Balanced Allocations on Graphs

Krishnaram Kenthapadi and Rina Panigrahy

Stanford University

Balls into Bins with 2 Choices on Graphs

- Overview
- A Positive Result
- A Negative Result

Choosing Bins in Groups

Balls into Bins with 2 Choices on Graphs

- **Classic Setup for 2-Choice Load Balancing**

- **Balls into Bins with 2 Choices on Graphs**

Balls into Bins with 2 Choices on Graphs

- **Classic Setup for 2-Choice Load Balancing**
  - n balls and n bins
  - To insert a ball, choose 2 bins at random
  - Insert the ball into the least-loaded bin
  - Max load: $\Theta(\log\log n)$ whp $(1 - 1/n^{\Omega(1)})$
- **Balls into Bins with 2 Choices on Graphs**

Balls into Bins with 2 Choices on Graphs

- **Classic Setup for 2-Choice Load Balancing**
  - n balls and n bins
  - To insert a ball, choose 2 bins at random
  - Insert the ball into the least-loaded bin
  - Max load: $\Theta(\log \log n)$ whp $(1 - 1/n^{\Omega(1)})$
- **Balls into Bins with 2 Choices on Graphs**
  - Bins are vertices in an undirected graph $G = (V, E)$
  - To insert a ball, choose an edge in E at random
  - Insert the ball into the least-loaded bin
  - Max load compared to classic setup?

Balls into Bins with 2 Choices on Graphs

- **Classic Setup for 2-Choice Load Balancing**
  - n balls and n bins
  - To insert a ball, choose 2 bins at random
  - Insert the ball into the least-loaded bin
  - Max load: $\Theta(\log \log n)$ whp $(1 - 1/n^{\Omega(1)})$
- **Balls into Bins with 2 Choices on Graphs**
  - Bins are vertices in an undirected graph $G = (V, E)$
  - To insert a ball, choose an edge in E at random
  - Insert the ball into the least-loaded bin
  - Max load compared to classic setup?
    - Classic setup corresponds to a complete graph

Building Intuition: Bad Graphs for Max Load

- **What if we take away too many edges from $\mathbf{K_n}$?**

Building Intuition: Bad Graphs for Max Load

- **What if we take away too many edges from $\mathbf{K_n}$?**
  - Poor load balancing
  - Max load at least as bad as standard (1-choice) balls into bins
  - $\Omega\left(\frac{\log n}{\log\log n}\right)$

Building Intuition: Bad Graphs for Max Load

- **What if we take away too many edges from $\mathbf{K_n}$?**
  - Poor load balancing
  - Max load at least as bad as standard (1-choice) balls into bins
  - $\Omega\left(\frac{\log n}{\log\log n}\right)$
- **What if large variation in vertex degree?**
  - Example: star graph with n vertices (min degree 1, max degree $n-1$)

Building Intuition: Bad Graphs for Max Load

- **What if we take away too many edges from $\mathbf{K_n}$?**
  - Poor load balancing
  - Max load at least as bad as standard (1-choice) balls into bins
  - $\Omega\left(\frac{\log n}{\log\log n}\right)$
- **What if large variation in vertex degree?**
  - Example: star graph with n vertices (min degree 1, max degree $n-1$)
  - If leaf vertex has large load, then so does center!

Building Intuition: Bad Graphs for Max Load

- **What if we take away too many edges from $\mathbf{K_n}$?**
  - Poor load balancing
  - Max load at least as bad as standard (1-choice) balls into bins
  - $\Omega\left(\frac{\log n}{\log\log n}\right)$
- **What if large variation in vertex degree?**
  - Example: star graph with n vertices (min degree 1, max degree $n-1$)
  - If leaf vertex has large load, then so does center!
  - Power of 2 choices gone: max load roughly like 1-choice setup

Power of 2 Choices on Graphs: Results

- For n^{ϵ}-regular graphs, maximum load is $\log\log n + O(1/\epsilon) + O(1)$ for large enough ϵ.
- For non-regular graphs max load is $\Omega\left(\frac{\log n}{\log\log n}\right)$ even when minimum degree is as large as n^{ϵ}, $\epsilon < 1$.

