An Exact Algorithm for Solving the Vertex Separator Problem

DIDI BIHA Mohamed[1], **and MEURS Marie-Jean** [2]

[1] Université d'Avignon et des Pays de Vaucluse,
Laboratoire d'Analyse Non linéaire et Géométrie (EA 2151), F-84018 Avignon, France.
mohamed.didi-biha@univ-avignon.fr
present address : LMNO, Université de Caen,
BP 5186, F14032 Caen Cedex, France
didibiha@math.unicaen.fr

[2] Université d'Avignon et des Pays de Vaucluse,
Laboratoire Informatique d'Avignon (EA 931), F-84911 Avignon, France.
marie-jean.meurs@univ-avignon.fr

Abstract: Given $G = (V, E)$ a connected undirected graph and a positive integer $\beta(|V|)$, the vertex sperator problem is to find a partition of V into noempty three classes A, B, C such that there is no edge between A and B, max$\{|A|, |B|\} \leq \beta(|V|)$ and $|C|$ is minimum. In this paper we consider the vertex separator problem from a polyhedral point of view. We introduce new classes of valid inequalities for the associated polyhedron. Using a natural lower bound for the optimal solution, we present successful computational experiments.

Key words. graph partitioning, vertex separator, polyhedral approach.

1 Introduction

Let $G = (V, E)$ be a connected undirected graph and $\beta(n)$ be a positive integer, where $n = |V|$. The vertex separator problem (VSP for short) is, given G and $\beta(n)$, to find a partition of V into three nonempty classes A, B, C such that

(i) There is no edge between A and B;

(ii) max$\{|A|, |B|\} \leq \beta(n)$;

(ii) $|C|$ is minimum.

The subset C is called a separator. For convenience, a partition $\{A, B, C\}$ of V which satisfies (i) and (ii) will be also called a separator.

This problem, originated in the area of VLSI design, appears in a wide range of applications. In communication networks , a separator is seen as a bottleneck when a graph represents the network. To check the capacity and brittleness of a network, the separator is used to find bounds and brittle nodes. In the field of graph algorithms, the computation of balanced small-sized separators is very useful, especially for divide-and-conquer algorithms. In bioinformatics and computational biology, separators are wanted in grid graphs providing a simplified representation of proteins. Fu et al.[4] explore the protein folding problem in finding the conformation with minimal energy, i.e. a conformation with the maximal number of non-consecutive vertices occupying neighboring positions.

The VSP is NP-hard [6, 2]. The problem is still NP-hard even if G is planar graph [5]. When $\beta(n) = n - k$ for some positive constant k, the problem becomes polynomially solvable [1]. As it was mentionned in [1], the VSP is trivial if $\beta(n) = 1$ and it is polynomially solvable if $\beta(n) \geq n - 1$.

Despite the fact that VSP has been the subject of extensive research, the first polyhedral approach was done only in 2005 by Balas and De Souza [1, 8]. The present paper concerns only the VSP from a polyhedral point of view.

This paper is organized as follows. In the next section, we present the polyhedron associated with the VSP and we introduce new classes of valid inequalities for this polyhedron. We introduce a natural lower bound for the optimal solution. This lower bound can be calculated in a polynomial time. In Section 3 we present experimental results on large instances of VSP. Our computational results compare favorably with those obtained by Balas and De Suza. Some concluding remarks are given in Section 4.

The remainder of this section is devoted to more definitions and notations.

The graphs we consider are finite, undirected and connected. We denote a graph by $G = (V, E)$, where V is the node set and E is the edge set. If e is an edge with end-nodes u and v, then we write $e = (uv)$. If $W \subseteq V$, the set of edges having one end-node in W and the other one in $\overline{W} = V \setminus W$ is called a *cut* and is denoted by $\delta(W)$. The set of edges having both end-nodes in W will be denoted $E(W)$. If W_1, W_2 are disjoint subsets of V, then $[W_1, W_2]$ denotes the set of edges of G which have one node in W_1 and the other one in W_2. For $U \subseteq V$ we denote by $G(U)$ the induced subgraph on U (i.e., $G(U) = (U, E(U))$. If $F \subseteq E$, then $V(F)$ denotes the set of nodes of F and $G(F)$ the subgraph of G induced by F.

