Shiftable Interval Graphs

Federico Malucelli(*), Sara Nicoloso(**)

Abstract: A Shiftable Interval Graph (SIG) is defined by a set of intervals and a set of windows associated with the intervals. Each interval does not have a fixed position, but it is allowed to move, provided that it remains completely contained into its window. Once a position has been fixed for all the intervals, the graph becomes an usual interval graph. In this paper we address the problem of finding the position of the intervals, which minimizes or maximizes some classical measures of the graph, such as clique number, stability number, chromatic number, clique cover number. We mainly focus on complexity aspects, bounds and solution algorithms. Some problems are solvable in polynomial time, others are proved to be NP-hard. Moreover some subclasses of SIG's ,for which exist polynomial algorithms exist, are characterized. Many practical applications can be reduced to problems on SIG's, and SIG's seem to be an interesting modeling framework.

1. Introduction and general definitions

In the present paper the class of Shiftable Interval Graphs (SIG's) is introduced as an extension of the class of interval graphs (IG's) (Golumbic 1980). For this class of graphs we will study the well known concepts of clique, independent set, cover by clique, coloring. In particular we will analyze the complexity of determining some characteristic measures on SIG's (such as min or max clique number, min or max stability number, etc.). When possible we will devise efficient algorithms. For the NP-complete problems, we shall propose lower and upper bounds and identify subclasses of easy instances.

A SIG S is defined by a set of n triples $t_i = <l_i,r_i,\lambda_i>$ of non-negative integer numbers satisfying $r_i - l_i \geq \lambda_i > 0$, i.e. $S \equiv \{t_i = <l_i,r_i,\lambda_i> \in Z^3_+ \colon r_i - l_i \geq \lambda_i > 0$, for $i = 1,\ldots,n\}$. The pair $[l_i,r_i]$ will be called *window* w_i and the value λ_i will be called the *length of the interval* associated with window w_i.

It is easy to think of a SIG as a set of intervals each of which is free to move within the corresponding window, i.e. such that the left endpoint of the i^{th} interval does not lay on the left of l_i, and the right endpoint of the same interval does not lay on the right of r_i. The exact position of each interval within its window is easily described by means of the *placement* $\varphi = [\varphi_1, \varphi_2,\ldots, \varphi_n]$, a vector the j^{th} component of which represents the distance between the left endpoint of the j^{th} interval and the left endpoint, l_j, of the corresponding window w_j. A placement φ is *feasible* if $0 \leq \varphi_j \leq r_j - l_j - \lambda_j$ for all j. Thus, once φ_j has been fixed to $\bar{\varphi}_j$, the coordinate of the left and right endpoints of the j^{th} interval are given by $l_j + \bar{\varphi}_j$ and $l_j + \bar{\varphi}_j + \lambda_j$, respectively.

In what follows, the pair $(t_i,\bar{\varphi}_i)$ will represent the fact that the i^{th} interval has been placed according to $\bar{\varphi}_i$, that is the pair $(t_i,\bar{\varphi}_i)$ represents the interval $[l_i + \bar{\varphi}_i, l_i + \bar{\varphi}_i + \lambda_i]$.

(*) Dipartimento di Elettronica e Informazione - Politecnico di Milano, Via Ponzio 34/5 20133 Milano, Italy. e-mail: malucell@elet.polimi.it

(**) IASI - CNR, Viale Manzoni 30, 00185 Roma, e-mail nicoloso@iasi.rm.cnr.it

By *interval model* $M(\bar{\varphi})$ we shall indicate the set $M(\bar{\varphi}) = \{(t_1,\bar{\varphi}_1), (t_2,\bar{\varphi}_2),\ldots, (t_n,\bar{\varphi}_n)\}$, which is, in fact, a set of intervals of the real line. The intersection graph of the intervals in $M(\bar{\varphi})$ will be denoted by $G(\bar{\varphi})$.

We say that any two intervals $[a,b]$ and $[a',b']$ intersect when $a' < b \leq b'$. The intersection graph G of the intervals is, clearly, an interval graph. Interval graphs are deeply studied in the literature; often we will exploit some properties of interval graphs to approach the problems defined on SIG's.

The set of all interval graphs $G(\varphi)$ obtained by varying φ in all possible ways is called the *family* F_S associated with the given SIG S. Notice that different values of the placement vectors φ, hence different interval models, may give rise to the same interval graph $G(\varphi)$.

A minimization (maximization, respectively) problem on a SIG S is defined as follows:

Given:	a SIG $S \equiv \{t_i = \langle l_i,r_i,\lambda_i\rangle \in Z_+^3 : r_i - l_i \geq \lambda_i > 0$, for $i = 1,\ldots,n\}$ and a function f: $F_S \rightarrow Z_+$,
Find:	a graph $G \in F_S$,
Such That:	$f(G)$ is minimum (maximum, resp.) over all graphs in F_S.

