A Tale of Santa Claus, Hypergraphs and Matroids

Sami Davies, Thomas Rothvoss, Yihao Zhang

UNIVERSITY *of* WASHINGTON

The Santa Claus problem (Restricted Max-Min Fair Allocation)

The Santa Claus problem (Restricted Max-Min Fair Allocation)

For W a set of **gifts**, X a set of **children**, where child i has value p_{ij} in $\{0, p_j\}$ for gift j, find assignment $\sigma : W \to X$ maximizing $\min_{i \in X} \sum_{j \in \sigma^{-1}(i)} p_{ij}$

Limit values in *restricted*

The Santa Claus problem (Restricted Max-Min Fair Allocation)

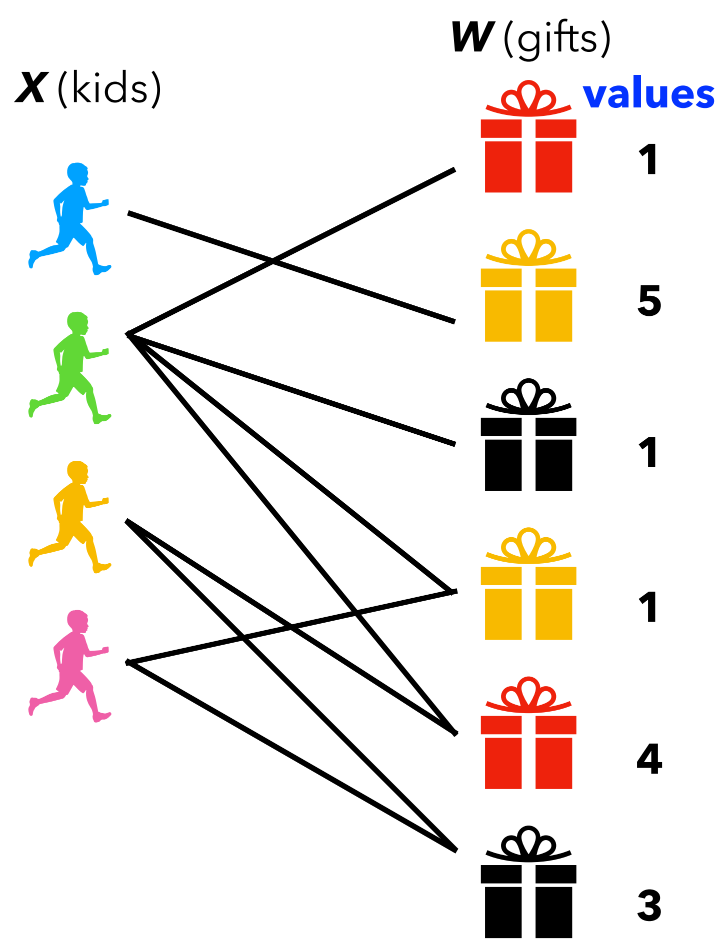
For W a set of **gifts**, X a set of **children**, where child i has value p_{ij} in $\{0, p_j\}$ for gift j, find assignment $\sigma : W \to X$ maximizing $\min_{i \in X} \sum_{j \in \sigma^{-1}(i)} p_{ij}$

Limit values in *restricted*

The Santa Claus problem (Restricted Max-Min Fair Allocation)

For W a set of **gifts**, X a set of **children**, where child i has value p_{ij} in $\{0, p_j\}$ for gift j, find assignment $\sigma : W \to X$ maximizing $\min_{i \in X} \sum_{j \in \sigma^{-1}(i)} p_{ij}$

