The temporal explorer who returns to the base [1]

Eleni C. Akrida[†], *George B. Mertzios*[♮], *and Paul G. Spirakis*[†,§]

[†]Department of Computer Science, University of Liverpool, UK

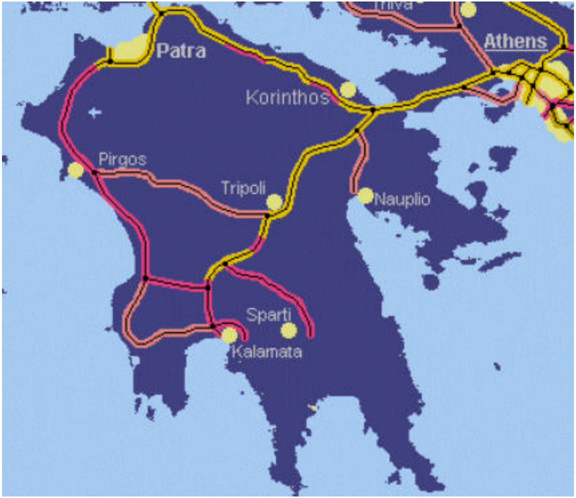
[♮]Department of Computer Science, Durham University, UK

[§]Department of Computer Engineering & Informatics, University of Patras, Greece

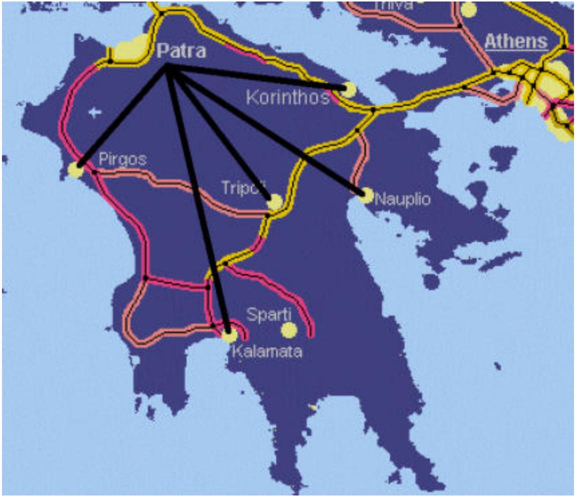
July 9, 2018

[1]Supported by the NeST initiative of the School of EEE and CS at the University of Liverpool and by the EPSRC Grants EP/P020372/1 and EP/P02002X/1.

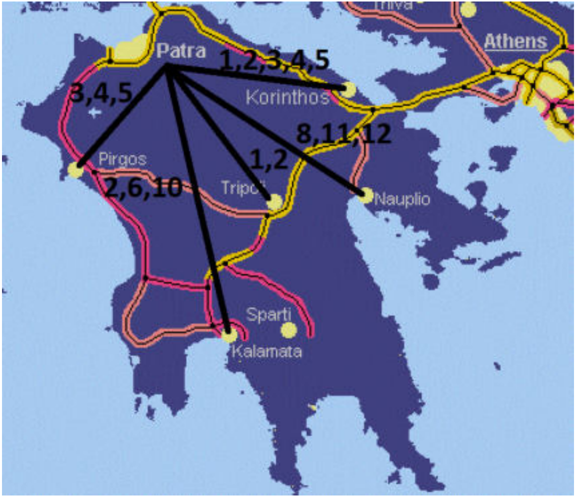
Returning to the base



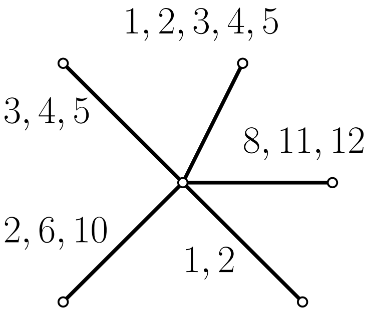
Returning to the base



Returning to the base



Returning to the base



Temporal Graphs

Definition (Temporal Graph)

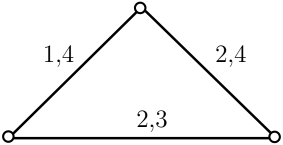
Let $G = (V, E)$ be a graph. A temporal graph on G is a pair (G, L), where $L : E \to 2^{\mathbb{N}}$ is a time-labeling function, called a *labeling* of G, which assigns to every edge of G a set of discrete-time labels. The labels of an edge are the *discrete time instances* at which it is available.

