In Which Graph Structures Can We Efficiently Find Temporally Disjoint Paths and Walks?

Pascal Kunz
Hendrik Molter
Meirav Zehavi

Based on a publication in IJCAI'23

Parameterized Complexity

Parameterized problem.

Each instance is associated with a parameter *k*.

Fixed-Parameter Tractability

$$k << n$$

$$f(k) \cdot n^{O(1)} = O^*(f(k))$$

$$\binom{n}{k} \text{ ... or worse!}$$

Fixed-Parameter Tractability

$$k << n$$

$$f(k) \cdot n^{O(1)} = O^*(f(k))$$

$$\binom{n}{k}$$ *... or worse!*

Unfortunately, not all problems are FPT!

Parameterized Analysis and Temporal Graphs.

- Nowadays, there is a very large number of works on this topic.
- Parameters: Solution size, structural parameters, **new parameters** (e.g., maximum duration).
- Classification (FPT, XP & W[1]-hard, para-NP-hard), Optimality under (S)ETH, Kernelization. Even: Counting, FPT-approximation, …

$$f(k) \cdot n^{O(1)}$$

$$n^{f(k)}$$

Parameterized Analysis and Temporal Graphs.

- Nowadays, there is a very large number of works on this topic.
- Parameters: Solution size, structural parameters, **new parameters** (e.g., maximum duration).
- Classification (FPT, XP & W[1]-hard, para-NP-hard), Optimality under (S)ETH, Kernelization. Even: Counting, FPT-approximation, …

Check Hendrik Molter's works.

- **Survey:** ``As Time Goes By: Reflections on Treewidth for Temporal Graphs'' in ``Treewidth, Kernels and Algorithms'', 2020.

Outline

Background

Our Contribution

Some Technical Details

Open Problems

Background

The Disjoint Paths Problem. Given a graph G and a multiset of k terminal pairs $\{(s_i,t_i)\colon i\in[k]\}$, does there exist a collection of pairwise disjoint paths $\{P_i\colon i\in[k]\}$ where for every $i\in[k]$, the endpoints of P_i are s_i and t_i?

Background

The Disjoint Paths Problem. Given a graph G and a multiset of k terminal pairs $\{(s_i,t_i): i\in[k]\}$, does there exist a collection of pairwise disjoint paths $\{P_i: i\in[k]\}$ where for every $i\in[k]$, the endpoints of P_i are s_i and t_i?

The Graph Minors Project (23 papers, 1983-2004). Any minor-closed family of graphs can be characterized by a finite set of forbidden minors.

- ❖ The inspiration behind the birth of Parameterized Analysis.
- ❖ Yielded numerous concepts, algorithms and structural results.

Background

The Disjoint Paths Problem. Given a graph G and a multiset of k terminal pairs $\{(s_i,t_i): i\in[k]\}$, does there exist a collection of pairwise disjoint paths $\{P_i: i\in[k]\}$ where for every $i\in[k]$, the endpoints of P_i are s_i and t_i?

The Graph Minors Project (23 papers, 1983-2004). Any minor-closed family of graphs can be characterized by a finite set of forbidden minors.

- ❖ The inspiration behind the birth of Parameterized Analysis.
- ❖ Yielded numerous concepts, algorithms and structural results.

Disjoint Paths Is a Central Component of the Graph Minors Project.

Background

Known Results on General Graphs.

Galactic FPT Algorithms: $f(k)n^3$ (RS, ’95), $f(k)n^2$ (KKR, ’12).

No Polynomial Kernel: (FHW, ‘80).

On Directed Graphs: NP-hard even for k=2 (BTY, ‘11).

Background

Known Results on General Graphs.

Galactic FPT Algorithms: $f(k)n^3$ (RS, ’95), $f(k)n^2$ (KKR, ’12).

No Polynomial Kernel: (FHW, ‘80).

On Directed Graphs: NP-hard even for k=2 (BTY, ‘11).

Extensively Studied on Special Graph Classes. E.g., planar graphs, split graphs, chordal graphs, …

Background

Temporal Paths/Walks. A temporal walk in a temporal graph is a sequence $P=((v_0,v_1,t_1),(v_1,v_2,t_2),\ldots,(v_{r-1},v_r,t_r))$, $t_1<t_2<\ldots<t_r$, such that each (v_{i-1},v_i) is an edge in the temporal graph at time t_i. When no vertex is repeated, it is a temporal path.

Background

Temporal Paths/Walks. A temporal walk in a temporal graph is a sequence $P=((v_0,v_1,t_1),(v_1,v_2,t_2),\ldots,(v_{r-1},v_r,t_r))$, $t_1<t_2<\ldots<t_r$, such that each (v_{i-1},v_i) is an edge in the temporal graph at time t_i. When no vertex is repeated, it is a temporal path.

The walk occupies each v_i at time $[t_i,t_i+1]$ where $t_0=t_1$ and $t_{r+1}=t_r$.

