On partitioning interval and circular-arc graphs into proper interval subgraphs with applications

Frédéric Gardi *

Laboratoire d'Informatique Fondamentale,
Parc Scientifique et Technologique de Luminy,
Case 901 - 163, Avenue de Luminy,
13288 Marseille Cedex 9, France
Frederic.Gardi@lif.univ-mrs.fr

Abstract. In this note, we establish that any interval or circular-arc graph with n vertices admits a partition into $O(\log n)$ proper interval subgraphs. This bound is shown to be asymptotically sharp for an infinite family of interval graphs. Moreover, the constructive proof yields a linear-time and space algorithm to compute such a partition. The second part of the paper is devoted to an application of this result, which has actually inspired this research: the design of an efficient approximation algorithm for a $\mathcal{NP}$-hard problem of planning working schedules.

1 Introduction

An undirected graph G=(V,E) is an *interval graph* if to each vertex $v \in V$ can be associated an open (resp. closed) interval I_v of the real line, such that any pair of distinct vertices u, v are connected by an edge of E if and only if $I_u \cap I_v \neq \emptyset$. The family $\{I_v\}_{v \in V}$ is an *interval representation* of G; the left and right endpoints of I_v are respectively denoted by $le(I_v)$ and $re(I_v)$. The edges of the complement graph $\overline{G}$ are transitively orientable by setting $u \to v$ if $r_u < l_v$; the orientation of the edges induces a partial order called interval order (we shall write $I_u \prec I_v$ if $r_u < l_v$). In the same way, the intersection graph of collections of arcs on a circle is called *circular-arc graph.* A circular-arc representation of an undirected graph G which fails to cover some point p on the circle will be topologically the same as an interval representation of G. In effect, we can cut the circle at p and straighten it out a line, the arcs becoming intervals. It is easy to notice therefore, that every interval graph is a circular-arc graph.

An interval graph G is called *proper interval graph* if there is an interval representation of G such that no interval contains properly another. A nice result of Roberts (1969, *cf.* [13, 6]) establishes that proper interval graphs coincide with *unit interval graphs*, the interval graphs having an interval representation such that all intervals have the same size, and $K_{1,3}$*-free interval graphs*, the interval graphs without induced copy of a tree composed of one central vertex and three leaves.

* The author is a Ph.D. student in Computer Science and Mathematics.

The main result. Interval and circular-arc graphs have been intensively studied for several decades by both discrete mathematicians and theoretical computer scientists. These two classes of graphs are particulary known for providing numerous models in diverse areas like scheduling, genetics, psychology, sociology, archæology and others. For surveys on all results and applications concerning interval and circular-arc graphs, the interested reader is referred to [13, 6, 8].

In this note, the problem of *partitioning interval or circular-arc graphs into proper interval subgraphs* is investigated. Two questions can be raised concerning this problem. The first, rather asked by the mathematician is: could you find good lower and upper bounds on the size of a minimum partition of an interval or circular-arc graph into proper interval subgraphs ? The second, rather asked by the computer scientist is: could you find an efficient algorithm to compute such a minimum partition ? An answer to the first question is given in this paper, through the following theorem. Although the result provides some advances on the second question (discussed in Conclusion), this one remains open at our knowledge.

Theorem 1. *Any interval graph or circular-arc graph with n vertices admits a partition into $O(\log n)$ proper interval subgraphs. Moreover, this bound is asymptotically sharp for an infinite family of interval graphs.*

The constructive proof of the result (described Section 2) yields a linear-time and space algorithm to compute such a partition. Thereby, this result could find applications in the design of approximation algorithms for hard problems on interval or circular-arc graphs, since many untractable problems for these graphs become easier for proper interval graphs. In the second part of the paper, we present such a kind of application in the area of working schedules planning, which has actually inspired this research.

Applications. The problem of *planning working schedules* holds an important place in operations research and business administration. In a schematic way, the problem consists in the *assignment of fixed tasks to employees in the form of shifts*. The tasks of the shift allocated to an employee, which induce his working schedules, must be pairwise *disjoint* (non-intersecting). Here a problem derived from schedules planning problems solved by the firm Prologia - Groupe Air Liquide [12] is considered. This fundamental problem, denoted WSP, is defined as follows. Let $\{T_i\}_{i=1,\ldots,n}$ be a set of tasks having respective starting and ending dates (l_i, r_i). The *regulation* imposes that any employee cannot execute more than k tasks. Given that the tasks allocated to an employee must not overlap, build an optimal planning according to the following objectives: on a *first level*, reduce the number of shifts or employees (*productivity*) and then on a *second level*, balance the planning (*social*) and prevent as well as possible the future modifications of the planning (*robustness*).

Since the tasks are simply some intervals of the real line, the WSP problem can be reformulated in graph-theoretic terms as the problem of *coloring an interval graph such that each color marks at most k vertices.* When the planning is *cyclic*, we obtain the same coloring problem with *circular-arc graphs*. In this model, the optimization criteria become respectively: to minimize the number of colors (P), balance the number of vertices in each color class (S) and maximize the smallest gap existing between two consecutive intervals or arcs having the same color (R). In fact, the criterion R prevents overlappings when some intervals or arcs are delayed or put forward. Hence, a solution to WSP is called (P)-*optimal* (resp. (S, R)-*optimal*) if it is optimal according to criterion P (resp. criteria S *and* R). Then, a $(P|S, R)$-*optimal* solution is defined to be one which is (S, R)-optimal *among* all (P)-optimal solutions.