Temporal Graphs

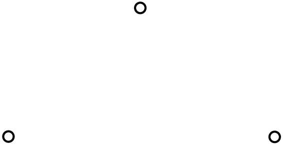
Definition (Temporal Graph)

Let $G = (V, E)$ be a graph. A temporal graph on G is a pair (G, L), where $L : E \to 2^{\mathbb{N}}$ is a time-labeling function, called a *labeling* of G, which assigns to every edge of G a set of discrete-time labels. The labels of an edge are the *discrete time instances* at which it is available.

temporal graph:



temporal instances:

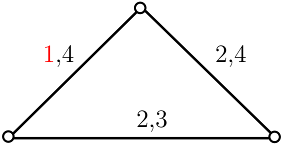


Temporal Graphs

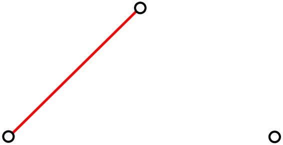
Definition (Temporal Graph)

Let $G = (V, E)$ be a graph. A temporal graph on G is a pair (G, L), where $L : E \to 2^{\mathbb{N}}$ is a time-labeling function, called a *labeling* of G, which assigns to every edge of G a set of discrete-time labels. The labels of an edge are the *discrete time instances* at which it is available.

temporal graph:



temporal instances:

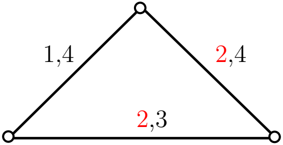


Temporal Graphs

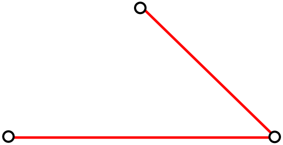
Definition (Temporal Graph)

Let $G = (V, E)$ be a graph. A temporal graph on G is a pair (G, L), where $L : E \to 2^{\mathbb{N}}$ is a time-labeling function, called a *labeling* of G, which assigns to every edge of G a set of discrete-time labels. The labels of an edge are the *discrete time instances* at which it is available.

temporal graph:



temporal instances:

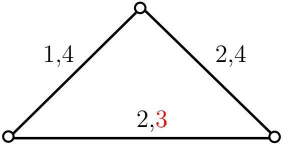


Temporal Graphs

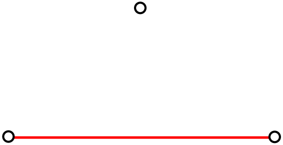
Definition (Temporal Graph)

Let $G = (V, E)$ be a graph. A temporal graph on G is a pair (G, L), where $L : E \to 2^{\mathbb{N}}$ is a time-labeling function, called a *labeling* of G, which assigns to every edge of G a set of discrete-time labels. The labels of an edge are the *discrete time instances* at which it is available.

temporal graph:



temporal instances:

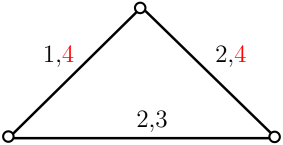


Temporal Graphs

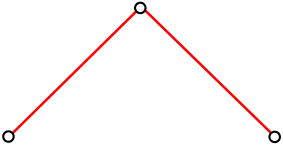
Definition (Temporal Graph)

Let $G = (V, E)$ be a graph. A temporal graph on G is a pair (G, L), where $L : E \to 2^{\mathbb{N}}$ is a time-labeling function, called a *labeling* of G, which assigns to every edge of G a set of discrete-time labels. The labels of an edge are the *discrete time instances* at which it is available.

temporal graph:



temporal instances:



Temporal Graphs

Definition (Temporal Star)

A temporal star is a temporal graph (G_s, L) on a star graph $G_s = (V, E)$. We denote by c the center of G_s.

