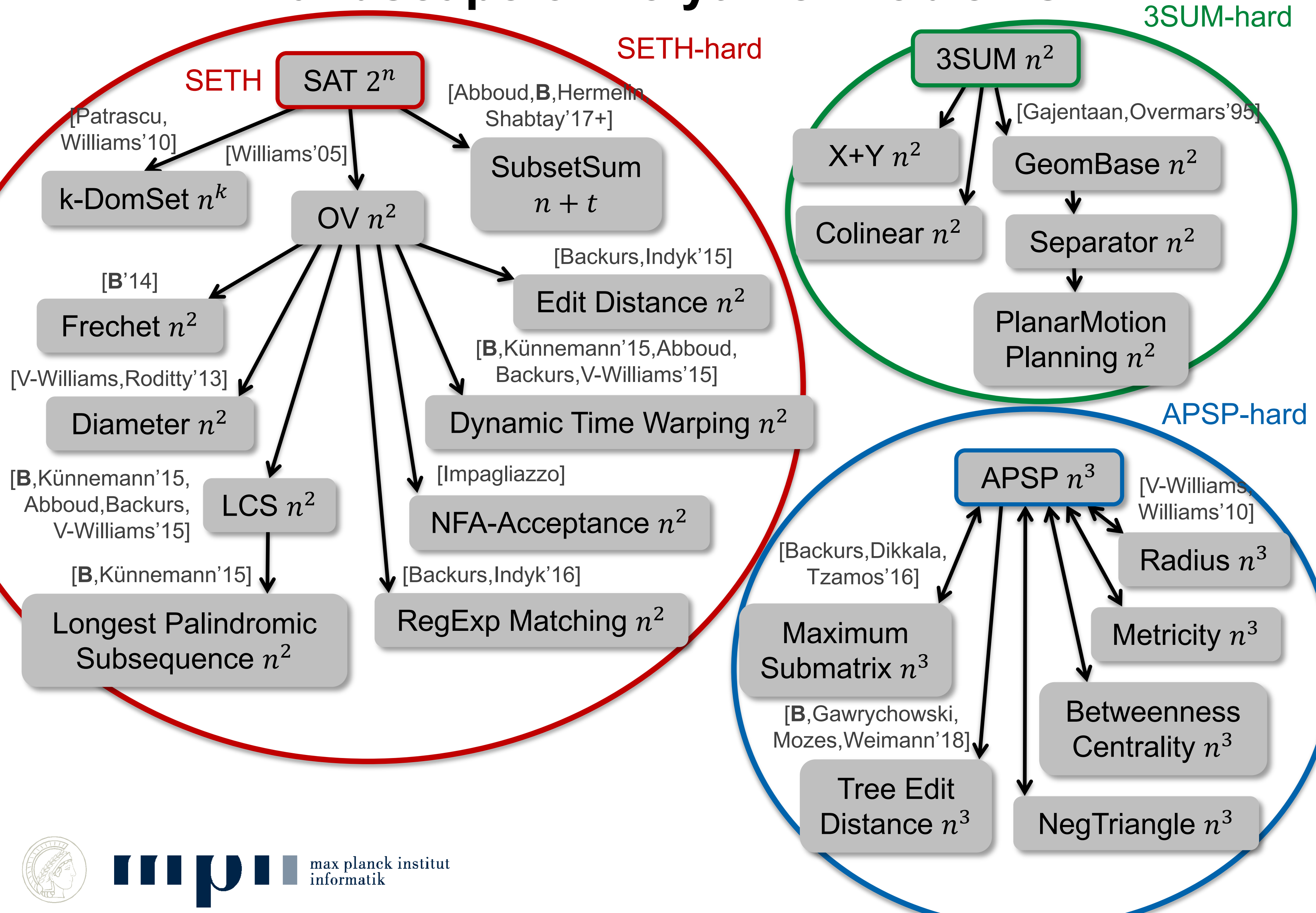
Fine-Grained Complexity - Hardness in P

Lecture 2: APSP

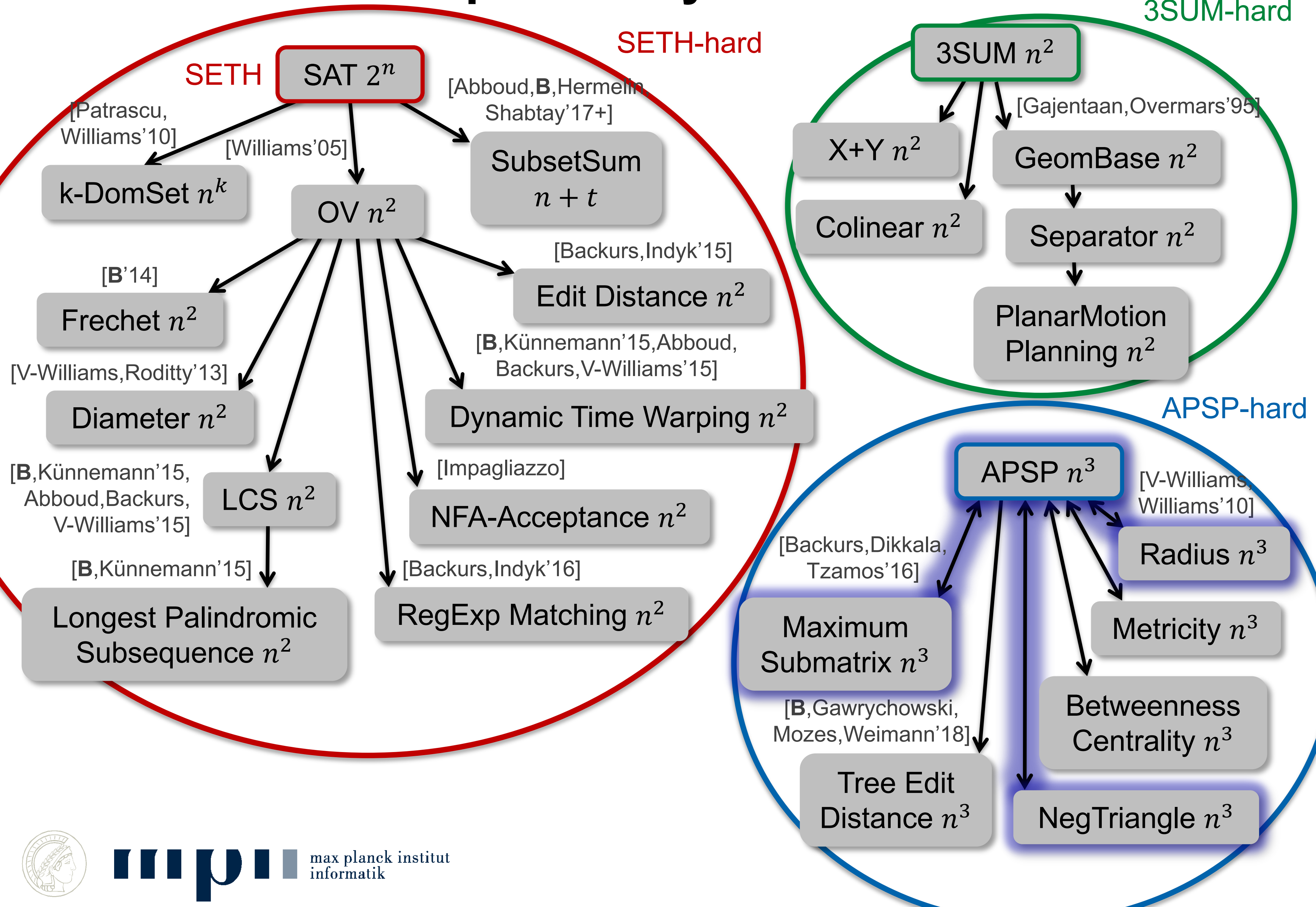
Karl Bringmann

Landscape of Polytime Problems



max planck institut
informatik

Landscape of Polytime Problems

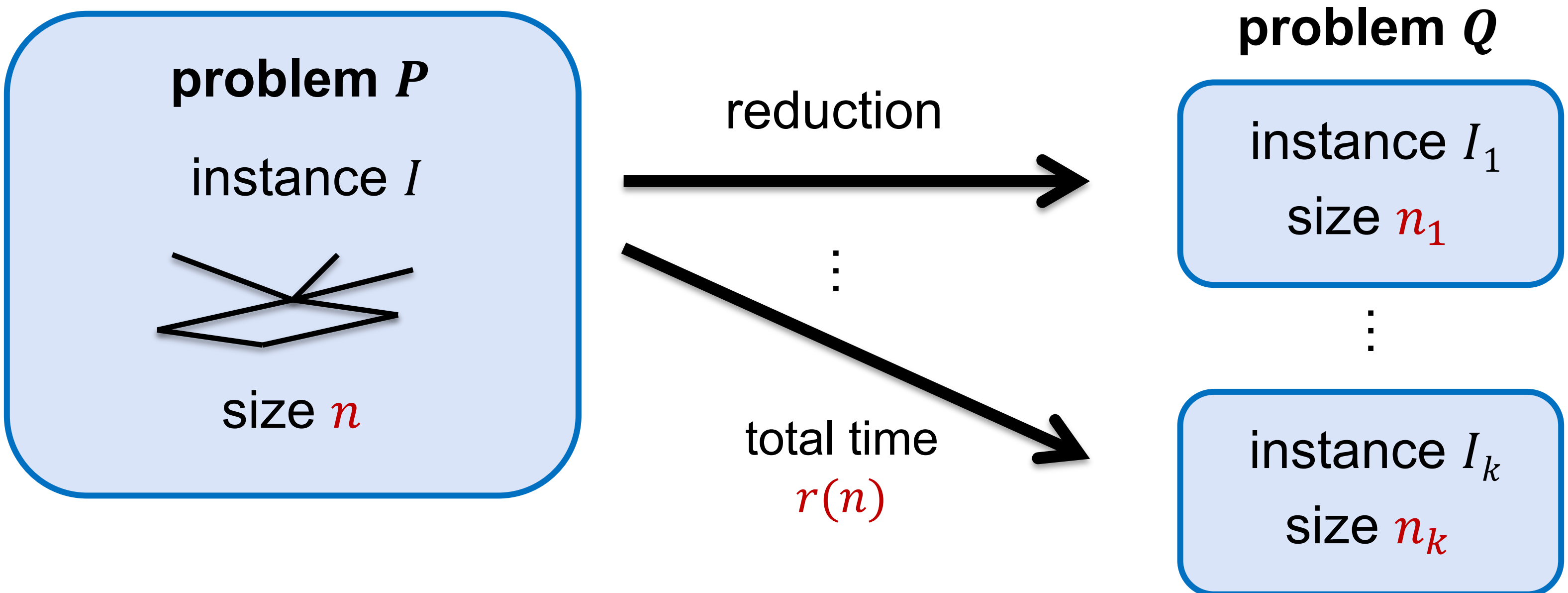


max planck institut informatik

Subcubic Reductions

A **subcubic reduction** from P to Q is

an algorithm A for P with **oracle** access to Q s.t.:



Properties:

for any instance I, algorithm $A(I)$ correctly solves problem P on I

A runs in time $r(n) = O(n^{3-\gamma})$ for some $\gamma > 0$

for any $\varepsilon > 0$ there is a $\delta > 0$ s.t. $\sum_{i=1}^{k} {n_i}^{3-\varepsilon} \leq n^{3-\delta}$

Problem Definitions

Problem All-Pairs-Shortest-Paths (APSP):

given a weighted directed graph G, compute the (length of the) **shortest path between any pair** of vertices

each edge has a weight in $\{1, .., n^c\}$

APSP-Hypothesis:

$\forall \varepsilon > 0$: APSP has no $O(n^{3-\varepsilon})$-time algorithm

there exists $c > 0$ such that

Algorithms:

$O(n^3)$ [Floyd'62,Warshall'62]

...

