GROW 2017: Open Problems

Please send your open problems and comments to Lalla Mouatadid (Lalla@cs.toronto.edu) to be included here.

Irena Penev

A *ring* is a graph R whose vertex set can be partitioned into $k \geq 4$ nonempty sets, say $X_1, \dots, X_k$ (with subscripts understood to be in $\mathbb{Z}_k$), such that for all $i \in \mathbb{Z}_k$, X_i can be ordered as $X_i = \{u_1^i, \dots, u_{|X_i|}^i\}$ so that $X_i \subseteq N_R[u_{|X_i|}^i] \subseteq \dots \subseteq N_R[u_1^i] = X_{i-1} \cup X_i \cup X_{i+1}$. Under these circumstances, we say that the ring R is of *length* k, as well as that R is a k*-ring*. Furthermore, we say that $(X_1, \dots, X_k)$ is a *good partition* of the ring R. A ring is *odd* (resp. *even*) if its length is odd (resp. even). The following lemma readily follows from the definition of a ring.

Lemma. *Let G be a graph, and let $(X_1, \dots, X_k)$, with $k \geq 4$ and subscripts understood to be in $\mathbb{Z}_k$, be a partition of $V(G)$. Then G is a k-ring with good partition $(X_1, \dots, X_k)$ if and only if all the following hold:*

(a) $X_1, \dots, X_k$ are cliques;

(b) for all $i \in \mathbb{Z}_k$, X_i is anticomplete to $V(G) \setminus (X_{i-1} \cup X_i \cup X_{i+1})$;

(c) for all $i \in \mathbb{Z}_k$, some vertex of X_i is complete to $X_{i-1} \cup X_{i+1}$;

(d) for all $i \in \mathbb{Z}_k$, and all distinct $y_i, y_i' \in X_i$, one of y_i, y_i' dominates the other.

Even rings are easily seen to be polynomially colorable. Indeed, for even k (and notation as in the definition of a ring), we simply assign color j to x_j^i when i is odd, and we assign color $\omega(R) - j + 1$ to x_j^i when i is even.

Question. *Are odd rings polynomially colorable?*

Remark: It is not hard to show that all holes in a k-ring are of length k. Thus, if odd rings are NP-hard to color, then even-hole-free graphs are NP-hard to color.

Sang-il Oum

Branch-width of hypergraphs was defined in the paper of Robertson and Seymour [Graph Minors X, 1991].

Question: Do you know any work concerning algorithms to find a branch-decomposition of a hypergraph?

I've been preparing a manuscript with Jisu Jeong and Eun Jung Kim, that is going to give an $O(n^3)$-time algorithm, for n-vertex hypergraphs, to find a branch-decomposition of width at most k or confirm that the branch-width is larger than k for fixed k. I wonder whether there is any linear time algorithm known.

Till Miltzow

1 Reunite with Fifi

Imagine it is a beautiful spring day, after a long and cold winter. You are very excited and decide to go for a long hike into the nature with our puppy Fifi (no leash involved). You decide to go to the lake Oeschienensee in the Pochtenalp. Well, it happens what must have happened eventually, while you are enjoying the beautiful scenery, your puppy hunts a butterfly all around the lake. After some time, it gets a little chilly and you decide its time to get home and you start collecting Fifi. Luckily, you have a perfect map of the lake and you know the precise location of Fifi, as you connected her dog collar with a GPS device. So you can just go to her and collect her. However, the puppy also starts to feel lonely and homesick. Also the puppy knows perfectly where its owner is, even without GPS (Researchers still do not know how Fifi is doing this.). Now things become messy as the puppy tries to go towards you with sheer infinite speed. The problem is that Fifi gets easily trapped in local bend of the lake, where it stays without approaching you further. Note that Fifi would never enter the lake or deviate from the path around the lake, as there is a forest around the lake with misterious creatures that scare her (Probably only toads and crickets).

Question. *Show that you can always reunite with Fifi.*

Although it seems almost completely absurd that you cannot fetch Fifi, note that it is not sufficient to just walk around the lake in clockwise direction, as can be seen in Figure 1. It is easy

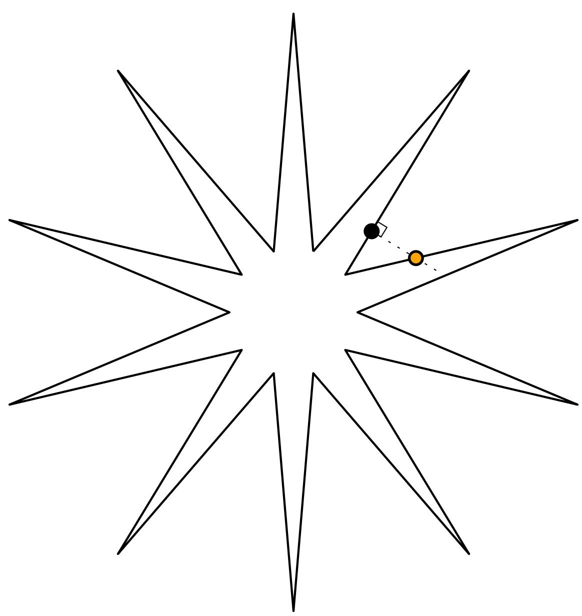

Figure 1: You are represented in orange and Fifi is represented in black. Note that Fifi is stuck in a local optimum and will not move unless you move. If you continue to walk in clockwise direction around the lake it is easy to check that Fifi will follow you, but is never able to get quite to you.

to show that you can fetch Fifi if the layout of the lake is rectilinear, but the argument does not carry over to general polygons. It is also known that the polygon that represents the lake boundary must be simple.

2 Coloring Unit Segment Graphs

In the result, I presented during the workshop, we showed that colouring n unit disk can be done in roughly $2^{O(\sqrt{n})}$ faster for a constant number of colours. This is based on separator theorems. Those separator theorems can be proved using volume arguments and thus work for any fat objects.

Those precise separator theorems do not exists for line segments and indeed, we can show that 6-colouring segment intersection graphs cannot be done in $2^{o(n)}$, unless the exponential time hypothesis (ETH) fails.

The lower bound and upper bound both fail for **unit** segments. This motivates the following question:

Question. *Can we colour unit segment intersection graphs with a constant number of colours (Say* 6 *or more.) in sub exponential time or is there a lower bound assuming the ETH or another complexity assumption?*

Laurent Feuilloley

A pattern is an ordered graph were couples of nodes can be related by three types of relations: plain edges, dashed edges, and non-edges. A subgraph of an ordered graph matches a pattern, if the plain edges are present, the dashed edges are absent (and there is no constraint on non-edges). Given a family of patterns, we can define the class of the graphs that have an ordering that avoids all the patterns, that is an ordering such that no ordered subgraph matches a pattern of the family.

The problems is: what about directed graphs? We can define similar concepts, then are the classes interesting? Also, for the undirected case it is known that we can recognize the classes based on patterns on three nodes in polynomial time, is this also true in the directed case ?