In other words, an optimization problem on a SIG S consists in identifying an interval graph $G \in F_S$ on which $f(G)$ attains its optimum value.

If f is defined as a max–type function itself, a minimization problem on a SIG S turns out to be a min–max problem. This happens, for example, when f is defined as the clique number of G. In fact, in this case the optimization problem consists in finding a graph $G \in F_S$ whose MAXimum complete subgraph has MINimum size. By similar reasoning we obtain min–min, max–min, and max–max problems.

Given an interval graph G we will denote by $\omega(G)$, $\chi(G)$, $\alpha(G)$, $k(G)$, $d(G)$ the clique number (i.e. the size of a complete subgraph of maximum size), the chromatic number (i.e. the size of a coloring of minimum size), the stability number (i.e. the size of an independent set of maximum size), the clique cover number (i.e. the size of a minimum sized covering by complete subgraphs), and the size of the minimum dominating set. The problems of determining a $G \in F_S$ which minimizes (maximizes) $\omega(G)$, $\chi(G)$, $\alpha(G)$, $k(G)$ and $d(G)$ will be denoted by min (max) $\omega(S)$, $\chi(S)$, $\alpha(S)$, $k(S)$ and $d(S)$. Note that given an interval graph defined on n nodes and m edges, the problems of determining $\omega(G)$, $\chi(G)$, $\alpha(G)$, $k(G)$ takes $O(n \log n)$ time (Gupta et al. 1982), while determining $d(G)$ takes $O(n+m)$ time (Bertossi 1986, Farber 1984).

For some of these problems we shall devise polynomial algorithms. Other problems will be proved to be NP-hard, and we shall prove some lower and upper bound for them. Finally, we shall try to characterize subclasses of SIG's for which the problems that are difficult in the general case, can be solved in polynomial time.

Many practical applications can be reduced to these kinds of problems on SIG's. Take as an example some scheduling problems where jobs with ready and due dates are to be scheduled on a set of identical machines: the ready and due dates of a job can be seen as

the left and right end point of a window, respectively, and its processing time as the interval length associated to the same window. We will show how minimizing or maximizing $\omega(S)$, $\chi(S)$, $\alpha(S)$, $k(S)$ and $d(S)$ can be interpreted in this environment.
In a companion paper (Bonfiglio et al. 1997) the problems related to the size of the dominating set are explored from both theoretical and computational viewpoints.

The paper introduces first some definitions of particular classes of SIG's and some basic properties (Section 2). Then it considers the problems related to the clique and the chromatic number (Section 3), the problems related to the clique cover and the stability number (Section 4). A final section contains some concluding remarks and some directions for future work.

2. Definitions and basic properties

This section is devoted to discussing the relationship between a given SIG S, the graphs of the family F_S, the interval models which can arise, and the same entities of the so-called *derived* SIG which will be introduced in a while.

It is convenient to define the intersection graph $H = (V,E_H)$ of the set of windows $\{[l_i,r_i]: i=1,\ldots, n\}$ where the nodes of V are in one–to–one correspondence with the windows of the given SIG, and an edge connects two nodes u, v if and only if the corresponding windows intersect. We now observe the following:

Observation 2.1: Any interval graph $G = (V,E_G) \in F_S$ is a partial subgraph of $H = (V,E_H)$, in the sense that $E_G \subseteq E_H$.

Clearly, it is not true that all the partial subgraphs of H are interval graphs (Fig. 1(c)), nor that every partial subgraph of H belongs to F_S, even though it may be an interval graph (Fig. 1 (d)), as the following Figure shows, where $S \equiv \{t_1$=<1,7,4>, t_2=<3,13,3>, t_3=<2,7,1>, t_4=<4,6,1>, t_5=<8,12,2>}.

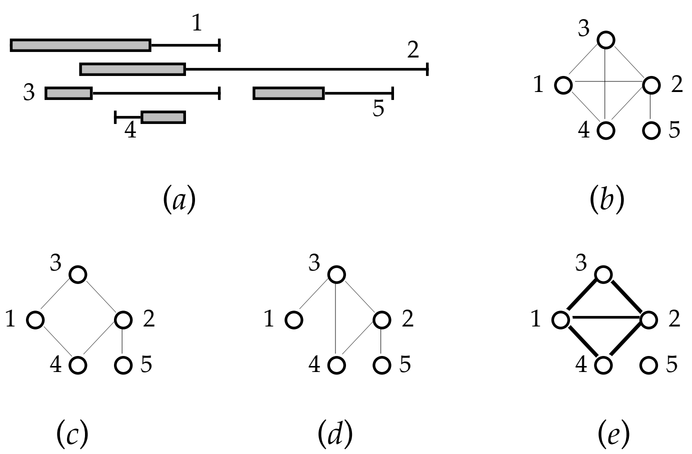

Fig. 1 — (*a*) a SIG and a feasible placement; φ (*b*) the graph H;
(*c*) a partial subgraph of H which is not an interval graph;
(*d*) a partial subgraph of H which is an interval graph but does not belong to F_S,
(*e*) a partial subgraph of H which is an interval graph and belongs to F_S (in fat it is $G(\varphi)$.

