Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 26, no. 1, pp. 199–205 (2022)
DOI: 10.7155/jgaa.00588

A Faster Algorithm for Maximum Independent Set on Interval Filament Graphs

Darcy Best[1,2] *Max Ward*[1,3]

[1]Both authors had equal contributions
[2]Google
[3]School of Computer Science, University of Adelaide

Submitted: October 2021	Reviewed: January 2022	Revised: March 2022
Accepted: May 2022	Final: May 2022	Published: June 2022
Article type: Concise paper		Communicated by: A. Wolff

Abstract. We provide an algorithm requiring only $O(N^2)$ time to compute the maximum weight independent set in an N-vertex interval filament graph. This implies an $O(N^4)$-time algorithm to compute the maximum weight induced matching in such graphs. Both algorithms significantly improve upon the previous best complexities for these problems. Previously, the maximum weight independent set and maximum weight induced matching problems required $O(N^3)$ and $O(N^6)$ time respectively.

1 Introduction

In this article, we provide an improved algorithm for finding maximum weight independent sets in interval filament graphs. This implies an improved algorithm for finding maximum weight induced matchings in interval filament graphs. Interval filament graphs were characterised by Gavril in 2000 [3] and again in 2007 [4]. They include co-comparability graphs and polygon-circle graphs [3].

Gavril later generalized the definition to 3D interval filaments [4]. In the process, Gavril and others have given several definitions of interval filament graphs. These definitions all characterize interval filaments differently, however, they all give rise to the same underlying class of intersection graph.

Definition 1 (Gavril's first definition from [3]) *An interval filament is defined by a curve C in the xy-plane which has a left endpoint ℓ and a right endpoint r such that both endpoints define an interval on the x-axis. Also, each point in C lies on or above the x-axis. If the intervals of two interval filaments are disjoint, their curves do not intersect.*

An interval filament graph is the intersection graph of a family of interval filaments.

E-mail addresses: max.ward-graham@adelaide.edu.au (Max Ward)

In Definition 1, Gavril does not specify the nature of the curve. Since Gavril's findings hold if the curve is continuous but otherwise unconstrained, we assume this most general definition.

Definition 2 (Gavril's second definition from [4]) *Take Definition 1 and constrain each curve C such that it is the union of continuous interval curves, which are each described by functions. This implies that it stays in the interval $[\ell, r]$ on the x-axis (see [4] for more details).*

Cameron gives an additional definition.

Definition 3 (Cameron's definition from [2]) *Take Definition 1 and constrain each curve C such that it stays in the interval $[\ell, r]$ on the x-axis, but no further restrictions on the curves (see [2] for more details).*

We use Definition 1 since it includes all other definitions. We formalise it as follows.

Definition 4 (Interval Filament) *Consider an interval described by $\ell, r \in \mathbb{R}$, with $\ell \leq r$. A (possibly self-intersecting) continuous curve $C \subset \mathbb{R}^2$ is an interval filament with endpoints ℓ and r if:*

- $(\ell, 0) \in C$, $(r, 0) \in C$, *and*
- $y \geq 0$ *for all* $(x, y) \in C$

Definition 5 (Interval Filament Graph) *An interval filament graph is the intersection graph of a set of interval filaments (from Definition 4) such that if two interval filaments have disjoint intervals, then their curves do not intersect.*

See Figure 1 for an example interval filament graph.

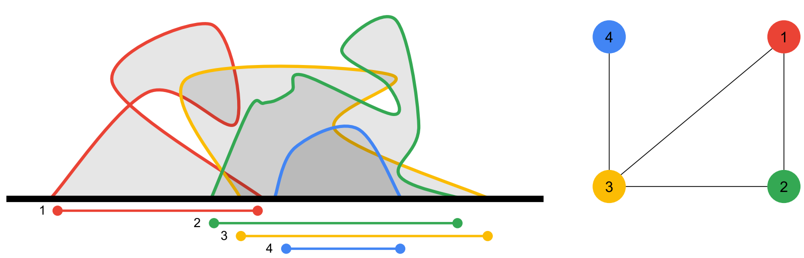

Figure 1: An example interval filament graph. The left depicts a family of interval filaments. The right shows the resulting intersection graph. Note that vertices 2 and 4 are not adjacent although the interval of vertex 4 is contained in the interval of vertex 2.

