Disjoint paths in tournaments

Paul Seymour, Princeton University

joint with Maria Chudnovsky, Alexandra Fradkin and Alex Scott
The $k$ disjoint paths problem in digraphs

Given $k$ pairs of vertices $(s_1, t_1), \ldots, (s_k, t_k)$ of a digraph $G$, decide whether there are disjoint directed paths in $G$ where the $i$th path is from $s_i$ to $t_i$ ($1 \leq i \leq k$).
The $k$ disjoint paths problem in digraphs

Given $k$ pairs of vertices $(s_1, t_1), \ldots, (s_k, t_k)$ of a digraph $G$, decide whether there are disjoint directed paths in $G$ where the $i$th path is from $s_i$ to $t_i$ ($1 \leq i \leq k$).

Theorem (Fortune, Hopcroft, Wyllie, 1980)

This is NP-complete, even for $k = 2$. 
Theorem

If the input digraph is a tournament:

- the two edge-disjoint paths problem is solvable in polynomial time (Bang-Jensen, 1991)
- the two vertex-disjoint paths problem is solvable in polynomial time (Bang-Jensen and Thomassen, 1992).
Theorem

If the input digraph is a tournament:
- the two edge-disjoint paths problem is solvable in polynomial time (Bang-Jensen, 1991)
- the two vertex-disjoint paths problem is solvable in polynomial time (Bang-Jensen and Thomassen, 1992).

Theorem (Our results)

For all fixed k, if the input digraph is a tournament:
- the k edge-disjoint paths problem is solvable in polynomial time (Fradkin, S., 2009)
- the k vertex-disjoint paths problem is solvable in polynomial time (Chudnovsky, Scott, S., 2010).
Vertex-disjoint undirected case

Two steps:

- If tree-width is at least $f(k)$, delete some vertex that does not change the problem.
Vertex-disjoint undirected case

Two steps:

- If tree-width is at least $f(k)$, delete some vertex that does not change the problem.
- If tree-width is less than $f(k)$, solve directly with dynamic programming.
Edge-disjoint case
Edge-disjoint case

Theorem

*If there is a big widget of either kind, then identifying u, v does not change whether the paths exist.*
Theorem

If there is a big widget of either kind, then identifying $u, v$ does not change whether the paths exist.

What if there is no big widget? Need to allow parallel edges.
Edge-disjoint case

**Theorem**

*Tournaments without the first widget of size \( t \) can be ordered with back-degree at most \( f(t) \).*
Edge-disjoint case

**Theorem**

*Tournaments without the first widget of size $t$ can be ordered with back-degree at most $f(t)$.*

**Theorem**

*Tournaments without either kind of widget of size $t$ can be ordered with cutwidth at most $f(t)$.*
Edge-disjoint case

Algorithm for $k$ edge-disjoint paths problem in a tournament:

- If there is a big widget (size at least $f(k)$), identify its ends and repeat;
Edge-disjoint case

Algorithm for $k$ edge-disjoint paths problem in a tournament:

- If there is a big widget (size at least $f(k)$), identify its ends and repeat;
- If there is no big widget, thin out parallel edges, get bounded cutwidth, use dynamic programming.
Edge-disjoint case

Algorithm for $k$ edge-disjoint paths problem in a tournament:

- If there is a big widget (size at least $f(k)$), identify its ends and repeat;
- If there is no big widget, thin out parallel edges, get bounded cutwidth, use dynamic programming.

- Fixed parameter tractable: running time $C_k |V(G)|^5$
- Extends to digraphs with bounded stability number (not fpt any more).
Theorem (Strengthened version)

For all $k$, there is a polynomial-time algorithm as follows:

- **Input:** Tournament $G$, pairs $(s_1, t_1), \ldots, (s_k, t_k)$, and integers $n_1, \ldots, n_k$
- **Output:** Decides whether there exist $k$ vertex-disjoint directed paths joining the pairs, where the $s_i - t_i$ path has length at most $n_i$. 
Vertex-disjoint case

**Linkage:** $k$ vertex-disjoint directed paths joining the pairs.

**MAIN RESULT**

- **Input:** Tournament $G$, and $(s_1, t_1), \ldots, (s_k, t_k)$.
- **Output:** Decides if there is a linkage (in polynomial time for fixed $k$).
Vertex-disjoint case

Idea: Construct an auxiliary digraph $H$, with two special vertices $S_0$, $T_0$; arrange that

- If there is a linkage in $G$ then there is an $S_0$, $T_0$-path in $H$
- If there is an $S_0$, $T_0$-path in $H$ then there is a linkage in $G$
Vertex-disjoint case

**Idea:** Construct an auxiliary digraph $H$, with two special vertices $S_0, T_0$; arrange that

- If there is a linkage in $G$ then there is an $S_0, T_0$-path in $H$
- If there is an $S_0, T_0$-path in $H$ then there is a linkage in $G$
- $H$ can be computed in polynomial time.
Vertex-disjoint case

Idea: Construct an auxiliary digraph $H$, with two special vertices $S_0, T_0$; arrange that

- If there is a linkage in $G$ then there is an $S_0, T_0$-path in $H$
- If there is an $S_0, T_0$-path in $H$ then there is a linkage in $G$
- $H$ can be computed in polynomial time.