The complexity of WSP for interval graphs was recently investigated with the single optimization criterion P. Bodlaender and Jansen [2] have shown that this is a $\mathcal{NP}$-hard problem even for fixed $k \geq 4$; the problem for $k = 3$ remains an open question at our knowledge. For $k = 2$, this is solved in linear time and space by matching techniques [1, 5]. Unless $\mathcal{P} = \mathcal{NP}$, the inherent hardness of the problem condemns us to design efficient heuristics for finding "good" solutions. In this way, linear-time approximations are presented for the WSP in the second part of the paper (Section 3). A classical algorithm is briefly described which achieves a constant worst-case ratio for the single criterion P. Unfortunately, such an algorithm offers no guarantee on the satisfiability of criteria S and R. Surprisingly, the WSP problem for proper interval graphs is proved to be solvable in a $(P|R, S)$-optimal way by a greedy algorithm. Thus, an idea is to partition the input interval graph into proper interval subgraphs and solve optimally the problem on each subgraph using the greedy. Obviously, the quality of such a *local optimization* depends strongly on how the input interval graph is partitionned. Hence, the theorem previously cited enables us to design a new algorithm which achieves a logarithmic worst-case ratio for criterion P, but moreover guarantees that $(P|R, S)$-*optima* are reached in a logarithmic number of subproblems. Finally, we remark that in real-life situations, *ie.* under certain conditions, the logarithmic worst-case ratio becomes *constant*.

Preliminaries. Before giving the first results, some useful notations and definitions are detailed. All the graph-theoretic terms not defined here can be found in [13, 6]. Let $G = (V, E)$ be an undirected graph. For simplicity, n and m denote respectively the number of vertices and edges of G throughout the paper. A *complete set* or *clique* is a set of pairwise connected vertices. The *clique number* $\omega(G)$ is the cardinality of the largest clique in G. On the opposite, an *independent set* or *stable* is a set of pairwise non-connected vertices. A *coloring* of G associates to each vertex one color in such a way that two connected vertices have different colors. In fact, a coloring of G corresponds to a partition of G into stables. The *chromatic number* $\chi(G)$ is the cardinality of a partition of G into the least number of stables. In the same way, $\chi(G, k)$ is defined to be the size of a minimum partition of G into stables of size at most k. The quality of our approximation algorithms in relation to the criterion P is measured by their *worst-case ratio* defined as $\sup_G\{|\mathcal{S}|/\chi(G, k)\}$ where $\mathcal{S}$ is any partition of G into stables of size at most k output by the algorithm.

2 The proof of Theorem 1

Although offering only a linear upper bound, the following lemma is crucial in the proof of the theorem.

Lemma 1. *Let $G = (V, E)$ be a $K_{1,t}$-free interval graph with $t \geq 3$. Then G admits a partition into $\lfloor t/2 \rfloor$ proper interval subgraphs. Moreover, this partition is computed in linear time and space.*

Proof. An algorithm is proposed for computing such a partition. Synthetically, the algorithm extracts and colors greedily some cliques of G with the set of colors $\{1, \dots, \lfloor t/2 \rfloor\}$; the output is the partition of G induced by these $\lfloor t/2 \rfloor$ colors.

Algorithm ColorCliques
input: a $K_{1,t}$-free interval graph $G = (V, E)$ with $t \geq 3$;
output: a partition of G into $\lfloor t/2 \rfloor$ proper interval subgraphs;
begin
  compute an interval representation $I_1, \dots, I_n$ of G;
  order $I_1, \dots, I_n$ according to the left endpoints;
  $\mathcal{C}^1 \leftarrow \cdots \leftarrow \mathcal{C}^{\lfloor t/2 \rfloor} \leftarrow \emptyset$, $i \leftarrow 1$, $j \leftarrow 1$;
  while $i \leq n$ **do**
    $C_j \leftarrow \{I_i\}$, $I_{left} \leftarrow I_i$, $i \leftarrow i + 1$;
    while $i \leq n$ and $I_{left} \cap I_i \neq \emptyset$ **do**
      $C_j \leftarrow C_j \cup \{I_i\}$;
      if $re(I_i) < re(I_{left})$ **then** $I_{left} \leftarrow I_i$;
      $i \leftarrow i + 1$;
    $c \leftarrow (j - 1) \bmod \lfloor t/2 \rfloor + 1$, $\mathcal{C}^c \leftarrow \mathcal{C}^c \cup \{C_j\}$, $j \leftarrow j + 1$;
  return $\mathcal{C}^1, \dots, \mathcal{C}^{\lfloor t/2 \rfloor}$;
end;

