Optimization Problems in Multiple-Interval Graphs

AYELET BUTMAN

Holon Institute of Technology

DANNY HERMELIN

University of Haifa

MOSHE LEWENSTEIN

Bar Ilan University

AND

DROR RAWITZ

Tel Aviv University

Abstract. Multiple-interval graphs are a natural generalization of interval graphs where each vertex may have more then one interval associated with it. We initiate the study of optimization problems in multiple-interval graphs by considering three classical problems: MINIMUM VERTEX COVER, MINIMUM DOMINATING SET, and MAXIMUM CLIQUE. We describe applications for each one of these problems, and then proceed to discuss approximation algorithms for them.

Our results can be summarized as follows: Let t be the number of intervals associated with each vertex in a given multiple-interval graph. For MINIMUM VERTEX COVER, we give a $(2 - 1/t)$-approximation algorithm which also works when a t-interval representation of our given graph is absent. Following this, we give a t^2-approximation algorithm for MINIMUM DOMINATING SET which

40

A preliminary version of this article was presented at the *ACM-SIAM Symposium on Discrete Algorithms* (*SODA*) [Butman et al. 2007].

The work on this article was done at the Caesarea Rothschild Institute.

D. Hermelin was supported by the Adams Fellowship of the Israel Academy of Sciences and Humanities. M. Lewenstein was supported by a German-Israel Foundation (GIF) Young Scientists program research grant.

Authors' addresses: A. Butman, Department of Computer Science, Holon Institute of Technology, Holon, Israel; e-mail: ayeletb@hit.ac.il; D. Hermelin, Department of Computer Science, University of Haifa, Haifa 31905, Israel, e-mail: danny@cri.haifa.ac.il; M. Lewenstein, Department of Computer Scince, Bar Ilan University, Ramat Gan 529000, Israil, e-mail: moshe@cs.biu.ac.il; D. Rawitz, Department of Electrical Engineering, Tel Aviv University, Tel Aviv 69978, Israel, e-mail: rawitz@eng.tau.ac.il.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2010 ACM 1549-6325/2010/03-ART40 $10.00
DOI 10.1145/1721837.1721856 http://doi.acm.org/10.1145/1721837.1721856

adapts well to more general variants of the problem. We then proceed to prove that MAXIMUM CLIQUE is **NP**-hard already for 3-interval graphs, and provide a $(t^2 - t + 1)/2$-approximation algorithm for general values of $t \geq 2$, using bounds proven for the so-called transversal number of t-interval families.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*Computations on discrete structures*; G.2.1 [**Discrete Mathematics**]: Combinatorics—*Combinatorial algorithms*; G.2.2 [**Discrete Mathematics**]: Graph Theory—*Graph algorithms*

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Approximation algorithms, minimum vertex cover, minimum dominating set, maximum clique, multiple-interval graphs, t-interval graphs

ACM Reference Format:

Butman, A., Hermelin, D., Lewenstein, M., and Rawitz, D. 2010. Optimization problems in multiple-interval graphs. ACM Trans. Algor. 6, 2, Article 40 (March 2010), 18 pages.
DOI = 10.1145/1721837.1721856 http://doi.acm.org/10.1145/1721837.1721856

1. *Introduction*

Interval graphs are one of the most popular and well-understood graph classes in algorithmic graph theory. They have numerous applications in various areas, most of which can be modeled by classical graph-theoretic problems. As an example, basic scheduling and storage problems translate to finding minimum colorings and clique covers in appropriate interval graphs [Golumbic 1980].

A natural generalization of interval graphs are *multiple-interval* graphs. A multiple-interval graph is an intersection graph of a family of *multiple intervals*, where a multiple-interval is the union of a finite number of disjoint intervals over the real line. Many problems that translate to interval graph problems extend naturally to multiple-interval graph problems. Scheduled tasks become multi-tasks, storage items require non-linear storage space, and so forth. However, in contrast to interval graphs, most of these problems turn out to be **NP**-hard.

In this article, we consider three classical optimization problems in multiple-interval graphs: MINIMUM VERTEX COVER, MINIMUM DOMINATING SET, and MAXIMUM CLIQUE. Since all three are **NP**-hard, our study focuses on designing approximation algorithms for these problems.

1.1. APPLICATIONS AND MOTIVATION.

Interval graphs and closely related relatives have been studied extensively due to their wide applicability in modeling many real life problems. Below, we consider three applications that correspond to the three problems we consider for multiple-interval graphs in this paper.

Loss Minimization. MAXIMUM INDEPENDENT SET is probably the most widely applicable problem in multiple-interval graphs [Bar-Yehuda et al. 2002; Bar-Yehuda and Rawitz 2005]. Usually, the applications for this problem are in scheduling or resource allocation scenarios. Since MINIMUM VERTEX COVER is the complement problem of MAXIMUM INDEPENDENT SET, onc can apply this problem in most of these scenarios, where one is interested in minimizing loss occurring due to unscheduled tasks or unfulfilled allocation requests.

For instance, Bar-Yehuda et al. [2002] describe an application for MAXIMUM INDEPENDENT SET in transmission of continuous-media data such as video on demand. Here, each multiple-interval represents multiple time segments in which a client requests data, allowing him to see a movie for instance, with intermediate breaks.

Two client requests are in conflict if there is some time period in which they overlap. An optimal schedule therefore corresponds to the maximum (weight) independent set in the corresponding multiple-interval graph. In terms of loss minimization, the minimum loss is obtained by not supplying the requests which correspond to the minimum weight vertex cover in the corresponding multiple-interval graph. Another application mentioned by Bar-Yehuda et al. [2002] is scheduling RAM and processor time for multiple real-time programs on multi-tasking systems. Here, loss minimization corresponds to the minimum number of programs to be removed from the schedule so that all remaining programs can run concurrently. We mention also that loss minimization for 1-interval requests, where intervals have an additional demand attribute associated with them, has been considered by Bar-Noy et al. [2001].

Employee Monitoring. Consider the following scenario: The sales manager of the ACME department store chain wants to monitor the salespersons at one of his stores, since sales are not looking too good this year. For this, he would like to assemble a group of monitor employees from the entire ACME chain, whose job is to inspect the salespersons of the store during their work shift. Now, at the beginning of each week he is handed the working schedule of all the salespersons in the store, in addition to the working schedule of the monitoring employees. Each schedule is represented by a multiple-interval which corresponds to multiple shifts in the week, that is, multiple time segments in which the salesperson or monitor can be at the store. The manager seeks to find the minimum number of monitor employees needed to inspect each one of salespersons at least once during the week, so the remaining monitor employees can be assigned monitor duties in other stores of the ACME chain. If we assume that for a salesperson to be inspected by a monitor, it is enough that they are both together in the store for some time period during the week, the problem above translates to MINIMUM DIRECTED DOMINATING SET. Furthermore, if the monitor employees are trusted salespersons that work at the store under examination, the problem translates to MINIMUM DOMINATING SET.

Communication Clique. Suppose you want to organize an international workshop of computer science specialists. In your community, you have many people fluent in many different languages. You wish to invite as many specialists as possible, but you want all them to be able to communicate together since you're on a tight schedule and anxious to obtain the long anticipated breakthrough in your research. If we consider the real line as a finite discrete set of points, where each point represents a different langauge, we can assign a multiple-interval (where each interval is a single point) to each specialist in the community, according to the multiple languages he is fluent in. The problem above then translates to MAXIMUM CLIQUE in the corresponding multiple-interval graph. One can imagine that the scenario described above appears in other real-life situations, for example, servers and communication protocols, transmitters and frequency intervals, and so forth.

1.2. OUR RESULTS. We initiate the study of combinatorial optimization problems on multiple-interval graphs. As mentioned above, our study focuses on three classical problems: MINIMUM VERTEX COVER, MINIMUM DOMINATING SET, and MAXIMUM CLIQUE. Below, we briefly describe our results for each of these problems. We use the term t-interval to refer to a multiple-interval which is the union of t disjoint intervals.

For MINIMUM VERTEX COVER, we give an approximation algorithm with ratio equalling $(2 - 1/t)$. This algorithm consists of two phases. The first phase is a clean-up phase that is based on the *local ratio technique*. At the end of this phase, the remaining t-interval graph is *hereditarily sparse*, that is, each of its induced subgraphs has a bounded average degree. In the second phase we apply known techniques for computing vertex covers in hereditarily sparse graphs. We note that our algorithm works even when a t-interval representation of the input graph is absent.

