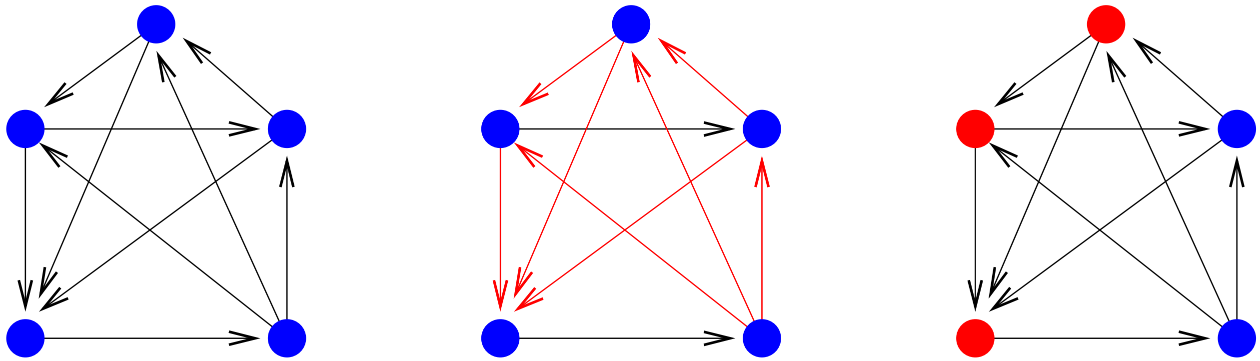
Algorithmic aspects of finding large acyclic subgraphs of directed graphs

Alantha Newman
G-SCOP, Grenoble

Directed Graphs

A directed graph (digraph) is a graph where each edge has a direction (i.e., an arc).



We look for either

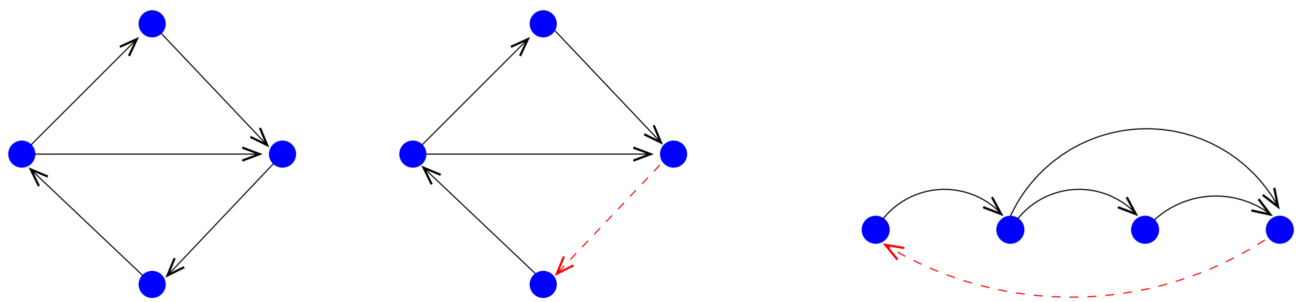
- a set of arcs that contains no directed cycles (i.e., acyclic subgraph),
- or a set of vertices whose induced subgraph contains no directed cycles (i.e., acyclic set).

Part I: Acyclic Subgraphs of Directed Graphs

The Maximum Acyclic Subgraph Problem

Given a directed graph $G = (V, E)$, the maximum acyclic subgraph problem is to find a maximum cardinality subset of the edges that is acyclic.

This problem is also known as the linear ordering problem.



Complement of the minimum feedback arc set problem: find a minimum weight subset $F \subseteq E$ such that $G' = (V, E \setminus F)$ is acyclic.

(A max acyclic subgraph in an undirected graph is a spanning tree.)

Simple $\frac{1}{2}$-Approximation

Take any ordering of the vertices. Either the set of forward edges or the set of backward edges contains at least half of the edges.

Exist instances where $OPT \approx |E|/2$, so we can't do better in general.

Open Problem: Can we find acyclic subgraph of size $\geq \frac{(1+\epsilon)}{2} OPT$?

NP-hard to approximate to within better than $\frac{14}{15}$ [Austrin, Manokaran, Wenner 2015].

Unique-Games hard to do better than half [Guruswami, Håstad, Manokaran, Raghavendra, Charikar 2008].

Goal is to distinguish between instances where

- $OPT \approx |E|$, and
- $OPT \approx |E|/2$.

In first case, we want solution $\gg |E|/2$.

In latter case, we want an upper bound $\ll |E|$.

Approximating MAS

$G = (V, E)$ is a digraph.

Let A be symmetric adjacency matrix for underlying (undirected) graph of G.

Let λ be the minimum (non-zero) eigenvalue of the normalized Laplacian for G, $L := I - \frac{1}{d}A$.

Theorem: There is an algorithm for maximum acyclic subgraph with approximation ratio $\frac{2}{4-\lambda}$.

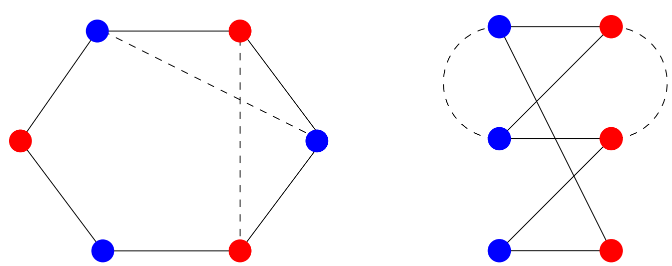
$\lambda = \min_{x \in \mathrm{R}^n - \{0\}} \frac{x^t L x}{x^t x}$.

