Exact and Approximation Algorithms for Linear Arrangement Problems

Alain Quilliot
LIMOS CNRS UMR 6158
LABEX IMOBS3
Université Blaise Pascal
Bat ISIMA, BP 10125
Campus des Cézeaux,
63173 Aubière, France
Email: alain.quilliot@isima.fr

Djamal Rebaine
UQAC
Département d’Informatique
Chicoutimi, Saguenay, Quebec
Canada
Email: Djamal.Rebaine@uqac.ca

Abstract—We present here new results and algorithms for the Linear Arrangement Problem (LAP). We first propose a new lower bound, which links LAP with the Max Cut Problem, and derive a LIP model as well as a branch/bound algorithm for the general case. Then we focus on the case of interval graphs: we first show that our lower bound is tight for unit interval graphs, and derive an efficient polynomial time approximation algorithm for general interval graphs.

I. INTRODUCTION

Let $G = (X, E)$ be a non oriented graph where $X$ and $E$ respectively denote the vertices and edges of $G$. The Linear Arrangement Problem (LAP) consists of finding a one-to-one mapping $\phi$ from $X$ to $\{1, \ldots, |X|\}$ that minimizes:

$$ f(G, \phi) = \sum_{(x, y) \in E} |\phi(y) - \phi(x)| $$

The LAP problem has applications (see [3, 11]) in Information Retrieval and Industrial Storage, and may also appear as a sub-problem of some Network Design models (see [4]). It is, even in practice, a very difficult combinatorial optimization problem. The corresponding decision LAP was first shown to be NP-complete for arbitrary graphs (see for example [8, 9]) and next, for interval graphs [5] and bipartite graphs [9]. However, polynomial time algorithms were designed for trees [4], unit interval graphs [6], paths, cycles, complete bipartite graphs, grid graphs [11] and restricted series-parallel graphs [1]. A survey is available in [3].

Since LAP is NP-hard, even for graphs which usually turn most difficult problems into time-polynomial ones, the heuristic approach is therefore justified to deal with it. So the goal of this theory oriented study is to provide tools for the design of exact and approximation LAP algorithms with some focus on interval graphs. The paper is organized as follows. Section 2 introduces an linear ordering based reformulation of LAP. In Section 3, we propose a general lower bound, which links LAP with the well-known Max Cut Problem, and next derive, in Section 4 and 5, an ILP model together with a branch/bound algorithm. Finally, in section 6 we restrict our study to the case of interval graphs and propose an approximation algorithm, whose efficiency is briefly tested in Section 7.

II. NOTATIONS, DEFINITIONS, LAP REFORMULATION

A simple (non oriented) graph with no loop is denoted by $G = (X, E)$: $X (E)$ is the node (edge) set of $G$. We denote by $(x, y) = (y, x)$ an edge with end-nodes $x$ and $y$ in $X$. If $A \subseteq X$, then $G_A$ is the sub-graph induced by $A$ from $G$. If $x \in X$, then $\Gamma(x) = \{y \in X \text{ such that } (x, y) \in E\}$ is the neighbour set of $x$. The complementary graph $G^c = (X, E^c)$ of $G$ is defined by: $E^c = \{(x, y) \text{ such that } (x, y) /\in E \text{ and } x \neq y\}$. A triangle of $G$ is a clique with 3 nodes. An anti-edge is a pair $e = (x, y) = (y, x), x \neq y$, such that $(x, y) /\in E$. A fork with root $x$ is any (non oriented) triple $f = \{(x, y, z) = (x, z, y) \text{ such that } (x, y), (x, z) \in E, \text{ and } (y, z) /\in E\}$. An anti-fork with root $z$ is any triple $f' = \{(x, y, z) = (y, x, z) \text{ such that } (x, y), (y, z) \in E \text{ and } (x, z) /\in E\}$.

A triangle $\{(x, y, z) = (y, z, x) \text{ such that } (x, y), (x, z) \in E \text{ and } (y, z) /\in E\}$.