For MINIMUM DOMINATING SET, we present a t^2-approximation algorithm which also applies for the more general case in which we are given two subsets: a subset $\mathcal{B} \subseteq \mathcal{F}$ that contains t-intervals that should be dominated, and a subset $\mathcal{R} \subseteq \mathcal{F}$ that contains t-intervals that may be used to dominate the t-intervals in $\mathcal{B}$. This more general version of the problem is equivalent to MINIMUM DIRECTED DOMINATING SET, since we do not require $\mathcal{R}$ and $\mathcal{B}$ to be disjoint. Our algorithm can also be applied when $\mathcal{B}$ contains t_B-intervals and $\mathcal{R}$ contains t_R-intervals, and in this case it computes $(t_B \cdot t_R)$-approximate solutions. We note that this version of the problem contains problems such as MINIMUM RECTANGLE TRANSVERSAL ($t_B = 2, t_R = 1$, and the intervals in $\mathcal{R}$ are points), MINIMUM t-INTERVAL TRANSVERSAL ($t_B = t$, $t_R = 1$ and the intervals in $\mathcal{R}$ are points), and MINIMUM SET COVER with at most t blocks of consecutive ones in each column of the constraint matrix ($t_B = 1$, $t_R = t$, and the intervals in $\mathcal{B}$ are points) as special cases. In fact, our algorithm extends the 2-approximation algorithm for MINIMUM RECTANGLE TRANSVERSAL from Gaur et al. [2002], the t-approximation algorithm for MINIMUM t-INTERVAL TRANSVERSAL that is implied by combining [Hassin and Segev 2005; Hassin and Tamir 1991] and the t-approximation algorithm for MINIMUM SET COVER with at most t blocks of consecutive ones from Hochbaum and Levin [2006].

MAXIMUM CLIQUE differs from the two problems above in that it was previously unknown whether the problem was hard, even for large values of t. We prove that MAXIMUM CLIQUE is already **NP**-complete for the case of $t = 3$. In addition, we present a $(t^2-t+1)/2$-approximation algorithm for the problem, when $t \geq 2$, using recent bounds proven for the transversal number of t-interval families in Kaiser [1997].

1.3. RELATED WORK. Multiple-interval graphs have been studied extensively from the graph-theoretic aspect. We briefly list some of the main results. The class of graphs with maximum degree Δ are $\lceil(\Delta + 1)/2\rceil$-interval graphs [Griggs and West 1980], while the complete bipartite graph $K_{m,n}$ is a $\lceil(mn + 1)/(m + n)\rceil$-interval graph [Trotter and Harary 1978]. Every graph with n vertices is a $\lceil(n + 1)/4\rceil$-interval graph, and the complete bipartite graph $K_{\lfloor n/2\rfloor,\lceil n/2\rceil}$ is an extremal example of this [Griggs 1979]. The class of planar graphs is a subclass of 3-interval graphs [Scheinerman and West 1983]. Finally, the problem of determining whether a given graph is t-interval is **NP**-complete for $t \geq 2$ [West and Shmoys 1984].

Many variants of multiple-interval covering problems have been studied by the combinatorial optimization and discrete geometry communities. Hochbaum and Levin [2006] studied the problem of covering points by t-intervals. The inverse problem of covering t-intervals by points, also called MINIMUM t-INTERVAL TRANSVERSAL, has been studied along with many of its variants in Gaur et al. [2002], Hassin and Megiddo [1991], Hassin and Segev [2005], and Hassin and Tamir [1991]. Obtaining upper bounds for the transversal number of any t-interval family has been considered by Alon [1998], Gyráfás and Lehel [1985], Kaiser [1997], and Tardos [1995].

Computational biology is another field where multiple-interval graph problems have been recently considered. For instance, Bafna et al. [1996] studied the problem of finding the maximum weight subset of non-overlapping local alignments between two genomic sequences. This problem translates to finding a maximum weight independent set in a restricted subclass of 2-interval graphs. In Aumann et al. [2005] studied another restricted subclass of t-interval graphs in the context of high throughput genotyping. In Blin et al. [2004], Crochemore et al. [2005], and Vialette [2004], 2-intervals were used to model secondary structure of RNA sequences, and secondary structure prediction scenarios were modeled by variants of the MAXIMUM INDEPENDENT SET problem in 2-interval graphs.

Classical optimization problems have been considered on several geometric intersection graphs. The first results of these type are now part of the classic literature [Brandstädt et al. 1999; Golumbic 1980; McKee and McMorris 1999]. Recently, this line of research has been extended to consider intersection graphs of various types of geometric objects, for example, intersection graphs of discs or ellipses in the plane [Clark et al. 1990; Erlebach et al. 2005; Hunt et al. 1998], intersection graphs of axis parallel rectangles [Agarwal et al. 1998; Berman et al. 2001], and intersection graphs of general fat objects [Chan 2003].

Finally, probably the most relevant work to ours is that of Bar-Yehuda et al. [2002] who studied the MAXIMUM INDEPENDENT SET problem in t-interval graphs. They gave a $2t$-approximation algorithm for this problem, in addition to a $2t$-approximation algorithm for the MINIMUM COLORING problem. We mention also Bar-Yehuda and Rawitz [2005], where a more general variant of MAXIMUM INDEPENDENT SET has been studied.

1.4. BASIC NOTATIONS AND TERMINOLOGY. We next briefly discuss notation and terminology that will be used throughout the article.

Let $i_1, i_2, \ldots, i_t$ be t disjoint closed intervals of the real line. The *t-interval* $I = (i_1, i_2, \ldots, i_t)$ is the union of these t intervals, i.e., $I = \bigcup_{j=1}^{t} i_j$. Given a pair of t-intervals $I = (i_1, i_2, \ldots, i_t)$ and $J = (j_1, j_2, \ldots, j_t)$, these two t-intervals *intersect* if they share a common point, i.e., $(\bigcup_{k=1}^{t} i_k) \cap (\bigcup_{\ell=1}^{t} j_\ell) \neq \emptyset$. A pair of nonintersecting t-intervals is said to be *disjoint*.

Let $\mathcal{F} = \{I_1, \ldots, I_n\}$ be a family of t-intervals. The *underlying family of intervals* of $\mathcal{F}$, denoted $\mathcal{I}(\mathcal{F})$, is the family of all intervals that compose the t-intervals in $\mathcal{F}$. That is, $\mathcal{I}(\mathcal{F}) = \{i \in \{i_1, i_2, \ldots, i_t\} \mid I = (i_1, i_2, \ldots, i_t) \in \mathcal{F}\}$. The *intersection graph* $\Omega_{\mathcal{F}}$ of $\mathcal{F}$, is a graph with a one-to-one correspondence between its vertices and $\mathcal{F}$ such that two vertices are connected in $\Omega_{\mathcal{F}}$ if their corresponding t-intervals in $\mathcal{F}$ intersect. We say that $\Omega_{\mathcal{F}}$ is a *t-interval graph* to emphasize that it is an intersection graph of family of t-intervals for some given $t \in \mathbb{N}^+$. A t-interval family $\mathcal{F}$ is a *t-interval representation* of a graph G if $G = \Omega_{\mathcal{F}}$.

We are concerned with vertex covers, dominating sets, and cliques in t-interval graphs. In terms of t-interval families, a subset $\mathcal{C} \subseteq \mathcal{F}$ is a cover of $\mathcal{F}$ if $\mathcal{F} \setminus \mathcal{C}$ is pairwise disjoint. A subset $\mathcal{D}$ of a t-interval family $\mathcal{F}$ is a dominating set of $\mathcal{F}$ if for any t-interval in $\mathcal{F} \setminus \mathcal{D}$ there is a t-interval in $\mathcal{D}$ which intersects it. A subset $\mathcal{K} \subseteq \mathcal{F}$ is called a *clique* if it is pairwise intersecting. Given a weight function $w : \mathcal{F} \to \mathbb{Q}^+$, the MINIMUM VERTEX COVER and MINIMUM DOMINATING SET problems in t-interval graphs ask to find the minimum weight cover and dominating set of $\mathcal{F}$, and the MAXIMUM CLIQUE problem asks to find the maximum weight clique of $\mathcal{F}$. A subset $\mathcal{C} \subseteq \mathcal{F}$ is an *α-approximate cover* of $\mathcal{F}$, if $\mathcal{C}$ is a cover of $\mathcal{F}$ with $w(\mathcal{C}) \leq \alpha \cdot w(\mathcal{C}_{opt})$, where $\mathcal{C}_{opt}$ is the minimum weight cover of $\mathcal{F}$. We define α-approximate dominating sets of $\mathcal{F}$ similarly. An *α-approximate clique* of $\mathcal{F}$ is a subset $\mathcal{K} \subseteq \mathcal{F}$ with $w(\mathcal{K}) \geq \alpha \cdot w(\mathcal{K}_{opt})$, where $\mathcal{K}_{opt}$ is the maximum weight clique of $\mathcal{F}$.