A subset D of G is called dominator if for every $v \in V \setminus D$ there exists some node $d \in D$ such that $(vd) \in E$.

2 The vertex separator polyhedron

Let $G = (V, E)$ be an undirected graph and $\beta(n)$ be a positive integer, where $n = |V|$. For every separator $\{A, B, C\}$ we associate an incidence vector $(x, y) \in \mathbb{R}^{2n}$ defined by $x_v = 1$ if $v \in A$ and 0 otherwise, and $y_v = 1$ if $v \in B$ and 0 otherwise. The convex hull of the incidence vectors of all separator is called the vertex separator polyhedron and denoted $P(G, \beta)$, that is,

$$P(G, \beta) = \text{conv}\{(x, y) : (x, y) \,\text{is the incidence vector of a separator}\}$$

The VSP is equivalent to solving the linear programm

$$max\{\sum_{v \in V}(x_v + y_v),\ (x, y) \in P(G, \beta)\}.$$

If (x, y) is the incidence vector of the separator $\{A, B, C\}$, then (x, y) satisfies the following inequalities :

$$\begin{array}{lll}
x_u + y_v \leq 1 & \forall (uv) \in E & (2.1)\\
x_v + y_u \leq 1 & \forall (uv) \in E & (2.2)\\
x_v + y_v \leq 1 & \forall v \in V & (2.3)\\
1 \leq \sum_{v \in V} x_v \leq \beta(n) & & (2.4)\\
1 \leq \sum_{v \in V} y_v \leq \beta(n) & & (2.5)\\
x_v \in \{0, 1\} & \forall v \in V & (2.6)\\
x_v \geq 0, y_v \geq 0 & \forall v \in V & (2.7)
\end{array}$$

Inequalities (2.1) and (2.2) come from the fact that there is no edge between A and B. Inequalities (2.3) come from the fact that $A \cap B = \emptyset$. Inequalities (2.4) and (2.5) come from the fact that $A \neq \emptyset \neq B$ and $\max\{|A|, |B|\} \leq \beta(n)$.

In [1], Balas and De Souza have shown the following.

Proposition 2.1 $P(G, \beta) = conv\{(x, y) \in \mathbb{R}^{2n} : \ (x, y) \,\text{satisfies}\, (2.1) - (2.6)\}$.

Notice that only the vector x is restricted to be integer.

Balas and De Souza have studied the polyhedron $P(G, \beta)$ and given several families of facet defining inequalities. Central to their investigation is the relationship between separators and dominators : In a connected graph, any separator and any connected dominator have at least one vertex in common. More formally, they have proved that If $\{A, B, C\}$ is a separator, then $|C \cap D| \geq 1$

for every connected dominator $D \subset V$.

Our approach is based on the following simple remark.

Remark 2.2 *If $\{A,B,C\}$ is a separator and a and b are two nodes such that $a \in A$ and $b \in B$, then C and any path from a to b have at least one vertex in common.*

Let u and v be two nonadjacent nodes. Denote by α_{uv} the maximum number of node-disjoint paths between u and v. Define

$$\alpha_{min} = \min\{\alpha_{uv} : u\,, v \in V\,, (uv) \notin E\}.$$

It is clear that if $\{A,B,C\}$ is a separator, then $|C| \geq \alpha_{min}$. Thus, α_{min} is a lower bound of the cardinality of any separator. Thus, the following inequality is valid for $P(G,\beta)$:

$$\sum_{v \in V}(x_v + y_v) \leq n - \alpha_{min} \tag{2.8}$$

Proposition 2.3 *Let S be a subset of V and $\alpha^S = Min\{\bar{\alpha}_{uv},\ u, v \in S,\ (uv) \notin E\}$, where $\bar{\alpha}_{uv}$ is the maximum number of node-disjoint paths between u and v in $G(S)$. The following inequality is valid for $P(G,\beta)$.*

$$\sum_{s \in S}(x_s + y_s) \leq |S| - Min\{\alpha^S\ ,\ |S| - \beta(n)\} \tag{2.9}$$

