Load Balancing with Bounded Convergence in Dynamic Networks

Michael Dinitz  Jeremy Fineman  Seth Gilbert  Calvin Newport
Load Balancing in Graphs
Load Balancing in Graphs

- Work $x_u$ for each vertex $u$
- $T = \sum u x_u$
- Goal: redistribute work so that every node has approximately $T/n$ load
Load Balancing in Graphs

- Work $x_u$ for each vertex $u$
  - $T = \sum_u x_u$
- Goal: redistribute work so that every node has approximately $T/n$ load
- Can only send work along edges of graph, no global knowledge (Local Load Balancing)
- Synchronous rounds: transfer work along edges in each round
Desirable Properties
Desirable Properties

• In each round, edges used form a matching
  • Each node sends/receives work from at most one other node in each round
  • Important for some applications/models [Cybenko '89]
  • Dimension Exchange

• Works in dynamic graphs
  • Sequence of graphs $H = (G_1 = (V, E_1), G_2 = (V, E_2), ...)$, each connected
  • Distributed: each node sees only load of neighbors in current graph

• Converges *quickly*
Desirable Properties

• In each round, edges used form a matching
  • Each node sends/receives work from at most one other node in each round
  • Important for some applications/models [Cybenko '89]
  • Dimension Exchange

• Works in dynamic graphs
  • Sequence of graphs $H = (G_1 = (V, E_1), G_2 = (V, E_2), \ldots)$, each connected
  • Distributed: each node sees only load of neighbors in current graph

• Converges quickly

• Many results getting 2/3 — can we get all three?
Model
Model

• Beginning of round $r$:
  • Graph $G_r = (V, E_r)$
  • $x_u(r-1)$ work at node $u$
  • Each node $u$ knows work at neighbors (not total work or $n$)
Model

- **Beginning of round** \( r \):
  - Graph \( G_r = (V, E_r) \)
  - \( x_u(r-1) \) work at node \( u \)
  - Each node \( u \) knows work at neighbors (not total work or \( n \))

- **In round** \( r \):
  - Local computation to determine matching \( M_r \)
  - If \( \{u,v\} \in M_r \), distribute \( x_u(r-1) + x_v(r-1) \) between \( u \) and \( v \) to get 
    \( x_u(r) \) and \( x_v(r) \)
Model

• Beginning of round \( r \):
  • Graph \( G_r = (V, E_r) \)
  • \( x_u(r-1) \) work at node \( u \)
  • Each node \( u \) knows work at neighbors (not total work or \( n \))

• In round \( r \):
  • Local computation to determine matching \( M_r \)
  • If \( \{u,v\} \in M_r \), distribute \( x_u(r-1) + x_v(r-1) \) between \( u \) and \( v \) to get \( x_u(r) \) and \( x_v(r) \)

• Goal: \( \tau \)-convergence
  • \( |x_u(u) - x_v(r)| \leq \tau \) for all \( u, v \in V \)
Example
Example
Example
Example
Example
Example

5-Converged
Results: Upper Bound

**Theorem:** There is an algorithm which achieve \( \tau \)-convergence after

\[
O \left( \min \left( n^2 \log \left( \frac{Tn}{\tau} \right) , \frac{Tn \log n}{\tau} \right) \right)
\]

rounds with high probability
**Theorem**: There is an algorithm which achieve $\tau$-convergence after

$$O \left( \min \left( n^2 \log \left( \frac{Tn}{\tau} \right) , \frac{Tn \log n}{\tau} \right) \right)$$

rounds with high probability

- Match if $\tau = \Theta(T/n)$
  - $O(n^2 \log n)$ rounds
  - If small enough constant, loads within multiplicative factor

- Easy, simple algorithm (Max-Neighbor)
Results: Lower Bound

**Theorem**: No randomized algorithm can achieve $O(T/n)$-convergence in $o(n^2)$ rounds against an online adaptive adversary.
Results: Lower Bound

**Theorem:** No randomized algorithm can achieve $O(T/n)$-convergence in $o(n^2)$ rounds against an online adaptive adversary.

- Adversary in each round $r$:
  - Sees current work distribution $\{x_u(r-1)\}_{u \in V}$
  - Chooses graph $G_r = (V, E_r)$
  - Does not see random coins used by algorithm in round $r$

- Max-Neighbor upper bound holds
Max Neighbor (round $r$)

- Node $u$ flips fair coin to decide whether to send or receive
  - If send, then $u$ sends proposal to $\text{argmax}_{v \in N(u)}(|x_v(r-1) - x_u(r-1)|)$
  - If receive, accept proposal from $\text{argmax}_{v \in S}(|x_v(r-1) - x_u(r-1)|)$ (where $S$ is neighbors of $u$ who sent a proposal to $u$)

- If $u$ accepts proposal from $v$, they are connected in round $r$
  - Set $x_u(r) = x_v(r) = \frac{1}{2}(x_u(r-1) + x_v(r-1))$
Example

