On a DAG Partitioning Problem

Abbas Mehrabian
amehrabi@uwaterloo.ca
joint work with Soroush Alamdari

University of Waterloo

Workshop on Web Algorithms, Dalhousie University
June 23rd, 2012

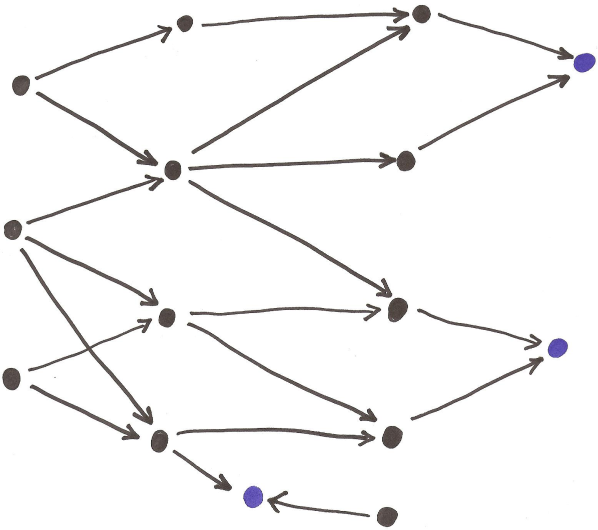
Problem Statement

Leskovec, Backstrom, Kleinberg 2009

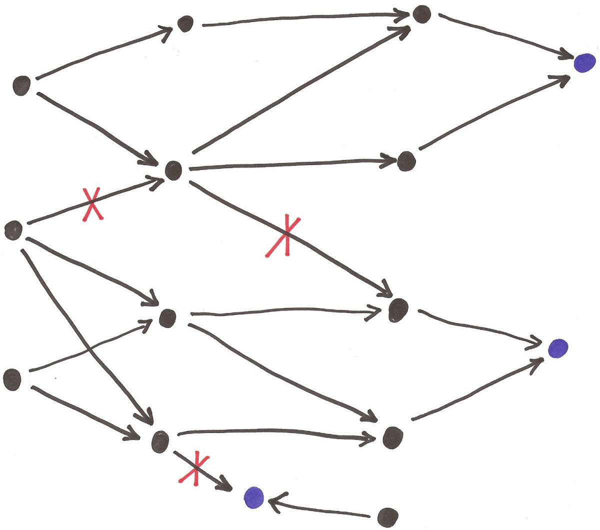
Given a directed acyclic graph G with arc weights,

Delete a set of arcs of minimum total weight so that each of the resulting connected components has exactly one sink.