Since computing an ordered interval representation is done in $O(n+m)$ time and space [4, 9], the algorithm runs in linear time and space. This correctness is established by showing that the color class $\mathcal{C}^c$ induces a proper interval graph for any $c \in \{1, \dots, \lfloor t/2 \rfloor\}$. Let $\mathcal{C}^c = \{C_1^c, \dots, C_q^c\}$ be the set of cliques assigned to $\mathcal{C}^c$ by the algorithm (in the order of their extraction). If $q \leq 2$ then $\mathcal{C}^c$ is trivially $K_{1,3}$-free. Otherwise, suppose that $\mathcal{C}^c$ contains an induced subgraph $K_{1,3}$ with I_a its central vertex and $I_b \prec I_c \prec I_d$ its three leaves. Clearly, the leaves belong to disjoint cliques: set $I_b \in C_u^c$, $I_c \in C_v^c$ and $I_d \in C_w^c$ with $u < v < w \in \{1, \dots, q\}$. According to the algorithm, I_a belongs necessarily to C_u^c. Now, from every clique C_j colored by the algorithm between C_u^c and C_w^c, select the interval having the smallest right endpoint in C_j and add it to the set S initially empty. We claim that S induces a stable of size at least $2\lfloor t/2 \rfloor + 1$. If two intervals of S are intersecting, then they belong to the same colored clique, a contradiction. At least $\lfloor t/2 \rfloor$ cliques are colored by the algorithm from C_u^c to C_v^c exclusive and still at least $\lfloor t/2 \rfloor$ from C_v^c to C_w^c exclusive. Thus, S contains at least $2\lfloor t/2 \rfloor + 1$ elements, which proves the claim. Since $I_a \in C_u^c$ and $I_a \cap I_d \neq \emptyset$, I_a intersects every interval in S except maybe the one most to right which belongs to C_w^c. This last interval is replaced in S by the interval I_d; in effect, I_d cannot intersect the last but one interval of S (otherwise $I_d \notin C_w^c$, a contradiction). Finally, since $2\lfloor t/2 \rfloor + 1 \geq t$ for all $t \geq 3$, we obtain that at least t disjoint intervals are overlapped by I_a, which is in contradiction with the fact that G is $K_{1,t}$-free. Therefore, the color class $\mathcal{C}^c$ induces well a $K_{1,3}$-free interval graph, *ie.* a proper interval graph by Roberts theorem (*cf.* [13, 6]), and the whole correctness of the algorithm is established. □

Remark. In Algorithm ColorCliques, the assignment of colors is done according to the basic ordering $\{1, \ldots, \lfloor t/2 \rfloor\}$. The correctness holds by using any permutation of the set $\{1, \ldots, \lfloor t/2 \rfloor, 1, \ldots, \lfloor t/2 \rfloor\}$, repeated as many time as necessary to complete the assignment (the proof remains the same). Notably, this implies that there exists at least $(2t)!/2^t t!$ non-isomorphic partitions of a $K_{1,t}$-free interval graph into proper interval graphs. Note that determining the minimum value t for which G is $K_{1,t}$-free can be done in $O(n^2)$ time by computing the largest stable [7] contained in each interval of its representation $I_1, \ldots, I_n$.

Lemma 2. *Any interval graph $G = (V, E)$ admits a partition into less than $\lceil \log_3((n+1)/2) \rceil$ $K_{1,5}$-free interval subgraphs. Moreover, this partition is computed in linear time and space.*

Before giving the proof of the lemma, we need to establish this useful claim.

Claim. Any interval graph $G = (V, E)$ admits an open (resp. closed) interval representation such that every interval has positive integer endpoints lower than n (resp. $2n$). Moreover, this representation is computed in linear time and space.

Proof. Let $A = (a_{ij})$ be the maximal cliques-versus-vertices incidence matrix of G. A $(0,1)$-matrix has the consecutive 1's property for columns if its rows can be permuted in such a way that the 1's in each column occur consecutively. A well-known characterization of interval graphs is that the matrix A has the consecutive 1's property for columns and no more than n rows (Fulkerson-Gross 1965, *cf.* [6]). Thereby, consider a representation of A with the 1's consecutive in each column and for each $v \in V$, set $le(I_v) = \min\{i \mid a_{iv} = 1\}$ and $re(I_v) = \max\{i \mid a_{iv} = 1\}$. Clearly, the open interval representation $\{I_v\}_{v \in V}$ is such that every endpoint is in $\{1, \ldots, n\}$. This interval representation is correct because two intervals are intersecting if and only if their two corresponding vertices are connected. Computing the matrix A with consecutive 1's is done in $O(n+m)$ time and space [9]. Therefore, the complexity of the previous construction is linear. Finally, a closed interval representation is obtained from the previous open interval representation. Sort all the endpoints (left and right mixed) in the ascendant order. For $i = 1, \ldots, 2n$, assign to the i^{th} endpoint the value i and then redefine the n intervals as closed with their new endpoints in $\{1, \ldots, 2n\}$. Since the order on the endpoints is unchanged, the interval graph remains the same. Moreover, sorting $2n$ integers in $\{1, \ldots, n\}$ is done in $O(n)$ time using $O(n)$ space, which concludes the proof. □

Proof (of Lemma 2). According to the Claim, compute in linear time and space an open interval representation $I_1, \dots, I_n$ of G with endpoints in $\{1, \dots, n\}$ and denote by ℓ the maximum length of an interval ($\ell \leq n-1$). Then, partition the intervals according to their length into $\lceil \log_3((\ell+2)/2) \rceil$ subsets as follows: $\mathcal{I}_1$ contains the intervals of length $\{1,2,3,4\}$, $\mathcal{I}_2$ the intervals of length $\{5,6,\dots,16\}, \dots, \mathcal{I}_i$ the intervals of length $\{2.3^{i-1}-1, \dots, 2.3^i-2\}$. We affirm that each subset $\mathcal{I}_i$ induces a $K_{1,5}$-free interval graph. Indeed, the contrary implies that one interval of $\mathcal{I}_i$ contains properly three disjoint intervals whose sum of lengths is lower than $2.3^i - 4$, which is a contradiction (the minimum sum of three intervals is $3(2.3^{i-1}-1) = 2.3^i - 3$). Note that the proof remains correct by starting with a closed interval representation with endpoints in $\{1, \dots, 2n\}$ and partitioning such that each set $\mathcal{I}_i$ contains the intervals of length $\{4.3^{i-1}-3, \dots, 4.3^i-4\}$ for $i = 1, \dots, \lceil \log_3((\ell+4)/4) \rceil$ (here $\ell \leq 2n-1$). □

Remark. In fact, we can prove more generally that any interval graph $G = (V, E)$ admits a partition into $O(\log_t n)$ $K_{1,t+2}$-free interval subgraphs for any integer $t \geq 3$.