Characterization of λ used by [Arora, Khot, Kolla, Steurer, Tulsiani, Vishnoi 2008] for Unique Games algorithm:

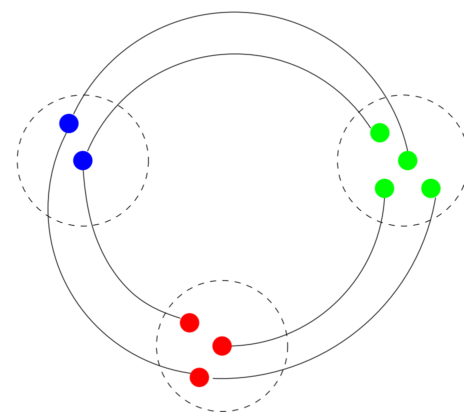
$$\lambda = \min \frac{\mathbb{E} \sum_{uv \in E} |z_u - z_v|^2}{\mathbb{E} \sum_{u,v \in V} |z_u - z_v|^2}.$$

Cut Problems

The goal of a cut problem is to partition the vertices into two (or more) sets so as to optimize a given objective function.



Maximum Cut



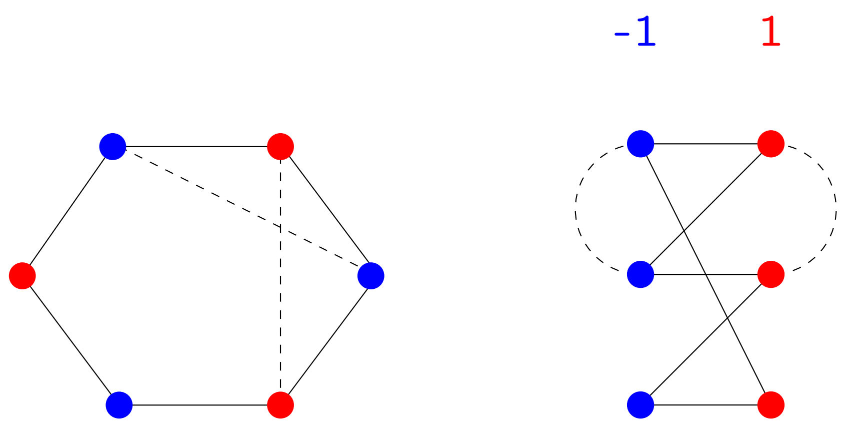
3-Coloring

Semidefinite Programming for Max Cut

Max Cut SDP [Goemans, Williamson 94]

$$\max \quad \sum_{ij \in A} \frac{1}{2}(1 - v_i \cdot v_j)$$

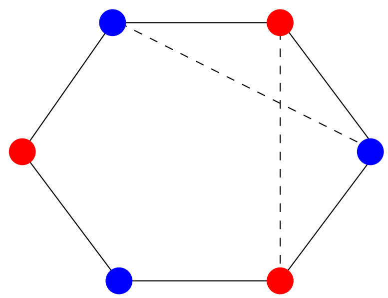
$$v_i \quad \in \quad \{-1, 1\} \quad \forall i \in V.$$

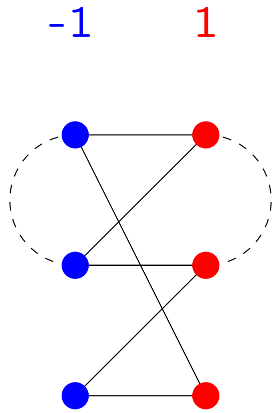


Semidefinite Programming for Max Cut

Max Cut SDP [Goemans, Williamson 94]

$$\begin{array}{rcl} \max & & \displaystyle\sum_{ij \in A} \frac{1}{2}(1 - v_i \cdot v_j) \\ v_i & \in & \mathrm{R}^n \quad \forall i \in V. \\ v_i \cdot v_i & = & 1. \end{array} \tag{1}$$

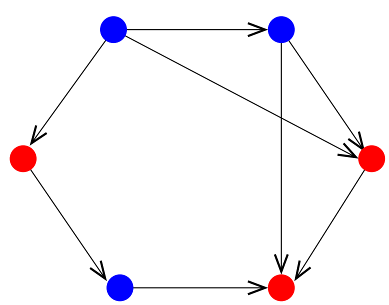


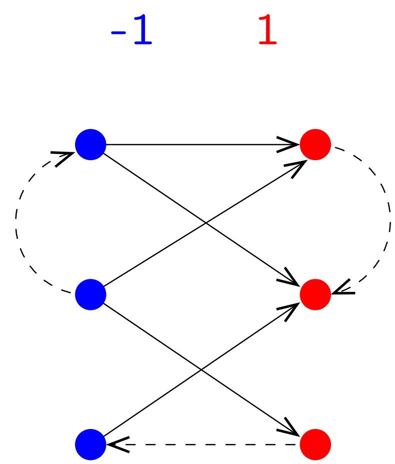


Maximum Directed Cut SDP[GW94]