The Upper Bound

Theorem

For a Δ-regular graph, the maximum load is
$$\log\log n + O\left(\frac{\log n}{\log(\Delta/\log^4 n)}\right) + O(1)$$

The Upper Bound

Theorem

For a Δ-regular graph, the maximum load is

$$\log\log n + O\left(\frac{\log n}{\log(\Delta/\log^4 n)}\right) + O(1)$$

- Taking $\Delta = n^\epsilon$ where $\epsilon > 8\log\log n/\log n$ yields $\log\log n + O(1/\epsilon) + O(1)$

The Upper Bound

Theorem

For a Δ-regular graph, the maximum load is
$$\log\log n + O\left(\frac{\log n}{\log(\Delta/\log^4 n)}\right) + O(1)$$

- Taking $\Delta = n^\epsilon$ where $\epsilon > 8\log\log n/\log n$ yields $\log\log n + O(1/\epsilon) + O(1)$
- **Can we apply the layered induction method?**

The Upper Bound

Theorem

For a Δ-regular graph, the maximum load is

$$\log\log n + O\left(\frac{\log n}{\log(\Delta/\log^4 n)}\right) + O(1)$$

- Taking $\Delta = n^\epsilon$ where $\epsilon > 8\log\log n/\log n$ yields $\log\log n + O(1/\epsilon) + O(1)$
- **Can we apply the layered induction method?**
  - No. p_{i+1} is no longer p_i^2. If one end of an edge has i balls, the other end is more likely to have i balls.
    (Can it have less than i?)

The Upper Bound

Theorem

For a Δ-regular graph, the maximum load is
$$\log\log n + O\left(\frac{\log n}{\log(\Delta/\log^4 n)}\right) + O(1)$$

- Taking $\Delta = n^\epsilon$ where $\epsilon > 8\log\log n/\log n$ yields $\log\log n + O(1/\epsilon) + O(1)$
- **Can we apply the layered induction method?**
  - No. p_{i+1} is no longer p_i^2. If one end of an edge has i balls, the other end is more likely to have i balls.
    (Can it have less than i?)
- **Argue that "bad" sequences of events (inserts) leading to high load at any bin do not occur (whp)**
  - Challenge: Identify the bad (witness) sequences and bound the probab. of their union

The Witness Tree Method: schema

- **Define a representation of witness sequences as trees**
  - A witness sequence is a sequence of events leading to high load at some bin
  - This is the creative part

The Witness Tree Method: schema

- **Define a representation of witness sequences as trees**
  - A witness sequence is a sequence of events leading to high load at some bin
  - This is the creative part
- **Show that all possible witness trees do not occur (whp)**
  - Let $T_1, T_2, \ldots$ denote the events correponding to witness trees for a given graph G
  - Show $\mathbf{Pr}[\cup_i T_i] \leq \frac{1}{n^{\Omega(1)}}$
  - union bound over all witness trees
    - bound the number of possible witness trees
    - bound the probability of occurence of any particular tree

The Witness Tree Method: schema

- **Define a representation of witness sequences as trees**
  - A witness sequence is a sequence of events leading to high load at some bin
  - This is the creative part
- **Show that all possible witness trees do not occur (whp)**
  - Let $T_1, T_2, \ldots$ denote the events correponding to witness trees for a given graph G
  - Show $\mathbf{Pr}[\cup_i T_i] \leq \frac{1}{n^{\Omega(1)}}$
  - union bound over all witness trees
    - bound the number of possible witness trees
    - bound the probability of occurence of any particular tree
- **The witness tree method is not particular to balls and bins problems**

The Witness Tree Method: schema

- **Define a representation of witness sequences as trees**
  - A witness sequence is a sequence of events leading to high load at some bin
  - This is the creative part
- **Show that all possible witness trees do not occur (whp)**
  - Let $T_1, T_2, \ldots$ denote the events correponding to witness trees for a given graph G
  - Show $\mathbf{Pr}[\cup_i T_i] \leq \frac{1}{n^{\Omega(1)}}$
  - union bound over all witness trees
    - bound the number of possible witness trees
    - bound the probability of occurence of any particular tree
- **The witness tree method is not particular to balls and bins problems**
- **Why not a witness graph?**

Deriving the Upper Bound using the Witness Tree Method

- Root: a bin with too many balls: $l + c$
- Each of the top l balls had an alternate bin choice
- Set the l alternate choices to be the children of the root
- Recurse for each of the l children of root
- Stopping condition: leaf nodes have load c
- **Observations**
    - Every edge in tree is in original graph
    - For a parent with load x, the ith balls from top has height $x - i + 1$
    - Alternative bin (child) must have $x - i$ balls contributing to the bad sequence
    - The ith child of root has load $\geq l + c - i$ and $l - i$ children
    - Cycles may be problem, ignore for now
    - Number of nodes in tree is $f(l) = 1 + \sum_{i=1}^{l} f(l - i) = 2^l$

