RAIRO-Oper. Res. 52 (2018) 1123–1145
https://doi.org/10.1051/ro/2017011

RAIRO Operations Research
www.rairo-ro.org

LOWER AND UPPER BOUNDS FOR THE LINEAR ARRANGEMENT PROBLEM ON INTERVAL GRAPHS

Alain quilliot[1], Djamal rebaine[2,*] and Hélène toussaint[1]

Abstract. We deal here with the *Linear Arrangement Problem* (LAP) on *interval* graphs, any interval graph being given here together with its representation as the *intersection* graph of some collection of intervals, and so with related *precedence* and *inclusion* relations. We first propose a lower bound *LB*, which happens to be tight in the case of *unit interval* graphs. Next, we introduce the restriction PCLAP of LAP which is obtained by requiring any feasible solution of LAP to be consistent with the *precedence* relation, and prove that PCLAP can be solved in polynomial time. Finally, we show both theoretically and experimentally that PCLAP solutions are a good approximation for LAP on *interval* graphs.

Mathematics Subject Classification. 90C27.

Received June 23, 2015. Accepted March 2, 2017.

1. Introduction

Let $G = (X,\, E)$ be a non oriented graph where X and E respectively denote the set of nodes and the set of edges of G. The *Linear Arrangement* problem (LAP) consists in finding a one-to-one mapping ϕ from X to $\{1,\, \ldots,\, |X|\}$ that minimizes the quantity: $LAP(\mathrm{G},\, \phi) = \Sigma_{(x,y)\in E}\ |\phi(y) - \phi(x)|$.

LAP and similar layout problems appear in different contexts, for example in Electrical Engineering and Telecommunications [11, 20, 24], Biology [18], Human Sciences [21], Information Retrieval [2, 31] or Warehouse Management [24]. In any case, one has to store or locate objects on a line in such a way two contiguous (adjacent) objects remain the closest possible according to this storage strategy.

LAP was first shown to be *NP-Hard* for arbitrary graphs (see *e.g.* [7, 11, 12, 15]) and next, for some restricted classes of graphs such as *interval* graphs [9] and b*ipartite* graphs [15]. Furthermore, non approximability results were presented in [18]. However, polynomial time algorithms were also developed for special graphs such as trees [7], *unit interval* graphs [10, 32], paths, cycles, complete graphs, complete *bipartite* graphs and *grid* graphs [13, 18, 19], *outer-planar* graphs [14], *chord* graphs [28], and restricted *series-parallel* graphs [1]. Surveys are available in [11, 18, 21, 27].

By the same way, general lower bounds may be found in [5, 6, 19], which involve linear programming formulations. Different kinds of heuristics and exact algorithms (branch and bound and dynamic programming) may be

Keywords. Interval Graphs, Linear Ordering.

[1] LIMOS, UMR CNRS 6158, LABEX IMOBS3, Université Blaise Pascal, Clermont-Ferrand, France.

[2] Département d'Informatique et de Mathématique, Université du Québec à Chicoutimi, Saguenay, QC G7H 2B1, Canada.

[*] Corresponding author: djamal.rebaine@uqac.ca

found in [11, 21, 22, 25–27, 29, 30], together with numerical experiments which make appear that, in the general case, LAP is an extremely difficult problem. The goal of the present study is mainly to provide tools for the handling of *interval* graphs.

The paper is organized as follows. Section 2 introduces notations and definitions, and also includes a reformulation of LAP, which involves linear orderings of X and a notion of *elementary break*. In Section 3, we derive from this reformulation a general lower bound $LB(G)$ for any *interval* graph $G = (X, E)$, which happens to be tight for *unit interval* graphs and which never misses optimality by more than 2.7 % in the experiments which we conduct on general *interval* graphs at the end of Section 3. In Section 4, we show how to compute, in polynomial time, an optimal solution of the restriction CLAP of LAP which imposes any feasible solution of LAP to be *consistent* with the *precedence* relation. This solution provides us with an upper bound $PCGB^*(G)$ for LAP. In Section 5 we bound the gap between $PCGB^*(G)$ and $LB(G)$ through both a theoretical result and a numerical experiment.

2. Notations, definitions, and LAP reformulation

An undirected graph (with no loop) is denoted by $G = (X, E)$, with node set X and edge set E. Any edge with end-nodes x and y in X is denoted by (x, y). Since G is undirected, (x, y) and (y, x) are the same. We call *anti-edge* of G any pair $[x, y]$, $x \neq y$, such that $(x, y) \notin E$, and we denote by E^c the set the *anti-edges* of G. If $A \subseteq X$, then G_A is the *proper sub-graph* induced by A into G. If $x \in X$, then $N(x)$ denotes the *neighbor* set $N(x) = \{y \in X \text{ such that } (x, y) \in E\}$. In case we are simultaneously dealing with several graphs (in Sect. 5), then we specify the related graph while using the notation $N(G, x)$. Finally we denote by $Z = A \cup^{\mathrm{Ex}} B$ any partition of a set Z into 2 disjoint subsets A and B (eventually empty), and by $\wedge$ and $\vee$, respectively the logical operators AND and OR.

2.1. Interval graphs

Let us recall that, if S is some given set, and if $F$is some collection of subsets of S, then the *intersection* graph induced by S and F is the undirected graph whose node set is F and whose adjacency relation is defined by:

– f, $f' \in F$ are adjacent in the related *intersection* graph if and only if $f \neq f'$ and $f \cap f' \neq Nil$, where *Nil* denotes the empty subset of S, that means if and only if f and f' are *intersecting*.

An undirected graph (with no loop) $G = (X, E)$ is an *interval graph* if it is possible to associate, with any node $x \in X$, a closed interval $I(x) = [o(x), d(x)]$ of the real line in such a way that x and y are adjacent in G if and only if $x \neq x'$ and $I(x)$ and $I(y)$ are intersecting. This also means that if we identify X with the interval collection $I = \{I(x) = [o(x), d(x)], x \in X\}$, then G is the *intersection* graph of this interval collection.

G is an *unit interval* graph if the intervals $I(x)$ may be chosen with length equal to 1, for every $x \in X$. It is known (see for instance [10]) that G is an *unit interval* graph if and only if those intervals $I(x)$ may be chosen in such a way that no interval is included into another one. Every time we talk here about *unit interval* graph, we refer to this weaker characterization.

2.2. Interval representations

Let $G = (X, E)$ be an *interval graph*. We call *interval representation* of G any interval collection $I = \{I(x) = [o(x), d(x)], x \in X\}$ such that x and y are adjacent in G if and only if $x \neq x'$ and $I(x)$ and $I(y)$ are intersecting. In such a case, we identify X with the collection $I = \{I(x), x \in X\}$, and consider G as the *intersection* graph of this interval collection. It is known that such an *interval representation* of the graph G may be chosen in such a way that all end-points $o(x)$, $d(x)$, $x \in X$, are distinct. Assuming that $G = (X, E)$ is the *intersection* graph of an interval collection X and that this collection X satisfies this *distinct end-point hypothesis*, then we can introduce additional relations between the nodes of G:

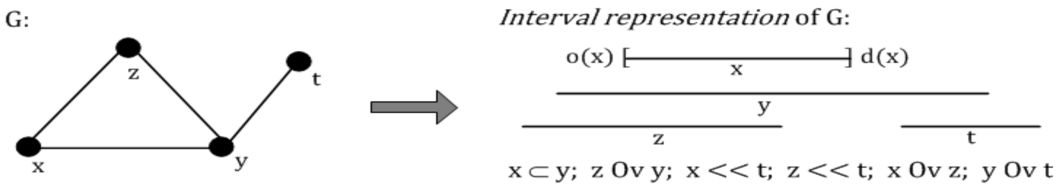

FIGURE 1. An Interval graph and its interval representation.

- *Inclusion* relation $\subset$: $x \subset y$ if $o(x) > o(y)$ and $d(y) > d(x)$,
- *Precedence* relation $\ll$: $x \ll y$ if $d(x) < o(y)$,
- *Overlap* relation Ov: $x\ \ Ov\ \ y$ if $o(x) < o(y) < d(x) < d(y)$.

So we assume, throughout the rest of this paper, that any *interval* graph $G = (X, E)$ is defined as being the *intersection* graph of an interval collection X, whose end-points are pair-wise distinct. If the relation $\subset$ is empty, then we talk about a *unit interval* graph. Figure 1 below shows such an *interval* graph G, together its related *precedence*, *inclusion* and *overlap* relations.

2.3. Linear orderings

A *linear ordering* σ of a set X is an order relation σ (transitive, anti-symmetric) such that for any pair x, y in X, $x \neq y$, we either have $x\ \sigma\ y$ or $y\ \sigma\ x$. One may also view σ as a way to sequence the elements of X and represent it as a list.

Given a linear ordering σ of a set X, and x, y, z in X, all distinct: we say that y and z are on the *same side with respect to* x (according to σ) if we have either $((x\ \sigma\ y) \wedge (x\ \sigma\ z))$ or $((y\ \sigma\ x) \wedge (z\ \sigma\ x))$. If, for instance X is the set $\{A, B, C, D, E\}$ and if σ is the sequence $\{B, E, C, D, A\}$ then we see that:

- C and A are on the *same side with respect* to E;
- C and A are not on the *same side with respect* to D.

Given an interval collection X, whose *intersection* graph defines the *interval* graph $G = (X, E)$, and some linear ordering σ of X. We say that σ is *precedence consistent* if it is *consistent* to the *precedence* relation $\ll$, meaning that, for any pair x, y such that $x \ll y$, then it must be the case that $x\ \sigma\ y$. Finally we denote by x *σ-can* y (the *canonical linear ordering* between x and y) if $o(x) \ll o(y)$ holds for every pair $x, y \in X$.

Example 2.1. If we refer to the graph $G = (X, E)$ of Figure 1, then we see that: z *σ-can* y *σ-can* x *σ-can* t.

2.4. Reformulation of the linear arrangement problem

Let $G = (X, E)$ be a graph, and let σ be a *linear* ordering of X. For any edge $e = (x, y)$, $z \in X - \{x, y\}$, we set $EB(e, z, \sigma) = 1$ if x and y are on the *same side with respect* to z according to σ and 0 otherwise, and call this quantity the *elementary break* of e by z according to σ.

We also set $GB(G, \sigma) = \Sigma_{e,z}\ EB(e, z, \sigma)$ and call the quantity $GB(G, \sigma)$ the *Global Break* of G according to σ.

Explanation. Figure 2 shows the *elementary break* and the *Global Break* values induced by some *linear* ordering σ on a graph $G = (X, E)$.

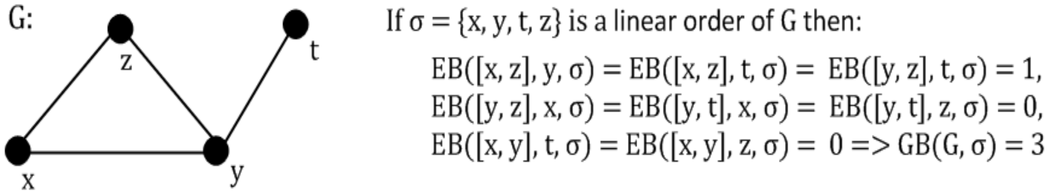

FIGURE 2. Elementary breaks and Global Break of G according to σ.

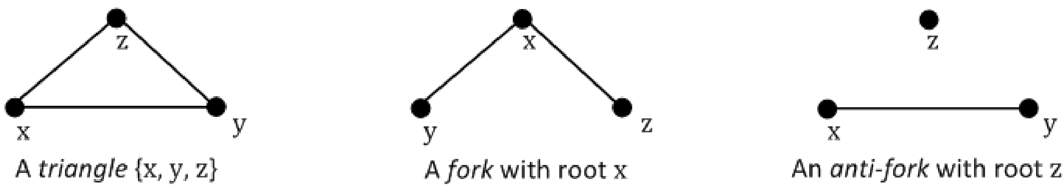

FIGURE 3. Triangles, forks, anti-forks.

Let us denote by we denote by $GB^*(G) = \text{Inf}_{\ \sigma}\ GB(G,\, \sigma)$ the optimal (minimal) *Global Break* value, and let us denote by $LAP^*(G) = \text{Inf}_{\ \phi}\ LAP(\text{G},\, \phi)$ th optimal value for LAP. Then we may state the following *Reformulation* Lemma, which will be the key for our approach for LAP:

Reformulation Lemma 0: *LAP*(G) = GB*(G) + Card(E).*

Proof. Given a *linear* ordering σ of X, together with the one-to-one mapping $\phi(\sigma)$ from X into $\{1,\, \ldots,$ Card$(X)\}$ which naturally derives from σ. We see that:

$$\begin{aligned} LAP(G, \phi(\sigma)) &= \Sigma_e =_{(x,y)\in E}\ j\phi(\sigma)(y) - -\phi(\sigma)(x)j \\ &= \Sigma_e(1 + \Sigma_z EB(e, z, \sigma)) \\ &= \text{Card}(E) + \Sigma_{e,z} EB(e, z, \sigma) = GB(G, \sigma) + Card(E). \end{aligned}$$

Since the correspondence σ ->$\phi(\sigma)$ is one-to-one and since any mapping ϕ from X into $\{1, \ldots,$ Card$(X)\}$ may be written $\phi(\sigma)$ for some linear ordering σ, solving LAP means computing σ which minimizes $GB(G,\, \sigma)$. We conclude. □

2.5. Triangles, Forks, Anti-Forks, Strong Triangles and Forks

A *triangle* of G is a clique with 3 nodes. We denote by $Tr(G)$ the number of *triangles* in G. A *fork* with *root* x is any pair $f = (x,\, [y,\, z])$ made of a node x and an *anti-edge* $[y, z]$ such that $(x,\, y) \in$ E and $(x,\, z) \in E$. An *anti-fork* with *root* z is any pair $h = ((x,\, y),\, z)$ made of an edge $(x,\, y)$ and a node z, such that $[x,\, z]$ and $[y,\, z] \in E^c$ (see Fig. 3).