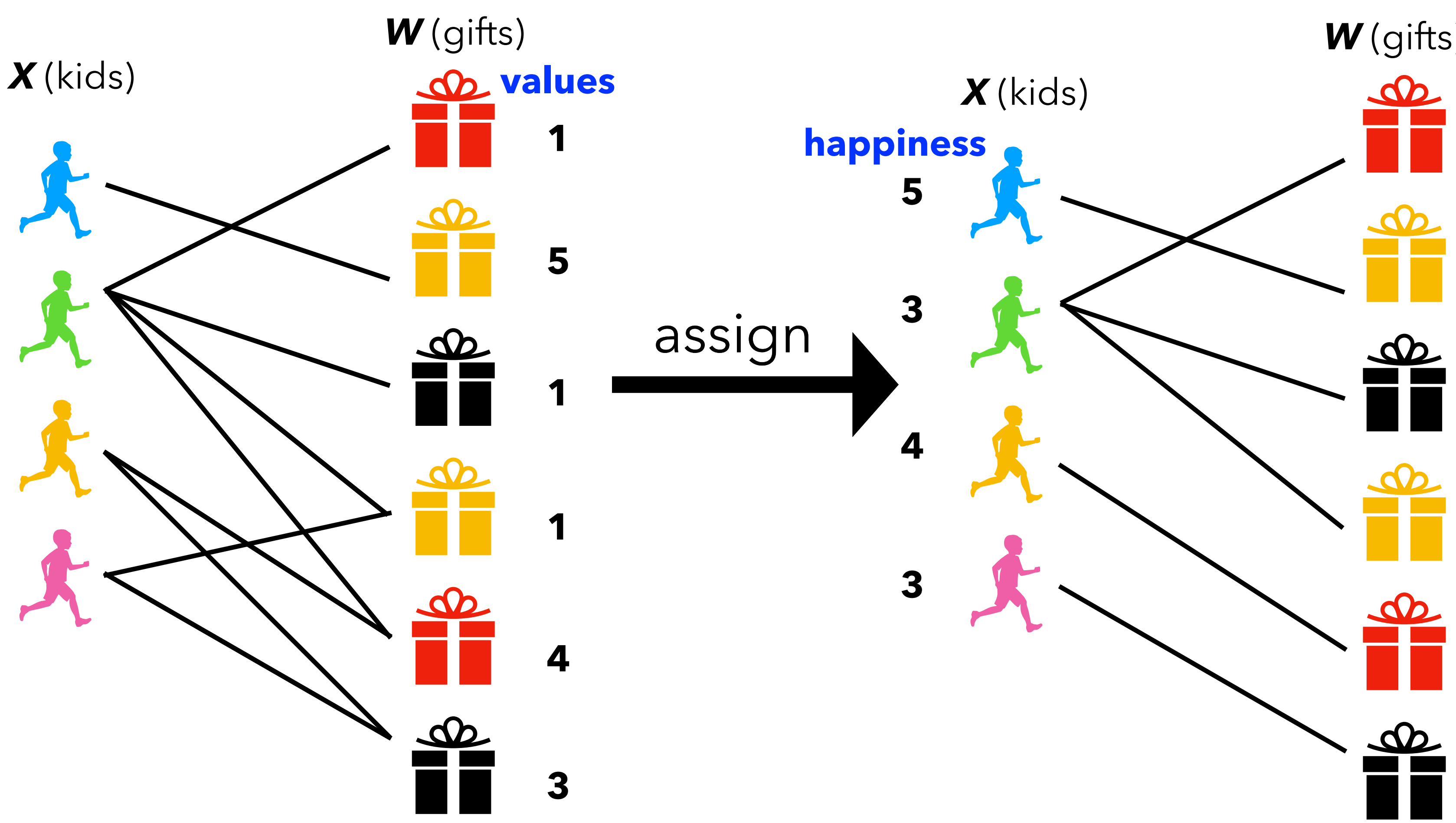
Limit values in *restricted*



The Santa Claus problem (Restricted Max-Min Fair Allocation)

For W a set of **gifts**, X a set of **children**, where child i has value p_{ij} in $\{0, p_j\}$ for gift j, find assignment $\sigma : W \to X$ maximizing $\min_{i \in X} \sum_{j \in \sigma^{-1}(i)} p_{ij}$

Limit values in *restricted*



"Dual" to classic jobs-machines scheduling

Prior Work on Santa Claus

[Bezakova, Dani '05] NP-hard to approximate Santa Claus within factor <2

[Annamalai, Kalaitzis, Svensson '15] 12.3-approx. algorithm use existence of a solution of a configuration LP (CLP)

[Cheng, Mao '19] CLP has integrality gap between 2 and 3.808

$\mathscr{C}(i,T)$ = sets of gifts giving child i value at least T. Exponentially many variables.