Later on, due to the relation between graphs in F_S and feasible placements, we will refer to the problem of finding a particular $G \in F_S$ as equivalent to that of finding a feasible placement which induces G.

We distinguish two kinds of edges:

Definition 2.2: An edge $(u,v) \in E_H$ is *strong* if and only if $(u,v) \in E_G$ for any $G \in F_S$.

Definition 2.3: An edge $(u,v) \in E_H$ is *weak* if and only if there exists at least one graph $G \in F_S$ such that $(u,v) \notin E_G$.

Obviously, given any edge $(u,v) \in E_H$ there always exists at least one graph $G \in F_S$ such that $(u,v) \in E_G$.

According to the above definition, the set E_H^S of all strong edges and the set E_H^W of all weak edges form a partition of E_H.

From the above observations it follows that the family F_S has a finite cardinality, in fact $|F_S| \leq 2^{|E_H|} < 2^{n^2}$. This is a peculiar property of SIG's, for they allow to map an infinite number of interval models into the finite set F_S. This follows from the fact that φ_i may assume one out of an infinite number of real values.

Definition 2.4: A vertex v is *short* (*long*, respectively) if the corresponding triple t_v verifies $r_v - l_v \leq 2\lambda_v$ ($r_v - l_v > 2\lambda_v$).

The triple t_i = <4,13,6> corresponds to a short vertex, the triple t_j = <l,16,3> corresponds to a long one.

According to the above definition, the set V_S of all short vertices and the set V_L of all long vertices form a partition of the vertex set V.

Lemma 2.5:

No strong edge connects two long vertices.

Proof:

By contradiction. Consider two long vertices u,v connected by a strong edge (u,v). We claim that there exist placements φ_u, φ_v such that the two corresponding intervals do not intersect. In particular, either $\varphi_u = 0$ and $\varphi_v = r_v - l_v - \lambda_v$, or $\varphi_u = r_u - l_u - \lambda_u$ and $\varphi_v = 0$, are such that one interval lays completely to the left of the other one, proving the lemma. ❑

Let $\mathrm{Adj}^S(v)$ be the set of vertices connected to v by a strong edge.

Theorem 2.6

Let v be a long vertex. Then $\mathrm{Adj}^S(v)$ defines a subset of vertices mutually connected by strong edges .

Proof:

Consider any two vertices $x,y \in \mathrm{Adj}^S(u)$ (clearly, x,y are short vertices, in light of Lemma 2.5). Whatever the placement φ_x, φ_y, and φ_u are, intervals x, y intersect interval u, and clearly intersect each other, as edges (x,u), (y,u) are strong. This proves the claim. ❑

Notice that $\{v\} \cup \mathrm{Adj}^S(v)$ induce a complete subgraph in K (thus in H).

It is convenient to introduce the graph K defined on the set V of vertices and the set E_H^S. Notice that $E_H^S = \bigcap_{G \in F_S} E_G$.

We now introduce the following definition:

Definition 2.7: A SIG is called *degenerate* if all edges of H are strong.

Notice that in this case one necessarily has $|F_S| = 1$ and (the unique) $G \in F_S$ is isomorphic to H.

Degenerate SIG's arise, for example, when only one feasible placement is possible, that is $\varphi=0$ (case 1), or when intervals *may* assume different positions within their windows, nevertheless the intersection between any two interval persists whatever their placement is (cases 2 and 3):

(1) – when $\lambda_i = r_i - l_i$ for all $t_i \in S$;

(2) – when given any two triples $t_u, t_v \in S$ one has that

(*i*) $l_u \leq l_v \leq r_u \leq r_v$ implies $l_u + \lambda_u \geq r_v - \lambda_v$

(*ii*) $l_u \leq l_v < r_v \leq r_u$ implies both $l_u + \lambda_u \geq r_v - \lambda_v$ and $l_v + \lambda_v \geq r_u - \lambda_u$;

(3) – when $\lambda_i = r_i - l_i - k$, for any $t_i \in S$, for a fixed $k \geq 0$, and for any two triples $t_u, t_v \in S$ one has that:

(*i*) $l_u \leq l_v \leq r_u \leq r_v$ implies $r_u - l_v \geq 2k$

(*ii*) $l_u \leq l_v < r_v \leq r_u$ implies both $r_u - l_v \geq 2k$ and $r_v - l_u \geq 2k$.