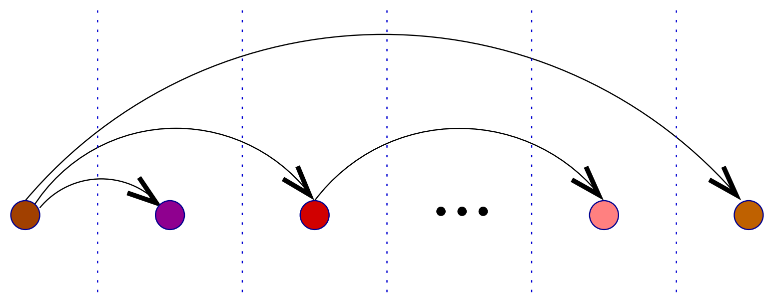
$$\max \quad \sum_{ij \in A} \frac{1}{4}(1 - v_i \cdot v_j + v_i \cdot v_0 - v_j \cdot v_0)$$

$$\begin{array}{rcll} v_i & \in & \{-1, 1\} & \forall i \in V. \\ v_0 & = & 1. & \end{array}$$





An Ordering is a Series of Cuts



A vertex ordering can be precisely described by $n - 1$ bipartitions of the vertices.

Ordering SDP

Our SDP is equivalent to a formulation using only unit vectors.

There are $n+1$ vectors for every vertex.

The following is an integral solution for a graph on four vertices in which vertex i is in position i in the ordering.

$$\begin{aligned}
\{v_1^0,\ v_1^1,\ v_1^2,\ v_1^3,\ v_1^4\} &= \{-1,\ 1,\ 1,\ 1,\ 1\},\\
\{v_2^0,\ v_2^1,\ v_2^2,\ v_2^3,\ v_2^4\} &= \{-1,-1,\ 1,\ 1,\ 1\},\\
\{v_3^0,\ v_3^1,\ v_3^2,\ v_3^3,\ v_3^4\} &= \{-1,-1,-1,\ 1,\ 1\},\\
\{v_4^0,\ v_4^1,\ v_4^2,\ v_4^3,\ v_4^4\} &= \{-1,-1,-1,-1,\ 1\}.
\end{aligned}$$

An SDP for Linear Ordering

$$\max \sum_{i,j\in V} \sum_{1\le h<\ell\le n} \frac{1}{4} w_{ij} (v_i^h - v_i^{h-1}) \cdot (v_j^\ell - v_j^{\ell-1})$$

$$\begin{array}{rcll}
(v_i^h - v_i^{h-1}) \cdot (v_j^\ell - v_j^{\ell-1}) & \geq & 0 & \forall i,j \in V, h, \ell \in [n] \\
v_i^{\frac{n}{2}} \cdot (\sum_{j=1}^{n} v_j^{\frac{n}{2}}) & = & 0 & \forall i \in V \\
v_i^h \cdot v_i^h & = & 1 & \forall i, h \in [n] \\
v_i^0 \cdot v_0 & = & -1 & \forall i \in V \\
v_i^n \cdot v_0 & = & 1 & \forall i \in V \\
v_i^h & \in & \{1, -1\} & \forall i, h \in [n].
\end{array}$$

Objective Function:

$$\max \sum_{i,j \in E} \sum_{1 \leq h < \ell \leq n} \frac{1}{4}(v_i^h - v_i^{h-1}) \cdot (v_j^\ell - v_j^{\ell-1})$$

An edge (i, j) only contributes 1 to the objective function when:

$$v_i^{h-1} = v_j^{\ell-1} = -1 \text{ and } v_i^h = v_j^\ell = 1.$$

Valid:

.	.	.	h-1	h	.	.	.	ℓ-1	ℓ	.	.	.
.	.	.	-1	1	.	.	.	1	1	.	.	.
.	.	.	-1	-1	.	.	.	-1	1	.	.	.

$$(v_i^h - v_i^{h-1}) \cdot (v_j^\ell - v_j^{\ell-1}) \geq 0$$

Invalid:

.	.	.	h-1	h	.	.	.	ℓ-1	ℓ	.	.	.
.	.	.	-1	-1	.	.	.	1	-1	.	.	.
.	.	.	-1	1	.	.	.	1	1	.	.	.

Difference Cuts

Problem of finding a cut that maximizes the difference between forward and backward edges is polytime solvable.

$$\max \sum_{ij \in A} (x_i - x_j), \quad 0 \leq x_i \leq 1.$$

Optimal solutions to this LP are integral.

If there is a cut with difference more than $\delta|E|$, then we can have a max acyclic subgraph with size more than $\delta|E| + \frac{1}{2}(1-\delta)|E| = (\frac{1+\delta}{2})|E|$.

Worst case is when digraph is Eulerian.

An Upper Bound on OPT

Assume digraph is Eulerian. All cuts have difference 0.

Consider the "cut value" associated with "middle vectors" $x_u = v_u^{n/2}$.

"Fractional" value that crosses the cut: $C := \sum_{(u,v)\in E} \frac{1 - x_u \cdot x_v}{2}$

$\text{Forward}_C + \text{Backwards}_C = C$ and $\text{Forward}_C = \text{Backwards}_C$.

