THE MULTI-TERMINAL VERTEX SEPARATOR PROBLEM: POLYHEDRAL ANALYSIS AND BRANCH-AND-CUT

Conference Paper - October 2015

CITATIONS
3

READS
98

4 authors, including:

Denis Cornaz
Paris Dauphine University
9 PUBLICATIONS 55 CITATIONS

Youcef Magnouche
Paris Dauphine University
7 PUBLICATIONS 4 CITATIONS

Sebastien Martin
Huawei Technologies
28 PUBLICATIONS 39 CITATIONS

Some of the authors of this publication are also working on these related projects:

Project
Tire-shape-optimization View project

Project
IEEE-CoDIT’19 conference (Paris-France, April 2019) View project
THE MULTI-TERMIAL VERTEX SEPARATOR PROBLEM: POLYHEDRAL ANALYSIS AND BRANCH-AND-CUT

Denis Cornaz 1 Youcef Magnouche 1 A.Ridha Mahjoub 1 Sébastien Martin 2
1 Laboratoire d’analyse et modélisation de systèmes pour l’aide à la décision,
Université Paris-Dauphine, Paris, France.
{denis.cornaz, youcef.magnouche, ridha.mahjoub}@dauphine.fr
2 Laboratoire de conception, optimisation et modélisation des systèmes,
Université de Lorraine, Metz, France.
sebastien.martin@univ-lorraine.fr

ABSTRACT
Let $G = (V \cup T, E)$ be an undirected graph such that $V$ is a set of vertices, $E$ a set of edges and $T$ a set of terminal vertices. The Multi-terminal vertex separator problem consists in partitioning $V$ into $k + 1$ subsets $\{S, V_1, \ldots, V_k\}$ minimizing the size of $S$ and such that there is no edge between two subsets $V_i$ and $V_j$ and each subset $V_j$ contains exactly one terminal. Set $S$ is called a separator. In this paper, we show that this problem is NP-complete. We discuss the problem from a polyhedral point of view. We describe some valid inequalities and characterize when they define facets. Using this we develop a Branch-and-Cut algorithm.

Keywords: Combinatorial optimization, Polyhedral approach, Branch-and-Cut, Complexity, Vertex separator problem.

1 INTRODUCTION:
Let $G = (V \cup T, E)$ be a simple graph with $V$ a set of vertices, $E$ a set of edges and $T$ a set of terminal vertices. The multi-terminal vertex separator problem, MTVSP for short, consists in finding the smallest subset $S \subseteq V$ called a separator such that the graph induced by $V \setminus S$ contains $k$ disjoint components and each component contains exactly one terminal. This problem is a variant of the vertex separator problem that consists in partitioning $V$ into $k + 1$ subsets $S, V_1, \ldots, V_k$ in such a way that $S$ is minimum and there is no edge between two subsets $V_i$ and $V_j$. The MTVSP is equivalent to finding the smallest node subset $S$ such that each path between two terminals intersects $S$. The MTVSP has applications in different areas like VLSI conception, linear algebra, connectivity problems and parallel algorithms. Many variants of the vertex separator problem have been studied [2] [3] [4]. In [2] Balas and Suza studied the following problem. Given a simple graph $G = (V, E)$ and an integer $\beta(n)$ with $n = |V|$, partition $V$ into three subsets $A, B$ and $C$ such that $|C|$ is minimum, no vertex in $A$ is incident to a vertex in $B$ and $\max(|A|, |B|) \leq \beta(n)$. In [5] authors studied another variant of the problem. Let $G = (V, E)$ be a simple graph and $a, b \in V$ two terminal nodes. The problem here is to partition $V$ into three subsets $A, B$ and $C$ minimizing $|\delta(C)|$ such that no vertex in $A$ is incident to a vertex in $B$ and $a \in A, b \in B$. This problem can be solved in polynomial time. It is equivalent to a minimum cut problem in a transformed graph.

The paper is organized as follows, in Section 2 we discuss the complexity of the MTVSP. In Section 3

* Corresponding Author
we propose two 0–1 linear programming formulations for the problem. In Section 4 we study the problem from polyhedral point of view and propose some valid inequalities. In Section 5 we present a branch-and-cut algorithm along with some experimental results.