Proposition 1. *Any interval graph (resp. circular-arc graph) $G = (V, E)$ admits a partition into less than $2\lceil \log_3((n+1)/2) \rceil$ (resp. $2\lceil \log_3((n+1)/2) \rceil + 1$) proper interval subgraphs. Moreover, this partition is computed in linear time and space.*

Proof. The proof of the bound for interval graphs follows immediately the combination of Lemmas 2 and 1 (with $t = 5$). For circular-arc graphs, compute first a circular-arc representation of G in linear time and space [10]. Now, choose one point p on the circle and compute the set of vertices V^* corresponding to the arcs which contain p. By observing that V^* forms a clique and the subgraph induced by $V \setminus V^*$ is an interval graph, we obtain the desired bound for circular-arc graphs (any clique induces trivially a proper interval graph). □

The first half of Theorem 1 is established through the previous proposition, while the second is established via the next proposition.

Proposition 2. *For infinitely many r, the complete r-partite graph $H_r = (S_1 \cup \dots \cup S_r, E)$ with $|S_1| = 1, \dots, |S_r| = 3^{r-1}$ admits no partition into less than $\log_3(2n+1)$ proper interval subgraphs.*

Proof. An interval representation of the graph H_r is built by defining recursively the r stables $S_1, \dots, S_r$ as follows. The stable S_1 consists of one open interval of length 3^{r-1}. For all $i = 2, \dots, r$, the stable S_i is obtained by copying the stable S_{i-1} and subdivising each interval of this one into three open intervals of equal length (see Fig. 2 in Appendix A for an example of construction). The resulting stables $S_1, \dots, S_r$ induce well a complete r-partite graph. Note that the number of vertices of H_r is given by $(*)$ $n = \sum_{i=1}^{r} 3^{i-1} = (3^r - 1)/2$.

Since any stable induces trivially a proper interval graph, H_r admits a partition into r proper interval subgraphs. Now, using induction, we show that any minimum partition of H_r into proper interval subgraphs has the cardinality $p(H_r) = r$. First, one can easily verify that $p(H_1) = 1$ or $p(H_2) = 2$; then, the induction basis is $p(H_{i-1}) = i - 1$ for $i > 2$. Now, suppose that $p(H_i) < i$ and consider a partition of H_i into $i-1$ sets $\mathcal{I}_1, \ldots, \mathcal{I}_{i-1}$ of proper intervals. Without loss of generality, the single interval $I^* \in S_1$ belongs to $\mathcal{I}_1$. We claim that the intervals of $\mathcal{I}_1 \setminus I^*$ induce at most two disjoint cliques. In effect, the contrary implies the existence of an induced subgraph $K_{1,3}$ in $\mathcal{I}_1$ (with I^* as central vertex and one interval in each disjoint clique as leaves). According to this claim, at least one interval of S_2 and all the intervals stemming from its subdivision in $S_3, \ldots, S_i$ do not belong to $\mathcal{I}_1$. Clearly, such a set of intervals induces the graph H_{i-1} and by induction hypothesis, needs $i-1$ sets to be partitionned into proper interval subgraphs. However, only the $i-2$ sets $\mathcal{I}_2, \ldots, \mathcal{I}_{i-1}$ are available to realize that, which leads to a contradiction. This completes the induction by obtaining that $p(H_i) = i$ for $i > 2$. The equality $(*)$ is finally used to obtain $p(H_r) = \log_3(2n + 1)$. $\square$

Corollary 1. *For every $t \geq 3$, a $K_{1,t}$-free interval graph with at most $\lfloor (3t-4)/2 \rfloor$ vertices exists which admits no partition into less than $\lfloor \log_3(t-1) \rfloor + 1$ proper interval subgraphs.*

Proof. The graph H_r defined in Proposition 2 is clearly $K_{1,t}$-free for $t \in \{3^{r-1} + 1, \ldots, 3^r\}$. By simple calculation, we deduce that H_r has at most $\lfloor (3t-4)/2 \rfloor$ vertices and admits no partition into less than $\lfloor \log_3(t-1) \rfloor + 1$ proper interval subgraphs for $t \in \{3^{r-1} + 1, \ldots, 3^r\}$. $\square$

3 Applications to working schedules planning

A classical approximation. In this subsection, a classical algorithm is presented to approximate WSP with interval graphs. Here are two propositions, partially established in [5], which are behind its proof.

Proposition 3. *A minimum coloring of an interval graph $G = (V, E)$ such that the number $s(G)$ of stables consisting of only one vertex is as small as possible is computed in linear time and space.*

Proposition 4. *Let $G = (V, E)$ be an interval graph and k an integer. If G is colored such that each color is used at least k times, then G admits an optimal partition into $\lceil n/k \rceil$ stables of size at most k. Moreover, this partition is computed in linear time and space given the coloring in input.*

Algorithm 2-ApproxWSP
input: an interval graph $G = (V, E)$, an integer k;
output: a solution $\mathcal{S}$ to the WSP problem for G;
begin
  compute a minimum coloring $\mathcal{C} = \{S_1, \ldots, S_{\chi(G)}\}$ of G with $s(G)$ minimum;
  $\mathcal{S} \leftarrow \emptyset$;
  for each $S_i \in \mathcal{C}$ **do**

if $|S_i| < k$ **then** $\mathcal{C} \leftarrow \mathcal{C} \setminus \{S_i\}$, $\mathcal{S} \leftarrow \mathcal{S} \cup \{S_i\}$;
compute an optimal partition $\mathcal{S}_k$ of $\mathcal{C}$ into stables of size at most k;
$\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{S}_k$;
return $\mathcal{S}$;
end;

Theorem 2. *Algorithm 2-ApproxWSP achieves in linear time and space the asymptotic worst-case ratio* $2(k-1)/k$ *for the criterion* P*. Moreover, this worst-case ratio is tight.*

Proof. Omitted here (see Appendix B for details). □

Remark. A similar algorithm can be designed to approximate WSP for circular-arc graphs with worst-case ratio 3: first determine in linear time a coloring using less than 2 $\omega(G)$ colors and then use Proposition 4, which remains correct for circular-arc graphs, to find a solution to WSP.

