Optimising reachability by reordering

Kitty Meeks

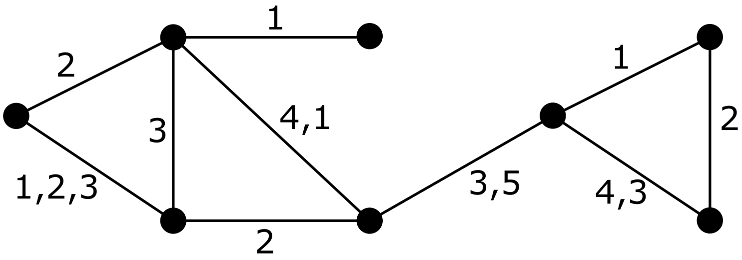
School of Computing Science, University of Glasgow

Algorithmic Aspects of Temporal Graphs II, Patras, Greece

8th July 2019

Joint work with Jessica Enright (University of Edinburgh) and Fiona Skerman (Masaryk University)

Reachability sets in temporal graphs



Assigning times to reduce reachability

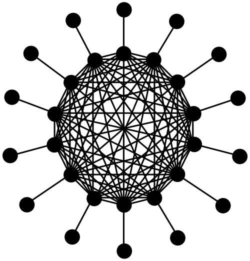
- Suppose we can change the *order* in which the edges are active, without changing the number of active times for each edge.

Assigning times to reduce reachability

- Suppose we can change the *order* in which the edges are active, without changing the number of active times for each edge.
- Our goal is to minimise the size of the *largest* reachability set.

Assigning times to reduce reachability

- Suppose we can change the *order* in which the edges are active, without changing the number of active times for each edge.
- Our goal is to minimise the size of the *largest* reachability set.



Our basic problem

The *temporal reachability set* of a vertex v in a temporal graph $(G, \mathcal{T})$ is the set of vertices u such that there exists a strict temporal path from v to u; we include v in this set.

The *maximum temporal reachability* of a temporal graph is the maximum cardinality of the temporal reachability set of any vertex v in the graph.

Our basic problem

The *temporal reachability set* of a vertex v in a temporal graph $(G, \mathcal{T})$ is the set of vertices u such that there exists a strict temporal path from v to u; we include v in this set.

The *maximum temporal reachability* of a temporal graph is the maximum cardinality of the temporal reachability set of any vertex v in the graph.

SINGLETON MIN-MAX REACHABILITY TEMPORAL ORDERING

Input: A graph $G = (V, E)$, and a positive integer k

Question: Is there a bijective function $\mathcal{T} \colon E \to [|E|]$ such that maximum temporal reachability of $(G, \mathcal{T})$ is at most k?

The problem is hard

Theorem

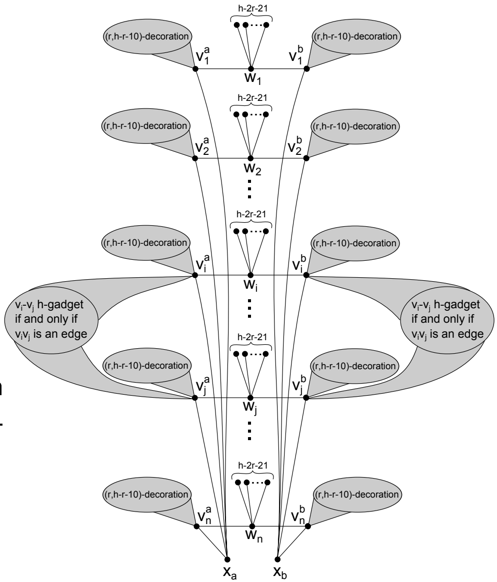
SINGLETON MIN-MAX REACHABILITY TEMPORAL ORDERING is NP-complete.

The problem is hard

Theorem

SINGLETON MIN-MAX REACHABILITY TEMPORAL ORDERING is NP-complete.