Definition (Time edge)

Let $e = \{u, v\}$ be an edge of the underlying graph of a temporal graph and consider a label $l \in L(e)$. The ordered triplet (u, v, l) is called *time edge*.

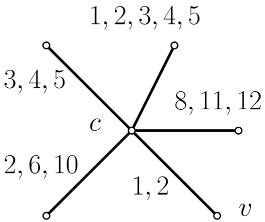
Temporal Graphs

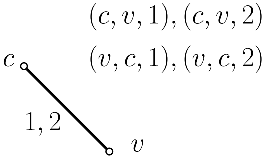
Definition (Temporal Star)

A temporal star is a temporal graph (G_s, L) on a star graph $G_s = (V, E)$. We denote by c the center of G_s.

Definition (Time edge)

Let $e = \{u, v\}$ be an edge of the underlying graph of a temporal graph and consider a label $l \in L(e)$. The ordered triplet (u, v, l) is called *time edge*.





Temporal Graphs

Definition (Journey)

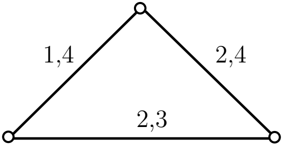
A *temporal path* or *journey* j from a vertex u to a vertex v ((u, v)*-journey*) is a sequence of time edges (u, u_1, l_1), (u_1, u_2, l_2), $\ldots$, (u_{k-1}, v, l_k), such that $l_i < l_{i+1}$, for each $1 \leq i \leq k-1$.

Temporal Graphs

Definition (Journey)

A *temporal path* or *journey* j from a vertex u to a vertex v (*(u, v)-journey*) is a sequence of time edges (u, u_1, l_1), (u_1, u_2, l_2), $\ldots$, (u_{k-1}, v, l_k), such that $l_i < l_{i+1}$, for each $1 \leq i \leq k-1$.

temporal graph:

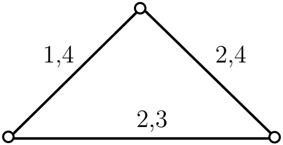


Temporal Graphs

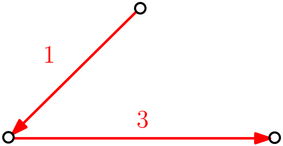
Definition (Journey)

A *temporal path* or *journey* j from a vertex u to a vertex v (*(u,v)-journey*) is a sequence of time edges $(u, u_1, l_1), (u_1, u_2, l_2), \ldots, (u_{k-1}, v, l_k)$, such that $l_i < l_{i+1}$, for each $1 \leq i \leq k-1$.

temporal graph:



journey:



Temporal Graphs

Definition (Exploration)

A (partial) exploration of a temporal star is a journey J that starts and ends at the center of G_s which visits some nodes of G_s; its size $|J|$ is the number of nodes of G_s that are visited by J.

Temporal Graphs

Definition (Exploration)

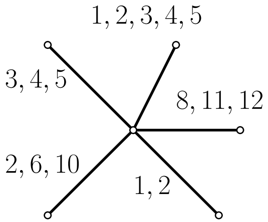
A (partial) exploration of a temporal star is a journey J that starts and ends at the center of G_s which visits some nodes of G_s; its size $|J|$ is the number of nodes of G_s that are visited by J.

- We "enter" (resp. "exit") an edge when we cross it from center to leaf (resp. leaf to center) at a time on which the edge is available.
- We can assume that in an exploration the entry to any edge e is followed by the exit from e at the earliest possible time. Waiting at a leaf (instead of exiting as soon as possible) does not help in exploring more edges.