$|E| - MAS_{SDP} = FAS_{SDP} \geq \text{Backwards}_C \geq \frac{C}{2}$

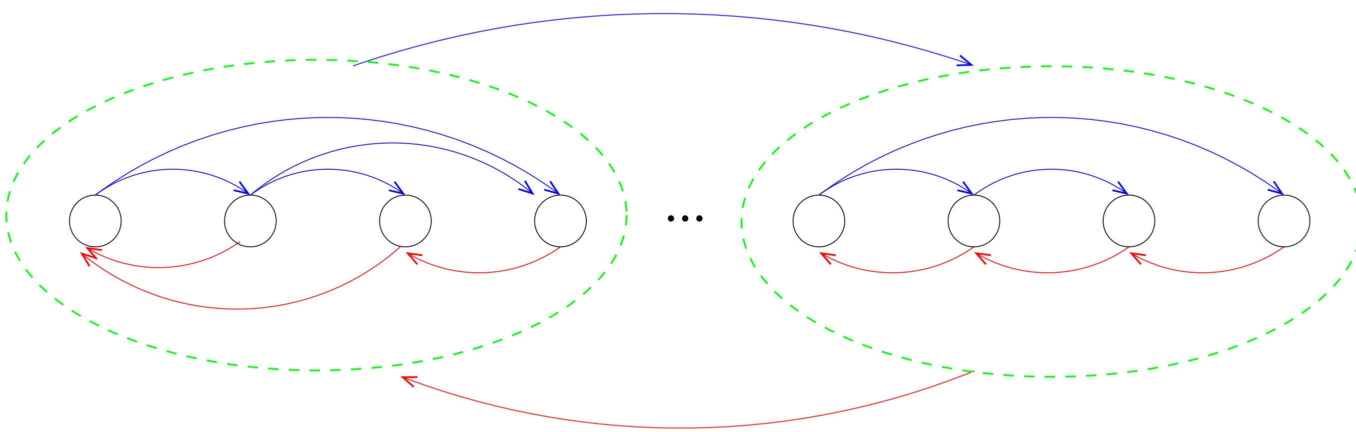
$MAS_{SDP} \leq |E| - \frac{C}{2} \leq |E| - \frac{\lambda |E|}{4}$

Magic step: $\lambda = \min \frac{\mathbb{E} \sum_{uv \in E} |z_u - z_v|^2}{\mathbb{E} \sum_{u,v \in V} |z_u - z_v|^2} \Rightarrow C \geq \frac{\lambda |E|}{2}$.

Directed Cuts

For each of the $n-1$ cuts defined by vectors $\{v_i^k\}$ for $n-1$ possible values of k, the forward value is:

$$\frac{1}{4}\sum_{ij\in A}\sum_{h\leq k,\ \ell>k}(v_i^h - v_i^{h-1})\cdot(v_j^\ell - v_j^{\ell-1})$$



The backward value is:

$$\frac{1}{4}\sum_{ij\in A}\sum_{h>k,\ \ell\leq k}(v_i^h - v_i^{h-1})\cdot(v_j^\ell - v_j^{\ell-1})$$

An Upper Bound on OPT

If there is a cut with difference $\delta|E|$, we get $\frac{(1+\delta)|E|}{2}$

Consider the "cut value" associated with "middle vectors" $x_u = v_i^{n/2}$.

"Fractional" value that crosses the cut: $C := \sum_{(u,v)\in E} \frac{1-x_u \cdot x_v}{2}$

$|E| - MAS_{SDP} = FAS_{SDP} \geq \frac{C-\delta|E|}{2}$

$MAS_{SDP} \leq |E| - \frac{C}{2} + \frac{\delta|E|}{2} \leq |E| - \frac{\lambda|E|}{4} + \frac{\delta|E|}{2}$

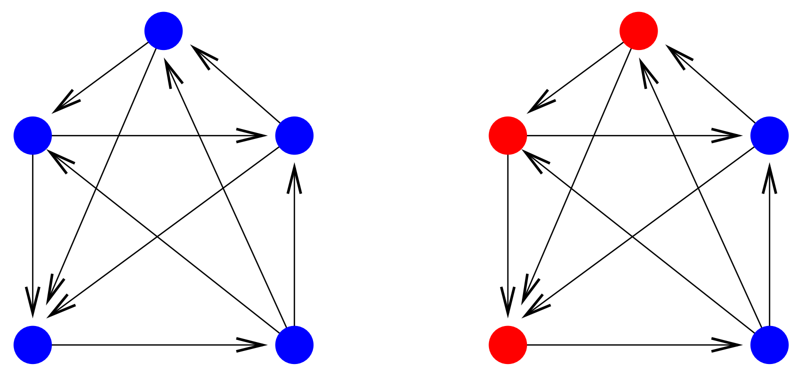
$\lambda = \min \frac{\mathbb{E}\sum_{uv\in E}|z_u - z_v|}{\mathbb{E}\sum_{u,v\in V}|z_u - z_v|} \Rightarrow C \geq \frac{\lambda|E|}{2}$.

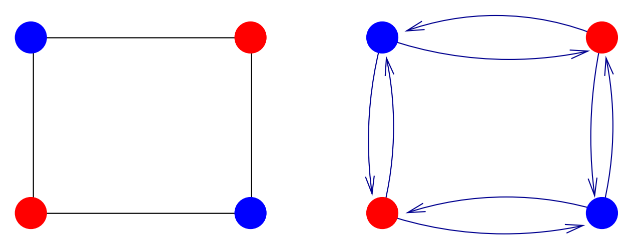
Part II: Induced Acyclic Subgraphs of Directed Graphs

Joint work with Felix Klingelhoefer
(who made many pictures/slides).