$$\sum_{C \in \mathscr{C}(i,T)} z_{i,C} = 1 \qquad \forall i \in X$$

$$\sum_{C : j \in C} \sum_{i} z_{i,C} \leq 1 \qquad \forall j \in W$$

$$z \geq 0.$$

^ sol'n can be approx using the ellipsoid method

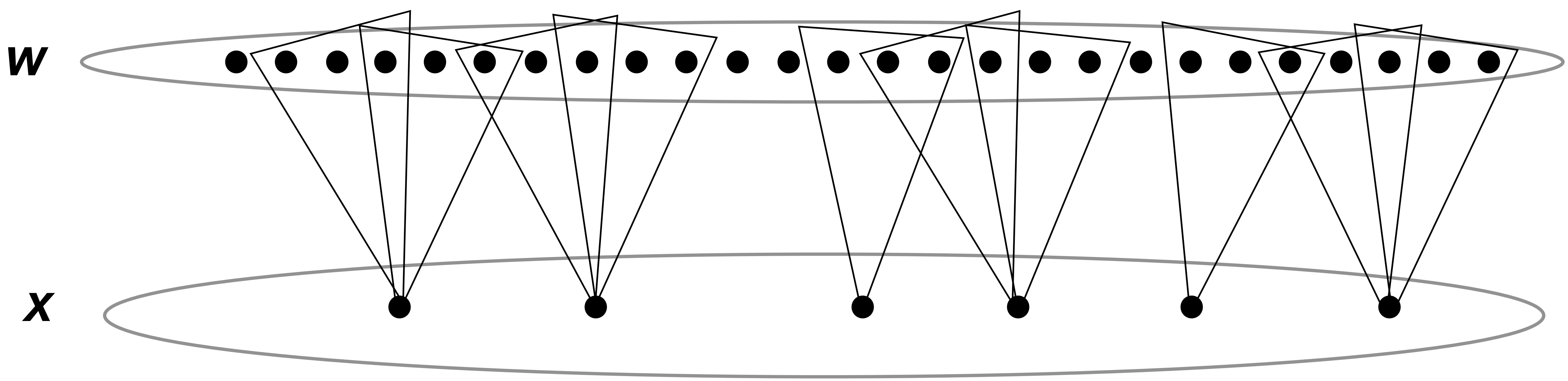
Hypergraph Matchings

Reframe allocation problems as bipartite hypergraph matching problems

Hypergraph Matchings

A **hypergraph** $\mathscr{H} = (X \cup W, \mathscr{E})$ is **bipartite** if for all e in $\mathscr{E}$, $|e \cap X| = 1$.

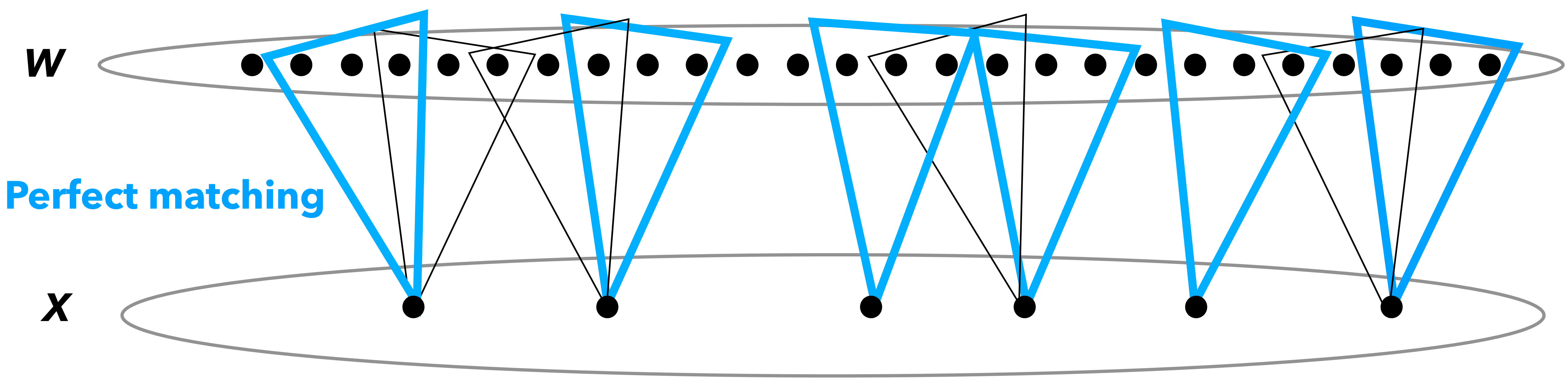
Hyperedges $F \subseteq \mathscr{E}$ form a ***X*-perfect matching** if disjoint and every node in X is contained in exactly one edge in F.



Hypergraph Matchings

A **hypergraph** $\mathscr{H}$ =($X \cup W$, $\mathscr{E}$) is **bipartite** if for all e in $\mathscr{E}$, $|e \cap X|=1$.

Hyperedges $F \subseteq \mathscr{E}$ form a ***X*-perfect matching** if disjoint and every node in X is contained in exactly one edge in F.



Hypergraph Matchings

CLP solution = fractional X-perfect matching on $(X \cup W, \cup_{i\in X} \mathscr{C}(i,T))$.

$\mathscr{C}(i,T)$ = sets of gifts giving child i value at least T

$$\sum_{C\in\mathscr{C}(i,T)} z_{i,C} = 1 \qquad \forall i \in X$$
$$\sum_{C:j\in C} \sum_{i} z_{i,C} \le 1 \qquad \forall j \in W$$
$$z \ge 0.$$

Rewrite CLP in hypergraph matching notation

$$(LP) \qquad \sum_{e\in\mathscr{C}(i,T)} z_e = 1 \qquad \forall i \in X$$
$$\sum_{\substack{e \in \cup_{i\in X} \mathscr{C}(i,T) \\ : w\in e}} z_e \le 1 \qquad \forall w \in W$$
$$z \ge 0.$$

Say restricting instance to have extra structure.

Hypergraph Matchin

Finding perfect matchings in bipartite hypergraphs is NP-hard.

When do there exist perfect matchings?

When, and how, can we find them efficiently?

[Haxell '95] Let $\mathscr{H} = (X \cup W, \mathscr{E})$ be a bipartite hypergraph with $|e| \leq r$ for all e in $\mathscr{E}$. Then either $\mathscr{H}$ contains a X-perfect matching or there are subsets $X' \subset X$ and $W' \subset W$ so that all hyperedges incident to X' intersect W' and $|W'| \leq (2r - 3)(|X'| - 1)$.

Generalization of augmenting paths in bipartite graphs

[Annamalai '15, Annamalai, Kalaitzis, Svensson '15] Use **augmenting tree** to make Haxell's argument polynomial (with some slack) and obtain 12.3 approx. for Santa Claus.

