Brief Announcement: Distributed Derandomization Revisited

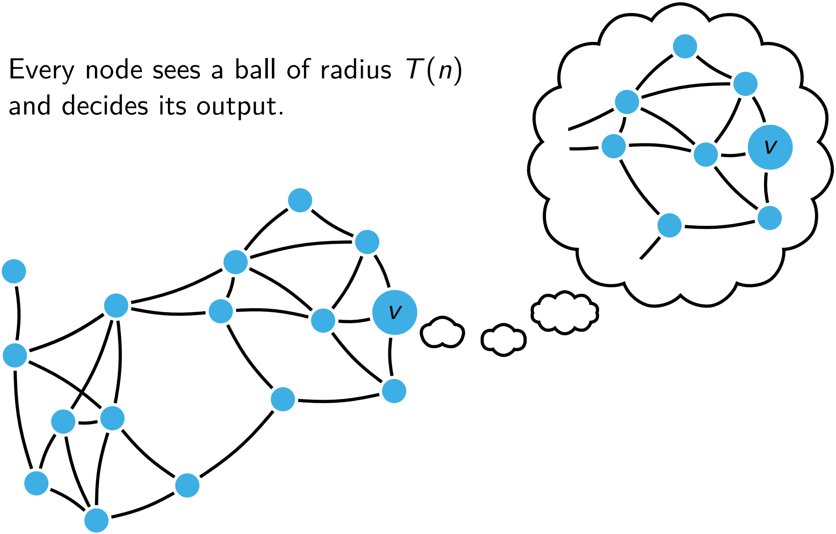
Sameep Dahal [1] Francesco d'Amore [1] Henrik Lievonen [1]
Timothé Picavet [1 2] Jukka Suomela [1]

[1]Aalto University, Finland

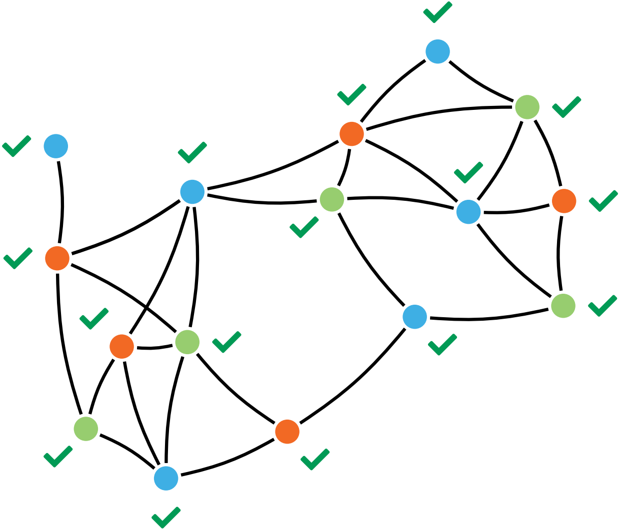
[2]ENS de Lyon, France (now at LaBRI)

The LOCAL model

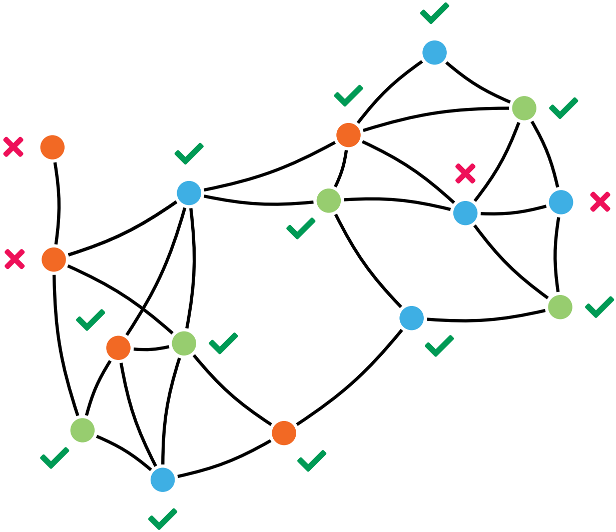
Every node sees a ball of radius $T(n)$ and decides its output.



Locally checkable labeling problems (LCLs)



Locally checkable labeling problems (LCLs)



Prior work and limitations

Theorem (Chang, Kopelowitz, and Pettie[1])

Let $\mathcal{A}_{\mathsf{rand}}$ a randomized LOCAL algorithm solving an LCL problem and ***that uses at most*** $\mathbf{r(n)}$ ***random bits****.*

[1] Yi-Jun Chang, Tsvi Kopelowitz, and Seth Pettie. An exponential separation between randomized and deterministic complexity in the LOCAL Model. SIAM Journal on Computing, 2019.

Prior work and limitations

Theorem (Chang, Kopelowitz, and Pettie[1])

*Let $\mathcal{A}_{\text{rand}}$ a randomized LOCAL algorithm solving an LCL problem and **that uses at most $\mathbf{r(n)}$ random bits**.*

derandomization

$\mathcal{A}_{\text{rand}}$ → $\mathcal{A}_{\text{det}}$

random
$T_{\text{rand}}(n)$ *complexity*

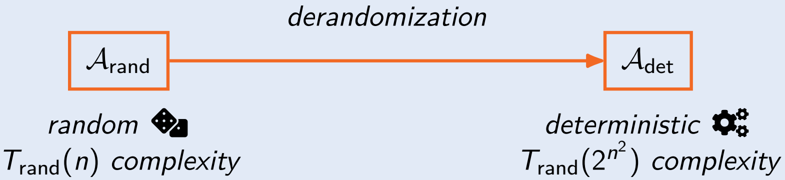
deterministic
$T_{\text{rand}}(2^{n^2})$ *complexity*

[1] Yi-Jun Chang, Tsvi Kopelowitz, and Seth Pettie. An exponential separation between randomized and deterministic complexity in the LOCAL Model. SIAM Journal on Computing, 2019.

