Conditional hardness and equvalences for

GRAPH PROBLEMS

Virginia V. Williams
Stanford

Undirected G, weigthed, n nodes, m edges

Important graph parameters

- Eccentricities: the eccentricity $e(x)$ of x is $\max_y d(x, y)$
- Diameter: $\max_x e(x) = \max_{x, y} d(x, y)$
- Radius: $\min_x e(x) = \min_x \max_y d(x, y)$
- Median: $\min_x \sum_y d(x, y)$

Best algorithms: compute all pairs shortest paths (APSP):
$\sim n^{3-o(1)}$ for dense graphs ($m \sim n^2$), $\sim n^2$ for sparse ($m \sim n$)

Can one get $n^{3-\varepsilon}$ for dense? Can one get $n^{2-\varepsilon}$ for sparse?
What about for approximations?

Outline

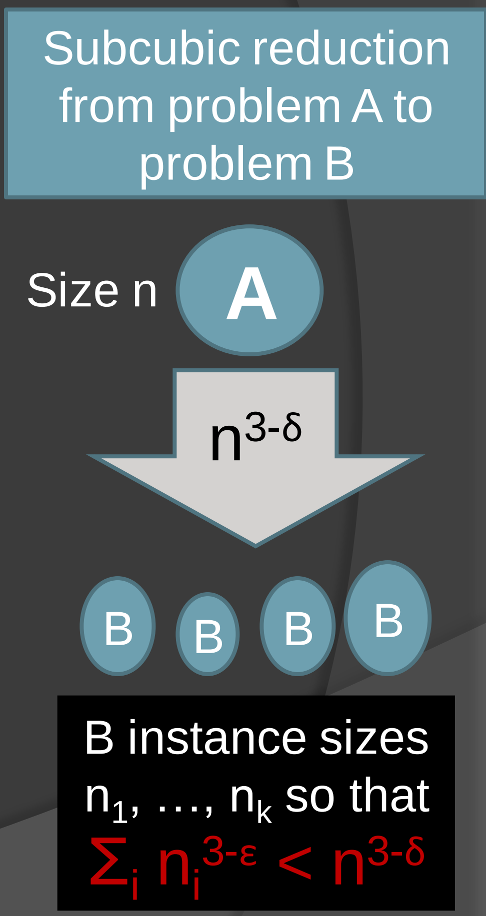
- Hardness and equivalences for dense graphs
- Hardness for sparse graphs

The dense graph regime: beating n^3

Theorem [VW'10, AGV'14]: APSP is equivalent to Radius, Median and many other graph problems, under *subcubic reductions*.

Equivalence of problems A and B means:

any $O(n^{3-\varepsilon})$ time alg for problem B can be converted to an $O(n^{3-\delta})$ time alg for problem A, and vice versa.



APSP Research

Big Open Problem: APSP in *truly* subcubic time?

APSP Conjecture: APSP requires $n^{3-o(1)}$ time.

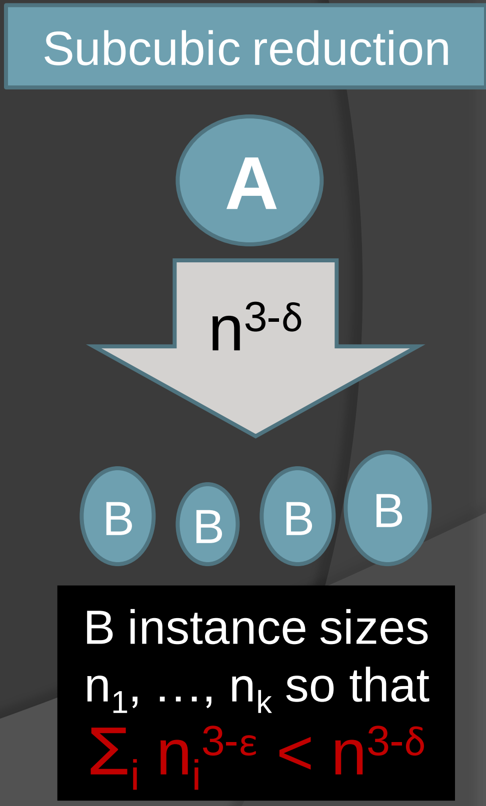
Author	Runtime	
Fredman	$\mathbf{n^3} \log \log^{1/3} n / \log^{1/3} n$	1976
Takaoka	$\mathbf{n^3} \log \log^{1/2} n / \log^{1/2} n$	1992
Dobosiewicz	$\mathbf{n^3} / \log^{1/2} n$	1992
Han	$\mathbf{n^3} \log \log^{5/7} n / \log^{5/7} n$	2004
Takaoka	$\mathbf{n^3} \log \log^{2} n / \log n$	2004
Zwick	$\mathbf{n^3} \log \log^{1/2} n / \log n$	2004
Chan	$\mathbf{n^3} / \log n$	2005
Han	$\mathbf{n^3} \log \log^{5/4} n / \log^{5/4} n$	2006
Chan	$\mathbf{n^3} \log \log^{3} n / \log^{2} n$	2007
Han, Takaoka	$\mathbf{n^3} \log \log n / \log^{2} n$	2012
Williams	$\mathbf{n^3} / \exp(\sqrt{\log n})$	2014

The dense graph regime: beating n^3

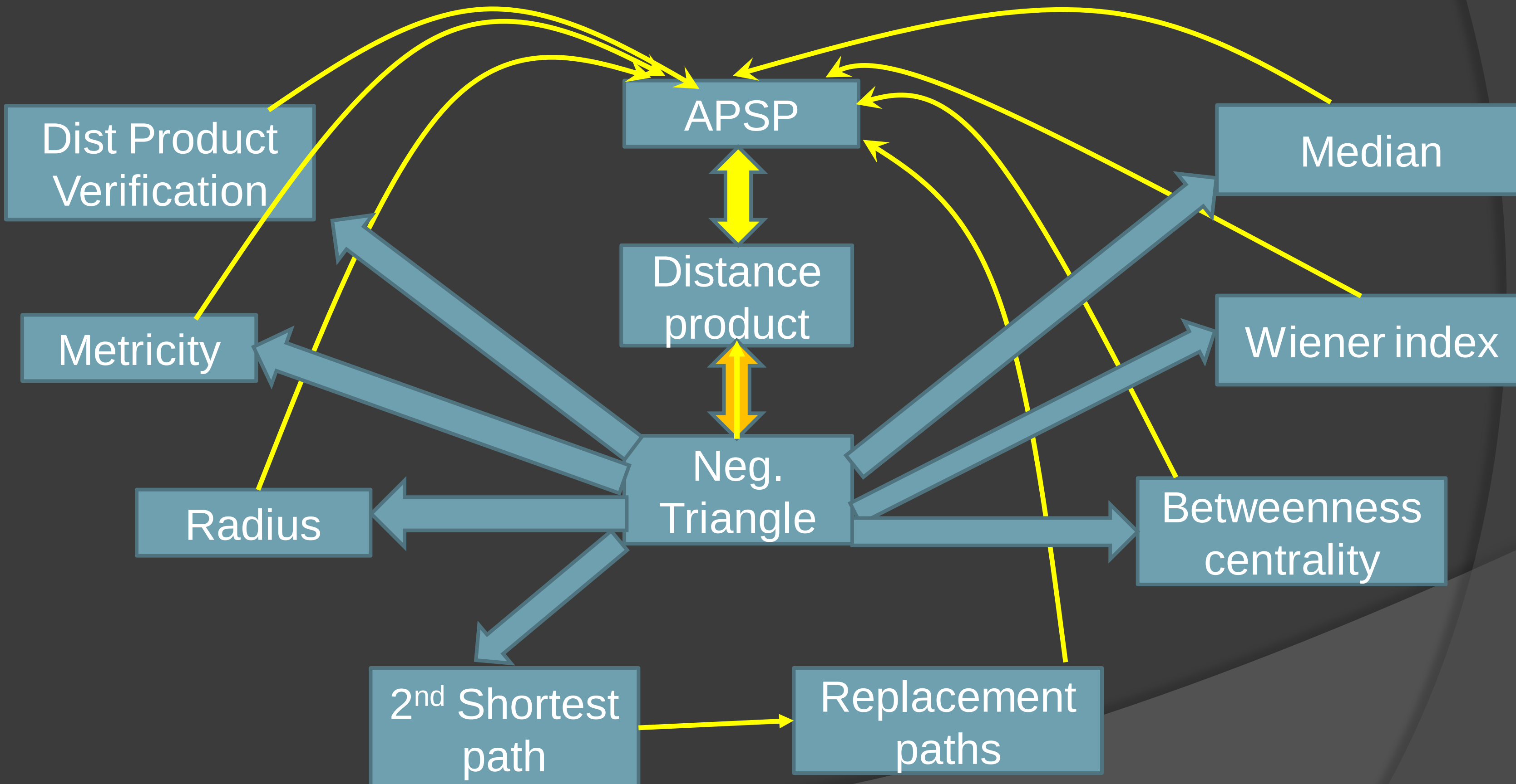
Theorem [VW'10, AGV'14]: APSP is equivalent to Radius, Median and many other graph problems, under *subcubic reductions*.