We denote by $G = (V \cup T, E)$ a simple graph with $V$ a set of vertices, $T$ a collection of terminal vertices and $E$ a set of edges. We denote by $n$ the size of the set $V$ and $k$ the number of terminals in $T$. Given a vertex $v \in V \cup T$, we denote by $N(v) \subseteq (V \cup T)$ the set of vertices incident to $v$ and by $d(v)$ the size of $N(v)$ called degree of $v$ in $G$. Given a subset $R \subseteq (V \cup T)$, we denote by $N(R) \subseteq (V \cup T)$ the set of vertices incident to at least one vertex in $R$. Let $\delta(v)$ be the set of edges incident to $v$ and $\delta(R)$ the set of edges having exactly one vertex in $R$. Let $C \subseteq \mathbb{Z}'$ be a vector, $C(R)$ is equivalent to $\sum_{v \in R} C(v)$. Let $H = (U, I)$ be a subgraph of $G$. We denote by $H \subseteq V$ the subset $U$ and by $H \subseteq E$ the subset $I$. The *internal vertices* of a path are all vertices of the path except the extremities. A path having its extremities in $T$ is called a *terminal path*. In this paper we consider the following hypotheses:

- There is no edge between two terminals, otherwise the problem has no solution.
- For each pair of terminals $t_i, t_j \in T$, we have $N(t_i) \cap N(t_j) = \emptyset$. Otherwise all vertices of $N(t_i) \cap N(t_j)$ belong to every separator.
- For each vertex $v \in V$, there is at least one path, between two terminals, containing $v$. Otherwise $v$ cannot belong to a minimal separator.
- The graph $G$ is connected.

2 **COMPLEXITY ANALYSIS**

In this section we consider the three-terminal vertex separator problem (TTVSP). It has been shown that the (TTVSP) is NP-complete [10]. In this section we give a simpler of this result using a polynomial reduction from the minimum vertex cover set problem $VC$. The $VC$ problem is a well-known NP-complete problem. It consists of finding the smallest subset of vertices such that all edges have at least one vertex in it.

It is clear that the TTVS problem is in NP. Let $H = (U, E')$ be a simple graph. We construct a graph $G = (V_1 \cup V_2 \cup V_3 \cup T, E)$ from the graph $H$ using the following operations:

- add three vertices $t_1, t_2$ and $t_3$ in $T$.
- for each vertex $u \in U$, add three vertices, $v^u_1$ in $V_1$, $v^u_2$ in $V_2$ and $v^u_3$ in $V_3$
- for each vertex $u \in U$, add three edges $t_1v^u_1$, $t_2v^u_2$ and $t_3v^u_3$ in $E$.
- for each vertex $u \in U$, add two edges $v^u_1v^u_2$ and $v^u_2v^u_3$ in $E$.
- for each edge $uv \in E'$, add two edges $v^u_1v^v_2$ and $v^v_1v^u_2$ in $E$.

![Figure 1: Graph transformation](image)

Figure 1 illustrates the above graph transformation. Let $S \subseteq V_1 \cup V_2 \cup V_3$ be a separator and $v^u_1, v^u_2, v^u_3 \in V$ three vertices associated with each vertex $u \in U$.  

2
**Proposition 1:** For a vertex \( u \in U \), if \( S \) is the smallest separator of \( G \) then either \( v_1^u, v_2^u \in S \) and \( v_3^u \not\in S \) or \( v_1^u, v_2^u \not\in S \) and \( v_3^u \in S \).

**Proof.**

If \( v_1^u \) belongs to the separator \( S \) and not \( v_2^u \) and \( v_3^u \) then there is a path from \( t_3 \) to \( t_2 \). If \( v_1^u \) and \( v_3^u \) belong together to the separator \( S \) but not \( v_2^u \) then we can replace \( v_3^u \) by \( v_2^u \). Clearly, the separator remains the smallest since \( v_3^u \) is only incident to \( v_1^u \), \( v_2^u \) and \( t_3 \). If \( v_1^u \), \( v_2^u \) and \( v_3^u \) belong together to \( S \), then \( S \setminus \{v_3^u\} \) is also a separator, a contradiction with the minimality of \( S \).

**Proposition 2:** The smallest vertex cover set in \( H \) is of size \( q \) if and only if the smallest separator in \( G \) is of size \( q + |U| \).