$(v_1, \ldots, v_k)$ to $(v'_1, \ldots, v'_k)$ is a **spiderstep** if for some $i$, $v'_i$ is an out-neighbour of $v_i$, and $v'_j = v_j$ for $j \neq i$. 
Vertex-disjoint case

Idea: Construct an auxiliary digraph $H$, with two special vertices $S_0, T_0$; arrange that

- If there is a linkage in $G$ then there is an $S_0, T_0$-path in $H$
- If there is an $S_0, T_0$-path in $H$ then there is a linkage in $G$
- $H$ can be computed in polynomial time.

$(v_1, \ldots, v_k)$ to $(v'_1, \ldots, v'_k)$ is a spiderstep if for some $i$, $v'_i$ is an out-neighbour of $v_i$, and $v'_j = v_j$ for $j \neq i$.

First try: $V(H)$ is all $k$-tuples of distinct vertices of $G$; $H$-adjacency is spidersteps.
Vertex-disjoint case

Idea: Construct an auxiliary digraph $H$, with two special vertices $S_0$, $T_0$; arrange that

- If there is a linkage in $G$ then there is an $S_0$, $T_0$-path in $H$
- If there is an $S_0$, $T_0$-path in $H$ then there is a linkage in $G$
- $H$ can be computed in polynomial time.

$(v_1, \ldots, v_k)$ to $(v'_1, \ldots, v'_k)$ is a spiderstep if for some $i$, $v'_i$ is an out-neighbour of $v_i$, and $v'_j = v_j$ for $j \neq i$.

First try: $V(H)$ is all $k$-tuples of distinct vertices of $G$; $H$-adjacency is spidersteps.

- $G$-linkage $\Rightarrow$ $H$-path YES (too easy)
Vertex-disjoint case

Idea: Construct an auxiliary digraph $H$, with two special vertices $S_0, T_0$; arrange that

- If there is a linkage in $G$ then there is an $S_0, T_0$-path in $H$
- If there is an $S_0, T_0$-path in $H$ then there is a linkage in $G$
- $H$ can be computed in polynomial time.

$(v_1, \ldots, v_k)$ to $(v'_1, \ldots, v'_k)$ is a spiderstep if for some $i$, $v'_i$ is an out-neighbor of $v_i$, and $v'_j = v_j$ for $j \neq i$.

First try: $V(H)$ is all $k$-tuples of distinct vertices of $G$; $H$-adjacency is spidersteps.

- $G$-linkage $\Rightarrow$ $H$-path YES (too easy)
- $H$-path $\Rightarrow$ $G$-linkage ??
- $H$ computible YES
Vertex-disjoint case

Second try: $V(H)$ is all $(k + 2)$-tuples $(v_1, \ldots, v_k, A, B)$, where $v_1, \ldots, v_k \in V(G)$ are distinct and $A, B$ are disjoint subsets of the other vertices of $G$.

$(v_1, \ldots, v_k, A, B)$ is $H$-adjacent to $(v'_1, \ldots, v'_k, A', B')$ if:

- $(v_1, \ldots, v_k)$ to $(v'_1, \ldots, v'_k)$ is a spiderstep, changing $v_i$
Vertex-disjoint case

Second try: $V(H)$ is all $(k + 2)$-tuples $(v_1, \ldots, v_k, A, B)$, where $v_1, \ldots, v_k \in V(G)$ are distinct and $A, B$ are disjoint subsets of the other vertices of $G$.

$(v_1, \ldots, v_k, A, B)$ is $H$-adjacent to $(v'_1, \ldots, v'_k, A', B')$ if:

- $(v_1, \ldots, v_k)$ to $(v'_1, \ldots, v'_k)$ is a spiderstep, changing $v_i$
- $v'_i \in A$, and $v_i \in B'$;
Vertex-disjoint case

Second try: \( V(H) \) is all \((k + 2)\)-tuples \((v_1, \ldots, v_k, A, B)\), where \(v_1, \ldots, v_k \in V(G)\) are distinct and \(A, B\) are disjoint subsets of the other vertices of \(G\).