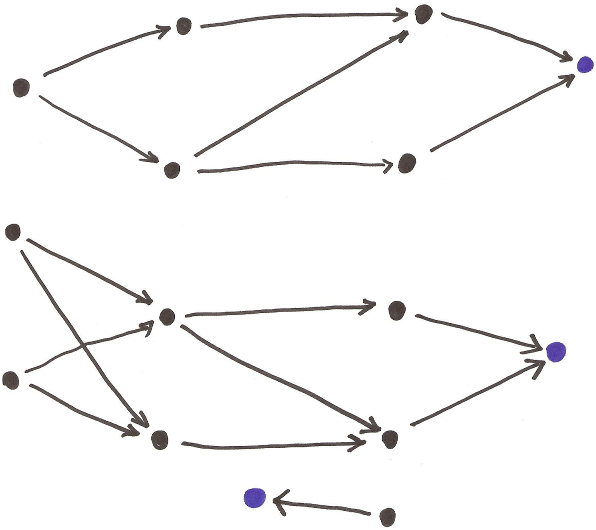
Example

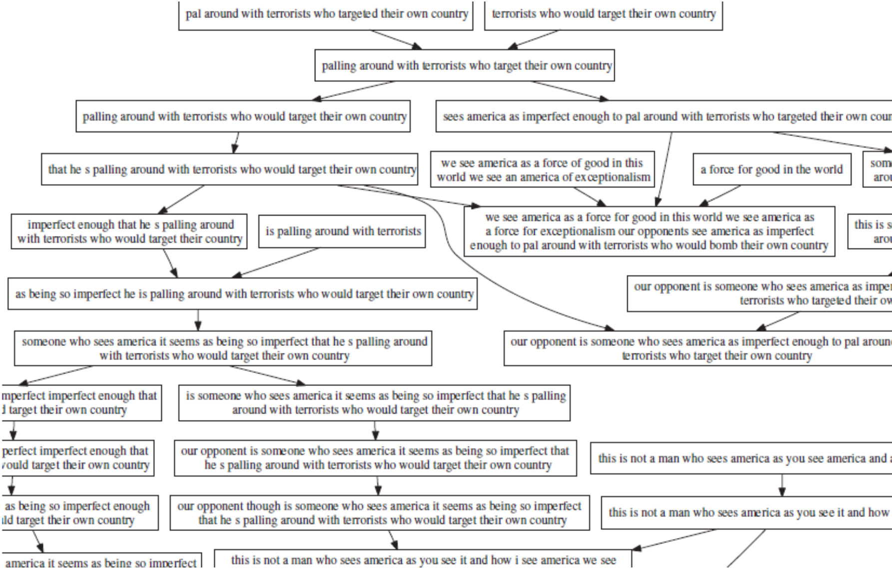


Example

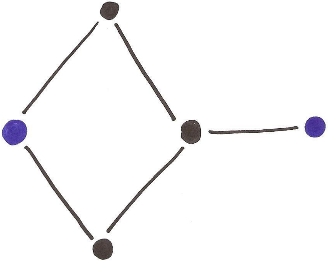


Example

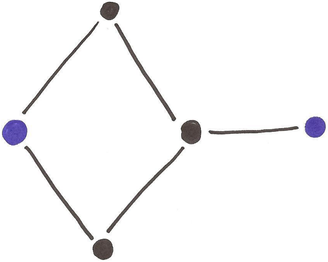


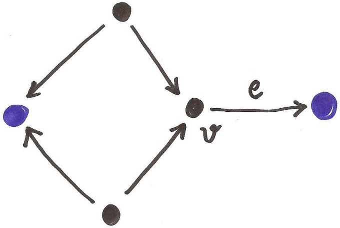

pal around with terrorists who targeted their own country
terrorists who would target their own country
palling around with terrorists who target their own country
palling around with terrorists who would target their own country
sees america as imperfect enough to pal around with terrorists who targeted their own coun
that he s palling around with terrorists who would target their own country
we see america as a force of good in this world we see an america of exceptionalism
a force for good in the world
imperfect enough that he s palling around with terrorists who would target their country
is palling around with terrorists
we see america as a force for good in this world we see america as a force for exceptionalism our opponents see america as imperfect enough to pal around with terrorists who would bomb their own country
as being so imperfect he is palling around with terrorists who would target their own country
our opponent is someone who sees america as imper terrorists who targeted their ow
someone who sees america it seems as being so imperfect that he s palling around with terrorists who would target their own country
our opponent is someone who sees america as imperfect enough to pal aroun terrorists who target their own country
is someone who sees america it seems as being so imperfect that he s palling around with terrorists who would target their own country
our opponent is someone who sees america it seems as being so imperfect that he s palling around with terrorists who would target their own country
this is not a man who sees america as you see america and
our opponent though is someone who sees america it seems as being so imperfect that he s palling around with terrorists who would target their own country
this is not a man who sees america as you see it and how
this is not a man who sees america as you see it and how i see america we see

Multiway Cut Problem



Multiway Cut Problem





NP-hardness

Leskovec et al. proved that DAG Partitioning is NP-hard.

Multiway Cut with k terminals $\Rightarrow$ DAG Partitioning with k sinks.

Approximation Algorithms?

Theorem (Călinescu, Karloff, Rabani 1998)

Multiway cut with k terminals is $(\frac{3}{2} - \frac{1}{k})$-approximable.