= send
Example

- send
Example

= send
Example

= send
Example

- = send
Analysis Outline
Analysis Outline

• Potential function $\varphi(r) = \sum_{u,v \in V} |x_u(r) - x_v(r)|$
  
  • Initially: $\varphi(0) \leq Tn^2$
  
  • $\tau$-converged if $\varphi(r) \leq \tau$
Analysis Outline

• Potential function $\varphi(r) = \sum_{u,v \in V} |x_u(r) - x_v(r)|$
  - Initially: $\varphi(0) \leq Tn^2$
  - $\tau$-converged if $\varphi(r) \leq \tau$

• Want to show potential drops “quickly”
  - Step 1: lower bound potential drop by other function $D_r$
  - Step 2: with constant probability $D_r$ at least “maximum gap” (good round)
  - Step 3: with high probability, after $O\left(n^2 \log \left(\frac{Tn}{\tau}\right)\right)$ rounds, enough good rounds to drop potential below $\tau$
Step I
Step I

- **Lemma**: \( \varphi(r-1) - \varphi(r) \geq D_r \)
  - \( d_{u,v}(r) = |x_u(r) - x_v(r)| \)
  - \( M_r = \{\{u,v\} : u,v \text{ connected in round } r\} \)
  - \( D_r = \sum_{\{u,v\} \in M_r} d_{u,v}(r-1) \)
Step I

- **Lemma:** $\varphi(r-1) - \varphi(r) \geq D_r$
  - $d_{u,v}(r) = |x_u(r) - x_v(r)|$
  - $M_r = \{\{u,v\} : u,v \text{ connected in round } r\}$
  - $D_r = \sum_{\{u,v\} \in M_r} d_{u,v}(r-1)$

- Get $D_r$ drop directly from pairs in $M_r$, just need to show other gaps don’t increase in total
Step 1

- **Lemma:** $\varphi(r-1) - \varphi(r) \geq D_r$
  - $d_{u,v}(r) = |x_u(r) - x_v(r)|$
  - $M_r = \{\{u,v\} : u,v \text{ connected in round } r\}$
  - $D_r = \sum_{\{u,v\} \in M_r} d_{u,v}(r-1)$

- Get $D_r$ drop directly from pairs in $M_r$, just need to show other gaps don’t increase in total

- Intuition: $M_r$ is one edge

![Diagram showing edge connections](image)
Step I

- **Lemma:** $\varphi(r-1) - \varphi(r) \geq D_r$
  - $d_{u,v}(r) = |x_u(r) - x_v(r)|$
  - $M_r = \{\{u,v\} : u,v$ connected in round $r\}$
  - $D_r = \sum_{\{u,v\} \in M_r} d_{u,v}(r-1)$

- Get $D_r$ drop directly from pairs in $M_r$, just need to show other gaps don’t increase in total

- Intuition: $M_r$ is one edge
Bounding $D_r$

- **Lemma:** $D_r \geq t_{\max}(r-1) / O(\log n)$ with constant probability
  - $d_{u,v}(r) = |x_u(r) - x_v(r)|$
  - $t_{\max}(r) = \max_{u,v \in V} d_{u,v}(r)$
Bounding $D_r$

- **Lemma:** $D_r \geq t_{\text{max}}(r-1) / O(\log n)$ with constant probability
  - $d_{u,v}(r) = |x_u(r) - x_v(r)|$
  - $t_{\text{max}}(r) = \max_{u,v \in V} d_{u,v}(r)$
Bounding $D_r$

- **Lemma**: $D_r \geq t_{\max}(r-1) / O(\log n)$ with constant probability
  - $d_{u,v}(r) = |x_u(r) - x_v(r)|$
  - $t_{\max}(r) = \max_{u,v \in V} d_{u,v}(r)$

\[
\sum_{i} d_{u_i, u_{i+1}}(r - 1) \geq d_{u_{\max}, v_{\max}}(r - 1) = t_{\max}(r - 1)
\]
Bounding $D_r$

- **Lemma:** $D_r \geq t_{\max}(r-1) / O(\log n)$ with constant probability
  - $d_{u,v}(r) = |x_u(r) - x_v(r)|$
  - $t_{\max}(r) = \max_{u,v \in V} d_{u,v}(r)$

\[
\sum_i d_{u_i,u_{i+1}}(r-1) \geq d_{u_{\max},v_{\max}}(r-1) = t_{\max}(r-1)
\]

- Would be great if each edge was in matching independently with constant probability, but not true
Bounding $D_r$

- Fix edge $\{u_i, u_{i+1}\}$. With constant probability there is edge $\{v, w\} \in M_r$ s.t.
  - $v, w$ at distance at most 3 from $u, v, and$
  - $d_{v, w}(r - 1) \geq d_{u_i, u_{i+1}}(r - 1)$

\[
\begin{array}{cccccccc}
12 & 16 & 14 & 30 & 45 & 36 & 15 & 50 \\
\end{array}
\]

$u_{\text{max}} = u_0$ $u_1$ $u_2$ $u_3$ $u_4$ $u_5$ $u_6$ $v_{\text{max}} = u_7$
Bounding $D_r$

• Fix edge $\{u_i, u_{i+1}\}$. With constant probability there is edge $\{v, w\} \in M_r$ s.t.
  