Explanation. The above Figure 3 illustrates what are respectively a *triangle*, a *fork* and an *anti-fork*. We notice that an *anti-fork* of G is nothing more than a *fork* in the complementary graph of G.

Let us suppose now that $G = (X,\, E)$ is an *interval* graph. Then we say that:

– a *fork* $f = (x,\, [y,\, z])$ with root x is *strong* if there exists $t \in \{y,\, z\}$ such that $t \subset x$ (see Fig. 4).

Explanation. Figure 4 shows a collection of 3 intervals which define a *strong fork*.

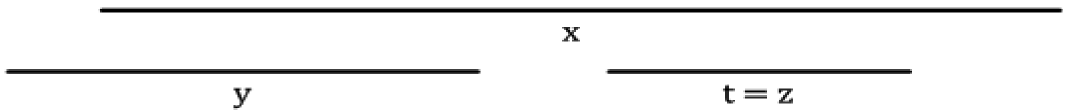

FIGURE 4. Strong fork $f = (x, [y, z])$.

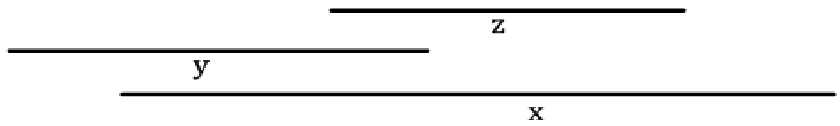

FIGURE 5. Strong triangle.

– a *triangle* $\{x,\ y,\ z\}$ is a *strong triangle* if at least one node in $\{x,\ y,\ z\}$ is included into another one (for instance $z \subset x$) as pictured by Figure 5.

Explanation. Figure 5 shows a collection of 3 intervals which define a *strong triangle.*

Those definitions allow us to state the following Lemma 2.2, which will be the basis, in Sections 3 and 4, for the computation of lower bounds for the $GB^*(G)$ value.

Lemma 2.2. *Let us consider some linear ordering σ of G, and let us set, for any $x \in X$: $Fk(G,\ \sigma,\ x) =$ Card($\{$forks $f = (x,\ [y,\ z])$, such that y and z on the same side according to σ with respect to $x\}$. Let also set:*

– *$Fk(G,\ \sigma) = \Sigma\ _x\ Fk(G,\ \sigma,\ x)$;*
– *$AFk(G,\ \sigma) =$ Card($\{$anti-forks $h = ((x,\ y),\ z)$, such that x and y not on the same side according to σ with respect to $z\}$).*

Then the following equality holds: $GB(G, \sigma) = Tr(G) + Fk(G, \sigma) + AFk(G, \sigma)$. (E1)

Proof. We know that $GB(G,\ \sigma) = \Sigma_{e,z}\ EB(e,\ z,\ \sigma)$. So, in order get (E1), we consider an edge $e = (x,\ y)$, together with a node z, different from x and y, and notice that computing the number of non null $EB((u,\ v),\ w,\ \sigma)$ values such that $\{x,\ y,\ z\} = \{u,\ v,\ w\}$ leads us to consider 3 cases:

– **Case 1.** x, y and z define a *triangle.*

Then $EB(e,\ z,\ \sigma) = 1$ if either $x\ \sigma\ z\ \sigma\ y$ or $y\ \sigma\ z\ \sigma\ x$. In such a case no quantity $EB((x,\ z),\ y,\ \sigma)$, $EB((z,\ y),\ x,\ \sigma)$ is equal to 1. We deduce that $\Sigma_{e=(x,y),z}$ such that $\{x, y, z\}$ is a triangle $EB(e,\ z,\ \sigma)$ is equal to the number $Tr(G)$ of *triangles* of G.

– **Case 2.** z is adjacent to exactly one node t in {x, y} . We denote by t^* the other node.

Then $EB(e,\ z,\ \sigma) = 1$ if either $t\ \sigma\ z\ \sigma\ t^*$ or $t^*\ \sigma\ z\ \sigma\ t$. In such a case, $EB((t,\ z),\ t^*,\ \sigma) = 0$. Conversely, if $EB((t,\ z),\ t^*,\ \sigma) = 1$ then we see that $EB(e,\ z,\ \sigma) = 0$. So the *fork* $f = (t,\ [t^*,\ z])$ gives rise to a non null value $EB((t,\ t^*),\ z,\ \sigma) + EB((t,\ z),\ t^*,\ \sigma)$ (which is then equal to 1) if and only if t^* and z are on the *same side with respect* to t according to σ. We deduce:

$\Sigma_{e=(x,y),z \text{adjacent to exactly 1 extremity of e}}\ EB(e,\ z,\ \sigma) = Fk(G,\ \sigma)$.

– **Case 3.** z is adjacent neither to x nor y, which means that $h = ([x,\ y],\ z)$ is an *anti-fork* with root z. Then the only situation in which the *elementary break* $EB(e,\ z,\ \sigma)$ can be different from 0 corresponds to the case when z is located *between* x and y according to σ (x and y are not on the *same side with respect* to z), that means when h is taken into account in $AFk(G,\ \sigma)$. Therefore, we have:
$\Sigma_{e=(x,y),z \text{such that } (x,z)\notin E \text{and} (y,z)\notin E}\ EB(e,\ z,\ \sigma) = AFk(G,\ \sigma)$.

According this, we have: $\Sigma_{e,z}\ EB(e,\ z,\ \sigma) = \Sigma_{e=(x,y),z \text{ such that} \{x,y,z\} \text{ is a triangle}}\ EB(e,\ z,\ \sigma)$
$+ \Sigma_{e=(x,y),z \text{ adjacent to exactly 1 extremity of } e}\ EB(e,\ z,\ \sigma) + \Sigma_{e=(x,y),z \text{ adjacent to no extremity of } e}\ EB(e,\ z,\ \sigma)$.

This proves (E1). □

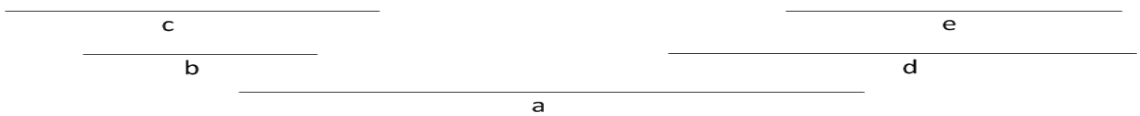

Figure 6. Illustration of the values m(x) and MC(x).

3. A Lower Bound LB for LAP in the case of *INTERVAL* graphs

Given an *interval* graph $G = (X, E)$. Computing a *linear* ordering σ of X means deciding, for every node x, which nodes of the *neighbor* set $N(x)$ are going to be located before or after x according to σ. More precisely, Lemma 0 tells us that if $\{x, y, z\}$ defines a *triangle*, then, whatever be the way x, y and z are ordered according to σ, $\{x, y, z\}$ gives rise to exactly one not null *elementary break* value $EB((u, v), w, \sigma)$ value such that $\{x, y, z\} = \{u, v, w\}$. But, if $(x, [y, z])$ is a *fork* with root x, then it gives rise to a non null *elementary break* value $EB((x, u), v, \sigma)$ such that $\{u, v\} = \{y, z\}$ if and only if y and z are on the *same side with respect* to x according to σ. Intuitively, that means that, in case $f = (x, [y, z])$ is a *fork* with root x, a good ordering σ should be, as often as possible, such that y and z are not on the *same side with respect* to x according to σ.

This remark leads us to denote, for any pair A, B of disjoint subsets of X, by $\rho(A, B)$ the number of *anti-edges* $[x, y]$, with one extremity in A and the other in B, that means the *cut-size* over the *anti-edge* set induced by partition A, B of node set X:

– $\rho(A, B) = \text{Card}(\{[y, z] \in E^c \text{ such that } y \in A, z \in B\}\ )$.

In case A or B is empty, then $\rho(A, B)$ is clearly null. Then, for any node $x \in X$, we set:

– $m(x) = \text{Card}\ (\{[y, z] \in E^c \text{ such that } y, z \in N(x)\}$;
– $MC(x) =$ Largest value $\rho(A, B)$ taken for all partitions $A \cup^{\text{Ex}} B$ of $N(x)$.

Remark 3.1. Since notation $Z = A \cup^{\text{Ex}} B$ may be used in the case when A or B (or both) are empty, we see that if x is a *pendant* node, then $MC(x) = 0$. On another side, we notice that $MC(x)$ is the optimal value of the *Unit Cost Max-Cut* (see *e.g.* [4, 8, 17]) instance defined on the complementary graph of the induced sub-graph $G_N(x)$.

Example 3.2. Let us suppose that G is the *intersection* graph defined by the collection of intervals as described in Figure 6 below. We see that:

- $m(a) = 4$; $MC(a) = 4$ because of the partition $A = \{b, c\}$, $B = \{d, e\}$ of $N(a)$
- $m(b) = m(c) = m(d) = m(e) = 0$; $Tr(G\ ) = 2$

Explanation. As just told above, we are going to carry on a local approach in order to build *ad hoc* linear ordering σ of the interval collection X and evaluating them: for every x in X, we are going to distribute the nodes of $N(x)$ before and after x according to σ, while trying both to ensure the consistency of those local distribution processes and to avoid *elementary breaks* on the edges of the sub-graph induced by $\{x\} \cup N(x)$. What Theorem 3.4 and Lemma 3.3 are going to show now is that locally minimizing, for a given x, the number of *elementary breaks* induced by this distribution process, is equivalent to solve the *Unit Cost Max-Cut* problem on the complementary graph of the induced sub-graph $G_{N(x)}$: as a matter of fact, the number of *elementary breaks* locally induced by distributing $N(x)$ into two subsets A (those y such that $y\ \sigma\ x$) and B (those y such that $x\ \sigma\ y$) will be checked to be equal to the number of *anti-edges* $[y, z]$ which are entirely located in A or in B, that means to the difference $m(x) - \rho(A, B)$. So, the difference $m(x) - MC(x)$ will provide us with the smallest number of *elementary breaks* that such a local distribution process may induce when performed on a given node x.

3.1. The Lower Bound Statement

This allows us to define the following quantity $LB\ (G) = Tr(G) + \Sigma_x(m(x) - MCx))$. We claim that $LB(G)$ provides us with a lower bound for the $GB^*(G)$ value. Let us first notice that above definitions about $\rho(A, B)$, $m(x)$ and $MC(x)$ allow us to state the following Lemma 3.3:

Lemma 3.3. *For any $x \in X$, we have $m(x) - \rho(A, B) = Fk(G, \sigma, x)$, where $Fk(G, \sigma, x)$ is defined according to the statement of Lemma 2.2.*

Proof. It is a direct application of the definitions. □

Then we may state the main result of this section:

Theorem 3.4. *$GB^*(G) \geqslant LB(G)$.*

Proof. Once again, we consider some linear ordering σ of G. Then, for any $x \in X$, we derive from σ a partition $A(x,\sigma) \cup^{\mathrm{Ex}} B(x, \sigma)$ of $N(x)$, by setting:
$A(x, \sigma) = \{ y \in N(x)$, such that $y\ \sigma\ x\}$; $B(x, \sigma) = \{ y \in N(x)$, such that $x\ \sigma\ y\}$.
Since $\rho(A(x, \sigma), B(x, \sigma)) \leqslant MC(x)$, we use Lemma 3.3 in order to deduce that, for any $x \in X$:
$m(x) - MC(x) \leqslant Fk(G, \sigma, x)$.
By summing on x we get (see the definition of $Fk(G, \sigma)$ in the statement of Lem. 2.2):
$Fk\ (G, \sigma) = \Sigma_x Fk(G, \sigma, x) \geqslant \Sigma_x m(x) - MC(x)$.
Since Lemma 2.2 states that $GB\ (G, \sigma) = Tr(G) + Fk(G, \sigma) + AFk(G, \sigma)$, we conclude. □

Remark 3.5. The above result holds for any undirected graph. Still, if we remove the *Interval Graph* hypothesis and consider trees instead, then the ratio between $GB^*(G)$ and $LB(G)$ can be arbitrarily large. This is due to the fact that $LB(G)$ does not involve any bound (but the trivial bound 0) on the term $AFk(G, \sigma)$: but if G is for instance a tree, then we see that $AFk(G, \sigma)$ strongly contributes to the $GB(G, \sigma)$ value. Conversely, in the case of *interva*l graphs, this value $AFk(G, \sigma)$ will not play any significant role and will disappear if we impose the ordering σ to be *precedence consistent*. So, experiments given at the end of this section will make appear a value of the ratio $(GB^*(G) - LB(G))/LB(G)$ close to 1% in the average, and Section 5 will prove that the error value $GB^*(G) - LB(G)$ cannot exceed the number of *strong triangles*.