[Davies, Rothvoss, Zhang '18] When X forms a **matroid**, use augmenting tree to find hypergraph matching on some basis of the matroid.

Our Main Result for Santa Claus

Our Main Result for Santa Claus

The Santa Claus problem admits a *(4+ε)*-approximation algorithm in time $n^{\Theta_\varepsilon(1)}$.

When gift values are “well-separated”, can approach a 3-approx

Contribution
Our Main ~~Result for Santa Claus~~

We exploit an underlying matroid to design a simple, new framework for scheduling problems.

- Introduce a more general problem, Matroid Max-Min Allocation
- Use an LP with $O(n^2)$ variables and constraints (simpler than CLP)
- Best approximation for Santa Claus (concurrent with Cheng, Mao '19)

Matroids

Matroids

X = ground set, $\mathcal{I} \subset 2^X$

Matroid $\mathcal{M} = (X, \mathcal{I})$ generalizes linear independence in vector spaces

Independent sets $\mathcal{I}$ satisfy:

- Nonemptyness: $\emptyset$ in $\mathcal{I}$
- Monotonicity: for all $A' \subseteq A$ with A in $\mathcal{I}$, A' in $\mathcal{I}$
- Exchange property: for A, B in $\mathcal{I}$ with $|A| < |B|$, there exists x in $B \setminus A$ such that $A \bigcup x$ in $\mathcal{I}$

Bases of a matroid: $\mathcal{B}(\mathcal{M})$, set of maximal independent sets

Base polytope: $P_{\mathcal{B}(\mathcal{M})} = conv\{\chi(S) \in \{0,1\}^X : S \text{ is a basis of } \mathcal{M}\}$

Matroid Max-Min Allocation

Matroid Max-Min Allocation

Reduce to a general problem we call **Matroid Max-Min Allocation**

Matroid Max-Min Allocation

Reduce to a general problem we call **Matroid Max-Min Allocation**

Setting matroid $\mathcal{M} = (X, \mathcal{I})$, bipartite graph $G = (X \cup W, E)$, resources W to distribute to X, values $p_j \geq 0$ for resource j in W

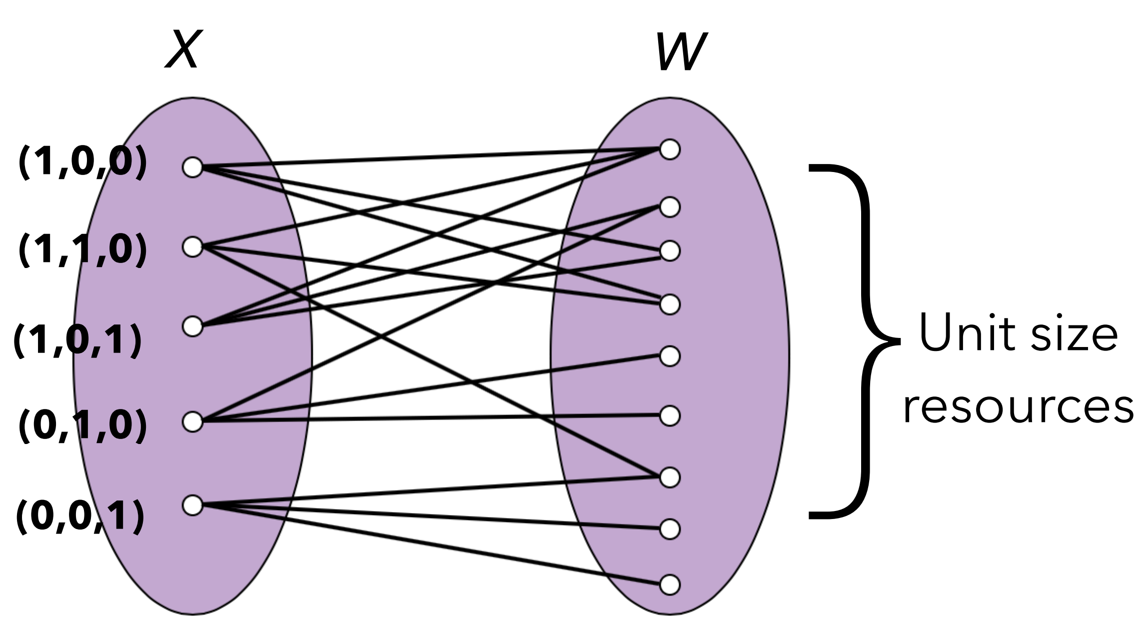
Goal find basis $S \in \mathcal{B}(\mathcal{M})$ and assignment $\sigma : W \to S$ with $(\sigma(i), j)$ in E maximizing over all S $\min_{i \in S} \sum_{j \in \sigma^{-1}(i)} p_j$

Matroid Max-Min Allocation

Reduce to a general problem we call **Matroid Max-Min Allocation**

Setting matroid $\mathcal{M} = (X, \mathcal{I})$, bipartite graph $G = (X \cup W, E)$, resources W to distribute to X, values $p_j \geq 0$ for resource j in W