Proof. If $G(S)$ is not connected or $|S| \leq \beta(n)$, then the inequality (2.9) is trivially true. Now suppose that $G(S)$ is connected and $|S| > \beta(n)$. Let (x,y) be an incidence vector for a certain separator $\{A,B,C\}$. If $S \subseteq C$, then the left-side of the inequality (2.11) is equal zero. It is then obvious that the inequality holds. Now suppose that $S \cap (A \cup B) \neq \emptyset$. Since $|S| > \beta(n)$, we have $A \cap S \neq S \neq B \cap S$. If $S \subset A \cup C$ (resp. $S \subset A \cup C$), then $|(A \cup B) \cap S| = |A \cap S|$ (resp. $|(A \cup B) \cap S| = |B \cap S|$). Thus, $|(A \cup B) \cap S| \leq \beta(n)$ and hence,

$$\sum_{s \in S}(x_s + y_s) \leq \beta(n) = |S| - (|S| - \beta(n)).$$

If $A \cap S \neq \emptyset \neq B \cap S$, then there are two nonadjacent nodes s_1 and s_2 such that $s_1 \in A \cap S$ and $s_2 \in B \cap S$. By Remark 2.2, C and any path in $G(S)$ from s_1 to s_2 have at least one vertex in common. Consequently, we have $|C \cap S| \geq \bar{\alpha}_{s_1 s_2} \geq \alpha^S$. Thus, $|(A \cup B) \cap S| \leq |S| - \alpha^S$ and hence,

$$\sum_{s \in S}(x_s + y_s) \leq |S| - \alpha^S.$$

The last two inequalities implie that inequality (2.11) holds. □
The inequality (2.9) is very interesting. First it is a generalization of the inequality (2.9), which as we will see later is the key of our successful computational experiments. Second, the parameter $\beta(n)$ appears in this inequality.
In the following we will give an example to illustrate this inequality.

Let $\hat{G} = (V, E)$ be the graph of Figure 1. Let us consider the instance of VSP with $\hat{G}$ and $\beta(n) = 3$. Let (x, y) be the fractional solution given in paranthesis (the first number (respectively, second number) corresponds to x_v (respectively, y_v)). It is not hard to see that (x, y) is an extreme point of the polytope given by the inequalities (2.1)-(2.5) and (2.7). This polytope is the linear relaxation of $P(G, \beta)$. Let $S = \{v_1, \ldots, v_5\}$. We have $\alpha^S = 2$ and $|S| - \beta(n) = 2$. The inequality (2.9) cuts off the fractional extreme point (x, y). By standard polyhedral techniques we can also proof that inequality (2.9) defines a facet of $P(\hat{G}, 3)$.

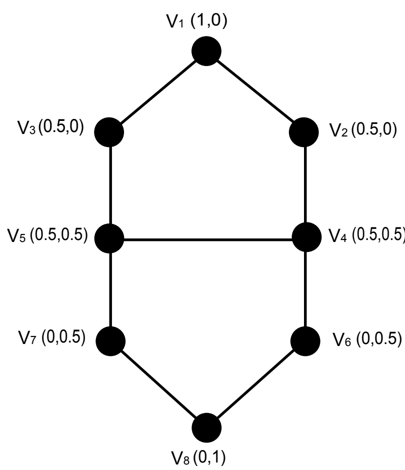

Figure 1: Fractional extreme point

3 Computational experiments

The purpose of our numerical experiments is to mesure the effect of the lower bound α_{min} by comparing our results with those obtained by De Souza and Balas. Thus, the numerical tests have been performed on the same two classes of instances considered by De Souza and Balas [8].