An *independent set* is a subset of vertices in a graph such that no two share an edge. If each vertex has an associated weight, then a *maximum weight independent set* (MWIS) is the independent set with the largest weight over all independent sets in a graph. An induced matching is a subset of edges in a graph that is a matching such that the induced subgraph on that matching is exactly that matching. A *maximum weight induced matching* (MWIM) is an induced matching that has the largest weight of edges over all induced matchings in the graph. Both MWIS and MWIM are NP-hard to compute [5, 6].

Independent sets are related to both matchings and induced matchings. First, the *line graph* $L(G)$ of a graph G is constructed by making a vertex for each edge in G and connecting these vertices if the corresponding edges in G share a vertex. A matching of G is a set of edges such that no two edges share an endpoint. In $L(G)$, a matching in G corresponds to a set of vertices in $L(G)$ such that no two of these vertices are adjacent (*i.e.*, they form an independent set in $L(G)$). Second, let's consider the *square* G^2 of a graph G, which is constructed by adding edges that do not already exist between vertices connected by a path of length two. Consider an edge e in an induced matching of G. No edge e' sharing an endpoint with e can be in the matching. Furthermore, no edge e'' whose endpoints are adjacent (connected by an edge) to an endpoint of e can be in the induced matching (otherwise, that edge connecting e with e'' would be in the induced graph). Thus, all edges in an induced matching correspond to vertices that are at least distance 2 from one another in $L(G)$, which tells us that an induced matching in G corresponds to an independent set in $[L(G)]^2$. Overall, this gives us a one-to-one relationship between independent sets in $L(G)$ and matchings in G, as well as between independent sets in $[L(G)]^2$ and induced matchings in G.

Previous Algorithms and Our Contribution

Gavril [3] gives an $O(|V|^3)$-time algorithm for finding a MWIS in an interval filament graph $G = (V, E)$. Importantly, Gavril's algorithm finds the clique in the complement graph rather than using the interval filaments directly. We improve MWIS in interval filament graphs to $O(|V|^2)$ using the interval filaments directly.

Cameron [2] noted that interval filament graphs have properties that imply a polynomial-time algorithm to find a MWIM in interval filament graphs given a polynomial-time algorithm to find a MWIS in interval filament graphs. In particular, the square of the line graph $[L(G)]^2$ for an interval filament graph is also an interval filament graph. The MWIS in $[L(G)]^2$ are in one-to-one correspondence with the MWIM in G. This, in conjunction with Gavril's $O(|V|^3)$-time algorithm, implies an $O(|E|^3)$-time algorithm for the MWIM problem. We are able to use Cameron's result to reduce the time needed to compute an MWIM in interval filament graphs to $O(|E|^2)$.

In our complexity analysis below, we do our computations in terms of S, the set of interval filaments, rather than the actual graph, G, to emphasize our algorithm's reliance on the underlying interval filaments.

2 From Interval Filaments to a Graph

A family of interval filaments and their intersections define an interval filament graph. To get the graph model containing the set of edges, the intersections between the interval filaments must be computed. Previous work did not address this problem and assumed that the graph model and corresponding interval filaments are given together [3, 4]. For completeness, we give a fast algorithm for obtaining the graph model given the interval filaments.

Creating the intersection graph from the interval filaments is highly dependent on the representation of the interval filaments. For example, if the interval filaments are arbitrary curves (as in Figure 1), determining if two interval filaments intersect can be quite tricky, and a completely separate problem.

In a simple case, where each interval filament is a polyline with no self-intersections, we have the following.

