Robust Optimization in the Presence of Uncertainty:
Counting approximately-shortest paths in directed acyclic graphs

Matúš Mihalák, Rastislav Šrámek, Peter Widmayer

ETH Zürich

pre-WALCOM-school, Chennai, Feb. 11, 2014
What?
Given $s$ and $t$:

- Shortest path
Given $s$ and $t$:  
- Shortest path  
- All shortest paths
Given $s$ and $t$:

- Shortest path
- All shortest paths
- $k$ shortest paths
Given $s$ and $t$:

- Shortest path
- All shortest paths
- $k$ shortest paths
- All paths
Given $s$ and $t$:

- Shortest path
- All shortest paths
- $k$ shortest paths
- All paths
- Count paths no longer than $L$. 

$s$-t paths
Problem (1)

*Given a weighted directed acyclic graph $G$, two vertices $s$ and $t$, and a threshold length $L$, how many $s$-$t$ paths no longer than $L$ are there in $G$?*
Count short paths in a DAG

Problem (1)

*Given a weighted directed acyclic graph* $G$, two vertices $s$ and $t$, and a threshold length $L$, how many $s$-$t$ paths no longer than $L$ are there in $G$?
Problem (1)

Given a weighted directed acyclic graph $G$, two vertices $s$ and $t$, and a threshold length $L$, how many $s$-$t$ paths no longer than $L$ are there in $G$?

#P-complete, reduction from partition.
Count short paths in a bi-criteria DAGs

Problem (2)

Given a directed acyclic graph $G$ with two sets of edge-weights, two vertices $s$ and $t$, and threshold lengths $L_1, L_2$, how many $s$-$t$ paths no longer than $L_1$ when evaluated on the first set of edge lengths and no longer than $L_2$ when evaluated on the second set of edge lengths are there in $G$?
Why?
Shortest paths on DAGs

- HMM Viterbi decoding
- Biological sequence alignment
- De-novo peptide sequencing
- Dynamic programming
How?
Counting paths on DAGs

Vertices in topological order, $s = v_1$, $t = v_n$.

How many paths no longer than $\ell$ from $v_1$ to $v_i$?
Counting paths on DAGs

Vertices in topological order, \( s = v_1, t = v_n \).

How many paths no longer than \( \ell \) from \( v_1 \) to \( v_i \)?

\( \ell \in [28, 32) \)
Counting paths on DAGs

Vertices in topological order, $s = v_1$, $t = v_n$. 

How many paths no longer than $\ell$ from $v_1$ to $v_i$?

$\ell \in [28, 32)$
Counting paths on DAGs (2)

$\tau(v_i, a)$: Minimum length $\ell$ so that there are at least $a \, v_1 \ldots v_i$ paths no longer than $\ell$.

$\tau(v_n, a) \leq L \leq \tau(v_n, a + 1)$
Recurrence for counting paths on DAGs

Paths to \( v_i \) come from:

\[
\tau(v_i, a) = \min_{a_1, \ldots, a_d} \left( \tau(p_1, a_1) + l_1 + \ldots + \tau(p_d, a_d) + l_d \right)
\]

Problems:

1. \( a \) runs from 0 to \( 2^n - 2 \).
2. Trying all possible \( a_1, \ldots, a_d \) is exponential in maximum in-degree.
Recurrence for counting paths on DAGs

Paths to $v_i$ come from:

$$
\tau(v_i, a) = \max \left\{ \tau(p_1, a_1) + l_1, \ldots, \tau(p_d, a_d) + l_d \right\}
$$
Recurrence for counting paths on DAGs

Paths to $v_i$ come from:

\[ \tau(v_i, a) = \min_{a_1, \ldots, a_d} \max_{\sum a_j = a} \left\{ \begin{array}{l} \tau(p_1, a_1) + l_1 \\ \vdots \\ \tau(p_d, a_d) + l_d \end{array} \right\} \]
Recurrence for counting paths on DAGs

Paths to $v_i$ come from:

$$
\tau(v_i, a) = \min_{a_1, \ldots, a_d} \max_{\sum a_j = a} \left\{ \begin{array}{c}
\tau(p_1, a_1) + l_1 \\
\vdots \\
\tau(p_d, a_d) + l_d
\end{array} \right. 
$$

Problems:

1. $a$ runs from 0 to $2^{n-2}$.
2. Trying all possible $a_1, \ldots, a_d$ is exponential in maximum in-degree.
Discretization

Division points at $(1 + \varepsilon)^k \rightarrow 1 + \varepsilon$ precision.
Division points at $(1 + \varepsilon)^k \rightarrow 1 + \varepsilon$ precision.