A fork with root $x$ is any (non oriented) triple $f = \{(x, y, z) = (x, z, y) \text{ such that } (x, y), (x, z) \in E, \text{ and } (y, z) /\in E\}$. An anti-fork with root $z$ is any triple $f' = \{(x, y, z) = (y, x, z) \text{ such that } (x, y), (y, z) \in E \text{ and } (x, z) /\in E\}$.

Figure 1: Triangles, Forks and Anti-Forks

LAP Reformulation: A linear ordering of a set $X$ is a binary order relation $\sigma$ such that, for any pair $x, y$ in $X$, $x \neq y$, we have either $x \sigma y$ or $y \sigma x$. Given a graph $G = (X, E)$ and a linear ordering $\sigma$ of $X$. For any edge $e = (x, y)$, we set $BE(e, \sigma) = (elementary break of e by $z$ according to $\sigma$) = 1 if $x \sigma z \sigma y \text{ or } y \sigma z \sigma x$, and 0 otherwise. We set $BG(G, \sigma) = (Global Break of G according to $\sigma$) = $\Sigma_{e,z} BE(e, z, \sigma)$. If $\phi(\sigma)$ is the one-to-one mapping from $X$ into $\{1, \ldots, |X|\}$ which derives from $\sigma$, then: $f(G, \phi(\sigma)) = \Sigma_{(x,y) \in E} |\phi(y) - \phi(x)| = BG(G, \sigma) + |E| = \Sigma_{e,z} BE(e, z, \sigma) + |E|$. 

III. LINKING MAX CUT AND LAP: A GENERAL LOWER BOUND

Computing a good linear ordering $\sigma$ of the vertices of a graph $G = (X, E)$ means efficiently deciding, for any $x \in X$, which vertices of the neighbour set $\Gamma(x)$ are located before $x$ according to $\sigma$. This local decision process may be linked to the well-known Max Cut Problem [2,7,10]:

Max Cut Problem: Let $H = (Z, F)$ be a simple graph. We denote by $Z = A \cup^{\text{Ex}} B$ any partition of $Z$ into 2 disjoint subsets, and $\text{Cut}(A, B)$ the number of edges of $H$ with one extremity into $A$ and the other into $B$. Solving the Max Cut Problem means seeking a partition $Z = A \cup^{\text{Ex}} B$ that maximizes $\text{Cut}(A, B)$. We denote by Max-Cut(H) the related optimal value.

Let us consider now some graph $G = (X, E)$. We denote by $Tr(G)$ the number of triangles of $G$. For any vertex $x$ in $X$, we have:

- $\text{m}(x) = \text{number of edges of } H(x); V(x) = \text{m}(x) - \text{Max-Cut}(H(x))$.

Theorem 1: For any graph $G = (X, E)$, we have: $\text{LAP}(G) \geq Tr(G) + \sum_x V(x)$.

Proof: Let us consider a linear ordering $\sigma$ of $G$, and set:

- $\text{Fk}(G, \sigma) = \text{number of forks } f = \{x, y, z\}, f$ with root $x$, of $G$, such that $((x \sigma y) \land (x \sigma z)) \lor ((y \sigma x \land z \sigma x))$.
- $\text{AFk}(G, \sigma) = \text{number of anti-forks } f = \{x, y, z\}$ of $G$, $f$ with root $z$, such that $(x \sigma z \sigma y) \lor (y \sigma z \sigma x)$.

Let us first check that: $BG(G, \sigma) = Tr(G) + \text{Fk}(G, \sigma) + \text{AFk}(G, \sigma)$.

(E1)

In order to do so, we consider an edge $e = (x, y)$, and a node $z$, different from $x$ and $y$. While counting $\text{BE}(e, z, \sigma)$, we consider three cases:

Case 1: $x, y$ and $z$ define a triangle. Then $\text{BE}(e, z, \sigma) = 1$ if either $x \sigma z \sigma y$ or $y \sigma z \sigma x$. In such a case no quantity $\text{BE}((x, z), y, \sigma), \text{BE}((z, y), x, \sigma)$ is equal to $1$. So, if $x, y, z$ define a triangle, there exists exactly one node $t$ in $\{x, y, z\}$ such that $\text{BE}(e(t), t, \sigma) = 1$, where $e(t)$ is the edge which is defined by $\{x, y, z\} - t$.