Equivalence of problems A and B means:

any $O(n^{3-\varepsilon})$ time alg for problem A can be converted to an $O(n^{3-\delta})$ time alg for problem B, and vice versa.



Some known equivalences to APSP



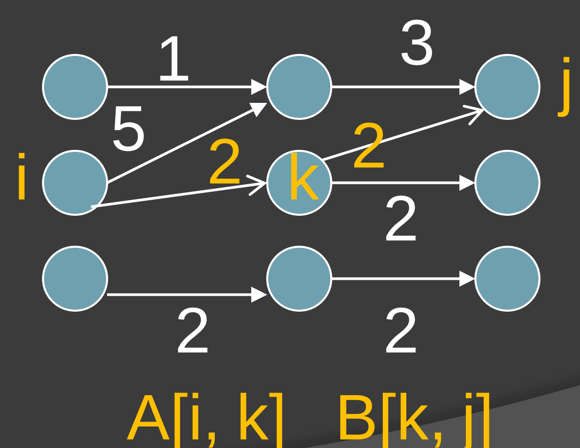
Distance product

Distance product: Given two matrices A, B:

$$(A * B)[i, j] = \min_k (A[i, k] + B[k, j])$$

APSP in T(n) time $\Longrightarrow$ Dist. Prod. in T(n) time

$$A \quad \begin{matrix} 1 & \infty & \infty \\ 5 & 2 & \infty \\ \infty & \infty & 2 \end{matrix}$$



$$\begin{matrix} 3 & \infty & \infty \\ 2 & 2 & \infty \\ \infty & \infty & 2 \end{matrix} \quad B$$

Distance product and APSP

Distance product in $T(n)$ time $\Rightarrow$ APSP in $T(n) \log n$ time

Fischer, Meyer'71

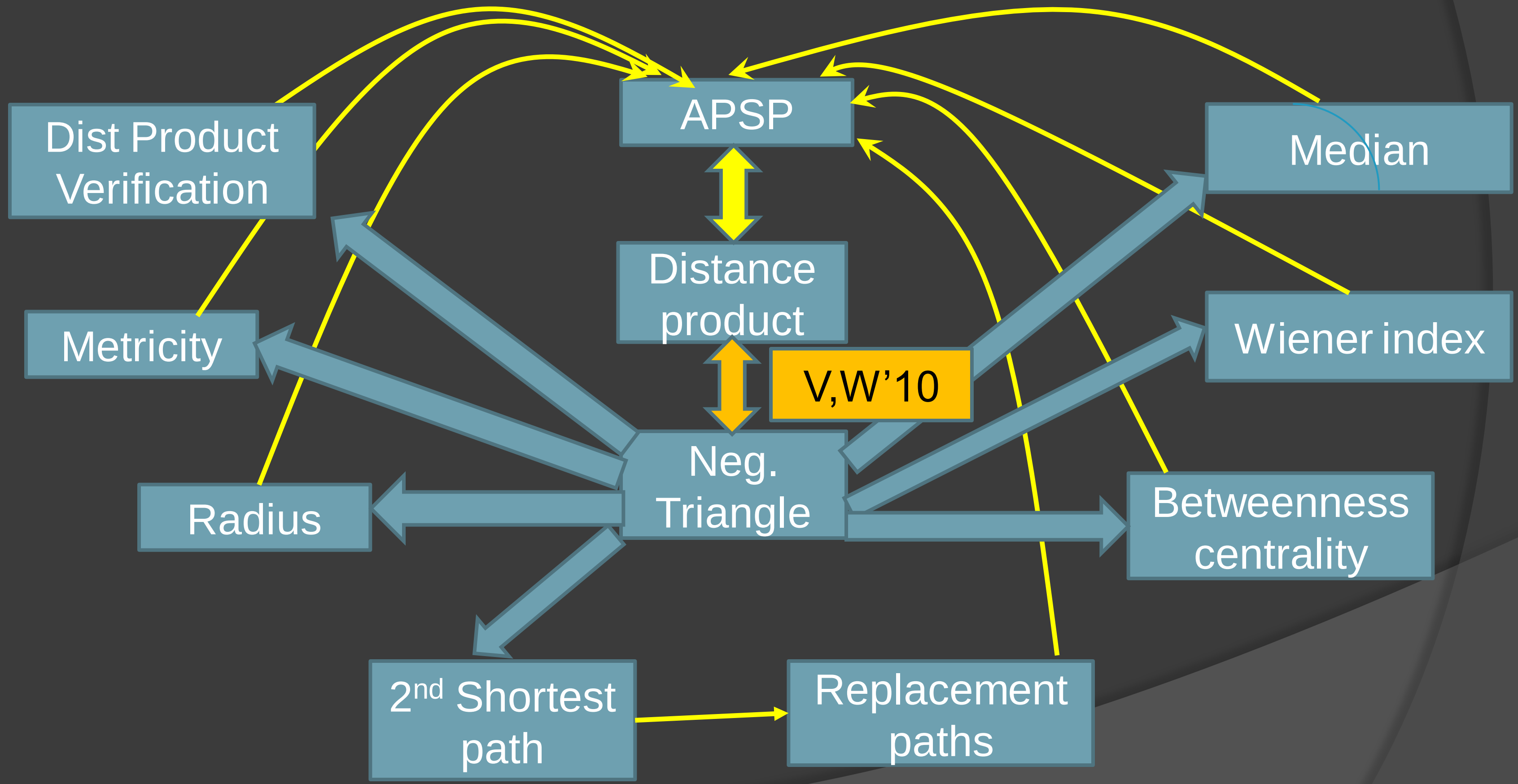
Weighted adjacency matrix of a graph:

$$A[u,v] = \begin{cases} w(u,v) & \text{for edges } (u,v) \text{ and} \\ \infty & \text{for non-edges } (u,v) \\ 0 & \text{for } u=v \end{cases}$$

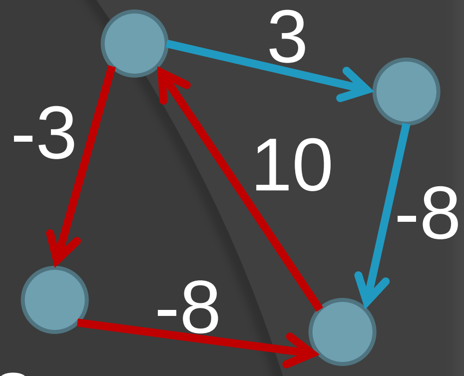
$A^k[u,v]$ = weight of shortest path on $\leq k$ edges

APSP and Distance product are equivalent.