Definition 2.8: A SIG S is *proper* if and only if any two mutually intersecting windows w_i, w_j verify $l_i < l_j < r_i < r_j$.

The triples of a proper SIG can be numbered in such a way that $i < j$ if and only if $l_i \leq l_j < r_i \leq r_j$. In what follows we shall always assume that the triples of a proper SIG's are numbered according to this criterion.

Notice that if S is proper, then H is a proper interval graph, also known as *unit interval graph* (Golumbic 1980).

Among the class of proper SIG's we distinguish the following two cases:

Definition 2.9: A SIG S is *proper non-decreasing* if and only if S is proper and $\lambda_i \leq \lambda_j$ for any $i < j$.

Definition 2.10: A SIG S is *proper non-increasing* if and only if S is proper and $\lambda_i \geq \lambda_j$ for any $i < j$.

Definition 2.11: A SIG S is a (0,1)–SIG if and only if $\lambda_i = r_i - l_i - 1$, for any $i = 1,\ldots,n$.

Notice that every interval of a (0,1)–SIG can assume either one of at most two positions, namely the leftmost or the rightmost one, thus $\varphi_i \in \{0,1\}$, $i = 1,\ldots,n$. In this case most edges are strong; in particular, if every pair i,j of mutually intersecting windows verifies $\min\{r_i,r_j\} - \max\{l_i,l_j\} \geq 3$, then all edges are strong, and the (0,1)–SIG is degenerate.

A certain subSIG of the given SIG S plays a very important role in many situations. We define it as follows:

Definition 2.12: The *derived* SIG S_d associated to the given SIG S is obtained by removing from S all triples whose window properly contains another window of S.

Note that, clearly, a derived SIG is proper. Moreover, the intersection graph H_d of the set of windows in S_d is an induced subgraph of H. Notice also that $S_d \equiv S$ if and only if S is proper, and that every graph $G \in \mathrm{F}_{S_d}$ is a partial subgraph of H.

3. The clique number and chromatic number problems

As noticed before, once a placement φ is given, the intersection relation among the intervals in $M(\varphi)$ can be represented by means of a graph $G(\varphi) \in \mathrm{F}_S$, which clearly is an interval graph, and the equality $\omega(G(\varphi)) = \chi(G(\varphi))$ holds for it (Golumbic 1980). This implies that the result of the optimization of $\omega(S)$ immediately applies to the same optimization of $\chi(S)$. Notice that assuming that l_i, r_i and λ_i are integer for all i=1,…,n, w.l.o.g. we can limit ourselves to integer φ.

3.1 *Minimization of* $\omega(S)$ *and* $\chi(S)$

3.1.1 Computational complexity

Consider the following decision problem:

Problem 3.1
Given a SIG S and a positive integer $K \leq n$, find a feasible placement $\varphi \in \mathrm{Z}_+^n$ such that $\omega(G(\varphi)) \leq K$.

Theorem 3.2
Let S *be* a SIG such that $l_i = 0$ and $r_i = \rho$, $i = 1,\ldots,n$. Problem 3.1 is NP-complete in the strong sense.
Proof:
Problem 3.1 is easily seen to be in NP. The proof is by reduction from 3-PARTITION, which is NP–complete in the strong sense (Garey and Johnson 1979): Given $3m$+1 positive integers $b_1, \ldots, b_{3m}$, B with $\frac{B}{4} < b_j < \frac{B}{2}$ for any $j = 1, \ldots, 3m$ and $\sum_{j=1}^{3m} b_j = mB$, find a partition of the b_j's into m subsets $\beta_1, \ldots, \beta_m$, such that $\sum_{b_j \in \beta_i} b_j = B$ and $|\beta_i| = 3$ for any $i = 1,\ldots,m$. Now, given any instance of 3-PARTITION we construct in polynomial time a corresponding instance of Problem 3.1, as follows: we set $K = m$, and with each integer b_j we associate a triple $t_j = \langle l_j, r_j, \lambda_j \rangle$, where $l_j = 0$, $r_j = B$ and

$\lambda_j = b_j$, for any $j = 1,\ldots,3m$. It is easy to see that a feasible placement φ exists if and only if 3-PARTITION is a YES-instance. ❑

The following corollary is a trivial consequence of the previous theorem.

Corollary 3.3

Given an arbitrary SIG S, the problem of minimizing $\omega(S)$ and the problem of minimizing $\chi(S)$ are NP-hard in a strong sense.

3.1.2 Lower bound

Theorem 3.4

$\omega(K) \leq \min \omega(S)$.