Directed Graphs and (Induced) Acyclic Sets

An acyclic set is a set of vertices whose induced subgraph contains no cycles.

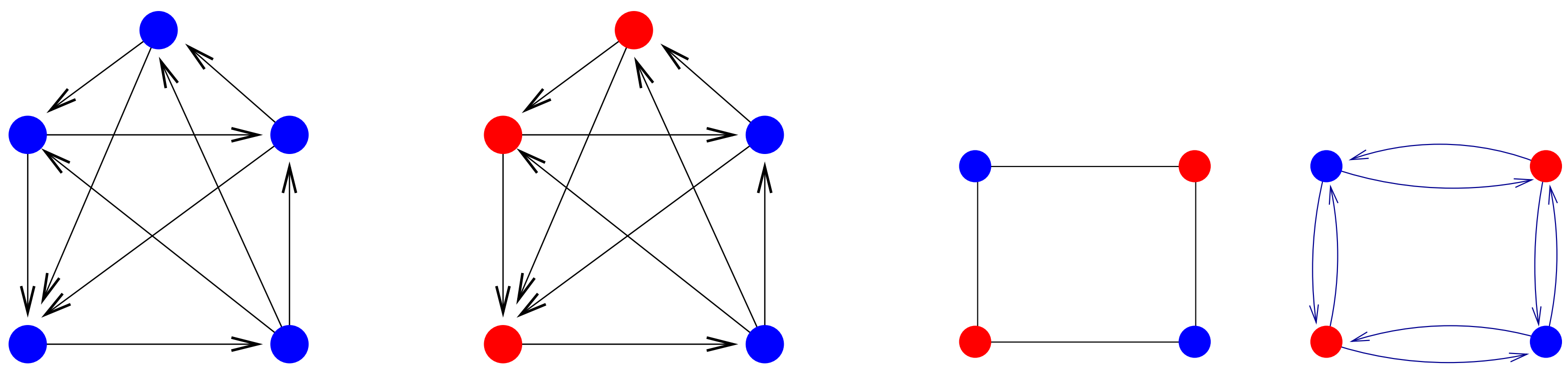




The problem of finding a large stable set in an undirected graph can be reduced to finding a large (induced) acyclic set by replacing each edge with a directed 2-cycle.

Directed Graphs and Dicoloring

A dicoloring of a directed graph is a partition of its vertices into acyclic sets.



The problem of coloring an undirected graph can be reduced to dicoloring by replacing each edge with a directed 2-cycle.

For oriented graphs without 2-cycles, 2-dicoloring is NP-hard.

If we can dicolor a digraph with c colors, then there exists an acyclic set of size at least n/c.

Acyclic Sets in Oriented Digraphs

We can always find an acyclic set containing $O(\log n)$ vertices.

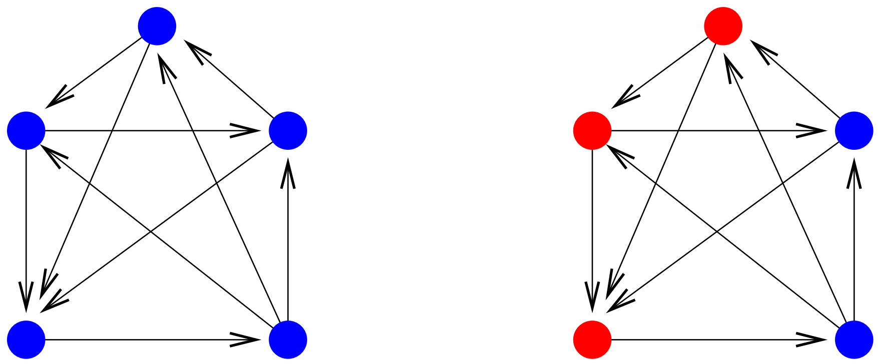
It is hard to find induced acyclic set of size at least $n^{1/2+\epsilon}$ in an oriented digraph containing such a set of size $n^{1-\epsilon}$.

What about promise of being 2-colorable? (i.e., There exists an acyclic set of size at least $n/2$.)

- NP-hard.
- UG-hard to dicolor with any constant number of colors [Svensson 2013].
- How to get any $o(n)$ colors?

Acyclic Sets in Tournaments

Tournament is an orientation of a complete graph.



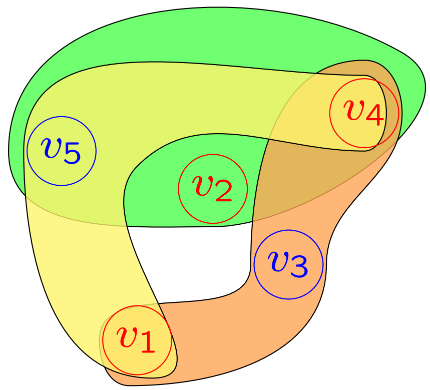
Also very hard in general.

Recall that if we can color with c colors, then find acyclic set of size n/c

Currently, we don't know how to find acyclic sets in tournaments without going through coloring ...

Tournaments and Coloring

Coloring tournaments is a special case of coloring 3-uniform hypergraphs, where every directed triangle is represented by a hyperedge.



Erdős Hajnal Conjecture

The following is a famous conjecture by [Erdős and Hajnal 1989].

Conjecture: For every graph H, there exists a positive constant $\epsilon(H)$ such that every H-free graph on n vertices contains a clique or an independent set of size $\Omega(n^{\epsilon(H)})$.