**Proof.**

\( (\Rightarrow) \) Let \( R \subseteq U \) be the vertex cover set of size \( q \). If for each vertex \( u \in R \), we add its corresponding vertices \( v_1^u \) and \( v_2^u \) in set \( S \), and for each vertex \( u \in U \setminus R \) we add its corresponding vertex \( v_3^u \) in set \( S \), then \( S \) is a separator in \( G \) of size \( 2q - q + |U| = q + |U| \). Moreover, \( S \) is the smallest separator.  

\( (\Leftarrow) \) For a terminal path containing \( v_1^u \in V_1 \) and \( v_2^u \in V_2 \), either \( v_1^u \in S \) or \( v_2^u \in S \). We know that there is an edge between \( v_1^u \) and \( v_2^u \) if there exists an edge \( uv \in E' \). It then follows that the corresponding vertices of \( S \cap V_3 \) represent a vertex cover set in \( H \) of cardinality \( q \). If \( v_1^u \not\in S \) or \( v_2^u \not\in S \), then \( v_3^u \in V_3 \) belongs to \( S \). So the separator is of size \( 2q - q + |U| = q + |U| \), and the vertex cover in \( H \) is of size \( q \). It is clear that if \( S \) is the smallest in \( G \), then the vertex cover is in \( H \).

3 FORMULATIONS

In this section we propose two different \( 0 - 1 \) linear formulations for the problem, the first one has a polynomial number of variables and constraints and uses double indices. The second has a polynomial number of variables but an exponential number of constraints.

### 3.1 Double indices formulation

Let \( x \in \{0,1\}^{(V \cup T) \times T} \) such that:

\[ x_{vt} = \begin{cases} 1, & \text{if the vertex } v \text{ belongs to the subset } V_t, \text{ for every } v \in (V \cup T), t \in T \\ 0, & \text{otherwise.} \end{cases} \]

\[
\max \sum_{v \in V} \sum_{t \in T} x_{vt} \\
\quad x_{ut} + \sum_{k \in T(t)} x_{vk} \leq 1 \quad \forall (uv) \in E, \forall t \in T, \tag{1} \\
\quad \sum_{t \in T} x_{vt} \leq 1 \quad \forall v \in (V \cup T), \tag{2} \\
\quad x_{tt} = 1 \quad \forall t \in T, \tag{3} \\
\quad x_{vt} \in \{0,1\} \quad \forall t \in T, \forall v \in (V \cup T) \tag{4}
\]

### 3.2 Natural formulation

Let \( T \) be the set of terminal paths between each pair of terminals. Let \( x \in \{0,1\}^V \) such that:

\[ x_v = \begin{cases} 1, & \text{if the vertex } v \text{ belongs to the separator,} \\ 0, & \text{otherwise.} \end{cases} \text{ for every } v \in V \]

3
We notice that the first formulation has \((n + k)k\) variables and the second has only \(n\) variables. Inequalities \((5)\) in this latter formulation, which are in an exponential number can be separated in polynomial time. Since the second formulation has less variables, we consider it for our analysis.

### 4 POLYHEDRAL ANALYSIS

For \(S \subseteq V\), let \(x^S \in \{0,1\}^V\) be the vector given by \(x_v = 1\) if \(v \in S\) and \(x_v = 0\) otherwise. \(x^S\) is called the incidence vector of \(S\). Let \(P(G,T)\) be the convex hull of solutions of the above program, that is, \(P(G,T) = \text{conv}(x \in \{0,1\}^V \mid x\) satisfies \((5)\)).

#### 4.1 Dimension and valid inequality

We have the following results:

**Proposition 3:** Polytope \(P(G,T)\) is full dimensional.

**Proposition 4:** For \(v \in V\), inequality \((6)\) defines facet of \(P(G,T)\).

**Proposition 5:** For a vertex \(v \in V\), inequality \((7)\) defines a facet of \(P(G,T)\) if and only if, vertex \(v\) does not belong to a terminal path containing two internal vertices.

#### 4.2 Path inequalities

**Theorem 1:** Inequality \((5)\) associated with a path \(P_{t_i t_j}\) defines a facet of \(P(G,T)\) if and only if:

- \(P_{t_i t_j}\) is inclusionswise minimal, that is only two internal vertices from \(P_{t_i t_j}\) connected to a terminal.
- No vertex \(v \not\in P_{t_i t_j}\) is incident to a terminal \(t \not\in P_{t_i t_j}\) and to two vertices of \(P_{t_i t_j}\).
- No vertex from \(P_{t_i t_j}\) is incident to more than two vertices from \(P_{t_i t_j}\).