$O\left(n^3/2^{\Omega(\log n)^{1/2}}\right)$ [Williams'14]

Problem Definitions

Problem All-Pairs-Shortest-Paths (APSP):

given a weighted directed graph G, compute the (length of the) **shortest path between any pair** of vertices

APSP-Hypothesis:

$\forall \varepsilon > 0$: APSP has no $O(n^{3-\varepsilon})$-time algorithm

Problem Min-Plus Matrix Product:

each entry in $\{1,..,n^c,\infty\}$

given $n \times n$-matrices A, B, define their min-plus product as the $n \times n$-matrix C with

$$C_{i,j} = \min_{1 \le k \le n} A_{i,k} + B_{k,j}$$

Naive algorithm: $O(n^3)$

Subcubic Equivalences

this is surprising!

this is useful!

compute all pairwise distances in a graph

APSP — non-trivial $O(n^3)$-algorithm, output size n^2

$\Updownarrow$

compute matrix C with

$$C_{i,j} = \min_{1 \le k \le n} A_{i,k} + B_{k,j}$$

Min-Plus Product — trivial $O(n^3)$-algorithm, output size n^2

$\Updownarrow$

All-Pairs-Negative-Triangle

$\Updownarrow$

Negative Triangle — trivial $O(n^3)$-algorithm, output size 1

Problem NegTriangle:

Given a weighted directed graph G

Decide whether there are vertices i, j, k s.t.

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

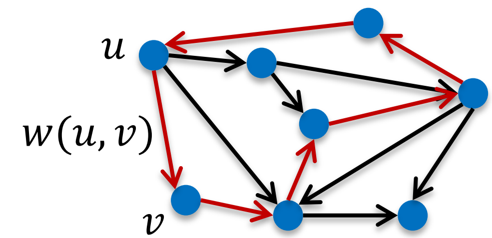
[Vassilevska-Williams,Williams'10]

[Abboud,Grandoni,Vassilevska-Williams'15]

Easy Application: Minimum Weight Cycle

Given a weighted directed graph G,

find the smallest weight of any (directed) cycle



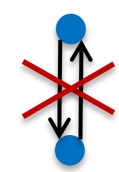
MinWeightCycle → APSP

compute all pairwise distances $d(u,v)$,

the minimum weight of any cycle is $\min_{(u,v)\in E} w(u,v) + d(v,u)$

NegTriangle → MinWeightCycle

Can assume that there are no double edges, since input graph is tripartite



Let $M \geq w(i,j)$ for all i,j

Add $10 \cdot M$ to each edge weight

Then a cycle with k edges has length in $[9 \cdot M \cdot k, 11 \cdot M \cdot k]$

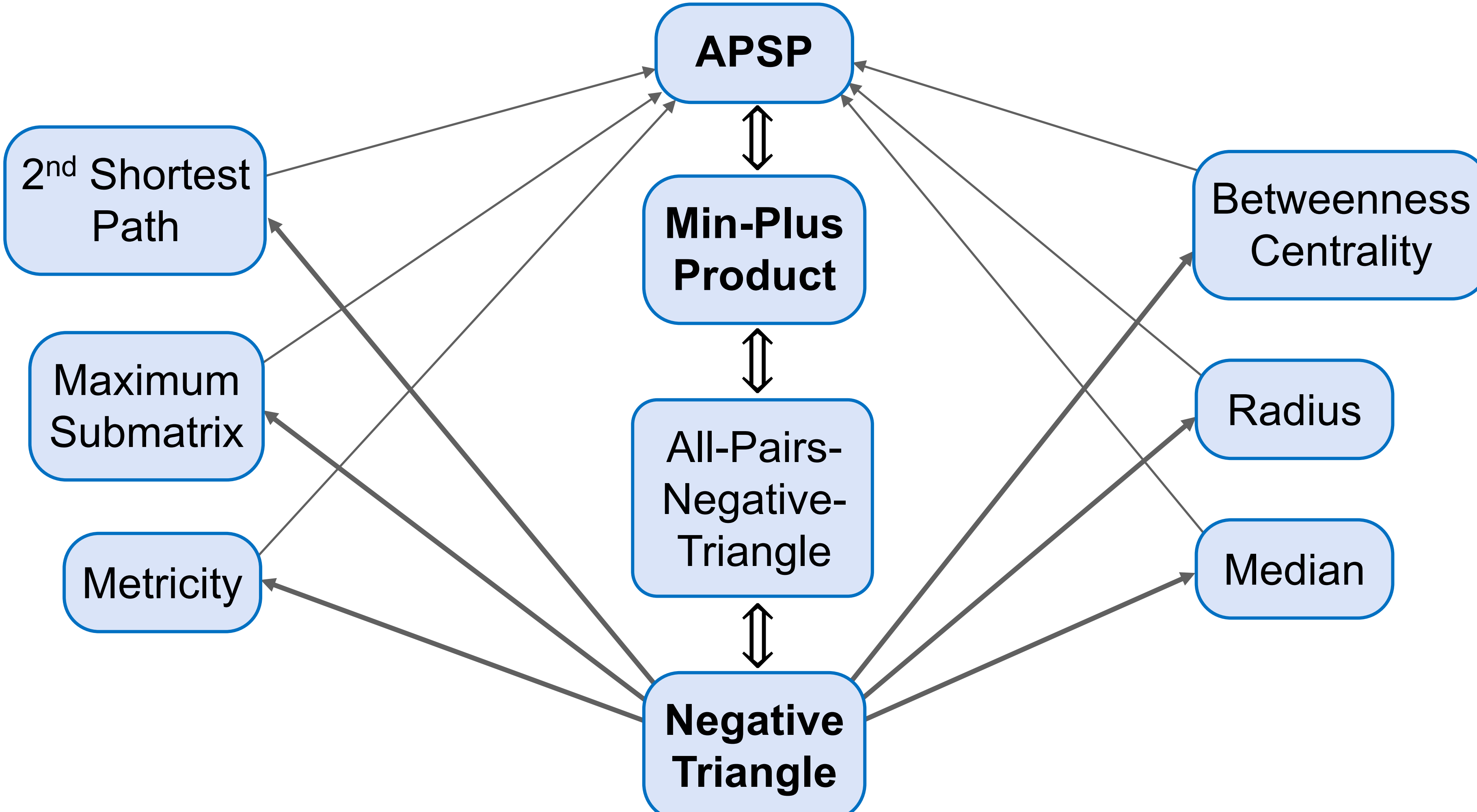
So any triangle has smaller length than any 4-cycle, 5-cycle, ...

So minimum weight of any cycle is $< 30 \cdot M$ iff there is a negative triangle

More Applications

this is surprising!

this is useful!



[Vassilevska-Williams,Williams'10]

[Abboud,Grandoni,Vassilevska-Williams'15]

max planck institut informatik

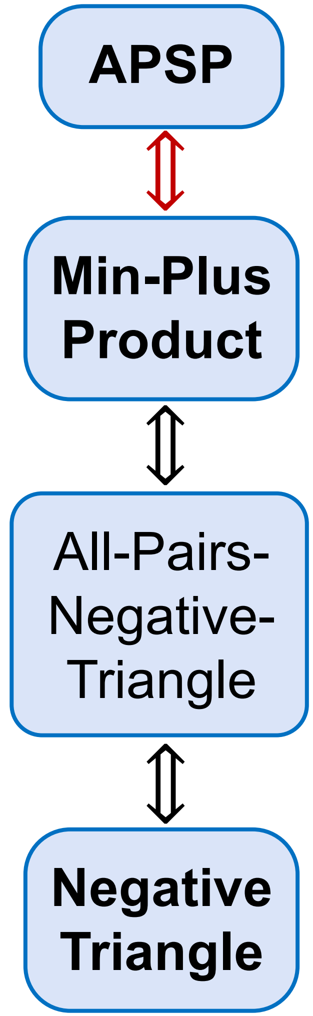
I. Equivalence of APSP and NegTriangle

II. Example Applications

III. Further Topics

IV. Conclusion

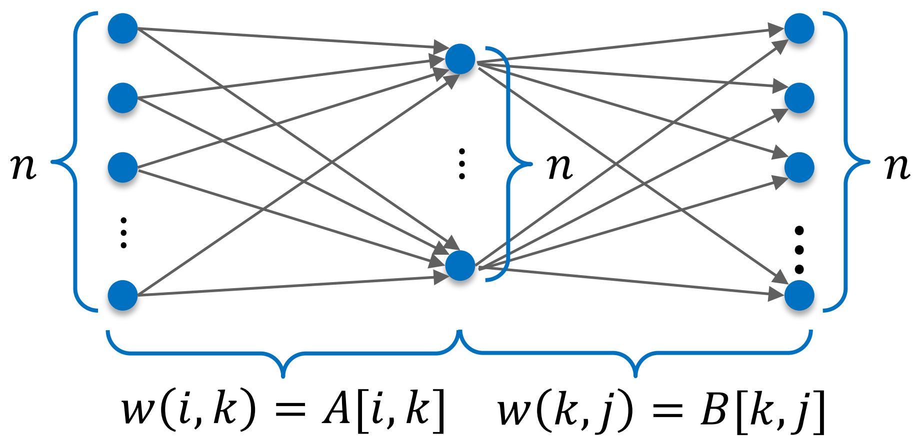
Subcubic Equivalences



APSP $\Leftrightarrow$ Min-Plus-Product

Thm: If APSP is in time $T(n)$ then Min-Plus Product is in time $O(T(n))$.

Proof: Given matrics A, B, construct graph:



„APSP is very powerful"

APSP $\Leftrightarrow$ Min-Plus-Product

Thm: If APSP is in time $T(n)$ then Min-Plus Product is in time $O\big(T(n)\big)$.

Thm: If Min-Plus Product is in time $T(n)$ then APSP is in $O(T(n)\log n)$.

Proof: Given graph G with adjacency matrix A

Add selfloops with cost 0, this yields adjacency matrix $\hat{A}$

Square $\lceil \log n \rceil$ times using Min-Plus Product:

$$B := \hat{A}^{2^{\lceil \log n \rceil}}$$

Then $B_{i,j}$ is the length of the shortest path from i to j

Property: $\left(\hat{A}^k\right)_{i,j}$ = *length of shortest path from* i *to* j *using* $\leq k$ *hops*

APSP ⇔ Min-Plus-Product

Thm: If APSP is in time $T(n)$ then Min-Plus Product is in time $O(T(n))$.

Thm: If Min-Plus Product is in time $T(n)$ then APSP is in $O(T(n)\log n)$.

APSP and Min-Plus Product are **subcubic equivalent**

Cor: APSP has an $O(n^{3-\varepsilon})$ algorithm for some $\varepsilon > 0$ if and only if Min-Plus Product has an $O(n^{3-\delta})$ algorithm for some $\delta > 0$

Cor: Min-Plus Product is in time $O\left(n^3/2^{\Omega(\log n)^{1/2}}\right)$

Subcubic Equivalences

APSP

$\Updownarrow$

Min-Plus Product

$\Updownarrow$

All-Pairs-Negative-Triangle

$\Updownarrow$

Negative Triangle

Triangle Problems

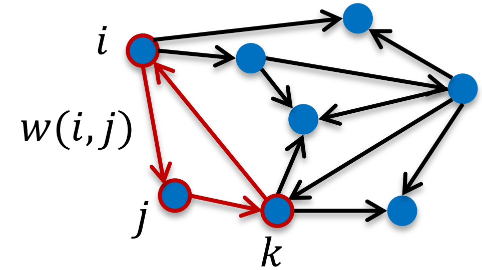
each edge has a weight in $\{-n^c, \ldots, n^c\}$

Negative Triangle

Given a weighted directed graph G

Decide whether **there are vertices $\boldsymbol{i, j, k}$** s.t.

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



Naive algorithm: $O(n^3)$

Intermediate problem:

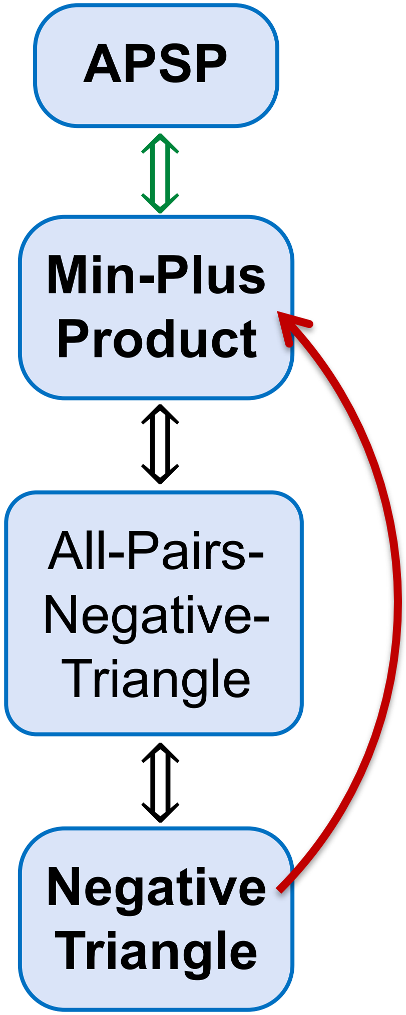
All-Pairs-Negative-Triangle

Given a weighted directed graph G with vertex set $V = I \cup J \cup K$

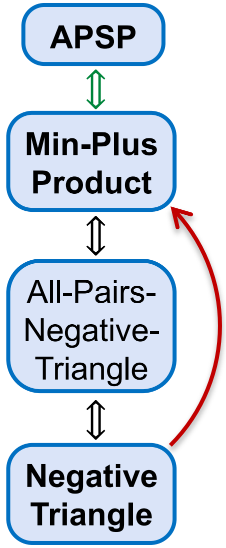
Decide **for every $\boldsymbol{i} \in \boldsymbol{I}, \boldsymbol{j} \in \boldsymbol{J}$ whether there is a vertex $\boldsymbol{k} \in \boldsymbol{K}$** s.t.

