SAND2014-17450PE

Generalized Hypergraph Matching via Iterated Packing and Local Ratio

Ojas Parekh
Sandia Labs

David Pritchard
University of Southern California

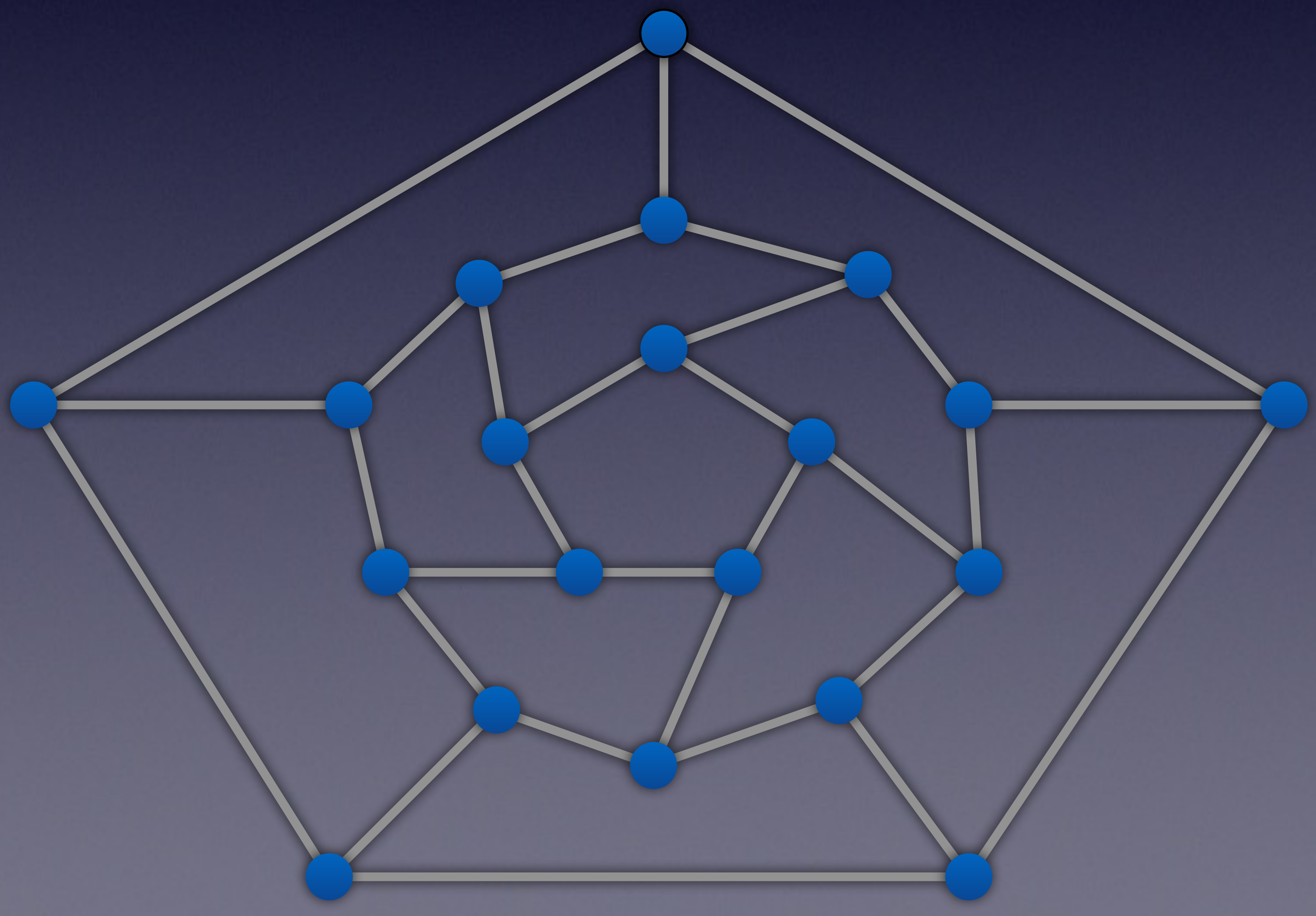
Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Approximation Algorithms

- Efficient heuristic with bound on worst case performance
- Trades off efficiency for performance
- α-approximation earns profit at least OPT/α on every instance

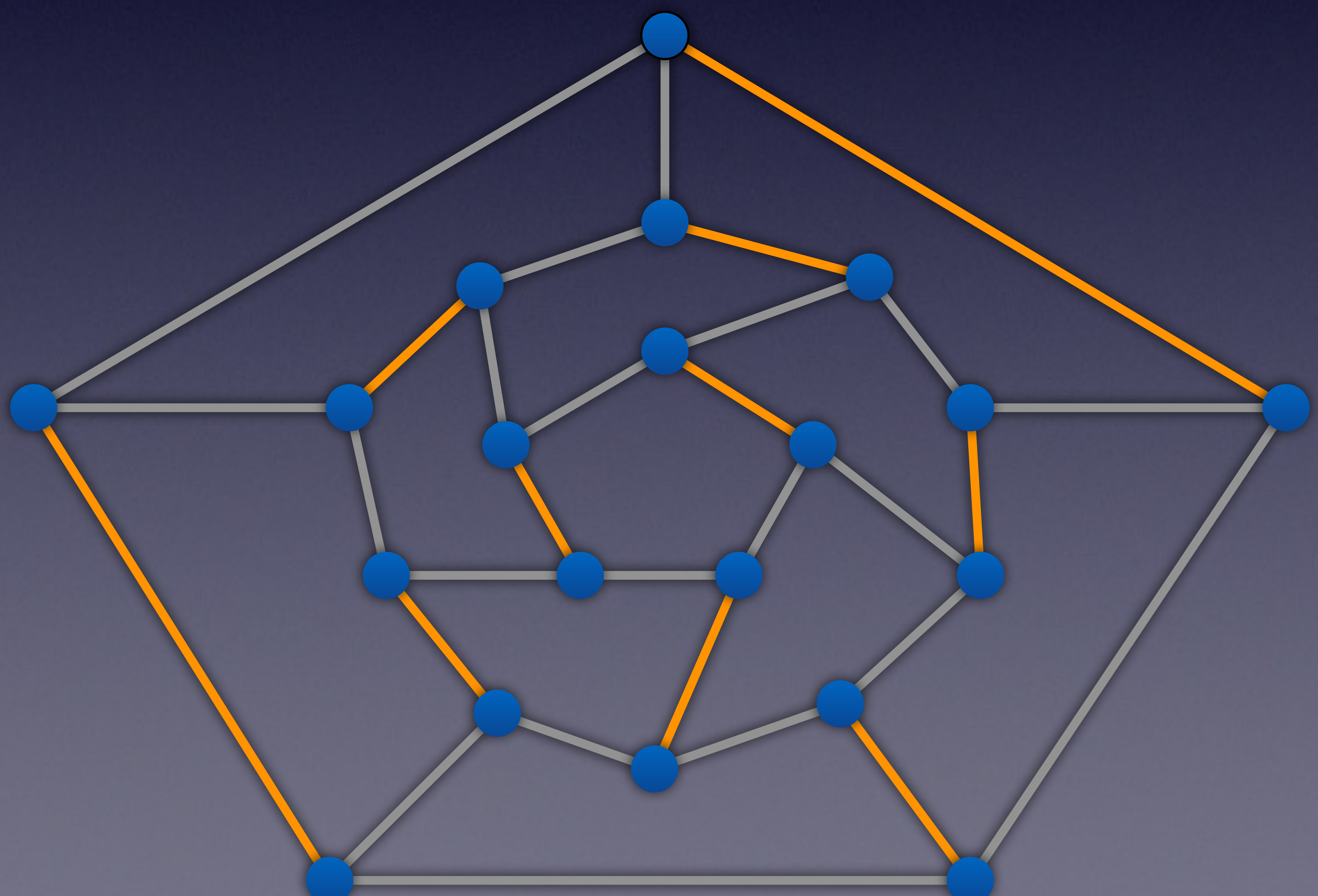
Matching

- Input: (weighted) undirected graph G = (V, E).
- $M \subseteq E$ is a matching if each node appears in at most edge in M.
- Max matching: find a max cardinality (weight) matching.



Matching

- Input: (weighted) undirected graph G = (V, E).
- M ⊆ E is a matching if each node appears in at most edge in M.
- Max matching: find a max cardinality (weight) matching.



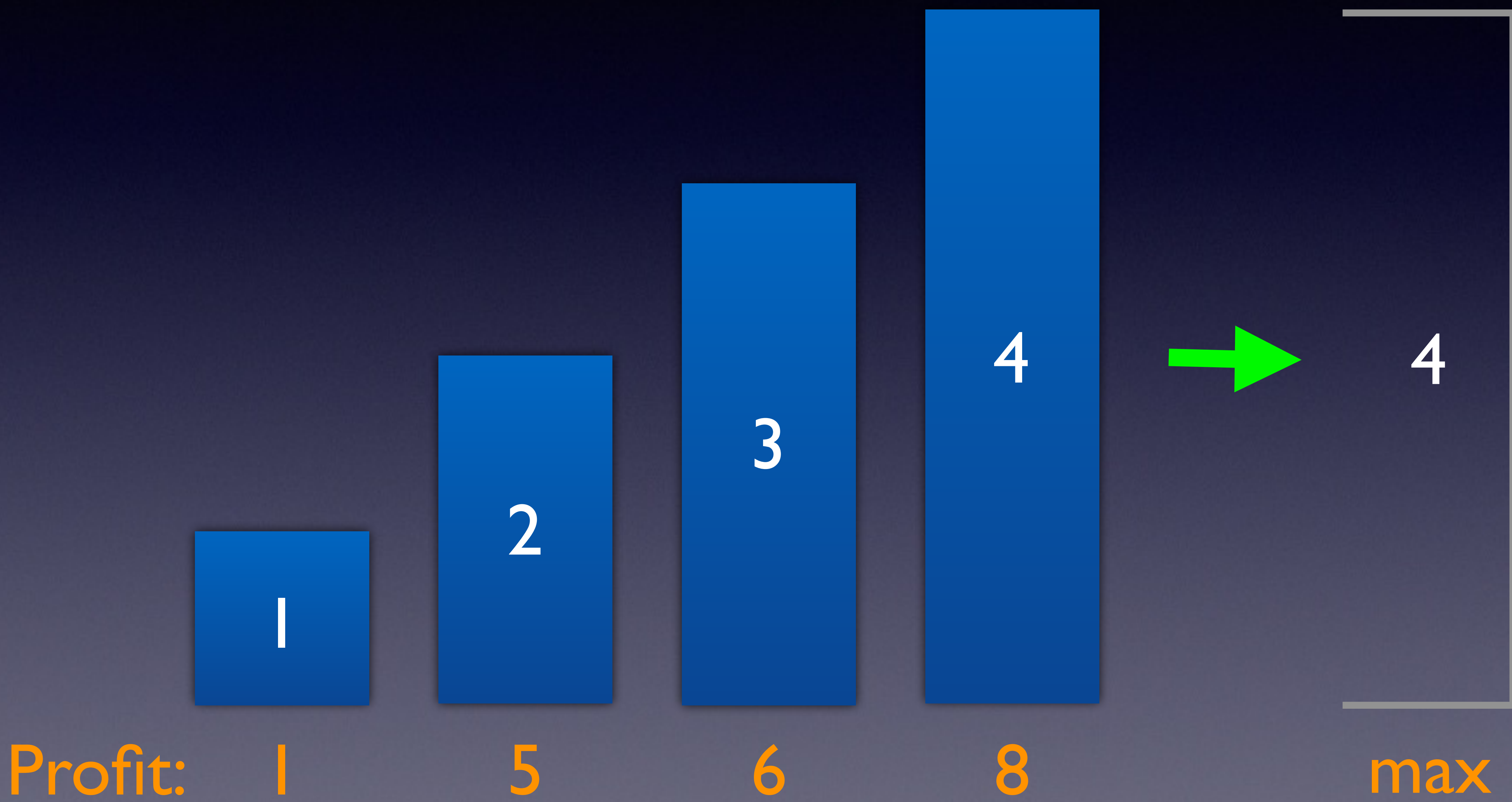
Matching: Extensions

- Vertex capacties: b-matching
  - Subgraph with at most b_v edges at v
- Edge capacities
  - Copies of edges may be selected