Deriving the Upper Bound using the Witness Tree Method: Top-level argument

- Show there cannot exist a set of $2^l = \log n$ bins connected by a tree where each edge represents a distinct ball and each bin has an additional c balls (whp)
  - $2^l = \log n \Rightarrow l = \log \log n$ (max load $\log \log n + c$)
- **The union bound**
  - A bound the number of possible trees on $\log n$ vertices × a bound the probability of such a tree

Deriving the Upper Bound using the Witness Tree Method: The Union Bound

- Number of different (non-isomorphic) tree shapes on m nodes is at most 4^m
- Number of trees of a given shape is at most $n\Delta^{m-1}$
- Probab. of a particular tree of any shape is at most $n^{m-1}(2/n\Delta)^{m-1}$
- Prob. that each bin has additional c balls is at most $(e/c)^{cm}$
- The prob. of union of all possible witness trees is at most the product of the above: $4^m n2^{m-1}(e/c)^{cm} \leq n[8(e/c)^c]^m$
  - The $n2^{m-1}$ factor is the product of the second and third items
- $8(e/c)^c = 1/4$ gives probability $1/n^2$ when $m = \log n$

Deriving the Upper Bound using the Witness Tree Method: The $n2^{m-1}$ factor

- Number of trees of a given shape on m nodes is number ways to choose the m bins out of n: at most $n\Delta^{m-1}$
  - n ways to choose root, and Δ ways to choose each of the $m-1$ edges
- Prob. that a particular excess ball hits a particular edge in graph is $2/(n\Delta)$
  - Number of edges in Δ-regular graph is $n\Delta/2$
  - Edges are chosen at random
- Prob. of an excess ball in each of the $m-1$ tree edges is at most $(2/n\Delta)^{m-1}$
- The product $n^{m-1}(2/n\Delta)^{m-1} \times n\Delta^{m-1} = n2^{m-1}$

Deriving the Upper Bound using the Witness Tree Method: The $(e/c)^{cm}$ factor

- Total additional balls cm
  - each bin in tree has to have c additional balls
- The cm balls can be chosen in $\binom{n}{cm}$ ways
- The cm balls choose tree edges with prob. $(\frac{m\Delta}{n\Delta})^{cm}$
- The probability that each bin has additional c balls is at most $\binom{n}{cm}(\frac{m\Delta}{n\Delta})^{cm} \leq (\frac{en}{cm})^{cm}(\frac{m\Delta}{n\Delta})^{cm}$

Deriving the Upper Bound using the Witness Tree Method:Cycle-producing edges

- Cycle producing edges can result in load larger than $l + c$ at some bins.
- Witness trees as above but with $\log^2 n$ bins and $p = \frac{k \log n}{\log(\Delta/\log^4 n)}$ excess edges do not occur (whp $1/n^{k-1}$ for large enough k).
- Construct a tree starting $l + c + p$ balls at the root
  - expand the edges of top p balls in first tree level and recurse
  - Number of nodes 2^l in each subtree.
  - Total: $2^l p + 1 = p \log n + 1 \leq k \log^2 n$
  - thus max load is at most $l + p + c = \log\log n + O(\frac{\log n}{\log(\Delta/\log^4 n)}) + O(1)$ whp $\square$.

A Lower Bound

Lemma

For every $\epsilon < 1$ there exist non-regular graphs with min degree n^ϵ such max load is $\Omega\left(\frac{\log n}{\log\log n}\right)$ whp

Example: Complete bipartite graph

- left partition $n - \Delta$ bins, right partition: Δ bins
- Essential idea:
  - Load will increase quickly in the smaller partition
  - Power of 2-choice gone.

A Lower Bound: Top-level Argument

- Break n insertions into $n/\log n$ phases of $\log n$ balls
- When $\Delta = n^{\epsilon}$, for up to $\Omega\left(\frac{\log n}{\log\log n}\right)$ phases:
  - After each phase i, each vertex in the smaller partition will have at least i balls in each bin
  - A fraction $1/(4\log n)^{i}$ of larger partition will also have load i
  - Inductive proof

Overview and Results

- Bins are arranged in a line. Bins within distance d are adjacent in G
- Group n bins into cn/d disjoint superbins of d/c consecutive bins, $c \geq 2$
- Pick 2 superbins at random and select superbin with lesser **total load**
- Ties broken to the left
- Place ball in least loaded bin within selected superbin
- max load is $\frac{\log\log n}{d \ln \phi_c}$, ϕ_c is a constant
- Similar result can be obtained with d ranndom choices and breaking ties to the left.