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

Subcubic Equivalences



Neg-Triangle to Min-Plus-Product



Given a weighted directed graph G on vertex set $\{1, \dots, n\}$

Adjacency matrix A:

$A_{i,j}$ = weight of edge (i, j), or ∞ if the edge does not exist

1. Compute Min-Plus Product $B \coloneqq A * A$:

$$B_{i,j} = \min_k A_{i,k} + A_{k,j}$$

2. Compute $\min_{i,j} A_{j,i} + B_{i,j}$

$$= \min_{i,j,k} A_{j,i} + A_{i,k} + A_{k,j}$$

= the smallest weight of any triangle

thus we solved Negative Triangle

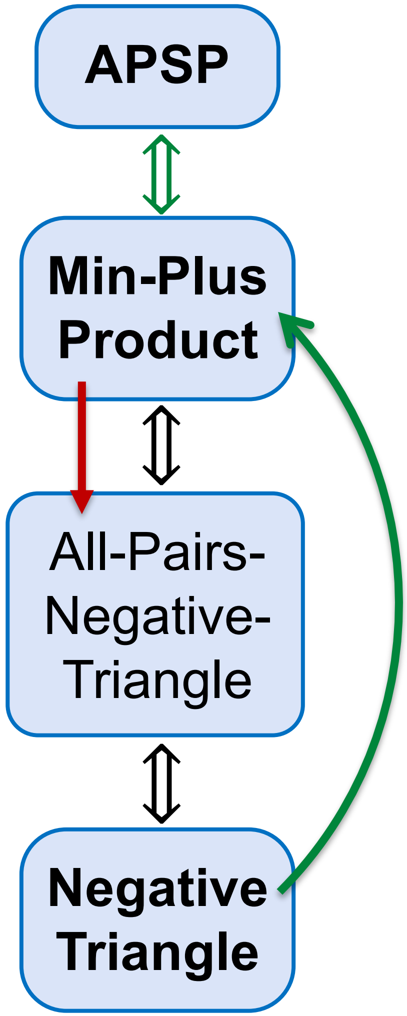
$$A: \quad \begin{matrix} 3 & 1 & \infty & \infty \\ \infty & \infty & 4 & \infty \\ 1 & 5 & \infty & 2 \\ 2 & \infty & 7 & 1 \end{matrix}$$

Running Time: $T_{\mathrm{NegTriangle}}(n) \leq T_{\mathrm{MinPlus}}(n) + O(n^2)$

→ subcubic reduction

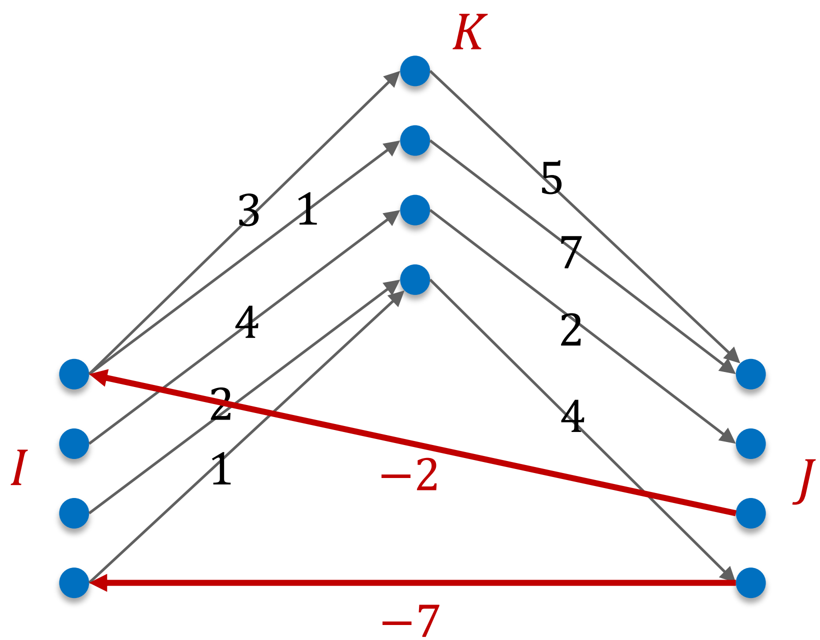
max planck institut
informatik

Subcubic Equivalences

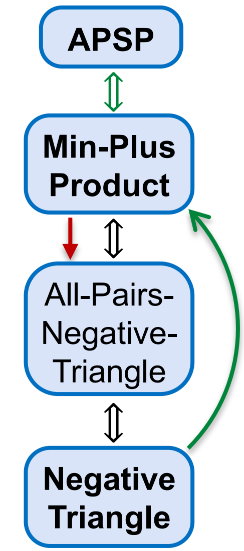


Min-Plus to All-Pairs-Neg-Triangle

$$A = \begin{matrix} 3 & 1 & \infty & \infty \\ \infty & \infty & 4 & \infty \\ \infty & \infty & \infty & 2 \\ \infty & \infty & \infty & 1 \end{matrix}$$



$$B = \begin{matrix} 5 & \infty & \infty & \infty \\ 7 & \infty & \infty & \infty \\ \infty & 2 & \infty & \infty \\ \infty & \infty & \infty & 4 \end{matrix}$$



$n = 4$ in the picture

Add all edges from J to I with (carefully chosen) weights $w(j,i)$

Run All-Pairs-Negative-Triangle algorithm

Result: for every i,j, is there a k such that $w(j,i) + w(i,k) + w(k,j) < 0$?

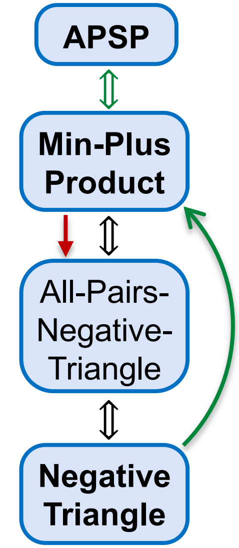
$$\Leftrightarrow w(i,k) + w(k,j) < -w(j,i)$$

WANTED: Min-Plus: for every i,j: $\min_k w(i,k) + w(k,j)$

= minimum number z s.t. there is a k s.t. $w(i,k) + w(k,j) < z + 1$

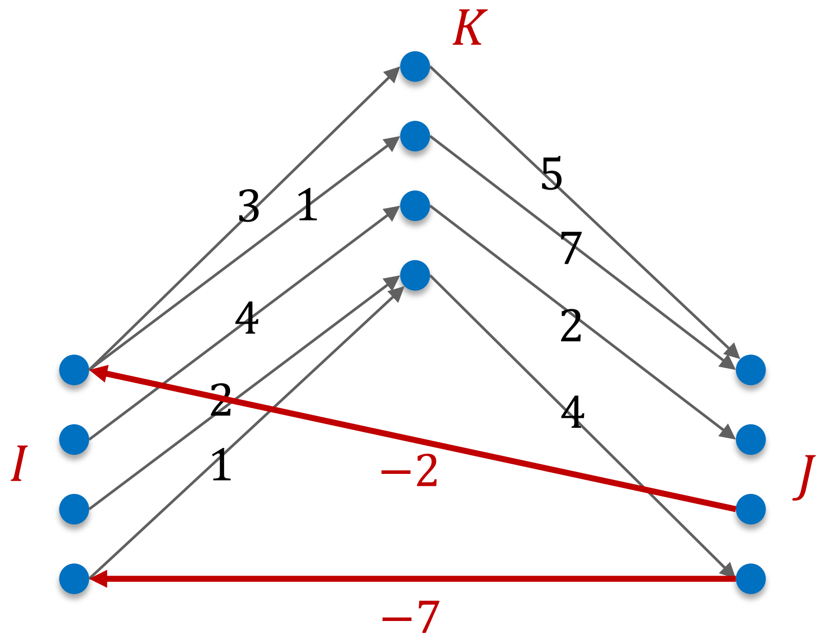
binary search via $w(j,i)$! **simultaneous** for all i,j!

Min-Plus to All-Pairs-Neg-Triangle



$$\begin{matrix} 3 & 1 & \infty & \infty \\ \infty & \infty & 4 & \infty \\ \infty & \infty & \infty & 2 \\ \infty & \infty & \infty & 1 \end{matrix}$$

A



$$\begin{matrix} 5 & \infty & \infty & \infty \\ 7 & \infty & \infty & \infty \\ \infty & 2 & \infty & \infty \\ \infty & \infty & \infty & 4 \end{matrix}$$

B

$n = 4$ in the picture

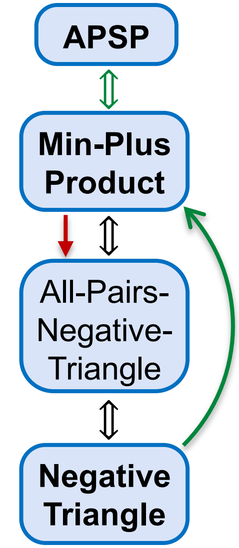
binary search via $w(j, i)$! **simultaneous** for all i, j!

need that all (finite) weights are in $\{-n^c, \dots, n^c\}$

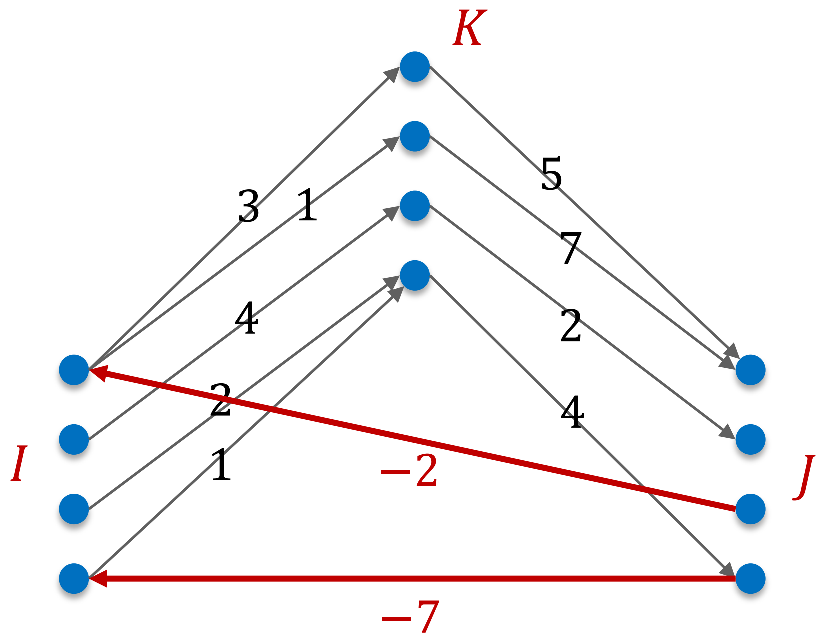
each entry of Min-Plus Product is in $\{-2n^c, \dots, 2n^c, \infty\}$

binary search takes $\log_2(4n^c + 1) = O(\log n)$ steps

Min-Plus to All-Pairs-Neg-Triangle



$$A = \begin{matrix} 3 & 1 & \infty & \infty \\ \infty & \infty & 4 & \infty \\ \infty & \infty & \infty & 2 \\ \infty & \infty & \infty & 1 \end{matrix}$$



$$B = \begin{matrix} 5 & \infty & \infty & \infty \\ 7 & \infty & \infty & \infty \\ \infty & 2 & \infty & \infty \\ \infty & \infty & \infty & 4 \end{matrix}$$