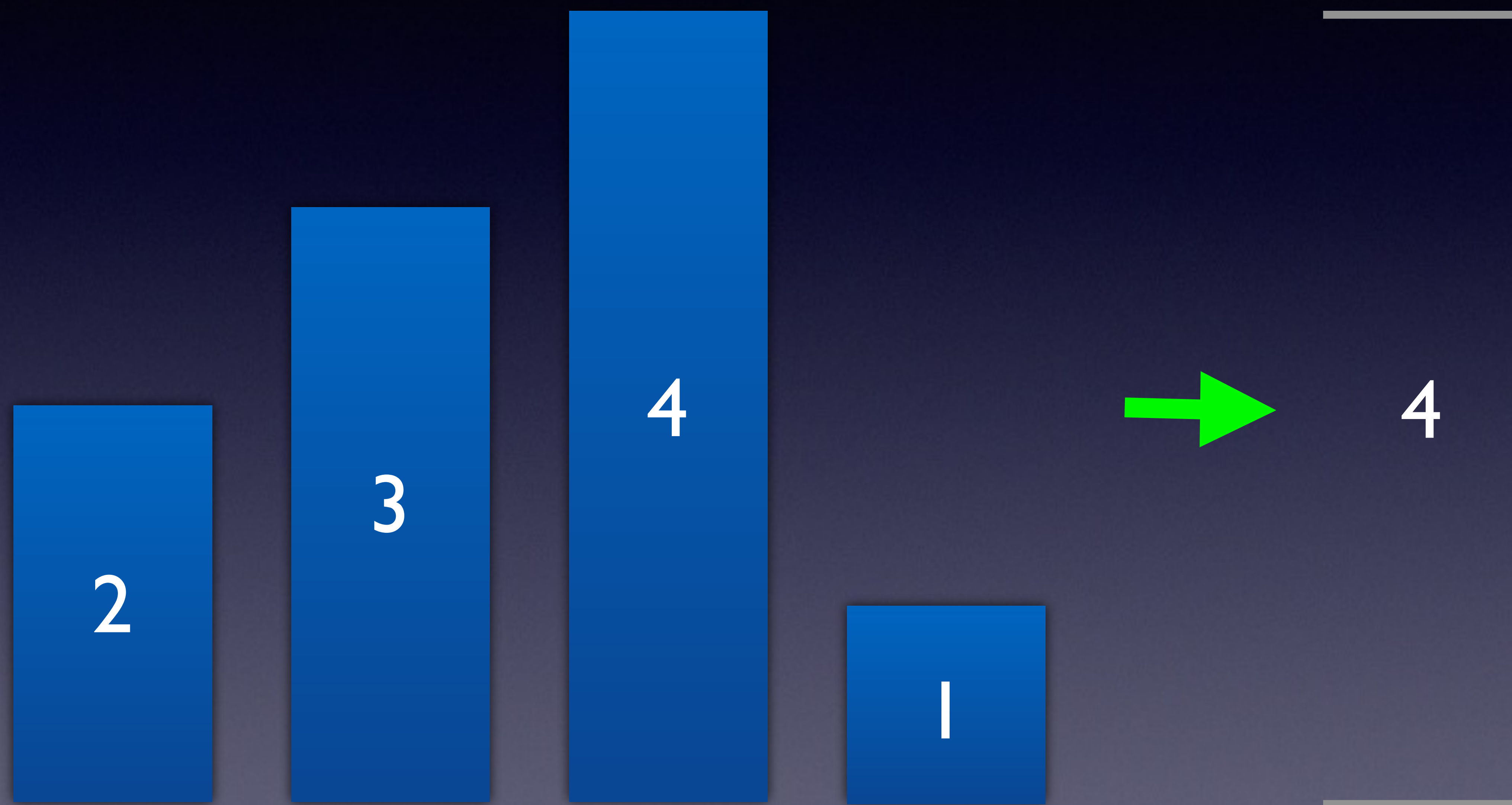
Demands

- Facilitate modeling of:
  - All or nothing scenarios
  - Economies of scale
  - Transportation: indivisible shipments
  - Scheduling: unrelated machines
  - *Knapsack* is prototypical problem

Knapsack

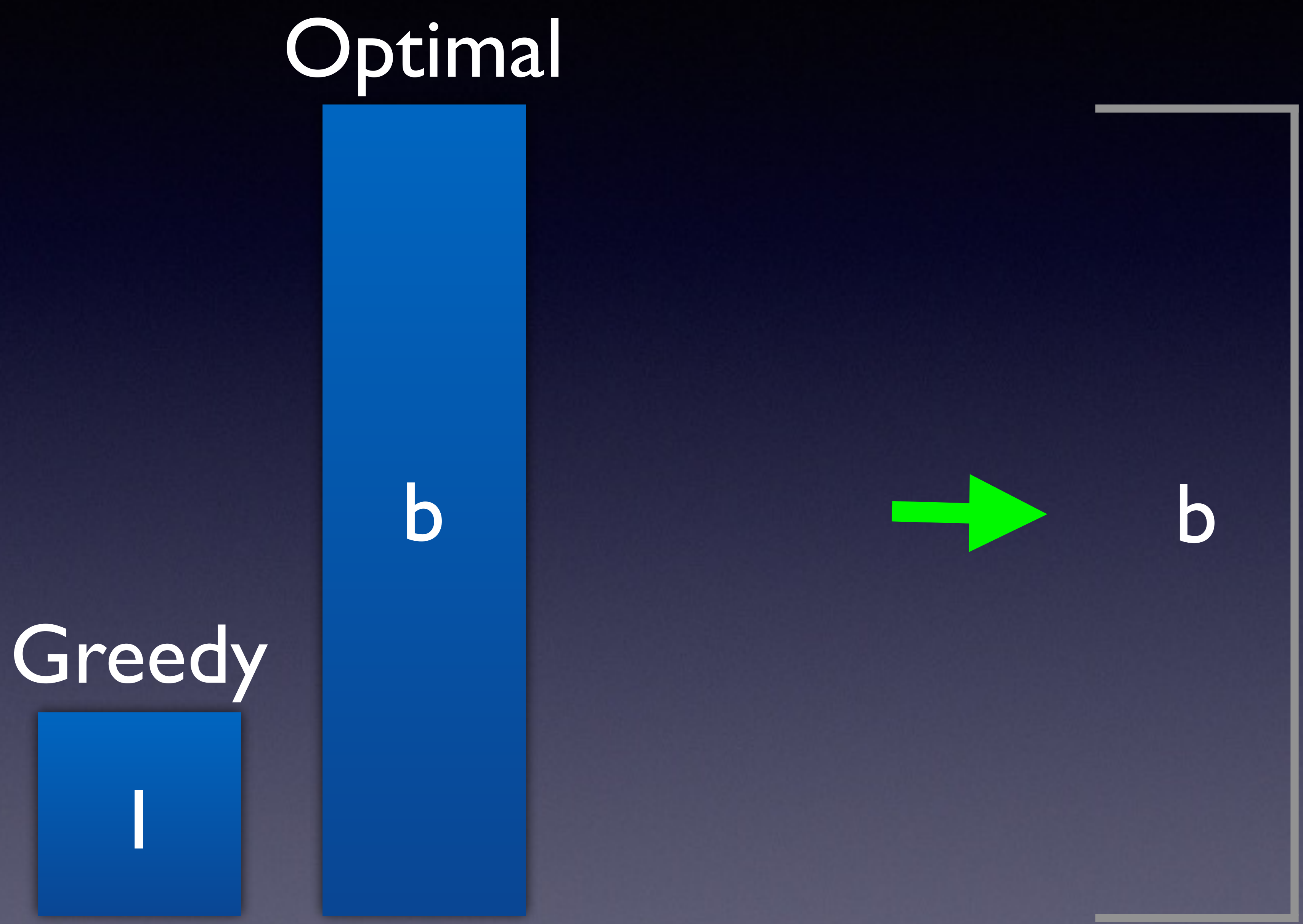


Knapsack: Greedy Algorithm



Profit:	5	6	8	1	max
P/S:	2.5	2	2	1	

Knapsack: Greedy Performs Poorly



Profit:	2	b	max
P/S:	2	1	

Knapsack: Approximation Algorithms

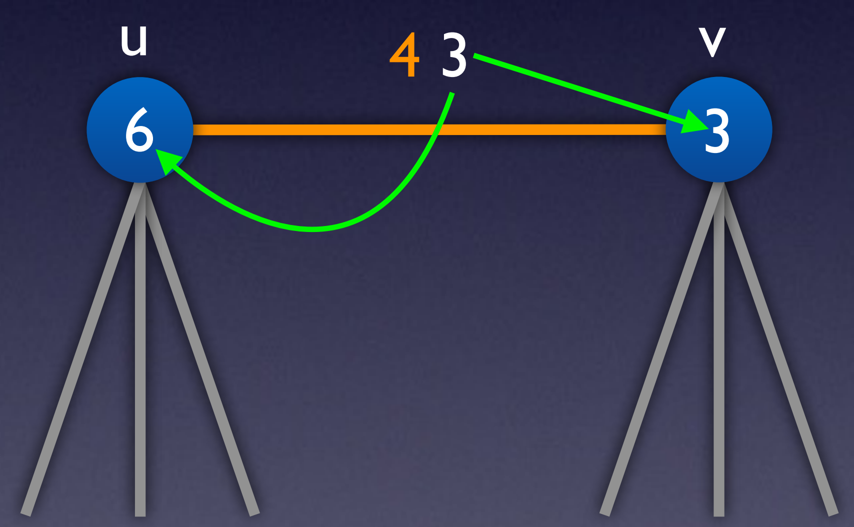
- NP-complete
- Modified greedy: 2-approx
- Natural LP relax: 2 integrality gap
- Dynamic programming: (F)PTAS

Our Problem

- Generalize matching in two directions
  - demand matching: knapsack at each vertex
  - k-hypergraph matching: each hyperedge $\leq$ k vertices
  - k-hypergraph demand matching: both

Demands in Graphs

Edges: profits (p) & demands (d)
Vertices: capacities (b)



Selecting above edge earns profit of 4 and leaves 3 and 0 units of capacity at u and v, respectively.

