) $\frac{\theta_{(HC)} - \theta_{(ARTF)}}{\theta_{(HC)}}$
5	25	40	16	72	64	0	0
7	49	84	36	176	144	4.5	0
10	100	180	81	422	324	16.5	3.1
15	225	420	196	1072	784	54.1	10.6

We remark that (i) CH-ELAP is always first ranked, (ii) IIH could reach the lower bound $\boldsymbol{\theta_{opt}}$ of a NFCB for any input CB i.e. the CB obtained by CH(DEC) or CH(ELAP). As to ARTF, ARTF-ELAP is the best as already seen for random graphs. However, the version using decreasing sorting (DEC) is the first one time. We can also notice that the difference as well the ratio between the obtained and optimal lengths (i.e $\boldsymbol{\theta - \theta^*}$ and $\boldsymbol{(\theta - \theta^*)/\theta^*}$ increase with $\boldsymbol{N}$ (hence $\boldsymbol{n}$). However, they are smaller than those corresponding to CH. Let us add that the ratio corresponding to the improvement obtained by applying IIH as well as ARTF increases with $\boldsymbol{N}$ hence $\boldsymbol{n}$.

When comparing our results to those obtained by one among the best known heuristics for grid graphs i.e. NT [6], we notice that ARTF-ELAP outperforms NT 3 times.

C. Real graphs

The third graph set involves 11 real graphs corresponding to water supply networks [3] as follows : (i) 4 graphs of Tunisian cities (Testour, Medjez el beb, Chebba and Tunis) ; (ii) 1 graph of a Danish city ; (iii) 1 graph of the Vietnamese city of Hanoi (Triple Hanoi) ; 2 graphs of South Korean cities (Bak Ryun and GoYang) ; and (iv) 3 graphs of unknown locations (Boss1, Boss2 and Boss3).

Table 5 FCB Lengths for Real Graphs

Graph	n	m	c	CH (DEC)	IIH	ARTF (DEC)	CH (ELAP)	IIH	ARTF (ELAP)
Testour	12	16	5	22	**19**	-	24	**19**	-
Medjez el beb	26	34	9	40	**38**	-	38	**38**	-
Chebba	36	49	14	79	64	71	87	64	**68**
Tunis	86	137	52	352	245	**302**	336	246	303
Denmark	172	200	29	127	**116**	-	131	114	**116**
Triple Hanoi	92	100	9	117	**99**	-	108	**99**	-
BakRyun	36	52	17	102	80	89	98	82	**87**
GoYang	23	31	9	40	**38**	-	41	**38**	-
Boss1	128	147	20	126	**119**	-	127	**119**	-
Boss2	122	149	28	189	164	**170**	178	164	**170**
Boss3	173	249	77	707	403	600	522	394	**513**

The results we obtained confirm the previous ones. Indeed, the approach using ELAP outperforms the one using Decreasing sort since it is first ranked 10 times i.e. 90.9% whereas the other is first ranked 8 times i.e. 72.7%. However, we can remark that for this set of real graphs, IIH kept sometimes the 'fundamentality' (6 times out of 11 for DEC and 5 times for ELAP). This thus justify our choice consisting in first constructing an FCB then improve it instead of starting from a good quality non FCB then 'fundamentalising' it.

IV. CONCLUSION & PERSPECTIVES

Through our contribution presented in this paper, we could design a new approach for constructing an FCB of reduced length, based on first the conjunction of constructive and iterative improving heuristics. Since this may lead in general to a non FCB, we designed an ultimate '*fundamentalising'* phase corresponding to two algorithms, the first being based on a theorem due to Syslo [19] and the second capable to 'fundamentalise' any non FCB. A series of experiments led to some satisfactory practical results. However, we can deduce some interesting points that would be studied in the future. We may cite the following :

- Improve the efficiency of IIH
- Apply directly ARTF on an optimal (non fundamental) cycle basis
- Experiment our approach on other 'tough' benchmark graphs such as triangular graphs (see a sample in Figure 1)

Design efficient parallel algorithms for solving the addressed problem (we already achieved a study on this point but it deserves more deepening and experiments)

REFERENCES

[1] E. Amaldi, L. Liberti, F. Maffioli and N. Maculan, "Algorithms for finding minimum fundamental cycle bases in graphs", Electronic notes in discrete mathematics, vol. 17, pp. 29-33, 2004.