$n = 4$ in the picture

binary search via $w(j,i)$! **simultaneous** for all i,j!

for all i,j: initialize $m(i,j) := -2n^c$ and $M(i,j) := 2n^c$

repeat $\log(4n^c)$ times:

for all i,j: set $w(j,i) := -\lceil (m(i,j) + M(i,j))/2 \rceil$

compute All-Pairs-Negative-Triangle

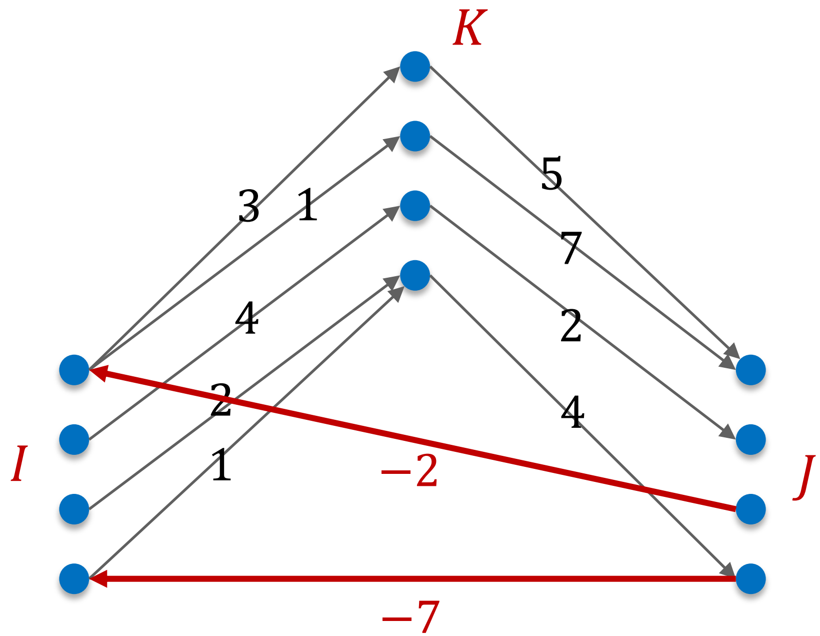
for all i,j: if i,j is in negative triangle: $M(i,j) := -w(j,i) - 1$

otherwise: $m(i,j) := -w(j,i)$

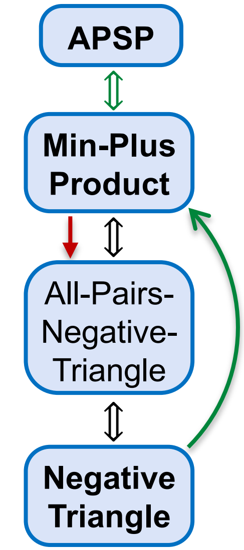
(missing: handling of ∞)

Min-Plus to All-Pairs-Neg-Triangle

$$A = \begin{matrix} 3 & 1 & \infty & \infty \\ \infty & \infty & 4 & \infty \\ \infty & \infty & \infty & 2 \\ \infty & \infty & \infty & 1 \end{matrix}$$



$$B = \begin{matrix} 5 & \infty & \infty & \infty \\ 7 & \infty & \infty & \infty \\ \infty & 2 & \infty & \infty \\ \infty & \infty & \infty & 4 \end{matrix}$$



$n = 4$ in the picture

binary search takes $\log_2(4n^c + 1) = O(\log n)$ steps

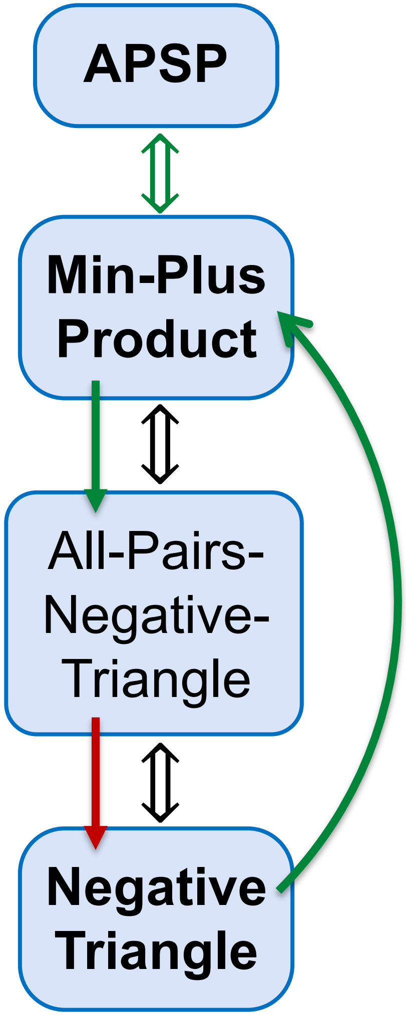
$T(n)$ algorithm for All-Pairs-Neg-Triangle yields
$O(T(n) \log n)$ algorithm for Min-Plus Product

In particular: $O(n^{3-\varepsilon})$ algorithm for All-Pairs-Neg-Triangle for some $\varepsilon > 0$ implies $O(n^{3-\varepsilon})$ algorithm for Min-Plus Product for some $\varepsilon > 0$

max planck institut informatik

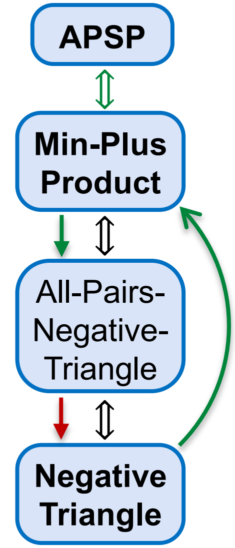
$\rightarrow$ subcubic reduction

Subcubic Equivalences



max planck institut
informatik

All-Pairs-Neg-Triangle to Neg-Triangle



Negative Triangle Given graph G

Decide whether there are vertices i, j, k such that

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

All-Pairs-Negative-Triangle Given graph G with vertex set $V = I \cup J \cup K$

Decide for every $i \in I, j \in J$ whether there is a vertex $k \in K$ such that

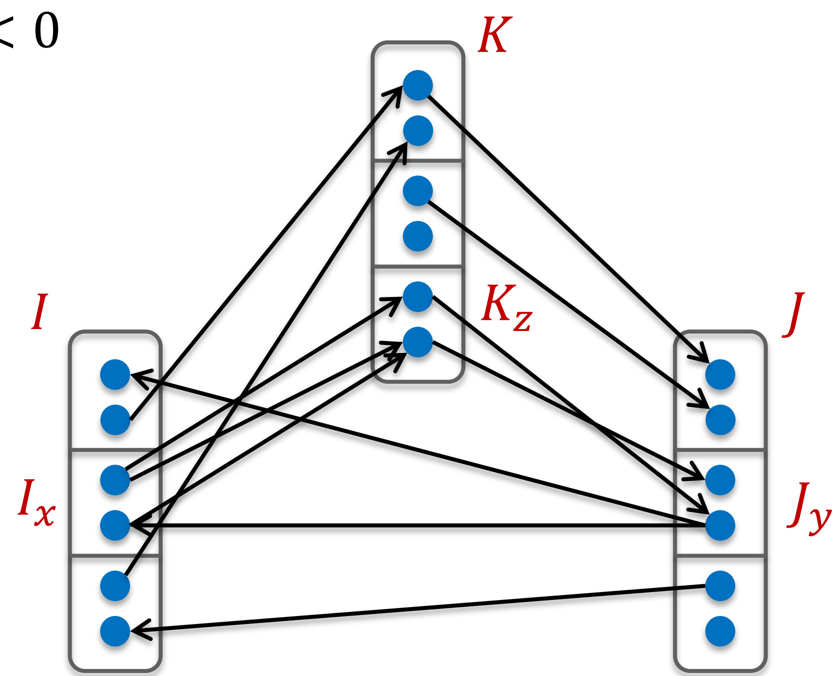
$$w(j,i) + w(i,k) + w(k,j) < 0$$

Split I, J, K into n/s parts of size s:

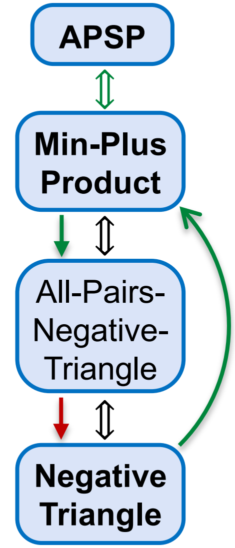
$$I_1, \dots, I_{n/s}, J_1, \dots, J_{n/s}, K_1, \dots, K_{n/s}$$

For each of the $(n/s)^3$ triples (I_x, J_y, K_z):

consider graph $G[I_x \cup J_y \cup K_z]$



All-Pairs-Neg-Triangle to Neg-Triangle



Initialize C as $n \times n$ all-zeroes matrix

For each of the $(n/s)^3$ triples of parts (I_x, J_y, K_z):

While $G[I_x \cup J_y \cup K_z]$ contains a negative triangle:

Find a negative triangle (i, j, k) in $G[I_x \cup J_y \cup K_z]$

Set $C[i, j] := 1$

Set $w(i, j) := \infty$

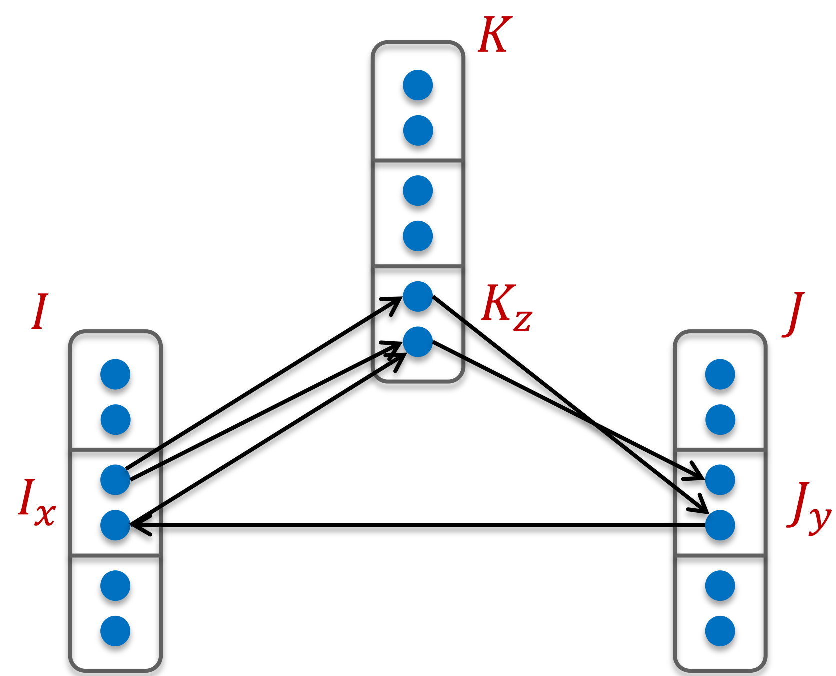
(i, j) is in no more negative triangles

✔ guaranteed termination:

can set $\leq n^2$ weights to ∞

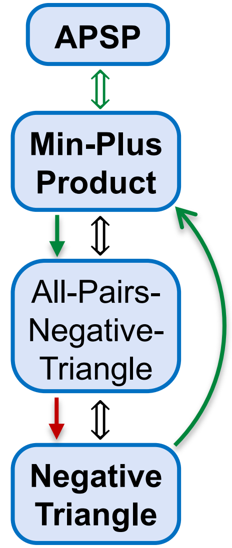
✔ correctness:

if (i, j) is in negative triangle, we will find one



max planck institut informatik

All-Pairs-Neg-Triangle to Neg-Triangle



Find a negative triangle (i, j, k) in $G[I_x \cup J_y \cup K_z]$

*How to **find** a negative triangle*
*if we can only **decide** whether one exists?*

Partition I_x into ${I_x}^{(1)}, {I_x}^{(2)}$, J_y into ${J_y}^{(1)}, {J_y}^{(2)}$, K_z into ${K_z}^{(1)}, {K_z}^{(2)}$