Demand Matching

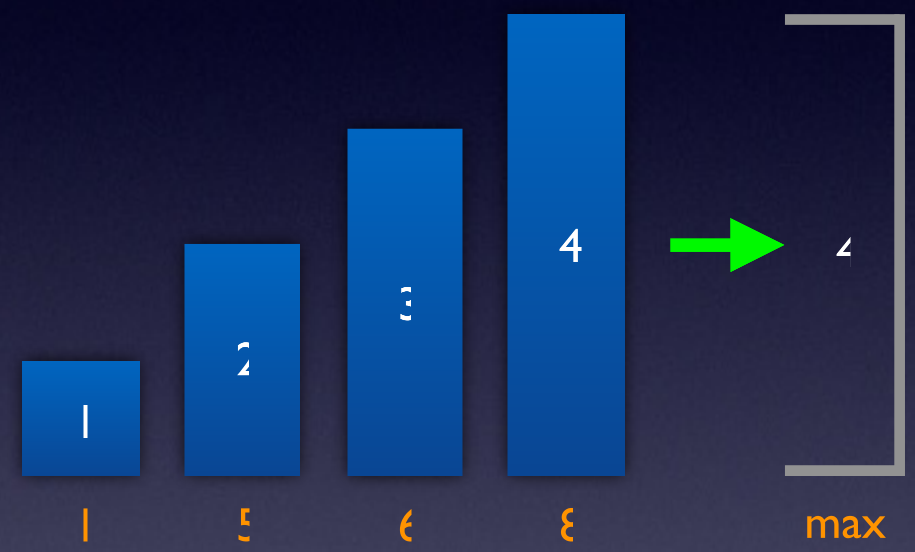
- M $\subseteq$ E is a demand matching if for each vertex v:

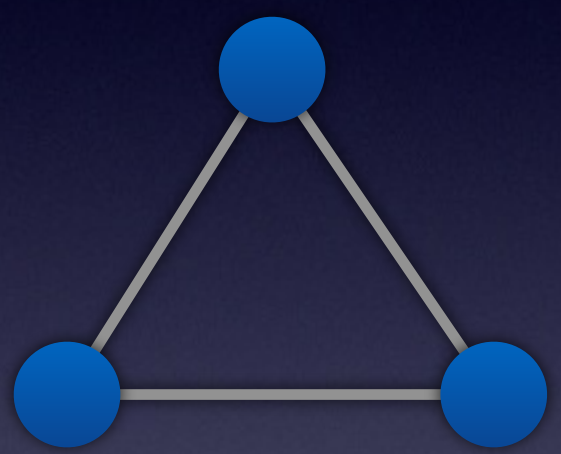
  $\sum_{e \in M \cap \delta(v)} d_e \leq b_v$

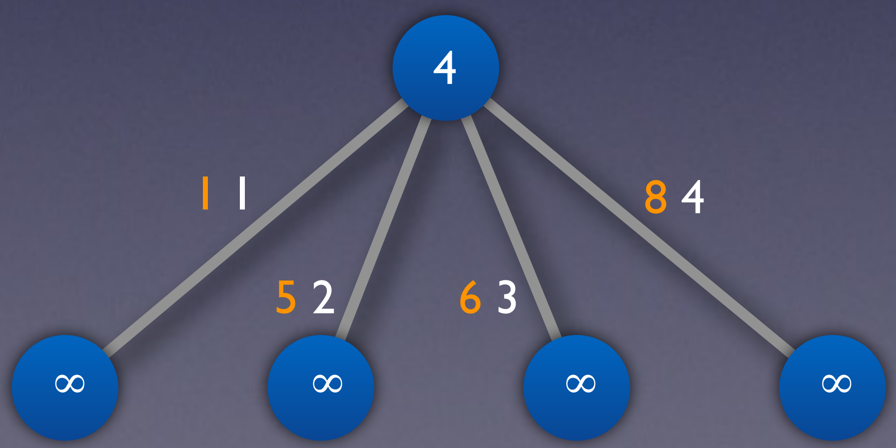
- Goal: find max profit demand matching
- Introduced by Shepherd & Vetta 2002
- APX-complete generalization of both matching and knapsack