The instances in the first class, called DIMACS graphs, come from the DIMACS challenge on graph coloring. The graphs included have no more than 150 vertices, exception the instance *myciel17* which has 191 vertices. The second class of instances, called MatrixMarket graphs (*MM* graphs for short), is composed by the intersection graphs of the coefficient matrices of systems of linear equation of the form $Ax = b$. The intersection graph of matrix A denoted $G(A) = (V_A, E_A)$, is the graph defined as follows: V_A is in bijection with the columns of A and an edge $(ij) \in E_A$ if and only if there exists an equation in $Ax = b$ such that both variables x_i and x_j have a nonzero coefficient. If the linear system is solved in a divide-and-conquer strategy, it will be divided into two smaller subsystems that are solved separately. However, the complete solution of those subsystems depends on the variables belonging to both systems. The algorithmic cost of merging the solutions of the subsystems to obtain the solution of the original system increases with the number of such variables. The efficiency of the algorithm also requires that the size of the subsystems must be bounded. Thus, the problem of choosing the best way to partition the linear system is reduced to a VSP defined on $G(A)$ (See De Souza and Balas [8]). Three categories of *MM* graphs were generated. The first one, called *MM-I*, corresponds to all matrices with 20 to 100 columns. The second category, called *MM-II*, was obtained from the matrices whose number of columns range from 100 to 200. The third category of instances, denoted by *MM-HD*, only contains graphs of density at least 35%. The 104 instances from the datasets presented above can be downloaded from [9].

Based on the arguments for the development of efficient divide-and-conquer algorithms [7], De Souza and Balas assigned the value $\lfloor \frac{2n}{3} \rfloor$ to $\beta(n)$. They compared five codes, two of them variants of the branch and bound code XPRESS and three of them combining branch and bound with a cutting planes procedure they have developed. All runs were executed on a desktop PC equiped with a Pentium 4 processor, with clock frequency of 2.5 *GHz* and 2 *GB* of memory.
Our tests were performed on Laptop Computer equiped with a Pentium M740 processor, with clock frequency of 1.73 *GHz* and 1 *GB* of memory (Notice that the computer we used is less performant than the one used by De Souza and Balas). They have been conducted in two steps. We first calculated α_{min}, the lower bound of any cardinality of any separator. This achived by calculating p max-flow, where $p = |\{(uv) : u, v \in V,\ (uv) \notin E\}|$. We used the code developed by Cherkassky and Goldberg [3] for the maximum flow problem(Source code available at [10]). We then solved the following mixed-integer program using Ilog-CPLEX 9.0 [11]

$$\text{Max} \sum_{v \in V} (x_v + y_v)$$

subject to

$$\begin{aligned}
x_u + y_v \leq 1 \quad , \quad x_v + y_u \leq 1 & \qquad \forall (uv) \in E \\
x_v + y_v \leq 1 & \qquad \forall v \in V \\
\sum_{v \in V} (x_v + y_v) \leq n - \alpha_{min} & \\
1 \leq \sum_{v \in V} x_v \leq \lfloor \frac{n - \alpha_{min}}{2} \rfloor & \\
1 \leq \sum_{v \in V} y_v \leq \beta(n) \quad , \quad 1 \leq \sum_{v \in V} x_{ia} \leq \beta(n), & \\
x_v \in \{0, 1\} & \qquad \forall v \in V \\
y_v \geq 0 & \qquad \forall v \in V
\end{aligned}$$

The inequality $\sum_{v \in V} x_v \leq \lfloor \frac{n - \alpha_{min}}{2} \rfloor$ comes from the following: let $\{A, B, C\}$ be a separator. We can suppose without loss of generality $|A| \leq |B|$. Since $|A| + |B| \leq n - \alpha_{min}$, we have $|A| \leq \lfloor \frac{n - \alpha_{min}}{2} \rfloor$.

To make a comparison with the results of De Souza and Balas, we focused on the following issues:

- The ability to obtain the optimal solution
- The number of Branch-and-Bond nodes explored
- The running times

The results of our computational experiments with those obtained by De Souza and Balas [8] are presented in Tables 1 to 4, where each line corresponds to one specific instance. The first column contains the instance name followed, in parenthesis, by the number of vertices of the graph and $d = \frac{2|E|}{|V|(|V|-1)}$ its density. The second column contains the value of α_{min}. The third column contains the optimal solution value. The forth column contains the number of Branch-and-Bound nodes explored in our experiments. The fifth column contains the number of Branch-and-Bound nodes explored by De Souza and Balas. The last two columns give respectively our running times and the one of De Souza and Balas. For every instance, we compare our result with the best one among the five codes used by De Souza and Balas. Our time running does not include the calculation of α_{min}. In fact, the time spend on calculating α_{min} is negligible.