A greedy for proper interval graphs. Here a greedy algorithm is presented which solves the WSP problem for proper interval graphs.

Algorithm GreedyProperWSP
input: a proper interval graph $G = (V, E)$, an integer k;
output: a solution $\mathcal{S}$ to the WSP problem for G;
begin
compute a proper interval representation $I_1, \dots, I_n$ of G;
order $I_1, \dots, I_n$ according to the left endpoints;
compute $\omega(G)$ and $\chi(G,k) \leftarrow \max\{\omega(G), \lceil n/k \rceil\}$;
$S_1 \leftarrow \cdots \leftarrow S_{\chi(G,k)} \leftarrow \emptyset$;
for i **from** 1 **to** n **do**
$j \leftarrow (i-1) \bmod \chi(G,k) + 1$, $S_j \leftarrow S_j \cup \{I_i\}$;
$\mathcal{S} \leftarrow \{S_1, \dots, S_{\chi(G,k)}\}$;
return $\mathcal{S}$;
end;

Computing an ordered proper interval representation of G is done in $O(n+m)$ time and space [3] and $\omega(G)$ is computed in $O(n)$ time [7]. Consequently, the algorithm runs in linear time and space.

Lemma 3. *The output solution* $\mathcal{S}$ *is* $(P|S)$*-optimal.*

Proof. First, we claim that the output stables $S_1, \dots, S_{\chi(G,k)}$ have a size at most k. According to the algorithm, the stables have the same size (to within one unity if n is not a multiple of k). Then, the existence of one stable of size strictly larger than k implies that $n > k\chi(G,k)$, a contradiction. Additionally, this establishes the (S)-optimality of $\mathcal{S}$. Now, suppose that two intervals I_u, I_v with $u < v$ are intersecting in the stable S_j for any $j \in \{1, \dots, \chi(G,k)\}$. By the algorithm, we have $u = j + \alpha\chi(G, k)$ and $v = j + \beta\chi(G, k)$ with $\alpha < \beta$. When the intervals are proper, the right endpoints have the same order as the left endpoints. Then, the intervals $I_u, I_{u+1}, \ldots, I_{v-1}, I_v$ include the portion $[l_v, r_u]$ of the real line and also induce a clique of size $v - u + 1 = (\beta - \alpha)\chi(G, k) + 1 \geq \chi(G, k) + 1$. Such a clique implies that $\omega(G) > \chi(G, k)$, which is a contradiction and the correctness of the solution $\mathcal{S}$ is entirely proved. To conclude, $\mathcal{S}$ is $(P|S)$-optimal because $\max\{\omega(G), \lceil n/k \rceil\}$ is a lower bound for $\chi(G, k)$. □

Lemma 4. *The output solution $\mathcal{S}$ is $(P|R)$-optimal.*

Proof (Sketch). The (P)-optimality of $\mathcal{S}$ is established by Lemma 2. Now, suppose that the set $S_1, \ldots, S_{\chi(G,k)}$ is not $(P|R)$-optimal. Define $S_1^*, \ldots, S_{\chi(G,k)}^*$ to be a $(P|R)$-optimal solution and g^* the minimum gap between two consecutive intervals of this solution. Remind that the intervals $I_1, \ldots, I_n$ are ordered according to the left endpoints and $I_{v,t}$ denotes the interval of rank t in the stable S_v^*. We claim that for all $i = 1, \ldots, n$, the interval $I_i \in S_u^*$ can be moved at the rank $t = \lfloor (i-1)/\chi(G, k) \rfloor + 1$ of the stable set S_v^* with $v = (i - 1) \bmod \chi(G, k) + 1$, without decreasing g^*. After such an operation, the resulting set $S_1^*, \ldots, S_{\chi(G,k)}^*$ coincide exactly with the solution $S_1, \ldots, S_{\chi(G,k)}$ of the greedy, which establishes its $(P|R)$-optimality. The claim is proved by an inductive process whose initial step is done as follows. If $I_1 \in S_u^*$ with $u \neq 1$, exchange the entire set of intervals of S_u^* with the one of S_1^*. Clearly, g^* is not deteriored (no gap is modified) and I_1 is correctly placed. Now, the inductive step is proved; the intervals $I_1, \ldots, I_{i-1}$ are considered to be correctly placed. The interval $I_i \in S_u^*$ shall be moved to the stable S_v^* if $u \neq v$. Then, two cases are distinguished.