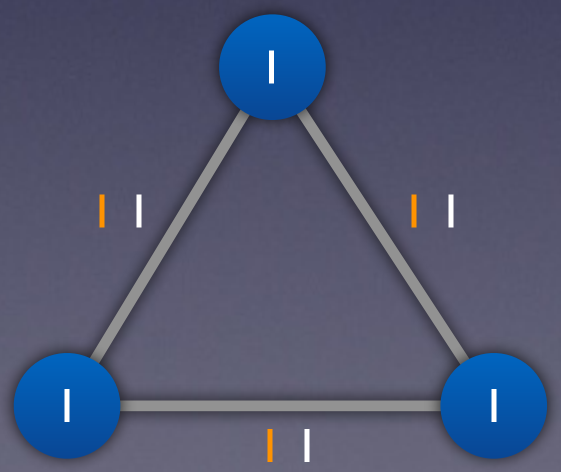
Demand Matching

- Generalizes matching



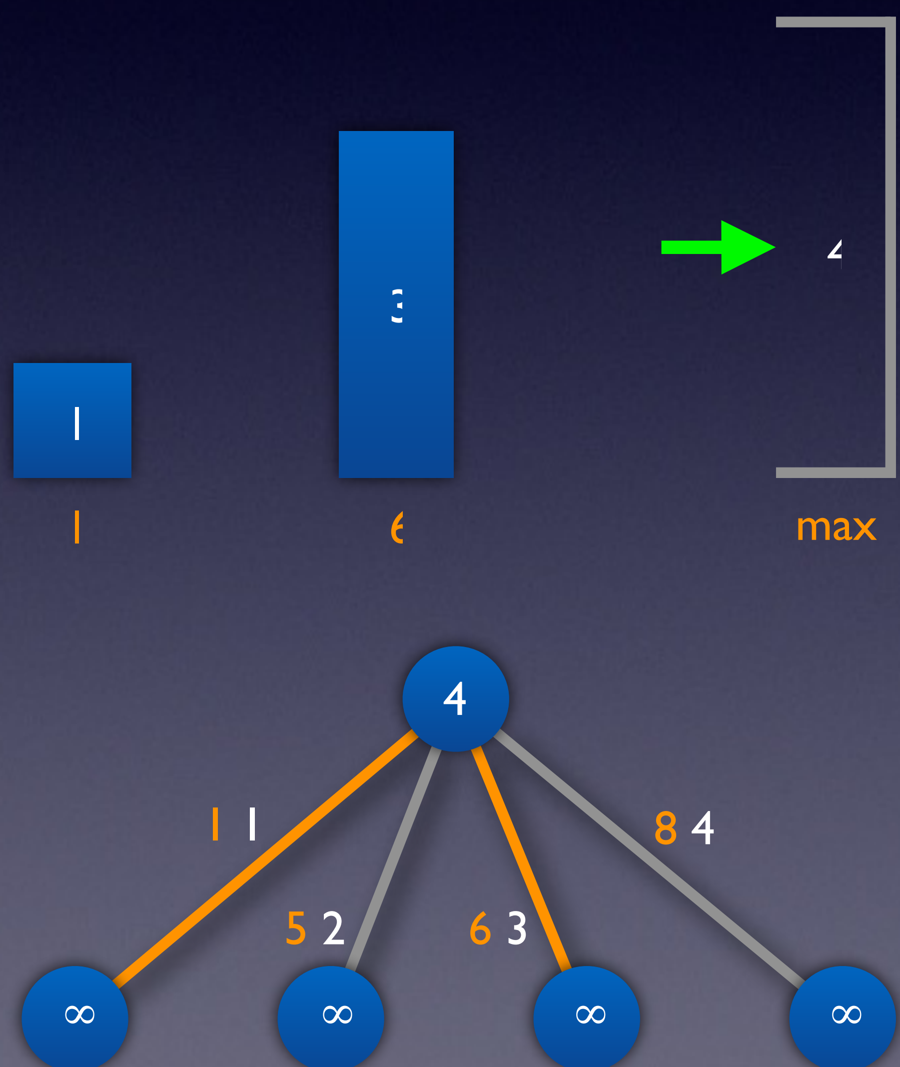




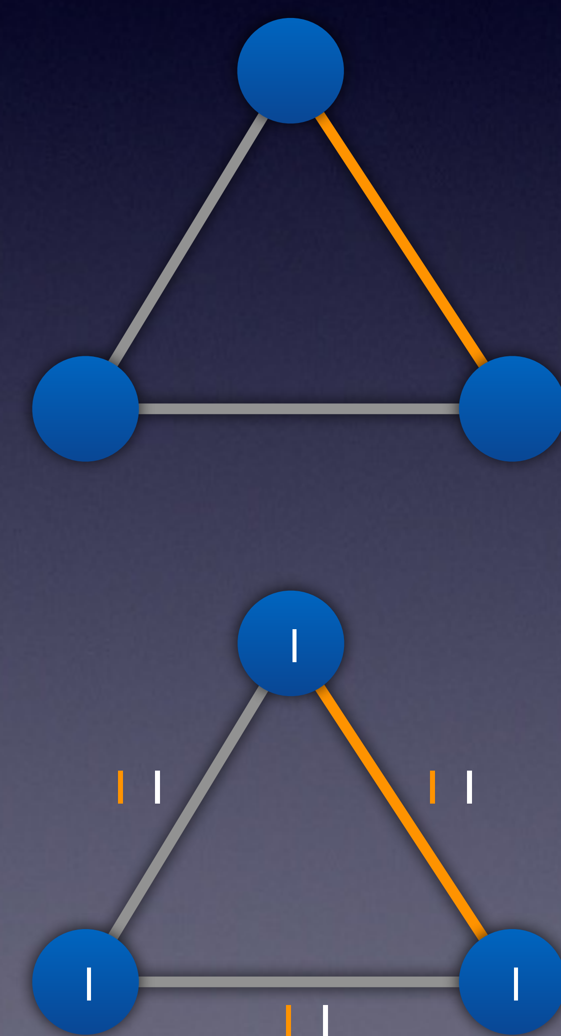


Demand Matching

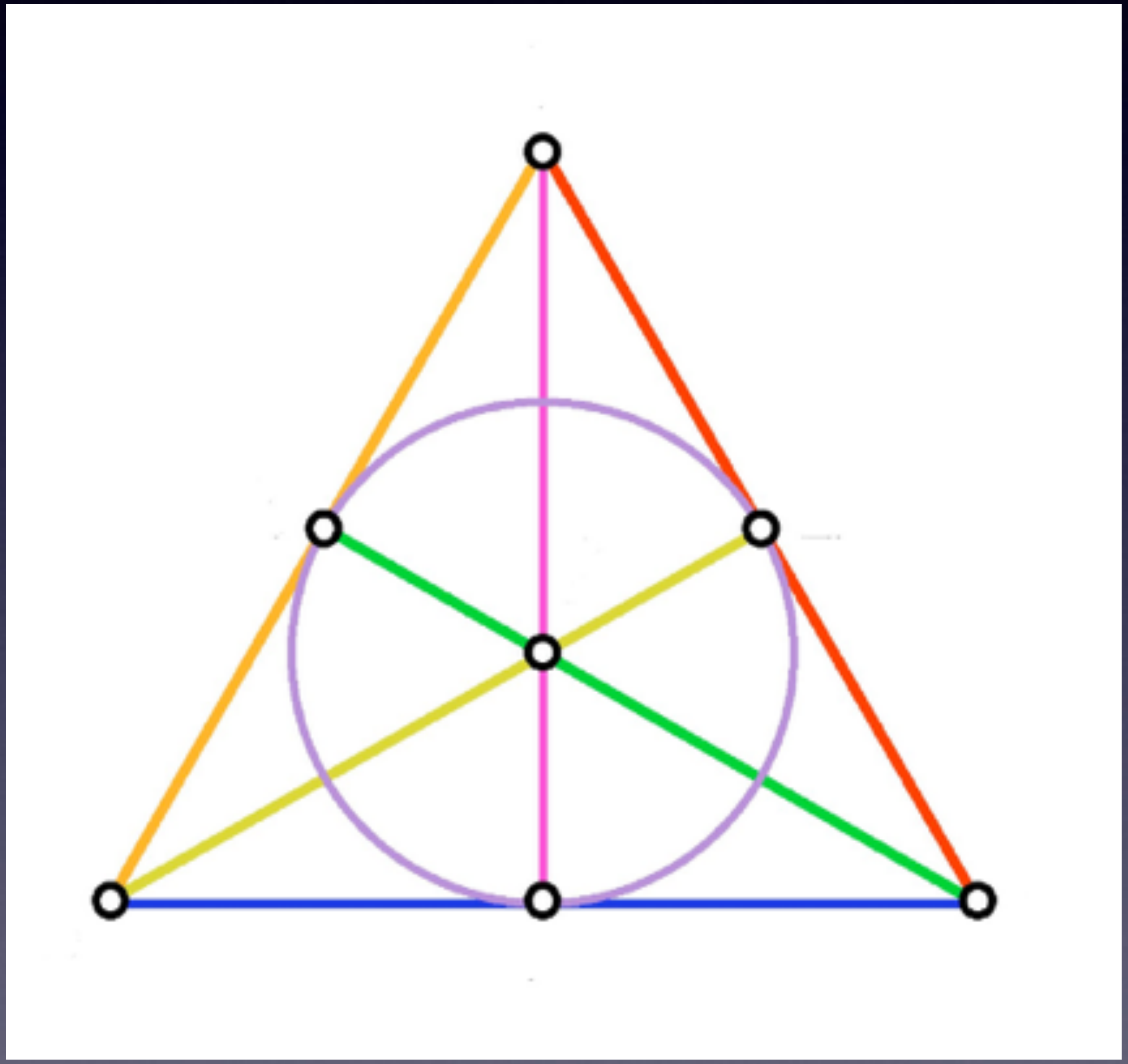
- Generalizes knapsack



- Generalizes matching



k-Hypergraph Matching

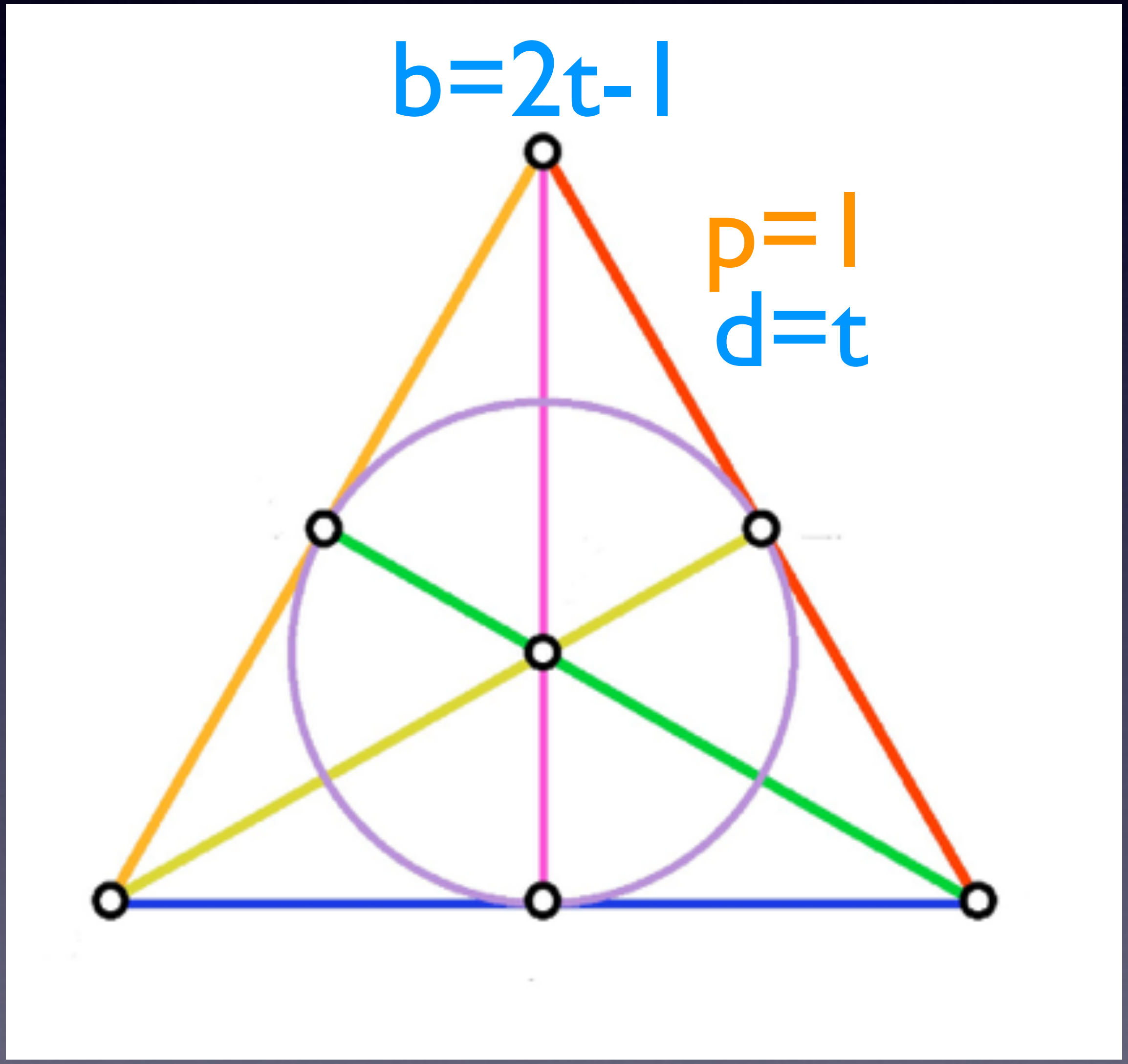


- Each hyperedge has $\leq$ k vertices
- Find: max weight collection of non-intersecting hyperedges
- E.g. k=3, |V|=7, |E|=7

k-Hypergraph Matching

- NP-complete
- Natural LP: $k-1+1/k$ integrality gap
- No $\Omega(k/\log(k))$ unless P=NP [Hazan, Safra, & Schwartz 2006]

k-Hypergraph Demand Matching



- Vertex v: capacity b_v
- Hyperedge e: demand d_e
- $M \subseteq E$ is a k-HDM if: $\sum_{e \in M \cap \delta(v)} d_e \leq b_v$
- Packing IPs with at most k nonzeroes per column

Results

	Previous	This work	Integrality Gap
k-HDM	8k	2k	$\geq 2(k-1+1/k)$
Dem. Match.	3.5	3	≥ 3
2-CS-PIP	4	3	≥ 3

[1] Chekuri, Ene, & Korula 2009
[2] Bansal, Korula, Nagarajan, & Srinivasan 2010
[3] Shepherd & Vetta 2002
[4] Pritchard & Chakrabarty 2009

Iterative Packing

- Iterative rounding handles soft constraints
- Emulate features of iterative rounding:
  - Conceptually simple
  - Large frac values facilitate approx
    - Suffices to find single large edge/iter
  - Illustrate with matching

Matching IP

$$
\begin{aligned}
Max \quad & \sum_{e \in E} p_e x_e \\
s.t. \quad & \sum_{e \in \delta(v)} x_e \leq 1 & \forall u \in V \\
& x_e \in \{0,1\} & \forall e \in E
\end{aligned}
$$

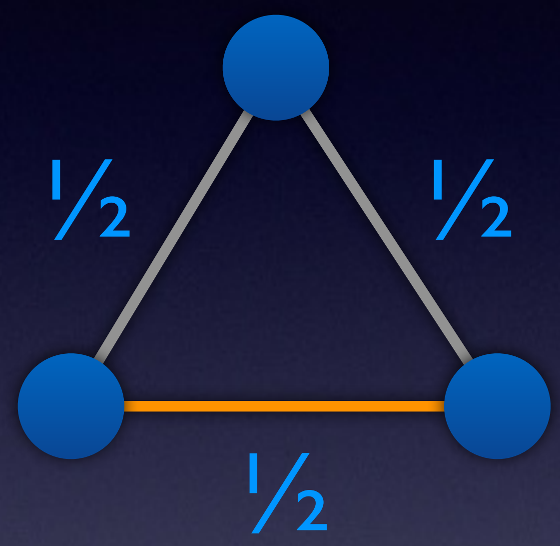
- Edge e is in matching (x_e=1) or not (x_e=0)
- At most one edge picked touching vertex u

Matching LP Relaxation

$$\begin{aligned} & Max \quad \sum_{e \in E} p_e x_e \\ & s.t. \quad \sum_{e \in \delta(v)} x_e \leq 1 \qquad \forall u \in V \\ & \qquad\qquad x_e \in [0,1] \quad \forall e \in E \end{aligned}$$

- Edge e takes real value between 0 and 1
- Values at each vertex must sum ≤ 1

Integrality gap



IP: 1
LP: 3/2

- Assume all edge profits are 1
- This matching LP has gap of 3/2

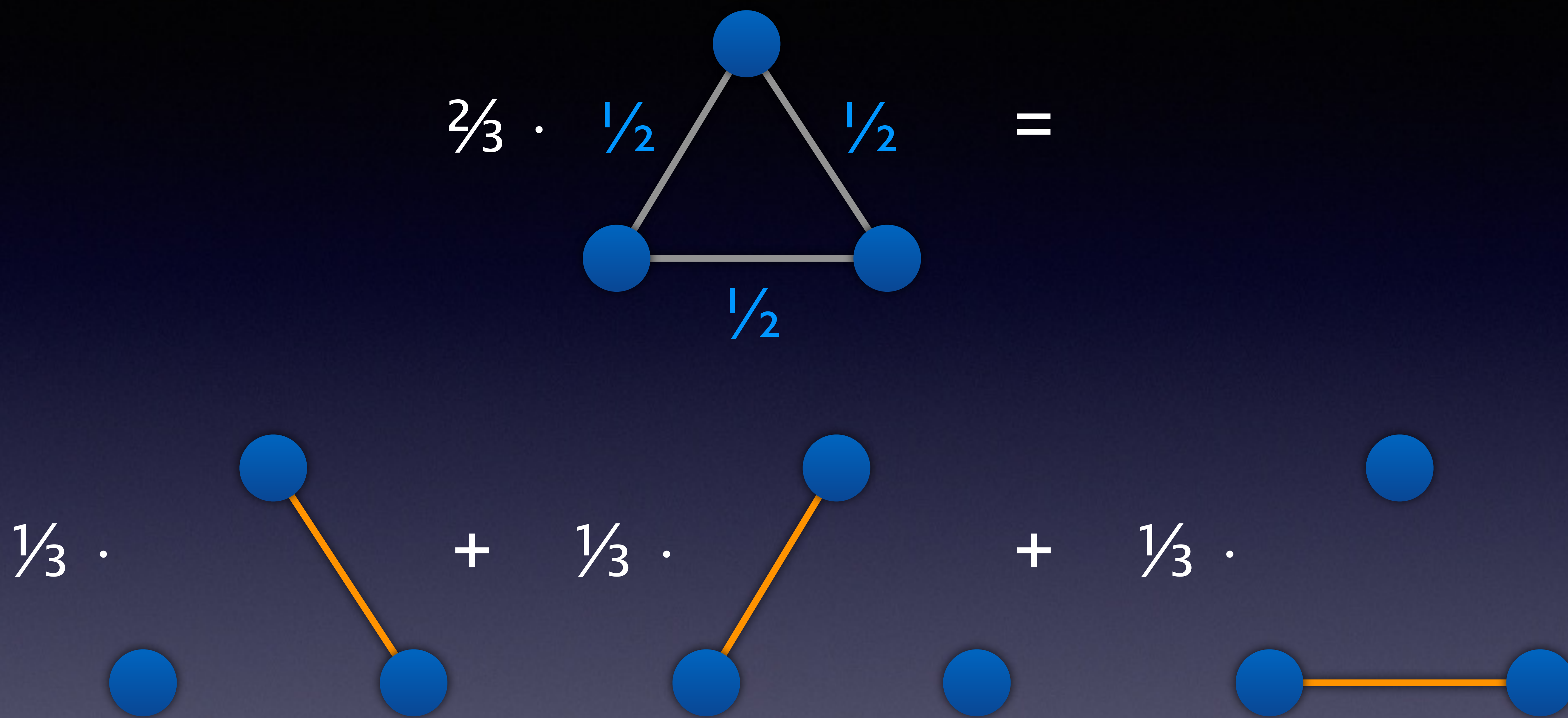
Convex Decompositions, Approximately

- What to do with fractional solutions?
- Find an approx convex decomposition:

$$\frac{1}{\alpha} \cdot x \leq \sum_i \lambda_i \chi^i, \sum_i \lambda_i = 1$$

- x is a frac sol and $\alpha \geq 1$ is a paramater chosen as small as possible

Convex Decomposition: Example



- Picking best matching gives $\geq$ ⅔ profit of frac sol
- Is α=3/2 optimal?

