max planck institut
informatik

# Truly Subcubic Algorithms for Language Edit Distance and RNA Folding via Fast Bounded-Difference Min-Plus Product

*Karl Bringmann*, Fabrizio Grandoni, Barna Saha, Virginia Vassilevska Williams

June 11, 2017

# Bounded Differences (BD) Matrices

Integer matrix $M$ has **BD** if for all $i, j$:

$$|M[i,j] - M[i,j+1]| \leq 1 \quad \text{and}$$

$$|M[i,j] - M[i+1,j]| \leq 1$$

| | | | |
|---|---|---|---|
| 2 | 2 | 3 | 2 |
| 1 | 1 | 2 | 3 |
| 2 | 1 | 2 | 3 |
| 1 | 0 | 1 | 2 |

More generally: $\boldsymbol{W}$**-BD** when differences are at most $W$

# (min,+) Product

For $n \times n$-matrices $A, B$, their (min,+) product $C = A * B$ is defined by

$$C[i,j] = \min_k A[i,k] + B[k,j]$$

(min,+) product is equivalent to All Pairs Shortest Paths [Fischer,Meyer'71]

trivial algorithm: $O(n^3)$

best known algorithm: $n^3/2^{\Theta(\sqrt{\log n})}$ [Williams'14]

---

Standard matrix multiplication: $C[i,j] = \sum_k A[i,k] \cdot B[k,j]$

time $O(n^\omega)$ where $\omega \leq 2.373$

# (min,+) Product

For $n \times n$-matrices $A, B$, their (min,+) product $C = A * B$ is defined by

$$C[i,j] = \min_k A[i,k] + B[k,j]$$

(min,+) product is equivalent to All Pairs Shortest Paths [Fischer,Meyer'71]

trivial algorithm: $O(n^3)$

best known algorithm: $n^3 / 2^{\Theta(\sqrt{\log n})}$ [Williams'14]

**Big Open Problem:** ***Is (min,+) product in time $O(n^{3-\varepsilon})$ for some $\varepsilon > 0$?***

***Study special cases!***

# (min,+) Product for Structured Matrices

**Matrices with small entries:** [Alon,Galil,Margalit'97]

If $A, B$ have entries in $\{-T, \dots, T\} \cup \{\infty\}$
then $A * B$ can be computed in time $\tilde{O}(Tn^{\omega})$

Sketch:

$$C[i,j] = \min_k A[i,k] + B[k,j] \longrightarrow A'[i,j] = x^{A[i,j]}$$

$$C'[i,j] = \sum_k A'[i,k] \cdot B'[k,j]$$

$$C[i,j] = \text{degree of highest monomial in } C'[i,j]$$

# (min,+) Product for Structured Matrices

**Matrices with small entries:** [Alon,Galil,Margalit'97]

If $A, B$ have entries in $\{-T, \dots, T\} \cup \{\infty\}$
then $A * B$ can be computed in time $\tilde{O}(Tn^{\omega})$

**Matrices with few distinct entries:** [Yuster'09]

If each row of $A$ has a *small* number of distinct entries,
then for arbitrary $B$ we can compute $A * B$ in *truly subcubic* time

**Question:** ***Is (min,+) product in time $O(n^{3-\varepsilon})$ for BD matrices?***

***Why care about BD matrices?***

# 1st Application: Language Edit Distance (LED)

*for simplicity:* $|G| = O(1)$

**CFG Parsing:**

Given a context-free grammar $G$ and a string $s$ of length $n$, is $s$ in $L(G)$?