3.2. Evaluating the Lower Bound LB(G)

In order to check the quality of $LB(G)$ as a lower bound for $GB^*(G)$, we may first address the case when $G = (X, E)$ is an *unit interval* and notice that:

Proposition 3.6. *In case $G = (X, E)$ is an unit interval graph, then $GB^*(G) = LB(G)$.*

Proof. It is only a matter of checking that $GB(G, \sigma\text{-}can) = Tr(\mathrm{G}) \leqslant LB(\mathrm{G}) \leqslant GB(G, \sigma\text{-}can)$, which will imply that σ-*can* achieves the lower bound $LB(G)$ and so that $GB^*(G) = LB(G)$. Let $f = (x, [y, z])$ be a *fork* of G. We may suppose $y \ll z$ and deduce from the fact that G is an *unit interval* graph that y σ-*can* x σ-*can* z. It comes that the quantity $Fk(G, \sigma\text{-}can)$ of Lemma 2.2 is equal to 0. By the same way, let $h = ((x, y), z)$ be an *anti-fork* of G: then x and y are on the *same side with respect* to z according to σ-*can* (we do not use here the fact that G is a *unit interval* graph). It comes that $AFk(G, \sigma\text{-}can) = 0$. Then Lemma 2.2 allows us to conclude. □

This proposition yields the following corollary, which confirms the optimality of σ-*can* in the case of *unit interval* graphs as stated in [10].

Corollary 3.7. *If $G = (X, E)$ is an unit interval graph, then the canonical linear ordering σ-can is an optimal solution of LAP.*

Also, we may compare $GB^*(G)$ and $LB(G)$ in the case of *interval* graphs through numerical experiments. We compute the values $MC(x)$ which are involved into the $LB(G)$ lower bound while applying the CPLEX 12 library to the following ILP model:

ILP Model *P-MC(x)*:
{Set $\Delta(x) = \{[y, z] \in E^c$ such that both y and z are in $N(x)\}$;
$Compute\ \{0, 1\} - valued\ vectors\ U = (U_y, y \in N(x))\ and\ T = (T_{y,z}, [y, z] \in \Delta(x))$
Subject to:

- For any $[y, z] \in \Delta(x), T_{y,z} \leqslant U_y$;
- For any $[y, z] \in \Delta(x), T_{y,z} \leqslant 1 - -U_z$;

And which *maximizes* $\Sigma_{(y,z)\in\Delta(x)} T_{y,z}\}$.

Explanation. Any vector U provides us in a natural way with a partition ($A = \{y$ such that $U_y = 1\}$, $B = N(x) - A$) of $N(x)$ and value $T_{y,z}$ tells us if we simultaneously have $y \in A$ and $z \in B$.

We compute $GB^*(G)$ while applying the CPLEX 12 library to the following ILP model:

ILP Model *P-LAP(G)*:
{*Compute* the {0, 1} -valued vectors $W = (W_{x,y}, x, y \in X, x \neq y), T = (T_f, f = (x, [y, z])\, fork of\, G), T* = (T*_h, h = ((x, y), z) anti-forkofG)$, which *satisfy the constraints*:

(1) For any x, $y \in X$, $x \neq y$, $W_{x,y} + W_{y,x} = 1$;
(2) For any x, y, $z \in X$, all distinct, $W_{x,z} \leqslant W_{x,y} + W_{y,z}$;
(3) For any *fork* $f = (x, [y, z])$, $T_f + W_{\mathrm{x,y}} + W_{x,z} \geqslant 1$ and $T_f + W_{\mathrm{y,x}} + W_{z,x} \geqslant 1$;
(4) For any *fork* $f = (x, [y, z])$, $T_f \geqslant W_{x,y} + W_{x,z}$ - 1;
(5) For any *anti-fork* $h = ((x, y), z)$, $T^*{}_h + W_{x,z} + W_{y,z} \geqslant 1$ and $T^*{}_h + W_{z,x} + W_{z,y} \geqslant 1$;
(6) For any *anti-fork* $h = ((x, y), z)$, $T^*{}_h \geqslant W_{z,x} - W_{z,y}$ - 1;

and which *minimizes*: $\Sigma_{f\ fork\ of\ G} T_f + \Sigma_{h\ anti\text{-}fork\ of\ G}\ T^*{}_h\}$.

Explanation. This model follows Lemma 2.2 and the equality $GB(G, \sigma) = Tr(G) + Fk(G, \sigma) + AFk(G, \sigma)$. For any x, y in X, $W_{x,y}$ tells us whether $x\ \sigma\ y$ or $y\ \sigma\ x$: constraint (1) tells that either $x\ \sigma\ y$ or $y\ \sigma\ x$ must hold in an exclusive way, and (2) expresses the transitivity of σ. For any *fork* $f = (x, [y, z])$, $T_f = 1$ means that y and z are on the *same side with respect* to x according to σ : constraint (3) tells us that if $T_f = 0$, then $W_{x,y}$ and $W_{x,z}$ are not allowed to take the same value and constraint (4) tells us that if $T_f = 1$ then they are not allowed to take different values. By the same way, for any *anti-fork* $h = ((x, y), z), T^*{}_h = 1$ means that x and y are on the *same side with respect* to z according to σ: Then constraints (5) and (6) work as constraints (3) and (4). It comes that $GB^*(G)$ is equal to the sum of $Tr(G)$ and the optimal value of the Program *P-LAP*(G).

We generate instance groups with parameters Card $(X) = 20, 50, 80, 100$. Such sizes may look small, but they allow us to get exact $GB^*(G)$ values through application of the CPLEX 12 library to the above LAP ILP model. Each instance group consists of 10 randomly generated instances, with a same number Card(X) of nodes and a same mean interval length L, every instance being defined as a collection X of *end-point-distinct* intervals of the real line obtained through the following procedure ***Generate***:

Generate(L: *Mean Interval Length*)
Fix the number Card(X) of intervals;
For $x = 1, \ldots$Card(X) do

Randomly *Generate*, through uniform distribution random sorting in the interval [0,1], the midst point $\Pi(x)$ of the interval x of X;
Randomly *Generate*, through uniform distribution random sorting in the interval [0,2L], the length $L(x)$ of the interval x of X;
Do in such a way that all the end-points of the resulting intervals be distinct.

TABLE 1. Comparing LB(G) and GB* (G).

Instance	X	Aver. *Card(E)*	Aver. H	*GAP-LB (%)*				
				Aver.	Min	Max	*CPU-LB* (s)	*CPU-LAP* (s)
GR1	20	41.4	24.7	0	0	0	<0.1	0.2
GR2	50	283.8	104.9	0.04	0	0.2	<0.1	69.4
GR3	50	406.0	158.5	0.3	0	1.4	4.2	883.3
GR4	50	606.1	305.3	0.02	0	0.1	10.1	884.8
GR5	50	634.1	305.6	0.04	0	0.2	5.8	670.4
GR6	80	31.3	10.8	0	0	0	0.02	66.4
GR7	80	152.0	108.6	0	0	0	1.6	37.4
GR8	80	300.1	105.5	0.4	0	2.7	0.1	247.9
GR9	80	566.9	205.6	0.01	0	0.09	6.9	1360.9
GR10	100	235.3	75.8	0.2	0	0.7	0.1	178.0
GR11	100	298.3	100.5	0.3	0	2.0	0.1	257.9
GR12	100	305.2	186.7	0.09	0	0.5	6.9	998.5
GR13	100	665.5	302.0	0.4	0	2.7	7.0	1605.4
GR14	100	469.6	164.9	0.2	0	1	0.2	3783.0
GR15	100	702.9	249.0	0.08	0	0.5	0.6	7564.0

Though the number Card(E) of edges of the resulting *interval* graph $G = (X, E)$ is not a parameter of this procedure, we indirectly control it through the parameter L, since the expected value of Card(E) increases with the value of L. This allows us to characterize an instance by its related Card(E) value and by the number H of arcs in the oriented graph induced on X by the *inclusion* order: since *unit interval* graphs, which make LAP easy to handle, are those *interval* graphs for which the induced *inclusion* order $\subset$ is empty, we may consider that H provides us with a kind of distance from graph G to the class of *unit interval graph* and so a kind of difficulty index. For every group, we compute the average, min and max gap $GAP\text{-}LB = (GB^*(G) - LB(G))/GB^*(G)$, as well as related average running times *CPU-LAP* and *CPU-LB*. Related results come in the following Table 1.

Comment: Table 1 makes appear that $LB(G)$ coincides very often with optimal value $GB^*(G)$. It is important to notice that, while the max value of *GAP-LB* may vary in a significant way depending on the instance group (from 0% to 2.7% in the above table), these variations cannot be related here to an increase of the size of the instances (values Card(X), Card(E) and H.) Since it would probably be possible to improve $P\text{-}LAP(G)$ and $P\text{-}MC(x)$ ILP formulations, CPU times values have to be taken here as mere additional information: *CPU-LB* happens here to be small in comparison *CPU-LAP*; still, it tends to increase fast with the size of G. As a matter of fact, implementing a branch and bound algorithm for LAP which involves the $LB(G)$ lower bound would require designing an efficient *ad hoc* algorithm for the computation of every quantity $MC(x)$. This last point remains an open question.

3.3. Discussing the complexity of the LB(G) computation

As far as we know, the complexity of the *Unit Cost Max-Cut* problem defined on the complementary graph of an *interva*l graph is still unknown, even if this graph is a *unit interval* graph. That means that we are not able to tell whether computing the $MC(x)$ quantities can be done in polynomial time or not, and this makes one ask himself whether computing $MC(x)$ quantities is really simpler than solving LAP. It is not easy to answer such a question, though above experiments make appear that, because ILP models related to $MC(x)$ are simpler and smaller than LAP ILP models, CPU times required for the computation of LB are small in comparison to CPU times required to full resolution of our LAP model.

In order to provide us with a deeper insight, let us first consider the case of *unit interval graphs.* If $G = (X, E)$ is a *unit interval* graph, and if $x \in X$, then it happens that any interval y in $N(x)$ intersects either $o(x)$

or $d(x)$. A consequence is that the complementary graph of the sub-graph of G induced by $N(x)$ is a bipartite graph, which makes the related *Unit Cost Max-Cut* problem become trivial.

If $G = (X, E)$ is a general *interval* graph, then we are going to check here that solving *Unit Cost Max-Cut* on the complementary graph of G may be done through a dynamic programming algorithm, which, in case we restrict ourselves to graphs with bounded maximal cliques, becomes time-polynomial. In order to do it, we suppose that the elements of X are labeled $X = \{x_1, \ldots, x_n\}$ in such a way that $o(x_i) < o(x_{i+1})$. Then we build an oriented time/state oriented acyclic graph $H = (Z, T)$ in such a way that any partition $X = A \cup^{\mathrm{Ex}} B$, which may be viewed as a succession of decisions: *assign x_i to A or to B*, also appears as a path in H from some initial node to some final node. We do it by setting:

- $S_i = \{j < i \text{ such that } o(x_i) < d(x_j)\} \cup \{j\} = \{ j \leqslant i \text{ such that } x_i = x_j \text{ or } (x_i, x_j) \in E\}$;
- U_i = Set of all S_i indexed $\{0, 1\}$ -valued vectors; we denote by $\mathbf{0}$ the null vector;
- V_i = Set of all 3-uple (p, q, u), where $u \in U_i$, and p, q are non negative integral numbers such that $p + q + \mathrm{Card}(S_i) = i$.

Then digraph H comes as follows:

- Its node set Z is the set of all pairs $(i = 1,\ldots n, v \in V_i)$, augmented with a special node *End*;
- Its arc set T is defined as the set all pairs:
  - $((n, v), End)$: the *length* of such an arc is equal to 0;
  - $((i, (p, q, u)), (i + 1, (p^*, q^*, u^*)))$, such that:
    - for any $j \in S_i \cap S_{i+1}$, we have $u_j = u^*_j$;
    - $p^* = p + \mathrm{Card}(\{j \in S_i - S_{i+1} \text{ such that } u_j = 0\})$; $q^* = i + 1 - p^* - \mathrm{Card}(S_{i+1})$.

In case $u_{i+1^*} = 1$, then the *length* of such an arc is equal to p^* else it is equal to q^*.