2. *Minimum Vertex Cover*

In the following section, we consider the MINIMUM VERTEX COVER problem for t-interval graphs. Recall that $2t - 1$ bounded degree graphs are t-interval graphs [Griggs and West 1980], and so MINIMUM VERTEX SET is **APX**-hard in t-interval graphs with $t \geq 2$, by the **APX**-hardness results given in Papadimitriou and Yannakakis [1991] for bounded degree graphs. We present an approximation algorithm with performance ratio equalling $2 - 1/t$.

The general outline of our algorithm is as follows. We first initiate a cleaning phase on $\mathcal{F}$. At the end of this phase, we remain with a family of t-intervals $\mathcal{F}' \subseteq \mathcal{F}$ which has the following convenient property: There are no three pairwise intersecting intervals in the underlying interval family of $\mathcal{F}'$, or in other words, $\Omega_{\mathcal{I}(\mathcal{F}')}$ does not contain a clique of size three. We call t-interval families with this property *flat*. Flat t-interval families are convenient for our purposes since their intersection graphs are *hereditarily sparse* (every induced subgraph has a bounded average degree), and we can apply known techniques for these types of graphs. The crux is therefore in the cleaning phase. We show that we can clean $\mathcal{F}$ in order to obtain a flat subset $\mathcal{F}' \subseteq \mathcal{F}$ in such a way that if $\mathcal{C}' \subseteq \mathcal{F}'$ is an α-approximate cover of $\mathcal{F}$ then $\mathcal{C}' \cup (\mathcal{F} \setminus \mathcal{F}')$ is a $\max\{\alpha, 1.5\}$-approximate cover of $\mathcal{F}$. Hence, using the known algorithm for hereditarily sparse graphs, we obtain our desired $2-1/t$ ratio.

2.1. THE CLEANING PHASE.

The cleaning phase is based on the Local-Ratio technique [Bar-Yehuda and Even 1985] which in turn is based on the Local-Ratio Theorem. In our terms, this theorem is stated as follows.

THEOREM 2.1 (LOCAL RATIO [BAR-YEHUDA AND EVEN 1985]). *Let $\mathcal{F}$ be a family of t-intervals and let w, w_1, and w_2 be weight functions for $\mathcal{F}$ such that $w = w_1 + w_2$. Then, if $\mathcal{C} \subseteq \mathcal{F}$ is an α-approximate cover for $\mathcal{F}$, both with respect to w_1 and with respect to w_2, then $\mathcal{C}$ is also an α-approximate cover with respect to w.*

A typical local ratio algorithm is recursive. In each recursive step, the algorithm first collects all zero elements to its solution. It then defines a weight function w_1 in such a way that $w_2 = w - w_1$ still assigns non-negative weights to all elements, and at least one element with non-zero weight with respect to w will get zero weight with respect to w_2. The algorithm then recursively solves the instance of the problem with w_2 as the given weight function, and fixes the returned solution so it will be a good approximate solution with respect to both w_1 and w_2. By the Local-Ratio theorem, this solution is guaranteed to be a good approximation with respect to w as well.

In Figure 1, we present algorithm LR-Cover, which applies the Local-Ratio technique for cleaning $\mathcal{F}$. A similar cleaning phase was used in Bar-Yehuda and Even [1985] in order to get rid of short odd cycles in the input graph. In its recursive basis, algorithm LR-Cover invokes algorithm Flat-Cover which specializes in flat

Algorithm LR-Cover($\mathcal{F}, w$)

Data: A set of t-intervals $\mathcal{F}$ and a weight function $w : \mathcal{F} \to \mathbb{Q}^+$.
Result: A cover $\mathcal{C}$ of $\mathcal{F}$.
begin
  1. **if** $\mathcal{F}$ is flat **then return** Flat-Cover($\mathcal{F}, w$).
  2. Select an endpoint p with $|\mathcal{K}_p| \geq 3$, where $\mathcal{K}_p = \{I \in \mathcal{F} \,|\, p \in I\}$.
  3. Select $I_0 \in \mathcal{K}_p$ with $w(I_0) = \min_{I \in \mathcal{K}_p} w(I)$.
  4. Define $w_1(I) = \begin{cases} w(I_0) & I \in \mathcal{K}_p, \\ 0 & \text{otherwise} \end{cases}$.
  5. Define $w_2 = w - w_1$.
  6. $\mathcal{F}^+ \leftarrow \{I \in \mathcal{F} \,|\, w_2(I) > 0\}$.
  7. $\mathcal{C} \leftarrow$ LR-Cover($\mathcal{F}^+, w_2$).
  8. $\mathcal{C} \leftarrow \mathcal{C} \cup \{I \in \mathcal{F} \,|\, w_2(I) = 0\}$.
  return $\mathcal{C}$.
end

FIG. 1. Algorithm LR-Cover.

t-interval families. We may assume with out loss of generality that the initial weight function w is positive.

LEMMA 2.1. *Suppose that algorithm Flat-Cover is an α-approximation algorithm for the* MINIMUM VERTEX COVER *problem in flat t-interval graphs. Then algorithm LR-Cover is a* $\max\{\alpha, 1.5\}$*-approximation algorithm for* MINIMUM VERTEX COVER *in general t-interval graphs.*

PROOF. First, observe that at each recursive call of LR-Cover, the size of $\mathcal{F}$ decreases at least by one, since the t-interval I_0 selected at Step 4 is excluded (along with possibly other t-intervals) from the next recursive call. Hence, LR-Cover invokes Flat-Cover after $\ell \leq n$ recursive calls, and is guaranteed to terminate assuming Flat-Cover terminates as well. The proof is by induction on the recursive calls of LR-Cover. At the ℓth recursive call, the lemma follows by the assumption that Flat-Cover is an α-approximation algorithm for flat families of t-intervals. Consider therefore, the ith recursive call of LR-Cover, $i \leq \ell$, and assume the lemma follows for any jth call, $i < j \leq \ell$.

Let $(\mathcal{F}, w)$ be the instance at the ith recursive call, and let $\mathcal{C} \subset \mathcal{F}$ be the family of t-intervals computed at step 7. Set $\beta = \max\{\alpha, 1.5\}$. By the inductive hypothesis, $\mathcal{C}$ is a β-approximate cover for $\mathcal{F}^+$ with respect to w_2. Step 8 ensures that $\mathcal{C}$ is also a cover for $\mathcal{F}$, since all t-intervals in $\mathcal{F} \setminus \mathcal{F}^+$ are added to $\mathcal{C}$. Also, after step 8, $\mathcal{C}$ remains β-approximate with respect to w_2, since only t-intervals with zero w_2-weight have been added to $\mathcal{C}$ at this step. We argue that after step 8, $\mathcal{C}$ is also β-approximate with respect to w_1.

Let opt_{w_1} be the weight of the optimal cover of $\mathcal{F}$ with respect to w_1, and let $\mathcal{K}_p$ be the subset of t-intervals defined at step 2. Set ε to be the weight of the t-interval I_0 selected at step 3. Since $\mathcal{K}_p$ is pairwise intersecting, any cover for $\mathcal{F}$ must include at least $|\mathcal{K}_p| - 1$ t-intervals of $\mathcal{K}_p$. Hence, since $w(I) = \varepsilon$ for any $I \in \mathcal{K}_p$, we have $\varepsilon(|\mathcal{K}_p| - 1) \leq opt_{w_1}$. On the other hand, $\sum_{I \in \mathcal{F}} w_1(I) = \varepsilon|\mathcal{K}_p|$, and so $\sum_{I \in \mathcal{C}} w_1(I) \leq \varepsilon|\mathcal{K}_p|$. It then follows that

$$\frac{\sum_{I \in \mathcal{C}} w_1(I)}{opt_{w_1}} \leq \frac{\varepsilon|\mathcal{K}_p|}{\varepsilon(|\mathcal{K}_p| - 1)} \leq \frac{3}{2} \leq \beta,$$

where the second inequality follows from the fact that $|\mathcal{K}_p| \geq 3$.