Some known equivalences to APSP



Negative triangle



Input: Graph G with integer edge weights

Output:

‘**Yes**’ if there exist nodes i,j,k in G such that

$$w(i,j) + w(j,k) + w(k,i) < 0$$

‘**No**’ otherwise.

Easy cubic time algorithm!!

In general, no $O(n^{3-\varepsilon})$ algorithm known.

$(A*B)[i, j] = \min_k (A[i, k]+B[k, j])$

Distance product to negative triangle

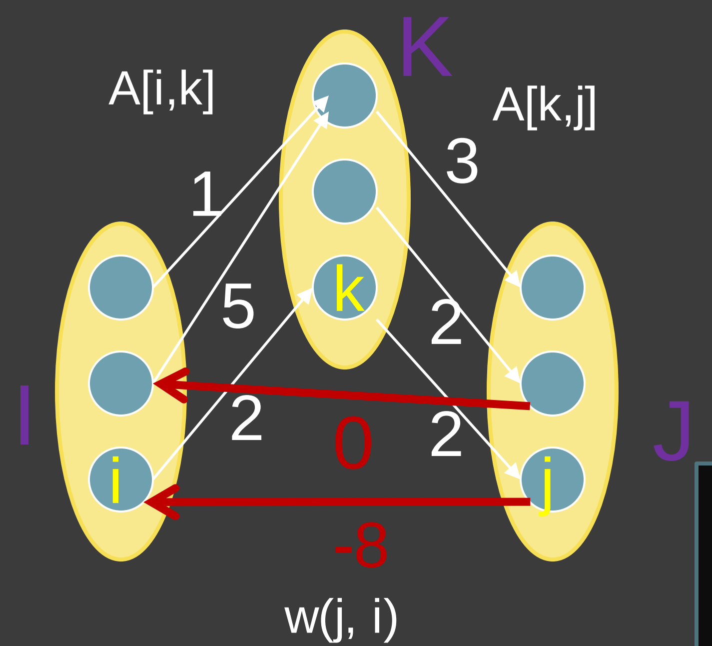
1. Distance product to
   All pairs negative triangles:
   For every j,i in G,
   is there a k so that $w(i,k) + w(k,j) < -w(j,i)$?

2. All pairs negative triangles to
   Negative triangle:
   Are there i,j,k in G, so that
   $$w(i,k) + w(k,j) < -w(j,i) \ ?$$

Reducing distance product to all pairs negative triangle

(A*B)[i, j] = $\min_k$ (A[i, k]+B[k, j])

$$A = \begin{matrix} 1 & \infty & \infty \\ 5 & \infty & \infty \\ \infty & \infty & 2 \end{matrix}$$



$$B = \begin{matrix} 3 & \infty & \infty \\ \infty & 2 & \infty \\ \infty & \infty & 2 \end{matrix}$$

Simultaneous binary search!

Add edges from J to I with carefully chosen weights w(· , ·)

All pairs negative triangles: for every j, i in J x I,

is there a k so that A[i,k]+B[k,j] < -w(j,i)?

then $\min_k$ A[i,k]+B[k,j] < -w(j,i)?

$(A*B)[i, j] = \min_k (A[i, k]+B[k, j])$

Distance product to negative triangle

1. Distance product to
   All pairs negative triangles:
   For every j,i in G,
   is there a k so that $w(i,k) + w(k,j) < -w(j,i)$?

2. All pairs negative triangles to
   Negative triangle:
   Are there i,j,k in G, so that
   $w(i,k) + w(k,j) < -w(j,i)$?

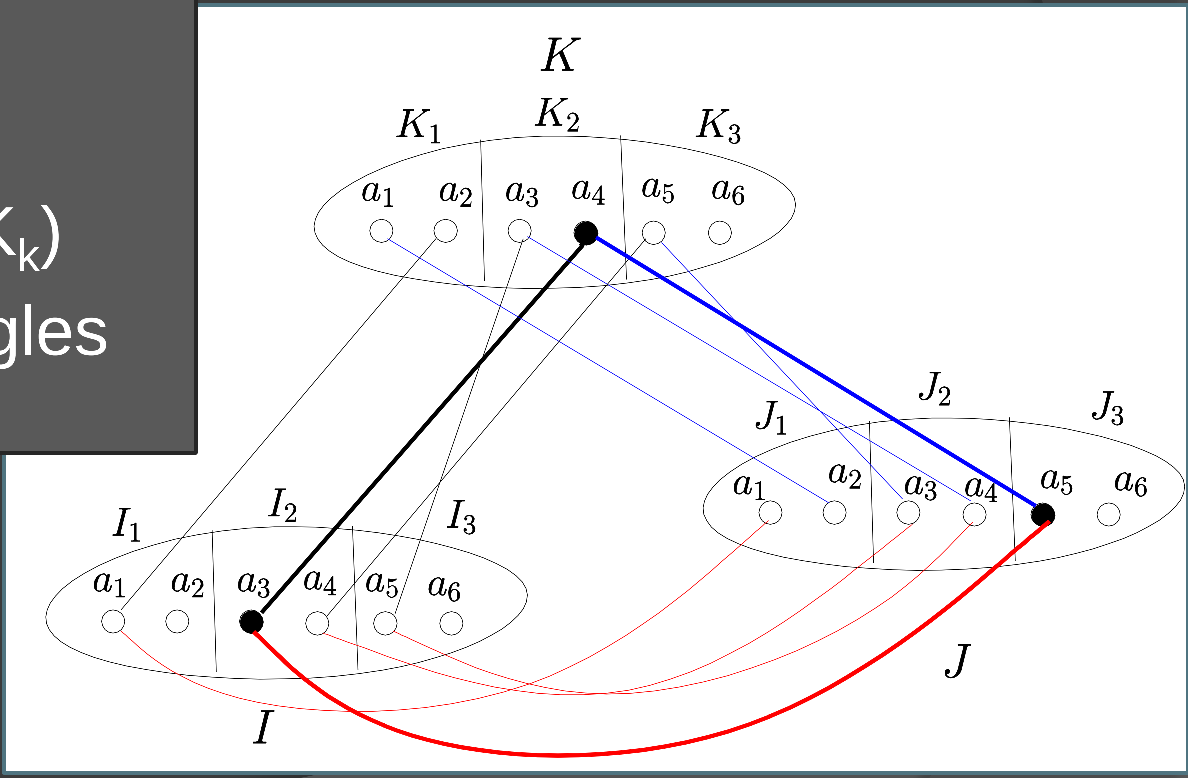
All pairs negative triangle to negative triangle

Idea:

1. Split I, J, K into pieces of small size s

2. Consider all $(n/s)^3$ triples of pieces (I_i,J_j,K_k)

3. Find negative triangles in each triple



All Pairs Negative Triangle

Initialize C : n x n matrix of all-zeros

For every triple (I_x,J_y,K_z) in turn:

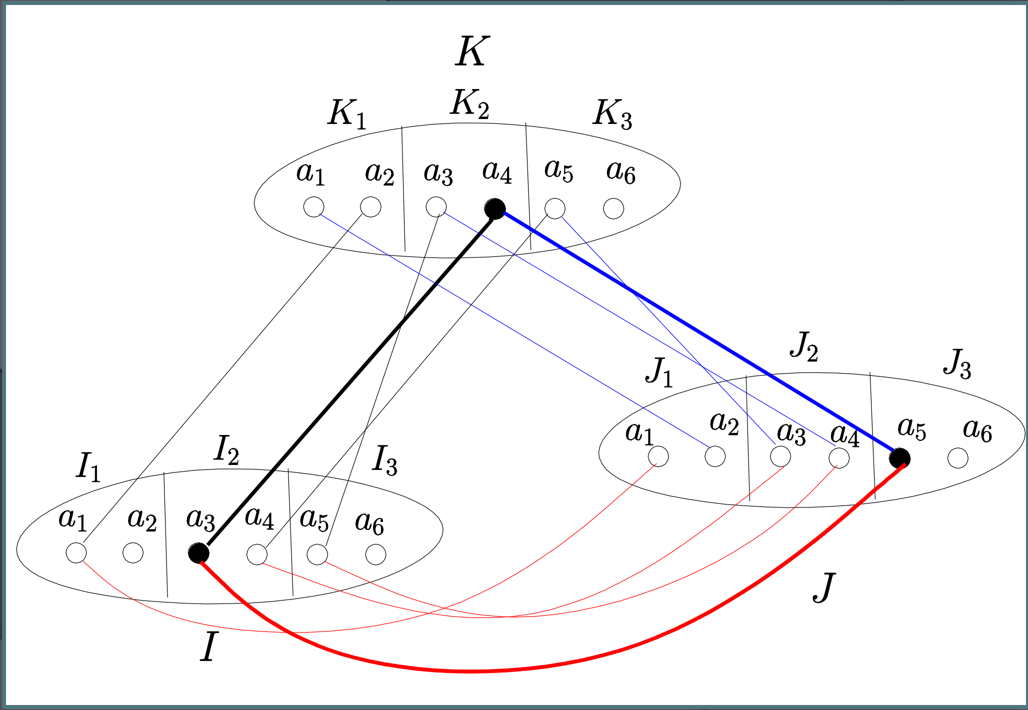
  while (I_x,J_y,K_z) has a negative triangle

    report negative triangle a_x,a_y,a_z

    set $C[a_x,a_y] = 1$

    set $w(a_x,a_y) = \infty$

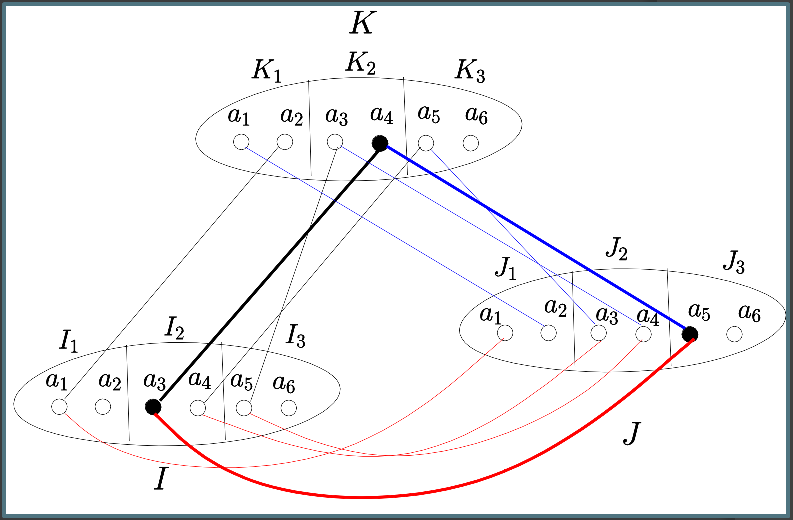
(a_x, a_y) doesn't appear in any more negative triangles!



All Pairs Negative Triangle

Initialize C : n x n matrix of all-zeros
For every triple (I_x,J_y,K_z) in turn:
while (I_x,J_y,K_z) has a negative triangle
report negative triangle a_x,a_y,a_z
set C[a_x,a_y] = 1
set w(a_x,a_y) = ∞

(a_x,a_y) *doesn't appear in any more negative triangles!*



Runtime:

[(#triples) + (#triangles found)] • T(neg. triangle in triple)

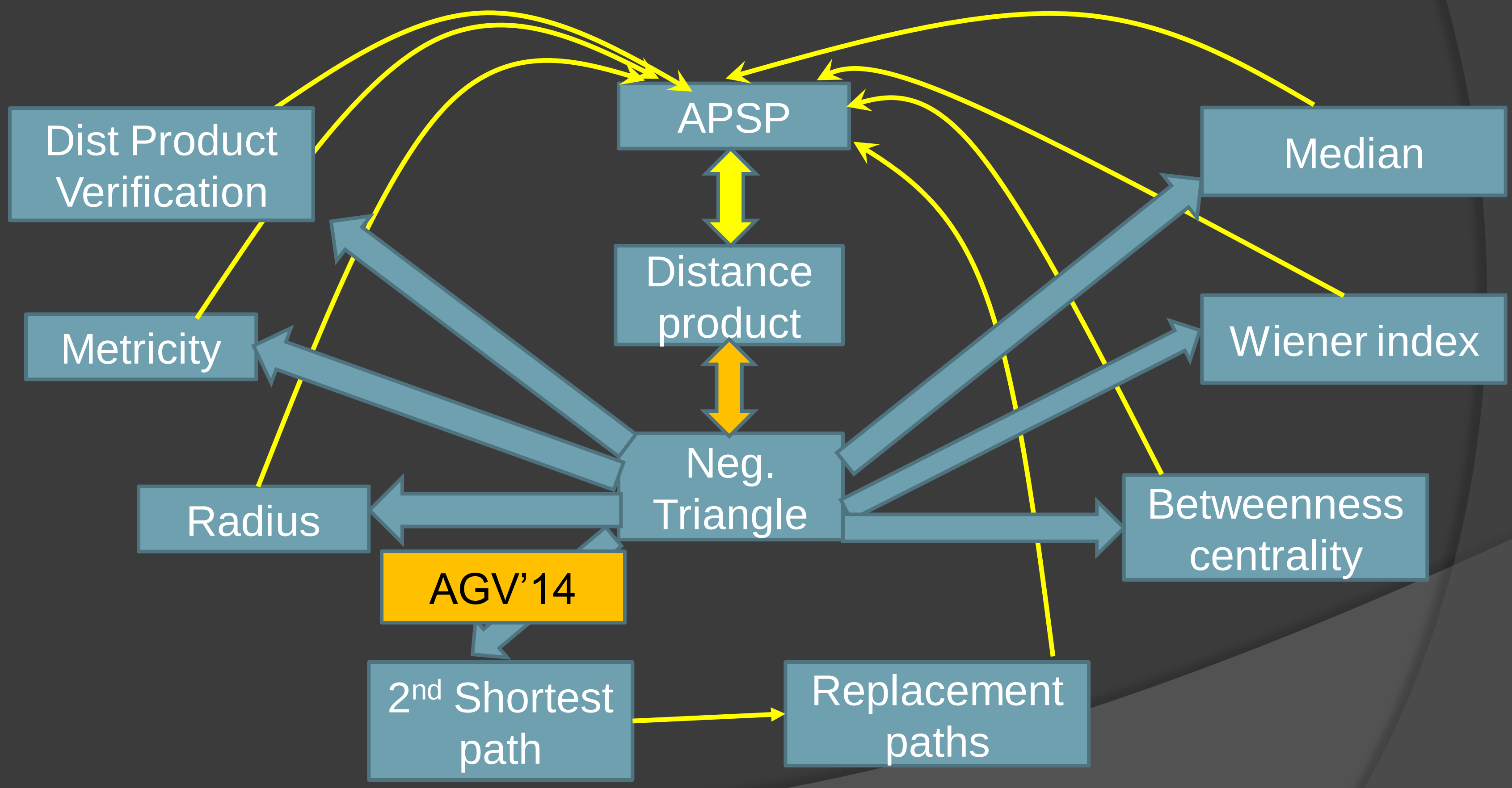
= $((n/s)^3 + n^2)$ • NegTriangle(s)