**Proof.**\((\Leftarrow)\)

- Suppose there exists a nonterminal vertex \(v \in P_{t_i t_j}\), adjacent to \(t_k \in T\{t_i, t_j\}\). It is clear that there exists a terminal path between \(t_k\) and \(t_j\). \(P_{t_k t_j}\) and inequality \((5)\) associated with \(P_{t_k t_j}\) can then be obtained from inequality \((5)\) associated with \(P_{t_k t_j}\) and inequalities \((7)\) associated with each nonterminal vertex of \(P_{t_k t_j}\).
- Suppose there exists \(v \in V\{P_{t_i t_j}\) and \(v\) adjacent to \(t_k \in T\{t_i, t_j\}\) and to two vertices of \(P_{t_i t_j}\). The following inequality is valid for \(P(G,T)\): \(x(P_{t_i t_j}) + x_v \geq 2\). This is obtained by chvátal-gomory procedure on inequalities \((5)\) induced by the paths \(P_{t_i t_j}, P_{t_k t_j}\) and \(P_{t_k t_j}\). Inequality \((5)\) associated with the path \(P_{t_i t_j}\) can be obtained from the above inequality and inequality \((6)\) of the vertex \(v\).
- If there exists a vertex \(v \in P_{t_i t_j}\) incident to more than two vertices of \(P_{t_i t_j}\), this means that there exists a terminal path \(P'_{t_i t_j}\) such that \(P'_{t_i t_j} \subseteq P_{t_i t_j}\). Inequality \((5)\) associated with \(P_{t_i t_j}\) can then be obtained from inequality \((5)\) of \(P'_{t_i t_j}\) and inequalities \((7)\) associated with vertices of \(P_{t_i t_j}\).\(P'_{t_i t_j}\).
(⇒) Denote by $ax \geq \alpha$ inequality (5). Let $bx \geq \beta$ be an inequality that defines a facet of $P(G, T)$. We suppose that $\{x \in P(G, T): ax = \alpha\} \subseteq \{x \in P(G, T): bx = \beta\}$. Since $P(G, T)$ is full dimensional, we need to prove that there exists $\rho$ such that $b = \rho a$.

For each vertex $v \in P_{t_i, t_j}$, define a separator $S^v = (V \setminus P_{t_i, t_j}) \cup \{v\}$. For each pair of vertices $u, v \in P_{t_i, t_j}$, the incidence vectors $x^{S^u}$ and $x^{S^v}$ are solutions of $P(G, t)$ and satisfy inequality (5) with equality. Hence, $ax^{S^u} = ax^{S^v}$, and hence $bx^{S^u} = bx^{S^v}$. Therefore:

$$b(u) = b(v) = \rho \quad \text{for all } u, v \in P_{t_i, t_j} \text{ and some scalar } \rho \in \mathbb{R}$$

For each vertex $v \in P_{t_i, t_j}$, let $u \in P_{t_i, t_j}$ be a vertex adjacent to $v$. If there is no $u \in P_{t_i, t_j}$ adjacent to $v$, then $u$ would represent any vertex of $P_{t_i, t_j}$. Consider the separator $S^u = S^u \setminus \{v\}$. The incidence vectors $x^{S^u}$ and $x^{S^u}$ are solutions of $P(G, t)$ and satisfy inequality (5) with equality. Hence, $ax^{S^u} = ax^{S^u}$ and hence $bx^{S^u} = bx^{S^u}$. This implies that:

$$b(v) = 0 \quad \forall v \notin P_{t_i, t_j}$$

\[\square\]

![Figure 2: examples of a star tree, clique star, terminal cycle and terminal tree](image)

### 4.3 Star tree inequalities

A **star tree** $J$ is a tree given by a root vertex $v_r \in V$ and $f$ vertex disjoint paths between $v_r$ and $f$ terminal vertices. Let $P_{t_i}$ be the set of internal vertices of a path between $v_r$ and a terminal vertex $t_i \in T$. Let $F_j \subseteq T$ be the leaf set of $J$. The following inequalities, are valid for $P(G, T)$:

$$\sum_{i \in F} x(P_t) + (f - 1)x_{v_r} \geq f - 1 \quad (9)$$

**Theorem 2:** Inequality (9) defines a facet of $P(G, T)$ if and only if the following hold:

- $f \geq 3$.
- No vertex $u \in J$ is incident to a terminal $t \in T \setminus F_j$.
- If two vertices $u, v \in J$ are not adjacent in the sub-graph induced by $J$, then $uv \notin E$.