We get $\Sigma e = (x, y), z \text{ such that } (x, y, z) \text{ is a triangle } \text{BE}(e, z, \sigma) = Tr(G)$.

Case 2: $f = \{x, y, z\}$ is a fork with root $x$. Then $\text{BE}(e(x), y, \sigma) = 1$ if either $x \sigma z \sigma y$ or $y \sigma z \sigma x$, and then $\text{BE}((x, z), y, \sigma) = 0$. Conversely, $\text{BE}((x, y), z, \sigma) = 1$ if either $x \sigma y \sigma z$ or $z \sigma y \sigma x$, and then $\text{BE}((x, y), z, \sigma) = 0$. So, $(x, y, z)$ yields an elementary break iff $y$ and $z$ are located on the same side of $x$ according to $\sigma$. Then:

$$\Sigma e = (x, y), z \text{ adjacent to exactly 1 extremity of } e \text{ BE}(e, z, \sigma) = \Sigma x \Sigma y,z \in \Gamma(x), (y, x) \notin E, y, z \text{ located the same way with respect to } x, \sigma 1 \leq \text{Fk}(G, \sigma).$$

Case 3: $f = \{x, y, z\}$ is an Anti-Fork with root $z$. Therefore, we have:

$$\Sigma e = (x, y), z \text{ such that } (x, z) \in E \text{ and } (y, z) \in E \text{ BE}(e, z, \sigma) = \text{AFk}(G, \sigma).$$

We get (E1) from the relation: $\Sigma e \text{ BE}(e, z, \sigma) = \Sigma e = (x, y), z \text{ such that } \text{BE}(e, z, \sigma) + \Sigma e = (x, y), z \text{ adjacent to no extremity of } e \text{ BE}(e, z, \sigma)$.

So, solving LAP means seeking a linear ordering $\sigma$ of the vertices of a graph $G = (X, E)$ which is induced by $\Gamma(x)$, such that $\text{BE}(e(x), y, \sigma) = \Sigma x \Sigma y,z \in \Gamma(x), (y, x) \notin E, y, z \text{ located the same way with respect to } x, \sigma 1 \leq \text{Max-Cut}(H(x))$. It follows that, for any $x \in X$, a feasible solution $A(x, \sigma) \cup^{\text{Ex}} B(x, \sigma)$ of Max Cut is defined on $H(x)$, by setting: $A(x, \sigma) = \{y \in \Gamma(x), such that \sigma y x\}$; $B(x, \sigma) = \{y \in \Gamma(x), such that \sigma x y\}$. Its value, in Max Cut sense, is: $m(x) - \Sigma y,z \in \Gamma(x), (y, x) \notin E, y, z \text{ located the same way with respect to } x, \sigma 1 \leq \text{Max-Cut}(H(x))$. We conclude.

Figure 2: Theorem 1 Counting Argument

Remarks: Small experiments make appear that this bound is often tight, specifically in the case of chordal graphs. Still, the Max Cut Problem, which has been extensively studied, is NP-Hard [9]. So, one may ask about the practical use of the above lower bound. The answer is two-sided:
A Quadratic Linear Formulation of LAP.