[2] E. Amaldi, L. Liberti, F. Maffioli and N. Maculan, "Mathematical models and a constructive heuristic for finding minimum fundamental cycle bases", Yugoslav journal of operations research, vol. 15 ed. 1, pp. 15-24, 2005.

[3] K. Bezzina, "Une approche pour la décomposition, la transformation et la réduction de la taille de problèmes séparables de flot de coût minimum dans un réseau", Unpublished engineering graduation thesis, Faculté des Sciences de Tunis : Département des Sciences de l'Informatique, Tunis, Tunisia, 2010.

[4] Z. Czech, M. Konopka and B. Majewski, "Parallel algorithms for finding a sub-optimal fundamental cycle set in a graph", Parallel computing, vol. 19 ed. 9, pp. 961-971, 1993.

[5] N. Deo, "Minimum-length fundamental cycle set", IEEE Transactions on circuits and systems CAS, vol. 26, pp. 894-895, 1979.

[6] N. Deo, N. Kumar and J. Parsons, "Minimum-length fundamental-cycle set problem: new heuristics and an empirical investigation", Congressus numerantium, vol. 107, pp. 141-154, 1995.

[7] N. Deo, G. Prabhu and M. S. Krishnamoorthy, "Algorithms for generating fundamental cycles in a graph", ACM Transactions on mathematical software, vol. 8 ed. 1, pp. 26-42, 1982.

[8] C. J. Egyhazy, "An algorithm for generating the minimum length fundamental cycles in a graph", Congressus numerantium, vol. 50, pp. 219-230, 1985.

[9] N. W. Gibbs, "Algorithm 491: Basic cycle generation", ACM Transactions on mathematical software, vol. 18 ed. 5, pp. 275-276, 1975.

[10] M. Gondran and M. Minoux, *Graphes et algorithmes*, Eyrolles, 1995.

[11] D. Gregor and A. Lumsdaine. (2013). Erdös-Renyi generator. [Online]. Available: http://www.boost.org/doc/libs/1_55_0/libs/graph/doc/erdos_renyi_generator.html

[12] M. Huber, "Implementation of algorithms for sparse cycle bases of graphs", Technische Universität München, Tech. Rep. 2003.

[13] T. Ito and M. Kizawa, "The matrix rearrangement procedure for graph-theoretical algorithms and its application to the generation of fundamental cycles", ACM Transactions on mathematical software, vol. 3 ed. 3, pp. 227-231, 1977.

[14] T. Kavitha, C. Liebchen, K. Mehlhorn, D. Michail, R. Rizzi, T. Ueckerdt and K. A. Zweig, "Cycle bases in graphs characterization, algorithms, complexity and applications", Computer science review, vol. 3 ed. 4, pp. 199-243, 2009.

[15] C. Liebchen, G. Wünsch, E. Köhler, A. Reich and R. Rizzi, "Benchmarks for strictly fundamental cycle bases", Lecture notes in computer science, vol. 4525, pp. 365-378, 2007.

[16] Z. Mahjoub, "Contribution à l'étude de l'optimisation des réseaux maillés", Unpublished state doctoral thesis, INP Toulouse, Toulouse, France, 1983.

[17] K. Paton, "An algorithm for finding a fundamental set of cycles of a graph", ACM Transactions on mathematical software, vol. 12 ed. 9, pp. 514-518, 1969.

[18] H. Sutter and A. Alexandrescu. (2013). Boost C++ Libraries. [Online]. Available: http://www.boost.org

[19] M. M. Syslo, "On cycle bases of a graph", Networks, vol. 9 ed. 2, pp. 123-132, 1979.

[20] R. E. Tarjan, "Depth-first search and linear graph algorithms", SIAM Journal on computing, vol. 1, pp. 146-160, 1972.

[21] P. Vismara, "Reconnaissance et représentation d'éléments structuraux pour la description d'objets complexes. Application à l'élaboration de stratégies de synthèse en chimie organique", Unpublished doctoral thesis, Université Montpellier II : Université des Sciences et Techniques du Languedoc, Montpelier, France, 1995.