**Proof.** (⇒)

- If $f = 2$, then the star tree inequality is equivalent to a path inequality associated with $J$.
- If a vertex $v \in P_t$ is incident to a terminal $t' \in T \setminus F_j$, then let the star tree $J'$ with all leaves in $F_j \setminus \{t\}$. Inequality (9) can be obtained from the star tree inequality associated with $J'$, the path inequality associated with $P'$ and the trivial inequalities. If $v_r$ is incident to a terminal $t' \in T \setminus F_j$, then let the a star tree $J'$ with all leaves in $F_j \cup \{t'\}$. Inequality (9) can be obtained from the star tree inequality associated with $J'$ and the trivial inequalities.
- Suppose there exist two vertices $u, v \in J$ not adjacent in the sub-graph induced by $J$ and $uv \in E$. a- $u \in P_t$ and $v \in P_{t'}$, for $t, t' \in F_j$ and $t \neq t'$. Then the following inequality:

###
\[
\sum_{t \in F_j} x (P_t) + (f - 2) x_{v_r} \geq f - 1
\]

is valid for \( P(G,T) \) and dominates inequality (9). Then, inequality (9) cannot be facet defining.

For \( u, v \in P_t \), for \( t \in F_j \). Let \( P_{uv} \subset P_t \setminus \{u, v\} \) be the internal vertices of the path between \( u \) and \( v \). Then the following inequality:

\[
\sum_{t \in F_j} x (P_t) + (f - 1) x_{v_r} - x(P_{uv}) \geq f - 1
\]

is valid for \( P(G,T) \) and dominates inequality (9). Then, inequality (9) cannot be facet defining.

\((\Rightarrow)\) Let us inequality (9) denote by \( ax \geq \alpha \). Let \( bx \geq \beta \) be an inequality that defines a facet of \( P(G,T) \) such that \( \{x \in P(G,T): ax = \alpha\} \subset \{x \in P(G,T): bx = \beta\} \). Since \( P(G,T) \) is full dimensional, we need to prove that there exists \( \rho \) such that \( b = \rho a \). For a terminal \( t \in F_j \), let a ring \( Q^t_i \subset (\bigcup (P_t \cup \{v_r\})) \) be a subset of \( f - 1 \) vertices containing exactly one vertex of each \( P_t \) for all \( l \in (F_j \setminus \{t\}) \), i.e., for all \( l \in (F_j \setminus \{t\}), |P_l \cap Q^t_i| = 1 \). Consider two vertices \( u_1, u_2 \in J \setminus (F_j \cup \{v_r\}) \). There exists two terminals \( t, t' \in F_j \) and two associated rings \( Q^t_i, Q^{t'}_i \) such that \( u_1 \in Q^t_i, u_2 \in Q^{t'}_i \) and \( Q^t_i \setminus \{u_1\} = Q^{t'}_i \setminus \{u_2\} \). Let \( S^{Q^t_i} = (V \setminus J) \cup Q^t_i \) (resp. \( S^{Q^{t'}_i} = (V \setminus J) \cup Q^{t'}_i \)). Clearly \( S^{Q^t_i} \) and \( S^{Q^{t'}_i} \) are two separators.

The incidence vectors \( x^{S^{Q^t_i}} \) and \( x^{S^{Q^{t'}_i}} \) satisfy inequality (9) with equality. Hence, \( ax^{S^{Q^t_i}} = ax^{S^{Q^{t'}_i}} \). Therefore \( bx^{S^{Q^t_i}} = bx^{S^{Q^{t'}_i}} \) implying that \( b(u_1) = b(u_2) \). As \( u_1 \) and \( u_2 \) are arbitrary, we deduce that:

\[
b(u) = b(v) = \rho \quad \forall u, v \in \bigcup (F_j \cup \{v_r\}) \text{ and a scalar } \rho \in \mathbb{R}
\]

Set \( S^0 = (V \setminus J) \cup \{v_r\} \). Clearly, \( S^0 \) is a separator, and its incidence vector satisfy inequality (9) with equality. Hence, \( ax^{S^0} = ax^{S^{Q^t_i}} \) and therefore, \( bx^{S^0} = bx^{S^{Q^t_i}} \). This yields:

\[
b(v_r) = \sum_{v \in Q^t_i} b(v) = (f - 1) \rho
\]