... is in time $\tilde{O}(n^{\omega})$ [L. Valiant'75]

**Language Edit Distance:** *„error-correcting CFG parsing"*

Given a CFG $G$ and a string $s$, compute minimum edit distance of $s$ to any string in $L(G)$

insertions, deletions, substitutions

... is in time $O(n^3)$ [Aho,Peterson'72]

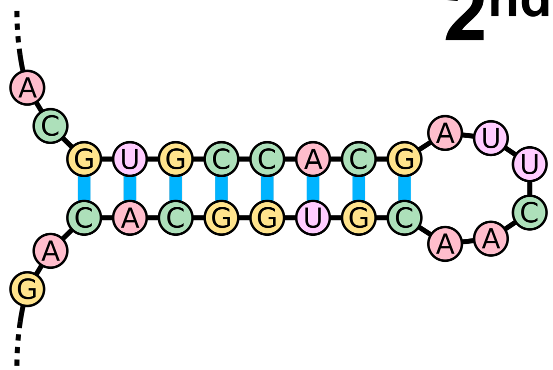
We show using Valiant's approach:

If **(min,+) product on BD matrices** is in time $O(n^{\alpha})$, then LED is in time $\tilde{O}(n^{\alpha})$

~8 page proof

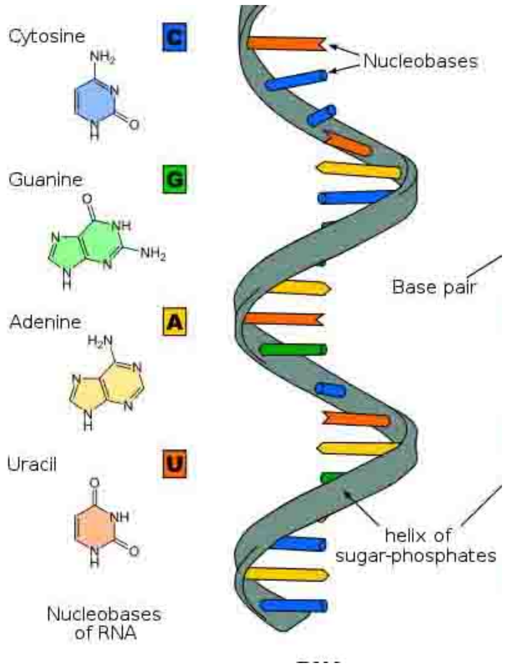
intuitive reason for BD: LED($s$) and LED($sc$) differ by $\leq 1$ for any symbol $c$

# 2nd Application: RNA Folding




RNA can be seen as a sequence of symbols from {A,C,G,U}

Biologists want to predict the secondary structure of RNA:




A can pair with U, and C can pair with G

Given an RNA sequence, find the largest set of matching pairs, such that no two pairs intersect

**AUUGCAG not allowed but AUUGCAG is okay**

... is in time $O(n^3)$ [Nussinov,Jacobson'80]

Disclaimer: No author of this paper is a biologist.

... can be cast as a LED problem (without substitutions)

If **(min,+) product on BD matrices** is in time $O(n^\alpha)$, then RNA Folding is in time $\tilde{O}(n^\alpha)$

max planck institut informatik

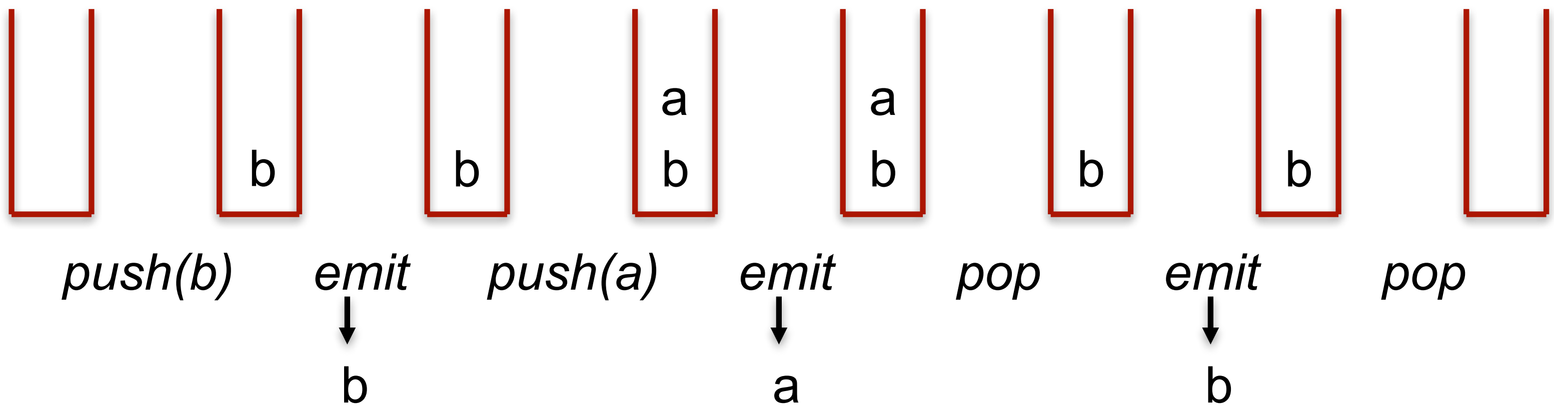
# 3rd Application: Optimal Stack Generation