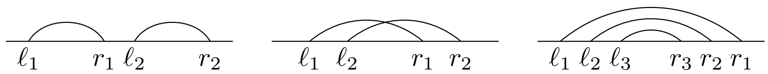

Figure 2: A visual representation of (P1), (P2), and (P3) from left to right.

Theorem 1 *Let $S = \{P_1, P_2, \ldots, P_N\}$ be a set of polylines that are not self-intersecting, where P_i contains t_i line segments. If $T = \sum t_i$ and K is the total number of intersections of the T line segments, then the intersection graph of S can be computed in $O(\min(K + T\log T, N \cdot T\log T))$ time.*

Proof: The key is to apply Balaban's algorithm [1] in two separate ways. Given a set of M line segments, Balaban's algorithm determines all intersections of the M line segments in $O(K' + M\log M)$ time, where K' is the total number of intersections. Balaban's algorithm can be slightly modified to determine if two non self-intersecting polylines intersect in $O(M\log M)$ time by simply exiting once an intersection is found. We will run both of the following algorithms in parallel, terminating both when one of them completes.

The first application of Balaban's algorithm is to determine all intersections of the line segments $P_1 \cup P_2 \cup \cdots \cup P_N$. This takes $O(K + T\log T)$ time. This first application is fast if K is relatively small. However, K can be as large as $\Theta(T^2)$.

The second application of Balaban's algorithm is simply used to determine if P_i intersects with P_j for $i \neq j$. For each pair (i, j), we can determine if P_i intersects with P_j in $O((t_i + t_j)\log(t_i + t_j))$ time. Each of the $t_i + t_j$ line segments involved contributes $O(\log(t_i + t_j)) = O(\log T)$ time. Since each line segment is involved in $N - 1$ tests, each line segment contributes $O(N\log T)$ time to the run time of the algorithm. Thus, in total, we spend $O(N \cdot T\log T)$ time in this second application of Balaban's algorithm. □

If each pair of interval filaments intersect in $O(1)$ places, then K is $O(|E|)$ in Theorem 1. Thus, we can find the intersection graph in $O(|E| + T\log T)$ time.

For the remainder of the article, we will assume that the intersection of two curves has previously been determined, so we may look it up in $O(1)$ time.

3 Maximum Weight Independent Set

The maximum weight independent set of a set of interval filaments will be computed using only three properties of interval filaments. Let F_1, F_2, F_3 be interval filaments with endpoints $\ell_i \le r_i$ for $i \in \{1, 2, 3\}$. These properties are visualized in Figure 2.

(P1) If $\ell_1 \le r_1 < \ell_2 \le r_2$, then F_1 and F_2 do not intersect.

(P2) If $\ell_1 \le \ell_2 \le r_1 \le r_2$, then F_1 and F_2 do intersect.

(P3) If $\ell_1 < \ell_2 < \ell_3 \le r_3 < r_2 < r_1$, F_1 does not intersect with F_2, and F_2 does not intersect with F_3, then F_1 does not intersect with F_3.

We now state our main result:

Theorem 2 *Let S be a set of weighted interval filaments (with their intersections known). The maximum weight independent set of the intersection graph defined by S can be computed in $O(|S|^2)$ time and space.*

Proof: For convenience, we will add the infinite interval filament $\sqcap$ with weight 0 to S:

$$(-\infty, 0) - (-\infty, \infty) - (\infty, \infty) - (\infty, 0).$$

Sort the $|S|$ left endpoints of the interval filaments based on their x-value (breaking ties arbitrarily) and index each interval filament by the corresponding order from 1 to $|S|$, left-to-right. Note that the infinite interval filament will have index 1. Furthermore, let after(i) be the index of the interval filament whose left endpoint comes immediately to the right of the right endpoint of interval filament i. If there is no interval filament whose left endpoint is to the right of interval filament i, then define after$(i) = |S| + 1$. This setup takes $O(|S| \log |S|)$ time to sort the interval filaments and $O(|S|)$ time to compute the after values if we sweep the endpoints from right to left.