We give a reduction from Minimum Bisection for Cubic Graphs.



General properties and consequences

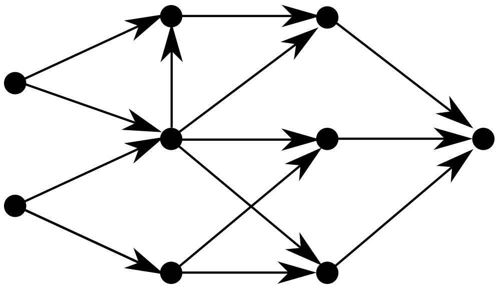
- *However we order the edges, the reachability set of every vertex contains all of its (out-)neighbours.*

General properties and consequences

- *However we order the edges, the reachability set of every vertex contains all of its (out-)neighbours.*

Theorem

The smallest achievable maximum reachability for a DAG is exactly the maximum out-degree plus one.



General properties and consequences

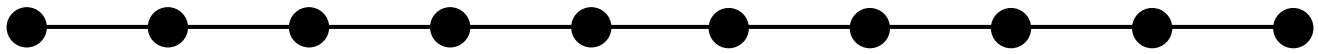
- *However we order the edges, there is a reachability set which contains all neighbours of two adjacent vertices.*

General properties and consequences

- *However we order the edges, there is a reachability set which contains all neighbours of two adjacent vertices.*

Theorem

For a path on at least five vertices, the smallest achievable maximum reachability is exactly four.



General properties and consequences

- *There is an order of edges that minimises the maximum reachability, so that all "leaf" edges are active before all others. Moreover, the relative order of leaf edges does not matter.*

General properties and consequences

- *There is an order of edges that minimises the maximum reachability, so that all "leaf" edges are active before all others. Moreover, the relative order of leaf edges does not matter.*

Theorem
There is an FPT algorithm to compute the optimum ordering of edges, parameterised by the number of vertices of degree at least two.

General properties and consequences

- *There is an order of edges that minimises the maximum reachability, so that all "leaf" edges are active before all others. Moreover, the relative order of leaf edges does not matter.*

Theorem

There is an FPT algorithm to compute the optimum ordering of edges, parameterised by the number of vertices of degree at least two.

Corollary

There is an FPT algorithm to compute the optimum ordering of edges for a tree, parameterised by the vertex cover number.

An algorithm for trees

The problem can be solved in linear time on trees when k (max permitted reachability) is bounded by a fixed constant:

An algorithm for trees

The problem can be solved in linear time on trees when k (max permitted reachability) is bounded by a fixed constant:

- Recall: if max degree $\geq k$ then we have a no-instance
- Starting at leaves, and working towards the root:
  - Consider all *relative* orderings of the edges incident with a single vertex v, and suppose that the edge from v to its parent is active at time t_v.

An algorithm for trees

The problem can be solved in linear time on trees when k (max permitted reachability) is bounded by a fixed constant:

- Recall: if max degree $\geq k$ then we have a no-instance
- Starting at leaves, and working towards the root:
  - Consider all *relative* orderings of the edges incident with a single vertex v, and suppose that the edge from v to its parent is active at time t_v.
  - For each possible ordering, and every pair (α, β) with $1 \leq \alpha, \beta \leq k$, determine whether there is an ordering of all edges incident with v and its descendants such that:
    1. no descendant of v has a reachability set of size $> k$,
    2. no descendant of v which reaches v before time t_v has size more than α, and
    3. at most β descendants of v are reachable from v after time t_v.

An algorithm for trees

The problem can be solved in linear time on trees when k (max permitted reachability) is bounded by a fixed constant:

- Recall: if max degree $\geq k$ then we have a no-instance
- Starting at leaves, and working towards the root:
  - Consider all *relative* orderings of the edges incident with a single vertex v, and suppose that the edge from v to its parent is active at time t_v.
  - For each possible ordering, and every pair (α, β) with $1 \leq \alpha, \beta \leq k$, determine whether there is an ordering of all edges incident with v and its descendants such that:
    1. no descendant of v has a reachability set of size $> k$,
    2. no descendant of v which reaches v before time t_v has size more than α, and
    3. at most β descendants of v are reachable from v after time t_v.