*for simplicity:* $|\Sigma| = O(1)$

**Optimal Stack Generation:**

Given a string $s$ over alphabet $\Sigma$, determine the shortest sequence of stack operations *push(.), emit, pop* s.t. performing these operations starting from an empty stack will emit $s$ and end with an empty stack

$s$ = bab




... is in time $O(n^3)$ (dynamic programming) [Tarjan'05]

We show: If **(min,+) product on O(1)-BD matrices** is in time $O(n^\alpha)$, then Optimal Stack Generation is in time $\tilde{O}(n^\alpha)$

intuitive reason for BD: OSG($s$) and OSG($sc$) differ by $\leq 3$ for any $c \in \Sigma$

# Main Result

... so we have seen that (min,+) product of BD matrices is well motivated

**Main Result:**

We can compute the (min,+) product of **BD** matrices

in randomized time $\cancel{O(n^{2.83})}$ and deterministic time $O(n^{2.87})$

**here:** $\boldsymbol{O(n^{2.9})}$

**Generalization:**

For $\boldsymbol{W}$**-BD** matrix $A$ with $W \ll n^{3-\omega} \approx n^{0.626}$ and **arbitrary** $B$

we can compute their (min,+) product in randomized **truly subcubic** time

max planck institut
informatik

# Algorithm Sketch

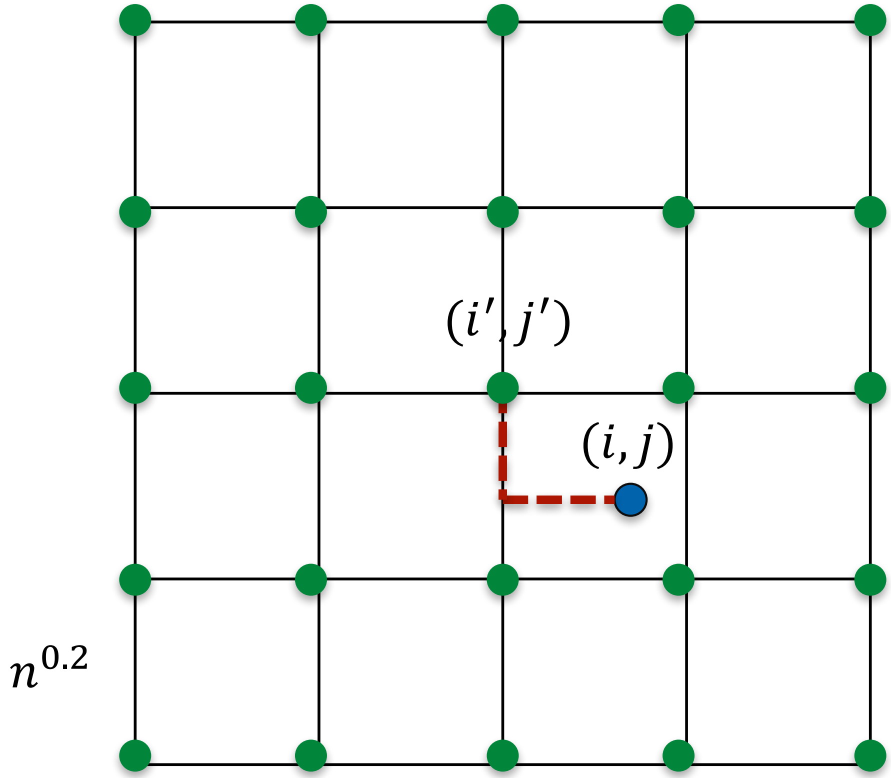
Input: BD matrices $A, B$. Want: $C[i,j] = \min_k A[i,k] + B[k,j]$