Temporal Graphs

Definition (Exploration)

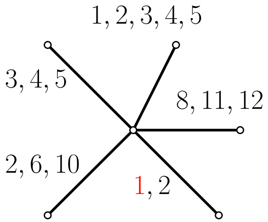
A (partial) exploration of a temporal star is a journey J that starts and ends at the center of G_s which visits some nodes of G_s; its size $|J|$ is the number of nodes of G_s that are visited by J.



Temporal Graphs

Definition (Exploration)

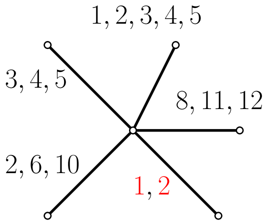
A (partial) exploration of a temporal star is a journey J that starts and ends at the center of G_s which visits some nodes of G_s; its size $|J|$ is the number of nodes of G_s that are visited by J.



Temporal Graphs

Definition (Exploration)

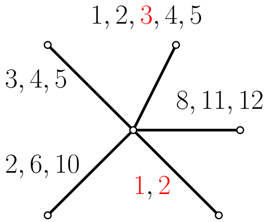
A (partial) exploration of a temporal star is a journey J that starts and ends at the center of G_s which visits some nodes of G_s; its size $|J|$ is the number of nodes of G_s that are visited by J.



Temporal Graphs

Definition (Exploration)

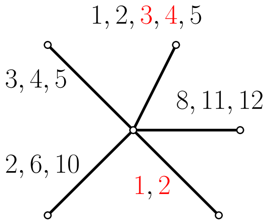
A (partial) exploration of a temporal star is a journey J that starts and ends at the center of G_s which visits some nodes of G_s; its size $|J|$ is the number of nodes of G_s that are visited by J.



Temporal Graphs

Definition (Exploration)

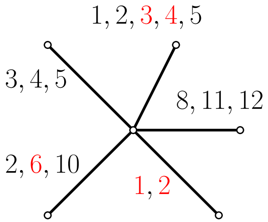
A (partial) exploration of a temporal star is a journey J that starts and ends at the center of G_s which visits some nodes of G_s; its size $|J|$ is the number of nodes of G_s that are visited by J.



Temporal Graphs

Definition (Exploration)

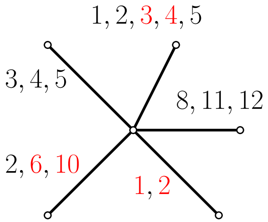
A (partial) exploration of a temporal star is a journey J that starts and ends at the center of G_s which visits some nodes of G_s; its size $|J|$ is the number of nodes of G_s that are visited by J.



Temporal Graphs

Definition (Exploration)

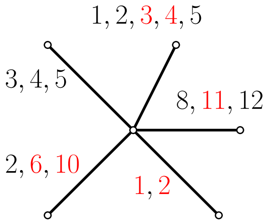
A (partial) exploration of a temporal star is a journey J that starts and ends at the center of G_s which visits some nodes of G_s; its size $|J|$ is the number of nodes of G_s that are visited by J.



Temporal Graphs

Definition (Exploration)

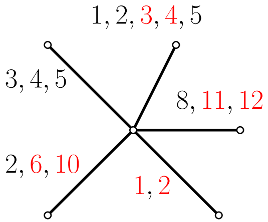
A (partial) exploration of a temporal star is a journey J that starts and ends at the center of G_s which visits some nodes of G_s; its size $|J|$ is the number of nodes of G_s that are visited by J.



Temporal Graphs

Definition (Exploration)

A (partial) exploration of a temporal star is a journey J that starts and ends at the center of G_s which visits some nodes of G_s; its size $|J|$ is the number of nodes of G_s that are visited by J.



The problems

StarExp(k)

Input: A temporal star (G_s, L) such that every edge has at most k labels.
Question: Is (G_s, L) explorable?