Main result

Theorem

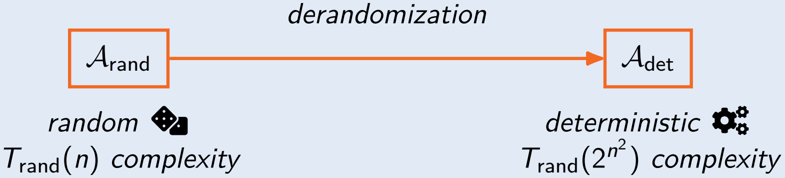
Let $\mathcal{A}_{\text{rand}}$ a randomized LOCAL algorithm solving an LCL ~~and that uses at most $r(n)$ random bits~~.



Main result

Theorem

*Let $\mathcal{A}_{\text{rand}}$ a randomized LOCAL algorithm solving a~~n LCL~~ **component wise verifiable problem** ~~and that uses at most $r(n)$ random bits~~.*



Proof Strategy

Idea:

Proof Strategy

Idea:

- lie about the number of vertices, say 2^{n^2} instead of n,

Proof Strategy

Idea:

- lie about the number of vertices, say 2^{n^2} instead of n,
- and find a good function f : IDs $\rightarrow$ infinite bit strings s.t. $\mathcal{A}_{\text{rand}}[f]$ is correct whp.

Proof Strategy

Idea:

- lie about the number of vertices, say 2^{n^2} instead of n,
- and find a good function f : IDs $\rightarrow$ infinite bit strings s.t. $\mathcal{A}_{\text{rand}}[f]$ is correct whp.

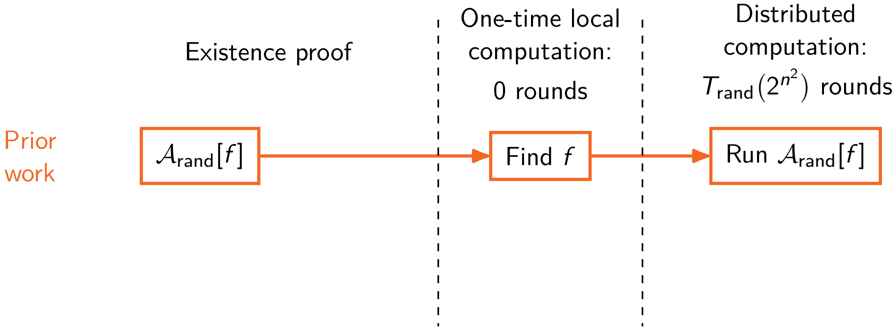
each vertex v uses $f(id(v))$ as random bit string

Proof Strategy

Idea:

- lie about the number of vertices, say 2^{n^2} instead of n,
- and find a good function f : IDs $\rightarrow$ infinite bit strings s.t. $\mathcal{A}_{\text{rand}}[f]$ is correct whp.

← each vertex v uses $f(id(v))$ as random bit string

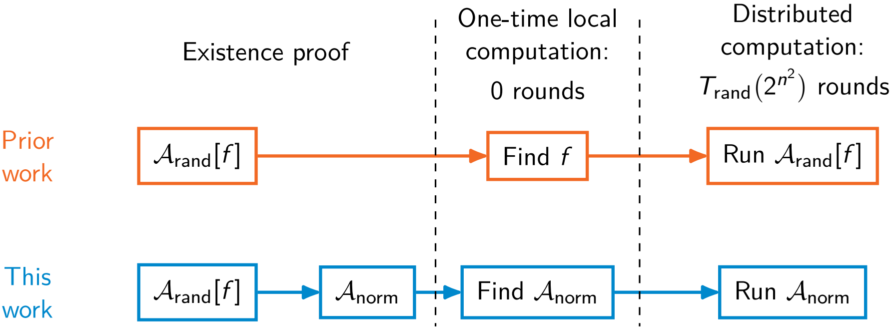


Proof Strategy

Idea:

- lie about the number of vertices, say 2^{n^2} instead of n,
- and find a good function f : IDs $\to$ infinite bit strings s.t. $\mathcal{A}_{\text{rand}}[f]$ is correct whp.

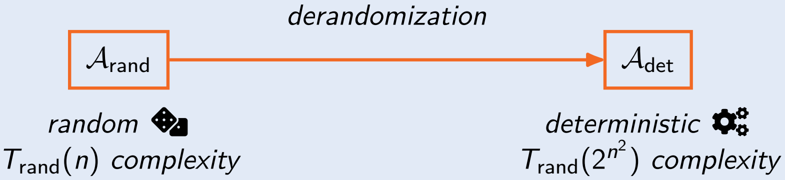
each vertex v uses $f(id(v))$ as random bit string



Extensions

Theorem

Let $\mathcal{A}_{\text{rand}}$ a randomized LOCAL algorithm solving an LCL problem on ***connected graphs****.*



Conclusion

- No more annoying bounded number of random bits assumption.
- The new derandomized algorithm is uniform in n.
- Generalization of the original theorem to:
  - component-wise verifiable problems,
  - and LCL problems on connected graphs.

Conclusion

- No more annoying bounded number of random bits assumption.
- The new derandomized algorithm is uniform in n.
- Generalization of the original theorem to:
  - component-wise verifiable problems,
  - and LCL problems on connected graphs.

Thanks for listening!