Since $G[I_x \cup J_y \cup K_z]$ contains a negative triangle,

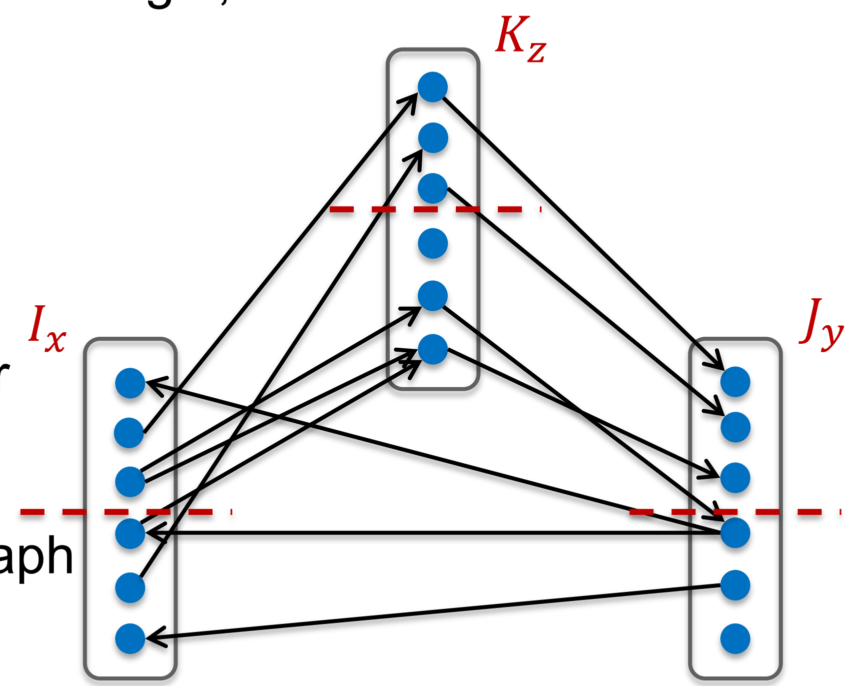
at least one of the 2^3 subgraphs

$$G[{I_x}^{(a)} \cup {J_y}^{(b)} \cup {K_z}^{(c)}]$$

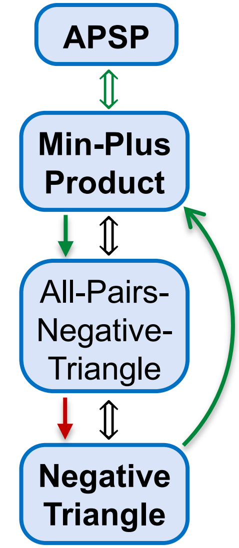
contains a negative triangle

Decide for each such subgraph whether it contains a negative triangle

Recursively find a triangle in one subgraph



All-Pairs-Neg-Triangle to Neg-Triangle



Find a negative triangle (i, j, k) in $G[I_x \cup J_y \cup K_z]$

*How to **find** a negative triangle*
*if we can only **decide** whether one exists?*

Partition I_x into ${I_x}^{(1)}, {I_x}^{(2)}$, J_y into ${J_y}^{(1)}, {J_y}^{(2)}$, K_z into ${K_z}^{(1)}, {K_z}^{(2)}$

Since $G[I_x \cup J_y \cup K_z]$ contains a negative triangle,

at least one of the 2^3 subgraphs

$$G[{I_x}^{(a)} \cup {J_y}^{(b)} \cup {K_z}^{(c)}]$$

contains a negative triangle

Decide for each such subgraph whether it contains a negative triangle

Recursively find a triangle in one subgraph

Running Time:

$T_{\text{FindNegTriangle}}(n) \leq$

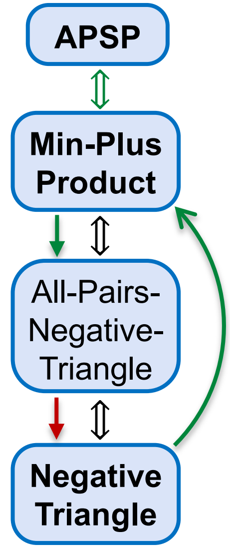
$$2^3 \cdot T_{\text{DecideNegTriangle}}(n)$$

$$+ \, T_{\text{FindNegTriangle}}(n/2)$$

$$= O(T_{\text{DecideNegTriangle}}(n))$$

max planck institut informatik

All-Pairs-Neg-Triangle to Neg-Triangle



Initialize C as $n \times n$ all-zeroes matrix

For each of the $(n/s)^3$ triples of parts (I_x, J_y, K_z):

  While $G[I_x \cup J_y \cup K_z]$ contains a negative triangle:

    Find a negative triangle (i, j, k) in $G[I_x \cup J_y \cup K_z]$

    Set $C[i, j] \coloneqq 1$

    Set $w(i, j) \coloneqq \infty$

$(*)$

Running Time:

$$(*) = O(T_{\text{FindNegTriangle}}(s)) = O(T_{\text{DecideNegTriangle}}(s))$$

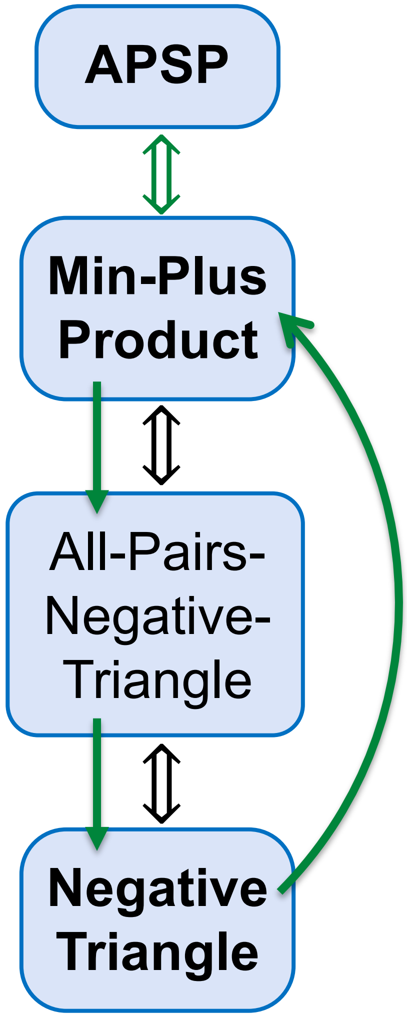
Total time: $\big((\#\text{triples}) + (\#\text{triangles found})\big) \cdot (*)$

$$\leq \big((n/s)^3 + n^2\big) \cdot T_{\text{DecideNegTriangle}}(s)$$

Set $s = n^{1/3}$ and assume $T_{\text{DecideNegTriangle}}(n) = O(n^{3-\varepsilon})$

Total time: $O\big(n^2 \cdot n^{1-\varepsilon/3}\big) = O(n^{3-\varepsilon/3})$

Subcubic Equivalences



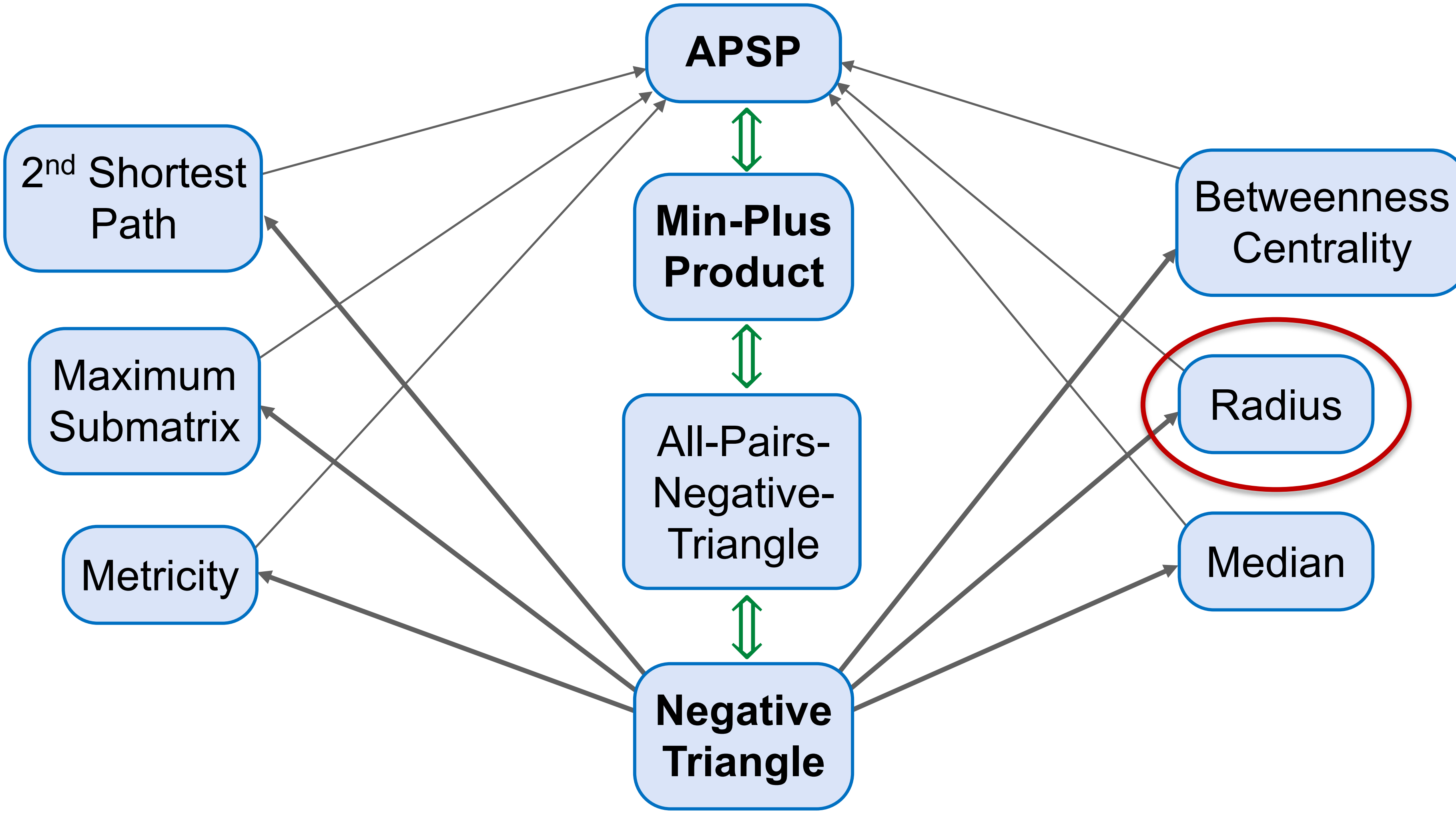
I. Equivalence of APSP and NegTriangle

II. Example Applications

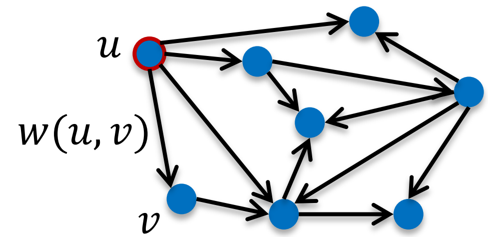
III. Further Topics

IV. Conclusion

Subcubic Equivalences



Radius

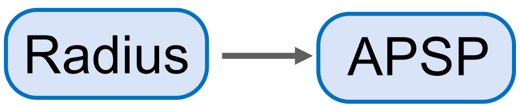


G is a weighted directed graph

$d(u,v)$ is the distance from u to v in G

Radius: $\min_u \max_v d(u,v)$

u is in some sense the *most central vertex*



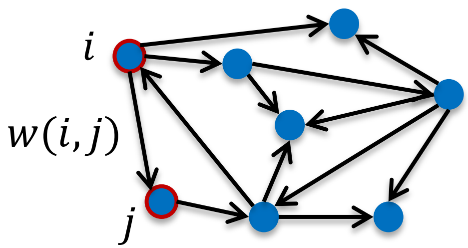
compute all pairwise distances,
then evaluate definition of radius in time $O(n^2)$