- **Variables**
  - $U_{x,y}$, $x$, $y$ such that $(x, y)$ edge $\in$ $E$: $U_{x,y} = 1$ $(0) \Leftrightarrow$ $y$ located to the right (left) of $x$
  - $V_{x,y}$, $x$, $y$ such that $(x, y)$ anti-edge $\in$ $E'$: $V_{x,y} = 1 (0) \Leftrightarrow$ $y$ located to the right (left) of $x$
  - $W_{x,y}$, $x \neq y$: $W_{x,y} = 1 (0) \Leftrightarrow$ $y$ located to the right (left) of $x$

- **Constraints** (Consistency)
  - For any $x, y, W_{x,y} + W_{y,x} = 1$
  - For any edge $(x, y) \in E$, $U_{x,y} = W_{x,y}$
  - For any anti-edge $(x, y) \in E'$, $V_{x,y} = W_{x,y}$
  - For any $x, y, z$, all distincts, $W_{x,y} + W_{y,z} \geq W_{x,z}$
  - For any $x, y, z$, all distincts, $(1 - W_{x,y}) + (1 - W_{y,z}) \geq (1 - W_{x,z})$

- **Minimize**

$$\sum_x \sum_{y,z} (1 - V_{x,y}) - \sum_x \sum_{y,z} (1 - U_{x,y})$$

If we denote by $Fork(G)$ the number of forks of the graph $G$, we easily get:

**Theorem 2**: The optimal value of this quadratic $\{0,1\}$
program is equal to $LAP(G) – Tr(G) – Fork(G)$.

This quadratic $\{0,1\}$ model may be easily turned into a
linear one by introducing additional vectors $S$ and $T$ as follows:

- $S = (S_\ell, f = (x, y, z)$, fork with root $x)$ subject to: $S_\ell \leq (1 - U_{x,y})$ and $S_f \leq U_{x,y}$; we consider here that forks are oriented, that means that $(x, y, z)$ and $(x, z, y)$ define 2 distinct forks with root $x$;

- $T = (T_\ell, g = (x, y, z)$, anti-fork with root $z)$ $T_\ell$ subject to: $T_\ell \geq (1 - V_{x,y})$ and $T_g \geq V_{x,y}$; we consider here that anti-forks are oriented, that means that $(x, y, z)$ and $(y, x, z)$ define 2 distinct anti-forks with root $z$.

Then minimizing the quadratic quantity
$$\sum_x \sum_{y,z} (1 - V_{x,y}) - \sum_x \sum_{y,z} (1 - U_{x,y})$$
means minimizing the linear quantity
$$\sum_x T_g - \sum_x S_f.$$
V. A BRANCH/BOUND ALGORITHM FOR LAP

We may derive from previous section an exact Branch/Bound method for LAP:

- **Branching** is performed by picking up some pair \((x, y)\) of nodes and considering the two alternatives \(x \sigma y\) and \(y \sigma x\) according to linear ordering \(\sigma\). Any sequence of such decisions may be extended through transitivity into a partial ordering of the node set \(X\);

- **Bounding** is performed through integer linear programming, while extending Theorem 1 in a natural way: we give Branching strategy comes in a natural way: we give  
- We say that \(G\) is a **Unit Interval graph** if intervals \([o(x), d(x)], x \in X\) may be chosen in such way that no pair \(x, y\) exists such that \(x \subset y\).

Then we see that the quantity \(Tr(G) + \sum_x V_d(x) + \sum_x W_d(x)\) provides us with a lower bound for the best (in LAP sense) linear extension of \(\sigma\).

- Branching strategy comes in a natural way: we give priority to pairs \((x, y)\) which define edges of the graph \(G\), and choose them in such a way the difference between the best alternative and the worst one is the largest possible.

VI. THE CASE OF INTERVAL GRAPHS: A RESTRICTED VERSION OF LAP

We first introduce additional definitions related to interval graphs:

- A simple graph with no loop \(G = (X, E)\) is an **interval graph** if it is the intersection graph of a set \([o(x), d(x)], x \in X, of closed intervals of the real line. Those intervals may be chosen such that points \(o(x), d(x), x \in X, are distinct. We assume this hypothesis to be always satisfied. We set:

- \(x \subset y\) if \(o(x) < o(y)\);

- \(x \sigma y\) if \(o(x) < o(y) < d(x)\);

- \(x \mathcal{O} y\) if \(o(x) < o(y) < d(x)\);

- **In case** \(X\) is an interval family with distinct endpoints, we say that a linear ordering \(\sigma\) of \(X\) is \((\mathcal{O}, \subset)-consistent\) if it is consistent with both orderings \(\mathcal{O}\) and \(\subset\). We denote by \(\sigma\)-can the canonical linear ordering, which is defined as follows: \(x \sigma\)-can \(y\) if, and only if, \(o(x) < o(y)\).