Goal find basis $S \in \mathcal{B}(\mathcal{M})$ and assignment $\sigma : W \to S$ with $(\sigma(i), j)$ in E maximizing over all S $\min_{i \in S} \sum_{j \in \sigma^{-1}(i)} p_j$

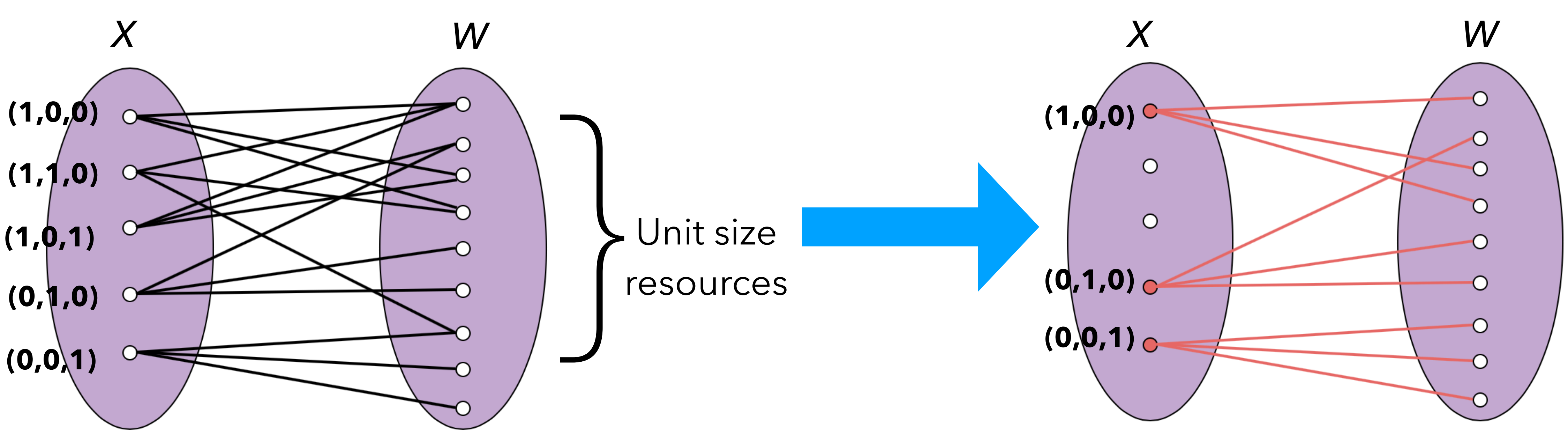


Matroid Max-Min Allocation

Reduce to a general problem we call **Matroid Max-Min Allocation**

Setting matroid $\mathcal{M} = (X, \mathcal{I})$, bipartite graph $G = (X \cup W, E)$
resources W to distribute to X, values $p_j \geq 0$ for resource j in W

Goal find basis $S \in \mathcal{B}(\mathcal{M})$ and assignment $\sigma : W \to S$ with $(\sigma(i), j)$ in E maximizing over all S $\min_{i \in S} \sum_{j \in \sigma^{-1}(i)} p_j$



Matroid Max-Min Allocation

Matroid Max-Min Allocation

For target objective value $T \geq 0$, the LP $Q(T)$ is the set of vectors satisfying: $(x, y) \in \mathbb{R}^X_{\geq 0} \times \mathbb{R}^E_{\geq 0}$

$$x \in P_{\mathcal{B}(\mathcal{M})} \qquad \sum_{j:(i,j)\in E} p_j y_{ij} \geq T \cdot x_i \forall i \in X \qquad \sum_{i:(i,j)\in E} y_{ij} \leq 1 \forall j \in W \quad y_{ij} \leq x_i \forall (i,j) \in E.$$

Matroid Max-Min Allocation

For target objective value $T \geq 0$, the LP $Q(T)$ is the set of vectors satisfying: $(x, y) \in \mathbb{R}^X_{\geq 0} \times \mathbb{R}^E_{\geq 0}$

$$x \in P_{\mathcal{B}(\mathcal{M})} \quad \sum_{j:(i,j)\in E} p_j y_{ij} \geq T \cdot x_i \forall i \in X \quad \sum_{i:(i,j)\in E} y_{ij} \leq 1 \forall j \in W \quad y_{ij} \leq x_i \forall (i,j) \in E.$$

Elements in basis are "well-covered"

Resources not over assigned

Expansion condition

Matroid Max-Min Allocation

For target objective value $T \geq 0$, the LP $Q(T)$ is the set of vectors satisfying: $(x, y) \in \mathbb{R}^X_{\geq 0} \times \mathbb{R}^E_{\geq 0}$

$$x \in P_{\mathcal{B}(\mathcal{M})} \qquad \sum_{j:(i,j)\in E} p_j y_{ij} \geq T \cdot x_i \forall i \in X \qquad \sum_{i:(i,j)\in E} y_{ij} \leq 1 \forall j \in W \qquad y_{ij} \leq x_i \forall (i,j) \in E.$$

Elements in basis are “well-covered”

Resources not over assigned

Expansion condition

Main technical result: Suppose $Q(T) \neq \emptyset$. Then one can find (x,y) in $Q((⅓-\varepsilon)T - ⅓ \max p_j)$ with x and y integral in time $n^{\Theta_\varepsilon(1)}$.