→ subcubic reduction

⟹ Radius is in time $O\left(n^3/2^{\Omega(\log n)^{1/2}}\right)$

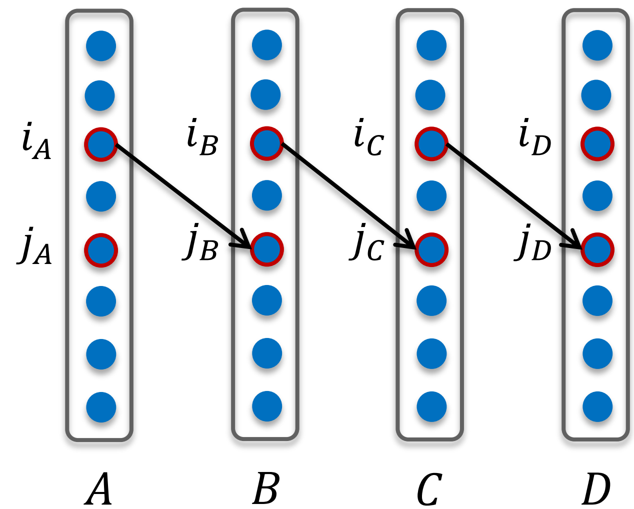
Negative Triangle to Radius

Negative Triangle instance:
graph G with n nodes,
edge-weights in $\{-n^c, \dots, n^c\}$



Radius instance:
graph H with $O(n)$ nodes,
edge-weights in $\{0, \dots, O(n^c)\}$

$M := 3n^c$

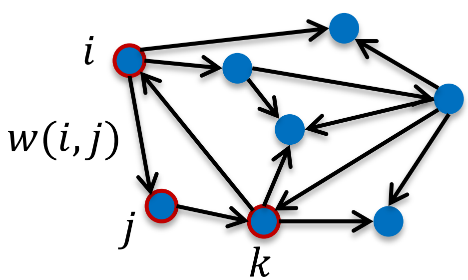


1) Make four layers with n nodes
2) For any edge (i,j): Add (i_A, j_B), (i_B, j_C),(i_C, j_D) with weight $M + w(i,j)$

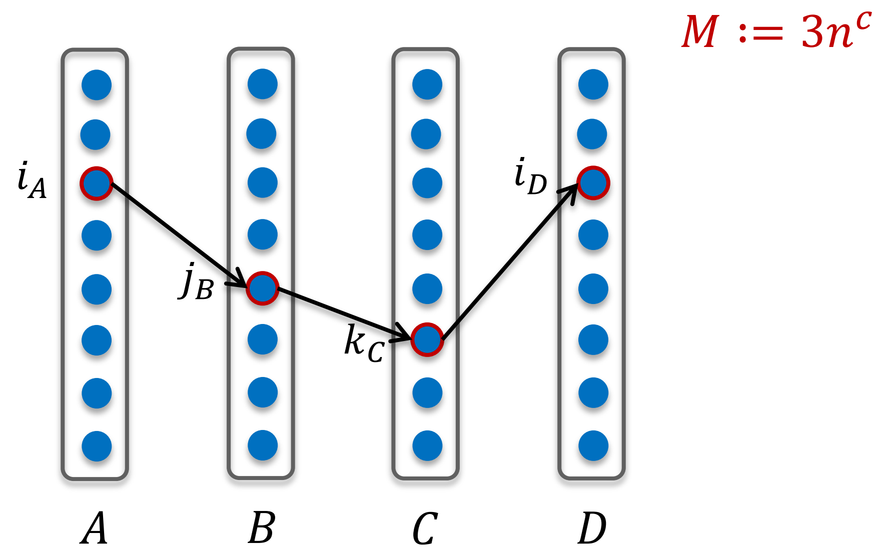
Negative Triangle to Radius

Negative Triangle instance:
graph G with n nodes,
edge-weights in $\{-n^c, \ldots, n^c\}$

Radius instance:
graph H with $O(n)$ nodes,
edge-weights in $\{0, \ldots, O(n^c)\}$



(i, j, k) has weight W



1) Make four layers with n nodes
2) For any edge (i, j): Add (i_A, j_B), (i_B, j_C),(i_C, j_D) with weight $M + w(i, j)$

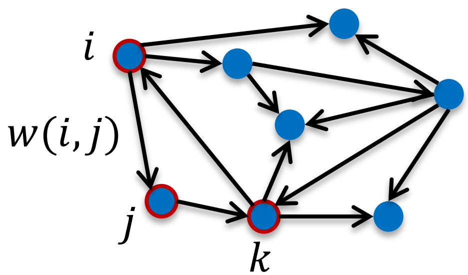
$\Leftrightarrow$ path has length $3M + W$

$\rightarrow$ $\exists i_A, j_B, k_C, i_D$-path of length $\leq 3M - 1$?

Negative Triangle to Radius

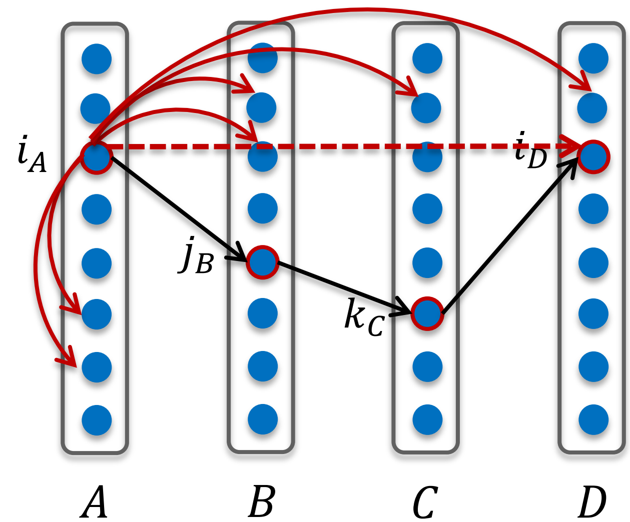
Negative Triangle instance:
graph G with n nodes,
edge-weights in $\{-n^c, \dots, n^c\}$

Radius instance:
graph H with $O(n)$ nodes,
edge-weights in $\{0, \dots, O(n^c)\}$



(i,j,k) has weight W

$M := 3n^c$



1) Make four layers with n nodes
2) For any edge (i,j): Add (i_A, j_B), (i_B, j_C),(i_C, j_D) with weight $M + w(i,j)$
3) Add edges of weight $3M - 1$ from any i_A to all nodes except i_D

$\Leftrightarrow$ path has length $3M + W$

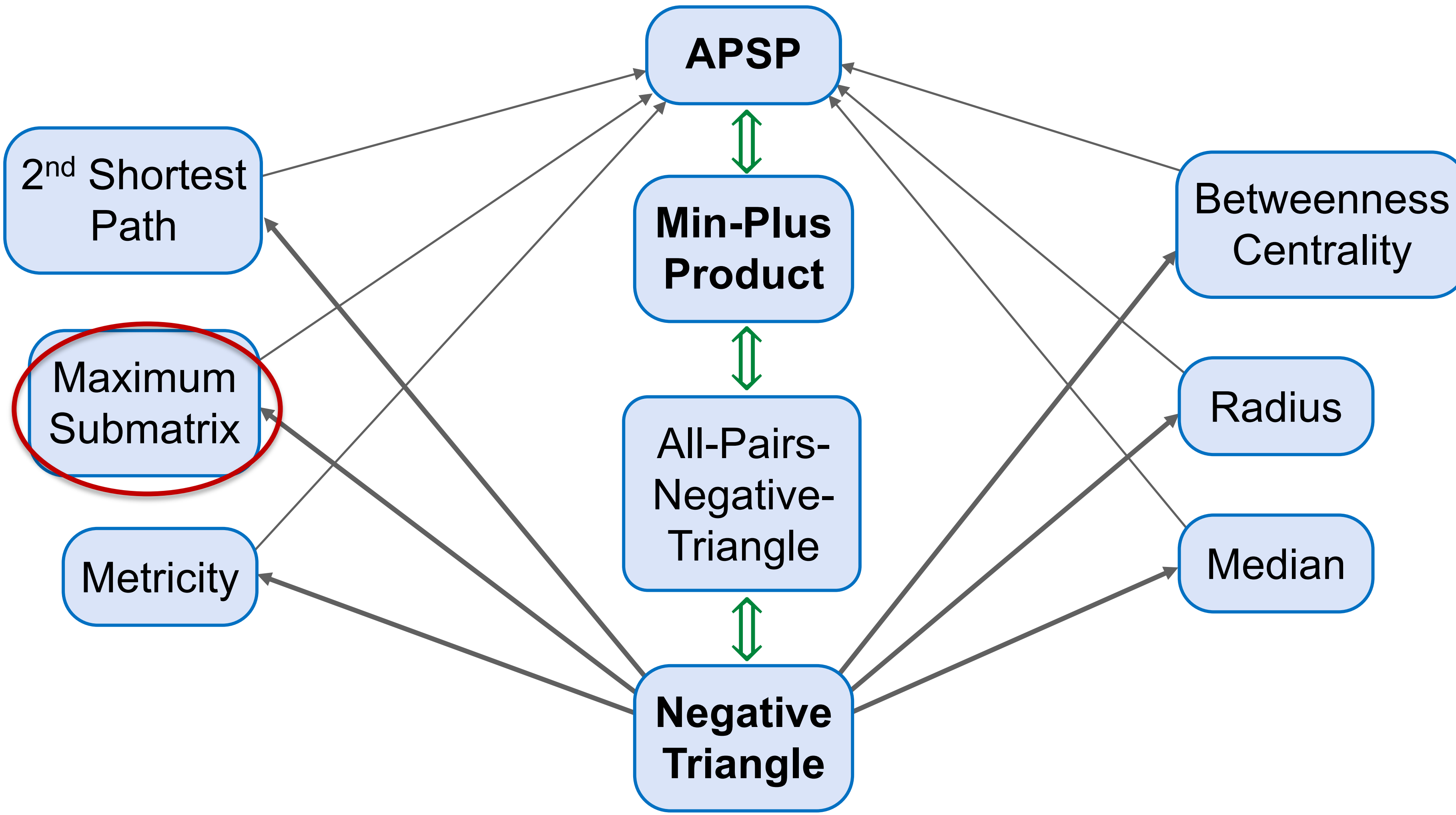
$\rightarrow$ $\exists i_A, j_B, k_C, i_D$-path of length $\leq 3M - 1$?

Radius: $\min_u \max_v d(u,v)$

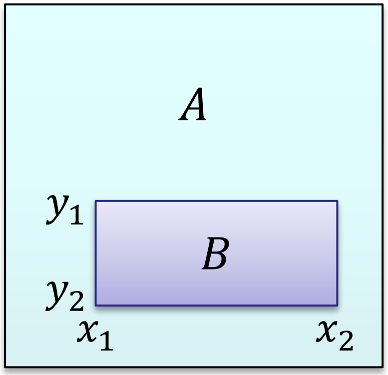
Claim: Radius of H is $\leq 3M - 1$ iff there is a negative triangle in G

max planck institut
informatik

Subcubic Equivalences



MaxSubmatrix



MaxSubmatrix:

given an $n \times n$ matrix A with entries in $\{-n^c, .., n^c\}$

$\Sigma(B) :=$ **sum of all entries** of matrix B

compute maximum $\Sigma(B)$ over all **submatrices** B of A

Thm: MaxSubmatrix is subcubic equivalent to APSP

[Tamaki,Tokuyama'98]
[Backurs,Dikkala,Tzamos'16]

there are $O(n^4)$ possible submatrices B

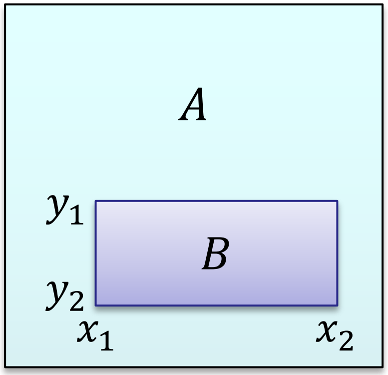
computing $\Sigma(B)$: $O(n^2)$

trivial running time: $O(n^6)$

Exercise: design an $O(n^3)$ algorithm

max planck institut
informatik

MaxSubmatrix



MaxSubmatrix:

given an $n \times n$ matrix A with entries in $\{-n^c, .., n^c\}$

$\Sigma(B) :=$ **sum of all entries** of matrix B

compute maximum $\Sigma(B)$ over all **submatrices** B of A

Thm: MaxSubmatrix is subcubic equivalent to APSP

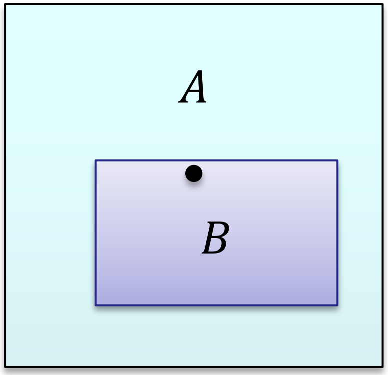
[Tamaki,Tokuyama'98]
[Backurs,Dikkala,Tzamos'16]