Proof:

It follows immediately recalling that K is a subgraph of any graph $G \in F_S$. ❑

We now describe how to compute $\omega(K)$. To this aim it is convenient to consider the graph $K' = (V_S, E_H^S)$. It is an interval graph, as it coincides with the intersection graph of the set of intervals $\{[r_u - \lambda_u, l_u + \lambda_u]$, for all short triples $t_u \in S\}$. Notice that $[r_u - \lambda_u, l_u + \lambda_u] \subseteq [l_u + \varphi_u, l_u + \varphi_u + \lambda_u]$ for any feasible placement φ_u. Thus $\omega(K')$ is easily computed. By Lemma 2.5 and Theorem 2.6 it follows that

$$\omega(K') \leq \omega(K) \leq \omega(K') + 1.$$

In order to compute the exact value of $\omega(K)$ it is sufficient to determine a long vertex v such that the set Adj(v) has maximum cardinality over all $v \in V_L$. If $|\text{Adj}(v)| = \omega(K')$, then $\omega(K) = \omega(K') + 1$. Otherwise $\omega(K) = \omega(K')$.

3.1.3 Polynomially solvable cases

Even though the problem of minimizing $\omega(S)$ has been just proved to be NP-hard on arbitrary SIG's, there are classes of SIG's for which problem 3.1 can be solved in polynomial time.

Algorithm mM-C (min Max Clique) described below outputs a feasible solution, if any, to Problem 3.1 defined on proper non-increasing SIG's. We assume that triples are numbered by increasing left endpoints. Throughout the algorithm we shall make use of the following definition:

Definition 3.5

Given a set J of intervals $[a_i, b_i]$ and an integer t we define the *rightmost t-clique point* $c_t(J)$ as the rightmost coordinate which belongs to exactly t intervals of J. $c_t(J) = 0$ if J is empty or no coordinate exists which belongs to exactly t intervals of J.

Fig. 2 shows an example.

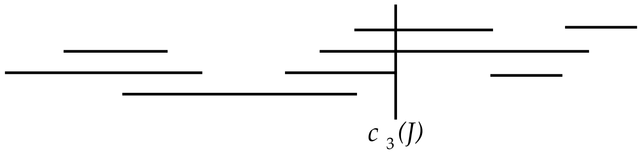

Fig. 2 — The rightmost 3-clique point (recall that intervals are open on the left).

```
Algorithm mM-C;
Input:      a proper non-increasing SIG S = {ti: i=1,…,n}, K;
Output:     a feasible placement φ such that ω(G(φ)) ≤ K;

Begin
For i = 1,…,n Do
        φi := undefined;
Let I := ø;
For i = 1,…,n Do
        Begin
        If cK(I) > ri − λi
        Then Return(NO) and Stop
        Else  Begin
          φi = max {0,  cK(I) − li};
          I := I ∪ {(ti,φi)};
          End;
        End;
Return(YES);
End.
```

Theorem 3.6

Algorithm mM-C correctly solves Problem 3.1 on proper non-increasing SIG's.

Proof:

We shall prove two facts, namely that: *i*) the clique number of the intersection graph of the set of intervals placed according to Algorithm mM–C does not exceed *K*, and *ii*) the algorithm returns NO if and only if the given SIG is a NO–instance.

The first claim is immediate: in fact the algorithm stops as soon as the distance between the current *K*–clique point and the right endpoint of the window under consideration is smaller than the corresponding interval length.

The "if"–part of the second claim follows from the fact that no feasible placement exists for the given instance.

Let us prove the "only if" part of the second claim. Assume that the algorithm returns NO when $i = h+1$, $I \equiv \{(t_1,\varphi_1),\ldots,(t_h,\varphi_h)\}$, and clearly $c_K(I) > r_{h+1} - \lambda_{h+1}$, and assume by contradiction that S is a YES–instance. We have to find a feasible placement $\tilde{\varphi}_{h+1}$ for interval h+1 so that it does not belong to a clique of cardinality larger than K. Consider the set J of all the already placed intervals which intersect coordinate $c_K(I)$, that is $J = \{i\text{: } i \leq h,\ l_i + \varphi_i < c_K(I) \leq l_i + \varphi_i + \lambda_i\}$). In order to place interval h+1 we have to shift some interval $j \in J$. Three cases arise:

a) no interval can be shifted so as to lay completely to the right of $c_K(I)$ or completely to the left of the left endpoint of interval h+1 in its rightmost placement;

b) there exists an interval $j \in J$ which can be shifted rightwards so as to lay to the right of $c_K(I)$, that is, there exists $j \in J$ that admits a feasible placement $\varphi'_j \geq \varphi_j$ such that its left endpoint $l_j + \varphi'_j$ lays not on the left of $c_K(I)$, namely $c_K(I) - l_j \leq \varphi'_j \leq r_j - l_j - \lambda_j$;

c) there exists an interval $j \in J$ which can be shifted leftwards so as to lay to the left of the left endpoint of interval h+1 in its rightmost placement, that is, there exists $j \in J$ that admits a feasible placement $0 \leq \varphi'_j \leq \varphi_j$ such that its right endpoint $l_j + \varphi'_j + \lambda_j$ lays not on the right of $r_{h+1} - \lambda_{h+1}$, namely $0 \leq \varphi'_j \leq r_{h+1} - \lambda_{h+1} - l_j - \lambda_j$.