1) Compute approximation $D[i,j] = C[i,j] \pm O(n^{0.2})$ *time $O(n^{2.6})$*

compute $C[i,j]$ exactly for all $i, j$ that are multiples of $n^{0.2}$

set $D[i,j]$ to some $C[i',j']$ by rounding $i, j$

*If $A, B$ are BD, then their (min,+) product is also BD*



# Algorithm Sketch

Input: BD matrices $A, B$. Want: $C[i,j] = \min_k A[i,k] + B[k,j]$

1) Compute approximation $D[i,j] = C[i,j] \pm O(n^{0.2})$

$A[i,k] + B[k,j] = C[i,j]$ implies $\underbrace{|A[i,k] + B[k,j] - D[i,j]| \leq O(n^{0.2})}_{\text{call these triples } (i,k,j) \textbf{ relevant}}$

then $C[i,j] = \min_{k:(i,k,j) \text{ relevant}} A[i,k] + B[k,j]$

# Algorithm Sketch

$(i,k,j)$ relevant:
$|A[i,k]+B[k,j]$
$-D[i,j]| \le O(n^{0.2})$

Input: BD matrices $A, B$. Want: $C[i,j] = \min_k A[i,k] + B[k,j]$

1) Compute approximation $D[i,j] = C[i,j] \pm O(n^{0.2})$

2) Cover most relevant triples:

fix $i^*, j^*$, and define matrices $A^*, B^*$

$$A^*[i,k] := (A[i,k] + \cancel{B[k,j^*]} - D[i,j^*]) - (\cancel{A[i^*,k]} + \cancel{B[k,j^*]} - D[i^*,j^*])$$

$$B^*[k,j] := (\cancel{A[i^*,k]} + B[k,j] - D[i^*,j])$$

(min,+) product $C^*$ of $A^*, B^*$:

$$C^*[i,j] = \min_k A^*[i,k] + B^*[k,j] \; = C[i,j] \underbrace{- D[i,j^*] + D[i^*,j^*] - D[i^*,j]}_{\text{can be cancelled afterwards}}$$

# Algorithm Sketch

$(i,k,j)$ relevant:
$|A[i,k]+B[k,j] - D[i,j]| \leq O(n^{0.2})$

Input: BD matrices $A, B$. Want: $C[i,j] = \min_k A[i,k] + B[k,j]$

1) Compute approximation $D[i,j] = C[i,j] \pm O(n^{0.2})$

2) Cover most relevant triples:

fix $i^*, j^*$, and define matrices $A^*, B^*$

$$A^*[i,k] \coloneqq (A[i,k] + B[k,j^*] - D[i,j^*]) - (A[i^*,k] + B[k,j^*] - D[i^*,j^*])$$

$$B^*[k,j] \coloneqq (A[i^*,k] + B[k,j] - D[i^*,j])$$

if $(i,k,j^*), (i^*,k,j^*), (i^*,k,j)$ are all relevant, then $|A^*[i,k]|, |B^*[k,j]| = O(n^{0.2})$

set all $\Omega(n^{0.2})$-entries of $A^*, B^*$ to $\infty$

then (min,+) product of $A^*$ and $B^*$ can be computed in time $\tilde{O}(n^{\omega+0.2})$

$(i,k,j)$ is „covered“ if $A^*[i,k]$ and $B^*[k,j]$ are $O(n^{0.2})$, i.e., not set to ∞

# Algorithm Sketch

$(i,k,j)$ relevant:
$|A[i,k] + B[k,j] - D[i,j]| \leq O(n^{0.2})$

$(i,k,j)$ is „covered“ if $A^*[i,k]$ and $B^*[k,j]$ are $O(n^{0.2})$ in some round