We have shown that at any recursive call of LR-Cover, the cover $\mathcal{C}$ returned is β-approximate both with respect to w_1 and w_2. Applying the Local-Ratio Theorem, we obtain that $\mathcal{C}$ is β-approximate with respect w at any recursive call, and we are done. □

2.2. FLAT t-INTERVAL FAMILIES. A graph is *hereditarily sparse* if each one of its induced subgraphs has a bounded average degree. The following lemma proves that the intersection graph of any flat t-interval family is hereditarily sparse.

LEMMA 2.2. *Let $\mathcal{F}$ be a flat t-interval family, $|\mathcal{F}| = n$, and let $G = \Omega_{\mathcal{F}}$ be the intersection graph of $\mathcal{F}$. Then, $|E(G)| \leq tn - 1$.*

PROOF. Consider the interval graph $G^* = \Omega_{\mathcal{I}(\mathcal{F})}$, the intersection graph of the underlying set of intervals of $\mathcal{F}$. Since any edge of G corresponds to at least one edge of G^*, we have $|E(G)| \leq |E(G^*)|$. Furthermore, since any vertex in G corresponds to at most t vertices of G^*, we have $|V(G^*)| \leq t|V(G)|$. To complete the proof, we argue that $|E(G^*)| \leq |V(G^*)| - 1$ by showing that G^* does not contain any cycles. To see this, note that G^* is an interval graph, and as such it is also chordal [Golumbic 1980]. Hence, if it contains a cycle, it also contains a clique of size three, contradicting the fact that $\mathcal{F}$ is flat. Combining all inequalities together, we get

$$|E(G)| \leq |E(G^*)| \leq |V(G^*)| - 1 \leq t|V(G)| - 1 = tn - 1,$$

and the lemma follows. □

Hochbaum [1983] presented a $(2-2/k)$-approximation algorithm for MINIMUM VERTEX COVER in graphs that can be colored with k colors. Her algorithm consists of two phases. First, it invokes an algorithm by Nemhauser and Trotter [1975] that partitions the vertex set into three sets V_1, V_2, and V_3 which satisfy the following three conditions:

(1) If a set $C \subseteq V_3$ is a vertex cover of $G[V_3]$ then $V_1 \cup C$ is a vertex cover of G.
(2) There exists an optimal vertex cover C_{opt} of G with $V_1 \subseteq C_{opt}$ and $V_2 \cap C_{opt} = \emptyset$.
(3) V_3 is a 2-approximate vertex cover for $G[V_3]$.

After applying this partitioning, Hochbaum's algorithm colors $G[V_3]$ with k colors. The computed solution is $V_1 \cup V_3 \setminus V_h$ where $V_h \subseteq V_3$ is the color set of maximum weight. An easy analysis shows that the performance ratio of Hochbaum's algorithm is indeed $2-2/k$, due to properties of the Nemhauser and Trotter partitioning, and due to the fact that $w(V_h) \geq w(V_3)/k$.

LEMMA 2.3. *If $\mathcal{F}$ is a flat t-interval family then one can find a $(2 - 1/t)$-approximate cover for $\mathcal{F}$ in polynomial time.*

PROOF. Let $\mathcal{F}$ be a flat t-interval family, and let $G = \Omega_{\mathcal{F}}$. By Lemma 2.2, we can order the vertices of G as $(v_1, \ldots, v_n)$, where each v_i has at most $2t - 1$ neighbors v_j with $i < j$. Hence, the greedy algorithm, which colors the vertices from v_n to v_1 and adds a new color only when necessary, uses at most $2t$ colors. It follows that using Hochbaum's algorithm, we can find an $(2 - 1/t)$-approximate cover for $\mathcal{F}$. □

By invoking Lemma 2.1 with the algorithm promised by Lemma 2.3 as algorithm Flat-Cover, we obtain the main result of this section.

THEOREM 2.2. *The* MINIMUM VERTEX COVER *problem in t-interval graphs can be approximated in polynomial time within a factor of* $2 - 1/t$.

It is worth mentioning that our algorithm works even when a t-interval representation of G is absent. In this case, the cleaning phase can be implemented (using more time) to clean all triangles from G. Since a triangle-free t-interval graph is flat, the second step can be carried out in exactly the same way.

3. *Minimum Dominating Set*

In the following section, we study the MINIMUM DOMINATING SET problem in t-interval graphs. This problem is known to be solvable in polynomial time for $t = 1$ and it is **APX**-hard for $t \geq 2$ via Griggs and West [1980] and Papadimitriou and Yannakakis [1991]. We present a t^2-approximation algorithm for t-interval graphs that is based on a reduction to the case of $t = 1$ (interval graphs), and a primal-dual algorithm which computes optimal solutions for MINIMUM DOMINATING SET in interval graphs. We also show that the analysis of our algorithm is essentially tight.

In this section, we actually need to solve a slightly more general variant of MINIMUM DOMINATING SET in which we are given two families of t-intervals, the red family $\mathcal{R} = \{I_1, \ldots, I_n\}$ and the blue family $\mathcal{B} = \{J_1, \ldots, J_m\}$, and the requirement is to find a minimum weight subset of $\mathcal{R}$ which dominates $\mathcal{B}$. We do not require the two families to be disjoint, and so if $\mathcal{R} = \mathcal{B}$ our variant reduces to the original MINIMUM DOMINATING SET problem. Naturally, we assume that for every blue t-interval in $\mathcal{B}$ there is at least one red t-interval in $\mathcal{R}$ which dominates it. Note that this more general version of the problem is equivalent to MINIMUM DIRECTED DOMINATING SET.

As mentioned above, our approximation algorithm consists of two parts. First, we present an algorithm based on the primal-dual method [Bar-Yehuda and Even 1981] that computes optimal solutions for the case of $t = 1$ in which both the red family and the blue family simply contain intervals. Afterwards, we present a t^2-approximation algorithm for the case of t-interval graphs using a reduction to the case of $t = 1$. It is important to note that the reduction is based on linear programming and this means that we cannot use any algorithm that solves the extended version of MINIMUM DOMINATING SET in interval graphs. The algorithm we use must supply a solution that is optimal with respect to the LP-relaxtion of the problem.

3.1. LINEAR PROGRAMMING FORMULATION.

We next briefly describe linear programming terminology that is essential for describing our algorithm. We assume basic knowledge of linear programming. Readers unfamiliar with the subject are referred to Karloff [1991].

Let $x(I)$ denote a real variable associated with the red t-interval $I \in \mathcal{R}$. Our generalized variant of MINIMUM DOMINATING SET can be formalized using the following linear integer program.

$$\begin{aligned} \min \quad & \sum_{I \in \mathcal{R}} w(I)x(I) \\ \text{s.t.} \quad & \sum_{I : J \cap I \neq \emptyset} x(I) \geq 1 \quad \forall J \in \mathcal{B} \\ & x(I) \in \{0, 1\} \qquad \forall I \in \mathcal{R} \end{aligned} \tag{DS}$$

where $x(I) = 1$ if I is in the dominating set. The linear relaxation of DS is obtained by replacing the integrality constraints by: $x(I) \geq 0$, for every $I \in \mathcal{R}$. We denote the linear relaxation of DS by LP-DS. The *integrality gap* of DS is the ratio between the optimal solutions of DS and LP-DS.

The *dual* program of LP-DS is:

$$\begin{array}{lll} \max & \sum_{J \in \mathcal{B}} y(J) & \\ \text{s.t.} & \sum_{J : I \cap J \neq \emptyset} y(J) \leq w(I) & \forall I \in \mathcal{R} \\ & y(J) \geq 0 & \forall J \in \mathcal{B} \end{array}$$

Here, the variables $y(J)$ correspond to blue t-intervals $J \in \mathcal{B}$. LP-DS is regarded as the *primal* program with respect to its dual. We will use x and y to denote primal and dual solution vectors respectively. A primal or dual solution is said to be *feasible* if it is indeed subject to all its corresponding constraints. It is known that for any pair of feasible primal-dual solutions x and y we have $\sum_{I \in \mathcal{R}} w(I)x(I) \geq \sum_{J \in \mathcal{B}} y(J)$, and that x and y are both optimal if and only if $\sum_{I \in \mathcal{R}} w(I)x(I) = \sum_{J \in \mathcal{B}} y(J)$.