In their computational experiments, De Souza and Balas fixed the maximum CPU time to 30 minutes. The instances for wich the time running of De Souza and Balas is indicated with "$\star$" are those whose the five codes developed by De Souza and Balas could not reach an optimal solution, or prove that such solution was reached, within the imposed time limit. In our experiments, we were able to find the optimal solution for all instances in small time. The only exception being instance *DSJC125.1* (Table 1.) for wich the optimal solution was found after 163 minutes. Notice that after 30 minutes, the value of the best integer solution was 91 (the value of optimal solution is 91) while the upper bound is equal 101. For this instance, the value of the best integer solution founded by De Souza and Balas was 89 while the upper bound was 102. In Table 1, we summarize the results for DIMACS instances.

Table 1. DIMACS instances

Instances	α_{min}	*Obj*	$\sharp$ *nodes*	$\sharp$ *nodes(SB)*	*time(s)*	*time(s) SB*
david (87,0.11)	1	81	113	52	0.984	0.19
DSJC125.9 (125,0.90)	103	22	0	34241	0.703	794.36
games120 (20,0.09)	2	102	8088	96920	121.047	429.02
miles500 (128,0.14)	2	119	145	318	7.156	2.11
miles750 (128,0.26)	6	113	938	369	62.797	9.83
miles1000 (128,0.40)	11	110	168	65	19.062	13.62
myciel3 (11,0.36)	3	8	0	19	0.000	0.00
myciel4 (23,0.28)	4	17	37	48	0.109	0.03
myciel5 (47,0.22)	5	37	290	169	0.828	0.28
myciel6 (95,0.17)	6	76	551	458	12.094	5.14
myciel7 (191,0.13)	7	156	14737	2441	880.875	160.59
queen6_6 (36,0.46)	15	21	0	81	0.016	1.42
queen7_7 (49,0.40)	18	31	0	263	0.063	7.78
queen8_8 (64,0.36)	21	43	0	3533	0.031	42.44
queen9_9 (81,0.33)	24	55	959	291471	14.375	1067.5
DSJC125.1 (125,0.09)	5	91	523328	110792	9783.080	(1800) $\star$
DSJC125.5 (125,0.50)	51	74	120	16768	8.610	(1800) $\star$
queen8_12 (96,0.30)	25	65	3175	80695	77.297	(1800) $\star$
queen10_10(100,0.30)	27	67	5412	73956	105.765	(1800) $\star$
queen11_11 (121,0.27)	30	81	13410	22756	456.781	(1800) $\star$
queen12_12 (144,0.25)	33	97	24635	10567	1245.080	(1800) $\star$

We recall that for every instance, to make a comparison with our result, we have chosen the best running time and the minimum number of nodes among the five codes developed by De Souza and Balas. From Table 1, we see that 6 instances could not be solved by De Souza and Balas within the time limite of 30 minutes by any of their codes. In fact, excepting for the instance DSJC125.1 they have obtained the optimal solution but they could not have the proof of their optimality. For example, in their experiments, the value of the best integer solution for the instance DSJC125.5 was 74 while the upper bound was 86.
The Table 1 shows that when the instances are easy, we obtain the same performance with the best code used by De Souza and Balas. On the contrary, for the hard instances our results compare very favorably with those obtained by De Souza and Balas. This conclusion is still true for MM instances as Tables 2 to 4 show.