Theorem (Alamdari, M 2012)

For any fixed $\epsilon > 0$ there is no polynomial $(n^{1-\epsilon})$-approximation algorithm for DAG Partitioning problem if $P \neq NP$.

Hardness result holds for DAGs with two sinks, unit weights, max outdegree 3

Approximation Algorithms?

Theorem (Călinescu, Karloff, Rabani 1998)

Multiway cut with k terminals is $(\frac{3}{2} - \frac{1}{k})$-approximable.

Theorem (Alamdari, M 2012)

For any fixed $\epsilon > 0$ there is no polynomial $(n^{1-\epsilon})$-approximation algorithm for DAG Partitioning problem if $P \neq NP$.

Hardness result holds for DAGs with two sinks, unit weights, max outdegree 3

A Positive Result

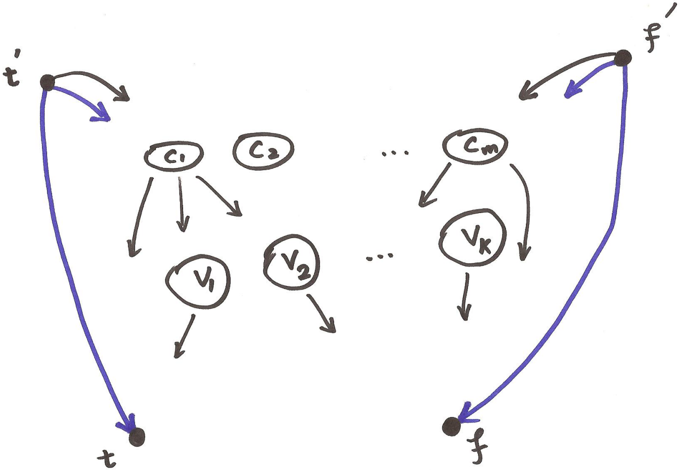
Theorem (Alamdari, M 2012)

The DAG Partitioning problem can be solved optimally on a DAG with bounded pathwidth.

Two Definitions

1. Unique sink property
2. The sink of a vertex

General Structure

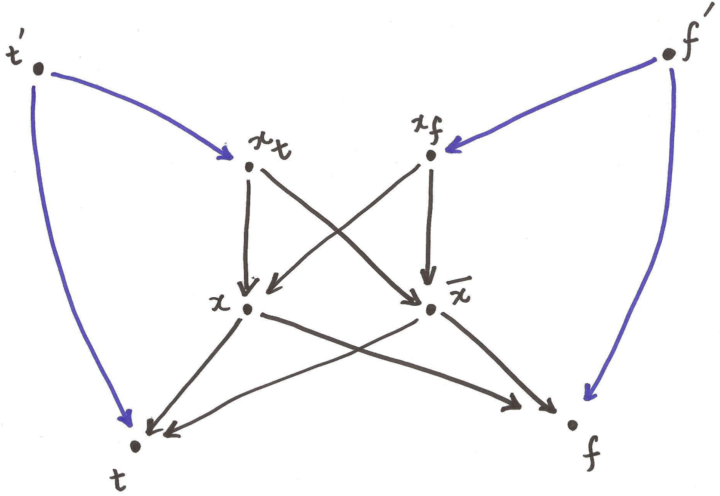


The Key Claim

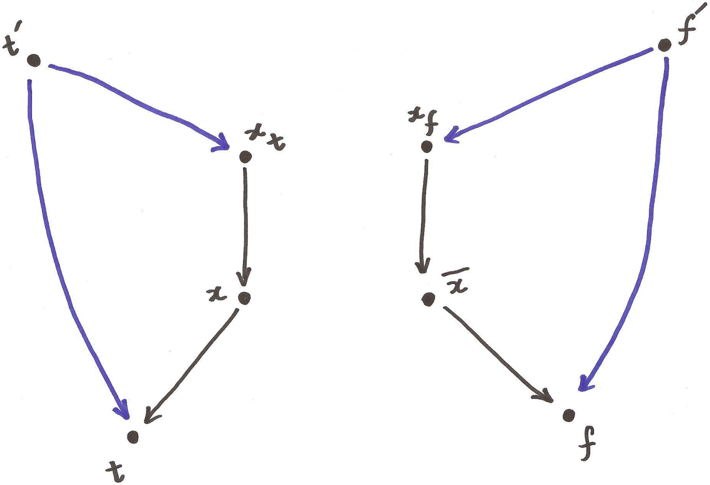
Claim

The 3-SAT instance is satisfiable $\Leftrightarrow$
DAG Partitioning has a solution that does not delete any blue arcs