We will the need the so-called *complementary slackness conditions*. For LP-DS, these conditions are stated as follows:

—The *primal conditions*: $\forall I \in \mathcal{R} : x(I) > 0 \Rightarrow \sum_{J : I \cap J \neq \emptyset} y(J) = w(I)$.

—The *dual conditions*: $\forall J \in \mathcal{B} : y(J) > 0 \Rightarrow \sum_{I : J \cap I \neq \emptyset} x(I) = 1$.

A pair of feasible primal-dual solutions x and y are both optimal if and only if they both satisfy the complementary slackness conditions.

3.2. ALGORITHM FOR INTERVAL GRAPHS. We next present a primal-dual algorithm that computes optimal dominating sets for 1-interval graphs. Our algorithm starts with a pair of primal-dual solutions $x = 0$ and $y = 0$. It has two phases. In the first, it gradually converts the primal solution to a feasible solution, while maintaining its integrality and the feasibility of the dual solution. The second phase is a reverse deletion phase, where the algorithm converts the primal solution to an optimal one, by assuring that the complementary slackness conditions are met.

Figure 2 depicts our primal dual algorithm PD-Dominate. We assume that $x = 0$ and $y = 0$ are the initial primal-dual solution vectors. Furthermore, we assume that the blue intervals $\mathcal{B} = J_1, \ldots, J_m$ are ordered in nondecreasing order of right endpoints.

LEMMA 3.1. *Algorithm PD-Dominate computes an optimal dominating set $\mathcal{D} \subseteq \mathcal{R}$ of $\mathcal{B}$.*

PROOF. The pair of primal and dual solutions x and y are both feasible by construction. To prove the lemma, we show that x is optimal by showing that both x and y satisfy the complementary slackness conditions.

First, by the construction of the primal solution, a primal variable is set to 1 only if its corresponding dual constraint is tight. Hence, the primal conditions are satisfied. To finish the proof, we need to show that the dual conditions are also satisfied, namely, that if $y(J) > 0$ then $\sum_{I : J \cap I \neq \emptyset} x(I) = 1$, for every $J \in \mathcal{R}$. Equivalently, we show that after the reverse deletion phase, if $y(J) > 0$, then there is exactly one red interval I that dominates J with $x(I) = 1$.

Algorithm PD-Dominate($\mathcal{R}, \mathcal{B}, w$)

Data: Two families of intervals $\mathcal{R} = \{I_1, \ldots, I_n\}$ and $\mathcal{B} = \{J_1, \ldots, J_m\}$, and a weight function $w : \mathcal{R} \rightarrow \mathbb{Q}^+$.
Result: A dominating set $\mathcal{D} \subseteq \mathcal{R}$ of $\mathcal{B}$.

```
begin
    1. for ℓ = 1 ... m do
           if J_ℓ does not intersect an interval I ∈ R with x(I) = 1 then
               (a) Increase y(J_ℓ) until some dual constraint becomes tight (y(J_ℓ) can be zero).
               (b) Select a red interval I ∈ R corresponding to a tight dual constraint on
               y(J_ℓ).
               (c) Set x(I) = 1.
           end
       end
    2. for ℓ = m ... 1 do
           if y(J_ℓ) > 0 then
               (a) Let x(I) be the variable which was set to 1 due to J_ℓ.
               (b) If x remains feasible when x(I) is set to zero then set x(I) = 0.
           end
       end
       return D = {I ∈ R | x(I) = 1}.
end
```

FIG. 2. Algorithm PD-Dominate.

We prove this claim using induction on the blue intervals in reverse order $(J_m, \ldots, J_1)$. (Recall that the blue intervals are ordered in a non-decreasing order of right endpoints.) At the recursive base, there are no intervals and there is nothing to prove. For the inductive step, we assume that the intervals $J_{\ell+1}, \ldots, J_m$ satisfy the claim, and show that it holds for J_ℓ as well. If $y(J_\ell) = 0$, then we are done. Otherwise, if $y(J_\ell) > 0$, it follows that when the algorithm reached J_ℓ in the first phase, $x(I) = 0$ for every red interval $I \in \mathcal{R}$ that intersects J. Hence, J_ℓ can only be dominated in $\mathcal{D}$ by red intervals that were taken after reaching J_ℓ in the first phase. If J_ℓ is the rightmost interval with non-zero dual variable, we are done. Otherwise, let J_k be the closest interval to the right of J_ℓ with non-zero dual variable. By the inductive assumption, we know that $\sum_{I : J_k \cap I \neq \emptyset} x(I) = 1$. Hence, when the algorithm returns to J_ℓ in the reverse deletion phase $\sum_{I : J_\ell \cap I \neq \emptyset} x(I) \leq 2$. Since the algorithm removes the red interval whose primal variable was set to 1 due to J_ℓ in case feasibility is maintained, we get that $\sum_{I : J_\ell \cap I \neq \emptyset} x(I) = 1$. □

The following is implied by the optimality of the primal-dual pair and the integrality of the primal solution.

COROLLARY 3.1. *The integrality gap of* DS *is* 1 *in the case of* $t = 1$.

3.3. ALGORITHM FOR t-INTERVAL GRAPHS. We are now in position to present our t^2-approximation algorithm for t-interval families. Our algorithm starts by solving the linear program LP-DS. Following this, it uses the optimal solution of LP-DS to construct an instance for the case of $t = 1$. A t^2-approximate solution for the original instance is then constructed from the optimal solution of the new instance.

Let x^* denote an optimal solution of LP-DS. We assume without loss of generality that each red t-interval in $\mathcal{R}$ consists of unique t intervals, i.e., that $|\mathcal{I}(\mathcal{R})| = t|\mathcal{R}|$. For a given blue t-interval $J = (j_1, \ldots, j_t) \in \mathcal{B}$, we select a unique *representative* $j \in \{j_1, \ldots, j_t\}$ which maximizes $\sum_{I \in \mathcal{R}, I \cap j \neq \emptyset} x^*(I)$. In other words, the representative j is the maximum dominated interval of J. We construct a new 1-interval instance $(\mathcal{R}', \mathcal{B}', w')$ as follows:

—$\mathcal{R}' = \{I' = i \mid i \in \mathcal{I}(\mathcal{R})\}$.
—$\mathcal{B}' = \{J' = j \mid J \in \mathcal{B}, j \text{ represents } J\}$.
—$w'(I') = w(I)/t$, where $I' = i$, $i \in \mathcal{I}(\mathcal{R})$, and I is the unique red t-interval which contains i.

We now invoke PD-Dominate$(\mathcal{R}', \mathcal{B}', w')$. Let $\mathcal{D}' \subseteq \mathcal{R}'$ be the dominating set returned by PD-Dominate$(\mathcal{R}', \mathcal{B}', w')$. We return the solution $\mathcal{D} = \{I \in \mathcal{R} \mid I' \in \mathcal{D}' \text{ is an interval of } I\}$. $\mathcal{D}$ is a feasible solution since all representatives are covered by $\mathcal{D}'$. Note that $\sum_{I \in \mathcal{D}} w(I) \leq t \cdot \sum_{I' \in \mathcal{D}} w'(I')$, and that this inequality is tight in case no two intervals of a t-interval in $\mathcal{D}$ are together in $\mathcal{D}'$.

LEMMA 3.2. *$\mathcal{D}$ is a t^2-approximate dominating set of $\mathcal{B}$.*

PROOF. First, note that by the construction of the algorithm above, $\mathcal{D}$ is a dominating set of $\mathcal{B}$. Let x and x' be the solution vectors corresponding to $\mathcal{D}$ and $\mathcal{D}'$, respectively. To prove the lemma, we show that $\sum_{I \in \mathcal{R}} w(I)x(I) \leq t^2 \cdot \sum_{I \in \mathcal{R}} w(I)x^*(I)$.

We define a fractional solution $\bar{x}$ for the instance $(\mathcal{R}', \mathcal{B}', w')$ as follows: $\bar{x}(I') = t \cdot x^*(I)$, where I' belongs to $\mathcal{I}(\mathcal{R})$ and I is the unique red t-interval that contains I'. Notice that

$$\sum_{I' \in \mathcal{R}'} w'(I')\bar{x}(I') = t \cdot \sum_{I \in \mathcal{R}} \frac{w(I)}{t} \cdot t \cdot x^*(I) = t \cdot \sum_{I \in \mathcal{R}} w(I)x^*(I) .$$

Next, we argue that $\bar{x}$ is feasible with respect to $(\mathcal{R}', \mathcal{B}', w')$. To see this, consider any blue t-interval $J = (j_1, \ldots, j_t)$ and its representative J'. By our selection of J', $\sum_{I \in \mathcal{R}, I \cap J' \neq \emptyset} x^*(I) \geq \frac{1}{t}$, since otherwise x^* would not be feasible. It follows that $\sum_{I' \in \mathcal{R}', I' \cap J' \neq \emptyset} \bar{x}(I') \geq 1$ for any $J' \in \mathcal{R}'$.