MaxStarExp(k)

Input: A temporal star (G_s, L) such that every edge has at most k labels.
Output: A (partial) exploration of (G_s, L) of *maximum* size.

Overview of results

- MaxStarExp(2) can be efficiently solved in $O(n \log n)$ time
- StarExp(3) can be solved in $O(n \log n)$ time
- StarExp(k) is NP-complete and MaxStarExp(k) is APX-hard, when $k \geq 6$
- Greedy 2-approximation algorithm for MaxStarExp(k)
- Characterisation of temporal stars with random labels that asymptotically almost surely admit a complete exploration

MaxStarExp(2) solution in $O(n \log n)$ time

MaxStarExp(2) is reducible to the Interval Scheduling Maximization Problem (ISMP).

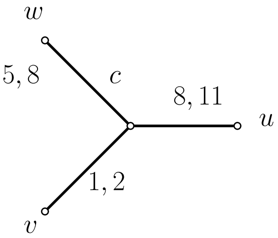
Interval Scheduling Maximization Problem (ISMP)

Input: A set of intervals, each with a start and a finish time.

Output: Find a set of non-overlapping intervals of maximum size.

MaxStarExp(2) solution in $O(n \log n)$ time

- Every edge e can be viewed as an interval to be scheduled.
- Any (partial) exploration of (G_s, L) corresponds to a set of non-overlapping intervals of the same size as the exploration, and vice versa.



Interval 1: (8, 11.5)
Interval 2: (1, 2.5)
Interval 3: (5, 8.5)

MaxStarExp(2) solution in $O(n \log n)$ time

Greedy optimal solution for ISMP:

1. Start with the set $S = E$ of all edges. Select the edge, e, with the smallest largest label (equivalent to the earliest finish time of the corresponding interval).
2. Remove from S the edge e and all conflicting edges.
3. Repeat until S is empty.

Time needed: $(|E| \log |E|) = O(n \log n)$

StarExp(3) solution in $O(n^2)$ time

- Note that if e has 2 labels, it must be explored by entering at the smallest and leaving at the largest label.
- The instance is reduced to a smaller one, with only edges with three labels, by removing all conflicting labels with the exploration of e from other edges.
- We reduce MaxStarExp(3) to 2SAT.
- For every edge e with labels l_1, l_2, l_3, we define the two possible exploration windows $[l_1, l_2], [l_2, l_3]$.
- We assign to e a Boolean variable x_e such that the truth assignment $x_e = 0$ (resp. $x_e = 1$) means that edge e is explored in the 1st interval (resp. 2nd interval).

StarExp(3) solution in $O(n^2)$ time

- For any two edges e_1 and e_2 with conflicting exploration windows, we add clauses:
  - $(x_1 \vee x_2)$ if the first exploration window of e_1 is conflicting with the first exploration window of e_2.
  - $(\neg x_1 \vee \neg x_2))$ if the second exploration window of e_1 is conflicting with the second exploration window of e_2.
  - $(\neg x_1 \vee x_2)$ if the second exploration window of e_1 is conflicting with the first exploration window of e_2.
  - $(x_1 \vee \neg x_2)$ if the first exploration window of e_1 is conflicting with the second exploration window of e_2.
- The constructed 2-CNF formula is satisfiable if and only if (G_s, L) is explorable.
- The formula contains $O(n^2)$ clauses in total, and thus the exploration problem can be solved in $O(n^2)$ time using a linear-time algorithm for 2SAT.

StarExp(3) solution in $O(n \log n)$ time

The idea:

- Reduce StarExp(3) to 2SAT where the number of clauses in the constructed formula is linear in n.
- Sort the $3n$ labels of (G_s, L) and scan through them to detect conflicts.

Hardness for $k \geq 6$ labels per edge

Theorem

StarExp(k)*is NP-complete and* MaxStarExp(k) *is APX-hard, for every* $k \geq 6$.