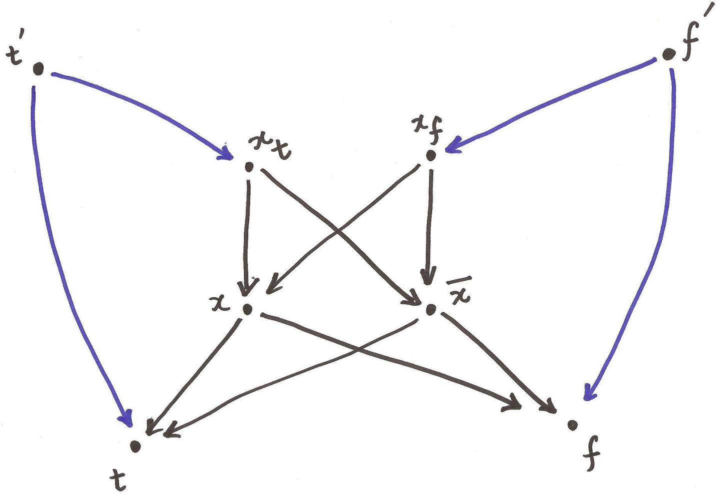
The Gadget for a Variable



The Gadget for a Variable

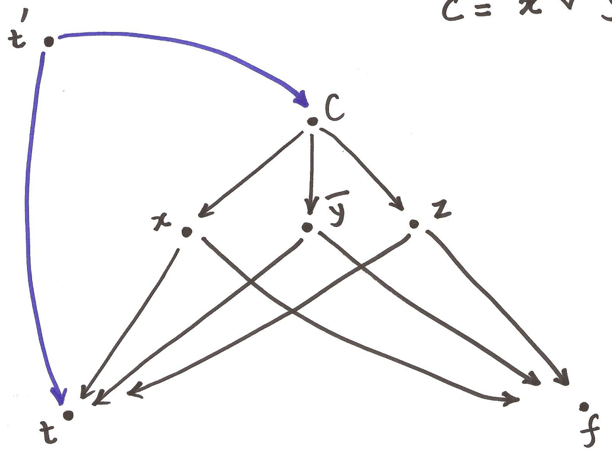


The Gadget for a Variable



The Gadget for a clause

$C = x \vee \bar{y} \vee z$



Avoiding Heavy Edges and Large Out-degrees

1. two sinks
2. blue (heavy) edges
3. large out-degrees

Definition of pathwidth

Definition

The family $\{W_i : 1 \leq i \leq m\}$ of subsets of $V(G)$ is a path decomposition of G if it satisfies:

(i) $\cup_{1 \leq i \leq m} W_i = V(G)$.

(ii) $\forall uv \in E(G)$, $\exists$ a bag containing u and v.

(iii) Every $v \in V(G)$ is contained in a consecutive set of bags.

Width of the path decomposition is $\max\{|W_i|\}$.

Given a path decomposition of width k, there is an algorithm with running time $2^{O(k^2)} n$.

Definition of pathwidth

Definition

The family $\{W_i : 1 \leq i \leq m\}$ of subsets of $V(G)$ is a path decomposition of G if it satisfies:

(i) $\cup_{1 \leq i \leq m} W_i = V(G)$.

(ii) $\forall uv \in E(G)$, $\exists$ a bag containing u and v.

(iii) Every $v \in V(G)$ is contained in a consecutive set of bags.

Width of the path decomposition is $\max\{|W_i|\}$.

Given a path decomposition of width k, there is an algorithm with running time $2^{O(k^2)}n$.