Input: BD matrices $A, B$. Want: $C[i,j] = \min_k A[i,k] + B[k,j]$

1) Compute approximation $D[i,j] = C[i,j] \pm O(n^{0.2})$

2) Cover most relevant triples:

initialize $\hat{C}[i,j] \coloneqq \infty$

repeat for $O(n^{0.3} \log n)$ rounds: *$\tilde{O}(n^{0.3})$ iterations*

pick $i^*, j^*$ randomly

$A^*[i,k] \coloneqq (A[i,k] + B[k,j^*] - D[i,j^*]) - (A[i^*,k] + B[k,j^*] - D[i^*,j^*])$

$B^*[k,j] \coloneqq (A[i^*,k] + B[k,j] - D[i^*,j])$

set all $\Omega(n^{0.2})$-entries of $A^*, B^*$ to $\infty$

compute (min,+) product $C^* = A^* * B^*$ *time $O(n^{\omega+0.2}) = O(n^{2.6})$*

$\hat{C}[i,j] \coloneqq \min\{\hat{C}[i,j],\ C^*[i,j] + D[i,j^*] - D[i^*,j^*] + D[i^*,j]\}$

**Lem:** After $O(n^\rho \log n)$ rounds there are $O(n^{3-\rho/3} + n^{2.5})$ **uncovered relevant** triples w.h.p. $= O(n^{2.9})$

*total time $\tilde{O}(n^{2.9})$*

# Algorithm Sketch

$(i,k,j)$ relevant:

$|A[i,k] + B[k,j] - D[i,j]| \le O(n^{0.2})$

$(i,k,j)$ is „covered“ if $A^*[i,k]$ and $B^*[k,j]$ are $O(n^{0.2})$ in some round

Input: BD matrices $A, B$. Want: $C[i,j] = \min_k A[i,k] + B[k,j]$

1) Compute approximation $D[i,j] = C[i,j] \pm O(n^{0.2})$

2) Cover most relevant triples

3) Enumerate uncovered relevant triples:

”for each uncovered relevant $(i,k,j)$:“

$$\hat{C}[i,j] \coloneqq \min\{\, \hat{C}[i,j], A[i,k] + B[k,j] \,\}$$

now $\hat{C}$ is correct output

# Algorithm Sketch

$(i,k,j)$ relevant:
$|A[i,k] + B[k,j] - D[i,j]| \le O(n^{0.2})$

Input: BD matrices $A, B$. Want: $C[i,j] = \min_k A[i,k] + B[k,j]$

$(i,k,j)$ is „covered“ if $A^*[i,k]$ and $B^*[k,j]$ are $O(n^{0.2})$ in some round

1) Compute approximation $D[i,j] = C[i,j] \pm O(n^{0.2})$

2) Cover most relevant triples

3) Enumerate uncovered relevant triples:

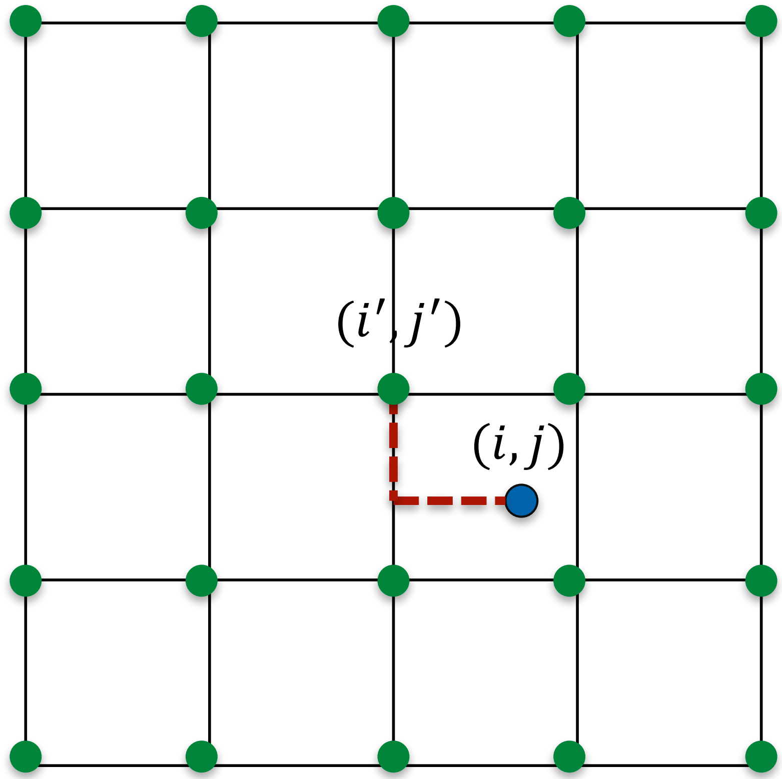
for all $i', k', j'$ divisible by $n^{0.2}$:

if $(i', k', j')$ is relevant and uncovered:

for all $i' - n^{0.2} < i \le i'$,
$k' - n^{0.2} < k \le k'$,
$j' - n^{0.2} < j \le j'$:

$\hat{C}[i,j] := \min\{ \hat{C}[i,j], A[i,k] + B[k,j] \}$

now $\hat{C}$ is correct output



# Algorithm Sketch

$(i,k,j)$ relevant:
$|A[i,k] + B[k,j] - D[i,j]| \leq O(n^{0.2})$

Input: BD matrices $A, B$. Want: $C[i,j] = \min_k A[i,k] + B[k,j]$

1) Compute approximation $D[i,j] = C[i,j] \pm O(n^{0.2})$

$(i,k,j)$ is „covered“ if $A^*[i,k]$ and $B^*[k,j]$ are $O(n^{0.2})$ in some round

2) Cover most relevant triples

3) Enumerate uncovered relevant triples:

for all $i', k', j'$ divisible by $n^{0.2}$: *$O(n^{2.4})$ iterations*

if $(i', k', j')$ is relevant and uncovered: *time $O(n^{0.3} \log n)$*

$$\left.\begin{array}{r} \text{for all } i' - n^{0.2} < i \leq i', \\ k' - n^{0.2} < k \leq k', \\ j' - n^{0.2} < j \leq j': \end{array}\right\} \text{either all or none are relevant and uncovered}$$

$\hat{C}[i,j] := \min\{\hat{C}[i,j], A[i,k] + B[k,j]\}$

*total time $O(n^{2.9})$ = number of relevant uncovered triples*

now $\hat{C}$ is correct output

*total time of algorithm $\tilde{O}(n^{2.9})$*

# Correctness

**Lem:** After $O(n^\rho \log n)$ rounds there are $O(n^{3-\rho/3} + n^{2.5})$ **uncovered relevant** triples w.h.p.

$(i,k,j)$ relevant:
$|A[i,k] + B[k,j] - D[i,j]| \le O(n^{0.2})$

$(i,k,j)$ is „covered“ if $A^*[i,k]$ and $B^*[k,j]$ are $O(n^{0.2})$ in some round

If $(i,k,j^*)$, $(i^*,k,j^*)$, $(i^*,k,j)$ are all relevant, then $(i,k,j)$ is covered

...

pick $i^*, j^*$ randomly

$A^*[i,k] \coloneqq (A[i,k] + B[k,j^*] - D[i,j^*]) - (A[i^*,k] + B[k,j^*] - D[i^*,j^*])$

$B^*[k,j] \coloneqq (A[i^*,k] + B[k,j] - D[i^*,j])$

set all $\Omega(n^{0.2})$-entries of $A^*, B^*$ to $\infty$

...