<u>Case $u < v$</u> (see Fig. 2 in Appendix A): $S_u^* = \{I_{u,1}, \ldots, I_{u,t}, I_i, \ldots, I_{u,j}, \ldots\}$ and $S_v^* = \{I_{v,1}, \ldots, I_{v,t-1}, I_{v,t}, \ldots, I_{v,j}, \ldots\}$. By induction hypothesis, we get $re(I_{v,t-1}) \leq re(I_{u,t})$ and $le(I_i) \leq le(I_{v,t})$. Since $re(I_{u,t}) < le(I_i)$, we obtain the inequalities (i) $re(I_{v,t-1}) \leq re(I_{u,t}) < le(I_i) \leq le(I_{v,t})$ which allow us to redefine $S_u^* = \{I_{u,1}, \ldots, I_{u,t}, I_{v,t}, \ldots, I_{v,j}, \ldots\}$ and $S_v^* = \{I_{v,1}, \ldots, I_{v,t-1}, I_i, \ldots, I_{u,j}, \ldots\}$. Two gaps are changed: $le(I_i) - re(I_{u,t})$ in S_u^* becomes $le(I_{v,t}) - re(I_{u,t})$ and $le(I_{v,t}) - re(I_{v,t-1})$ in S_v^* becomes $le(I_i) - re(I_{v,t-1})$. According to (i), the new gaps are larger than the minimum of the two old ones.

<u>Case $u > v$</u> (see Fig. 3 in Appendix A): $S_u^* = \{I_{u,1}, \ldots, I_{u,t-1}, I_i, \ldots, I_{u,j}, \ldots\}$ and $S_v^* = \{I_{v,1}, \ldots, I_{v,t-1}, I_{v,t}, \ldots, I_{v,j}, \ldots\}$. Here induction hypothesis provide the inequalities (ii) $re(I_{v,t-1}) \leq re(I_{u,t-1}) < le(I_i) \leq le(I_{v,t})$ and we redefine $S_u^* = \{I_{u,1}, \ldots, I_{u,t-1}, I_{v,t}, \ldots, I_{v,j}, \ldots\}$ and $S_v^* = \{I_{v,1}, \ldots, I_{v,t-1}, I_i, \ldots, I_{u,j}, \ldots\}$. According to (ii), the two new gaps in S_u^* and S_v^* are still larger than the minimum of the two old ones.

The analysis of these two cases shows the correctness of the inductive step and completes the proof of the claim. □

Theorem 3. *Algorithm GreedyProperWSP determines in linear time and space $(P|S, R)$-optimal solutions to the problem WSP for proper interval graphs.*

The logarithmic approximation with sub-optima. According to the previous discussions, a new approximation algorithm is designed for WSP with interval graphs.

Algorithm log-ApproxWSP
input: an interval graph $G = (V, E)$, an integer k;
output: a solution $\mathcal{S}$ to the WSP problem for G;
begin
$\mathcal{S} \leftarrow \emptyset$;
if G is a proper interval graph **then** $\mathcal{S} \leftarrow$ GreedyProperWSP(G, k);
else
partition G into $B(n)$ proper interval subgraphs $G_1, \ldots, G_{B(n)}$;
for each subgraph G_i **do** $\mathcal{S} \leftarrow \mathcal{S} \cup$ GreedyProperWSP(G, k);
return $\mathcal{S}$;
end;

Theorem 4. *Algorithm* log-*ApproxWSP achieves in linear time and space the absolute worst-case ratio* $\min\{k, B(n)\}$ *with* $B(n) = 2\lceil \log_3((n+1)/2) \rceil$ *for the criterion* P *and guarantees that* $(P|S,R)$*-optima are reached in* $B(n)$ *subproblems. Moreover, the worst-case ratio is asymptotically tight.*

Proof. Correctness and complexity follow from Theorems 1 and 3, plus the fact that recognizing a proper interval graph is done in linear time and space [3]. To complete the proof, the worst-case ratio is established. If G is a proper interval graph then $\mathcal{S}$ is optimal. Otherwise, we have $|\mathcal{S}| = \sum_{i=1}^{B(n)} \chi(G_i, k)$. By using the inequalities $\sum_{i=1}^{B(n)} \chi(G_i, k) \leq n \leq k \cdot \chi(G, k)$ and $\sum_{i=1}^{B(n)} \chi(G_i, k) \leq \sum_{i=1}^{B(n)} \chi(G, k) \leq B(n) \cdot \chi(G, k)$, we obtain the result.

Finally, an interval graph G is given which tights asymptotically the ratio $\min\{k, B(n)\}$ with $B(n) = 2\lfloor \log_3((n+1)/2) \rfloor$ and $k = B(n)$. The complete proof is not detailed here; without loss of generality, we assume that n is a multiple of $B(n)$ and set $N(n) = n/B(n) - 1$. The interval graph is modeled by the following set of open intervals. For $i = 1, \ldots, B(n)/2$, take one interval $(1, 2.3^i - 1)$, one interval $(1, 2.3^{i-1})$, $N(n)$ intervals $(2.3^{i-1}, 4.3^{i-1} - 1)$ and $N(n)$ intervals $(4.3^{i-1} - 1, 2.3^i - 2)$ (see Fig. 1 above for an example of construction). Note that the endpoints are well in $\{1, \ldots, n\}$ and G is not a proper interval graph. In this case, one can verify that the approximation ratio of Algorithm log-ApproxCIG$_k$ is

$$\frac{|\mathcal{S}|}{\chi(G,k)} = \frac{(B(n)/2)(2N(n)+1)}{2(B(n)/2) + N(n) - 1} = B(n) \cdot \frac{n - B(n)/2}{n + B^2(n) - 2B(n)} \underset{n \to \infty}{\longrightarrow} B(n) = k.$$

□

Remark. Algorithm log-ApproxCIG$_k$ produces $(P|S,R)$-optimal solutions when G is a proper interval graph. Besides, in real-life situations [12], the minimum