This general method can also be generalised to graphs of bounded treewidth.

General properties and consequences (cont.)

- Changing the relative order of two non-incident edges (while leaving all other relative orderings unchanged) doesn't change any reachability sets

General properties and consequences (cont.)

- Changing the relative order of two non-incident edges (while leaving all other relative orderings unchanged) doesn't change any reachability sets
- If the edge chromatic number of the graph is c, there is an ordering of the edges so that no reachability set has size more than 2^c:

General properties and consequences (cont.)

- Changing the relative order of two non-incident edges (while leaving all other relative orderings unchanged) doesn't change any reachability sets
- If the edge chromatic number of the graph is c, there is an ordering of the edges so that no reachability set has size more than 2^c:
  - we can decompose the edges into at most c sets of disjoint edges

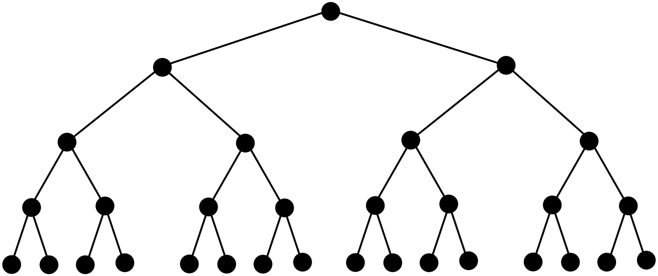
General properties and consequences (cont.)

- Changing the relative order of two non-incident edges (while leaving all other relative orderings unchanged) doesn't change any reachability sets
- If the edge chromatic number of the graph is c, there is an ordering of the edges so that no reachability set has size more than 2^c:
  - we can decompose the edges into at most c sets of disjoint edges
  - adding a set of disjoint edges, with active times later than any other edges, at most doubles the size of any reachability set

General properties and consequences (cont.)

Theorem

The smallest achievable maximum reachability for a (large enough) binary tree is exactly eight.



An approximation algorithm

Theorem

Given any graph G, we can compute a $\frac{2^{\Delta(G)+1}}{\Delta(G)+1}$-approximation to the smallest achievable maximum reachability in linear time, where $\Delta(G)$ denotes the maximum degree of G. Moreover, we can also compute an assignment of times to edges which achieves this approximation ratio in polynomial time.

An approximation algorithm

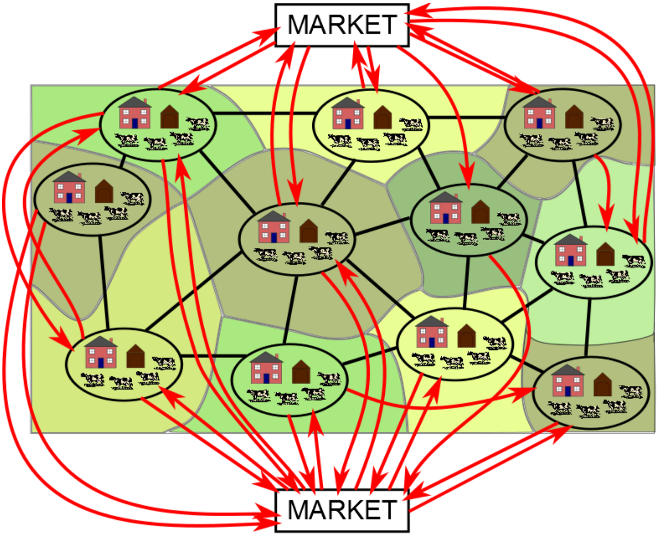
Theorem

Given any graph G, we can compute a $\frac{2^{\Delta(G)+1}}{\Delta(G)+1}$-approximation to the smallest achievable maximum reachability in linear time, where $\Delta(G)$ denotes the maximum degree of G. Moreover, we can also compute an assignment of times to edges which achieves this approximation ratio in polynomial time.