Approximate Convex Decompositions

- Optimal α is precisely integrality gap! [Carr & Vempala 2002]
- Need poly # of solutions for an approx alg
- Iterative packing greedily maintains approx convex decomp

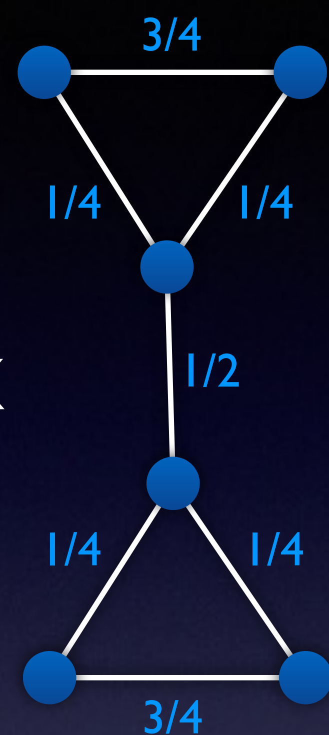
Iterative Packing

- Begin with fractional LP solution, x
- Goal is approx convex decomposition:

1. Remove edge e from G and x
2. Recursively obtain approx conv decomp of $1/\alpha \cdot x$ into integral solutions in G
3. Insert e into $1/\alpha \cdot x_e$ frac of integral solutions

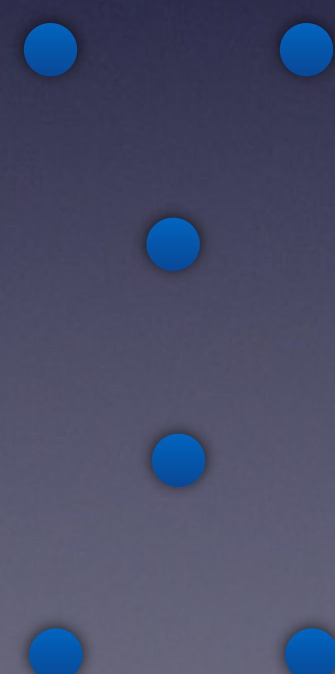
Iterative Packing: Example

x



½ · x

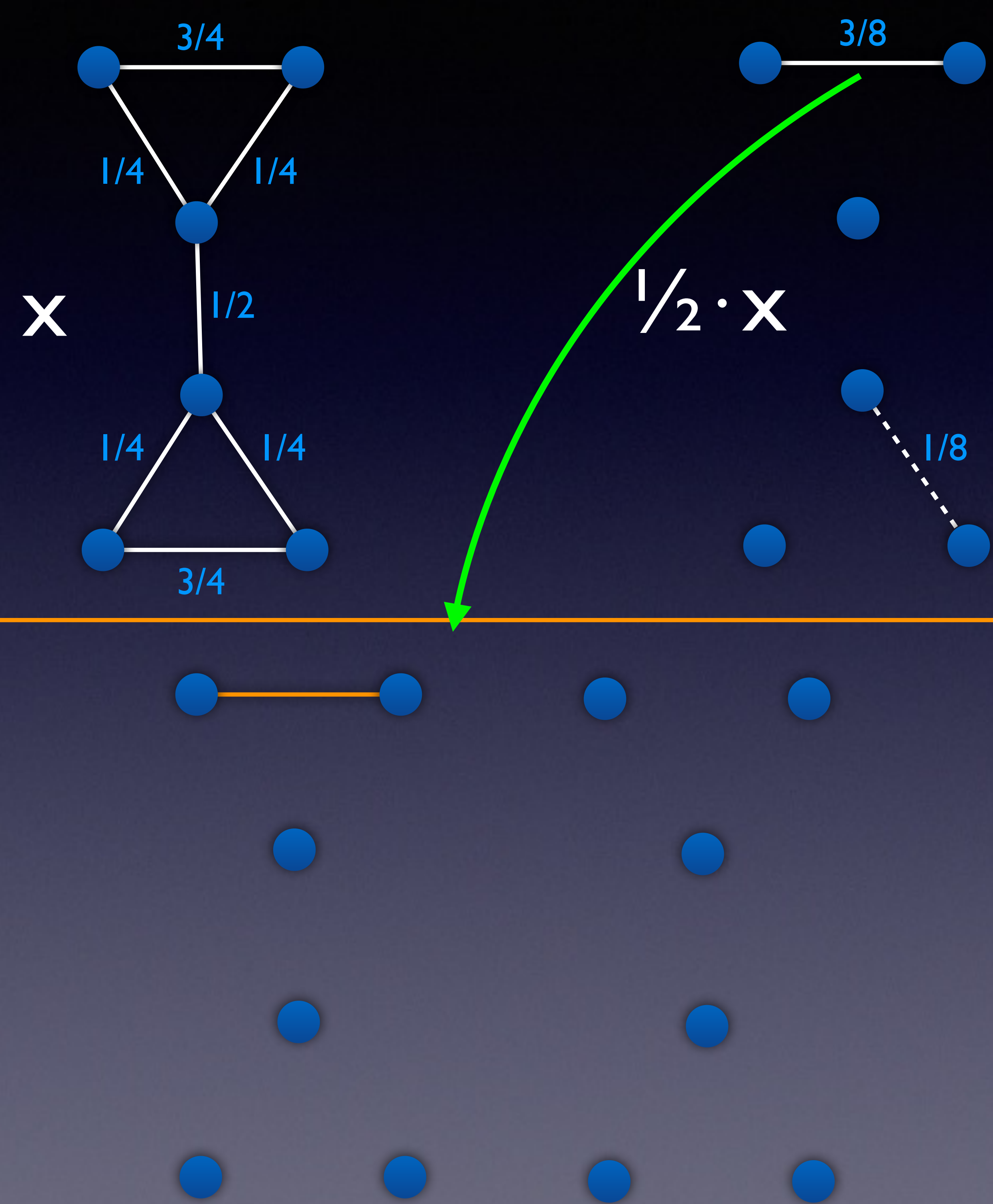
Convex Decomposition



Multiplier

1

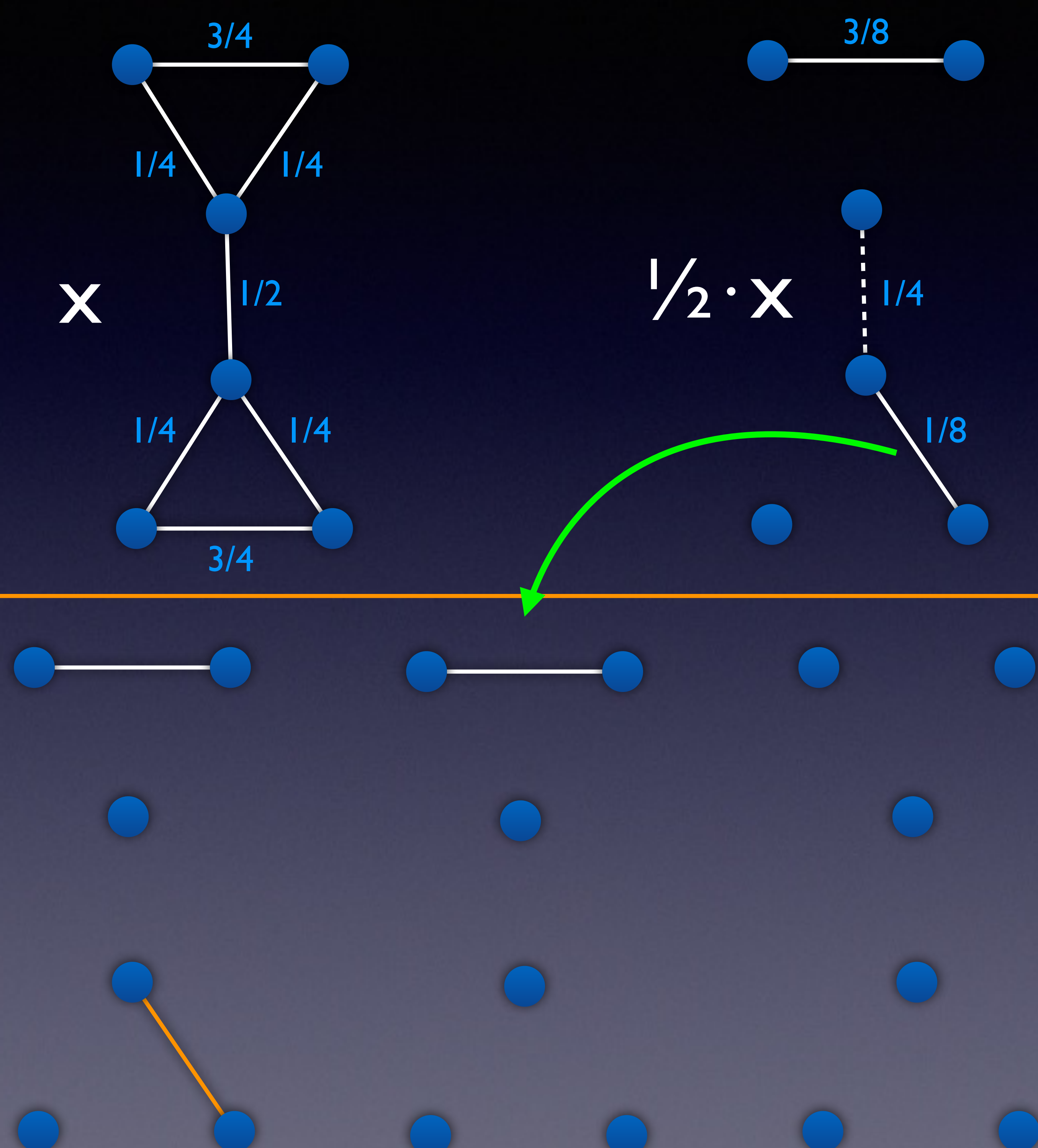
Iterative Packing: Example



C.D.

Mult. $\frac{3}{8}$ $1 - \frac{3}{8} = \frac{5}{8}$

Iterative Packing: Example

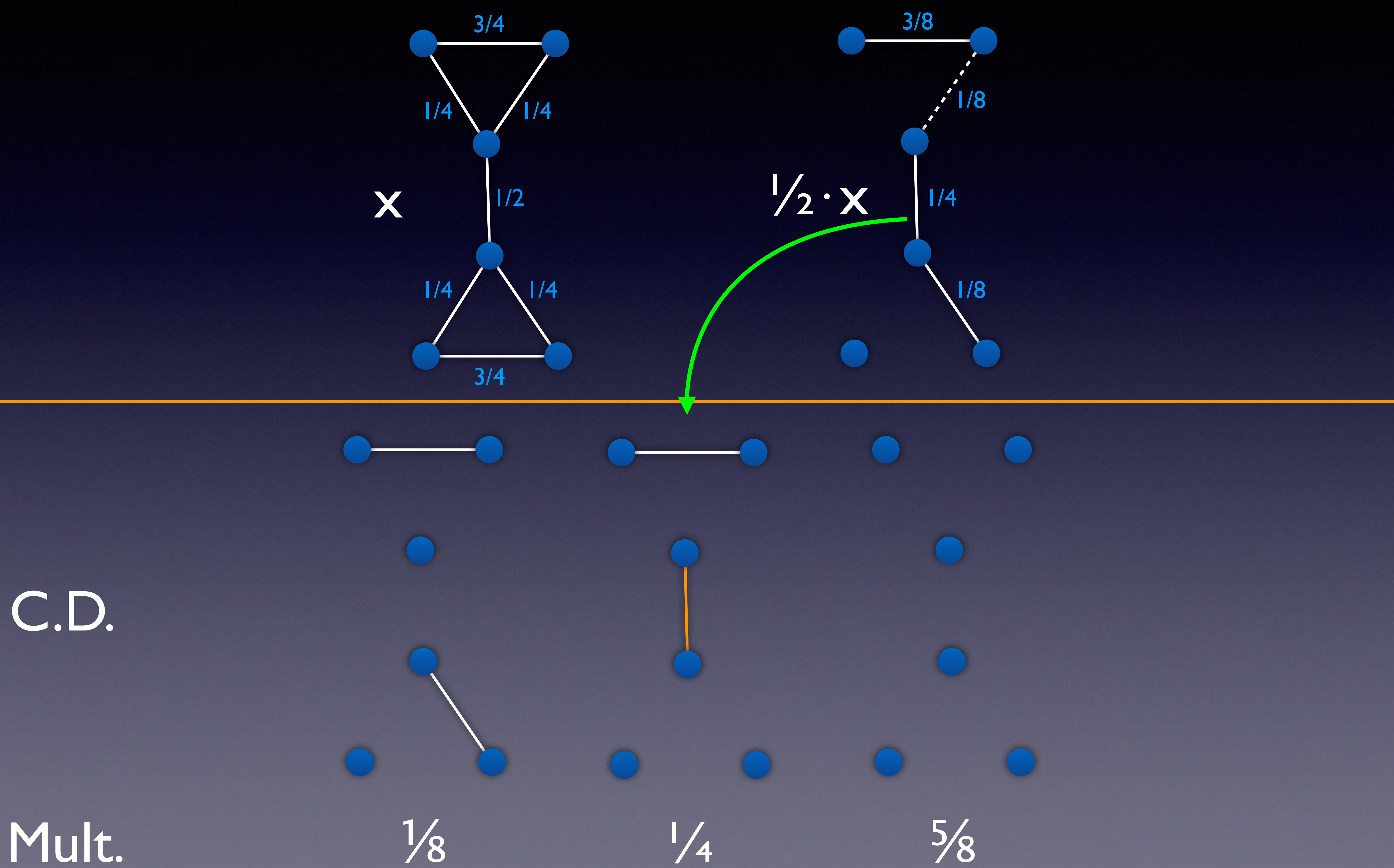


C.D.