NegTriangle(n) = n^d

= $n^{2+(d/3)}$ for $s=n^{1/3}$.

Subcubic if d<3.

Some known equivalences to APSP



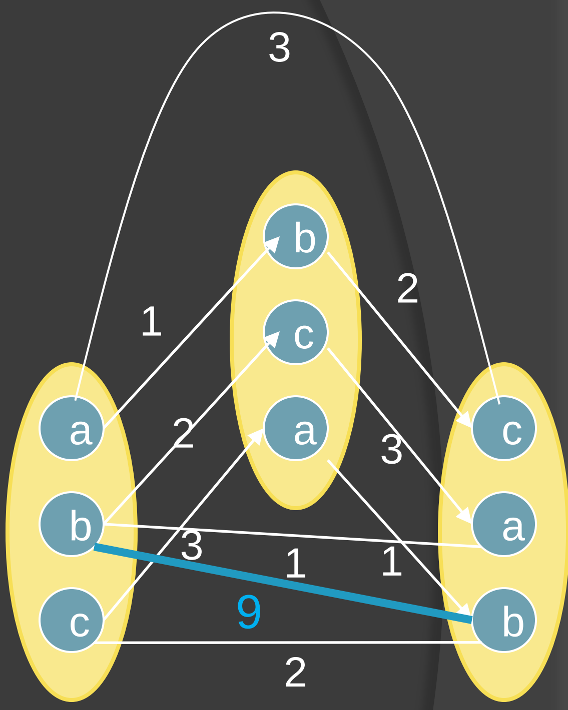
Reducing Neg. Triangle to Radius

WLOG,

in the Negative triangle problem:

- The given graph G is *tripartite*:

  create 3 copies of the vertex set and add 3 copies of each edge



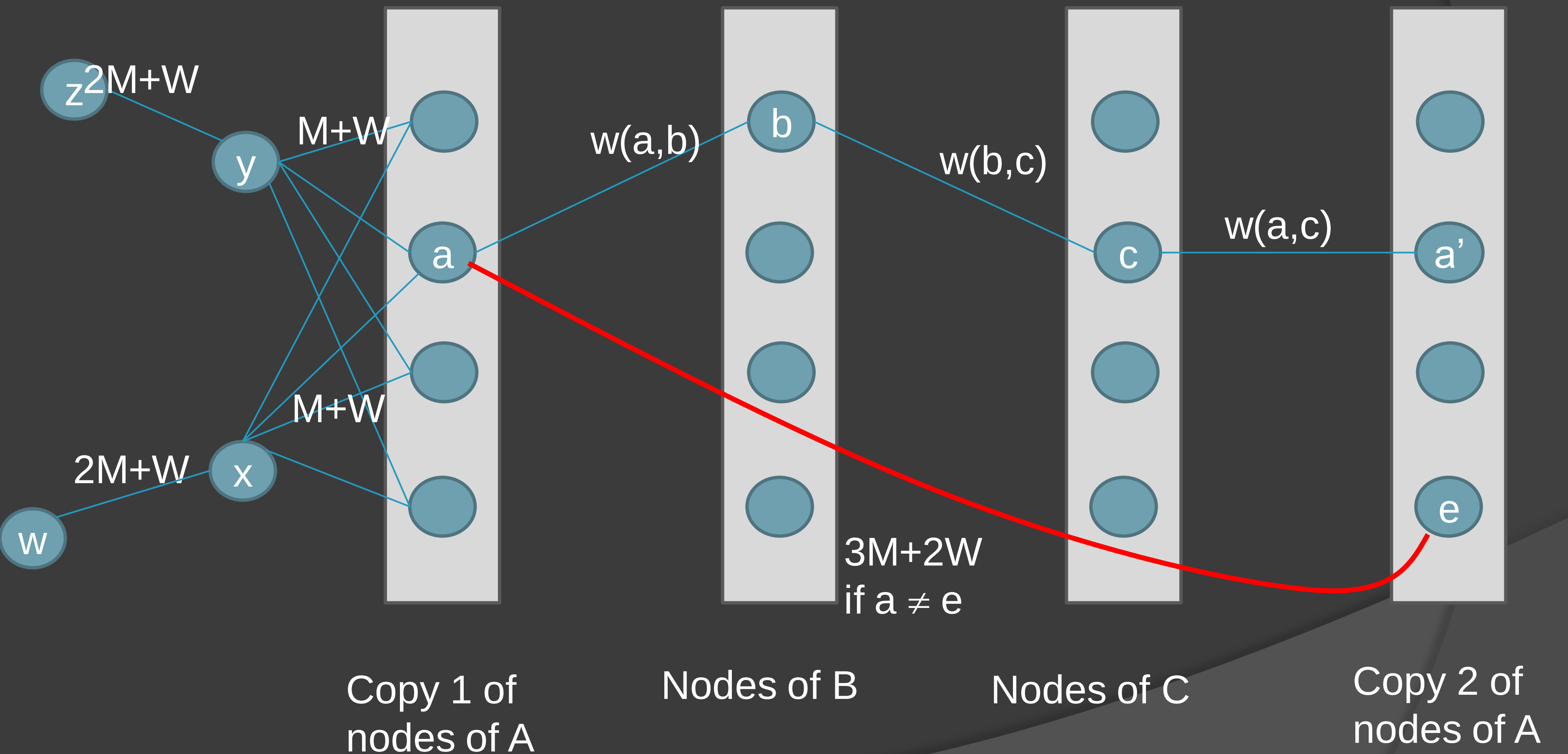
- G is a *complete* tripartite graph: if the largest edge weight were M, make each non-edge between different partitions into an edge of weight 3M. 3M is the new M.