MaxCenteredSubmatrix:

compute maximum $\Sigma(B)$ over all **submatrices** B of A **containing the center** of A

i.e. we require $x_1 \leq n/2 < x_2$ and $y_1 \leq n/2 < y_2$

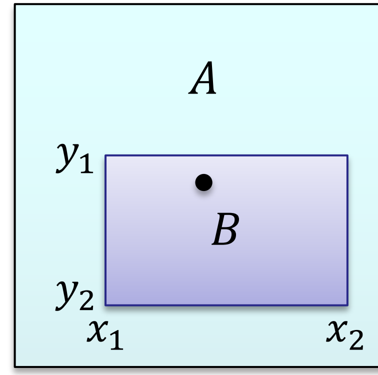
Thm: MaxCenteredSubmatrix is subcubic equ. to APSP



we only prove: NegativeTriangle $\leq$ MaxCenteredSubmatrix

Exercise: MaxCenteredSubmatrix $\leq$ APSP

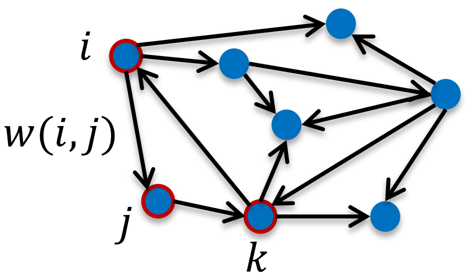
NegTriangle to MaxCentSubmatrix



***Positive* Triangle** instance:
graph G with n nodes,
edge-weights in $\{-n^c, \dots, n^c\}$

→

MaxCenteredSubmatrix:
$2n \times 2n$-matrix A
entries in $\{-n^{O(c)}, \dots, n^{O(c)}\}$



Claim: MaxCentSubmatrix of A is $> M$
iff G has a ***positive*** triangle

$M \coloneqq 2n^{c+3}$

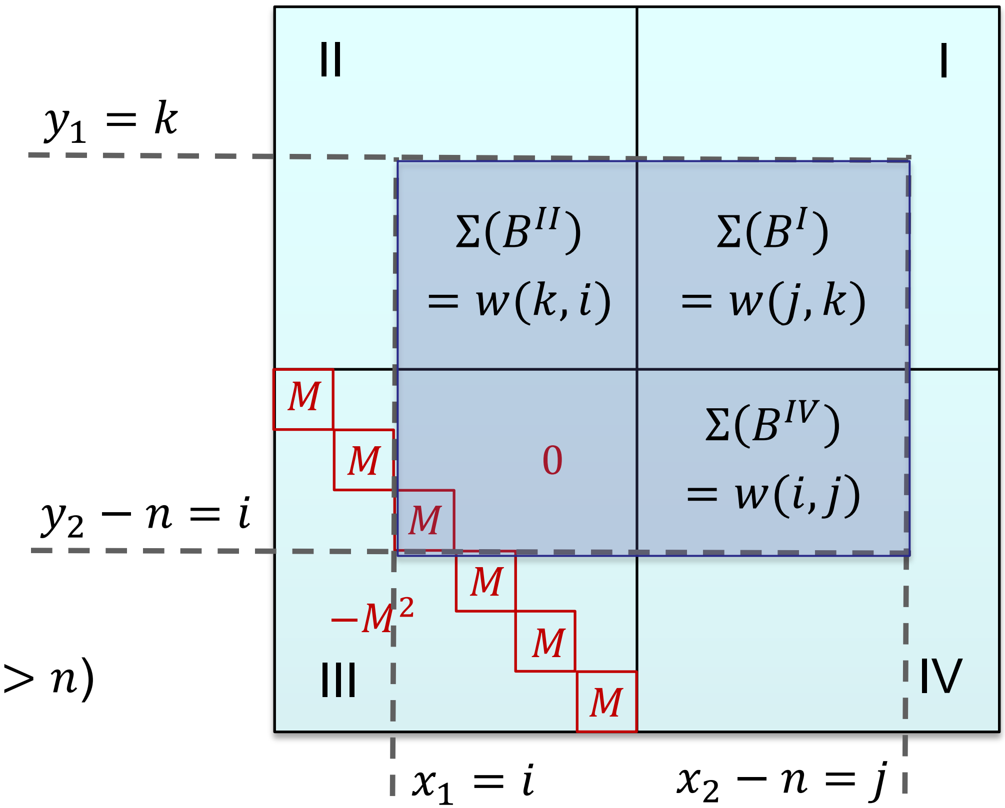
In quadrant II we want for any k, i:

$$\sum_{y=k}^{n} \sum_{x=i}^{n} A_{y,x} = w(k,i)$$

this is satisfied by defining:

$$A_{k,i} \coloneqq w(k,i) - w(k+1,i) - w(k,i+1) + w(k+1,i+1)$$

(where $w(x,y) \coloneqq 0$ for $x > n$ or $y > n$)



I. Equivalence of APSP and NegTriangle

II. Example Applications

III. Further Topics

IV. Conclusion

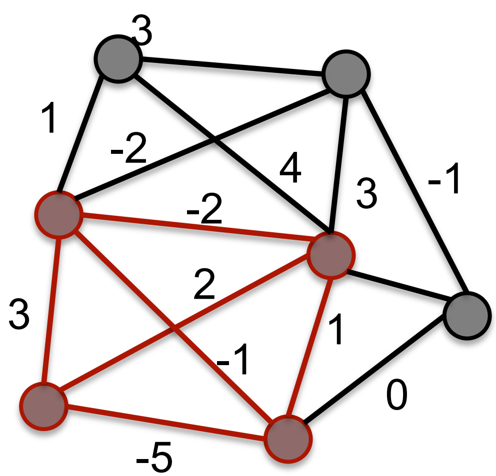
Weighted k-Clique

Problem Negative-k-Clique:

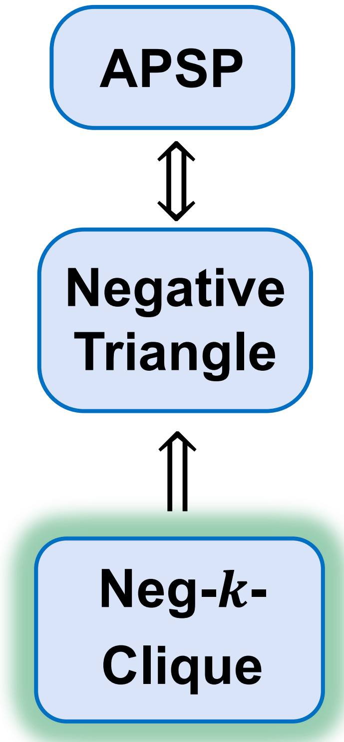
Given weighted directed graph G
is there a k-Clique with negative total edge-weight?

Neg-k-Clique-Hypothesis:

$\forall \varepsilon > 0, k \geq 3$: Neg-$k$-Clique has no $O(n^{k-\varepsilon})$ algorithm



"Yields more lower bounds since the input is sparser"



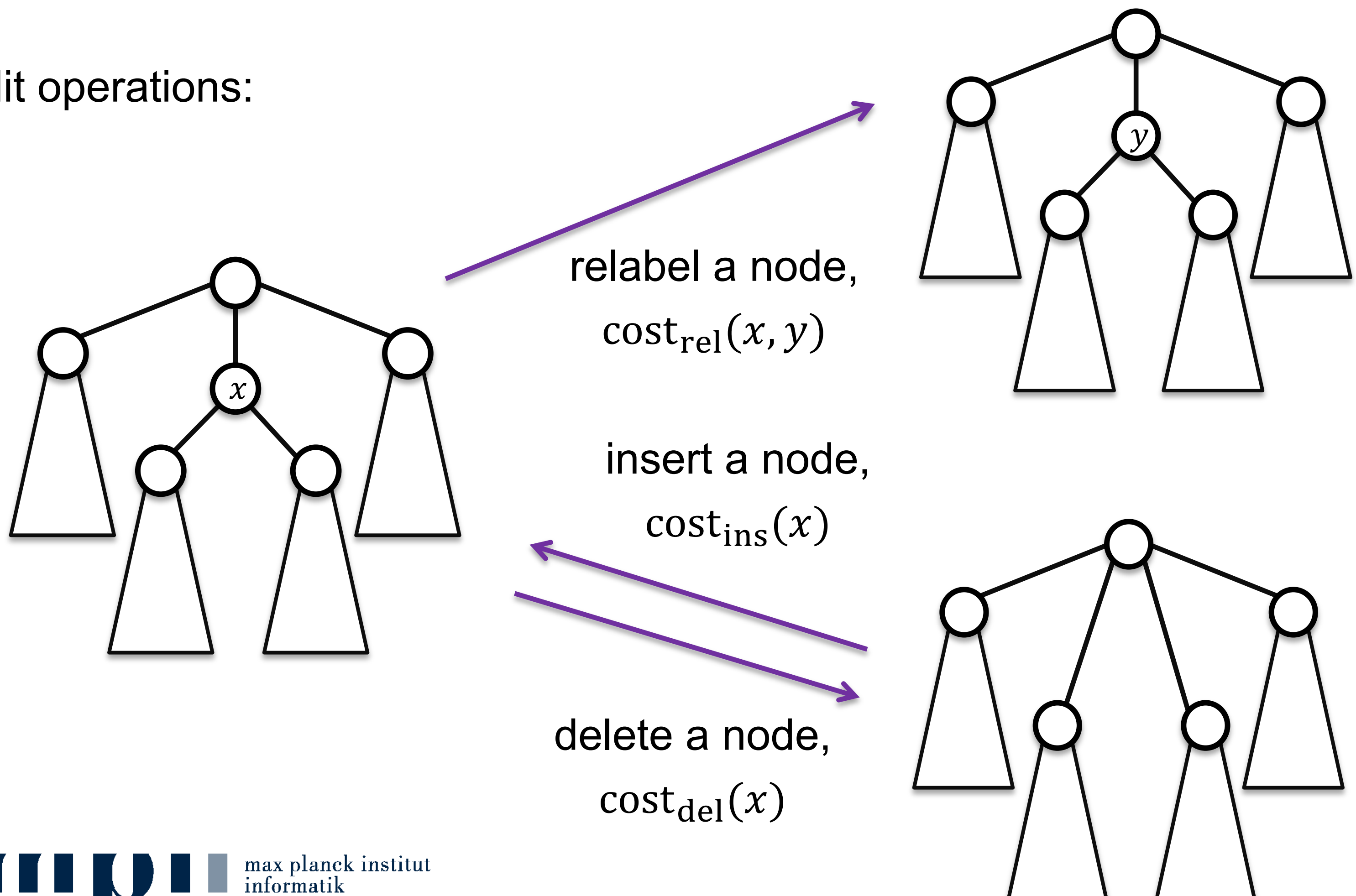
Tree Edit Distance

Input size: $O(n + |\Sigma|^2)$

on two rooted ordered trees T, T' with nodes labeled by Σ

determine minimum cost of edit operations transforming T into T'

edit operations:



Tree Edit Distance

Input size: $O(n + |\Sigma|^2)$

on two rooted ordered trees T, T' with nodes labeled by Σ

determine minimum cost of edit operations transforming T into T'

first algorithm: $O(n^6)$ [Tai'79]

a series of papers improved to: $O(n^3)$ [Demaine,Mozes,Rossman,Weimann'07]

Thm: [B.,Mozes,Gawrychowski,Weimann'18]

For alphabet $|\Sigma| = \Omega(n)$, a truly subcubic algorithm for tree edit distance implies a truly subcubic algorithm for **APSP**.

For $|\Sigma| = O(1)$, a truly subcubic algorithm for tree edit distance implies an $O(n^{k-\varepsilon})$ algorithm for **Neg-k-Clique**.

other applications: Max-Weight-Rectangle, Viterbi, ...