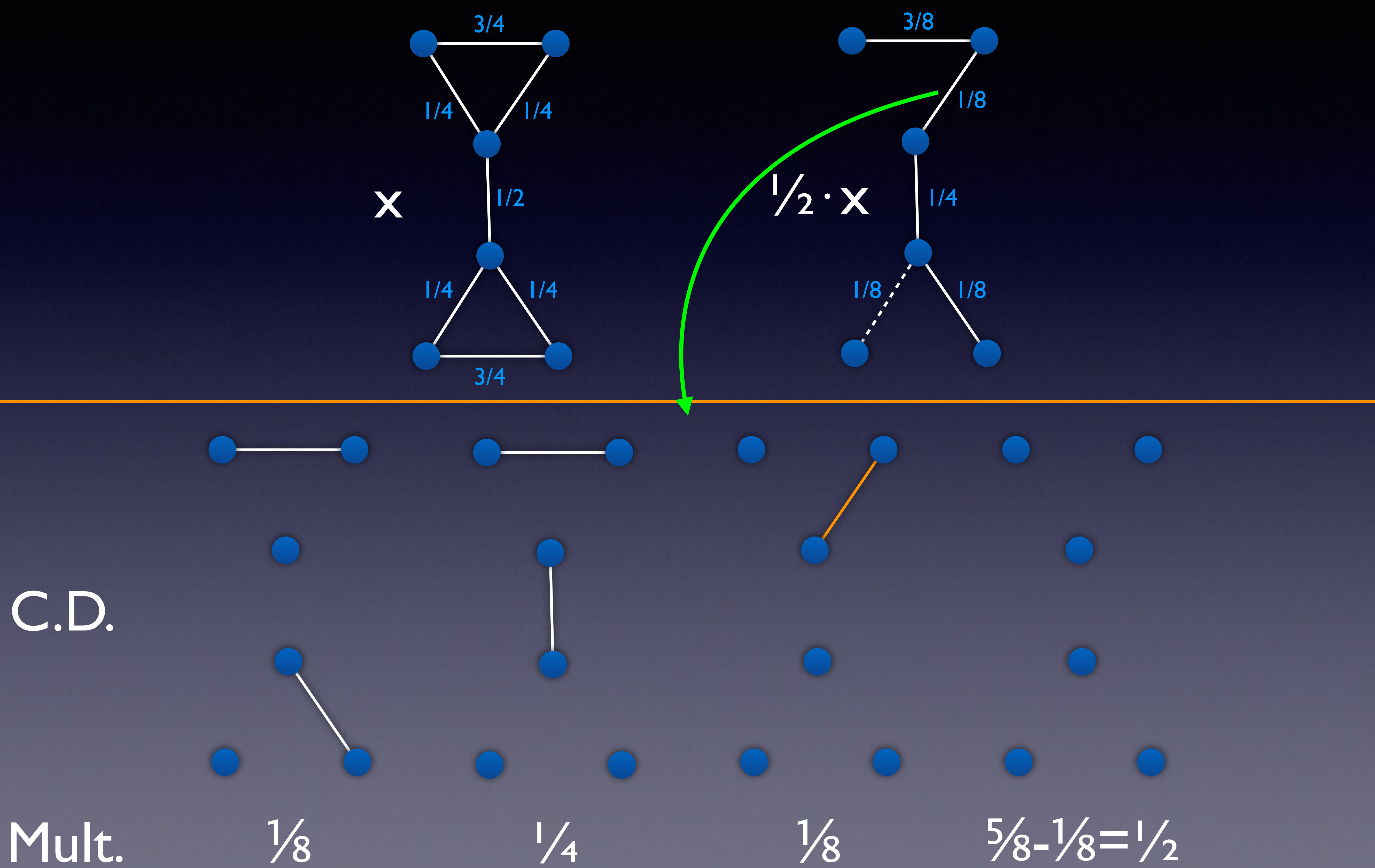
Mult.