Reducing Neg. Triangle to Radius

Tripartite G with partitions A,B,C, weights in {–M ,…, M}, find triangle of weight <0.
(wlog AxB, BxC, AxC are complete bipartite, M = no edge)

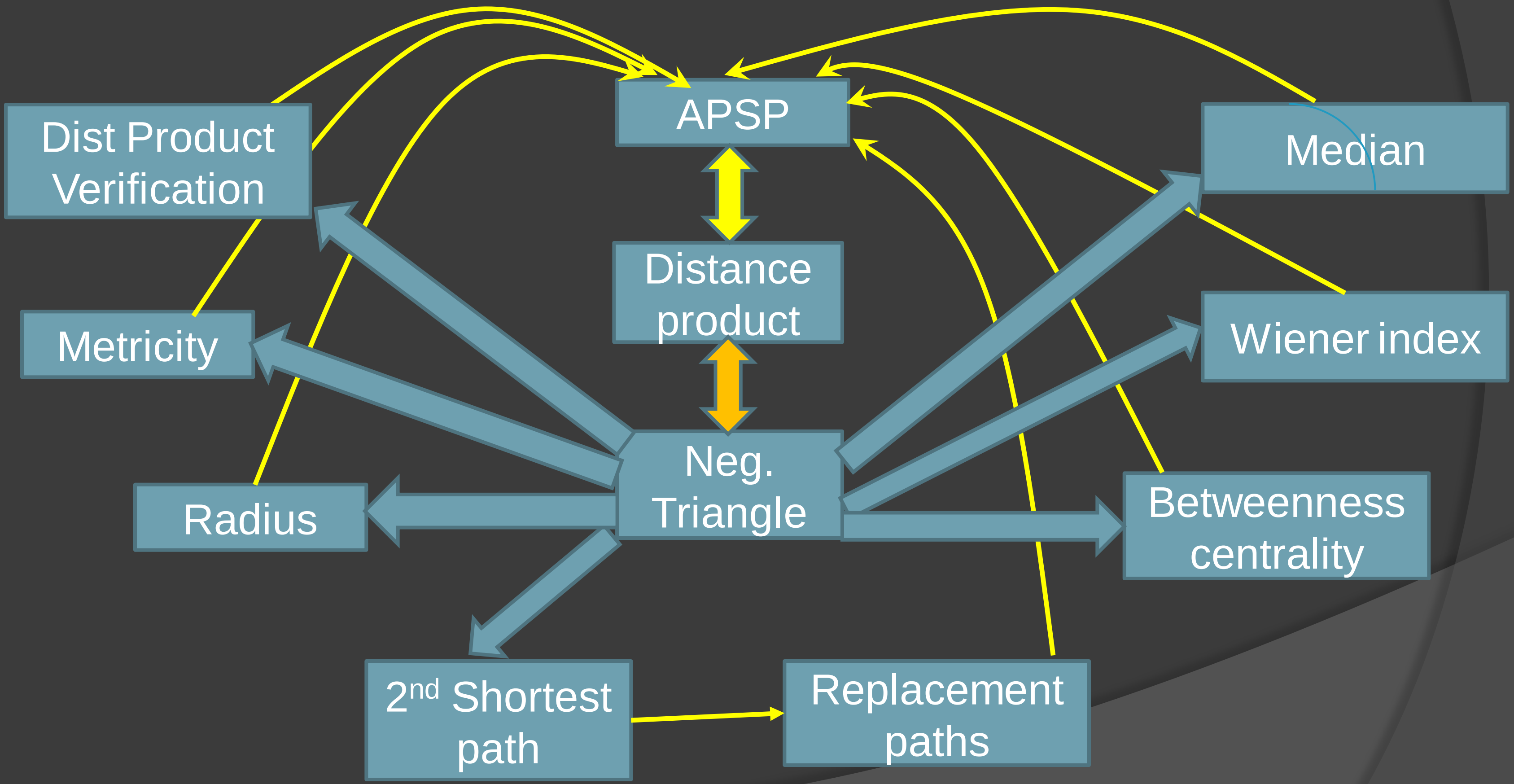
+M+W

Weights between W and 2M+W, find triangle of weight < 3M+3W. (W>>M: 4W>3M+3W)



The radius is < 3M+3W if and only if there was a negative triangle

Some known equivalences to APSP



Outline

- Hardness and equivalences for dense graphs
- Hardness for sparse graphs

Algorithms for sparse graphs

- *Eccentricities, diameter, radius, median* can be solved in $\tilde{O}(n^2)$ time in graphs with $\tilde{O}(n)$ edges, and this is the best known even for *unweighted* graphs!

- Output is a single integer, unlike APSP
- What about approximation algorithms?

Algorithms for sparse graphs

Best subquadratic time approximations:

- Diameter: 3/2-approx. in min $\{m^{3/2}, mn^{2/3}\}$ time [1]
- Radius: 3/2-approx. in min $\{m^{3/2}, mn^{2/3}\}$ time [1]
- All eccentricities: 5/3-approx. in $m^{3/2}$ time [1]
- Median: $(1+\varepsilon)$ approximation in m/poly(ε) time [2]

[1] Chechik et al.'14, [2] Indyk'99

We'll show that these approximation ratios are TIGHT for subquadratic algs (under conjectures).

Sparse graphs: conjectures

- Orthogonal vectors (OV):

given two sets U and V of n vectors in $\{0, 1\}^{O(\log n)}$, are there $u \in U$, $v \in V$ such that $u \cdot v = 0$?

OV conjecture (OVC): OV requires $n^{2-o(1)}$ time.

Theorem [W'04]: SETH implies OVC.

- Hitting set (HS):

given two sets U and V of n subsets of $[O(\log n)]$, is there some $u \in U$ such that for all $v \in V$, $u \cap v \neq \emptyset$?

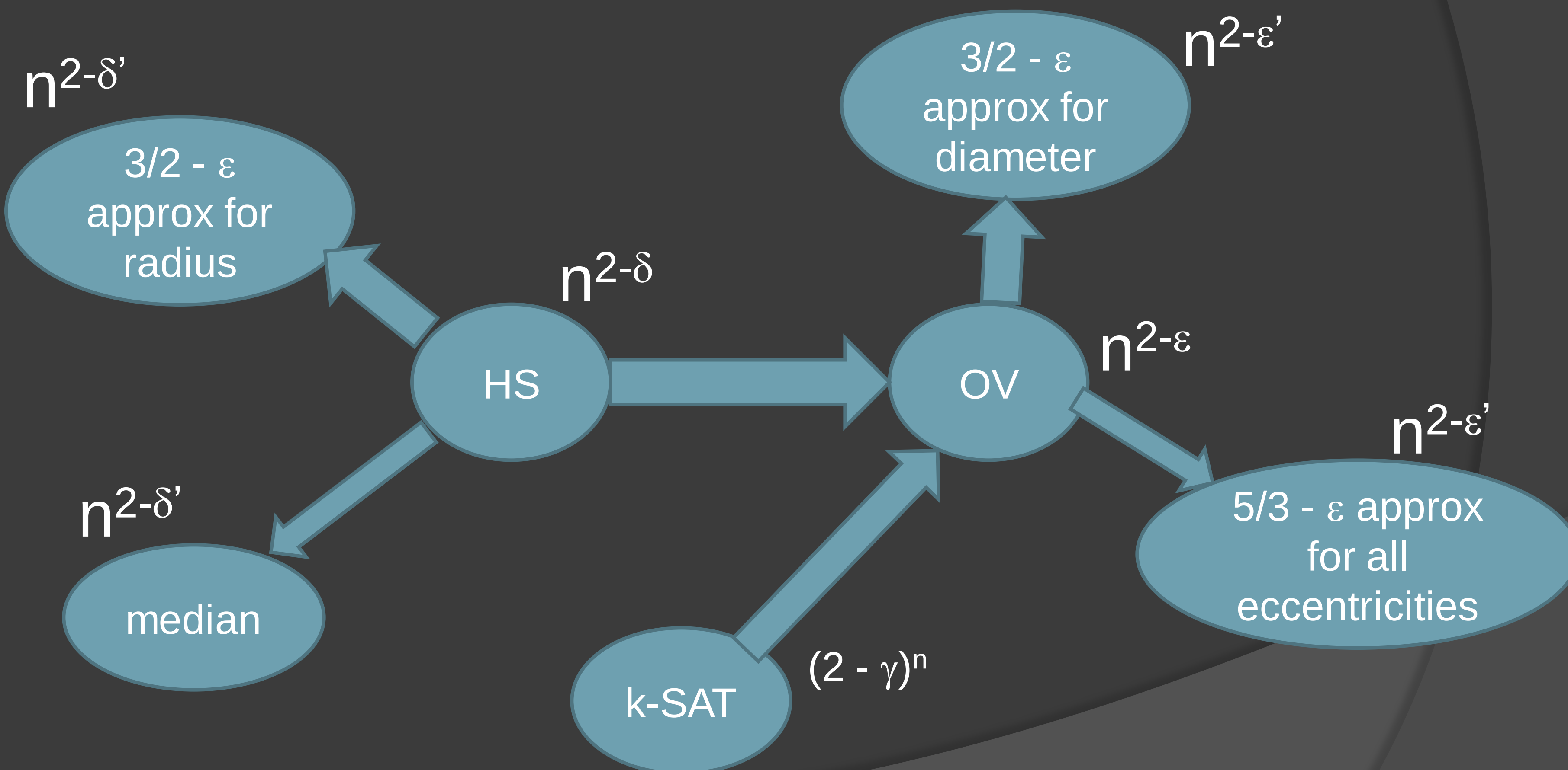
HS conjecture (HSC): HS requires $n^{2-o(1)}$ time.