We may state:

Proposition 3.8. *Solving Unit Cost Max-Cut on the complementary graph of G is equivalent to computing a largest path from the node (1,(**0**, 0, 0)) to the node End in the digraph H.*

Proof. It comes in a straightforward way from the construction of H. Node set Z represents the point of view of a decider which scans the set $X = \{x_1, \ldots, x_n\}$ and decides, at every step i, to assign x_{i+1} a value 0 (means x_{i+1} is put into A) or 1 (x_{i+1} is put into B). The state in which this decider finds himself before taking this decision is summarized by a 3-uple $(p, q, u) \in V_i$, whose meaning is:

- p intervals x_j such that $x_j \ll x_k$ for any $k \geqslant i$ have been assigned the value 0;
- $q$among those intervals have been assigned the value 1;
- Assignment vector u provides us, for every $j \leqslant i$ such that x_j may eventually be adjacent or equal to some x_k, $k \geqslant i$, with the value which have been assigned to interval x_j.

Node $(1, (\mathbf{0}, 0, 0))$ represents the initial state (step 1): x_1 is arbitrarily assigned the value 0.

Then arcs and their respective lengths express the transitions which take place when some value 0 or 1 is assigned to x_{i+1}, together with the resulting increase of the value $\rho\ (A, B)$. According to this standard dynamic programming scheme, one sees that any partition $X = A \cup^{\mathrm{Ex}} B$ may be understood as a path Γ from initial node $(1, (\mathbf{0}, 0, 0))$ to final node *End* in this acyclic oriented graph H, with value $\rho\ (A, B)$ equal to the length of Γ. We conclude. □

Proposition 3.9. *Let K be a given integral number. If we restrict ourselves to interval graphs G = (X, E) which do not admit cliques with more than K nodes, then Unit Cost Max-Cut can be solved in polynomial time on the complementary graph of G.*

Proof. It is only a matter of noticing that, since, for any i in the above algorithmic scheme, S_i defines a clique for the graph G, then the size of U_i never exceeds K. This implies that the number of nodes of the digraph H does not exceed $n.\ (2^K + 2)$ and allows us to conclude. □

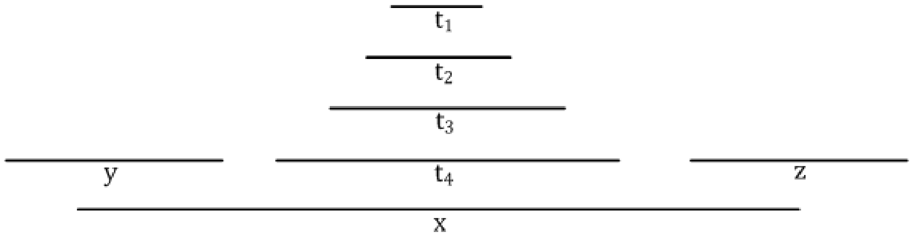

FIGURE 7. Non $\ll$-consistency of an optimal LAP solution.

4. An upper bound: The PCLAP restriction of LAP

In this Section 4, we are going to prove that, if we impose the linear ordering σ to be *precedence consistent*, then the resulting restriction of LAP may be solved in polynomial time. Since the *canonical* ordering σ-*can* is *precedence consistent*, this will provide us with a an upper bound for LAP which will improve the upper bound related to σ-*can*.

The *canonical* linear ordering σ-*can* is not optimal in the case of general *interval* graphs. As a matter of fact, optimal solution may even not be *precedence consistent* as illustrated by Figure 7 below, which shows an interval collection X whose related *interval* graph $G = (X, E)$ is such that optimal $GB^*(G)$ value has to be obtained through a non *precedence consistent* linear ordering σ: Ordering X in an optimal way requires here either locating y and z before x and $t_1 \dots t_4$, or the converse, and yields an optimal $GB^*(G)$ value equal to 11; conversely, any *precedence consistent* ordering σ first considers node y, next all the nodes which are contained into the clique $\{x, t_1 \dots t_4\}$, and finally node z, and yields a *Global Break* value $GB(G, \sigma)$ equal to 14.

Still, the above negative example is not fully representative of what happens if we impose the ordering σ to be *precedence consistent*. As a matter of fact, one may check that in many cases, it is possible to find an ordering σ of the node set X which is *precedence consistent* and whose *GB* value is either optimal or close to optimality. At the same time, one may ask whether imposing σ to be *precedence consistent* is not going to make LAP easier to solve. In order to stress the fact that we focus here on precedence consistent linear orderings and to avoid confusion with the general case, we denote by $PCGB(G, \sigma)$ the *Global Break* value induced by some *precedence consistent* linear ordering σ, and this leads us to define the following LAP restriction PCLAP: *Precedence Consistent Linear Ordering Problem*:

PCLAP: *Compute a precedence consistent linear ordering* σ *of* X *which minimizes* $PCGB(G, \sigma)$.

We denote by $PCGB^*(G) = Inf_{\sigma \in PC(X)}\ PCGB(G, \sigma)$ the optimal (minimal*) Global Break* value, taken on the set $PC(X)$ of all *precedence consistent* linear orderings σ.

The purpose of this section is to study PCLAP and show (Thm. 4.10) that PCLAP can be solved in an exact way in polynomial time. Before entering into the technical details, let us start by providing

the intuition behind this result and what is going to be process which will lead to main Theorem 4.10.

***PCGB*(G)* polynomial time computation (Thm. 4.10): sketch of the proof**. In order to show that $PCGB^*(G)$ can be computed in polynomial time, we are first going to revisit Theorem 1 in the case of *precedence consistent* linear orderings (Lem. 4.1), and check that, in this case, a *precedence consistent* version of *Unit Cost Max-Cut* can be solved in polynomial time (Lem. 4.2). Next (Lem. 4.4), we shall explain the way an *ad hoc* local resolution of *Unit Cost Max-Cut* will tell us, for every node x of our interval graph $G = (X, E)$, how to distribute the nodes $y \in N(x)$ such that Not $(x \subset y)$ before and after x, in order to compute in polynomial time a binary relation σ-*bal* which will be proved to define a *precedence consistent* linear ordering of X (Lems. 4.7 and 4.9). Finally, we shall use the fact that this distribution process locally implements an optimal solution of the *precedence consistent* version of *Unit Cost Max-Cut* in order to show, through a counting argument, that σ-*bal* achieves the lower bound of Lemma 4.1 and so is an optimal solution of PCLAP.

4.1. A lower bound for PCLAP: revisiting Theorem 2.6

Let us define the *non-dominant neighbor* set $ND(x)$ of a node x as the following subset of the neighbour set $N(x)$: $ND(x) = \{y \in N(x)$ such that Not $(x \subset y)\}$.

Then we say, for every $x \in X$, that a partition $Z = A \cup^{\text{Ex}} B$ of $ND(x)$ is *precedence consistent* if, for any y, $z \in ND(x)$ such that: $y \in A$ and $z \ll y$, then we also have $z \in A$. We denote by $MCC(x)$ the largest possible value $\rho(A, B)$, taken for all possible *precedence consistent* partition $Z = A \cup^{\text{Ex}} B$ of $ND(x)$. This allows us to state:

Lemma 4.1. *$PCGB^*(G) \geqslant Tr(G) + \Sigma_x(m(x) - MCC(x))$.*

Proof. We may first notice that the value $m(x)$, which has been defined in the previous section as the number of *anti-edges* in the sub-graph of G induced by $N(x)$, is also equal to the number of *anti-edges* in the sub-graph of G induced by $ND(x)$, due to the fact that no y such that $x \subset y$ is extremity of an *anti-edge* $[y, z] \in E^c$, such that $z \in N(x)$.

Let us consider now some *precedence consistent* linear ordering σ, together with an edge $e = (x, y) \in E$ and a node z, such that $EB(e, z, \sigma) = 1$: $((x, y), z)$ cannot be an *anti-fork* with root z, since *precedence consistency* of σ would impose x and y to be on the *same side with respect* to z. So we get, by following the proof of Lemma 2.2, that $PCGB(G, \sigma) = Tr(G) + Fk(G, \sigma)$.

Moreover, for any $x \in X$, we may derive from σ a *precedence consistent* partition $Z = A \cup^{\text{Ex}} B$ of $ND(x)$: $A = A(x,\sigma) = \{y \in ND(x)$, such that $y\ \sigma\ x\}$; $B = B(x, \sigma) = \{y \in ND(x)$, such that $x\ \sigma\ y\}$.

Let us now recall that $Fk(G, \sigma, x)$ was defined in 2.5 as the quantity Card($\{[y, z] \in E^c$ such that y, $z \in ND(x)$ are on the *same side with respect* to x according to $\sigma\}$) and that $Fk(G, \sigma) = \Sigma_x\ Fk(G, \sigma, x)$.

It comes that, for any node x: $Fk(G, \sigma, x) = m(x) - \rho(A(x, \sigma), B(x, \sigma))$.

Since $\rho(A(x, \sigma), B(x, \sigma)) \leqslant MCC(x)$, we get by summation that $PCGB(G, \sigma) = Tr(G) + Fk(G, \sigma) = Tr(G) + \Sigma_x\ Fk(G, \sigma, x) = Tr(G) + \Sigma_x\ (m(x) - \rho(A(x, \sigma), B(x, \sigma))) \geqslant Tr(G) + \Sigma_x\ (m(x) - MCC(x))$.

We conclude. ☐

4.2. Computing the Values MCC(x): dealing with Unit Cost Max-Cut in the case of precedence consistent partitions

As previously mentioned, the complexity of the *Unit Cost Max-Cut* problem defined on the complementary graph of an *interva*l graph is still unknown. But, conversely, we claim that $MCC(x)$ quantities may be obtained through straightforward application of a simple formula. In order to state this in a formal way, let us consider some node x of the graph $G = (X, E)$ and set, for every node z in $ND(x)$:

- $d^L(x, z) = \text{Card}(\{t \in ND(x)$ such that $t \ll z\})$;
- $d^R(x, z) = \text{Card}(\{t \in ND(x)$ such that $z \ll t\})$.

Explanation: In the above definition, 'L' holds for *Left* and 'R' for *Righ*t. The quantity $d^L(x, z) + d^R(x, z)$ provides us with the number of *anti-edges* of $ND(x)$ which involve node z. If z is located on the side A of a *precedence consistent* partition $ND(x) = A \cup^{\text{Ex}} B$, then we see that the *anti-edges* related to $d^L(x, z)$ are not going to be involved into $\rho(A, B)$ and that if it is located on side B, then the *anti-edges* related to $d^L(x, z)$ are not going to be involved into $\rho(A, B)$. So, intuitively, comparing $d^L(x, z)$ and $d^R(x, z)$ values provides us with information about the way we should distribute nodes z between A and Bin order to achieve the $MCC(x)$ value.

As a matter of fact, we may turn this intuition into the following Lemma 4.2:

Lemma 4.2. $MCC(x) = m(x) - \Sigma_{z \in ND(x)} Inf(d^L(x, z), d^R(x, z))$.

Proof. The idea of the proof is to first prove, through a simple counting argument, that for any *precedence consistent* partition $A \cup^{\text{Ex}} B$ of $ND(x)$, its *cut-size* value $\rho(A, B)$ does not exceed the quantity $m(x) - \Sigma_{z \in ND(x)}$

$\mathrm{Inf}(d^L(x, z), d^R(x, z))$, and next to make appear a *precedence consistent* partition $A^* \cup^{\mathrm{Ex}} B^*$ of $ND(x)$ which achieves this upper bound.

So let us first consider some *precedence consistent* partition $A \cup^{\mathrm{Ex}} B$ of $ND(x)$. It defines an oriented graph structure $(ND(x), K_{A,B})$ on the node set $ND(x)$, whose arc set $K_{A,B}$ is defined as follows: $(y, z) \in K_{A,B}$ if, and only if, $(y \in A, z \in A, y \ll z)$ or $(y \in B, z \in B, y \ll z)$. Since $A \cup^{\mathrm{Ex}} B$is *precedence consistent*, the number of arcs in this digraph with extremity equal to some given node $z \in A$ is equal to $d^L(x, z)$. It comes that the number of arcs of $K_{A,B}$ with both extremities in A, which is also equal to the number of *anti-edges* of $ND(x)$ with both extremities in A, is equal to $\Sigma_{z\in A} d^L(x, z)$. By the same way, the number of *anti-edges* of $ND(x)$ with both extremities in B is equal to $\Sigma_{z\in B} d^R(x, z)$. We get that $\mathrm{Card}(K_{A,B}) = \Sigma_{z\in A} d^L(x, z) + \Sigma_{z\in B} d^R(x, z) \geqslant \Sigma_{z\in ND(x)} \mathrm{Inf}(d^L(x, z), d^R(x, z))$ and that:

$$\rho(A,B) = m(x) - \mathrm{Card}(K_{A,B}) \leqslant m(x) - \Sigma_{z\in ND(x)} \mathrm{Inf}(d^L(x, z), d^R(x, z)).$$

We deduce $MCC(x) \leqslant m(x) - \mathrm{Card}(K_{A,B}) \leqslant m(x) - \Sigma_{z\in ND(x)} \mathrm{Inf}(d^L(x, z), d^R(x, z))$.