⅛   ⅜-⅛=¼   ⅝

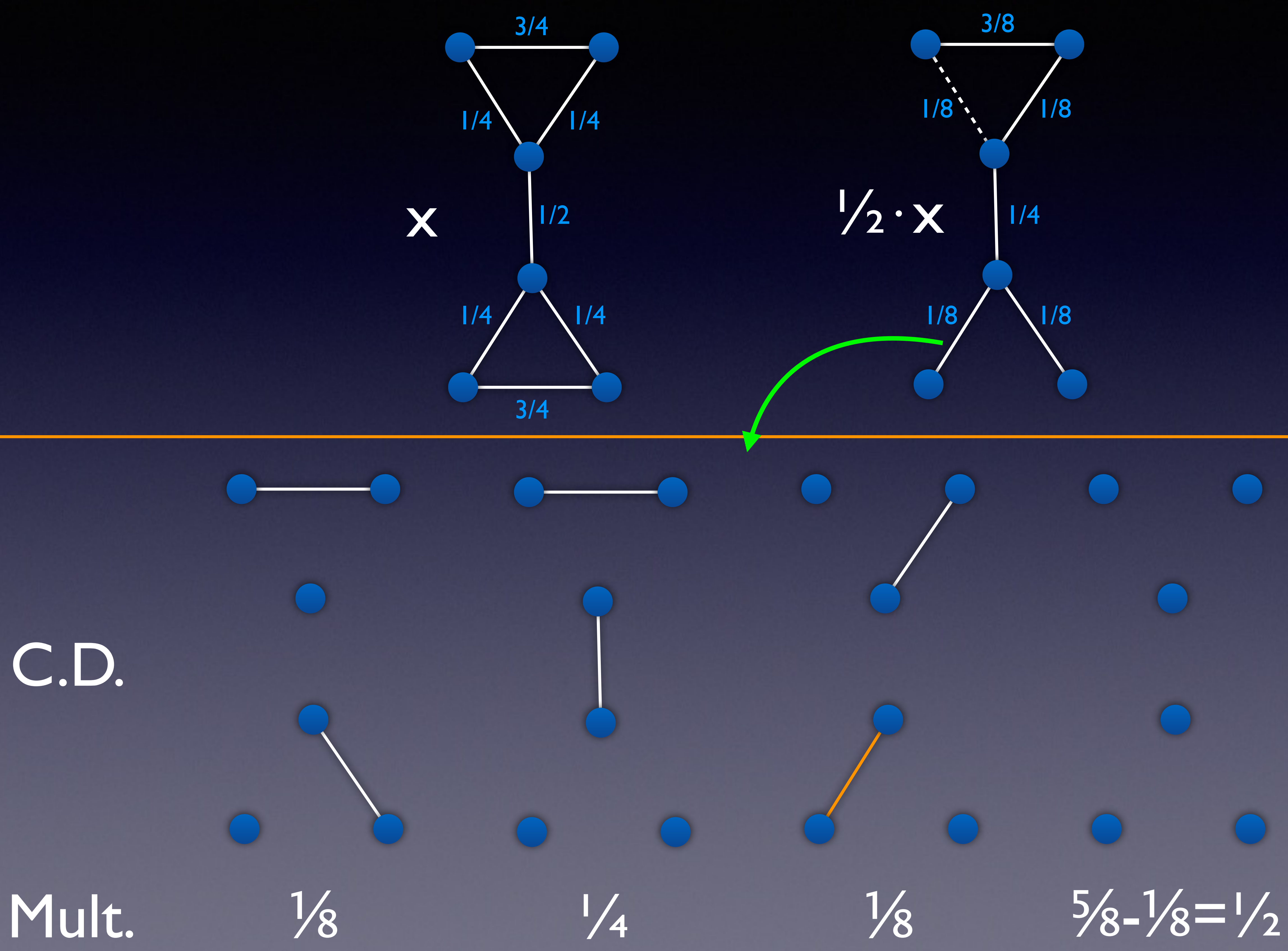
Iterative Packing: Example



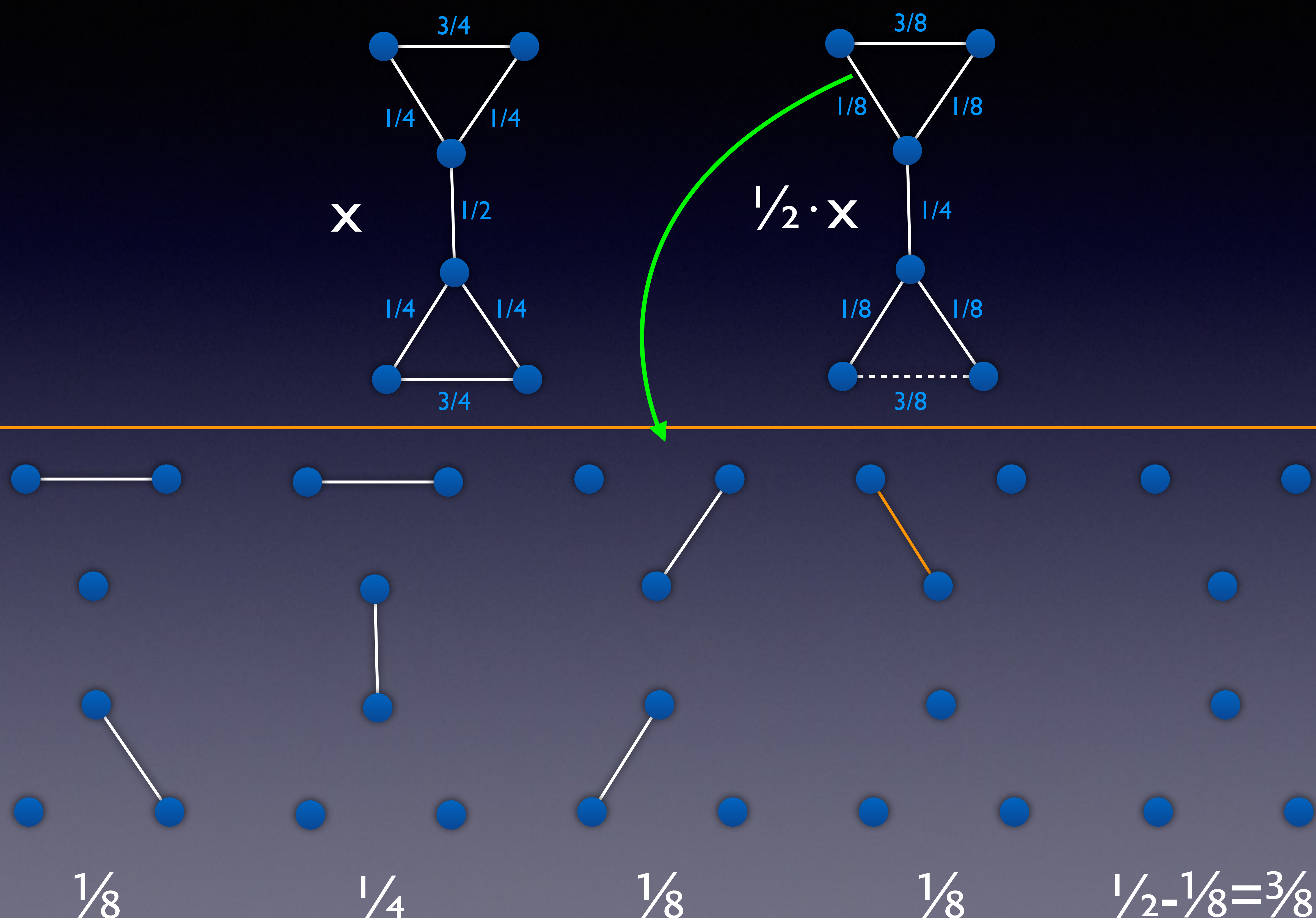
Iterative Packing: Example



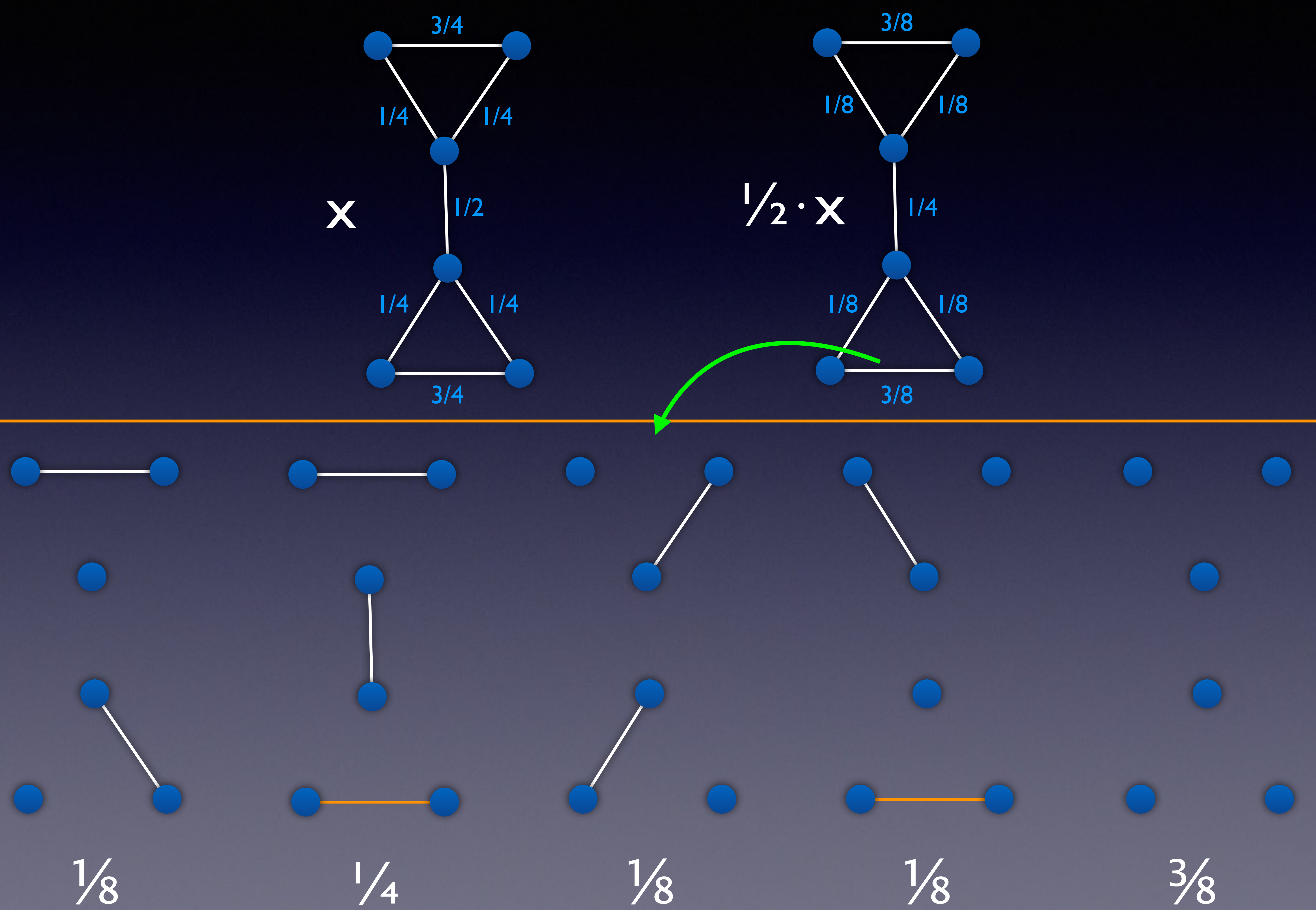
Iterative Packing: Example



Iterative Packing: Example



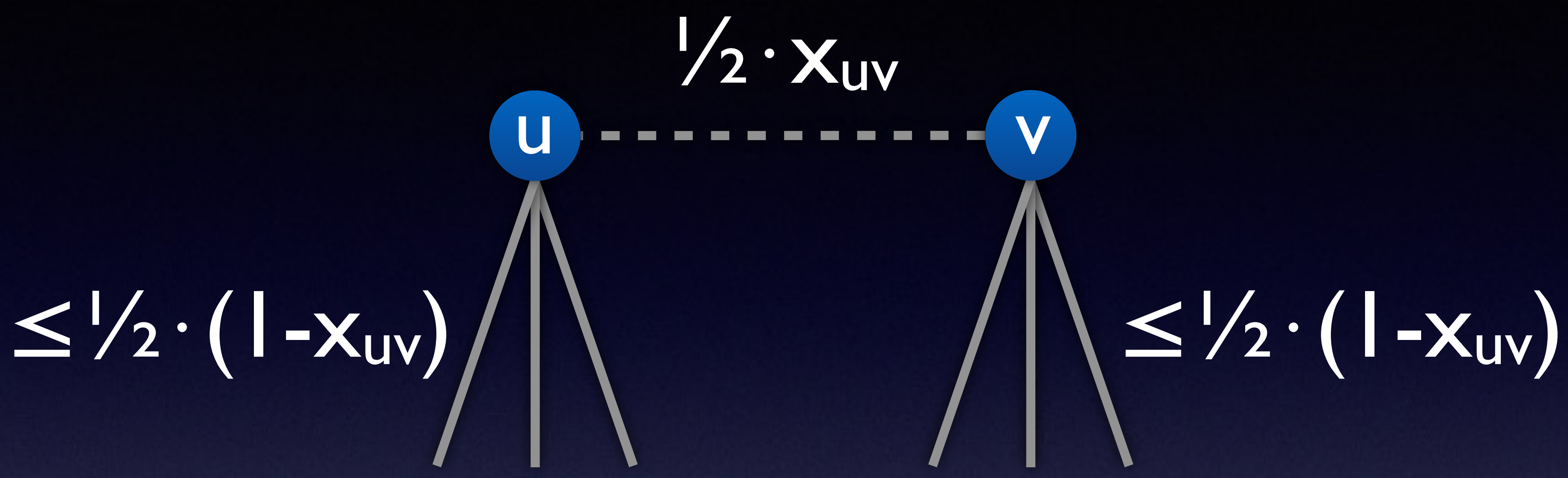
Iterative Packing: Example



Iterative Packing: ½-approx

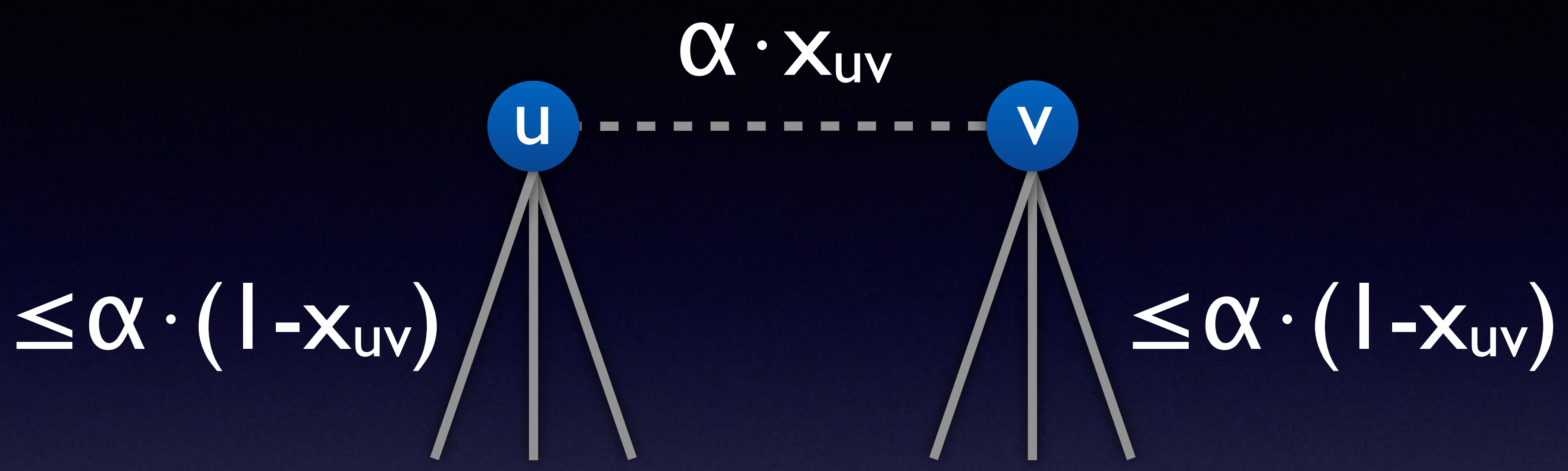
- At most one new solution per iteration
- Is ½-approx conv decomp always possible?
- Actually did better: $\frac{4}{5} \cdot x$
- When can we provably achieve $\geq \frac{1}{2} \cdot x$?

Iterative Packing: ½-approx



- Consider when edge uv inserted
- u, v each matched in $\leq \frac{1}{2} \cdot (1 - x_{uv})$ frac of sols
- Room for uv in $\geq 1 - 2 \cdot \frac{1}{2}(1 - x_{uv}) = x_{uv}$ frac
- Note: only needed $\frac{1}{2} \cdot x_{uv}$ frac

Iterative Packing: α-approx



- Consider when edge uv inserted
- u and v matched in $\leq \alpha \cdot (1-x_{uv})$ frac of sols
- Require: $2[\alpha \cdot (1-x_{uv})] + \alpha \cdot x_{uv} \leq 1$
- May choose $\alpha = 1/(2-x_{uv})$

k-HDM IP

$$
\begin{array}{lll}
Max & \sum_{e \in E} p_e x_e & \\
s.t. & \sum_{e \in \delta(v)} d_e x_e \leq b_u & \forall u \in V \\
 & x_e \in \{0,1\} & \forall e \in E
\end{array}
$$

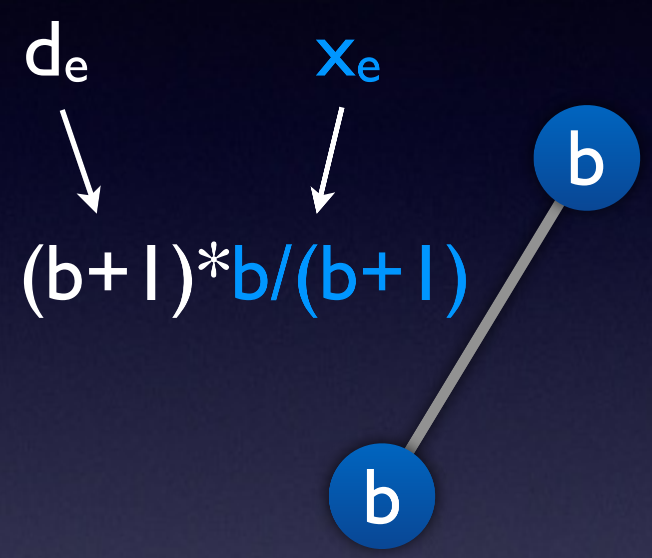
- Assume $|e| \leq k$ for all e
- (Hyper)edge e induces demand d_e at u in e
- Total demand induced at u must be $\leq b_u$

k-HDM LP

$$
\begin{aligned}
&Max && \sum_{e \in E} p_e x_e && \\
&s.t. && \sum_{e \in \delta(v)} d_e x_e \leq b_u && \forall u \in V \\
& && x_e \in [0,1] && \forall e \in E
\end{aligned}
$$