# Correctness

**Lem:** After $O(n^\rho \log n)$ rounds there are $O(n^{3-\rho/3} + n^{2.5})$ **uncovered relevant** triples w.h.p.

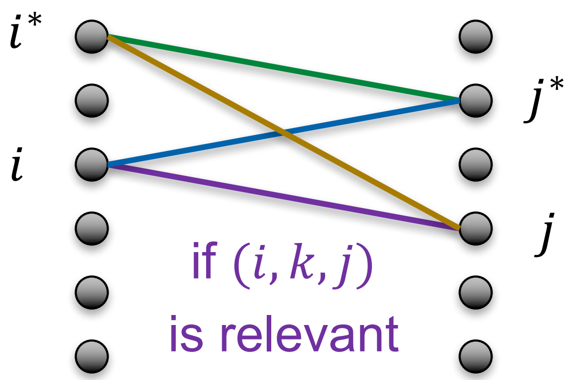
$(i,k,j)$ relevant:
$|A[i,k] + B[k,j] - D[i,j]| \leq O(n^{0.2})$

$(i,k,j)$ is „covered“ if $A^*[i,k]$ and $B^*[k,j]$ are $O(n^{0.2})$ in some round

If $(i,k,j^*)$, $(i^*,k,j^*)$, $(i^*,k,j\ )$ are all relevant, then $(i,k,j)$ is covered

When are $(i,k,j^*)$, $(i^*,k,j^*)$, $(i^*,k,j\ )$ all relevant?

bipartite graph $G_k$:




we „cover“ a relevant triple $(i,k,j)$ if $i,j,i^*,j^*$ form a 4-cycle in $G_k$

need: $G_k$ contains many 4-cycles

have: many relevant triples

**Lem:** Any bipartite graph with $m \geq 4n^{1.5}$ edges contains $\Omega(m^4/n^4)$ 4-cyles.

# Algorithm Recap

Input: BD matrices $A, B$. Want: $C[i,j] = \min_k A[i,k] + B[k,j]$

1) Compute approximation $D[i,j] = C[i,j] \pm O(n^{0.2})$

   compute $C[i,j]$ exactly for all $i, j$ that are multiples of $n^{0.2}$

   set $D[i,j]$ to some $C[i',j']$ by rounding $i, j$

2) Cover most relevant triples:

   initialize $\hat{C}[i,j] := \infty$

   repeat for $O(n^{0.3} \log n)$ rounds:

   pick $i^*, j^*$ randomly

   $A^*[i,k] := (A[i,k] + B[k,j^*] - D[i,j^*]) - (A[i^*,k] + B[k,j^*] - D[i^*,j^*])$

   $B^*[k,j] := (A[i^*,k] + B[k,j] - D[i^*,j])$

   set all $\Omega(n^{0.2})$-entries of $A^*, B^*$ to $\infty$

   compute (min,+) product $C^* = A^* * B^*$

   $\hat{C}[i,j] := \min\{ \hat{C}[i,j],\ C^*[i,j] + D[i,j^*] - D[i^*,j^*] + D[i^*,j] \}$

3) Enumerate uncovered relevant triples:

   for all $i', k', j'$ divisible by $n^{0.2}$:

   if $(i', k', j')$ is relevant and uncovered:

   for all $i' - n^{0.2} < i \le i',\ k' - n^{0.2} < k \le k',\ j' - n^{0.2} < j \le j'$:

   $\hat{C}[i,j] := \min\{ \hat{C}[i,j], A[i,k] + B[k,j] \}$

$(i,k,j)$ relevant:

$|A[i,k] + B[k,j] - D[i,j]| \le O(n^{0.2})$

$(i,k,j)$ is „covered“ if $A^*[i,k]$ and $B^*[k,j]$ are $O(n^{0.2})$ in some round

# Conclusion

we generalize the subcubic special cases of (min,+) matrix multiplication:

(min,+) product of **BD** matrices can be solved in rand. time $O(n^{2.83})$

this yields subcubic $O(n^{2.83})$ algorithms for:

- *Language Edit Distance*, a classic parsing problem from ‘72
- *RNA Folding*, a classic bioinformatics problem from ‘80
- *Optimal Stack Generation*, an open problem by Tarjan

**Open Problems:**

1) What is the right exponent?

[Abboud,Backurs,V-Williams15]

Conditional lower bounds imply that LED and RNA Folding are in $\widetilde{\Omega}(n^{\omega})$,

2) Find more applications of BD (min,+) product