In order to achieve our proof, let us consider now the following partition $A^* \cup^{\mathrm{Ex}} B^*$ of $ND(x)$:

$$A^* = \{z \text{ such that } d^L(x, z) \leqslant d^R(x, z)\} \ ; \ B^* = \{z \text{ such that } d^L(x, z) > d^R(x, z)\} \ .$$

This partition is *precedence consistent*: if for instance $z \in A^*$ (we proceed the same way if $z \in B^*$) and $t \ll z$ then we also have $d^L(x, t) \leqslant d^L(x, z) \leqslant d^R(x, z) \leqslant d^R(x, t)$, and so $t \in A^*$.

Since, for any z in A^*, t in B^*, $d^L(x, z) = \mathrm{Inf}(d^L(x, z), d^R(x, z))$ and $d^R(x, t) = \mathrm{Inf}(d^L(x, t), d^R(x, t))$ hold, the above computation yields $\rho(A^*, B^*) = m(x) - \mathrm{Card}(K_{A*,B*}) = m(x) - \Sigma_{z\in ND(x)} \mathrm{Inf}(d^L(x, z), d^R(x, z))$. We get that $A^* \cup^{\mathrm{Ex}} B^*$ achieves the $MCC(x)$ upper bound $m(x) - \Sigma_{z\in ND(x)} \mathrm{Inf}(d^L(x, z), d^R(x, z))$ and conclude. $\square$

Corollary 4.3. *Let F(x) be the anti-edge set* $\{[y, z] \in E^c$ *such that* $y, z \in ND(x)\}$ *, and let* $F_0(x)$ *be the subset of F(x) defined by* $F_0(x) = \{[y, z] \in F(x)$*, such that* $y \in A_0(x)$ *and* $z \in B_0(x)\}$ *. Then m(x) - MCC(x) is at most equal to Card(F(x) -* $F_0(x))/2$*.*

Proof. Since $m(x) - MCC(x) = \Sigma_{y\in ND(x)} \mathrm{Inf}(d^L(x,y), d^R(x, y))$, we must prove that:

$$2.(\Sigma_{y\in ND(x)} \mathrm{Inf}(d^L(x,y), d^R(x,y)) \leqslant \mathrm{Card}(F(x) - F_0(x)).$$

We proceed by induction on $\mathrm{Card}(ND(x) - (A_0(x) \cup B_0(x)))$, and, in case this cardinality remains fixed, on the number of edges in the sub-graph of G induced by $ND(x)$).

We first set $A = \{y \in ND(x) - (A_0(x) \cup B_0(x))$ such that $d^L(x,y) < d^R(x, y)\}$, $B = \{y \in ND(x) - (A_0(x) \cup B_0(x))$ such that $d^L(x,y) > d^R(x,y)\}$ and $C = \{y \in ND(x) - (A_0(x) \cup B_0(x))$ such that $d^L(x,y) = d^R(x, y)\}$. We notice that C is a clique of G.

Then we see that if $y \in C$, then withdrawing y from $ND(x)$ does not modify the values $\mathrm{Inf}(d^L(x,z), d^R(x, z))$ for the other elements z of $ND(x)$, and makes $\mathrm{Card}(F(x) - F_0(x))$ by $2.d^L(x,y) = 2d^R(x, y)$ decrease. It comes that we may suppose C to be empty.

If $y = [o(y), d(y)] \in A$ is such that there exists $z \in ND(x)$ such that $o(y) < d(z) < d(y)$, then we denote by $z(y)$ the argument for the smallest related $d(z)$ value and make $o(y)$ increase until becoming larger than $d(z(y))$. By doing this, we make $\mathrm{Inf}(d^L(x,y), d^R(x, y))$ increase by 1, as well as $\mathrm{Card}(F(x) - F_0(x))$. Since $z(y)$ must be either in A or in $A_0(x)$, $\mathrm{Inf}(d^L(x, z(y)), d^R(x, z(y)))$ remains unchanged, as well as all the other values $\mathrm{Inf}(d^L(x,z), d^R(x, z))$, $z \in ND(x) - \{y\}$. Then applying the induction hypothesis to the resulting interval collection allows us to conclude. Of course, we may proceed the same way with any element of B, while making the $d(y)$ values decrease. It comes that we may suppose that the restriction of $\ll$ to $A \cup B$ is a linear ordering such that for any $y \in A$, $z \in B$, we have $y \ll z$. As a matter of fact, we may set:

- $A = \{a_1, \ldots, a_p\}$ with $a_1 \ll \ldots \ll a_p$;
- $B = \{b_1, \ldots, b_q\}$ with $a_p \ll b_q \ll \ldots \ll b_1$.

Then, if $y \in A_0(x)$ does not intersect b_q and if $a_i \in A$ exists which intersects y, then we choose i maximal with this property and make $d(y)$ decrease until it becomes smaller than $o(a_i)$. By doing this, we make increase by 1 both Card$(F(x) - F_0(x))$ and $\Sigma_{z \in ND(G,x)}$ Inf$(d^L(x,z),\ d^R(x,\ z))$, which allows us to apply the induction hypothesis and conclude. Of course, we proceed the same way with $B_0(x)$, and so we may suppose that:

- For any $y \in A_0(x)$ we have either $y \ll a_1$ or y intersects b_q;
- For any $y \in B_0(x)$ we have either $y \ll b_1$ or y intersects a_p.

Let us set $r = $ Card$(\{y \in A_0(x)$ such that $y \ll a_1\}$ and $t = $ Card$(\{y \in B_0(x)$ such that $y >> b_1\}$. Then we see that:

- $q + t > r + p - 1 and q + t - 1 < r + p$ which also means $q + t = r + p$;
- $2.\Sigma_{z \in ND(G,x)}$Inf$(d^L(x,z),\ d^R(x,\ z)) = p.(p$-1) $+q.(q$-1) + $2.r.p$+ $2.t.q$;
- Card$(F(x) - F_0(x)) \geqslant (p.(p$-1) $+q.(q$-1) + $2.r.p$+ $2.t.q)/2$ $+p.q + p.t + r.q$.

At this point, it only the matter of a routine computation to check that the relation $q + t = r + p$ implies that $p.q + p.t + r.q \geqslant (p.(p$-1) $+q.(q$-1) + $2.r.p$+ $2.t.q)/2$ and the result. □

4.3. Solving PCLAP in an exact way

In this section, we prove that the lower bound of above Lemma 4.1 is equal to $PCGB^*(G)$. The main idea is to build an optimal PCLAP solution σ by first computing, for any node x, a *precedence consistent* partition $A \cup^{\text{Ex}} B$ of $ND(x)$ which maximizes $\rho(A,\ B)$, and by next setting $y\ \sigma\ x$ for any node y in A (locating $y$before x) and $x\ \sigma\ z$ for any node z in B (locating z after x). But then we face the risk that this way of defining σ induces a binary relation which is not a linear ordering. In order to deal with this issue, we first strengthen the notion of *precedence consistent* partition, and impose the *precedence consistent* partition $A \cup^{\text{Ex}} B$ of $ND(G,\ x)$ to be what we call a *strong precedence consistent* partition, that means to be such that:

- $\rho(A,\ B) = MCC(x)$;
- A contains $A_0(x) = \{y \in X$ such $y\ \ Ov\ \ x\}$;
- B contains $B_0(x) = \{y \in X$ such $x\ \ Ov\ \ y\}$;
- A is maximal, for the inclusion order, provided the 3 above requirements are satisfied.

Computing such a *strong precedence consistent* partition may be done according to following Lemma 3.4:

Lemma 4.4. *We get a strong $\ll$-consistent partition $A^*(x) \cup^{Ex} B^*(x)$ of $ND(x)$ by setting:*

- *$A^*(x) = \{z \in\ \ ND(x) - B_0(x)$ such that $d^L(x,\ z) \leqslant d^R(x,\ z)\}$;*
- *$B^*(x) = \{z \in ND(x)$ such that $d^L(x,\ z) > d^R(x,\ z)\} \cup B_0(x)$.*

This strong $\ll$-consistent partition is unique.

Proof. $A^*(x)$ and $B^*(x)$ above clearly define a *precedence consistent* partition such that $A_0(x) \subseteq A^*(x)$ and $B_0(x) \subseteq B^*(x)$. By following the same argument as in Lemma 3.2 and taking into account that, for any $y \in A_0(x)$ $(B_0(x))$ we have $d^L(x,\ z) = 0$ $(d^R(x,\ z) = 0)$, we see that:

$$\text{Card}(K_{A*(x),B*(x)}) = \Sigma_{z \in ND(x)}\ \text{Inf}(d^L(x,\ z),\ d^R(x,\ z)),$$

where $(K_{A*(x),B*(x)})$ is the arc set defined on $ND(x)$ by: $(y,\ z) \in K_{A*(x),B*(x)}$ if and only if:

$$(y \in A^*(x),\ z \in A^*(x),\ y \ll z) \text{ or } (y \in B^*(x),\ z \in B^*(x),\ y \ll z).$$

It comes that: $\rho(A^*(x),\ B^*(x)) = m(x) - \Sigma_{z \in ND(x)}$ Inf$(d^L(x,\ z),\ d^R(x,\ z)) = MCC(x)$.
Conversely, if $(A,\ B)$ is any *precedence consistent* partition, the relation

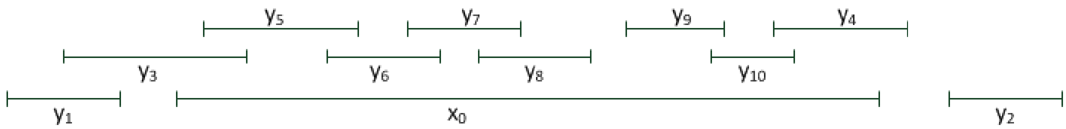

FIGURE 8. Distribution of the nodes of $X - x_{\mathbf{0}}$ with respect to $x_{\mathbf{0}}$ according to σ-bal.

$$\text{Card }(K_{A,B}) = \Sigma_{z\in A} d^L(x,z) + \Sigma_{z\in B} d^R(x,z)$$

which was involved in the proof of Lemma 3.2, also implies that $\rho(A,\ B) = MCC(x)$ holds if and only if all nodes z such that $d^L(x,\ z) < d^R(x,\ z)$ are in A and all nodes z such that $d^L(x,\ z) > d^R(x,\ z)$ are in B. Then any *precedence consistent* partition $A \cup^{\text{Ex}} B$ of $ND(x)$ such that $\rho(A,\ B) = MCC(x)$, $A_0(x) \subseteq A$ and $B_0(x) \subseteq B$, is also such that $A \subseteq A^*(x)$.We conclude. $\square$

Algorithmic definition of the *well-balanced σ-bal linear ordering.*

We assume here that we have been pre-processing the interval collection X in order to get:

- a $\{0,1\}$ -valued array AR, with indexation on $X.X$, such that $AR[x,\ y]$ means $((x\ Ov\ y) \vee (x \ll y))$;
- a vector R(as *Right*), which provides us, for any $x \in X$, with the number of nodes y such that $x \ll y$;
- a vector L (as *Left*), which provides, for any $x \in X$, with the number of nodes y such that $y \ll x$;
- an array $LIST_{\subset}$, which provides us, for any $x \in X$, with the list of nodes y such that $y \subset x$.

We denote by H the number of arcs of the oriented graph induced on set X by the $\subset$ ordering. This number H, which we already used in Section 4.2 as a kind of measure of the distance which may exist between graph G and the *unit interval* graph class, is going to be involved into the computation of the complexity of the *PCLAP* Algorithm below (Lem. 4.5). Notice that this number is not really involved in the algorithm itself.

Then we may provide an algorithmic definition of what we call the *well-balanced binary relation* associated with the interval collection X, and which we denote by σ-*bal*. This *well-balanced binary relation* σ-*bal* is defined as a complete extension of the $\ll$ and Ov orderings. That means that we consider that, for any pair x, y such that $x \ll y$ or $x\ Ov\ y$, we *a priori* have (preprocess) x σ-*bal* y (and consequently *Not* x σ-*bal* y). Then the way we decide to order nodes x, y of the interval graph $G = (X,\ E)$ in case $x \subset y$ or $y \subset x$ derives from application of the following *PCLAP* algorithm:

PCLAP Algorithm

Input: AR, R, L, $LIST_{\subset}$
Output: the relation x σ-*bal* y which extends AR when $x \subset y$ or $y \subset x$.
For $x \in X$ do
  For $y \in LIST_{\subset}[x]$ do
    If $R[y] - R[x] \leqslant L[y] - L[x]$ then set y σ-*bal* x else set x σ-*bal* y;

Figure 8 below shows an interval collection X, an interval x_0, and the way the nodes of $X - x_0$ are distributed before and after x_0 according to the σ-*bal* relation.