Reduction from 3SAT(3):

3SAT(3)

Input: A boolean formula F in CNF with variables $x_1, x_2, \ldots, x_p$ and clauses $c_1, c_2, \ldots, c_q$, such that each clause has at most 3 literals, and each variable appears in at most 3 clauses.
Output: Decision on whether the formula is satisfiable.

Hardness for $k \geq 6$ labels per edge

Theorem

StarExp(k)*is NP-complete and* MaxStarExp(k) *is APX-hard, for every* $k \geq 6$.

Reduction from 3SAT(3):

3SAT(3)

Input: A boolean formula F in CNF with variables $x_1, x_2, \ldots, x_p$ and clauses $c_1, c_2, \ldots, c_q$, such that each clause has at most 3 literals, and each variable appears in at most 3 clauses.
Output: Decision on whether the formula is satisfiable.

- Wlog assume every variable occurs once negated, $\neg x_i$, and at most twice non-negated, x_i.

The reduction

- (G_s, L) has one edge per clause, and three edges per variable (one "primary" and two "auxiliary") of F.
- The "primary" edge corresponding to a variable x has two pairs of labels, the 1st corresponding to $x = 0$ and the 2nd corresponding to $x = 1$.

The reduction

- Any edge corresponding to a clause containing x has an (entry, exit) pair of labels conflicting with the 1st pair of labels of the edge corresponding to x (associated with $x = 0$) but not with the 2nd pair.
- Any edge corresponding to a clause containing $\neg x$ has an (entry, exit) pair of labels conflicting with the 2nd pair of labels of the edge corresponding to x (associated with $x = 1$) but not with the 1st pair.
- For every variable x we have two "auxiliary" edges:
  - The first one to avoid entering and exiting the primary edge corresponding to x using labels from different pairs.
  - The second one to avoid entering an edge corresponding to some clause using a label associated with x and exiting using a label associated with a *different* variable y.

Greedy 2-approximation for MaxStarExp(k)

- Notice: there is an optimal solution exploring an edge e which has an exploration window with the earliest exit time.

Greedy 2-approximation for MaxStarExp(k)

- Notice: there is an optimal solution exploring an edge e which has an exploration window with the earliest exit time.
- Indeed, suppose that it is not the case and consider an optimal solution not exploring e.

Greedy 2-approximation for MaxStarExp(k)

- Notice: there is an optimal solution exploring an edge e which has an exploration window with the earliest exit time.
- Indeed, suppose that it is not the case and consider an optimal solution not exploring e.
- One can exchange the explored edge of this solution that has earliest exit time with the edge e using its first exploration window.

Greedy 2-approximation for MaxStarExp(k)

Input: a temporal star graph (G_s, L) with at most k labels per edge, $k \in \mathbb{N}^*$

Output: a (partial) exploration of (G_s, L)

Initialize the set of candidate edges to be $\mathcal{C} = E$;

Initialize the set of explored edges to be $Exp = \emptyset$;

t:=0;

while $\mathcal{C} \neq \emptyset$ **do**

  Find $e \in \mathcal{C}$ to be explored with entry time at least t and minimum exit time. Let t_0 be said exit time;

  Add e to the set of explored edges, Exp (with exploration window from t until t_0);

  Remove e from the set of candidate edges, $\mathcal{C}$;

  $t = t_0 + 1$;

  if *no* $e \in \mathcal{C}$ *has* 2 *labels greater or equal to* t **then**

    break;

  end

end

k random labels per edge: The setting

- Each edge of G_s receives k labels independently of other edges, and each label is chosen uniformly at random and independently of others from the set of integers $\{1, 2, \ldots, \alpha\}$, for some $\alpha \in \mathbb{N}$.