Table 2. MM-I instances

Instances	α_{min}	*Obj*	♯ *nodes*	♯ *nodes(SB)*	*time(s)*	*time(s) SB*
ash219 (85,0.06)	2	79	137	111	1.187	0.18
dwt72 (72,0.07)	2	68	26	59	0.500	0.07
can62 (62,0.11)	2	56	57	119	0.500	0.15
dwt66 (66,0.12)	4	62	8	31	0.312	0.06
bcspwr02 (49,0.15)	2	44	49	51	0.484	0.06
dwt__59 (59,0.15)	3	51	137	201	0.969	0.30
bcspwr01 (39,0.16)	2	36	13	13	0.125	0.02
ash85 (85,0.17)	6	72	593	971	8.610	2.87
dwt87 (87,0.19)	4	77	109	241	3.031	0.90
impcol_b (59,0.19)	3	49	428	380	1.532	0.59
west0067 (67,0.19)	3	56	195	185	1.578	0.51
will57 (57,0.19)	2	53	55	17	0.531	0.05
can96 (96,0.20)	14	72	7787	259231	154.921	1131.60
steam3 (80,0.23)	8	72	108	313	1.828	0.81
curtis54 (54,0.24)	5	46	77	167	0.625	0.20
can73 (73,0.25)	10	53	1138	6997	8.266	21.82
bfw62a (62,0.34)	3	59	7	11	0.579	0.05
ibm32 (32,0.36)	4	24	69	72	0.313	0.07
pores_1 (30,0.41)	5	22	48	7	0.204	0.09
can61 (61,0.47)	12	46	90	5	2.109	0.82
bcsstk01 (48,0.55)	18	30	0	39	0.063	1.63
can24 (24,0.57)	8	16	0	17	0.000	0.05
fidapm05 (42,0.61)	12	30	44	5	0.422	0.10
fidap005 (27,0.67)	9	18	28	5	0.109	0.04

Table 3. MM-II instances

Instances	α_{min}	*Obj*	♯ *nodes*	♯ *nodes(SB)*	*time(s)*	*time(s) SB*
L125.ash608 (125,0.05)	2	118	142	82	2.375	0.50
L125.will199 (125,0.05)	1	119	135	119	2.562	0.35
L125.west0167 (125,0.06)	1	121	69	73	1.906	0.22
ash331 (104,0.06)	2	97	183	212	2.812	0.82
west0132 (132,0.06)	1	126	105	85	4.297	0.42
rw136 (136,0.07)	1	121	1635	17386	28.875	57.57
bcspwr03 (118,0.08)	2	112	99	73	2.422	0.38
gre__115 (115,0.09)	3	95	5573	6659	82.812	28.77
L125.dw__162 (125,0.12)	6	116	469	214	8.360	1.15
L125.can__187 (125,0.13)	12	111	786	5353	16.984	23.22
L125.gre__185 (125,0.15)	6	104	1300	2337	30.578	31.40
L125.can__161 (125,0.16)	12	97	15063	218627	580.797	(1800) ⋆
L125.lop163 (125,0.16)	9	163	308	6319	16.031	33.22
can__144 (144,0.16)	18	126	1362	64203	31.250	443.49
lund_a (147,0.26)	11	118	385	2705	45.891	155.29
L125.bcsstk05 (125,0.35)	17	101	331	2301	30.812	54.29
L125.dwt__193 (125,0.38)	21	95	259	129	29.719	73.74
L125.fs_183_1 (125,0.44)	1	98	240	1175	35.016	36.21
bcsstk04 (132,0.68)	48	84	600	89	26.796	80.01
arc130 (130,0.93)	21	87	268	129	99.328	137.42