- Preprocess:

$y_1 \ll x_0 \ll y_2 =>$ y_1 σ-*bal* x_0 σ-*bal* y_2 ; y_3 Ov x_0 Ov y_4 $=>y_3$ σ-*bal* x_0 σ-*bal* y_4;
+ (action of the PCLAP Algorithm) :
$d^L(x_0, y_5) = 0$ and $d^{\mathrm{R}}(x_0, y_5) = 5$ $=>y_5$ σ-*bal* x_0 ; $d^L(x_0, y_6) = 1$ and $d^{\mathrm{R}}(x_0, y_6) = 4$ $=>y_6$ σ-*bal* x_0;
$d^L(x_0, y_7) = 2$ and $d^{\mathrm{R}}(x_0, y_7) = 3$ $=>y_7$ σ-*bal* x_0 ; $d^L(x_0, y_8) = 3$ and $d^{\mathrm{R}}(x_0, y_8) = 3$ $=>y_8$ σ-*bal* x_0;
$d^L(x_0, y_9) = 5$ and $d^{\mathrm{R}}(x_0, y_9) = 1$ $=>x_0$ σ-*bal* y_9 ; $d^L(x_0, y_{10}) = 5$ and $d^{\mathrm{R}}(x_0, y_{10}) = 0$ $=>x_0$ σ-*bal* y

Lemma 4.5 (PCLAP Algorithm's Complexity)**.** *The complexity of the PCLAP algorithm is $O(H)$.*

Proof. The double loop of $PCLAP$ is indexed on the number of relations $y \subset x$, x, $y \in X$, and any iteration inside this loop requires only a comparison and an assignment. □

Remark 4.6. The above $PCLAP$ algorithm only computes σ-*bal* for pairs x, y such that we have either $x \subset y$ or $y \subset x$. In case $x \ll y$ or x Ov y, it is an *a priori* decision which set x σ-*bal* y. But this point is not taken into account in the evaluation of the complexity of CLAP, since it is considered as part of the preprocess which yields the input of $PCLAP$ (the AR array).

At this point, we know that σ-*bal* is a complete extension of the $\ll$ and Ov partial orderings, but we still do not know whether it is a linear ordering and less whether it is an optimal PCLAP solution or not. In order to check that σ-*bal* is a linear ordering, we first state the following Lemma 3.7, which could have been used as well as a definition of the σ-*bal* relation, and which is only a kind of translation of the algorithmic definition induced by the $PCLAP$ algorithm.

Lemma 4.7. *The above defined binary relation $\sigma-$ bal is such that x σ-bal y if and only if one among the following relations, which are mutually exclusive, holds*:

(E2): $(x \ll y)$ or $(x$ Ov $y)$,

or

(E3): $(x \subset y)$ and $d^L(y, x) \leqslant d^R(y, x)$,

or

(E4): $(y \subset x)$ and $d^R(x, y) < d^L(x, y)$.

Proof. Configurations (E2), (E3) and (E4) are clearly mutually exclusive. Given x, y in X, we have either:

(E2*): ($x \ll y$ or x Ov y) or ($y \ll x$ or y Ov x),

or

(E3*): $(x \subset y)$

or

(E4*): $(y \subset x)$.

Configurations (E2*), (E3*), (E4*) are also mutually exclusive.

If (E2*) holds, then σ-*bal* coincides with the relation $AR = (\ll$ or $Ov)$, which means that $x\sigma$-*bal* y if and only if $AR[x, y] = 1$, or, in other words, if and only if (E2) holds.

If (E3*) holds then $x \in LIST_{\subset}[y]$ and the $CLAP$ algorithm sets x σ-*bal* y if and only if:

$$d^L(y,x) = R[x] - R[y] \leqslant L[x] - L[y] = d^R(y,x),$$

which also means $d^L(y,x) \leqslant d^R(y,x)$.

If (E4*) holds then $y \in LIST_{\subset}[x]$ and the $CLAP$ algorithm sets x σ-*bal* y if and only if:

$$d^L(x,y) = R[y] - R[x] > L[y] - L[x] = d^R(x,y),$$

which also means $d^R(x,y) < d^L(x,y)$.We conclude. □

Corollary 4.8. *Given $x \in X$. If we set $A^*(x) = \{y \in ND(x)$ such that $y\sigma$-bal $x\}$ and $B^*(x) = \{y \in ND(x)$ such that x σ-bal $y\}$ then we get a strong precedence consistent partition $A^*(x) \cup^{Ex} B^*(x)$ of $ND(x)$.*

Proof. It is a mere translation of Lemma 4.4 and above Lemma 4.7.

Let us now check that σ-*bal* is a *precedence consistent* linear ordering:

Lemma 4.9. *The σ-bal relation computed through the PCLAP algorithm is anti-symmetric, transitive and precedence consistent.*

Proof. The anti-symmetry of σ-*bal* comes from its mere definition.

As for transitivity, let us consider x, y, z such that x σ-*bal* y σ-*bal* z. We have to check that x σ-*bal* z. Non trivial configurations correspond to the following three cases (modulo symmetry):

- **Case 1.** $x \subset y \subset z$
  We know (Lem. 4.7) that $d^L(y, x) \leqslant d^R(y, x)$ and $d^L(z, y) \leqslant d^R(z, y)$. It comes that:
  $d^L(z, x) = d^L(z, y) + d^L(y, x) \leqslant d^R(z, x) = d^R(z, y) + d^R(y, x)$, and that $x\sigma$-*bal* z.
- **Case 2.** $x \subset z \subset y$
  We know (Lem. 4.7) that $d^L(y, x) \leqslant d^R(y, x)$ and $d^R(y, z) < d^L(y, z)$. It comes that:
  $d^L(z, x) = d^L(y, x) - d^L(y, z) \leqslant d^R(y, x) - d^R(y, z) = d^R(z, x)$, and that x σ-*bal* z.
- **Case 3.** y *Ov* z; $x \subset y$; $x \subset z$

We know (Lem. 4.7) that $d^L(y, x) \leqslant d^R(y, x)$. Since y *Ov* z, we also have:
$d^L(z, x) \leqslant d^L(y, x)$ and $d^R(y, x) \leqslant d^R(z, x)$. We deduce $d^L(z, x) \leqslant d^R(z, x)$ and x σ-*bal* z.

So we get the transitivity of σ-*bal*.

As for the *precedence consistency* of σ-*bal*, it comes in a trivial from the fact that σ-*bal* has been *a priori* built as an extension of $\ll$. The result is thus established. ☐

We are now in a position to prove the optimality of σ-*bal*. This optimality will derive from the fact that, for any $x \in X$, σ-*bal* distributes the elements of $ND(x)$ before and after x, in a way which induces a *strong precedence consistent* partition of $ND(x)$ and thus which achieve the $MCC(x)$) value.

Theorem 4.10. *The PCLAP Algorithm computes, in polynomial time, a relation σ-bal which is an optimal PCLAP solution and whose value PCGB*(G) satisfies:*

$$PCGB^*(G) = PCGB(G, \sigma\text{-}bal) = Tr(G) + \Sigma_x\ (m(x) - MCC(x)) = \Sigma_x\ \Sigma_{z\in \mathrm{ND(x)}}\ Inf(d^L(x, z),\ d^R(x, z)).$$

Besides, the following inequality holds:

$$Tr(G) + \Sigma_x \Sigma_{z\in ND(x)}\ Inf(d^L(x, z),\ d^R(x, z)) = PCGB^*(G) \leqslant Tr(G) + Strong\text{-}Fork/2,$$

where Strong-Fork denotes the number of strong forks of the interval graph G.

Explanation. The second part of the above result provides us with a magnitude order of the values of $GB^*(G)$ and $PCGB^*(G)$, and makes the *Strong-Fork* number discriminate the LAP instance defined by G from the time-polynomial *unit interval* graph case.

Proof. We get from Lemma 4.9 that σ-*bal* is a *precedence consistent* linear ordering and from Lemma 6 that the *PCLAP* Algorithm computes it in no more than $O(\mathrm{Card}(X)^2)$ instructions.

The main idea behind the proof is that (see Cor. 4.8), since σ-*bal* distributes, for any x $\in$ X, nodes of $ND(x)$ in a way which defines a *strong precedence consistent* partition $A^*(x) \cup^{\mathrm{Ex}} B^*(x)$, the $MCC(x)$ value is achieved by the *cut-size* $\rho(A^*(x),\ B^*(x))$ of this partition. Then it will be only a matter of summing the related local relations and to use Lemma 4.1 in order to conclude for the first part of the result. The second part will derive in a straightforward way of Corollary 4.3.

So let us enter into details and first prove the optimality of σ-*bal* (and, consequently, the inequality $PCGB(G, \sigma\text{-}bal) \leqslant PCGB(G, \sigma\text{-}can)$). We know (see proof of Lem. 4.1) that $PCGB(G, \sigma\text{-}bal) = Tr(G) + Fk(G, \sigma\text{-}bal)$, and that $Fk(G, \sigma\text{-}bal)$ may also be written: $Fk(G, \sigma\text{-}bal) = \Sigma_x\ Fk(G, \sigma\text{-}bal, x)$. But Lemma 4.4 and Corollary 4.3 tell us that, for any x, $Fk(G, \sigma\text{-}bal, \mathrm{x}) = \mathrm{m(x)}$ - $MCC(x)$. It comes that $PCGB(G, \sigma\text{-}bal) = Tr(G) + \Sigma_x\ (m(x)$ - $MCC(x))$. Then Lemma 4.1 yields $PCGB(G, \sigma\text{-}bal) = PCGB^*(G)$ and Lemma 4.2 tells us that: $PCGB(G, \sigma\text{-}bal) = PCGB^*(G) = Tr(G) + \Sigma_x \Sigma_{y\in ND(x)} \mathrm{Inf}(d^L(x, y),\ d^R(x, y))$.

In order to get the second part of the above theorem, we consider, for any $x \in X$, $F(x)$ and $F_0(x)$ as defined in the statement of Corollary 4.3. This corollary tells us that m(x) - $MCC(x) \leqslant \mathrm{Card}(F(x) - F_0(x))/2. But F(x) - F_0(x)$ may be written as: $F(x) - F_0(x) = \{[y,\ z] \in E^c$ such that $y \ll z$ and at least one node t among $\{y,\ z\}$ is such that $t \subset x\}$. So we see that there is a one-to-one correspondence between $F(x) - F_0(x)$ and the set of *strong forks* with root x. Since a *fork* has only one root, we get that $\Sigma_{x\in X}\ \mathrm{Card}(F(x) - F_0(x)) = Strong\text{-}Fork$, and so we may conclude. ☐

5. Bounding the Gap between $LB(G)$, $PCGB^*(G)$ and $GB^*(G)$

The purpose of this section is to bound the gap between the lower bound $LB(G)$ of Theorem 3.4 and the upper bound $PCGB^*(G)$.

Sketch of the approach: Since $PCGB^*(G) = Tr(G) + \Sigma_x(m(x) - MCC(x))$, we first bound the gap between $MC(x)$ and $MCC(x)$ values (Lem. 5.1). We do it by applying a transformation *Reduce* to some interval collection $ND(x)$, checking that the gap is null for this *Reduce*(x) interval collection (Lems. 5.2 and 5.3), and bounding the gap induced by reversing the *Reduce* transformation (Lem. 5.1). Next, we consider the interval collection X as a whole, and derive from our local reasoning about $MC(x)$ and $MCC(x)$ values a global bound for the gap between $LB(G)$ and $PCGB^*(G)$.

5.1. Bounding the gap between MC(x) and MCC(x) values

We consider an *interval* graph $G = (X, E)$, some node x, the set $ND(x)$, and set:

- $E_1(x) = \{(y, z) \in E$, such that y and $z \in ND(x) - (A_0(x) \cup B_0(x))\}$;
- $E_2(x) = \{(y, z) \in$ E, such that $y \in A_0(x) \cup B_0(x)$ and $z \in ND(x) - (A_0(x) \cup B_0(x))\}$.

Explanation. *Precedence consistent* orderings σ of X impose any y in $A_0(x)$ to be such that $y\ \sigma\ x$, and any $z$in $B_0(x)$ to be such that $x\ \sigma\ z$. But the example related to Figure 7 shows that relaxing *precedence consistency* may lead to put elements of $A_0(x)$ and $B_0(x)$ on the same side with respect to x, and put on the other side elements of $ND(x) - (A_0(x) \cup B_0(x))$ which are connected by a small number of *anti-edges*. At the end, this may induce the difference between $PCGB^*(G)$ and $GB^*(G)$. If we refer now to our local distribution process related to some node x and its *non dominant neighbour* set $ND(x)$, this gives rise to the intuition that the difference between $MC(x)$ and $MCC(x)$ values is going to be due to edges of the sub-graph induced by $ND(x)$ which involve at least one node in $ND(x) - (A_0(x) \cup B_0(x))$, that means to the edges in $E_1(x)$ and $E_2(x)$. On another side, taken as a whole, subsets $E_1(x)$ and $E_2(x)$ have to be linked with the *strong triangles*: the sum $\Sigma_{x\in X}$ Card$(E_1(x))$ provides us with the number of *strong triangles* $\{x, y, z\}$, which are such that both y and z are included into x and $\Sigma_{x\in X}$ Card$(E_2(x))$ enumerates all *strong* triangles $\{x, y, z\}$ such that x and y overlap, while eventually counting twice a same *triangle* $\{x, y, z\}$ in the case we simultaneously have $z \subset x$ and $z \subset y$. This last remark will help us in turning local Lemma 5.2 and 5.3 into global Theorem 5.5.