\((v_1, \ldots, v_k, A, B)\) is \(H\)-adjacent to \((v'_1, \ldots, v'_k, A', B')\) if:

- \((v_1, \ldots, v_k)\) to \((v'_1, \ldots, v'_k)\) is a spiderstep, changing \(v_i\)
- \(v'_i \in A\), and \(v_i \in B'\);
- \(A' \subseteq A\) and \(B \subseteq B'\).
Vertex-disjoint case

Second try: $V(H)$ is all $(k + 2)$-tuples $(v_1, \ldots, v_k, A, B)$, where $v_1, \ldots, v_k \in V(G)$ are distinct and $A, B$ are disjoint subsets of the other vertices of $G$.

$(v_1, \ldots, v_k, A, B)$ is $H$-adjacent to $(v'_1, \ldots, v'_k, A', B')$ if:

- $(v_1, \ldots, v_k)$ to $(v'_1, \ldots, v'_k)$ is a spiderstep, changing $v_i$
- $v'_i \in A$, and $v_i \in B'$;
- $A' \subseteq A$ and $B \subseteq B'$.

- $G$-linkage $\Rightarrow$ $H$-path YES (too easy)
- $H$-path $\Rightarrow$ $G$-linkage YES
- $H$ computible NO
Vertex-disjoint case

Ski: directed path $S$ with $f(k)$ vertices; has first vertex $r(S)$ (rear) and last vertex $t(S)$ (tip)

Ski set: $k$-tuple $(S_1, \ldots, S_k)$ of vertex-disjoint skis.
Vertex-disjoint case

**Ski:** directed path $S$ with $f(k)$ vertices; has first vertex $r(S)$ (rear) and last vertex $t(S)$ (tip)

**Ski set:** $k$-tuple $(S_1, \ldots, S_k)$ of vertex-disjoint skis.

$(S_1, \ldots, S_k)$ to $(S'_1, \ldots, S'_k)$ is a **skiing spider step** if

- for some $i$, $S_i \setminus r(S_i) = S'_i \setminus t(S'_i)$
- $S'_j = S_j$ for $j \neq i$. 


Vertex-disjoint case

**Ski**: directed path $S$ with $f(k)$ vertices; has first vertex $r(S)$ (rear) and last vertex $t(S)$ (tip)

**Ski set**: $k$-tuple $(S_1, \ldots, S_k)$ of vertex-disjoint skis.

$(S_1, \ldots, S_k)$ to $(S'_1, \ldots, S'_k)$ is a **skiing spider step** if

- for some $i$, $S_i \setminus r(S_i) = S'_i \setminus t(S'_i)$
- $S'_j = S_j$ for $j \neq i$.

$P(S_1, \ldots, S_k)$: set of all other vertices complete to some $P_i \setminus t(P_i)$;
$Q(S_1, \ldots, S_k)$: set of all other vertices complete from some $P_i \setminus r(P_i)$. 
Vertex-disjoint case

Skiing spider digraph: vertex set all \((k + 2)\)-tuples \((S_1, \ldots, S_k, A, B)\), where \((S_1, \ldots, S_k)\) is a ski set and \(A, B\) are disjoint subsets of the other vertices of \(G\), such that

- \(A \subseteq P(S_1, \ldots, S_k)\), and \(B \subseteq Q(S_1, \ldots, S_k)\)
- \(A \cup B = P(S_1, \ldots, S_k) \cup Q(S_1, \ldots, S_k)\).
Vertex-disjoint case

Skiing spider digraph: vertex set all \((k + 2)\)-tuples \((S_1, \ldots, S_k, A, B)\), where \((S_1, \ldots, S_k)\) is a ski set and \(A, B\) are disjoint subsets of the other vertices of \(G\), such that

- \(A \subseteq P(S_1, \ldots, S_k)\), and \(B \subseteq Q(S_1, \ldots, S_k)\)
- \(A \cup B = P(S_1, \ldots, S_k) \cup Q(S_1, \ldots, S_k)\).
Vertex-disjoint case

Skiing spider digraph: vertex set all $(k + 2)$-tuples $(S_1, \ldots, S_k, A, B)$, where $(S_1, \ldots, S_k)$ is a ski set and $A, B$ are disjoint subsets of the other vertices of $G$, such that

- $A \subseteq P(S_1, \ldots, S_k)$, and $B \subseteq Q(S_1, \ldots, S_k)$
- $A \cup B = P(S_1, \ldots, S_k) \cup Q(S_1, \ldots, S_k)$.