If case *a*) applies, no solution exists, and in fact no feasible placement exists such that the number of intervals crossing coordinate $c_K(I)$ is smaller than k+1.

If case *b*) applies, as the SIG is non increasing, $r_j - \lambda_j \geq c_K(I)$ implies that $r_{h+1} - \lambda_{h+1} \geq c_K(I)$, which contradicts the stopping condition of the algorithm, and $\tilde{\varphi}_{h+1} = c_K(I) - l_{h+1}$ is a feasible placement for interval h+1 which does not increase over K the number of mutually intersecting intervals.

If case *c*) applies, the situation is the following. Because of the algorithm behavior, in order not to exceed K, no interval can be shifted leftwards without shifting rightwards any other interval. This means that we have to find an interval $i < j$ which has to be moved rightwards in order to allow the leftwards shifting of j. Let $I' \equiv \{(t_1,\varphi_1), \ldots, (t_{j-1}\varphi_{j-1})\}$ and let J' be the set of all the intervals p which intersect coordinate $c_K(I')$, with $p<j$. The same kind of reasoning seen above on the possible shifting of an interval can be iteratively applied, until case *a*) applies (which is always the case, for example when $c_K(I') > 0$ for the first time during the execution of the algorithm), which ends the proof. Notice that $c_K(I')$ is always greater than 0, otherwise case *a*) would apply. ❑

The computational complexity of algorithm mM-C for proper non-increasing SIG's is O(n) if we assume that the windows are already sorted by increasing left endpoint.

Algorithm mM-C can be used to solve the optimization version of the problem by solving a sequence of problems for different values of K. A good way of operating consists of applying a dichotomic search on the values of K, as lower and upper bounds are known for it. Namely, one can apply a dichotomic search for K in the range $[1,\omega(H)]$, since $1 \leq \min \omega(S) \leq \max \omega(S) \leq \omega(H) \leq n$. The number of steps of the dichotomic search amounts to $O(\log \omega(H))$ which is bounded from above by $O(\log n)$. This finally gives an $O(n \log n)$ time algorithm to solve the optimization version of Problem 3.1 on proper non-increasing SIG's.

3.2 *Maximization of ω(S) and χ(S)*

A problem in some sense related to the one just studied is the problem of maximizing $\omega(G)$ over the set of all $G \in F_S$. Unlike the problem of minimizing $\omega(S)$, that of maximizing $\omega(S)$ on arbitrary SIG's, even in its optimization form, can be solved quite easily. Its formal statement is the following:

Problem 3.7
Given a SIG S and a positive integer $K \leq n$, find a feasible placement $\varphi \in Z_+^n$ such that $\omega(G(\varphi)) \geq K$.

One clearly has $\omega(G) \leq \omega(H)$ for any $G \in F_S$, as every G is a subgraph of H. Moreover, one can conclude that max $\omega(S) = \omega(H)$, where max $\omega(S) = \max \{\omega(G), G \in F_S\}$. In fact a feasible placement φ such that $\omega(G(\varphi)) = \omega(H)$ does always exist: be x any coordinate which belongs to exactly $\omega(H)$ windows (such a coordinate does always exist), be C be the set of indices of the windows intersecting it (i.e. $C = \{i\text{: } l_i < x \leq r_i\}$), set $\varphi_i \in [0, \min\{x - l_i, r_i - l_i - \lambda_i\}]$ for all $i \in C$, and $\varphi_i \in [0, r_i - l_i - \lambda_i]$ for all $i \notin C$. It is easy to see that all intervals i, with $i \in C$, intersect coordinate x, that is, $\omega(G(\varphi)) = \omega(H)$

As a consequence, the complexity of determining the placement vector φ which maximizes $\omega(S)$ is dominated by the complexity of determining a complete subgraph of H with maximum size, which requires $O(n \log n)$.

4. The stability number and clique cover number problems

For the same reasons discussed at the beginning of the previous section, equation $\alpha(G) = k(G)$ holds for any graph $G \in F_S$. This fact allows for concluding that, given a SIG S, the result of the optimization over $\alpha(S)$ immediately applies to the same optimization over $k(S)$. Notice that assuming that l_i, r_i and λ_i are integer for all i=1,…,n, w.l.o.g. we can limit ourselves to integer φ, like already done in Section 3.