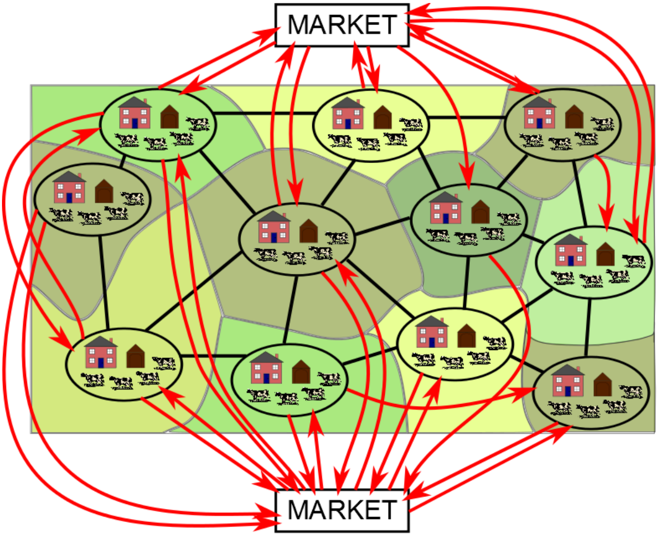
Corollary

Given any graph G of bounded maximum degree, we can compute a constant factor approximation in linear time.

Generalisations



Generalisations



- Specified classes of edges have to be active simultaneously.

Generalisations

Theorem

With this generalisation, the problem is NP-hard even if the input graph is (a) a directed acyclic graph, or (b) a tree.

Sets of simultaneous edges

Theorem

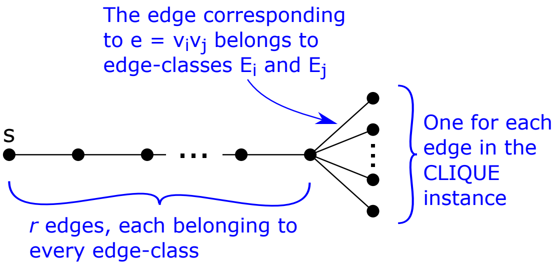
When we can specify classes of edges which must be active simultaneously, the problem is W[1]-hard parameterised by the vertex cover number, even when restricted to trees.

Sets of simultaneous edges

Theorem

When we can specify classes of edges which must be active simultaneously, the problem is W[1]-hard parameterised by the vertex cover number, even when restricted to trees.

- Reduction from Clique.
- We have one edge-class for each vertex in the Clique instance.

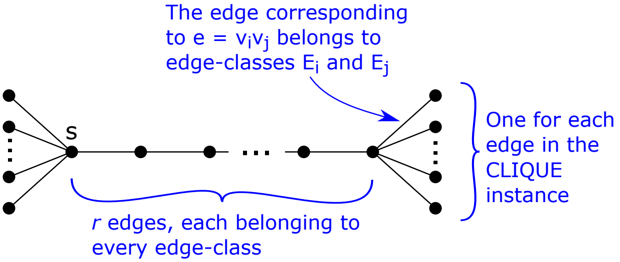


Sets of simultaneous edges

Theorem

When we can specify classes of edges which must be active simultaneously, the problem is W[1]-hard parameterised by the vertex cover number, even when restricted to trees.

- Reduction from Clique.
- We have one edge-class for each vertex in the Clique instance.



The edge-class interaction graph

The edge-class interaction graph has the set of edge-classes as its vertex set, and two edge-classes are adjacent if and only if some vertex of the original graph is incident with an edge from both classes.

The edge-class interaction graph

The edge-class interaction graph has the set of edge-classes as its vertex set, and two edge-classes are adjacent if and only if some vertex of the original graph is incident with an edge from both classes.