- Then, we say that a **fork** \(f = \{x, y, z\}\) with root \(x\) of such an interval graph \(G = (X, E)\) is a strong fork if there exists \(t \in \{y, z\}\) such that \(t \subset x\), and that a triangle \((x, y, z)\) is a strong triangle if at least some node is contained into another one (for instance \(z \subset x\)).

- We say that \(G\) is a **Unit Interval graph** if intervals \([o(x), d(x)], x \in X\) may be chosen in such way that no pair \(x, y\) exists such that \(x \subset y\).

- We finally say that a subset \(Y\) of \(X\) is a **Left-(\mathcal{O}, \subset)-Section** \((\text{Right-(\mathcal{O}, \subset)-Section})\) if, for any \(x, y \in X\) such that \(x \in Y\) and \((y \mathcal{O} x) \vee (y \subset x)\), then we also have \(y \in Y\) \((x \in Y)\).

**A. A Direct Application of Theorem 1 to Unit Interval Graphs**

In the case of unit interval graphs, Theorem 1 allows us to state:

**Theorem 2:** If \(G = (X, E)\) is a unit interval graph, then \(\sigma\)-can is an optimal solution of LAP.

**Proof:** Let us suppose that an elementary break \(e = (x, y), z, \sigma\)-can exists, and that \(x \mathcal{O} y\) which implies that \(x \sigma\)-can \(y\). If \(x \subset z\) then \(y \sigma\)-can \(z\) and \(z\) does not break \(e\). Similarly, if \(z \subset x\) then \(z \subset x\) and \(z\) does not break \(e\). It comes that \(x \cap z\) is not empty. By the same way, \(y \cap z\) is not empty and \([x, y, z]\) is a triangle. So, there is a one-to-one correspondence between triangles and elementary breaks. So, \(BG(G, \sigma\)-can\) = \(Tr(G)\), and we conclude. □
B. An Approximation Result.

In the case of general interval graphs, $\sigma$-can may not be optimal. As a matter of fact, optimal solution may even not be $(Ov, <<)$-consistent:

\[ \Rightarrow \text{LAP}(G) = 11, \text{optimal } \sigma\text{-opt such that } y \sigma\text{-opt } z \sigma\text{-opt } x, \]
while $BG(G, \sigma\text{-can}) = 14$

\[ \Rightarrow \text{Ex: } y \text{ before } z \text{ before } x \]

\[ \text{Figure 5: Theorem 2 Argument} \]

\[ \text{Figure 6: non consistency of } (Ov, <<) \]

Still, what can be easily checked is that $\sigma\text{-can}$ produces a 2-approximation if we refer to the standard definition of LAP:

**Theorem 3:** Given an interval graph $G = (X, E)$ with $m$ edges. Then the following inequality holds:

\[ BG(G, \sigma\text{-can}) \leq 2\text{LAP}(G) + m. \]

**Proof:** A Global Break oriented proof comes by induction on the cardinality of $X$. Let $x_0$ be the first (smallest) element of $X$ according to $\sigma\text{-can}$, and $\sigma\text{-opt}$ some optimal solution of LAP. Induction tells us that:

\[ \Sigma_{e, z \neq x_0} \text{BE}(e, z, \sigma\text{-can}) \leq m - \Pi_{z \neq x_0} \text{BE}(e, z, \sigma\text{-opt}). \]

Since all vertices of $\Gamma(o(x_0)) \cup \{x_0\}$ are consecutive according to $\sigma\text{-can}$, we get that:

\[ \Sigma_{x \in \Gamma(o(x_0))} \text{BE}((x_0, x), z, \sigma\text{-can}) = \]

\[ \Sigma_{x \in \Gamma(o(x_0))} \text{BE}((x_0, x), z, \sigma\text{-can}) = \Pi_{x \in \Gamma(o(x_0))}((\Gamma(o(x_0)) - 1)/2). \]