$n = |X|+|W|$

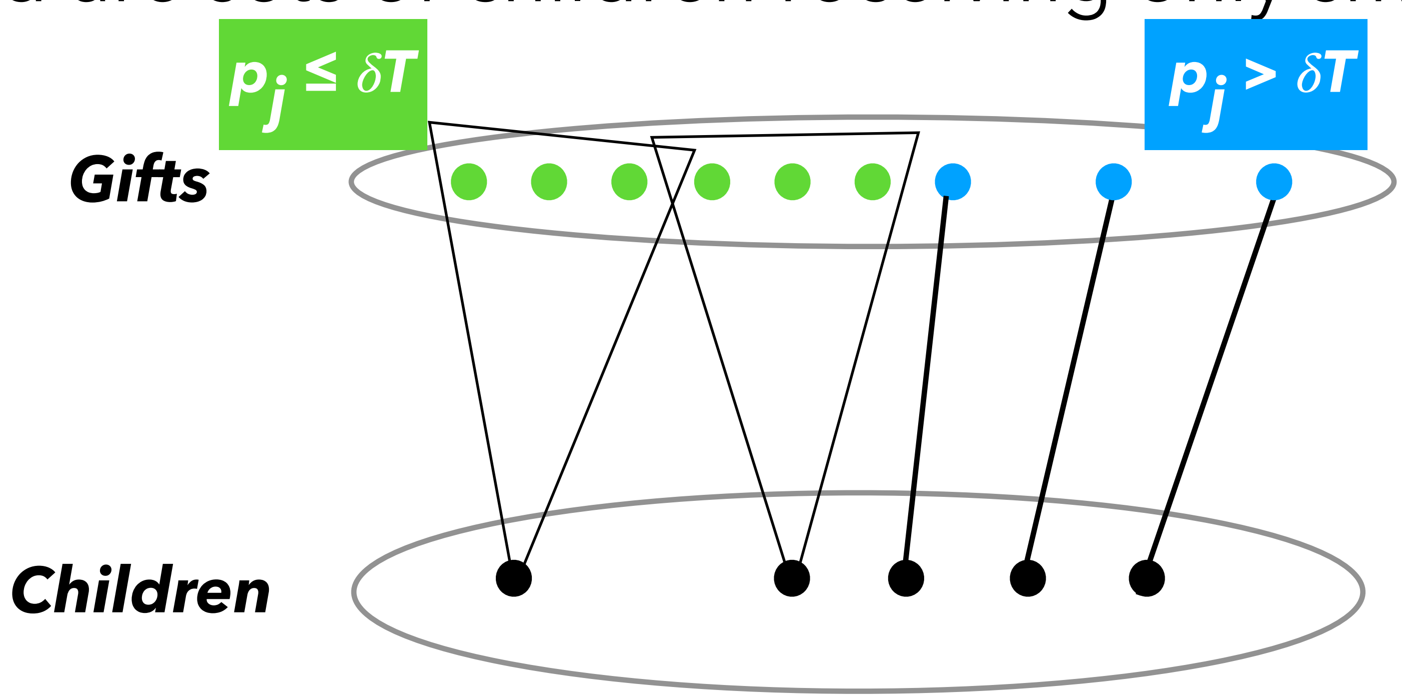
Santa Claus and Matroid Max-Min Allocation

Fix $\delta > 0$. Label gift j large if $p_j > \delta T$, small if $p_j \leq \delta T$

Let $\mathcal{I}$ = {$A \subseteq$ children s.t ∃ matching between A and large gifts}. (children, $\mathcal{I}$) forms a **matchable set matroid**, $\mathcal{M}$.

$\mathcal{M}^*$ = (children, $\mathcal{I}^*$) is the **co-matroid** for $\mathcal{I}^*$ = {$A \subseteq$ children s.t ∃ B in $\mathcal{B}(\mathcal{M})$ with $A \cap B = \varnothing$ }

Bases of co-matroid are sets of children receiving only small gifts



Santa Claus and Matroid Max-Min Allocation

Santa Claus and Matroid Max-Min Allocation

T = opt value. Relaxation $P(T, \delta)$- vectors $z \in \mathbb{R}^{M \times J}$ satisfying:

J_S, J_L: small, large gifts

z_{ij} : does kid i get gift j

A_j: kids who want gift j

Santa Claus and Matroid Max-Min Allocation

T = opt value. Relaxation *P(T, δ)*- vectors $z \in \mathbb{R}^{M \times J}$ satisfying:

$$\sum_{j \in J_S : i \in A_j} p_j z_{ij} \geq T \cdot \left(1 - \sum_{j \in J_L : i \in A_j} z_{ij}\right) \qquad \forall i \in X$$