With this setup, we may describe the algorithm. For simplicity, we say that an interval filament x (with endpoints ℓ_x and r_x) is *strictly under* interval filament y (with endpoints ℓ_y and r_y) if $\ell_y < \ell_x \leq r_x < r_y$ and x does not intersect y. Let $F(\ell, c)$ be the maximum weight of an independent set that only contains interval filaments that are strictly under interval filament ℓ and does not contain interval filaments $1, 2, \ldots, c-1$. Note that $F(1, 2)$ is the maximum weight over all independent sets of the graph.

To compute an arbitrary $F(\ell, c)$, we only have three situations to deal with:

(i) If $c = |S| + 1$, then $F(\ell, c) = 0$, since we have eliminated all interval filaments from consideration.

(ii) If interval filament c is not strictly under interval filament ℓ, then $F(\ell, c) = F(\ell, c+1)$ by the definition of F.

(iii) Otherwise, we must try both including interval filament c and not including interval filament c in our independent set and take the better answer between these options. If we choose to not include interval filament c, then $F(\ell, c) = F(\ell, c+1)$. If we choose to include interval filament c, then any other interval filaments included in the independent set must lie strictly under interval filament c or fully to the right of interval filament c. (Recall that interval filaments to the left of c are not taken into account by $F(\ell, c)$.) So in this case, $F(\ell, c) = F(c, c+1) + F(\ell, \text{after}(c)) + \text{weight}(c)$. In total,

$$F(\ell, c) = \max\left\{F(\ell, c+1), \quad F(c, c+1) + F(\ell, \text{after}(c)) + \text{weight}(c)\right\}.$$

At every step in the algorithm, $F(\ell, c)$ is only used when we are considering a case where interval filament ℓ is in an independent set we are constructing. This allows us to prove correctness just based on (P1), (P2), and (P3). Adding an interval filament c to the independent set cannot possibly intersect with other interval filaments already included without either intersecting another curve that is strictly under ℓ, but completely to the left of c, (this would violate (P1)) or intersecting an interval filament that is not strictly under interval filament ℓ (this would violate (P3)). The interval filaments added to the independent set from "$F(c, c+1)$" cannot intersect those from "$F(\ell, \text{after}(c))$" by (P1) and (P2).

The weight of the MWIS of the graph is $F(1, 2)$. We can use dynamic programming (DP) to store the values of F. Since F only depends recursively on DP values with a strictly larger second argument, we can fill the DP table bottom-up sweeping c from large to small and ℓ in any order (or use memoization). This DP solution takes $O(|S|^2)$ time and space. □

If we would like to actually find a MWIS, we can re-traverse the DP states, taking the optimal choice at each step. Given the corresponding interval filament graph, this algorithm runs in $O(|V|^2)$ time. This improves the running time of the previous best algorithm, which required $O(|V|^3)$ time [3].

4 Maximum Weight Induced Matching

Maximum weight induced matching for interval filament graphs can be solved in essentially the same way as the maximum weight independent set. Cameron [2] showed that the square of the line graph $[L(G)]^2$ of an interval filament graph is also an interval filament graph. Because an independent set in $[L(G)]^2$ is a maximum induced matching in G, Cameron showed that having a polynomial-time algorithm for finding a maximum independent set in an interval filament graph implies a polynomial-time algorithm for the maximum induced matching problem.

We use similar observations to solve the maximum weight induced matching problem. However, we give a simpler construction to that used by Cameron. The key observation is that the union of two intersecting interval filaments (with the new endpoints being the leftmost left endpoint and the rightmost right endpoints of the two interval filaments) also follows (P1), (P2), and (P3) from Section 3. This allows us to reuse our algorithm from the previous section.