On the other hand, if we refer to $\sigma\text{-opt}$, we get:

\[ \Sigma_{x \in \Gamma(o(x_0)), z \neq x} \text{BE}((x_0, x), z, \sigma\text{-opt}) \geq \]

\[ \Sigma_{x, z \in \Gamma(o(x_0))} \text{BE}((x_0, x), z, \sigma\text{-opt}) \geq \]

\[ [\Pi_{x \in \Gamma(o(x_0))}((\Gamma(o(x_0)) - 1)/2) - 1/2]. \]

We derive the result by combining (E2) and (E3). \( \square \)

### C. A Restricted Version CLAP of LAP.

However, experiments will show that best linear orderings are most often $(Ov, <<)$-consistent. So we are going to study the following restriction CLAP of LAP:

**$(Ov, <<)$-Consistent Linear Arrangement Problem (CLAP):** [Compute a $(Ov, <<)$-consistent linear ordering $\sigma$ which minimizes $BG(G, \sigma)$].

The following lemma bridges CLAP with Theorem 1.

**Lemma 1:** In the case the linear ordering $\sigma$ is $(Ov, <<)$-consistent, we have that: $BG(G, \sigma) = Tr(G) + SFk(G, \sigma)$, where $SFk(G, \sigma)$ is the number of strong forks $f = (x, y, z), x = \text{Root}(f)$ such that $((x, y) \land (x, z)) \lor (y, x) \land (z, x)$.

**Proof:** left to the reader (same proof as for Theorem 1). \( \square \)

Extending Theorem 1 to CLAP leads us to introduce a specific version of Max-Cut:

**$(Ov, <<)$-Consistent Unit Cost Max-Cut Problem (C-Max-Cut):** Given a graph $H = (Z, F)$, which is the complementary graph of an interval graph $H^c = (Z, F^c)$, and two disjoint subsets $A_0$ and $B_0$ of $Z$, such that:

- $A_0$ ($B_0$) is a $\text{Left-}(Ov, <<)$-Section ($\text{Right-}(Ov, <<)$-Section) of $H^c$;
- Both $A_0$ and $B_0$ define complete sub-graphs of $H^c = (Z, F^c)$.

Compute a partition $Z = A \cup B$, such that:

1. $A$ contains $A_0$ and is a $\text{Left-}(Ov, <<)$-Section of $H^c$;
2. $B$ contains $B_0$ and is a $\text{Right-}(Ov, <<)$-Section of $H^c$;
3. the number of edges of $H$ which connect $A$ and $B$ is $I \{((x, y) \in E, x \in A, y \in B)\}$ is the largest possible;
4. $A$ is maximal for the set inclusion order, provided 1, 2, 3 are satisfied.

We denote by $C\text{-Max-Cut}(H, A_0, B_0)$ the related optimal value. Then we set, for the interval graph $G = (X, E)$ and for any vertex $x$ in $X$:

- $\Gamma^{Ov}_c(x) = (Ov, <<)$-neighbour set of $x = \{y \in \Gamma_0(x), y \neq x, \text{ such that } (y \subset x) \lor (y, Ov, x) \lor (Ov, y)\}$;
- $H(x) = \text{complementary graph of the sub-graph induced by } \Gamma^{Ov}_c(x)$;
- $A_0(x) = \{y \in Z \text{ such that } y, Ov, x\}; B_0(x) = \{y \in Z \text{ such that } x, Ov, y\}$;
Then we get:

**Lemma 2:** \( CLAP(G) \geq Tr(G) + \sum_x CV(x) \).

**Proof:** For every \( x \in X \), we set \( \mathbf{E}^*(x) = \{ \text{non oriented pairs } (y, z) \text{ such that } y \subset x \text{ or } z \subset x \text{ holds} \} \).