For a vertex \( w \in J \), set \( S_w = S^{Q^t_i} \setminus \{w\} \). Clearly, \( S_w \) is a separator of \( G \) and its incidence vector satisfies inequality (9) with equality. Hence \( ax^{S_w} = ax^{S^{Q^t_i}} \). Therefore \( bx^{S_w} = bx^{S^{Q^t_i}} \). This implies that:

\[
b(w) = 0 \quad \forall w \notin J
\]

From the above equalities, it follows that \( b = \rho a \).

4.4 Clique star inequalities

A clique star is a graph defined by a clique induced by \( K_f \subset V \) and \( f \) vertex disjoint paths from each vertex of \( K_f \) to a terminal \( t \in T \). Let \( P^i_j \) be the internal vertices of the path from \( v_i \) to a terminal vertex \( t_i \in T \). The following inequality is valid for \( P(G,T) \):

\[
\sum_{v \in K_f} x (P^i_j) + x(K_f) \geq f - 1
\]

4.5 Terminal cycle inequalities

Let \( C \subseteq V \) be a cycle and \( Q \subseteq C \) be a vertex subset of \( C \) of size \( f \). A terminal cycle \( L \) is a graph given by \( C \) and \( f \) vertex disjoint paths between vertices of \( Q \) to \( f \) terminals of \( T \). The following inequality is valid for \( P(G,T) \):

\[
\sum_{v \in K_f} x (P^i_j) + x(K_f) \geq f - 1
\]
\[ x(C) \geq \left\lceil \frac{f}{2} \right\rceil \]

**Theorem 3:** A terminal cycle inequality defines a facet of \( P(G,T) \) if and only if the following hold:

- \( f \) is odd.
- If two vertices \( u, v \in Q \) are not incident in \( L \) then \((uv) \not\in E\).
- If there is a vertex \( w \) not in \( L \), adjacent to all vertices of \( Q' \subseteq Q \), then there exists a vertex cover in \( C \) of size \( \left\lceil \frac{|f|}{2} \right\rceil \) that contains at least \(|Q'| - 1 \) vertices of \( Q' \).

### 4.6 Terminal tree inequalities

A terminal tree \( R \) is a tree induced by \( R \subseteq V \) in \( G \) such that all the leaf vertices are terminals. For a vertex \( v \in R \), let \( d_R(v) \) be the number of edges that are incident to \( v \) in \( R \). Let \( f_R \) be the number of terminals of \( R \). For a terminal tree the following inequality is valid for \( P(G,T) \):

\[
\sum_{v \in R} (d_R(v) - 1)x(v) \geq f_R - 1
\]

### 5 BRANCH-AND-CUT ALGORITHM

We developed a branch-and-cut algorithm to solve the multi-terminal vertex separator problem. The path inequalities are separated in polynomial time by an exact algorithm. However we used some heuristic algorithms to separate the valid inequalities cited before. We compare the CPU time of the branch-and-cut with the one of the commercial solver Cplex using the double indices formulation. We use DIMACS graphs coloring instances \([9]\) by adding some terminal vertices. We use also some random graphs. In the following table \( n, m \) and \( k \) represent respectively the number of vertices, edges and terminals. \( Ps, St, Cs, Tt \) and \( Tc \) represent respectively the number of path inequalities, star tree inequalities, clique star inequalities, terminal tree inequalities and terminal cycle inequalities separated in the branch-and-cut algorithm. Nodes and Gap represent respectively the number of branching nodes and the gap given by the branch-and-cut algorithm. Finally, B&C and Cplex represent respectively the CPU Time of the branch-and-cut algorithm and Cplex in seconds.