Theorem [AVW'15]: HSC implies OVC.

Sparse graphs world

Best known subquadratic time approximations:

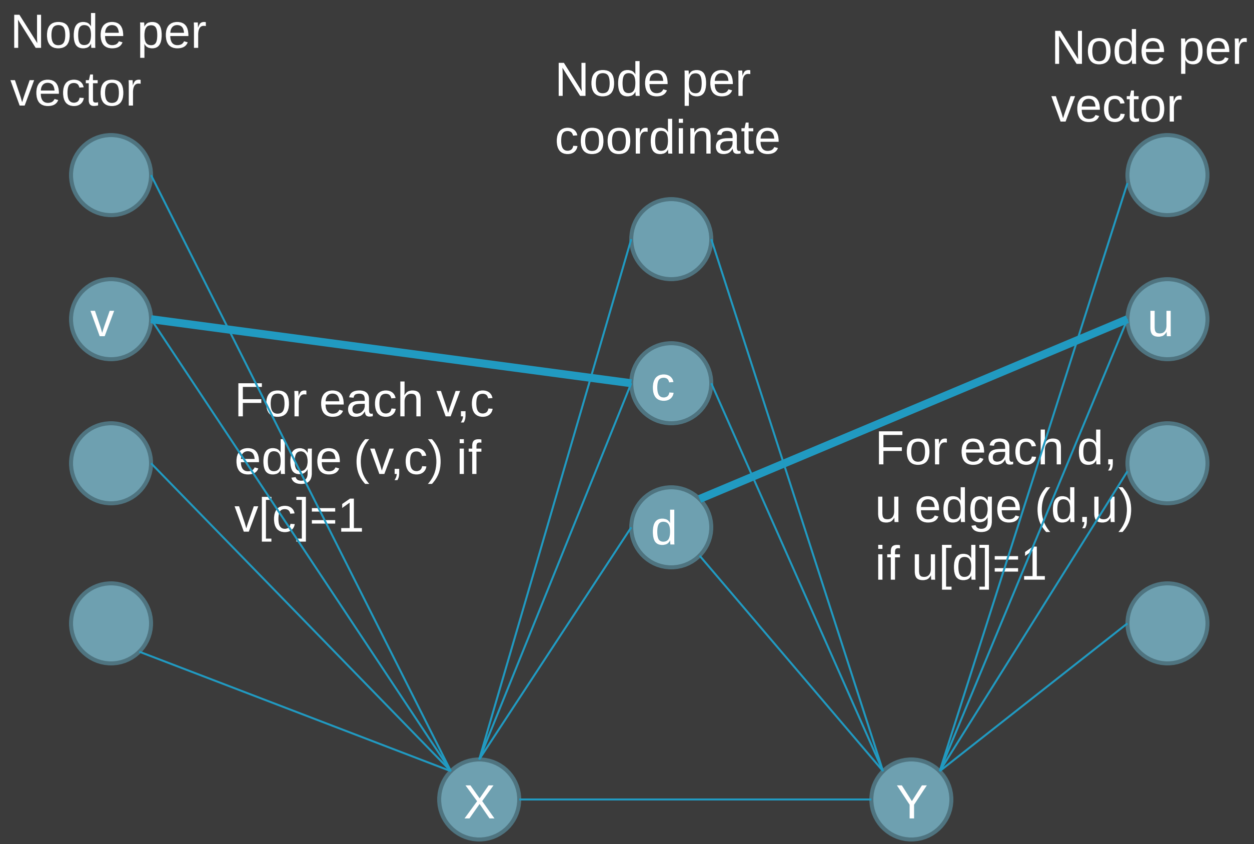
- Diameter: 3/2
- Radius: 3/2
- All eccentricities: 5/3
- Median: $(1+\varepsilon)$, any $\varepsilon > 0$



Reduce from OV on n vectors

Diameter 2 or 3 [RV'13]

Graph has O(n) nodes
m=O(n log n) edges



Diameter is 3 if exists orthogonal pair, and is 2 otherwise.

Thm: Diameter 2 or 3 in $O(m^{2-\varepsilon})$ time implies $O(n^{2-\delta})$ time for OV and hence SETH is false.

Any two vector nodes from the same side are at dist 2.

Any coordinate is at dist 2 from everyone, X and Y are at dist 2 from everyone.