  • $v, w$ at distance at most 3 from $u, v,$ and
  
  • $d_{v,w}(r - 1) \geq d_{u_i,u_{i+1}}(r - 1)$
Bounding $D_r$

- Fix edge $\{u_i, u_{i+1}\}$. With constant probability there is edge $\{v, w\} \in M_r$ s.t.
  - $v, w$ at distance at most 3 from $u, v$, and
  - $d_{v, w}(r - 1) \geq d_{u_i, u_{i+1}}(r - 1)$

- Independent for $i, i'$ with $|i' - i| > 6$ (distance bound)
- Constant prob. of logarithmic fraction of full path
Putting it Together

• Potential drop at least $D_r$

• With constant probability, $D_r \geq t_{\text{max}}(r-1) / O(\log n)$

• Potential at round $r$ at most $n^2 t_{\text{max}}(r-1)$

• So after about $n^2$ rounds, potential small enough to guarantee $\tau$-convergence
Lower Bound

• Claim (informal): any algorithm which in each round uses a matching needs $\Omega(n^2)$ rounds to get $(T/n)$-convergence
Lower Bound

- Claim (informal): any algorithm which in each round uses a matching needs $\Omega(n^2)$ rounds to get $(T/n)$-convergence.
Lower Bound

• Claim (informal): any algorithm which in each round uses a matching needs \( \Omega(n^2) \) rounds to get \((T/n)\)-convergence.

\[
\begin{align*}
n & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 \\
\end{align*}
\]

• If ALG used \( \{i, i+1\} \) in round r-1, swap (if necessary) so one with smaller work is on the right.
Lower Bound

- Claim (informal): any algorithm which in each round uses a matching needs $\Omega(n^2)$ rounds to get $(T/n)$-convergence.

- If ALG used \( \{i, i+1\} \) in round \( r-1 \), swap (if necessary) so one with smaller work is on the right.
Lower Bound

• Claim (informal): any algorithm which in each round uses a matching needs $\Omega(n^2)$ rounds to get $(T/n)$-convergence

• If ALG used \( \{i, i+1\} \) in round \( r-1 \), swap (if necessary) so one with smaller work is on the right
Lower Bound

• Claim (informal): any algorithm which in each round uses a matching needs $\Omega(n^2)$ rounds to get $(T/n)$-convergence

• If ALG used $\{i, i+1\}$ in round r-1, swap (if necessary) so one with smaller work is on the right
Lower Bound

• Claim (informal): any algorithm which in each round uses a matching needs $\Omega(n^2)$ rounds to get $(T/n)$-convergence

• If ALG used \{i, i+1\} in round r-1, swap (if necessary) so one with smaller work is on the right
Lower Bound

• Claim (informal): any algorithm which in each round uses a matching needs $\Omega(n^2)$ rounds to get $(T/n)$-convergence

• If ALG used $\{i, i+1\}$ in round $r-1$, swap (if necessary) so one with smaller work is on the right
Lower Bound

• Claim (informal): any algorithm which in each round uses a matching needs $\Omega(n^2)$ rounds to get $(T/n)$-convergence

• If ALG used $\{i, i+1\}$ in round r-1, swap (if necessary) so one with smaller work is on the right
Lower Bound

• Claim (informal): any algorithm which in each round uses a matching needs $\Omega(n^2)$ rounds to get $(T/n)$-convergence.

- If ALG used $\{i, i+1\}$ in round $r-1$, swap (if necessary) so one with smaller work is on the right.
Lower Bound

• Claim (informal): any algorithm which in each round uses a matching needs $\Omega(n^2)$ rounds to get $(T/n)$-convergence

• If ALG used \{i, i+1\} in round r-1, swap (if necessary) so one with smaller work is on the right
Lower Bound

• Claim (informal): any algorithm which in each round uses a matching needs $\Omega(n^2)$ rounds to get $(T/n)$-convergence

\[
\begin{align*}
3n/4 & \quad 3n/16 & \quad n/32 & \quad n/32 & \quad 0 & \quad 0 & \quad 0 & \quad 0
\end{align*}
\]

• If ALG used $\{i, i+1\}$ in round $r-1$, swap (if necessary) so one with smaller work is on the right
  • Lemma: best strategy for ALG is to split work equally across each edge it uses (EQUAL)
  • EQUAL takes $\Omega(n^2)$ rounds before significant weight on node $n$
Conclusion

• Load balancing upper and lower bounds:
  • Local (no global coordination)
  • At most one connections / node / round (matching)
  • Dynamic networks
  • Provably converges quickly (w.h.p.)

• Lots of interesting questions left!
  • Theory of dynamic graphs
  • Connection to smoothed analysis
  • Logarithmic gap between upper and lower bounds
  • Practice…
Thanks!