Theorem

There is a ordering of the edge-classes such that the maximum reachability is at most $(d+1)^{\chi}$, where

- *d is the maximum degree in any single edge class, and*
- *χ is the chromatic number of the edge-class interaction graph.*

The edge-class interaction graph

The edge-class interaction graph has the set of edge-classes as its vertex set, and two edge-classes are adjacent if and only if some vertex of the original graph is incident with an edge from both classes.

Theorem

There is a ordering of the edge-classes such that the maximum reachability is at most $(d+1)^{\chi}$, where

- *d is the maximum degree in any single edge class, and*
- *χ is the chromatic number of the edge-class interaction graph.*

Corollary

We can compute a d-approximation to the smallest achievable maximum reachability in polynomial time if the edge-class interaction graph is bipartite, where d is the maximum degree of any single edge class.

Future directions

- Explore new techniques for approximation algorithms (with better approximation ratios)

Future directions

- Explore new techniques for approximation algorithms (with better approximation ratios)
- Include lists of permitted times for each edge(-class), possibly with associated costs

Future directions

- Explore new techniques for approximation algorithms (with better approximation ratios)
- Include lists of permitted times for each edge(-class), possibly with associated costs
- Consider maximising the size of the smallest reachability set

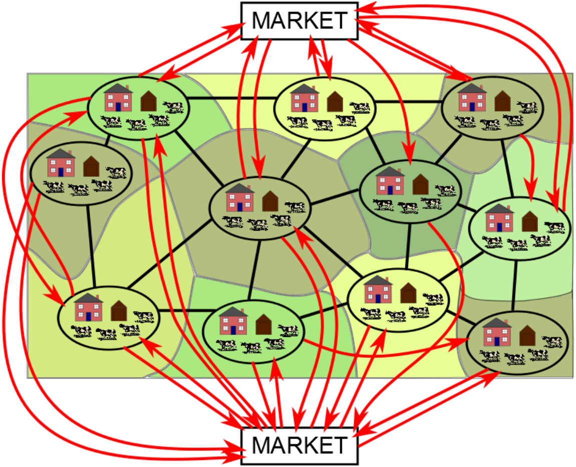
Future directions

- Explore new techniques for approximation algorithms (with better approximation ratios)
- Include lists of permitted times for each edge(-class), possibly with associated costs
- Consider maximising the size of the smallest reachability set
- What about minimising/maximising instead:
  - the *average* size of reachability sets, or
  - the *expected* worst-case or average reachability set in a model where each edge has an associated transmission probability?

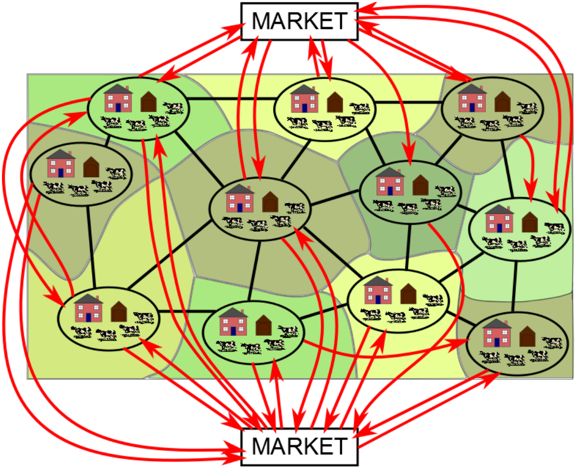
Future directions

- Explore new techniques for approximation algorithms (with better approximation ratios)
- Include lists of permitted times for each edge(-class), possibly with associated costs
- Consider maximising the size of the smallest reachability set
- What about minimising/maximising instead:
  - the *average* size of reachability sets, or
  - the *expected* worst-case or average reachability set in a model where each edge has an associated transmission probability?
- Systems with a combination of static and reorderable edges

Future directions



Future directions



THANK YOU

arxiv.org/abs/1802.05905