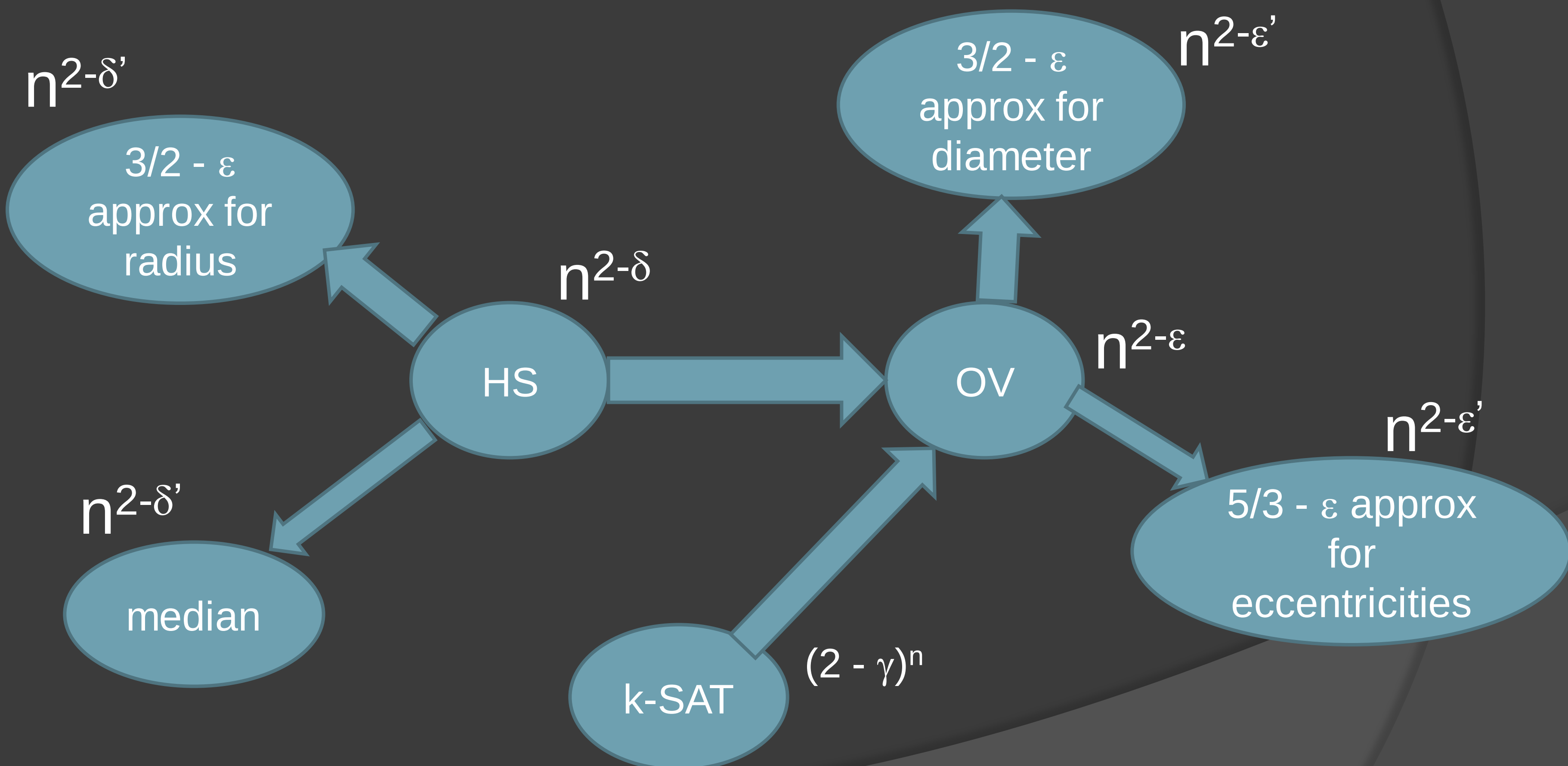
Two vectors u and v from different sides are at

dist 2 if exists a c with **u[c]=v[c]=1**, and at **dist 3** otherwise.

Sparse graphs world

Best known subquadratic time approximations:

- Diameter: 3/2
- Radius: 3/2
- All eccentricities: 5/3
- Median: $(1+\varepsilon)$, any $\varepsilon > 0$

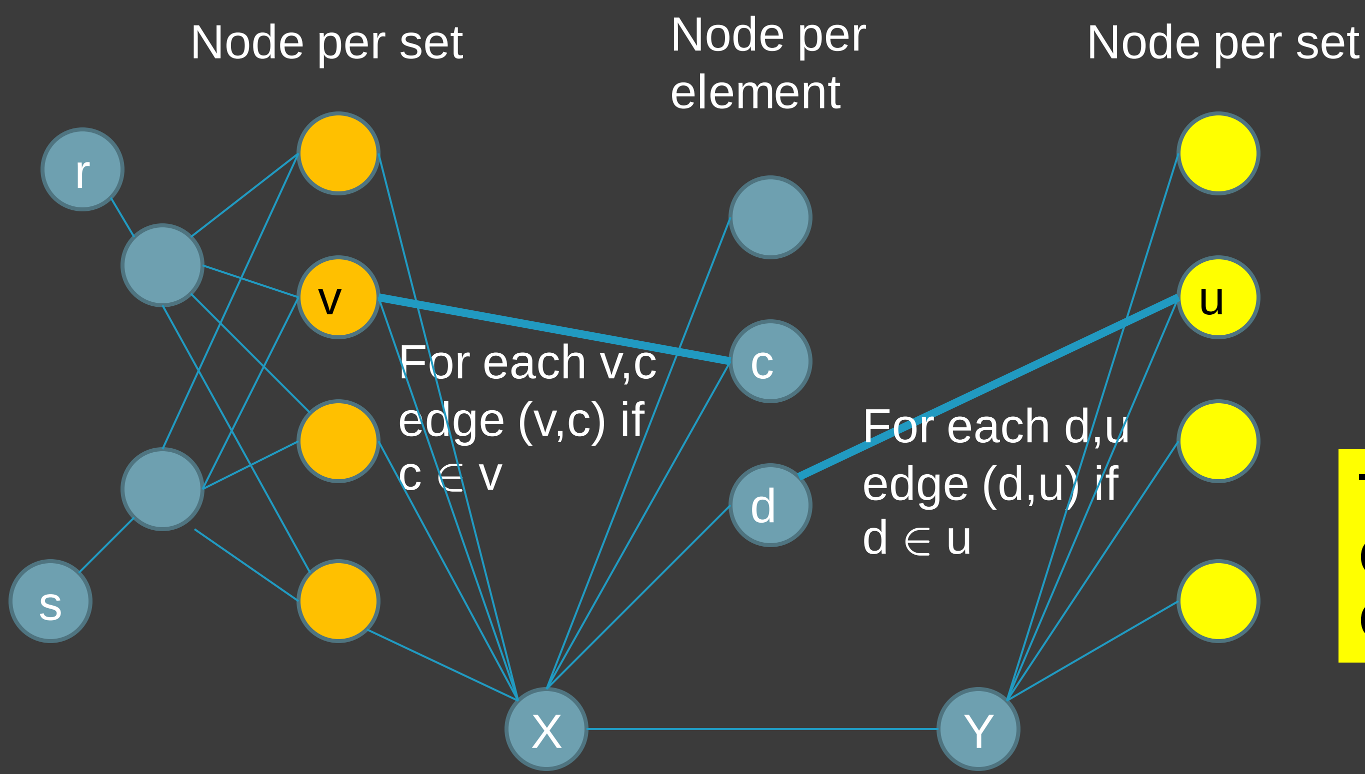


Reduce from HS on n sets

Graph has O(n) nodes
m=O(n log n) edges

Radius 2 or 3

[AVW'15]



The center node is orange, and the radius is 2 if a hitting set exists, and is 3 otherwise.

Thm: Radius 2 or 3 in $O(m^{2-\varepsilon})$ time implies $O(n^{2-\delta})$ time for HS.

Any two orange nodes are at dist 2.

Every orange node is at distance at most 2 from every non-yellow node.

Two sets u and v from different sides (one yellow, one orange) are at **dist 2** if exists a c with **c ∈ u ∩ v**, and at **dist 3** otherwise.

Every non-orange node is at distance at least 3 from r or s

Radius and diameter

- So far, any algorithm for one of radius and diameter has also been modified to work for the other
- The two problems only differ in the quantifiers: max,max vs min,max, but this also seems to make them different
- Diameter might be easier than APSP in dense graphs, or might be hard for a different reason
- Some problems are equivalent to Diameter in dense graphs (e.g. approximating betweenness centrality).

Open questions

- Any hardness for diameter in dense graphs?
- Other equivalent graph problems?
- Can we relate the sparse and dense cases to each other? E.g. does an $n^{1.9}$ time algorithm for sparse diameter imply an $n^{2.9}$ algorithm for dense diameter?
- Approximation hardness for dense graphs? The reductions do not preserve approximability.
- What about the runtimes for 3/2 approximating the diameter / radius? Are they optimal?

THANK YOU!