For any \( x \in X \), we get a feasible solution \( A(x, \sigma) \cup \mathbf{B}(x, \sigma) \) of the \( \text{C-Max-Cut} \) instance defined by \( H(x), A_0(x), B_0(x) \), by setting: \( A(x, \sigma) = \{ y \in \Gamma_{Ov}(x), \text{such that } y \sigma x \}; B(x, \sigma) = \{ y \in \Gamma_{\text{bal}}(x), \text{such that } x \sigma y \} \). Its value is:

\[ m(x) - \sum_{x,y \in \Gamma_{Ov}(x)} |E(x,y)\leq \sum_{x,y \in \Gamma_{\text{bal}}(x)} CV(x) \leq \sum_{x} CV(x) \].

It follows that, for any \( x \in X \):

\[ |E^*(x)| \geq \sum_{x,y \in \Gamma_{Ov}(x), y \in E, y,z \text{ located the same way with respect to } x, \sigma} 1 \]

\[ \geq CV(x) \] and \( \sum_{x} \mathbf{E}^*(x) \geq \sum x CV(x) \).

Then we get that \( Sfk(G, \sigma) = \sum x \mathbf{E}^*(x) \geq \sum_x CV(x) \).

We conclude. \( \square \)

**D. Solving \text{C-Max-Cut} and Evaluating \( CV(x) \)**

The complexity of the \text{Max Cut} problem in the case of the complementary graph of an interval graph is still an open issue. However, things are easier with \text{C-Max-Cut}:

**Theorem 4:** Given \((Z, F), A_0 \) and \( B_0 \) as in the definition of \text{C-Max-Cut}. Let us set, for every vertex \( z \in Z - (A_0 \cup B_0) \):

- \( d_H(A_0, z) = \{ t \in Z - A_0 \text{ such that } t \prec z, t \prec z \} \) + \{ t \in A_0 \text{ such that } t \prec z \};
- \( d_H(B_0, z) = \{ t \in Z - B_0 \text{ such that } z \prec t, z \prec t \} \) + \{ t \in B_0 \text{ such that } z \prec t \}.

Then we solve \text{C-Max-Cut} by setting:

\[ A = \{ z \in Z - (A_0 \cup B_0) \text{ such that } d_H(A_0, z) \geq d_H(B_0, z) \} \cup A_0; \]

\[ B = \{ z \in Z - (A_0 \cup B_0) \text{ such that } d_H(A_0, z) < d_H(B_0, z) \} \cup B_0. \]

**Proof:** Left to the reader. \( \square \)

**E. An Exact Solution \( \sigma\text{-bal} \) for \( CLAP \).**

We construct this solution \( \sigma\text{-bal} \), by setting, for any pair \( x, y \) in \( X \), \( \sigma\text{-bal} \) if, and only if, one among the following options holds:

- \((x \prec y) \text{ or } (x \text{Ov} y)\); \( \sigma\text{-can} \):
- \((x \prec y) \text{ and } d_{H(y)}(A_0(y), x) \leq d_{H(y)}(B_0(y), x); \sigma\text{-bal} \)
- \((y \prec x) \text{ and } d_{H(x)}(B_0(x), y) < d_{H(x)}(A_0(x), y); \sigma\text{-can} \).

**Figure 7:** A \( \sigma\text{-bal} \) construction

**Lemma 3:** The \( \sigma\text{-bal} \) relation is transitive.

**Proof:** Left to the reader. \( \square \)

We are now ready to state the optimality of \( \sigma\text{-bal} \).

**Theorem 5:** The relation \( \sigma\text{-bal} \) is an optimal solution of \( CLAP \), which satisfies:

1. \( BG(G, \sigma\text{-bal}) \leq BG(G, \sigma\text{-can}) \).
2. \( Tr(G) \leq BG(G, \sigma\text{-bal}) = Tr(G) + \sum_x CV(x) \leq Tr(G) + \text{Strong-Fork}/2 \), where \text{Strong-Fork} is the number of strong forks of the interval graph \( G \).

