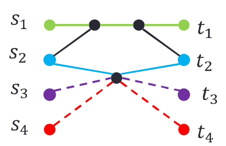
New Hardness Results for Routing on Disjoint Paths

Julia Chuzhoy[1], David H. K. Kim[2], Rachit Nimavat[3]

[1,3]Toyota Technological Institute at Chicago, [2]Computer Science Department, University of Chicago

NODE-DISJOINT PATHS (NDP) PROBLEM

- **Input:** Graph G, source-destination pairs $(s_1,t_1),\dots,(s_k,t_k)$
- **Output:** Route as many pairs as possible via node-disjoint paths



n: Number of graph vertices

Terminals: Vertices participating in demand pairs

$OPT_{NDP} = 2$
$OPT_{EDP} = 4$

Edge-Disjoint Paths Problem: Route as many demand pairs as possible via edge-disjoint paths

KNOWN RESULTS

- NP-Hard, even in planar graphs and grid graphs

  Goal: Route OPT/α demand pairs — α −approximation
- Where we stand?
  - General Case: $O(\sqrt{n})$ - Approximation vs $\approx \Omega\left(\sqrt{\log n}\right)$ - Hardness
  - Grid Graphs: $O(n^{1/4})$ - Approximation vs APX - Hardness
  - Planar Graphs: $O(n^{9/19})$ - Approximation vs APX - Hardness
- Similar situation, even in EDP (Grids ↔ Walls)
- What if we allow congestion?

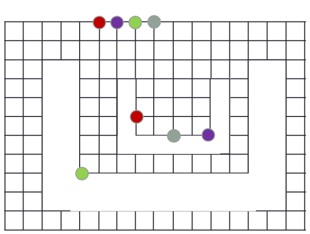
  Congestion 2 ⇒ polylog(k) − Approximation for NDP/EDP

OUR RESULT

$2^{\Omega(\sqrt{\log n})}$ - Hardness for NDP/EDP unless $NP \subseteq DTIME\left(n^{O(\log n)}\right)$ for:

- planar graphs
- max vertex degree 3
- all sources on the boundary of outer face

Here: Hardness for NDP for ***grids with holes*** with all sources on top row



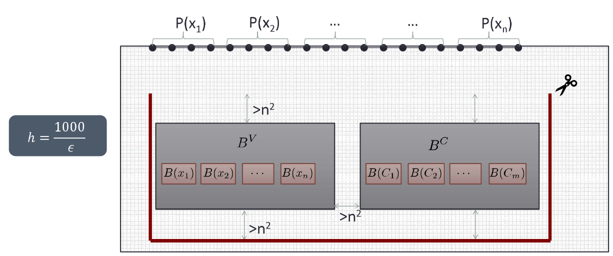
ROADMAP

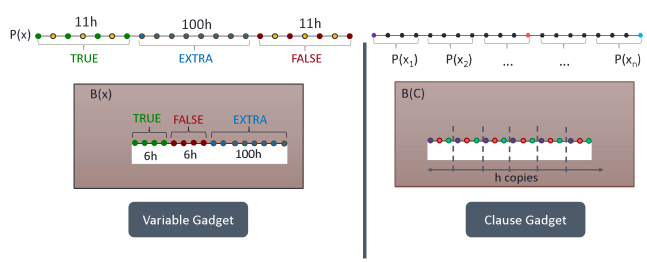
- Starting Point: 3SAT(5) instance φ
- [PCP Theorem] Unless P=NP, no efficient algorithm can distinguish between:
  - Yes-Instance: Some assignment satisfies all clauses
  - No-Instance: No assignment satisfies more than $(1-\epsilon)$-fraction of clauses
- Build NDP instance of size $N = n^{O(\log n)}$ such that:
  - φ is YI => Can route C_{YI} demand pairs
  - φ is NI => No solution routes more than C_{NI} demand pairs
- The gap: $\frac{C_{YI}}{C_{NI}} = 2^{\Omega(\log n)} = 2^{\Omega(\sqrt{\log N})}$

IDEA

- Construction in stages.
- Stage 1: Gap = $\Omega(1)$, Size = $O(poly\ n)$
- $\Theta(\log n)$ stages. In every stage: Gap grows by $\Omega(1)$, Size grows by $O(n \cdot \text{current-gap})$
- End: Gap = $2^{\Omega(\log n)}$, Size = $n^{O(\log n)}$

LEVEL 1 INSTANCE : BIRD'S EYE VIEW





Composable Instance!

- Can move the cut-out of Level 1 instance around
- Can move sources along the top boundary

LEVEL 1 : ANALYSIS

Yes Instance:

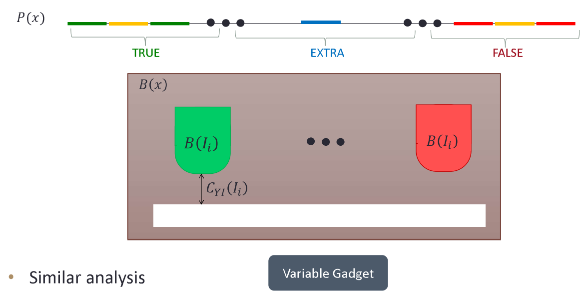
- x = True => Route all 'Extra' and 'True' pairs in B(x)
- x = False => Route all 'Extra' and 'False' pairs in B(x)
- Literal y = True in clause C => Route corresponding pairs in B(C)

No Instance:

- Can interpret routing in B^V as an assignment
- Too many pairs routed in B^C => Too many clauses satisfied!

LEVEL $i+1$: MATRYOSHKA DOLL

- Nested construction
- Replace each demand pair of Level 1 instance by a fresh copy of Level i instance



- Similar analysis

  Gap grows by $\Omega(1)$, Size grows by $O(n \cdot \text{current-gap})$

CONCLUSIONS AND FOLLOW-UP WORK

- $2^{\Omega(\sqrt{\log n})}$ - Hardness for NDP shown in *grids with holes*
- Better hardness?
  - $2^{\Omega(\log^{1-\delta} n)}$ – Hardness for NDP/EDP in grids/walls [ongoing work]
  - $n^{\Omega\left(\frac{1}{\log\log^2 n}\right)}$ – Hardness for NDP/EDP in grids/walls (assuming rETH) [ongoing work]
  - Polynomial hardness in general graphs?
- Better algorithms for grids?
  - $O(n^{1/4})$ – Approximation in grids vs $O(\sqrt{n})$ – approximation in general graphs
  - $2^{O(\sqrt{\log n})}$ – Approximation in grids if all sources lie on boundary [ongoing work]
- Congestion Minimization?