We say that $ND(x)$ is *reduced* if any interval which is not in $A_0(x) \cup B_0(x)$ may be considered as reduced to 1 point, that means if for any y, z in $ND(x) - A_0(x) \cup B_0(x)$ we either have $y \ll z$ or $z \ll y$. We denote by *Reduce*(x) the *reduced* interval collection derived from $ND(x)$ by replacing every y in $ND(x)$ by an interval $u(x)$ according to the following scheme:

- $u(y) = y$ if $y \in A_0(x) \cup B_0(x)$;
  (E5): $u(y)$ is reduced to the interval $[o(y), o(y) + \varepsilon]$, where ε very small, if $y \notin A_0(x) \cup B_0(x)$ and Card$(\{z \in A_0(x)$ such that $(y, z) \in E\}) \geqslant$ Card$(\{z \in B_0(x)$ such that $(y, z) \in E\})$;
  (E5'): $u(y)$ is reduced to the point $[d(y), d(y) + \varepsilon]$, where ε very small, if $y \notin A_0(x) \cup B_0(x)$ and Card$(\{z \in A_0(x)$ such that $(y, z) \in E\}) <$ Card$(\{z \in B_0(x)$ such that $(y, z) \in E\})$;

Figure 9 below shows the way we derive *Reduce*(x) from $ND(x)$.

$ND(x) =>$*Reduce*(x)

Of course, the definitions of $MC(x)$ and $MCC(x)$ apply to *Reduce*(x): to any pair (A, B) of disjoint subsets of $ND(x)$ corresponds in a one-to-one way a pair $(u(A) = \{u(x), x \in A\}$, $u(B) = \{u(x), x \in B\}$) of disjoint subsets of *Reduce*(x), with a $\rho_R(A, B)$ value defined by:

- $\rho_R(A, B) =$ Card$(\{[u(x), u(y)] \in$ *anti-edge* set induced by *Reduce*(x), such that $x \in A$ and $y \in B\})$.

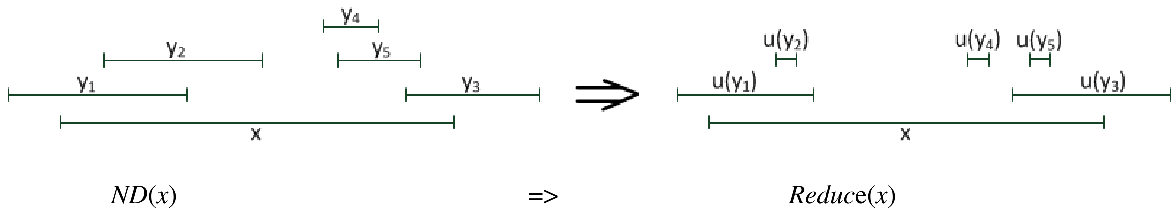

FIGURE 9. Deriving *Reduce*(x) from *ND*(x).

This allows us to set in a natural way:

- $MC_R(x)$ ($MCC_R(x)$) = the value $MC(x)$ ($MCC(x)$) computed while dealing with the *Reduce*(x) interval collection and replacing the values $\rho(A, B)$ by the $\rho_R(A, B)$ values.

Lemma 5.1. *The following inequalities hold*:
(E6): $MC(x) \leqslant MC_R(x) \leqslant MC(x) + Card(E_1(x)) + Card(E_2(x))$;
(E7): $MCC_R(x) \leqslant MCC(x) + Card(E_2(x))/2 + Card(E_1(x))$.

Proof. Any partition $A \cup^{\mathrm{EX}} B$ of $ND(x)$ yields a partition $\{u(x),\ x \in A\} \cup^{\mathrm{Ex}} \{u(x),\ x \in B\}$ of *Reduce*(x), and we have $\rho_R(A, B) \geqslant \rho(A, B)$. We deduce $MC(x) \leqslant MC_{\mathrm{R}}(x)$. If $\{u(x),\ x \in A\} \cup^{\mathrm{Ex}} \{u(x),\ x \in B\}$ is an optimal partition for *Reduce*(x), then the *anti-edges* which are involved in the computation of $\rho_R(A, B)$ and which are not involved in $\rho(A, B)$ correspond to edges of $E_1(x) \cup E_2(x)$ which have been suppressed by the transition from $ND(x)$ to *Reduce*(x). We deduce $MC_{\mathrm{R}}(\mathrm{x}) \leqslant MC(x) + \mathrm{Card}(E_1(x)) + \mathrm{Card}(E_2(x))$ and (E6).

As for (E7), we see that any *precedence consistent* partition $\{u(x),\ x \in A\} \cup^{\mathrm{Ex}} \{u(x),\ x \in B\}$ of *Reduce*(x) yields a *precedence consistent* partition $A \cup^{\mathrm{Ex}} B$ of $ND(x)$. The *anti-edges* which are involved in $\rho_R(A, B) - \rho(A, B)$ correspond either to edges of $E_1(x)$ or to edges of $E_2(x)$. In the case they correspond to edges of $E_2(x)$, (E5) and (E5') tell us that, for any $z$in $ND(x) - A_0(x) - B_0(x)$, no more than half part of the edges of $E_2(x)$ which may be written (y, z), $y \in A_0(x) \cup B_0(x)$ have been suppressed by the transition from $ND(x)$ to *Reduce*(x). We deduce that $\rho_R(A, B) \leqslant \rho(A, B) + \mathrm{Card}(E_1(x)) + \mathrm{Card}(E_2(x))/2$ and (E7). ☐

As a matter of fact, we may identify, for any y in $ND(x) - A_0(x) - B_0(x)$, the interval $u(y)$ of *Reduce*(x) and its end-points $o(u(y))$ and $d(u(y))$. So we consider a *strong precedence consistent* partition (see previous Sect. 4.3) $A^*\cup^{\mathrm{EX}}B^*$ of *Reduce*(x). $A^*\cup^{\mathrm{EX}}B^*$ induces an optimal $\rho_R(A^*, B^*)$ value). Then we say that *Reduce*(x) is *regular* if for any $y \in A^*$ and $z \in B^*$ we have $y \ll z$, and we claim:

Lemma 5.2. *In case Reduce*(x) *is regular then:* $MCC_R(x) = MC_R(x)$.

Proof. As a matter of fact, it is enough to prove that if $ND(x)$ is *reduced*, then $MCC(x) = MC(x)$. Since we are going to apply perturbations to current graph G throughout our reasoning, we use the $ND(G, x)$ notation for the *non-dominant neighbour* set when there is an ambiguity about the related graph. Let us set $n = \mathrm{Card}(ND(x))$ and m = Number of edges in the *interval* graph defined by $ND(x)$. We suppose that $ND(x)$ is *regular* and we proceed by induction on $n + m$. In order to do it, we label $\{y_1, \ldots, y_p, z_q, \ldots z_1\}$ the elements of $ND(x) - A_0(x) - B_0(x)$, in such a way that:

- $u(y_1) \ll \ldots \ll u(y_p) \ll u(z_q) \ll \ldots \ll u(z_1)$;
- $u(y_1), \ldots, u(y_p) \in A^*$; $u(z_1), \ldots, u(z_q) \in B^*$.

If $i \geqslant 1$ and $z \in A^* \cap A_0(x)$ exist such that $u(y_i) \ll z \ll u(y_{i+1})$, then we see that, because $ND(x)$ is *regular*, making $d(z)$ decrease until we get $z \ll u(y_i)$ makes m decrease by 1. Induction hypothesis applies to the resulting interval graph G'. But $A^*\cup^{\mathrm{EX}}B^*$ remains a *strong precedence consistent* partition of $ND(G', x)$ with unchanged $\rho(A^*, B^*)$ value. Since removing an edge of the graph induced by $ND(x)$ cannot make increase the related $MC(x)$ value, we conclude.

So, we may suppose that, for any $z \in A^* \cap A_0(x)$, we either have $z \ll u(y_1)$ or $u(y_p) < d(z)$, and we set: $A_0^{Inf}(x) = \{z \in A_0(x) \text{ such that } z \ll u(y_1)\}$ and $A_0^{Sup}(x) = \{z \in A_0(x) \text{ such that } d(z) > u(y_{\mathrm{p}})\}$. We may of course proceed the same way with B^* and suppose that for any $z \in B^* \cap B_0(x)$, we either have $u(z_1) \ll z$ or $u(z_q) > o(z)$. By the same way, we set: $B_0^{Inf}(x) = \{z \in B_0(x) \text{ such that } z \gg u(z_1)\}$ and $B_0^{Sup}(x) = \{z \in A_0(x)$ such that $u(z_q) > o(z)\}$.

Let us suppose now that $A_0^{Inf}(x)$ is not empty, and let us pick up y in $A_0^{Inf}(x)$ such that $d(y)$ is the largest possible. Replacing y by some interval $[d(y) - \varepsilon, d(y)]$, ε very small, turns the interval graph G into another interval graph G' in such a way that $ND(G', x)$ remains *reduced* and that A^* and B^* keep on defining a strong $\ll$ -consistent partition of $ND(G', x)$. Then we may apply the induction hypothesis and state that related values $MCC_{G'}(x)$ and $MC_{G'}(x)$ are equal. But $MCC_{G'}(x)$ and $MCC_G(x)$ are also equal since the value $\rho(A^*, B^*)$ remains unchanged. Since $MC_{G'}(x) \geqslant MC_G(x) = MC(x)$ we conclude. It comes that we may suppose that $A_0^{Inf}(x)$ is empty, and, by proceeding the same way with $B_0^{Inf}(x)$, that $B_0^{Inf}(x)$ is also empty.

Because of the *strong precedence consistency* of the partition $A^*\cup^{\mathrm{EX}}B^*$, we see that: $p - 1 \leqslant \mathrm{Card}(B_0^{Sup}(x)) + q$ and that : $q - 1 < \mathrm{Card}(A_0^{Sup}(x)) + p$. In case both $A_0^{Sup}(x)$ and $B_0^{Sup}(x)$ are non empty, we see that we must have (E8): $p \leqslant \mathrm{Card}(B_0^{Sup}(x)) + q$ or (E8-1):$q < \mathrm{Card}(A_0^{Sup}(x)) + p$. In case (E8) holds, we remove one element z from $B_0^{Sup}(x)$. Because of (E8), A^* and $B^* - \{z\}$ keep on defining a *strong precedence consistent* partition of the resulting $ND(G', x)$ interval collection. Induction hypothesis applies and yields $MCC_{G'}(x) = MC_{G'}(x)$. But we also have $MCC_G(x) = MCC_{G'}(x) + \mathrm{Card}(A^*)$. Since $\mathrm{Card}(A^*)$ is exactly the number of *anti-edges* with end-point z in $ND(G, x)$, we have $MC(x) = MC_G(x) \leqslant MC_{G'}(x) + \mathrm{Card}(A^*)$. Then we conclude. In case (E8-1) holds, we proceed the same way and see that we may suppose that at least one among $A_0^{Sup}(x)$, $B_0^{Sup}(x)$ is empty.

So we may suppose that both $A_0^{Inf}(x)$ and $B_0^{Inf}(x)$ are empty, and that at least one among $A_0^{Sup}(x)$, $B_0^{Sup}(x)$ (for instance $B_0^{Sup}(x)$) is empty. Then it is an easy matter to check that, if $A \cup^{\mathrm{EX}} B$ is a partition of $ND(x)$, then we may move the elements of $A_0^{Inf}(x)$ in such a way:

- we either have $A_0^{Inf}(x) \subseteq A$ or $A_0^{Inf}(x) \subseteq B$;
- the value $\rho(A, B)$ does not strictly decrease.

We may then rename A and B in such a way that $A_0^{Inf}(x) \subseteq A$. But once it is done, it is also an easy matter to check that if we set $k = \mathrm{Card}(A - A_0^{Inf}(x))$, then replacing in A the elements of $A - A_0^{Inf}(x)$ by the k first elements of $\{u(y_1), \ldots, u(y_p), u(z_q), \ldots, u(z_1)\}$ for the $\ll$ linear ordering does not make the value $\rho(A, B)$ strictly decrease. That means that $MCC(x) =$ and $MC(x)$. We conclude. □

Lemma 5.3. *In any case:*
(E9): *$MCC_R(x) = MC_R(x)$.*

Proof. We proceed by induction on the number of *anti-edges* of the *interval* graph defined by *Reduce*(x), while considering the case when *Reduce*(x) is *regular* as our bottom case (Lem. 4.1). In order to do it, we consider once again a *strong precedence consistent* partition $A^*\cup^{\mathrm{EX}}B^*$ of *Reduce*(x) and set:

- $L = \{\text{End-points } o(u(y)),\, y \in A^* - A_0(x),\, d(z),\, z \in A_0(x)\}$;
- $R = \{\text{End-points } d(u(y)),\, y \in B^* - B_0(x),\, o(z),\, z \in B_0(x)\}$.

In case *Reduce*(x) is not *regular* then there must exist t and t', $t < t$', which are consecutive in $L \cup R$, and such that $t \in R$ and $t' \in L$. Because of the definition of L and R the intersection $\{t, t'\} \cap (A_0(x) \cup B_0(x))$ cannot be empty. Then we see that switching t and t' makes increase by 1 the number of *anti-edges* of the *interval* graph defined by *Reduce*(x). By the same way, the quantity $\rho_R(L, R)$ also increases by 1. We conclude. □

Lemma 5.4. *The following inequalities hold*: $0 \leqslant MC(x) - MCC(x) \leqslant Card(E_1(x)) + Card(E_2(x))/2$.