**Sketch of the Proof:** From Lemma 3, we have that \( \sigma\text{-bal} \) is a \((\prec, \text{Ov})\)-consistent linear ordering. Then the optimality of \( \sigma\text{-bal} \) (and so, the fact that \( BG(G, \sigma\text{-bal}) \leq BG(G, \sigma\text{-can}) \)) derives, through a simple computation, from the fact that since it locally achieves, for any node \( x \), the lower bound \( CV(x) \), then it also globally achieves the lower bound of Lemma 2. \( \square \)

We easily deduce that this result has an algorithmic interpretation:
**Corollary 3** (left to the reader): Computing $\sigma$-bal may be done in $O(Arc-\subset)$ time, where Arc-\subset is the number of arcs of the digraph induced on $X$ by the $\subset$ ordering.

**VII. NUMERICAL EXPERIMENTS**

We implemented both the Branch/Bound algorithm of Section 5 and the Approximation algorithm $\sigma$-bal. We did it on a LINUX server CentOS 5.4, Processor Intel Xeon 3.6 GHZ, while using the CPLEX 12 library when dealing with integer linear programs. For interval graphs, our Branch/Bound scheme could solve, in no more than few minutes, instances with up to 40 nodes. This allowed us to perform a comparative analysis of the precision of the Lower Bound $LB(G) = Tr(G) + \Sigma x \cdot V(x)$ provided by Theorem 1, and of both approximation algorithms $\sigma$-can and $\sigma$-bal.

We use 10 instance groups related to $\text{Card}(X) = 10, 20, 30$ and 40, and for every instance group, compute:

- the mean gap $\text{LB-GAP} = (\text{LAP}(G) - \text{LB}(G))/\text{LB}(G)$ between the optimal LAP value and the lower bound LB;
- the mean gap $\text{CLAP-GAP} = (\text{BG}(G, \sigma\text{-bal}) - \text{LAP}(G))/\text{LAP}(G)$ between the optimal value of CLAP, computed by $\sigma$-bal, and the optimal value;
- the mean gap $\text{CAN-GAP} = (\text{BG}(G, \sigma\text{-can}) - \text{LAP}(G))/\text{LAP}(G)$ between the value defined by the canonical ordering $\sigma$-can, and the optimal value.

We get results which are described in the following Table 1.

<table>
<thead>
<tr>
<th>N</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
</tr>
</thead>
<tbody>
<tr>
<td>LB-GAP</td>
<td>9.8%</td>
<td>12.5%</td>
<td>10.7%</td>
<td>14.5%</td>
</tr>
<tr>
<td>CLAP-GAP</td>
<td>3.2%</td>
<td>4.5%</td>
<td>4.1%</td>
<td>5.0%</td>
</tr>
<tr>
<td>CAN-GAP</td>
<td>9.6%</td>
<td>12.3%</td>
<td>11.5%</td>
<td>10.9%</td>
</tr>
</tbody>
</table>

*Table 1: Comparative precision of lower bound LB and approximation solutions $\sigma$-can and $\sigma$-bal.*

Table 2 provides now the specific results related to $n = 10$, which by the way, gives an estimation of the LAP(G) values which may derive from interval graphs of this size:

<table>
<thead>
<tr>
<th>INSTANCE NUMBER</th>
<th>LB(G)</th>
<th>LAP(G)</th>
<th>CLAP(G)</th>
<th>CAN(G)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>12</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>13</td>
<td>16</td>
<td>17</td>
<td>19</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>13</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>11</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>15</td>
<td>15</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>8</td>
<td>13</td>
<td>15</td>
<td>15</td>
<td>17</td>
</tr>
<tr>
<td>9</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>10</td>
<td>9</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

*Table 2: Values LAP(G), LB(G), CLAP(G) and CAN(G) related with a 10 instances group with Card(X) = 10.*

**VIII. CONCLUSION**

This paper, with theoretical focus, proposes approximation results for the Linear Arrangement problem, in the case of interval graphs. Further research should be about the extension of our approaches to chordal graphs and circular graphs, as well as about the design of efficient exact algorithms.

**REFERENCES**