Background

Temporal Paths/Walks. A temporal walk in a temporal graph is a sequence $P=((v_0,v_1,t_1),(v_1,v_2,t_2),\dots,(v_{r-1},v_r,t_r))$, $t_1<t_2<\dots<t_r$, such that each (v_{i-1},v_i) is an edge in the temporal graph at time t_i. When no vertex is repeated, it is a temporal path.
The walk occupies each v_i at time $[t_i,t_i+1]$ where $t_0=t_1$ and $t_{r+1}=t_r$.

Disjointness. Two temporal walks are temporally disjoint if there does not exist a vertex that they occupy at the same time.

Background

Temporal Paths/Walks. A temporal walk in a temporal graph is a sequence $P=((v_0,v_1,t_1),(v_1,v_2,t_2),\ldots,(v_{r-1},v_r,t_r))$, $t_1<t_2<\ldots<t_r$, such that each (v_{i-1},v_i) is an edge in the temporal graph at time t_i. When no vertex is repeated, it is a temporal path.
The walk occupies each v_i at time $[t_i,t_i+1]$ where $t_0=t_1$ and $t_{r+1}=t_r$.

Disjointness. Two temporal walks are temporally disjoint if there does not exist a vertex that they occupy at the same time.

The Temporally Disjoint Paths (Walks) Problem. Defined similarly to the Disjoint Paths problem.

Background

Known Results. The problems were introduced by Klobas, Mertzios, Molter, Niedermeir and Zschoche in IJCAI’21. They proved:

- **For Temporally Disjoint Paths:**
  - NP-hard when k=2,
  - NP-hard on paths,
  - FPT on trees wrt k.
- **For Temporally Disjoint Walks:**
  - W[1]-hard wrt k,
  - XP wrt k.

Outline

Background

Our Contribution

Some Technical Details

Open Problems

Our Contribution

I. Temporally Disjoint Paths and Walks are W[1]-hard wrt *n* on stars.

Our Contribution

I. Temporally Disjoint Paths and Walks are W[1]-hard wrt *n* on stars.

But what if *k* is part of the parameterization? (Possibly, *k*>>*n* since we are given a multiset of terminal pairs.)

Our Contribution

Recall: KMMNZ'21 proved that TDP is FPT wrt *k* on trees.

II. Temporally Disjoint Paths is W[1]-hard wrt *k+vcn*.

III. Temporally Disjoint Paths is FPT wrt *k+fes*.

Our Contribution

Recall: KMMNZ’21 proved that TDP is FPT wrt *k* on trees.

II. Temporally Disjoint Paths is W[1]-hard wrt *k*+*vcn*.

III. Temporally Disjoint Paths is FPT wrt *k*+*fes*.

IV. Temporally Disjoint Walks is W[1]-hard wrt *k* on stars (*vcn*=1, *fes*=0).

Our Contribution

Recall: KMMNZ'21 proved that TDP is FPT wrt *k* on trees.

II. Temporally Disjoint Paths is W[1]-hard wrt *k*+*vcn*.

III. Temporally Disjoint Paths is FPT wrt *k*+*fes*.

IV. Temporally Disjoint Walks is W[1]-hard wrt *k* on stars (*vcn*=1, *fes*=0).

- **Different than the behavior on general graphs:** TDP is NP-hard when *k*=2, while TDW is XP wrt *k* KMMNZ'21.

Our Contribution

Recall: KMMNZ’21 proved that TDP is FPT wrt *k* on trees.

II. Temporally Disjoint Paths is W[1]-hard wrt *k*+*vcn*.

III. Temporally Disjoint Paths is FPT wrt *k*+*fes*.

IV. Temporally Disjoint Walks is W[1]-hard wrt *k* on stars (*vcn*=1, *fes*=0).

- **Different than the behavior on general graphs:** TDP is NP-hard when *k*=2, while TDW is XP wrt *k* KMMNZ’21.
- Resolves open questions by KMMNZ’21.

Our Contribution

Recall: KMMNZ'21 proved that TDP is FPT wrt *k* on trees.

II. Temporally Disjoint Paths is W[1]-hard wrt *k*+*vcn*.

III. Temporally Disjoint Paths is FPT wrt *k*+*fes*.

IV. Temporally Disjoint Walks is W[1]-hard wrt *k* on stars (*vcn*=1, *fes*=0).

- **Different than the behavior on general graphs:** TDP is NP-hard when *k*=2, while TDW is XP wrt *k* KMMNZ'21.
- Resolves open questions by KMMNZ'21.

IV. Temporally Disjoint Walks is FPT wrt *k* on paths.

Outline

Background

Our Contribution

Some Technical Details

Open Problems

Some Technical Details

IV. Temporally Disjoint Walks is FPT wrt *k* on paths.

Some Technical Details

IV. Temporally Disjoint Walks is FPT wrt *k* on paths.

Main Component. Consider the ``changes of direction'' of the solution walks. We prove that there exist a solution s.t.:

- Their number is O(*k*) per walk.
- They only occur in *O*(*k*)-sized ``regions'' surrounding sources and sinks.