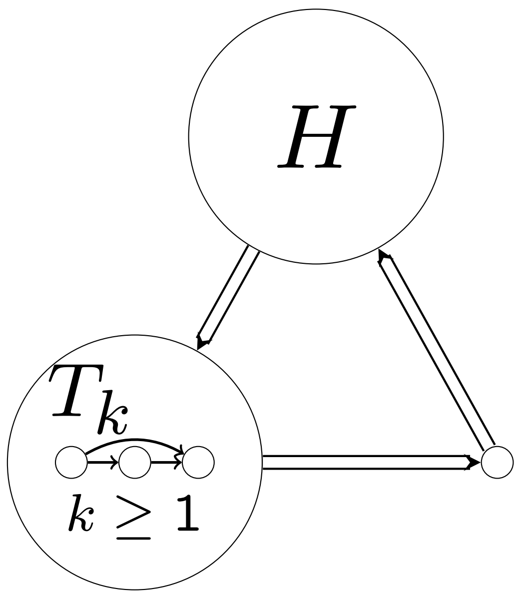
[Alon, Pach, Solymosi 2001] proved that it has an equivalent formulation in terms of tournaments.

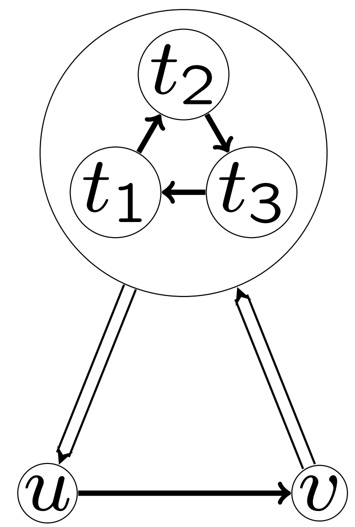
Conjecture: For every tournament T, there exists a positive constant $\epsilon(H)$ such that every H-free tournament on n vertices contains a transitive set of size $\Omega(n^{\epsilon(H)})$.

Heroes in tournaments

The class of tournaments H such that any H-free tournament has constant chromatic number (called **heroes**) has been completely defined [Berger, Choromanski, Chudnovsky, Fox, Loebl, Scott, Seymour, Thomassé 2013].

Heroes are exactly the tournaments with EH exponent $\epsilon(H) = 1$.





Proved by showing H-free tournaments have “constant” chromatic number, where constant depends on H. Not (completely) constructive.

State of the art for tournament coloring

Similar to 3-colorable graphs, we can ask what is the minimum number of colors required to color a 2-colorable tournament?

Tournament Type	Lower Bound	Upper Bound
2-Colorable tournaments	2 [CHZ 2007]	$\tilde{O}(n^{\frac{1}{5}})$ [KNS 2001]
3-Colorable tournaments	3 [FGSY 2019]	?
k-Colorable tournaments, $k \geq 2$	k [FGSY 2019]	?
General tournaments	?	$n/\log n$

Best known polynomial time inapproximability results and approximation algorithms for various tournament coloring problems.

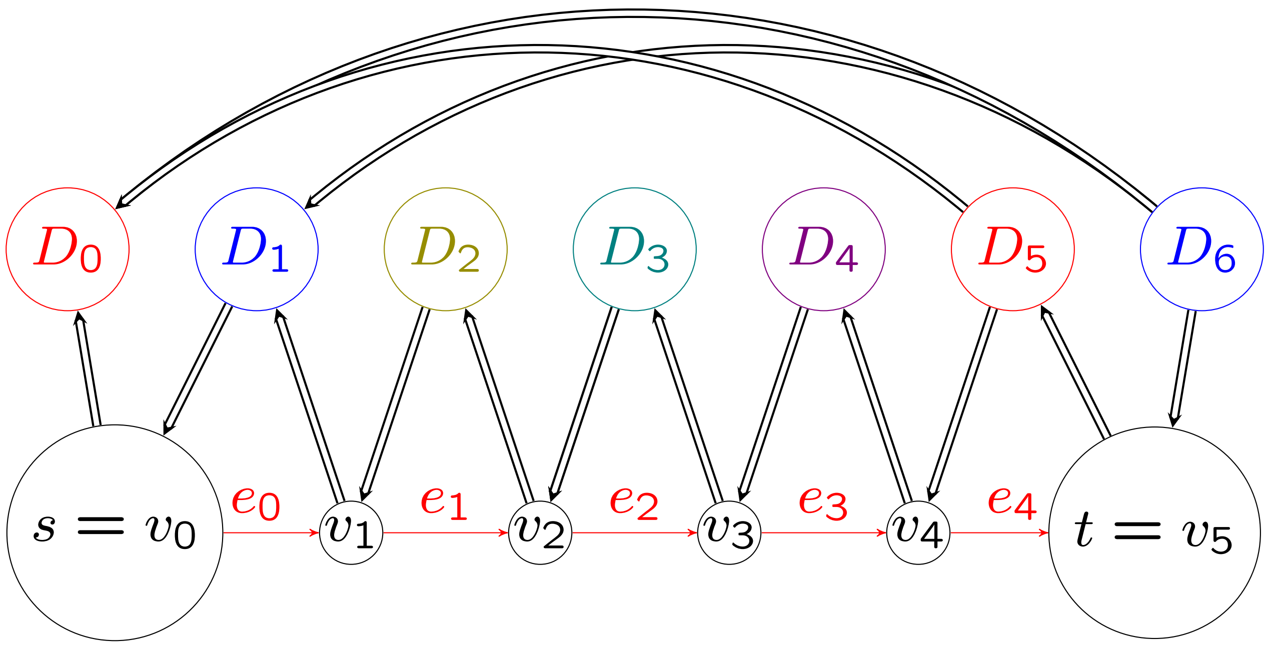
We can color a 2-colorable tournament with $O(n^{1/5})$ colors. (Works via iteratively finding induced acyclic sets.)