Kids with no large gifts "well-covered"

No gift over assigned

$$\sum_{i \in A_j} z_{ij} \leq 1 \qquad \forall j \in W$$

$$z_{ij} \leq 1 - \sum_{j' \in J_L : i \in A_{j'}} z_{ij'} \qquad \forall j \in J_S \; \forall i \in A_j .$$

Expansion condition- kids with large gift get no small gifts

J_S, J_L: small, large gifts

z_{ij} : does kid *i* get gift *j*

A_j: kids who want gift *j*

Santa Claus and Matroid Max-Min Allocation

T = opt value. Relaxation *P(T, δ)*- vectors $z \in \mathbb{R}^{M \times J}$ satisfying:

$$\sum_{j \in J_S : i \in A_j} p_j z_{ij} \geq T \cdot \left(1 - \sum_{j \in J_L : i \in A_j} z_{ij}\right) \qquad \forall i \in X$$

Kids with no large gifts "well-covered"

No gift over assigned

$$\sum_{i \in A_j} z_{ij} \leq 1 \qquad \forall j \in W$$

$$z_{ij} \leq 1 - \sum_{j' \in J_L : i \in A_{j'}} z_{ij'} \qquad \forall j \in J_S \ \forall i \in A_j .$$

Expansion condition- kids with large gift get no small gifts

J_S, J_L: small, large gifts

z_{ij} : does kid *i* get gift *j*

A_j: kids who want gift *j*

Instance of SC with objective value *T* → There exists *z* in *P(T, δ)* → There exists *(x*,y*)* in *Q(T)* (w.r.t. $\mathcal{M}^*$)

Santa Claus and Matroid Max-Min Allocation

Instance of SC with objective value T → There exists z in $P(T, \delta)$ → There exists (x^*,y^*) in $Q(T)$ (w.r.t. $\mathcal{M}^*$)

Main technical result: Suppose $Q(T) \neq \emptyset$. Then one can find (x,y) in $Q((⅓-\varepsilon)T - ⅓ \max p_W)$ with x and y integral in poly time

From main technical result: Find children receiving only small gifts and their gift assignments: their happiness ≥ $(⅓-\delta/3-\varepsilon)T$

Remaining children receive a large gift: their happiness ≥ δT

Children receive happiness ≥ $\min\left\{\left(\frac{1}{3} - \delta/3 - \varepsilon\right)T, \delta T\right\}$, set $\delta = 1/4$:

The Santa Claus problem admits a $(4+\varepsilon)$-approximation algorithm in time $n^{\Theta_\varepsilon(1)}$.

Main Technical Result

Main Technical Result

Q(T):

$$x \in P_{\mathcal{B}(\mathcal{M})} \quad \sum_{j:(i,j)\in E} p_j y_{ij} \geq T \cdot x_i \forall i \in X \quad \sum_{i:(i,j)\in E} y_{ij} \leq 1 \forall j \in W \quad y_{ij} \leq x_i \forall (i,j) \in E.$$

Main technical result: Suppose *Q(T)* ≠ ∅. Then one can find *(x,y)* in *Q((⅓-ε)T -⅓ max* p_W*)* with *x* and *y* integral in time $n^{\Theta_\varepsilon(1)}$.

Main Technical Result

Q(T):

$$x \in P_{\mathcal{B}(\mathcal{M})} \quad \sum_{j:(i,j)\in E} p_j y_{ij} \geq T \cdot x_i \forall i \in X \quad \sum_{i:(i,j)\in E} y_{ij} \leq 1 \forall j \in W \quad y_{ij} \leq x_i \forall (i,j) \in E.$$

Main technical result: Suppose *Q(T)* ≠ ∅. Then one can find *(x,y)* in *Q((⅓-ε)T -⅓ max* p_w*)* with *x* and *y* integral in time $n^{\Theta_\varepsilon(1)}$.

Language change: *hyperedges* in a bipartite *hypergraph*

$\mathcal{E}_t$: minimal bipartite hyperedges e with *val(e)* ≥ *t*

val(e)= sum of values of resources in *e*

Hypergraph *H=(X* ∪ *W,* $\mathcal{E}$*)* is bipartite if for all *e* in $\mathcal{E}$, |e ∩ *X*|=*1*.