4.1 *Maximization of α(S) and k(S)*

The present Section is devoted to the problem of maximizing $\alpha(G)$ over all $G \in F_S$.

Problem 4.1
Given a SIG S and a positive integer K, find a feasible placement $\varphi \in Z_+^n$ such that $\alpha(G(\varphi)) \geq K$.

4.1.1 Computational complexity

Theorem 4.2
The optimization version of Problem 4.1 is NP-hard.
Proof:
Consider the one-machine n jobs scheduling problem with ready and due times. The problem of minimizing the number of tardy jobs is NP-hard when the ready times are non-negative (Lenstra et al. 1977). This problem can be trivially reduced to the maximization of $\alpha(\cdot)$ on a suitable SIG S, constructed as follows: to each job i having ρ_i, δ_i, p_i as release date, due date, and processing time, respectively, we associate the triple $t_i = \langle\rho_i, \delta_i, p_i\rangle$; we then set S to be the set of all triples t_i, $i = 1,\ldots,n$. The size of the maximum independent set over all graphs $G \in F_S$ is equal to the number of jobs processed on time.
❑

The following corollary is an immediate consequence of the previous theorem.

Corollary 4.3
The problem of maximizing $k(S)$ is NP-hard.

4.1.2 Lower and upper bounds

Theorem 4.4
$\alpha(H) \leq \max \alpha(S) \leq \alpha(K)$.
Proof:
The inequalities follow immediately recalling that H and K are supergraph and subgraph, respectively, of any graph $G \in F_S$. ❑

We now describe how to compute $\alpha(K)$. We construct an independent set I as follows. V_L is an independent set, by Lemma. Consider any $v \in V_L$, by Theorem 2.6, only one vertex out of $\{v\} \cup \mathrm{Adj}^S(v)$ belongs to an independent set. The best choice is to insert v into I. Then I is completed including a maximum independent set of the interval graph induced by the vertex set $V_S \setminus (\bigcup_{v \in V_L} \mathrm{Adj}^S(v))$.

4.1.3 Polynomially solvable cases

Like already done for the minimization of $\omega(S)$, we can find classes of SIG's which allow for solving problem 4.1, i.e. for maximizing $\alpha(S)$, in polynomial time.

The maximization of $\alpha(S)$ on proper SIG's can be conducted in $O(n^2)$ time by making use of the algorithm by Kise et al. (Kise et al. 1978), which is based on a dynamic programming approach.

We now consider the class of proper non-decreasing SIG's. It is a subclass of the proper SIG's class and, of course, the algorithm by Kise et al. applies. Nevertheless, we now propose an $O(n)$ time algorithm for the problem of maximizing $\alpha(S)$ on them. Again we shall assume that triples are numbered by increasing left endpoints.

```
Algorithm MM-IS
Input:      a proper non-decreasing SIG S = {t_i: i=1,...,n};
Output:     a feasible placement φ such that α(G(φ)) is maximum;

Begin
For i = 1,...,n Do
   φ_i := undefined;
Let IS := ø;
p := l_1;
For i = 1,...,n do
   Begin
   If p ≤ r_i − λ_i
   Then Begin
        φ_i := max {0,p − l_i};
        IS := IS ∪ {(t_i,φ_i)};
        p := l_i + φ_i + λ_i
      End
```

End
End.

Theorem 4.5

Algorithm MM-IS for proper non-decreasing SIG's is correct.

Proof:

The proof follows from the observation that, at step i of the algorithm, intervals in IS isolate the maximum independent set for the subproblem defined by the first i triples (numbered by increasing left endpoints); moreover the placement φ_i is such that, among all possible independent sets of maximum size for the subSIG $S' \equiv \{t_j: j = 1,\dots,i\}$, it is one whose rightmost right endpoint is minimum. The proof is by induction. For $i = 1$ we have $IS = \{(t_1,0)\}$ and the above observation holds. Assume that the observation is true for i and consider i+1. Either interval i+1 can be added to IS and the observation holds true, or interval i+1 does not admit a feasible placement which allows its insertion into the independent set IS. In this second case we could either discard interval i+1 or discard another interval $h < i+1$ and possibly insert i+1 into IS with a suitable feasible placement; in this latter subcase, in particular, one has to find a new placement to all intervals $h+1,\dots,i$ and then possibly place i+1 in a feasible position, so as to insert it into IS. But, since $\lambda_{i+1} \geq \lambda_i$, this second case would produce an IS of the same size to which corresponds a value of p which can not be smaller than the one corresponding to the previous IS. ❏

If the intervals are already sorted, the complexity of MM-IS is $O(n)$.