k random labels per edge: The setting

- Each edge of G_s receives k labels independently of other edges, and each label is chosen uniformly at random and independently of others from the set of integers $\{1, 2, \ldots, \alpha\}$, for some $\alpha \in \mathbb{N}$.
- *Uniform random temporal star*, $G_s(\alpha, k)$.

k random labels per edge: The setting

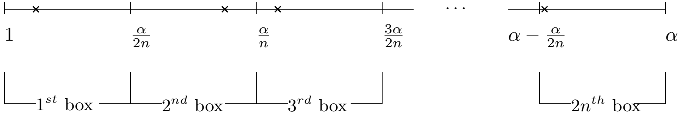
- Each edge of G_s receives k labels independently of other edges, and each label is chosen uniformly at random and independently of others from the set of integers $\{1, 2, \ldots, \alpha\}$, for some $\alpha \in \mathbb{N}$.
- *Uniform random temporal star*; $G_s(\alpha, k)$.
- Goal: investigate the explorability of a uniform random temporal star based on different values of α and k.

Case: $\alpha \geq 2n$ and $k \geq 6n \ln n$

Theorem

If $\alpha \geq 2n$ and $k \geq 6n \ln n$, then the probability that we can explore all edges of $G_s(\alpha, k)$ tends to 1 as n tends to infinity.

Proof sketch.



We show that for every edge of G_s, there will be asymptotically almost surely at least one of its labels that falls in the first box, one of its labels that falls in the second box, etc.

Observation

If for every edge $e \in E$ and for every box B_i there is at least one label of e that lies within B_i, then there exists an exploration of $G_s(\alpha, k)$.

Case: $\alpha \geq 4$ and $k = 2$

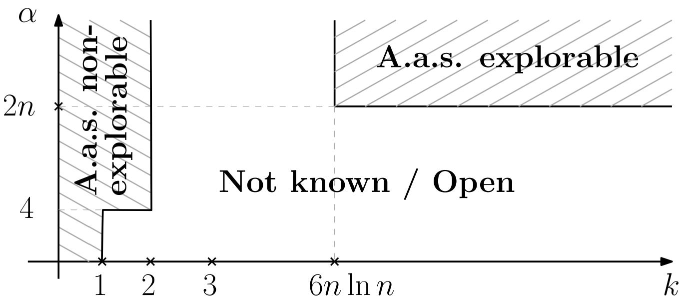
Theorem

If $\alpha \geq 4$ and $k = 2$, then the probability that we can explore all edges of $G_s(\alpha, k)$ tends to zero as n tends to infinity.

Proof idea.

- We introduce the notion of blocking pairs of edges.
- We show that for two particular edges, they are blocking asymtotically almost surely.
- We arbitrarily group all edges of $G_s(\alpha, 2)$ into $\lfloor \frac{n-1}{2} \rfloor$ independent pairs.
- If there is an exploration in $G_s(\alpha, 2)$, then there are no blocking pairs of edges in any such pairing.
- We show that there is no exploration asymptotically almost surely

Explorability of $G_s(\alpha, k)$



The shaded areas of the chart indicate the pairs (α, k) for which $G_s(\alpha, k)$ is asymptotically almost surely (a.a.s.) explorable and non-explorable, respectively.

Open problems

- Complexity of the maximization problem MaxStarExp(3)
- Complexity of StarExp(k) and MaxStarExp(k), for $k \in \{4, 5\}$
- Variation of StarExp(k) and MaxStarExp(k) where the consecutive labels of every edge are λ time steps apart, for some $\lambda \in \mathbb{N}$; complexity and/or best approximation factor

Open problems

- Complexity of the maximization problem MaxStarExp(3)
- Complexity of StarExp(k) and MaxStarExp(k), for $k \in \{4, 5\}$
- Variation of StarExp(k) and MaxStarExp(k) where the consecutive labels of every edge are λ time steps apart, for some $\lambda \in \mathbb{N}$; complexity and/or best approximation factor

Thank you

arxiv.org/abs/1805.04713