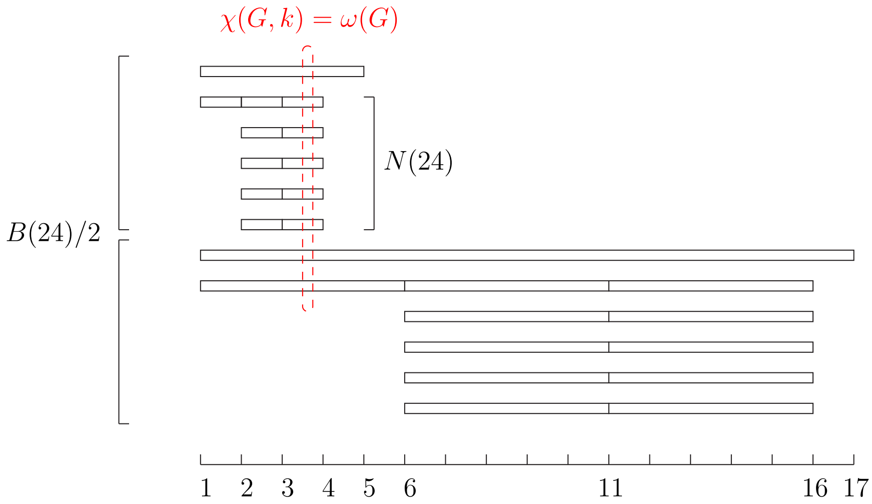

Fig. 1. An example of construction which tights the worst-case ratio with $n = 24$ ($k = B(24) = 4$, $N(24) = 5$): $\chi(G, k) = 8$ and $|\mathcal{S}| = 22$.

value t for which G is $K_{1,t}$-free is generally *small* (≤ 9). This allows direct partitionings into proper interval subgraphs by Algorithm ColorCliques and also the obtaining of *constant* worst-case ratios (≤ 4) for the criterion P. For example, for tasks of $1, 2, 3$ or 4 hours, we can obtain a 2-approximation and for tasks of $1, 2, \ldots, 8$ hours, a 3-approximation. Moreover, Algorithm log-ApproxWSP can be easily adapted for circular-arc graphs. In this case, its "real-life" worst-case ratio is nearly the same than the one obtained by the classical approach.

4 Conclusion

As a conclusion, we discuss some projections on the complexity of determining a minimum partition of a interval graph into proper interval subgraphs. In effect, answering to the mathematician has provided some hints for answering to the computer scientist.

First, we know now that a minimum partition of a $K_{1,5}$-free interval graph G into proper interval subgraphs is computed in linear time and space: if G is not a proper interval graph, then we can use Lemma 1 to partition G into 2 proper interval graphs (recognizing proper interval graph is done in linear time and space [3]). For $K_{1,6}$-free interval graphs (and also for arbitrary interval graphs), we conjecture that the problem is $\mathcal{NP}$-complete.

Finding a polynomial-time approximation algorithm with constant worst-case ratio for the problem seems to be difficult too. However, combining the previous remark with Lemma 2 enables us to design a linear-time approximation algorithm, similar to Algorithm log-ApproxWSP, which achieves the worst-case ratio $\ln n$ for this problem: if G is not a proper interval graph, then we can partition it into $\lceil \log_3((n+1)/2) \rceil$ $K_{1,5}$-free proper interval graphs (each of then are partitionned in linear time into a minimum number of proper interval subgraphs).

References

1. M.G. Andrews, M.J. Atallah, D.Z. Chen and D.T. Lee (2000). Parallel algorithms for maximum matching in complements of interval graphs and related problems. *Algorithmica* 26, 263–289.
2. H.L. Bodlaender and K. Jansen (1995). Restrictions of graph partition problems. Part I. *Theoretical Computer Science* 148, 93–109.
3. D.G. Corneil, H. Kim, S. Natarajan, S. Olariu and A. Sprague (1995). Simple linear time recognition of unit interval graphs. *Information Processing Letters* 55, 99–104.
4. D.G. Corneil, S. Olariu and L. Stewart (1998). The ultimate interval graph recognition algorithm ? In *Proc. 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 175–180, ACM Publications, New-York, NY.
5. F. Gardi (2003). Efficient algorithms for disjoint matchings among intervals and related problems. In *Proc. 4th International Conference on Discrete Mathematics and Theoretical Computer Science*, *LNCS* 2731, 168–180.
6. M.C. Golumbic (1980). *Algorithmic Graph Theory and Perfect Graphs.* Computer Science and Applied Mathematics Series, Academic Press, New-York, NY.
7. U.I. Gupta, D.T. Lee and J.Y.-T. Leung (1982). Efficient algorithms for interval graphs and circular-arc graphs. *Networks* 12, 459–467.
8. P.C. Fishburn (1985) *Interval Orders and Interval Graphs.* John Wiley & Sons, New-York, NY.
9. M. Habib, R. McConnel, C. Paul and L. Viennot (2000). Lex-BSF and partition refinement, with applications to transitive orientation, interval graph recognition and consecutive ones testing. *Theoretical Computer Science* 234, 59–84.
10. R. McConnel (2001). Linear time recognition of circular-arc graphs. In *Proc. 42nd Annual IEEE Symposium on Foundations of Computer Science*, pages 386–394, IEEE Computer Society Publications, Los Alamitos, CA.
11. S. Olariu (1991). An optimal greedy heuristic to color interval graphs. *Information Processing Letters* 37, 21–25.
12. Bamboo - Planification by Prologia - Groupe Air Liquide. http://prologianet.univ-mrs.fr/bamboo/bamboo_planification.html
13. F.S. Roberts (1978). *Graph Theory and its Application to the Problems of Society.* SIAM Publications, Philadelphia, PA.