4.2 *Minimization of* $\alpha(S)$ *and* $k(S)$

Now, we discuss the problem of finding a feasible placement φ which defines an interval graph G whose maximum independent set has minimum size. Thanks to the well-known relation which holds among the stability number $\alpha(\cdot)$ and the clique cover number $k(\cdot)$ of a perfect graph, we can re-state the problem as follows:

Problem 4.6

Given a SIG S and a positive integer K, find a feasible placement $\varphi \in Z^n_+$ such that $a(G(\varphi)) \leq K$.

The solution to this problem is easily found by applying the following algorithm: consider the intersection graph H of the set of windows; be Γ a minimum covering by cliques of the node set of H (it takes linear time to find); for each clique $C \in \Gamma$, let x be a coordinate belonging to every window whose corresponding node is in C; for any $i \in C$ set φ_i to a suitable value such that interval i intersects coordinate x (more formally: φ_i is to be such that $l_i + \varphi_i \leq x \leq l_i + \varphi_i + \lambda_i$, that is $0 \leq \varphi_i \leq \min\{x - l_i, r_i - l_i - \lambda_i\}$). If there exists windows whose corresponding node belongs to more than one clique, assign φ w.r.t. any one of them. It is easy to see that such an algorithm finds a placement vector φ which minimizes the size of both $\alpha(S)$ and $k(S)$ and requires a running time linear in the number n of windows.

5. Conclusions and future work

In this paper we introduced the class of Shiftable Interval Graphs (SIG's) and we analyzed some problems defined on them. The SIG's can be considered as a general framework which can be used to formulate many practical problems. A quite natural application is in the field of scheduling. As pointed out in the Proof of Theorem 4.2, a set of jobs can be represented by a SIG where release and due dates of jobs are represented by windows and the processing times are represented by interval lengths. In practice a schedule is given by the feasible placement of the intervals. Thus, minimizing the number of identical machines is equivalent to minimizing the clique number, or minimizing the number of tardy jobs is equivalent to maximizing the independent set. Other problems can find a direct interpretation in terms of SIG's. However, the study of the SIG's suggests many other interesting problems for example those related to the dominating set. In this case the SIG's have been a useful tool to prove new results. The purpose of this paper is mainly introductory, hence only few of the possible features of the SIG's have been dealt with; after the introduction of the basic definitions, we focused on the problems related to the clique number, the stability number and the size of the dominating set. Many other problems remain to be investigated, for example gaining a deeper insight into the maximum dominating set and minimizing/maximizing the size of a totally dominating set.

Notice that many problems that in the case of Interval Graphs have no sense, when SIG's are considered, gain interest and are far from being trivial; we refer for example to minimizing or maximizing the minimum clique, or the minimum independent set.

Another interesting problem is the recognition of SIG's. Given a graph G whose arcs can be active or not, and a set of relations between arcs defined as follows: for each arc (i,j) we have a set of *compatible arcs* $A(i,j)$ which are always active whenever arc (i,j) is active, and a set of *incompatible arcs* $\bar{A}(i,j)$ which are always inactive whenever arc (i,j) is active. The problem of the SIG recognition consist in determining if the relations between arcs can be modeled in the framework of a SIG and give a possible set of windows and intervals.

A natural and immediate extension is to consider the *Shiftable Circular Arc Graphs*, and to study the complexity of minimizing or maximizing the classical measures an these classes of graphs.

References

Bertossi, A.A., "Total Domination in Interval Graphs", *Inf. Proc. Lett.* 23, (1986), 131-134.

Bonfiglio P., F. Malucelli, and S. Nicoloso, "The dominating set problem on shiftable interval graphs", (1997), Technical Report no. 461, IASI-CNR Rome.

Farber M., "Domination, independent domination, and duality in strongly chordal graphs" *Discrete Applied Mathematics* 7, (1984) pp. 115-130.

Garey M. R., and D. S. Johnson, “Two-processor scheduling with start-times and deadlines”, *SIAM Journal on Computing* 6, (1977) pp. 416-426.

Garey M. R., and D. S. Johnson, *Computers and intractability A guide to the theory of NP-Completeness*, W. H Freeman and Company, New York, 1979.

Gupta U.I., D.T. Lee, and J.Y.-T. Leung, “Efficient Algorithms for Interval Graphs and Circular Arc Graphs”, *Networks* 12, (1982), 459-467.

Golumbic M. C., *Algorithmic Graph Theory and Perfect Graphs*., Academic Press Inc, San Diego, 1980.

Kise H., T. Ibaraki, and H. Mine, “A solvable case of the one machine scheduling problem with ready and due dates”, *Operations Research* 26, (1978) pp. 121-126.

Lenstra J. K., A. H. G. Rinnooy-Kan, and P. Brucker, “Complexity of machine scheduling problems”, *Annals of Discrete Mathematics* 1, (1977) pp. 343-362.