Table 4. MM-HD instances

Instances	α_{min}	*Obj*	$\sharp$ *nodes*	$\sharp$ *nodes(SB)*	*time(s)*	*time(s) SB*
L100.steam2 (100,0.36)	16	76	175	167	8.781	30.98
L100.cavity01 (100,0.37)	8	85	115	7	10.657	3.52
L80.cavity01 (80,0.38)	8	65	125	15	4.578	2.22
L80.fidap025 (80,0.38)	12	68	52	7	2.625	1.14
L120.fidap025 (120,0.39)	18	102	82	21	9.61	9.27
L80.steam2 (80,0.40)	12	61	115	29	4.218	3.15
L100.fidap021 (100,0.41)	15	85	119	5	5.469	2.64
L100.fidap025 (100,0.41)	12	82	310	11	13.578	5.41
L120.cavity01 (120,042)	8	99	138	15	16.375	16.57
L120.fidap021 (120,0.43)	15	98	192	27	16.765	17.36
L80.fidap021 (80,0.43)	11	65	99	7	4.031	2.43
L120.rbs480a (120,0.46)	23	88	209	233	21.282	98.84
L120.wm2 (120,0.47)	1	98	171	33	18.781	19.27
L100.rbs480a (100,0.52)	26	73	156	63	13.953	11.91
L80.wm3 (80,0.55)	1	63	184	13	8.406	3.81
L80.wm1 (80,0.57)	2	59	155	59	8.64	11.13
L80.rbs480a (80,0.58)	18	62	43	5	3.5	0.97
L80.wm2 (80,0.58)	1	61	153	15	7.36	2.89
L100.wm3 (100,0.59)	1	77	179	9	19.656	4.79
L120.e05r0000 (120,0.59)	30	90	82	11	9.828	7.51
L100.wm1 (100,0.60)	1	74	196	27	27.782	11.65
L120.fidap022 (120,0.60)	30	84	219	177	30.782	94.06
L100.wm2 (100,0.61)	1	76	190	15	18.485	7.37
L100.fidapm02 (100,0.62)	27	69	255	13	25.125	6.35
L120.fidap001 (120,0.63)	35	82	237	27	37.532	22.56
L100.e05r0000 (100,0.64)	30	70	120	47	7.937	14.94
L120.fidapm02 (120,0.65)	27	86	223	11	42.953	12.29
L80.fidapm02 (80,0.65)	27	53	50	7	3.281	1.97
L100.fidap001 (100,0.68)	36	64	100	39	5.141	9.82
L100.fidap022 (100,0.68)	38	62	168	143	13.985	30.62
L80.e05r0000 (80,0.68)	19	60	133	3	7.937	0.96
L80.fidap001 (80,0.72)	2	54	84	1	3.14	0.98
L80.fidap022 (80,0.76)	39	41	75	171	2.172	10.05
L80.fidap002 (80,0.77)	27	53	83	9	3.328	1.71
L80.fidap027 (80,0.80)	24	56	81	3	4.469	1.18
L100.fidap027 (100,0.81)	27	69	187	5	23.437	4.15
L100.fidap002 (100,0.82)	34	66	102	3	6.234	2.38
L120.fidap002 (120,0.82)	52	68	121	71	10.453	37.86
L120.fidap027 (120,0.85)	35	83	229	9	50.719	9.07

4 Concluding remarks

In this paper, we have studied the vertex separator problem from a polyhedral point of view. We have introduced some new valid inequalities for the vertex separator polyhedron. Using a natural lower bound we were able to solve to optimality all the instances generated by De Souza and Balas [8] in small time without using any sophisticated methods like a branch-and-cut algorithm. This leads us to believe that larger instances could be routinely solved with a cutting plane algorithm using this lower bound and the inequalities introduced by Balas and de Souza .

An important problem which deserves to be adressed is to study the separation problem of the inequality (2.9) and to use this inequality in the framework of a cutting plane algorithm for the VSP.

References

[1] E. Balas and C. de Souza, *The vertex separator problem : a polyhedral investigation*, Mathematical Programming, n. 3, 103(2005), pp. 583–608.

[2] T.N. Bui and C. Jones, *Finding Good Approximate Vertex and Edge Partitions is NP-Hard*, Information Processing Letters,42(1992), pp. 153–159.

[3] B.V. Cherkassky and A.V. Goldberg, *On Implementing Push-Relabel Method for the Maximum Flow Problem*, Algorithmica, 19(1997), pp. 390-410.

[4] B. Fu, S.A. Oprisan, and L. Xu, *Multi-Directional Width-Bounded Geometric Separator and Protein Folding*, ISAAC, (2005), pp. 995–1006.

[5] J. Fukuyama, *NP-completeness of the planar separator problems*, Journal of Graph Algorithms and Applications, 4(2006), pp. 317–328.

[6] M.R. Garey and D.S. Johnson, *Computers and Intractabiliy*, W.H. Freeman and Compagny, (1979)

[7] R.J. Lipton and R.E. Tarjan, *A separator theorem for planar graphs*, SIAM J. Numer. Anal., 36(1979), pp. 177–189.

[8] C. de Souza and E. Balas, *The vertex separator problem : algorithms and computations*, Mathematical Programming, n. 3, 103(2005), pp. 609–631.

[9] HTTP://www.ic.unicamp.br/~cid/Problem-instances/VSP.html

[10] HTTP://www.avglab.com/andrew/soft.html

[11] HTTP://www.ilog.com