Some Technical Details

IV. Temporally Disjoint Walks is FPT wrt *k* on paths.

Main Component. Consider the ``changes of direction'' of the solution walks. We prove that there exist a solution s.t.:

- Their number is O(*k*) per walk.
- They only occur in *O*(*k*)-sized ``regions'' surrounding sources and sinks.

How to Exploit This? ``Guess'' where all the changes of directions occur.

Some Technical Details

IV. Temporally Disjoint Walks is FPT wrt *k* on paths.

Main Component. Consider the ``changes of direction'' of the solution walks. We prove that there exist a solution s.t.:

- Their number is O(*k*) per walk.
- They only occur in *O*(*k*)-sized ``regions'' surrounding sources and sinks.

How to Exploit This? ``Guess'' where all the changes of directions occur. Then, reduce to Temporally Disjoint Paths with ``precedence constraints'': Replace each terminal pair by a multiset of terminals pairs.
(Partition a solution walk into solution paths by ``cutting'' it every time it changes direction.)

Some Technical Details

IV. Temporally Disjoint Walks is FPT wrt *k* on paths.

Main Component. Consider the ``changes of direction'' of the solution walks. We prove that there exist a solution s.t.:

- Their number is O(*k*) per walk.
- They only occur in *O*(*k*)-sized ``regions'' surrounding sources and sinks.

How to Exploit This? ``Guess'' where all the changes of directions occur. Then, reduce to Temporally Disjoint Paths with ``precedence constraints'': Replace each terminal pair by a multiset of terminals pairs.
(Partition a solution walk into solution paths by ``cutting'' it every time it changes direction.)
Then, use an algorithm similar to that for Temporally Disjoint Paths on trees.

Some Technical Details

Main Component. Consider the ``changes of direction'' of the solution walks. We prove that there exist a solution s.t.:

- Their number is O(*k*) per walk.
- They only occur in *O*(*k*)-sized ``regions'' surrounding sources and sinks.

How to Prove This? Show that a walk changes direction because:

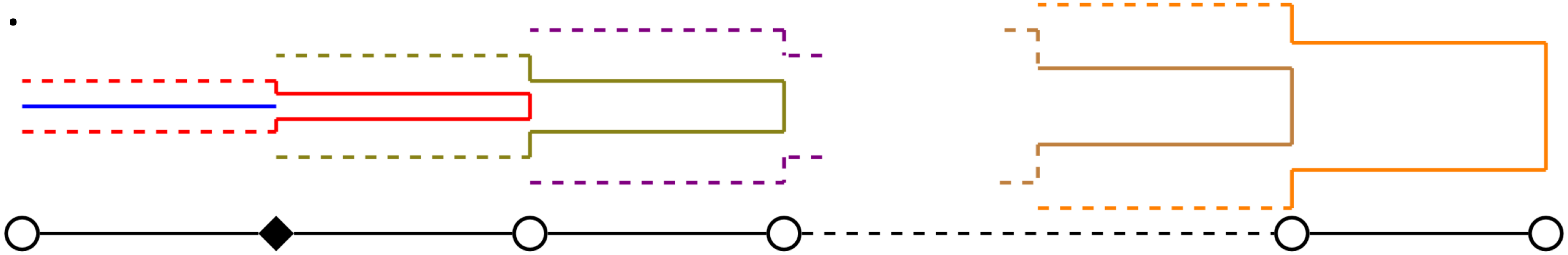
Some Technical Details

Main Component. Consider the ``changes of direction'' of the solution walks. We prove that there exist a solution s.t.:

- Their number is O(k) per walk.
- They only occur in $O(k)$-sized ``regions'' surrounding sources and sinks.

How to Prove This? Show that a walk changes direction because:

- It ``surrounds'' a solution walk that has just started or finished.
- It ``surrounds'' a solution walk that ``surrounds'' a solution walk that has just started or finished.
- … [k-1 times].



Outline

Background

Our Contribution

Some Technical Details

Open Problems

Open Problems

1. Consider parameters that are not related to *vc* and *fes*, and which are large on stars.
E.g.: Classify Temporally Disjoint Paths (Walks) wrt k + cutwidth/bandwidth.

Open Problems

1. Consider parameters that are not related to *vc* and *fes*, and which are large on stars.
E.g.: Classify Temporally Disjoint Paths (Walks) wrt k + cutwidth/bandwidth.

2. Our hardness results hold when each of the solution's paths/walks use only constantly many edges. But, in some of them, the maximum **duration** of the solution's paths/walks is large. So: Consider the duration as part of the parameterization.
E.g.: Classify Temporally Disjoint Paths (Walks) wrt k + maximum duration.