Appendix A

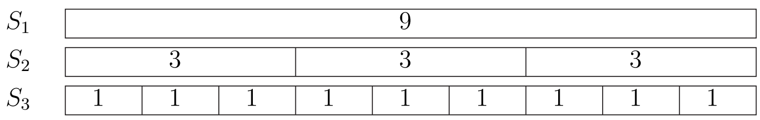

Fig. 2. The interval representation of G_3.

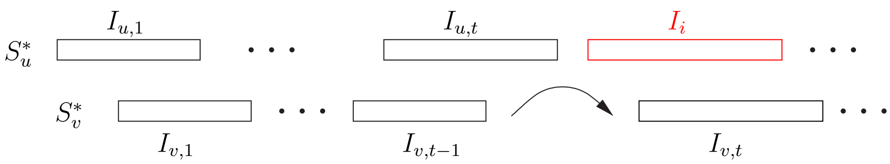

Fig. 3. The case $u < v$.

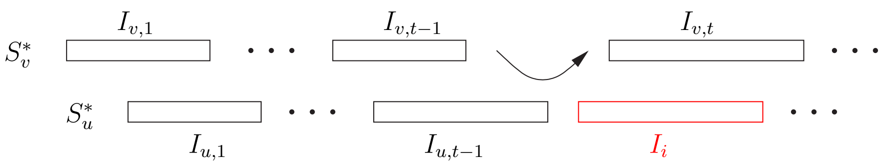

Fig. 4. The case $u > v$.

Appendix B

Proof (of Theorem 2). Correctness and complexity follow from Propositions 3 and 4. The worst-case ratio is established as follows. Denote by $\mathcal{C}_2$ the stables of size $\{2, \ldots, k-1\}$ in $\mathcal{C}$ and by $\mathcal{C}_k$ the stables of size at least k. The output solution $\mathcal{S}$ has the cardinality

$$|\mathcal{S}| = |\mathcal{C} \setminus \mathcal{C}_k| + \left\lceil \frac{\sum_{S_i \in \mathcal{C}_k} |S_i|}{k} \right\rceil \tag{1}$$

If $\mathcal{C}_k = \emptyset$ ($\chi(G,k) = \chi(G)$) or $\mathcal{C} \setminus \mathcal{C}_k = \emptyset$ ($\chi(G,k) = \lceil n/k \rceil$) then $\mathcal{S}$ is (P)-optimal. Otherwise, the following (in)equalities hold:

$$\sum_{S_i \in \mathcal{C}_k} |S_i| + \sum_{S_i \in \mathcal{C} \setminus \mathcal{C}_k} |S_i| = n \tag{2}$$

$$|\mathcal{C} \setminus \mathcal{C}_k| \leq \chi(G) - 1 \tag{3}$$

$$\chi(G,k) \geq \max\{\chi(G), \left\lceil \frac{n - s(G)}{k} \right\rceil\} \tag{4}$$

(2) is trivial and (3) ensues from the previous discussion. To prove (4), let $S_i = \{I_j\}$ be one of the $s(G)$ stables of size one. Since $\omega(G) = \chi(G)$, I_j belongs to any maximum clique of G. Therefore, $s(G)$ intervals cannot be matched in G, which implies that $\chi(G,k) \geq \lceil \frac{n-s(G)}{k} \rceil$. This inequality and the direct one $\chi(G,k) \geq \chi(G)$ yield (4). Now, by combining (2) and (4), we obtain

$$\frac{\sum_{S_i \in \mathcal{C}_k} |S_i|}{k} \leq \chi(G,k) + \frac{s(G) - \sum_{S_i \in \mathcal{C} \setminus \mathcal{C}_k} |S_i|}{k} \tag{5}$$

By the fact that $|S_i| = 1$ for every stable in $\mathcal{C} \setminus \{\mathcal{C}_2 \cup \mathcal{C}_k\}$ and (3), we have $s(G) - \sum_{S_i \in \mathcal{C} \setminus \mathcal{C}_k} |S_i| = -\sum_{S_i \in \mathcal{C}_2} |S_i| \leq -2(\chi(G) - 1)$. Hence, (5) becomes

$$\frac{\sum_{S_i \in \mathcal{C}_k} |S_i|}{k} \leq \chi(G,k) - \frac{2(\chi(G) - 1)}{k} \tag{6}$$

By integrating (3) and (6) into (1), we obtain finally $|\mathcal{S}| \leq \frac{2(k-1)}{k} \cdot \chi(G,k) + \frac{2}{k}$.

To conclude, the tightness of the worst-case ratio is shown. Set $n = kq$ with $q \geq 2$ and define the interval graph G as the union of the clique K_q (of size q) and $n - q$ isolated vertices. The first step of Algorithm 2-ApproxWSP can be done as follows: the vertices of K_q are uniformly distributed in q stables $S_1, \ldots, S_q$ where the isolated vertices are then placed in such a way that $|S_1| = kq - 2(q-1)$ and $|S_2| = \cdots = |S_q| = 2$ (note that $\chi(G) = n/k = q$). In this case, the output solution $\mathcal{S}$ has the size

$$|\mathcal{S}| = \left\lceil \frac{(k-2)\chi(G) + 2}{k} \right\rceil + \chi(G) - 1 \tag{7}$$

whereas an optimal one has simply the cardinality $\chi(G,k) = \chi(G) = n/k$. Therefore, by substitutions in (7), we obtain $|\mathcal{S}| \geq \frac{2(k-1)}{k} \cdot \chi(G,k) - \frac{k-2}{k}$ and the worst-case ratio $2(k-1)/k$ is asymptotically reached. □