By Corollary 3.1, we know that the integral solution x' is an optimal fractional solution for $(\mathcal{R}', \mathcal{B}', w')$. Hence, the weight of $\bar{x}$ is at least as high as the weight of x'. It follows that

$$\sum_{I \in \mathcal{R}} w(I)x(I) \leq t \cdot \sum_{I' \in \mathcal{R}'} w'(I')x'(I') \leq t \cdot \sum_{I' \in \mathcal{R}'} w'(I')\bar{x}(I') = t^2 \cdot \sum_{I \in \mathcal{R}} w(I) \cdot x^*(I)$$

and we are done. □

The following theorem summarizes the main result of this section.

THEOREM 3.2. *The* MINIMUM DOMINATING SET *problem in t-interval graphs can be approximated in polynomial time within a factor of t^2.*

3.4. TIGHTNESS OF ANALYSIS. We next show that the analysis of our algorithm is essentially tight. Specifically, we show that for every $t \geq 2$, there exists an instance on which the algorithm computes a solution whose weight is $\binom{t+1}{2}$, while the optimum is 1.

We describe an instance that consists of a family $\mathcal{F}$ of $t(t+1)$ t-intervals. First, we define a family of t-intervals $\mathcal{F}' = \{I_1, \ldots, I_{t+1}\}$, by partitioning the real line into $\binom{t+1}{2}$ segments, and then placing in each segment $S_{i,i'}$, $\{i, i'\} \subseteq \{1, \ldots, t+1\}$, two identical intervals, one belonging to I_i, and the other belonging to $I_{i'}$. Then,

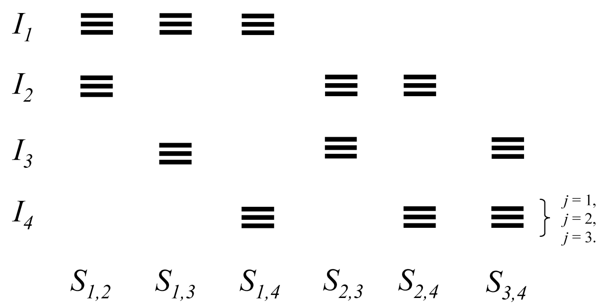

FIG. 3. The t-interval family $\mathcal{F}$ for $t = 3$.

we define $\mathcal{F}$ as the family that consists of t identical copies of each t-interval $\mathcal{F}'$, where $I_{i,j} \in \mathcal{F}$ is the jth copy of $I_i \in \mathcal{F}'$. An example of this construction with $t = 3$ is given in Figure 3.

We now make two observations. First, note that the family of t-intervals $\mathcal{F}$ is pairwise intersecting, that is, it is a clique. Second, note that we can determine any order between the intervals in each segment while making sure that $\mathcal{F}$ remains pairwise intersecting.

Now, since $\mathcal{F}$ is a clique it follows that any t-interval in $\mathcal{F}$ constitutes a dominating set. Assuming unit weights, any t-interval is also an optimal solution. We show that in the case of unit weights the algorithm may compute a solution whose weight is $\binom{t+1}{2}$.

We assume that the algorithm starts with the fractional optimal solution x^*, where $x^*(I_{ij}) = 1/t(t+1)$. Observe that x^* is indeed optimal. Now, in this case, each t-interval is equally dominated in each of its intervals. It follows that the algorithm may choose any interval as a representative. We assume that for a given $i \in \{1, \ldots, t+1\}$, the algorithm chooses representatives in different segments for each t-interval $I_{i,j}$, $j \in \{1, \ldots, t\}$. Hence, the 1-interval instance that is constructed by the algorithm contains two blue intervals in each segment $S_{i,i'}$.

Since all segments are disjoint, it follows that any solution of the 1-interval instance must contain at least one red interval in each segment $S_{i,i'}$. Since there can be any order between the red intervals in each segment, a solution that contains $\binom{t+1}{2}$ red intervals, each belonging to a different t-interval, is feasible and optimal. In this case, the solution returned by the algorithms contains $\binom{t+1}{2}$ t-intervals, and therefore its weight is $\binom{t+1}{2}$ as required.

4. *Maximum Clique*

In the following section, we consider the MAXIMUM CLIQUE problem. We show that the problem is hard for t-interval graphs with $t \geq 3$. Following this, we give a simple approximation algorithm for the problem which relies on bounds obtained for the transversal number of t-interval families (see Definition in Section 4.2).

4.1. **NP**-HARDNESS RESULT.

We begin by showing that MAXIMUM CLIQUE is **NP**-hard in 3-interval graphs. We show this be presenting a reduction from MAXIMUM 2-DNF. Recall that in MAXIMUM 2-DNF, we are asked to determine whether one can satisfy at least k clauses in a given 2-DNF formula. That is, whether there is a truth assignment to the boolean variables of the formula, such

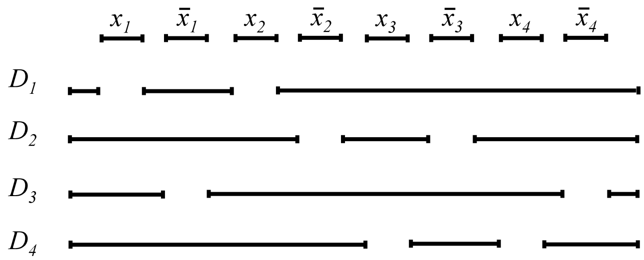

FIG. 4. An example of the construction at segment **I**. The clauses in this example are $D_1 = (\bar{x}_1 \wedge \bar{x}_2)$, $D_2 = (x_2 \wedge x_3)$, $D_3 = (x_1 \wedge x_4)$, and $D_4 = (\bar{x}_3 \wedge \bar{x}_4)$.

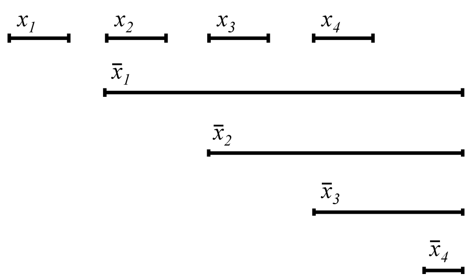

FIG. 5. Segment **II**.

that the value of at least k clauses in the formula is TRUE under this assignment. MAXIMUM 2-DNF is **NP**-hard due to the hardness result given in Kohli et al. [1994] for MINIMUM 2-CNF. We initially show that the weighted variant of MAXIMUM CLIQUE is **NP**-hard, and then relax the weighted condition.

Let $\Phi = D_1 \vee \cdots \vee D_m$ be a 2-DNF formula over a set of variables $\{x_1, \ldots, x_n\}$. We construct a 3-interval family $\mathcal{F}$ which contains a 3-interval for each clause and each literal (i.e., variable with or without negation). For convenience, we will describe our construction by defining the underlying interval family $\mathcal{I}_{\mathcal{F}}$ of $\mathcal{F}$ over three disjoint segments **I**, **II**, and **III**, of the real line.

In segment **I**, we first place $2n$ disjoint intervals, one for each literal 3-interval in $\mathcal{F}$. Then, we define a 3-interval in $\mathcal{F}$ corresponding to each clause $D_i \in \Phi$ as follows: Suppose D_i consists of the two literals α and β. The 3-interval of D_i is the 3-interval obtained by taking an interval which covers all of segment **I**, and then removing from it the two intervals that correspond to the negations of α and β. In this way, all 3-intervals that correspond to clauses are pairwise intersecting, since all of them contain, for instance, the right boundary point of segment **I**. Figure 4 gives an example of our construction at this segment.