- Demands must frac pack at each vertex

Integrality gap

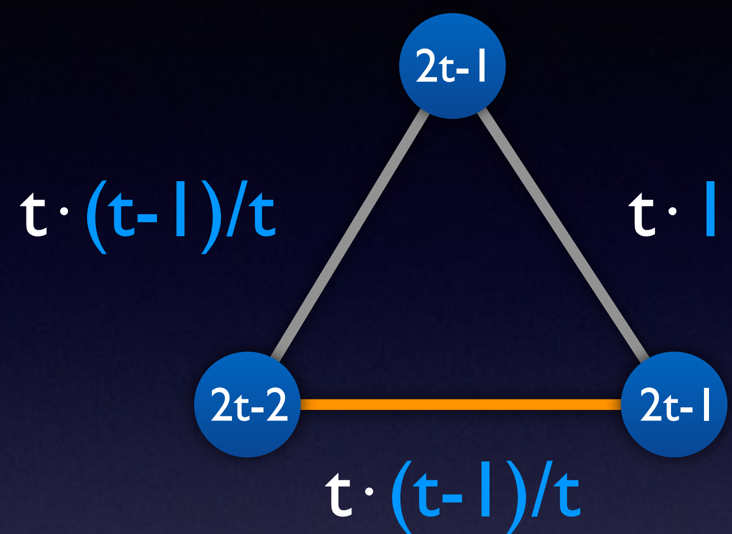


IP: 0

LP: >0

- IP is infeasible, yet LP is not
- Cheating since $d_e > b$
- Fix: no clipping-assumption
- Problem: integral edges

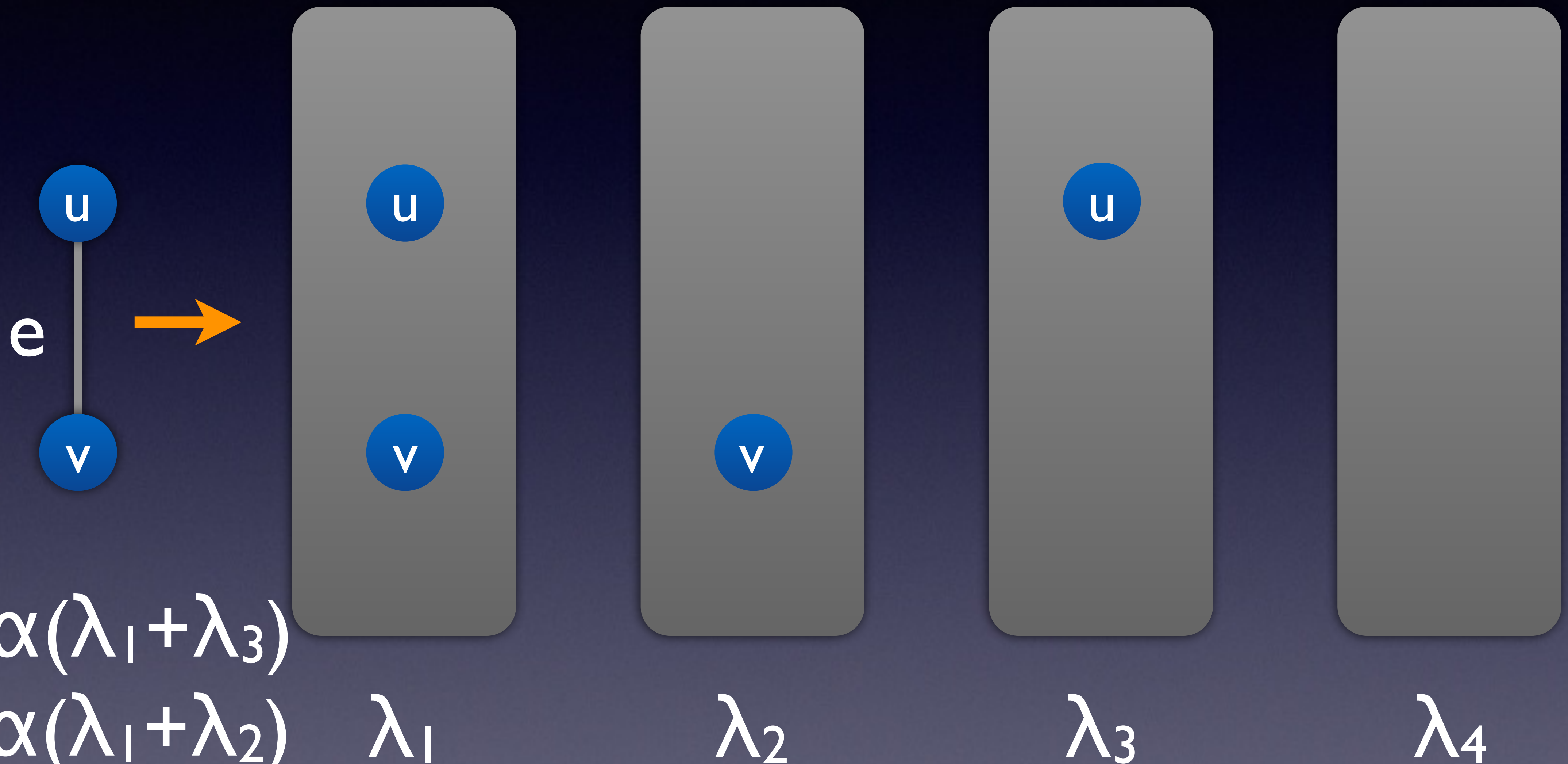
Integrality Gap



IP: 1
LP: ~3

- Essentially (k-hypergraph) matching
- LP "cheats" by exploiting demands
- Projective plane gives 2(k-1+1/k)
- We show these are essentially tight

Analysis



$\beta_u = \alpha(\lambda_1 + \lambda_3)$

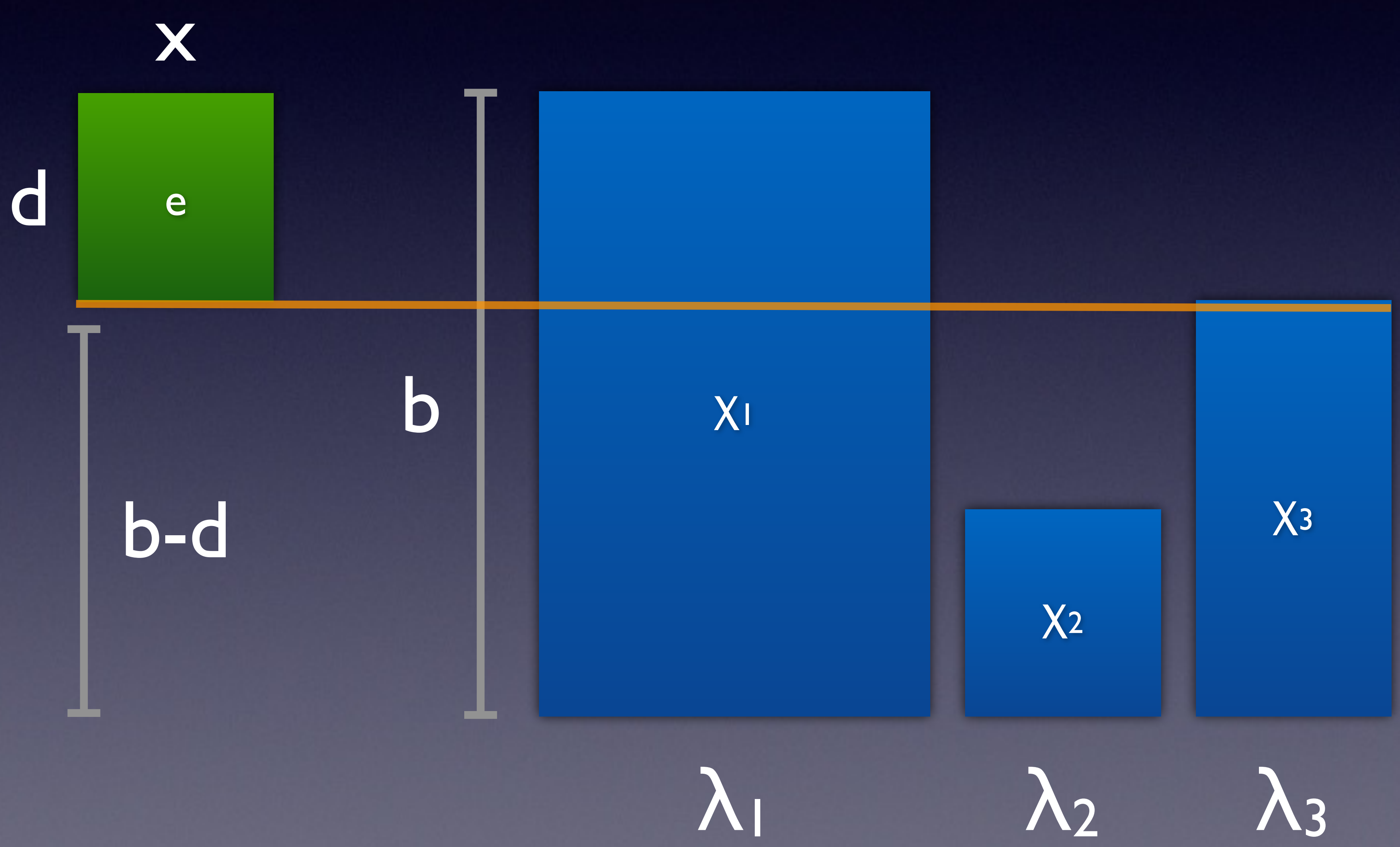
$\beta_v = \alpha(\lambda_1 + \lambda_2)$

- Consider when e is inserted
- Let $\beta_u = \alpha \cdot$ (frac of sols obstructing e at u)

Analysis

- β_u/α is frac of solutions obstructing e at u
- Bound: $\beta \geq \max_u \beta_u$
- Require: $(k \cdot \beta + x_e)/\alpha \leq 1$
- May choose: $\alpha = k \cdot \beta + x_e$
- Task: bound β

Packing with demands



$\beta=\alpha(\lambda_1+\lambda_3)$

Volume

$$\beta \leq \frac{b-d\cdot x}{b-d+1}$$

Height

Analysis: k-HDM

- Recall: α = k· β + x
- $\beta \leq \dfrac{b - d \cdot x}{b - d + 1}$
- β ≤ 1 for special cases (e.g. d=1, x=1)
- In general, bound is unwieldy
- Idea: order edges by demand

$$\beta \leq \frac{b - d \cdot x}{\max\{b - d + 1, d\}} \leq 2\frac{b - d \cdot x}{b + 1} \leq 2$$

Analysis: DM

- Observation [S & V]: 3-approximation on fractional support of an ext point
- Easier “nonconstructive” proof of I.G.
- Idea: order edges by x value (integral first)
  - New base case: fractional support
  - Need approx decomp for base case
  - [C & V] or directly

Iterative Packing

Problem	Solution	Ordering	Base Case	α
Matching	Arbitrary	Arbitrary	∅	2
Matching	Ext Point	Dec x	∅	3/2
k-HDM	Arbitrary	Dec d	∅	2k
2-CS-PIP	Ext Point	Integer	Frac	3*
k-H b-M	Ext Point	Chan & Lau	∅	k-1+1/k*

Open Problems

- 2k-approx for k-CS-PIP
- Better use of extreme points
- Demand versions of other problems
- Configuration LP (exponential # variables)
- Improvement for bipartite graphs

Thanks!