<table>
<thead>
<tr>
<th>instances</th>
<th>( n )</th>
<th>( m )</th>
<th>( k )</th>
<th>( Ps )</th>
<th>( St )</th>
<th>( Cs )</th>
<th>( Tt )</th>
<th>( Tc )</th>
<th>Nodes</th>
<th>Gap</th>
<th>B&amp;C</th>
<th>Cplex</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSJR500</td>
<td>500</td>
<td>99258</td>
<td>10</td>
<td>94</td>
<td>934</td>
<td>903</td>
<td>1</td>
<td>255</td>
<td>1</td>
<td>0</td>
<td>116.18</td>
<td>37.859</td>
</tr>
<tr>
<td>Anna</td>
<td>138</td>
<td>493</td>
<td>11</td>
<td>210</td>
<td>155</td>
<td>12</td>
<td>31</td>
<td>0</td>
<td>58</td>
<td>0</td>
<td>0.784</td>
<td>0.972</td>
</tr>
<tr>
<td>DSJC125</td>
<td>125</td>
<td>1472</td>
<td>11</td>
<td>61</td>
<td>665</td>
<td>158</td>
<td>136</td>
<td>103</td>
<td>7</td>
<td>2</td>
<td>13.245</td>
<td>1.272</td>
</tr>
<tr>
<td>games120</td>
<td>120</td>
<td>638</td>
<td>10</td>
<td>106</td>
<td>1421</td>
<td>136</td>
<td>6</td>
<td>40</td>
<td>31</td>
<td>11%</td>
<td>14.701</td>
<td>0.933</td>
</tr>
<tr>
<td>David</td>
<td>87</td>
<td>406</td>
<td>10</td>
<td>137</td>
<td>1220</td>
<td>54</td>
<td>42</td>
<td>1</td>
<td>137</td>
<td>31%</td>
<td>9.641</td>
<td>1.223</td>
</tr>
<tr>
<td>Jean</td>
<td>80</td>
<td>254</td>
<td>6</td>
<td>34</td>
<td>11</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>4%</td>
<td>0.143</td>
<td>0.172</td>
</tr>
<tr>
<td>Huck</td>
<td>74</td>
<td>391</td>
<td>9</td>
<td>155</td>
<td>81</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>36</td>
<td>13%</td>
<td>0.584</td>
<td>0.31</td>
</tr>
<tr>
<td>miles250</td>
<td>128</td>
<td>387</td>
<td>15</td>
<td>474</td>
<td>617</td>
<td>40</td>
<td>0</td>
<td>6</td>
<td>426</td>
<td>28%</td>
<td>6.722</td>
<td>1.195</td>
</tr>
<tr>
<td>myciel5</td>
<td>47</td>
<td>236</td>
<td>7</td>
<td>39</td>
<td>23</td>
<td>0</td>
<td>3</td>
<td>6</td>
<td>1</td>
<td>0%</td>
<td>0.376</td>
<td>0.12</td>
</tr>
<tr>
<td>myciel6</td>
<td>95</td>
<td>755</td>
<td>11</td>
<td>176</td>
<td>1354</td>
<td>0</td>
<td>3</td>
<td>14</td>
<td>17</td>
<td>7%</td>
<td>13.413</td>
<td>5.188</td>
</tr>
<tr>
<td>myciel7</td>
<td>192</td>
<td>2360</td>
<td>11</td>
<td>177</td>
<td>344</td>
<td>0</td>
<td>0</td>
<td>77</td>
<td>1</td>
<td>0%</td>
<td>4.609</td>
<td>11.013</td>
</tr>
<tr>
<td>myciel7</td>
<td>192</td>
<td>2360</td>
<td>11</td>
<td>177</td>
<td>344</td>
<td>0</td>
<td>0</td>
<td>77</td>
<td>1</td>
<td>0%</td>
<td>38.996</td>
<td>23.767</td>
</tr>
<tr>
<td>Queen8_8</td>
<td>64</td>
<td>728</td>
<td>8</td>
<td>79</td>
<td>83</td>
<td>8</td>
<td>81</td>
<td>15</td>
<td>1</td>
<td>0%</td>
<td>0.382</td>
<td>0.398</td>
</tr>
<tr>
<td>Queen8_12</td>
<td>96</td>
<td>1368</td>
<td>11</td>
<td>56</td>
<td>199</td>
<td>73</td>
<td>171</td>
<td>19</td>
<td>1</td>
<td>0%</td>
<td>2.136</td>
<td>1.895</td>
</tr>
</tbody>
</table>
The bold numbers represent the cases where the branch-and-cut is better than Cplex. We can see that when the number of vertices and edges is small, both of Cplex and branch-and-cut are efficient. For huge random graphs, when the density and the number of terminals are high, Cplex cannot solve instances. We can notice that Cplex is not efficient with high density graphs.

6 REFERENCES