Theorem 3 *Let S be a set of weighted interval filaments (with their intersections known). The maximum weight induced matching of the intersection graph defined by S can be computed in $O(|S|^4)$ time and space.*

Proof: Let

$$S' = \{a \cup b : a, b \in S, a \neq b, a \cap b \neq \emptyset\}.$$

All elements in S' satisfy the properties needed in Section 3, so we may run that algorithm in $O(|S'|^2)$ time, which is $O(|E|^2)$ in the corresponding interval filament graph. Note that S' can have $|S|^2$ elements, so the time complexity is $O(|S|^4)$ in the worst case. □

This improves the running time of the previous best algorithm, which required $O(|S|^6)$ time [2].

5 Final Remarks

We have described an $O(|S|^2)$-time algorithm that takes a set S of interval filaments with their intersections and finds a MWIS in the corresponding interval filament graph. This improves the previous best complexity, $O(|S|^3)$, for the problem [3]. The improvement largely comes from operating directly on the interval filaments. Further, we show that our findings imply a faster algorithm, $O(|S|^4)$, for the MWIM problem in interval filament graphs, which improves the previous best complexity from $O(|S|^6)$ [2].

Lower bounds on the time needed to solve these problems are open questions. There are interval filament graphs that make our algorithms run in $\Theta(|S|^2)$ time for MWIS and $\Theta(|S|^4)$ time for MWIM (see Figure 3). For circle graphs, which are a subset of interval filament graphs, an $O(|V|^2)$-time algorithm for MWIS exists [7] and an $O(|V|^3)$-time algorithm exists for MWIM [8]. Also, we wonder if faster algorithms exist for the unweighted variants of the problems.

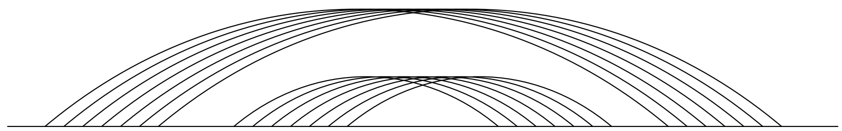

Figure 3: Worst case for our algorithm. The outer interval filaments will be ℓ once for every inner interval filament as c in the MWIS algorithm. Similarly, every pair of outer interval filaments will be ℓ once for every pair of inner interval filaments as c in MWIM.

References

[1] I. J. Balaban. An optimal algorithm for finding segments intersections. In *Proceedings of the Eleventh Annual ACM Symposium on Computational Geometry (SCG)*, page 211–219, 1995. doi:10.1145/220279.220302.

[2] K. Cameron. Induced matchings in intersection graphs. *Discrete Mathematics*, 278(1–3):1–9, 2004. doi:10.1016/S1571-0653(05)80122-2.

[3] F. Gavril. Maximum weight independent sets and cliques in intersection graphs of filaments. *Information Processing Letters*, 73(5–6):181–188, 2000. doi:10.1016/S0020-0190(00)00025-9.

[4] F. Gavril. 3D-interval-filament graphs. *Discrete Applied Mathematics*, 155(18):2625–2636, 2007. doi:10.1016/j.dam.2007.08.006.

[5] I. Kanj, M. J. Pelsmajer, M. Schaefer, and G. Xia. On the induced matching problem. *Journal of Computer and System Sciences*, 77(6):1058–1070, 2011. doi:10.1016/j.jcss.2010.09.001.

[6] R. E. Tarjan and A. E. Trojanowski. Finding a maximum independent set. *SIAM Journal on Computing*, 6(3):537–546, 1977. doi:10.1137/0206038.

[7] G. Valiente. A new simple algorithm for the maximum-weight independent set problem on circle graphs. In T. Ibaraki, N. Katoh, and H. Ono, editors, *Proceedings of the International Symposium on Algorithms and Computation (ISAAC)*, pages 129–137. Springer, 2003. doi:10.1007/978-3-540-24587-2_15.

[8] M. Ward, A. Gozzard, M. Wise, and A. Datta. A faster algorithm for maximum induced matchings on circle graphs. *Journal of Graph Algorithms and Applications*, 22(2):389–396, 2018. doi:10.7155/jgaa.00476.