Proof. The first part of the statement is trivial and directly follows from the definition of the quantities $MC(x)$ and $MCC(x)$. The second one comes in a straightforward way from (E6), (E7) of Lemma 5.1 and (E9) of Lemma 5.3 . □

We are now able to state the main result of this section: let us denote by *Strong-Tr* the number of *strong triangles* (defined in Sect. 2.5).

Theorem 5.5. *Let* $G = (X, E)$ *be an interval graph. Then we have:* $PCGB^*(G) - GB^*(G) \leqslant PCGB^*(G) - LB(G) \leqslant$ *Strong-Tr.*

Remark 5.6. Notice that this approximation result improves the result (see [9]) which states that $GB^*(G, \sigma\text{-}can) \leqslant 2.GB^*(G)+$ Card(E), and which also means that σ-*can* produces a 2-approximation if we refer to the standard definition of LAP.

Proof of Theorem 3. The idea which is behind the proof of Theorem 5.5 is to first turn, through summation local bounds involved in Lemma 5.4 into a global bound and then use what was told at the beginning of Section 5.1 about the link which exists between the subsets $E_1(x)$, $E_2(x)$, $x \in X$, and the *strong triangle* notion.

Theorem 4.10 says that $GB(G, \sigma\text{-}bal) = PCGB*(G) = Tr(G) + \Sigma_x\ (m(x) - MCC(x))$ and Theorem 3.4 says that GB*(G) $\geqslant Tr(G) + \Sigma_x\ (m(x) - MC(x)) = LB(G)$. But, because of Lemma 5.4 , we know that, for any $x \in X$, $MC(x) - MCC(x)$ is non negative and does not exceed Card$(E_1(x))$ + Card$(E_2(x))/2$.

The sum $\Sigma_{x\in X}$ Card$(E_1(x))$ provides us with the number of *strong triangles* $\{x, y, z\}$, which are such that both y and z are included into x. We denote by $Strong_1$ the number of those *strong* triangles. When it comes to $\Sigma_{x\in X}$ Card$(E_2(x))$, we see that it enumerates all *strong* triangles $\{x, y, z\}$ such that x and y overlap, while eventually counting twice a same *triangle* $\{x, y, z\}$ in the case we simultaneously have $z \subset x$ and $z \subset y$. Then, it follows that $\Sigma_{x\in X}$ Card$(E_2(x))/2$ does not exceed the number $Strong_2$ of *strong triangles* which are such that x and y overlap. So we get:

$GB(G, \sigma\text{-}bal)$ - $OPT(G) \leqslant C\text{-}OPT(G) - LB(G) \leqslant \Sigma_x\ (MC(x) - MCC(x)) \leqslant$

Σ_x Card$(E_1(x))$ + Card$(E_2(x))/2) = Strong_1 + \Sigma_x$ Card$(E_2(x))/2 \leqslant Strong_1 + Strong_2 =$ *Strong-Tr*.

Therefore, the result follows. □

Experimental comparison of $GB(G, \sigma\text{-}can)$, $GB^*(G)$ and $PCGB^*(G)$.

We use the same instances as at the end of Section 3, in order to get an evaluation of the gap which may exist between $GB(G, \sigma\text{-}can)$, $PCGB^*(G)$ and $GB^*(G)$:

- *GAP-BAL* is the gap *GAP-BAL*= $(GB(G, \sigma\text{-}bal)$ - $GB^*(G))/\ GB^*(G)$;
- *GAP-can* is the gap *GAP-CAN* = $(GB(G, \sigma\text{-}can)$ - $GB^*(G))/\ GB^*(G)$;
- CPU times never exceed 1 ms (10^{-3} second), for both $GB(G, \sigma\text{-}bal)$ and $GB(G, \sigma\text{-}can)$.

Comment: Above results show that the linear ordering σ-*bal* yields a very good approximation of $GB^*(G)$, since only instance group GR1 yields a maximal *GAP-BAL* value larger than 1%. It seems difficult to correlate the behavior of σ-*bal* to either the size of the instance, (the worst result corresponds here to the smallest size) or to the number $H$of inclusion relation. Conversely, one may see that the practical performance of the intuitive solution σ-*can* is not so good: though it solves the unit case in an exact way and may prevail itself with a worst case 2-approximation ratio 2-approximation if we refer to the standard definition of LAP (see above Rem. 5.6), it yields, in the general case and according to the above table, *GAP-CAN* values close to 30% in the average, and larger than 80% in the worst case. The fact that $LB(G)$ and $PCGB^*(G)$ provide us with respectively lower and upper bounds which are very close in average to the optimal value $GB^*(G)$ opens the way to the design of efficient branch and bound algorithms for LAP, provided we become able to design an efficient way to compute the $MC(x)$ values.

TABLE 2. Comparing $GB^*(G)$, $PCGB^*(G)$ and $GB(G, \sigma\text{-}can)$.

Instance	Card(X)	Card(E)	H	*GAP-CAN%*			*GAP-BAL%*		
				Mean	Min	Max	Mean	Min	Max
GR1	20	41.4	24.7	31.6	5.3	74.4	0.8	0	7.7
GR2	50	283.8	104.9	17.0	12.3	24.9	0.03	0	0.2
GR3	50	406.0	158.5	22.8	10.5	45.6	0.12	0	5.4
GR4	50	606.1	305.3	27.4	14.4	43.6	0	0	0
GR5	50	634.1	305.6	25.5	11.7	39.2	0.004	0	0.02
GR6	80	31.3	10.8	35.0	0	66.7	0	0	0
GR7	80	152.0	108.6	21.6	5.3	30.5	0	0	0
GR8	80	300.1	105.5	27.4	14.4	43.6	0	0	0
GR9	80	566.9	205.6	18.8	12.7	25.3	0	0	0
GR10	100	235.3	75.8	18.8	13.6	27.8	0.09	0	0.9
GR11	100	298.3	100.5	20.0	12.4	23.7	0.04	0	0.4
GR12	100	305.2	186.7	17.4	7.2	38.5	0.15	0	4.0
GR13	100	665.5	302.0	23.2	13.4	36.6	0.7	0	3.8
GR14	100	469.6	164.9	18.6	13.4	23.8	0	0	0
GR15	100	702.9	249.0	17.7	12.4	24.0	0	0	0

6. Conclusion

This paper addressed the *Linear Arrangement* problem while focusing on *interval* graphs. We first developed a new lower bound *LB*, tight for *unit interval* graphs, then solved in an exact way the restriction PCLAP of LAP which is obtained by imposing any linear arrangement to be *consistent* with the *Precedence* partial ordering, and ended by bounding the gap, from both a theoretical and an experimental point of view, which is induced by solving PCLAP instead of LAP.

Still, we could notice that important questions remain open. One of them is about the link between the *Unit Cost Max Cut* Problem and the LAP problem. Would it be possible to tell more about properties which would make *Unit Cost Max Cut* easy to solve on the complementary of *interval* graphs and about efficient related algorithms? This would open the way to the design of efficient branch and bound LAP algorithms for *interval* graphs. Also, our intuition is that the bound stated proposed in Theorem 5.5 could eventually be improved. Finally, one may ask whether the methods which have been used throughout this paper could be applied to classes of graphs which present some kind of similarities with interval graphs.

So, in the future, we plan trying to go further on these issues and more specifically studying the case of *circular* and *chordal* graphs.

Acknowledgements. This research was supported by LABEX IMOBS3 and FEDER funds, and by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] S. Achouri, T. Bossart and A. Munier−Kordon, A polynomial algorithm for MINDSC on a subclass of series parallel graphs. *RAIRO: OR* **49** (2009) 145–156.

[2] N. Ailon, M. Charikar and A. Newman, Aggregating inconsistent information: ranking and clustering. *Proc. of 37th ACM Symp. Theory Comput. (STOC)* (2005) 684–693.

[3] C. Berge, Graphes et Hypergraphes. Dunod Ed, Paris (1974).

[4] F. Barahona and A.R. Mahjoub, On the cut polytope. *Math. Program.* **36** (1986) 157–173.

[5] A. Caprara, Lower bounds for the minimum linear arrangement of a graph. *Electronic Notes Discrete Math.* **36** (2010) 843–849.

[6] A. Caprara, A. Letchford and J. Salazar, Decorous lower bounds for minimum linear arrangement. *INFORMS J. Comput.* **23** (2011) 26–40.

[7] F.K. Chung, On optimal linear arrangement of trees. *Comput. Maths. Appl.* **11** (1984) 43–60.

[8] V. Chvatal and C. Ebenegger, A note on line digraphs and the directed Max-Cut problem. *Discrete Appl. Math.* **29** (1990) 165–170.

[9] J. Cohen, F. Fomin, P. Heggernes, D. Kratsch and G. Kucherov, Optimal linear arrangement of interval graphs. *Proc. of MFCS'06*. Springer Verlag Berlin, Heidelberg (2006).

[10] D.G. Corneil, H. Kim, S. Natarajan, S. Olariu and A.P. Sprague, Simple linear time recognition of unit interval graphs. *Information Processing Lett.* **55** (1995) 99–104.

[11] J. Diaz, J. Petit and M. Serna, A survey on graph layout problems. *ACM Comput. Surveys* **34** (2002) 313–356.

[12] Even S., Shiloach Y., NP-Completeness of Several Arrangement Problems. Technical Report #43. Computer Science Department, *The Technion*, Haifa, Israel (1975).

[13] P. Fishburn, P. Tetali and P. Winkler, Optimal linear arrangement of a grid. *Disc. Math.* **213** (2000) 123–139.

[14] G.N. Frederickson and S.E. Hambrusch, Planar linear arrangements of outerplanar graphs. *IEEETCS: IEEE* Trans. Circuits Syst. **35** (1988) 323–333.

[15] M.R. Garey and D.S. Johnson, Computers and intractability: a guide to the theory of NP-completeness. Freeman and Co., New York (1979).

[16] M. Grotschel, The Sharpest Cut, MPS-SIAM Series on Optimization (2004).

[17] A. Guenoche, B. Vandeputte−Riboud and J.-B. Denis, Selecting varieties using a series of trials and a combinatorial ordering method. *Agronomy* **14** (1994) 363–375.

[18] S.B. Horton, *The optimal linear arrangement problem: algorithms and approximation*. Ph.D. Thesis, Georgia Institute of Technology (1997).

[19] S.B. Horton, R.G. Parker and R.B. Borie, On minimum cuts and the linear arrangement problem. *Discrete Appl. Math.* **103** (2000) 127–139.

[20] O. Hudry, Complexity of voting procedures, in Encyclopedia of complexity and System Sciences. Edited by R. Meyers. Springer (2009).

[21] S. Ilya, *The minimum linear arrangement problem*, M. Sc. thesis, Weismann Institute, Haifa, Israel (2003).

[22] Y. Koren and D. Harel, A multi-scale algorithm for the linear arrangement problem. In Revised Papers from the 28th International Workshop on Graph-Theoretic Concepts in Computer Science, WG '02. Springer Verlag (2002) 296–309.

[23] A. Lad, R. Ghani and Y. Yang and B. Kisiel, Towards optimal ordering of prediction task. *Proc. of SIAM Int. Conf. Datamining SDM09* (2009) 883–894.

[24] W. Liu and A. Vannelli, Generating lower bounds for the linear arrangement problem. *Discrete Appl. Math.* **59** (1995) 137–151.

[25] J. Petit, Approximation heuristics and benchmarking for the MinLA problem. Edited by R. Battiti and A. Bertossi. In *Alex '98* Proceedings, Univ. di Trento (1998) 112–128.

[26] J. Petit, Experiments on the minimum linear arrangement problem. *ACM J. Exp. Algorithmics* **8** (2003) 25–38.

[27] J. Petit, Addenda to the survey of layout problems. *Bull. Eur. Assoc. Theor. Comput. Sci.* **105** (2011) 177–201.

[28] P. Raoufi, H. Rostami and H. Bagherinezhad, An optimal time algorithm for the minimum linear arrangement of chord graphs. *J. Infor. Syst.* **238** (2013) 212–220.

[29] E. Rodriguez−Tello, J.-K. Hao and J. Torres−Jimenez, An effective two-stage simulated annealing algorithm for the Minimum Linear Arrangement problem. *Comput. Oper. Res.* **35** (2008) 3331–3346.

[30] R. Schwarz, A branch-and-cut algorithm with betweenness variables for the Linear Arrangement problem. Diplomarbeit. Universität Heidelberg (2010).

[31] A. Van Zuylen and D.B. Williamson, Deterministic algorithms for rank aggregation and other ranking and clustering problems. Edited by C. Kaklamakis and M. Skutella, Approximation and on line Algorithms. In Vol. 4927 of *Lect.* Notes Comput. Sci. Springer Berlin (2008) 260–273.

[32] J. Yuan and S. Zhou, Optimal labelling of unit interval graphs. *Appl. Math.* **10** (1995) 337–344.