[Backurs,Dikkala,Tzamos'16] [Backurs, Tzamos'17]

Weighted k-Clique

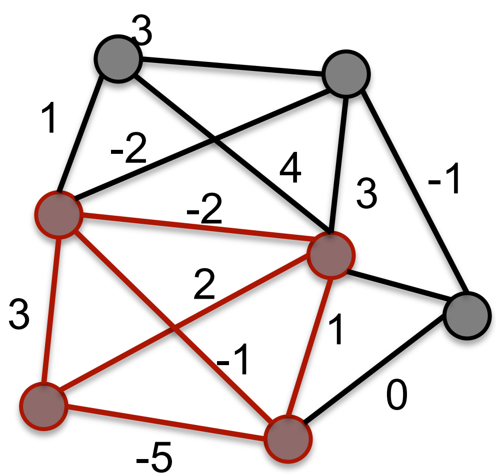
Problem Negative-k-Clique:

Given weighted directed graph G

is there a k-Clique with negative total edge-weight?

Neg-k-Clique-Hypothesis:

$\forall \varepsilon > 0, k \geq 3$: Neg-$k$-Clique has no $O(n^{k-\varepsilon})$ algorithm



APSP

$\Updownarrow$

Negative Triangle

$\Uparrow$

Neg-k-Clique

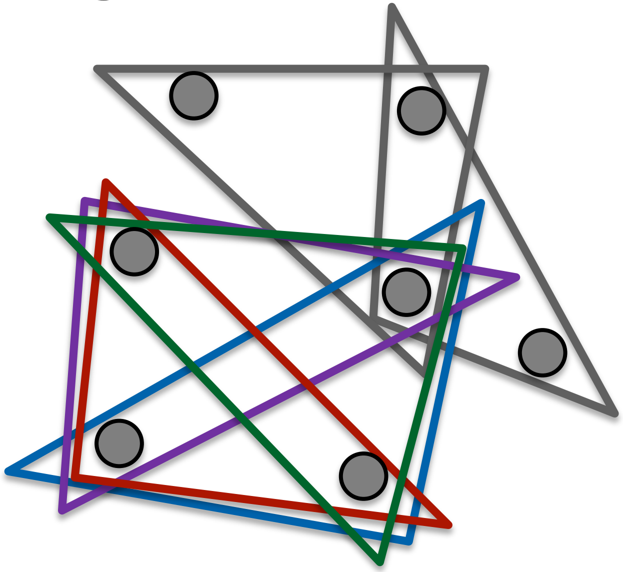
[Abboud,**B**,Dell, Nederlof'18]

OV

If OVH fails then Neg-k-Clique is in time $O(n^{k-\varepsilon})$

"Neg-k-Clique unifies OV and APSP"

(Weighted) k-Clique in Hypergraphs



r-hypergraph:

$G = (V, E)$ with $\mathrm{E} \subseteq \binom{V}{r}$

note: 2-hypergraph = graph

k-Clique in r-hypergraph:

vertices $v_1, \dots, v_k$ s.t.

for any $e \subseteq \{v_1, \dots, v_k\}$ of size r we have $e \in E$

$O(n^{0.79k})$ known for **k-Clique in graphs** [NP'85]

$O(n^{k-\varepsilon})$ not known for **Neg-k-Clique in graphs** or **k-Clique in 3-hypergraphs**

OVH fails:

$\Longrightarrow$ $O(n^{k-\varepsilon})$ for **Neg-k-Clique in r-hypergraphs**

[A**B**DN'18]

for any $k \gg r$ and weights bounded by $n^{f(k)}$

Proof Outline

Neg-$2k$-Clique r-hypergraphs $\xrightarrow{(1)}$ **ExactWeight-$2k$-Clique r-hypergraphs** $\xrightarrow{(2)}$ **$2k$-Clique $2r$-hypergraphs** $\xrightarrow{(3)}$ **OV**

(1) *From testing "≤" to testing "="*

(2) *Removing weights by increasing the arity*

(3) *Implementing Constraint Satisfaction Problems by Orthogonal Vectors*

Proof Outline – Step (2)

Removing weights by increasing the arity

ExactWeight-k-Clique

Given target t, graph G, weights w,
is there a k-clique of weight t?

$\rightarrow$

k-Clique

4-hypergraph G'

assume weights bounded by $W = O(n^{f(k)})$

Consider k-clique C with $\sum_{e \subseteq C} w(e) = t$

Base-B expansion: $t = \sum_\ell t_\ell \cdot B^\ell$, $w(e) = \sum_\ell w_\ell(e) \cdot B^\ell$

we have $\sum_{e \subseteq C} w(e) = t$

$\Leftrightarrow$ $\exists$ carries $c_\ell \in \{0, \dots, O(k^2)\}$ such that $c_\ell + \sum_{e \subseteq C} w_\ell(e) = t_\ell + c_{\ell+1} \cdot B$ $\forall \ell$

guess carries: blowup of $O(k^2)^{\log W / \log B} = n^{o(1)}$ for $B := \log n$

Proof Outline – Step (2)

Removing weights by increasing the arity

$W = O(n^{f(k)})$

$B := \log n$

ExactWeight-k-Clique

Given target t, graph G, weights w, is there a k-clique of weight t?

$\rightarrow$

k-Clique

4-hypergraph G'

New problem after guessing carries:

Find k-clique C with $c_\ell + \sum_{e \subseteq C} w_\ell(e) = t_\ell + c_{\ell+1} \cdot B \quad \forall \ell$

$\Leftrightarrow \sum_{e \subseteq C} w'_\ell(e) = 0 \quad \forall \ell$ with $w'_\ell(e) := c_\ell + \binom{k}{2} w_\ell(e) - t_\ell - c_{\ell+1} \cdot B$

$\Leftrightarrow \sum_\ell (\sum_{e \subseteq C} w'_\ell(e))^2 = 0$

$\Leftrightarrow \sum_{e_1, e_2 \subseteq C} \sum_\ell w'_\ell(e_1) \cdot w'_\ell(e_2) = 0$

$\Leftrightarrow \sum_{h \subseteq C, |h|=4} w''(h) = 0$ with weights bounded by $O\left(B^2 \frac{\log W}{\log B}\right) = \text{polylog}\, n$

guess all weights: $(\text{polylog}\, n)^{O(k^4)} = n^{o(1)}$ blowup

Proof Outline – Step (2)

Removing weights by increasing the arity

$W = O(n^{f(k)})$

$B := \log n$

ExactWeight-k-Clique

Given target t, graph G, weights w, is there a k-clique of weight t?

r-hypergraph

→

k-Clique

4-hypergraph G'

$2r$-hypergraph

New problem after guessing carries:

Find k-clique C with $c_\ell + \sum_{e \subseteq C} w_\ell(e) = t_\ell + c_{\ell+1} \cdot B \;\; \forall \ell$

$\Leftrightarrow \sum_\ell (c_\ell + \sum_{e \subseteq C} w_\ell(e) - t_\ell - c_{\ell+1} \cdot B)^2 = 0$

$\Leftrightarrow \sum_\ell (\sum_{e \subseteq C} w'_\ell(e))^2 = 0$ with $w'_\ell(e) := c_\ell + \binom{k}{2} w_\ell(e) - t_\ell - c_{\ell+1} \cdot B$

$\Leftrightarrow \sum_{e_1, e_2 \subseteq C} \sum_\ell w'_\ell(e_1) \cdot w'_\ell(e_2) = 0$

$\Leftrightarrow \sum_{h \subseteq C, |h|=4} w''(h) = 0$ with weights bounded by $O\left(B^2 \frac{\log W}{\log B}\right) = \operatorname{polylog} n$

guess all weights: $(\operatorname{polylog} n)^{O(k^2)} = n^{o(1)}$ blowup

Proof Outline – Putting it together

Neg-$2k$-Clique r-hypergraphs $\xrightarrow{(1)}$ **ExactWeight-$2k$-Clique r-hypergraphs** $\xrightarrow{(2)}$ **$2k$-Clique $2r$-hypergraphs** $\xrightarrow{(3)}$ **OV**

r-hypergraph

$G = (V, E)$

$n = O(|V|^k)$

$d = O(|V|^{2r})$

OV-Hypothesis: (moderate dimension)

$\forall \varepsilon, \delta > 0$: OV in $d = n^\delta$ has no $O(n^{2-\varepsilon})$-time algorithm

If OVH fails, then for some ε, δ OV is in time $O(n^{2-\varepsilon})$ in $d = n^\delta$

Then for any r and $k \geq 2r/\delta$, **Neg-$2k$-Clique in r-hypergraphs** is in $O(|V|^{2k-\varepsilon k})$

max planck institut informatik

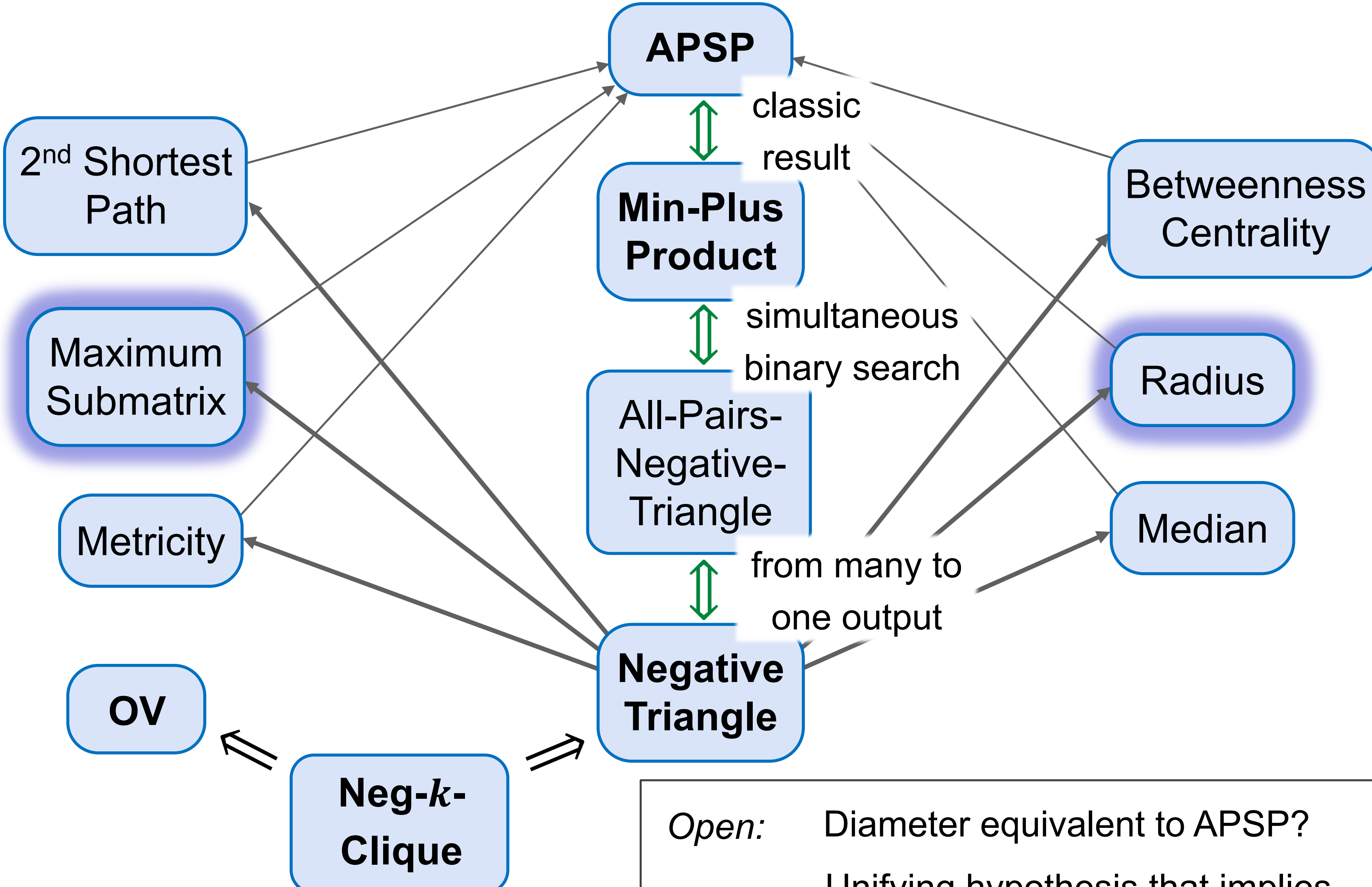
I. Equivalence of APSP and NegTriangle

II. Example Applications

III. Further Topics

IV. Conclusion

Conclusion



Open: Diameter equivalent to APSP?

Unifying hypothesis that implies OV-H, APSP-H and 3SUM-H?