The Subproblems

Alter the path decomposition so that every child is only slightly different with its parent.

The subproblems are of the form (H, X, P, F, D).

The Subproblems

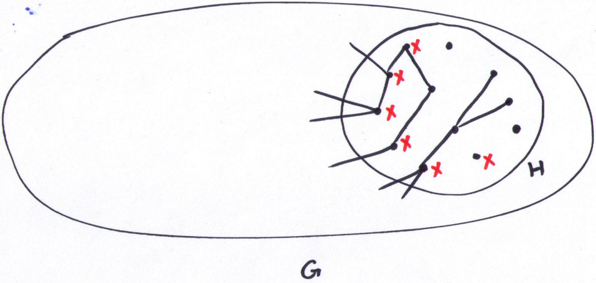
Alter the path decomposition so that every child is only slightly different with its parent.
The subproblems are of the form (H, X, P, F, D).

The subproblems

$(H, X, \qquad)$

$H \subseteq G$

$X \subseteq V(H)$



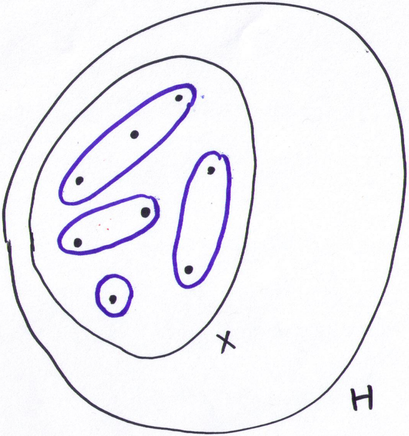
The subproblems

$(H, X, P, \quad)$

$H \subseteq G$

$X \subseteq V(H)$

$P \subseteq 2^{2^X}$



The subproblems

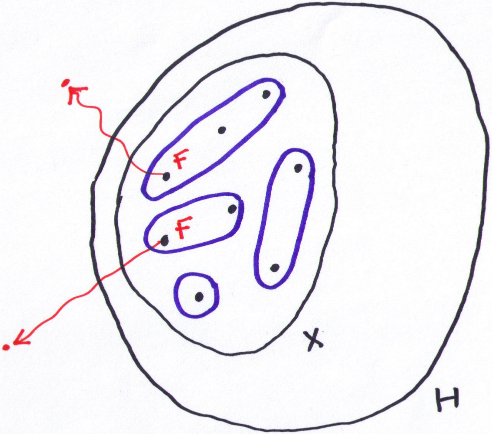
$(H, X, P, F, \quad)$

$H \subseteq G$

$X \subseteq V(H)$

$P \subseteq 2^{2^X}$

$F \subseteq X$



The subproblems

(H, X, P, F, D)

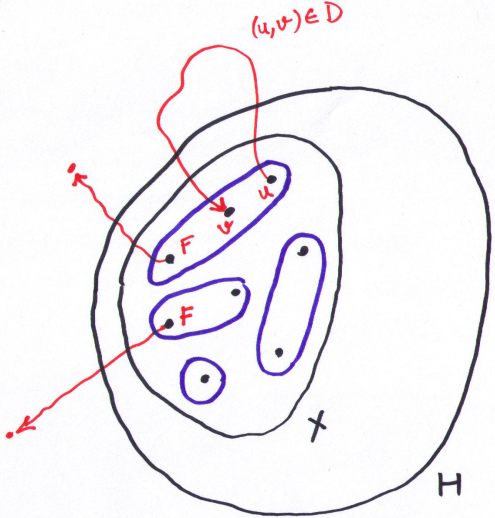
$H \subseteq G$

$X \subseteq V(H)$

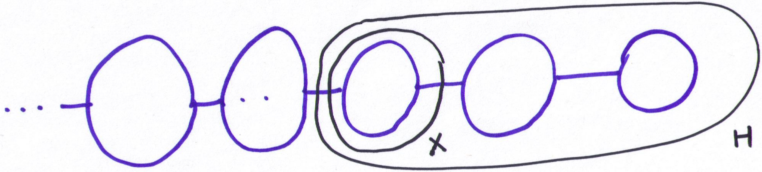
$P \subseteq 2^{2^X}$

$F \subseteq X$

$D \subseteq X^2$



The actual subproblems



The Running Time

The subproblems are of the form (H, X, P, F, D).