In reality we can err when evaluating the recurrence so we need to be even more precise, $\rightarrow (1 + \varepsilon)^{k/n}$. Number of different bins for $a$ is $O(n^2 \varepsilon^{-1})$. 
Minimization

\[ \tau(v_i, a) = \min_{a_1, \ldots, a_d} \max_{\sum a_j = a} \begin{cases} \tau(p_1, a_1) + l_1 \\ \vdots \\ \tau(p_d, a_d) + l_d \end{cases} \]

\[ \tau(p_i, a_i) \text{ is monotonous with increasing } a_i: \]
Minimization

$$\tau(v_i, a) = \min_{a_1, \ldots, a_d} \max_{\sum a_j = a} \left\{ \begin{array}{ll} \tau(p_1, a_1) + l_1 \\ \vdots \\ \tau(p_d, a_d) + l_d \end{array} \right\}$$

$$\tau(p_i, a_i)$$ is monotonous with increasing $$a_i$$:
Minimization

\[ \tau(v_i, a) = \min_{a_1, \ldots, a_d} \max_{\sum a_j = a} \left\{ \begin{array}{l}
\tau(p_1, a_1) + l_1 \\
\vdots \\
\tau(p_d, a_d) + l_d
\end{array} \right\} \]

\( \tau(p_i, a_i) \) is monotonous with increasing \( a_i \):
Minimization

\[
\tau(v_i, a) = \min_{a_1, \ldots, a_d} \max_{\sum a_j = a} \begin{cases} 
\tau(p_1, a_1) + l_1 \\
\vdots \\
\tau(p_d, a_d) + l_d 
\end{cases}
\]

\(\tau(p_i, a_i)\) is monotonous with increasing \(a_i\):
Minimization

\[ \tau(v_i, a) = \min_{a_1, \ldots, a_d} \max_{\sum a_j = a} \left\{ \begin{array}{l}
\tau(p_1, a_1) + l_1 \\
\vdots \\
\tau(p_d, a_d) + l_d
\end{array} \right\} \]

\( \tau(p_i, a_i) \) is monotonous with increasing \( a_i \):
Minimization

\[ \tau(v_i, a) = \min_{a_1, \ldots, a_d} \max_{\sum a_j = a} \left\{ \begin{array}{l} \tau(p_1, a_1) + l_1 \\ \vdots \\ \tau(p_d, a_d) + l_d \end{array} \right\} \]

\( \tau(p_i, a_i) \) is monotonous with increasing \( a_i \):
Minimization

\[ \tau(v_i, a) = \min_{a_1, \ldots, a_d} \max_{\sum a_j = a} \left\{ \tau(p_1, a_1) + l_1, \ldots, \tau(p_d, a_d) + l_d \right\} \]

\( \tau(p_i, a_i) \) is monotonous with increasing \( a_i \):
Minimization

\[
\tau(v_i, a) = \min_{a_1, \ldots, a_d} \max_{\sum a_j = a} \left\{ \begin{array}{l} \tau(p_1, a_1) + l_1 \\
\vdots \\
\tau(p_d, a_d) + l_d \end{array} \right. 
\]

\(\tau(p_i, a_i)\) is monotonous with increasing \(a_i\):

We can do all \(a\) at once resulting in \(O(mn^2\varepsilon^{-1})\) time complexity.
Bi-criteria version is inapproximable

Reduction from knapsack: Is there a selection such that 
\[ \sum w \leq W, \sum p \geq P? \]

Count paths no longer than \( W \) and \( -P \).
Bi-criteria version is inapproximable

Reduction from knapsack: Is there a selection such that
\[ \sum w \leq W, \sum p \geq P? \]

Count paths no longer than \( W \) and \( nC - P \).
Bi-criteria version is inapproximable

Reduction from knapsack: Is there a selection such that 
\[ \sum w \leq W, \sum p \geq P? \]

Count paths no longer than \( W \) and \( nC - P \).

Bi-criteria FPTAS for \#Knapsack [Gopalan et al., 2010].
Bi-criteria version (2)

\( \tau'(v_i, a, L_1) \) – minimum length \( L_2 \) such that there are at least \( a \) paths \( v_1 \rightarrow v_i \) no longer than \( L_1 \) in first instance and no longer than \( L_2 \) in second.

Goal: Find \( a \) such that \( \tau'(t, a, L_1) \leq L_2 \leq \tau'(t, a + 1, L_1) \).

\[
\tau'(i, a, L_1) = \min_{a_1, \ldots, a_d} \max_{\sum a_j = a} \left\{ \begin{array}{c}
\tau'(p_1, a_1, L_1 - l_1) + l_1' \\
\vdots \\
\tau'(p_d, a_1, L_1 - l_d) + l_d'
\end{array} \right\}
\]
Bi-criteria version (2)

\( \tau'(v_i, a, L_1) \) – minimum length \( L_2 \) such that there are at least \( a \) paths \( v_1 \rightarrow v_i \) no longer than \( L_1 \) in first instance and no longer than \( L_2 \) in second.

Goal: Find \( a \) such that \( \tau'(t, a, L_1) \leq L_2 \leq \tau'(t, a + 1, L_1) \).

\[
\tau'(i, a, L_1) = \min_{a_1, \ldots, a_d} \max_{\sum a_j = a} \left\{ \begin{array}{ll}
\tau'(p_1, a_1, L_1 - l_1) + l_1' \\
\vdots \\
\tau'(p_d, a_1, L_1 - l_d) + l_d' 
\end{array} \right. 
\]

\((1 + \varepsilon)\)-approximate number of paths with length at most \( L_1 \) in the first graph and at most \((1 + \delta)\) times different from \( L_2 \) in the second graph: \( O(mn^3 \varepsilon^{-1} \delta^{-1} \log n \log L_1) \).
Conclusion and future work

- Counting approximate solutions is relevant for robustness.
- Counting paths shorter than a threshold is $\#P$-complete but has a FPTAS.
- Bi-criteria version is inapproximable
- Results are applicable for a large class of counting problems.

- Force ordering upon other problems, for instance spanning trees.
Thank you!