Efficient coloring of 2-colorable tournaments

We use $\vec{\chi}_{\text{eff}}(T)$ to denote the number of colors with which a tournament T can be colored efficiently.

- **Problem:** How many colors do we need to efficiently color a 2-colorable tournament ?
  - At least 3 since 2-coloring 2-colorable tournaments is NP-hard.
  - At most $O(n^{\frac{1}{5}})$.
- **Theorem:** Every 2-colorable tournament can be efficiently colored with **10 colors**.

Path Decomposition



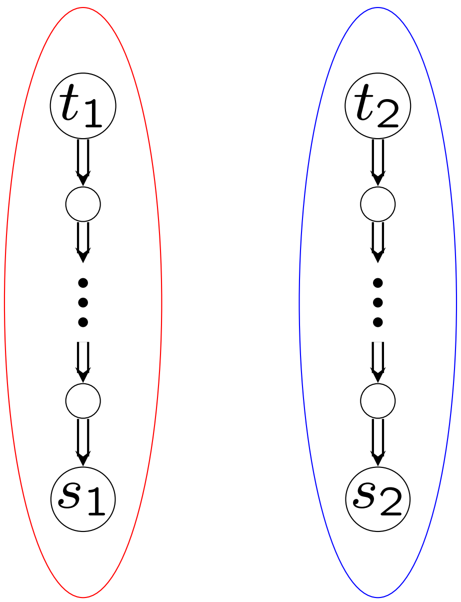
- Every vertex is in some set D_i.
- The red arcs form a shortest path from $s = v_0$ to $t = v_k$.
- There can be no long (≥ 5) forward arc between the D_i's.
- Every D_i is in the neighborhood of a red arc, except D_0 and D_{k+1}.
- $\vec{\chi}_{\text{eff}}(T) \leq 5 \cdot \max_i \vec{\chi}_{\text{eff}}(D_i)$.

Decomposition Lemma

- A c-good pair is a pair of vertices (s, t) such that $\vec{\chi}_{\text{eff}}(N^+(s) \cup N^-(t)) \leq c$.
- **Lemma:** If T has a c-good pair and if $\vec{\chi}_{\text{eff}}(N(e)) \leq c$ for each arc e, then $\vec{\chi}_{\text{eff}}(T) \leq 5c$.
- The c-good pair bounds D_0 and D_{k+1} and the condition $\vec{\chi}_{\text{eff}}(N(e)) \leq c$ bounds all the remaining D_i.

2-Colorable tournaments: Finding a good pair

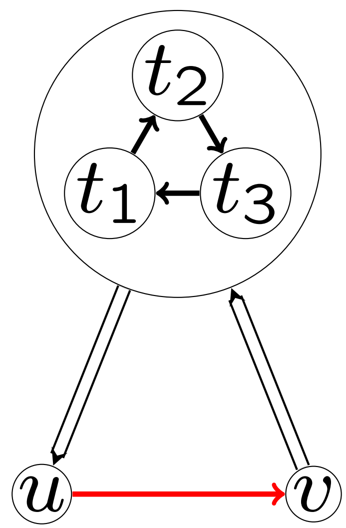
- **Lemma:** Every 2-colorable tournament has a 1-good pair.



- Every 2-colorable tournament has a partition into two transitive sets.
- t_1 can only have blue in-neighbors.
- s_1 can only have blue out-neighbors.
- $N^-(t_1) \cup N^+(s_1)$ is a subset of the blue vertices, thus acyclic.
- (s_1, t_1) and (s_2, t_2) both form 1-good pairs.

2-colorable tournaments: Bounding the edge neighborhood

Lemma: Every 2-colorable tournament can be efficiently partitioned into two 2-colorable light tournaments.

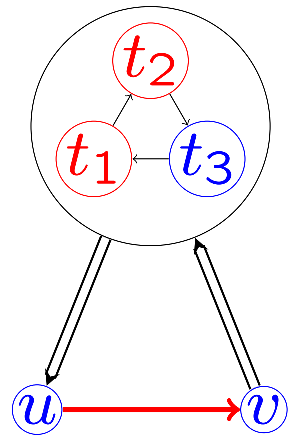


We say an edge is **heavy** if its neighborhood contains a triangle. The set of heavy edges must form a bipartite graph.

We can then 2-color that graph such that no color class contains a heavy edge.

2-colorable tournaments: Bounding the edge neighborhood

Lemma: Every 2-colorable tournament can be efficiently partitioned into two 2-colorable light tournaments.

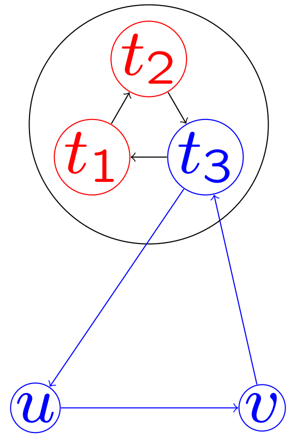


We say an edge is **heavy** if its neighborhood contains a triangle. The set of heavy edges must form a bipartite graph.