H $O(k^2 n)$

X 1

P 2^{k^2}

F 2^k

D 2^{k^2}

$2^{O(k^2)} n$ subproblems, each update in time $O(k^3)$.

The Running Time

The subproblems are of the form (H, X, P, F, D).

H $O(k^2 n)$

X 1

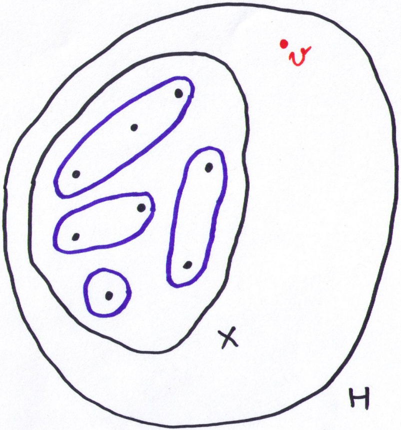
P 2^{k^2}

F 2^k

D 2^{k^2}

$2^{O(k^2)} n$ subproblems, each update in time $O(k^3)$.

updating - Example 1: Child has one vertex morethan parent



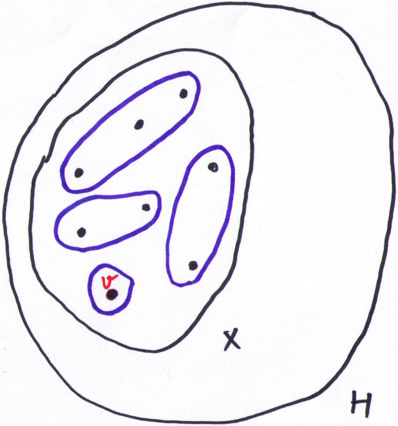
Take the minimum

over built subproblems!

updating - Example 2: child has one vertex less than parent

assume: v is singleton,

$v \notin F$

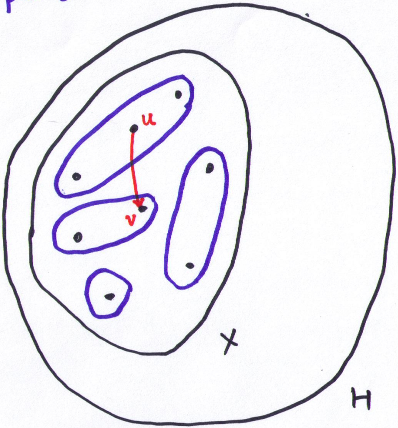


v is a sink,

delete it!

updating Example 3 : child has one edge less than parent

assume: u, v in different parts



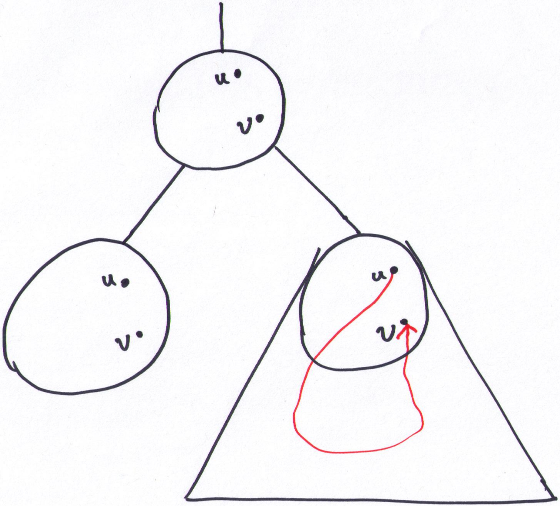
opt (parent) =

opt(child) + w (uv)

What About Treewidth?

OPEN. What about bounded-treewidth graphs!

The Trouble with Tree Decompositions



Thank you Slide

Thanks for your attention :-)