$(S_1, \ldots, S_k, A, B)$ is adjacent to $(S'_1, \ldots, S'_k, A', B')$ if

- $(S_1, \ldots, S_k)$ to $(S'_1, \ldots, S'_k)$ is a skiing spider step, changing $S_i$
- $t(S'_i) \in A$ and $r(S_i) \in B'$
- $A' \subseteq A$ and $B' \subseteq B'$. 
Vertex-disjoint case

Skiing spider digraph: vertex set all \((k + 2)\)-tuples \((S_1, \ldots, S_k, A, B)\), where \((S_1, \ldots, S_k)\) is a ski set and \(A, B\) are disjoint subsets of the other vertices of \(G\), such that

- \(A \subseteq P(S_1, \ldots, S_k)\), and \(B \subseteq Q(S_1, \ldots, S_k)\)
- \(A \cup B = P(S_1, \ldots, S_k) \cup Q(S_1, \ldots, S_k)\).

\((S_1, \ldots, S_k, A, B)\) is adjacent to \((S'_1, \ldots, S'_k, A', B')\) if

- \((S_1, \ldots, S_k)\) to \((S'_1, \ldots, S'_k)\) is a skiing spider step, changing \(S_i\)
- \(t(S'_i) \in A\) and \(r(S_i) \in B'\)
- \(A' \subseteq A\) and \(B \subseteq B'\).

- \(G\)-linkage \(\Rightarrow\) \(H\)-path YES (too easy)
- \(H\)-path \(\Rightarrow\) \(G\)-linkage YES
- \(H\) computible ???
Vertex-disjoint case

\[ |P(S_1, \ldots, S_k) \cap Q(S_1, \ldots, S_k)| \text{ is the confusion of } (S_1, \ldots, S_k). \]
Vertex-disjoint case

\[ |P(S_1, \ldots, S_k) \cap Q(S_1, \ldots, S_k)| \] is the confusion of \((S_1, \ldots, S_k)\).

Let \(H\) be a subdigraph of the skiing spider digraph; just those \((k + 2)\)-tuples with ski sets of bounded confusion.
Vertex-disjoint case

\[ |P(S_1, \ldots, S_k) \cap Q(S_1, \ldots, S_k)| \] is the confusion of \((S_1, \ldots, S_k)\).

Let \(H\) be a subdigraph of the skiing spider digraph; just those \((k + 2)\)-tuples with ski sets of bounded confusion.

- \(G\)-linkage \(\Rightarrow\) \(H\)-path \(\text{??}\)
- \(H\)-path \(\Rightarrow\) \(G\)-linkage \(\text{YES}\)
- \(H\) computible \(\text{YES}\)
Vertex-disjoint case

\[ |P(S_1, \ldots, S_k) \cap Q(S_1, \ldots, S_k)| \] is the confusion of \((S_1, \ldots, S_k)\).

Let \(H\) be a subdigraph of the skiing spider digraph; just those \((k + 2)\)-tuples with ski sets of bounded confusion.

- \(G\)-linkage \(\Rightarrow\) \(H\)-path ??
- \(H\)-path \(\Rightarrow\) \(G\)-linkage YES
- \(H\) computible YES

Need a theorem: if a linkage exists, a spider can ski along it with bounded confusion at each step.
**Vertex-disjoint case**

**Theorem (Key fact)**

Let $P_1, \ldots, P_k$ a linkage with union of minimum size. There is a numbering $v_1, \ldots, v_n$ of $V(G)$, increasing along each $P_i$, such that there is no $(k + 2)$-edge planar matching as in the figure.
Vertex-disjoint case

\[ s_1 \quad \cdots \quad s_k \]

\[ v_1 \cdots v_i \]

\[ t_1 \quad \cdots \quad t_k \]
Vertex-disjoint case
Vertex-disjoint case
Vertex-disjoint case

This proves:

**Theorem**

- **Input:** Tournament $G$, and $(s_1, t_1), \ldots, (s_k, t_k)$.
- **Output:** Decides if there is a linkage, in polynomial time (for fixed $k$).
### Theorem

- **Input:** Tournament $G$, and $(s_1, t_1), \ldots, (s_k, t_k)$.
- **Output:** Decides if there is a linkage, in polynomial time (for fixed $k$).

More generally: digraph $G$ is *$d$-path-dominant* if every vertex has an in- or out-neighbour in every $d$-vertex directed path.

### Theorem

- **Input:** $d$-path-dominant digraph $G$, and $(s_1, t_1), \ldots, (s_k, t_k)$.
- **Output:** Decides if there is a linkage.
- **Running time:** $O(n^t)$ where $t = 6k^2d(k + d) + 13k$