We can then 2-color that graph such that no color class contains a heavy edge.

2-colorable tournaments: Bounding the edge neighborhood

Lemma: Every 2-colorable tournament can be efficiently partitioned into two 2-colorable light tournaments.

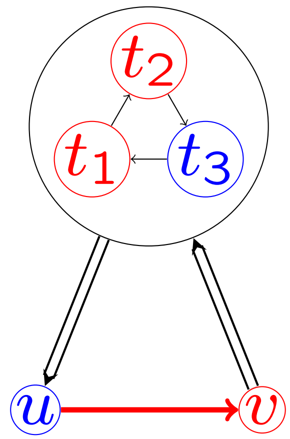


We say an edge is **heavy** if its neighborhood contains a triangle. The set of heavy edges must form a bipartite graph.

We can then 2-color that graph such that no color class contains a heavy edge.

2-colorable tournaments: Bounding the edge neighborhood

Lemma: Every 2-colorable tournament can be efficiently partitioned into two 2-colorable light tournaments.



We say an edge is **heavy** if its neighborhood contains a triangle. The set of heavy edges must form a bipartite graph.

We can then 2-color that graph such that no color class contains a heavy edge.

Coloring 2-colorable tournaments

We partition the 2-colorable tournament into two 2-colorable light tournaments, which can each be colored with 5 colors using the decomposition lemma, resulting in the following theorem:

Theorem: Every 2-colorable tournament can be efficiently colored with **10 colors**.

Coloring 3-Colorable Tournaments

Theorem: If we can efficiently color a 3-colorable graph G with k colors, then we can efficiently color a 3-colorable tournament with $50k$ colors.

Which implies the following bound using the current best bound for coloring 3-colorable graphs [Kawarabayashi, Thorup 2017].

Corollary: Let T be a 3-colorable tournament on n vertices. Then, $\vec{\chi}_{\text{eff}}(T) \leq O(n^{.19996})$.

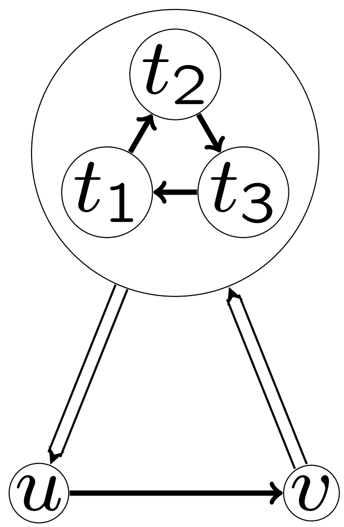
Other direction holds, too.

Problems of coloring 3-colorable graphs and 3-colorable tournaments with constantly many colors are essentially equivalent.

Coloring light tournaments

Recall **light tournaments** are tournaments with no heavy edge.

Problem: How many colors do we need to (efficiently) color light tournaments?



A heavy edge.

(A 2-colorable light tournament can be colored with 5 colors, but now we are interested in general light tournaments.)

Coloring light tournaments

It was known that light tournaments can be colored with constantly many colors, but proof was not constructive [Berger et al. 2013].

Proof can be made algorithmic with around 35 colors.

- Light tournaments have bounded chromatic number because hero-free (i.e., $\Delta(1, 1, C_3)$-free).
- [Berger et al.] use "jewel chain" to color.

To use the decomposition lemma, we need to find a good pair and bound the edge neighborhood.

- All edge neighborhoods are acyclic in light tournaments.
- We use jewel chain to find a good pair.

Any light tournament can be colored with at most 8 colors.

There is a light tournament that requires 3 colors. What is the truth?

Complexity of Coloring Tournaments

- Best known result is that it is NP-**hard** to color k-colorable tournaments with k colors, for $k \geq 2$. [Fox, Gishboliner, Shapira, Yuster 2019].
- We show it is NP-**hard** to color k-colorable tournaments with $2k-1$ colors.
- We show it is NP-**hard** to color 2-colorable tournaments with 3 colors.
- If it is NP-**hard** to color a 2-colorable tournament with 4 colors, then there is no poly-time algorithm to 2-color a 2-colorable light tournament.

Arc-bounded Tournaments

Light tournaments have $\chi(N(e)) \leq 1$ and have $\chi(T) \leq O(1)$.

[Haratyunyan, Le, Thomassé, Wu 19] showed if $\chi(N^+(u)) \leq t$, then $\chi(T) \leq f(t)$.

Theorem: if $\chi(N(e)) \leq t$, then $\chi(T) \leq g(t)$.

Question: If T or G has high chromatic number, what subgraph are forbidden?

Conjecture: If T has high chromatic number, then it has two sets A and B, each with high chromatic number, and all arcs from A to B [Scott, Seymour, Tung 2023].

Conjecture: If G has high chromatic number and clique size at most t, then it has two anti-complete sets, each with high chromatic number [Erdős, El-Zahar 1985].

[Scott, Seymour, Tung 2023] showed first implies second.

Theorem: Second conjecture implies first; Conjectures are equivalent.

Open problems

Find a transitive subset of size greater than $n/10$ in 2-colorable tournaments.

2-color 2-colorable light tournaments?

Get better algorithmic or complexity results for digraphs.