In segment **II**, we first place n disjoint intervals, one for each 3-interval in $\mathcal{F}$ which corresponds to a positive literal. The intervals are placed in the order of the variables, that is, the interval corresponding to x_i is placed to the left of the interval which corresponds to x_{i+1}. Next, for all $1 \leq i < n$, we add an interval corresponding to $\bar{x}_i$ that begins at the left endpoint of the interval corresponding to x_{i+1} and ends at the right boundary point of segment **II**. The interval which corresponds to $\bar{x}_n$ is placed to the right of the interval corresponding to x_n. In this way, all intervals corresponding to negative literals in this segment, intersect at the right boundary point. See example in Figure 5.

Segment **III** is defined symmetrically to **II**, where the roles of the intervals corresponding to positive literals and those corresponding to negative literals is reversed.

LEMMA 4.1. *Let $I \in \mathcal{F}$ be a 3-interval corresponding to a literal α. Then I intersects all other 3-intervals in $\mathcal{F}$ which correspond to literals, except the 3-interval which corresponds to the negation of α.*

PROOF. Assume, without loss of generality, that α is some positive literal x_i. Then, by construction, the 3-interval that corresponds to $\bar{x}_i$ does not intersect I. Furthermore, all 3-intervals that correspond to the other positive variables intersect I in segment **III**. To complete proof note that for all $j < i$, the 3-interval that corresponds to literal $\bar{x}_j$ intersects I in segment **II**, and for all $j > i$, the 3-interval that corresponds to literal $\bar{x}_j$ intersects I in segment **III**. □

We now assign to each 3-interval that corresponds to a literal weight $m + 1$ (= the number of clauses + 1), and to each 3-interval corresponding to a clause weight 1. We claim that this yields the following:

LEMMA 4.2. *There is a clique $\mathcal{K} \subseteq \mathcal{F}$ of weight $(m + 1)n + k$ iff there is an assignment that satisfies k clauses of the 2-DNF formula Φ.*

PROOF. Assume that there is an assignment satisfying k clauses of Φ. Let $\mathcal{K}_X \subseteq \mathcal{F}$ be the subset of 3-intervals corresponding to this assignment, and $\mathcal{K}_D \subseteq \mathcal{F}$ be the 3-intervals which correspond to the k satisfied clauses of Φ. Then, $\mathcal{K}_X$ is pairwise intersecting by Lemma 4.1, and $\mathcal{K}_D$ is pairwise intersecting by construction. Furthermore, each $I \in \mathcal{K}_X$ intersects each $J \in \mathcal{K}_D$ in segment **I**, since the assignment corresponding to $\mathcal{K}_X$ satisfies all clauses with 3-intervals in $\mathcal{K}_D$. It follows that $\mathcal{K} = \mathcal{K}_X \cup \mathcal{K}_D$ is a clique in $\mathcal{F}$, with weight $(m + 1)n + k$ as desired.

Conversely, assume that there is an $(m + 1)n + k$ weighted clique. Since the overall number of clauses is m it must be that the clique is formed from 3-intervals of n literals and k clauses. Since, by Lemma 4.1, x_i and $\bar{x}_i$ cannot be in a clique, the n variables form an assignment. By construction, as observed above, each clause corresponding to a t-interval in the clique must be satisfied. □

This immediately implies that the weighted variant of MAXIMUM CLIQUE is **NP**-hard in 3-interval graphs. This line of proof can be extended to show that the same is true for the unweighted variant by constructing $m + 1$ identical copies of each 3-interval corresponding to a literal, instead of one 3-interval of weight $m + 1$.

THEOREM 4.1. MAXIMUM CLIQUE *is* **NP**-*hard in t-interval graphs for $t \geq 3$.*

4.2. APPROXIMATION ALGORITHM. We next present a $(t^2 - t + 1)/2$ approximation algorithm for MAXIMUM CLIQUE. We begin with defining the notion of a transversal of a t-interval family: A *transversal* of $\mathcal{F}$ is a set of points $\{p_1, p_2, \ldots, p_\tau\}$ such that for every $I \in \mathcal{F}$ there is at least one $p_i \in \{p_1, p_2, \ldots, p_\tau\}$ with $p_i \in I$. The *transversal number* of $\mathcal{F}$, denoted $\tau(\mathcal{F})$, is the minimum size of any transversal of $\mathcal{F}$.

Note that there is no loss of generality in restricting the set of points in a transversal of $\mathcal{F}$ to be endpoints of intervals in the underlying family of intervals $\mathcal{I}_{\mathcal{F}}$. For a fixed τ, $1 \leq \tau \leq \tau(\mathcal{F})$, we say that a clique $\mathcal{K}$ of $\mathcal{F}$ is a τ*-clique* if it can be transversed with τ points. Hence, any clique in $\mathcal{F}$ is a $\tau(\mathcal{F})$-clique.

Algorithm 2-Clique($\mathcal{F}, w$)

Data: A set of t-intervals $\mathcal{F}$ and a weight function $w : \mathcal{F} \rightarrow \mathbb{Q}^+$.
Result: A 2-clique $\mathcal{K}$ of $\mathcal{F}$.
begin
  1. $\mathcal{K} \leftarrow \emptyset$.
  2. **foreach** *pair of endpoints* p, q *of intervals in* $\mathcal{I}_{\mathcal{F}}$ **do**
    a. Define $\mathcal{K}_p = \{I \in \mathcal{F} \mid p \in I\}$ and $\mathcal{K}_q = \{I \in \mathcal{F} \mid q \in I \wedge p \notin I\}$.
    b. Let G be the complement graph of $\Omega_{\mathcal{K}_p \cup \mathcal{K}_q}$.
    c. Compute the maximum weight independent set I in G.
    d. Let $\mathcal{K}' \subseteq \mathcal{K}_p \cup \mathcal{K}_q$ be the subset of t-intervals corresponding to I.
    e. if $w(\mathcal{K}') > w(\mathcal{K})$ then $\mathcal{K} \leftarrow \mathcal{K}'$.
  end
  return $\mathcal{K}$.
end

FIG. 6. Algorithm 2-clique.

Obtaining upper bounds for the transversal number of a family of t-intervals has been studied in Alon [1998], Gyráfás and Lehel [1985], Kaiser [1997], and Tardos [1995]. Kaiser [1997] proved a bound on the traversal number of such families in terms of their clique-cover number. The *clique-cover* number of a family of t-interval $\mathcal{F}$, denoted $\nu(\mathcal{F})$, is defined to be the minimum number ν such that $\mathcal{F}$ is the union of ν cliques.

THEOREM 4.2 (KAISER 1997). *For any family $\mathcal{F}$ of t-intervals, $\tau(\mathcal{F}) \leq (t^2 - t + 1)\nu(\mathcal{F})$.*

Note that this theorem implies that any clique in a $\mathcal{F}$ is a $(t^2 - t + 1)$-clique. We next show that the maximum weight 2-clique in $\mathcal{F}$ can easily be found. This will give a $(t^2 - t + 1)/2$ approximation algorithm for MAXIMUM CLIQUE.

In Figure 6, we present algorithm 2-Clique, which computes the maximum weight 2-clique of $\mathcal{F}$. Correctness of this algorithm is immediate. Also, observe that the graph G computed at step 2(b) is bipartite, with $\mathcal{K}_p$ and $\mathcal{K}_q$ corresponding to the two classes in the vertex partitioning. Since MAXIMUM INDEPENDENT SET is solvable in polynomial time in bipartite graphs [Garey and Johnson. 1979], algorithm 2-Clique can be carried out in polynomial time as well.

THEOREM 4.3. *The* MAXIMUM CLIQUE *problem in t-interval graphs can be approximated in polynomial time within a factor of* $(t^2 - t + 1)/2$.

5. *Discussion*

In this article, we initiated the study of optimization problems in multiple-interval graphs. Our study focused on the three classical problems: MINIMUM VERTEX COVER, MINIMUM DOMINATING SET, and MAXIMUM CLIQUE, for which we gave approximation algorithms which performance ratios $2 - 1/t$, t^2, and $(t^2 - t + 1)/2$, respectively. We also showed that MAXIMUM CLIQUE is **NP**-hard already for 3-interval graphs.

There are several open issues that stem from our study. Below we list some of these:

(1) Our algorithm for MINIMUM VERTEX COVER has a performance ratio which equals the currently best known ratio for sparse hereditary graphs of maximum

degree $2t - 1$. Can this be tightened to the best known ratio for bounded degree $2t - 1$ graphs [Halperin 2000]? Since these graphs are included in the class of t-interval graphs [Griggs and West 1980], this will, in some sense, give a satisfactory tight result.

(2) Can one improve the performance ratio of our algorithm for MINIMUM DOMINATING SET? The only known lower bound we have is $\Omega(\lg t)$ which follows from the lower bound known for MINIMUM DOMINATING SET in general graphs [Raz and Safra 1997], and from the fact that any graph is a t-interval graph with $t \geq (n + 1)/4$ [Griggs 1979].

(3) Is MAXIMUM CLIQUE **NP**-hard in 2-interval graphs? Is it **APX**-hard in t-interval graphs for any constant $t \in \mathbb{N}^+$? Also, the performance ratio given by our algorithm in Section 4.2 is most likely not tight. Can one obtain a substantially better ratio?

Another obvious direction for future research is to study other optimization problems in multiple-interval graphs. Some interesting candidates might be MINIMUM CLIQUE COVER, MINIMUM INDEPENDENT DOMINATING SET, MAXIMUM CUT, and so forth . . .

ACKNOWLEDGMENTS. We would like to thank Reuven Bar-Yehuda for fruitful discussions.

REFERENCES

AGARWAL, P., VAN KREVELD, M., AND SURI, S. 1998. Label placement by maximum independent set in rectangles. *Computat. Geom. Theory Appl. 11*.

ALON, N. 1998. Piercing d-intervals. *Disc. Computat. Geom. 19*, 333–334.

AUMANN, Y., LEWENSTEIN, M., MELAMUD, O., PINTER, R., AND YAKHINI, Z. 2005. Dotted interval graphs and high throughput genotyping. In *Proceedings of the 16th Annual Symposium on Discrete Algorithms (SODA)*. ACM, New York, 339–348.

BAFNA, V., NARAYANAN, B., AND RAVI, R. 1996. Nonoverlapping local alignments (weighted independent sets of axis parallel rectangles). *Disc. Appl. Math. 71*, 41–53.

BAR-NOY, A., BAR-YEHUDA, R., FREUND, A., NAOR, J., AND SHIEBER, B. 2001. A unified approach to approximating resource allocation and scheduling. *J. ACM 48*, 5, 1069–1090.

BAR-YEHUDA, R., AND EVEN, S. 1981. A linear time approximation algorithm for the weighted vertex cover problem. *J. Algorithms 2*, 198–203.

BAR-YEHUDA, R., AND EVEN, S. 1985. A local-ratio theorem for approximating the weighted vertex cover problem. *Ann. Disc. Math. 25*, 27–46.

BAR-YEHUDA, R., HALLDÓRSSON, M., NAOR, J., SHACHNAI, H., AND SHAPIRA, I. 2002. Scheduling split intervals. In *Proceedings of the 13th Annual Symposium on Discrete Algorithms (SODA)*. ACM, New York, 732–741.

BAR-YEHUDA, R., AND RAWITZ, D. 2005. Using fractional primal-dual to schedule split intervals with demands. In *Proceedings of the 13th Annual European Symposium on Algorithms*. 714–725.

BERMAN, P., DASGUPTA, B., MUTHUKRISHNAN, S., AND RAMASWAMI, S. 2001. Efficient approximation algorithms for tiling and packing problems with rectangles. *J. Algorithms 41*.

BLIN, G., FERTIN, G., AND VIALETTE, S. 2004. New results for the 2-interval pattern problem. In *Proceedings of the 15th Annual Symposium on Combinatorial Pattern Matching (CPM)*. 311–322.

BRANDSTÄDT, A., LE, V., AND SPINRAD, J. 1999. *Graph classes: A survey*. SIAM, Philadelphia, PA.

BUTMAN, A., HERMELIN, D., LEWENSTEIN, M., AND RAWITZ, D. 2007. Optimization problems in multiple-interval graphs. In *Proceedings of the 18th Annual Symposium on Discrete Algorithms (SODA)*. ACM, New York, 268–278.

CHAN, T. 2003. Polynomial-time approximation schemes for packing and piercing fat objects. *J. Algorithms 46*.

CLARK, B. N., COLBOURN, C. J., AND JOHNSON, D. S. 1990. Unit disk graphs. *Disc. Math. 86*, 165–177.
CROCHEMORE, M., HERMELIN, D., LANDAU, G., AND VIALETTE, S. 2005. Approximating the 2-interval pattern problem. In *Proceedings of the 13th Annual European Symposium on Algorithms (ESA)*. 426–437.
ERLEBACH, T., JANSEN, K., AND SEIDEL, E. 2005. Polynomial-time approximation schemes for geometric intersection graphs. *SIAM J. Comput. 34*.
GAREY, M., AND JOHNSON., D. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York.
GAUR, D. R., IBARAKI, T., AND KRISHNAMURTI, R. 2002. Constant ratio approximation algorithms for the rectangle stabbing problem and the rectilinear partitioning problem. *J. Algorithms 43*, 138–152.
GOLUMBIC, M. 1980. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, New York.
GRIGGS, J. 1979. Extremal values of the interval number of a graph, II. *Disc. Math. 28*, 37–47.
GRIGGS, J., AND WEST, D. 1980. Extremal values of the interval number of a graph. *SIAM J. Algeb. Disc. Meth. 1*, 1, 1–14.
GYRÁFÁS, A., AND LEHEL, J. 1985. Covering and coloring problems for relatives of intervals. *Disc. Math. 55*, 167–180.
HALPERIN, E. 2000. Improved approximation algorithms for the vertex cover problem in graphs and hypergraphs. In *Proceedings of the 11th Annual Symposium on Discrete Algorithms (SODA)*. ACM, New York, 329–337.
HASSIN, R., AND MEGIDDO, N. 1991. Approximation algorithms for hitting objects with straight lines. *Disc. Appl. Math. 30*, 1, 29–42.
HASSIN, R., AND SEGEV, D. 2005. Rounding to an integral program. In *Proceedings of the 4th International Workshop on Efficient and Experimental Algorithms*. 44–54.
HASSIN, R., AND TAMIR, A. 1991. Improved complexity bounds for location problems on the real line. *Oper. Res. Lett. 10*, 7, 395–402.
HOCHBAUM, D. 1983. Efficient bounds for the stable set, vertex cover and set packing problems. *Disc. Appl. Math. 6*, 243–254.
HOCHBAUM, D., AND LEVIN, A. 2006. Cyclical scheduling and multi-shift scheduling: complexity and approximation algorithms. *Disc. Optimiz. 3*, 4, 327–340.
HUNT III, H. B., MARATHE, M., RADHAKRISHNAN, V., RAVI, S., ROSENKRANTZ, D., AND STEARNS, R. 1998. NC-approximation schemes for NP- and PSPACE-hard problems for geometric graphs. *J. Algorithms 26*.
KAISER, T. 1997. Transversals of d-intervals. *Disc. Computat. Geom. 18*, 2, 195–203.
KARLOFF, H. 1991. *Linear Programming*. Birkhauser, Boston, MA.
KOHLI, R., KRISHNAMURTI, R., AND MIRCHANDANI, P. 1994. The minimum satisfiability problem. *SIAM J. Disc. Math. 7*, 275–283.
MCKEE, T., AND MCMORRIS, F. 1999. *Topics in intersection graph theory*. SIAM, Philadelphia, PA.
NEMHAUSER, G. L., AND TROTTER, L. E. 1975. Vertex packings: Structural properties and algorithms. *Math. Prog. 8*, 232–248.
PAPADIMITRIOU, C., AND YANNAKAKIS, M. 1991. Optimization, approximation, and complexity classes. *J. Comput. Syst. Sci. 43*, 425–440.
RAZ, R., AND SAFRA, S. 1997. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *Proceedings of the 29th Annual Symposium on the Theory of Computing (STOC)*. 475–484.
SCHEINERMAN, E., AND WEST, D. 1983. The interval number of a planar graph: three intervals suffice. *J. Combinat. Theory B 35*, 224–239.
TARDOS, G. 1995. Transversals of 2-intervals, a topological approach. *Combinatorica 15*, 123–134.
TROTTER, W. T., AND HARARY, F. 1978. On double and multiple interval graphs. *J. Graph Theory 2*, 137–142.
VIALETTE, S. 2004. On the computational complexity of 2-interval pattern matching problems. *Theore. Comput. Sci. 312*, 335–379.
WEST, D., AND SHMOYS, D. 1984. Recognizing graphs with fixed interval number is NP-complete. *Disc. Appl. Math. 8*, 295–305.

RECEIVED JUNE 2007; REVISED NOVEMBER 2008; ACCEPTED APRIL 2009