Main Technical Result

Q(T):

$$x \in P_{\mathcal{B}(\mathcal{M})} \qquad \sum_{j:(i,j)\in E} p_j y_{ij} \geq T \cdot x_i \forall i \in X \qquad \sum_{i:(i,j)\in E} y_{ij} \leq 1 \forall j \in W \quad y_{ij} \leq x_i \forall (i,j) \in E.$$

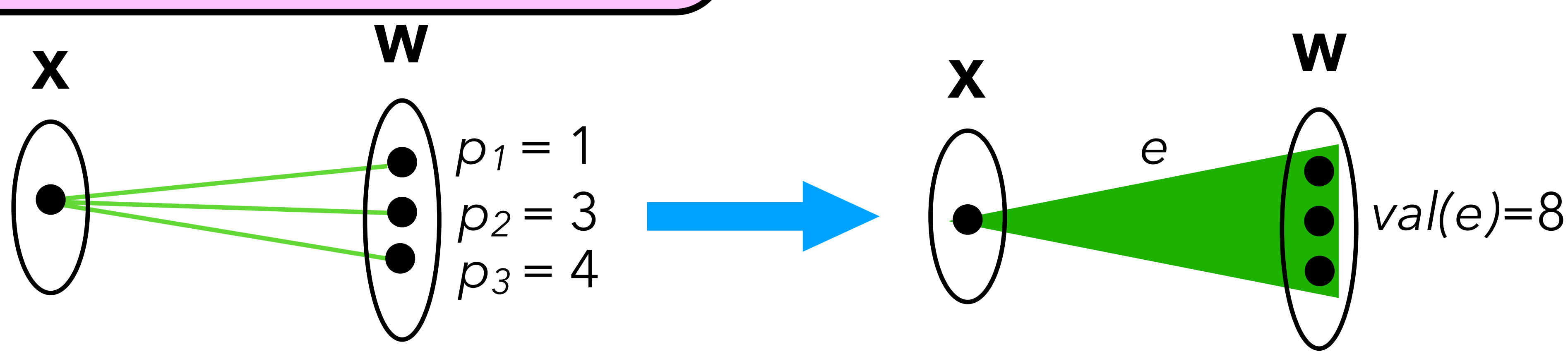
Main technical result: Suppose $Q(T) \neq \emptyset$. Then one can find (x,y) in $Q((⅓-\varepsilon)T - ⅓ \max p_w)$ with x and y integral in time $n^{\Theta_\varepsilon(1)}$.

Language change: *hyperedges* in a bipartite *hypergraph*.

$\mathcal{E}_t$: minimal bipartite hyperedges e with *val(e)* ≥ *t*

Hypergraph $H=(X \cup W, \mathcal{E})$ is bipartite if for all e in $\mathcal{E}$, $|e \cap X|=1$.

val(e)= sum of values of resources in *e*



Main Technical Result

Main Technical Result

Set δ = *max* p_w/T

To prove main technical result: find a basis *S* of $\mathcal{M}$ and a hypergraph matching $M \subseteq \mathcal{E}_{(\frac{1}{3}-\frac{\delta}{3}-\varepsilon)T}$ covering *S*

$\mathcal{E}_t$: bipartite hyperedges *e* with *val(e)*=*t*.

Main Technical Result

Set $\delta = max\ p_w/T$

To prove main technical result: find a basis S of $\mathcal{M}$ and a hypergraph matching $M \subseteq \mathcal{E}_{(\frac{1}{3}-\frac{\delta}{3}-\varepsilon)T}$ covering S

$\mathcal{E}_t$: bipartite hyperedges e with $val(e)=t$.

Our algorithm runs in rank($\mathcal{M}$) phases.

Main Technical Result

Set δ = *max* p_w/T

To prove main technical result: find a basis *S* of $\mathcal{M}$ and a hypergraph matching $M \subseteq \mathcal{E}_{(\frac{1}{3}-\frac{\delta}{3}-\varepsilon)T}$ covering *S*

$\mathcal{E}_t$: bipartite hyperedges *e* with *val(e)*=*t*.

Our algorithm runs in rank($\mathcal{M}$) phases.

Start of phase: $S \in \mathcal{I}$, with $S \setminus i_0$ covered by hypermatching $M \subseteq \mathcal{E}_{(\frac{1}{3}-\frac{\delta}{3}-\varepsilon)T}$

Main Technical Result

Set $\delta = max\ p_w/T$

To prove main technical result: find a basis S of $\mathcal{M}$ and a hypergraph matching $M \subseteq \mathcal{E}_{(\frac{1}{3}-\frac{\delta}{3}-\varepsilon)T}$ covering S

$\mathcal{E}_t$: bipartite hyperedges e with $val(e)=t$.

Our algorithm runs in rank($\mathcal{M}$) phases.

Start of phase: $S \in \mathcal{I}$, with $S \setminus i_0$ covered by hypermatching $M \subseteq \mathcal{E}_{(\frac{1}{3}-\frac{\delta}{3}-\varepsilon)T}$

During a phase: Build an *augmenting tree*. Swap sets of hyperedges in the tree to find more space.

Main Technical Result

Set δ = $max\ p_w/T$

To prove main technical result: find a basis S of $\mathcal{M}$ and a hypergraph matching $M \subseteq \mathcal{E}_{(\frac{1}{3}-\frac{\delta}{3}-\varepsilon)T}$ covering S

$\mathcal{E}_t$: bipartite hyperedges e with $val(e)=t$.

Our algorithm runs in rank($\mathcal{M}$) phases.

Start of phase: $S \in \mathcal{I}$, with $S \setminus i_0$ covered by hypermatching $M \subseteq \mathcal{E}_{(\frac{1}{3}-\frac{\delta}{3}-\varepsilon)T}$

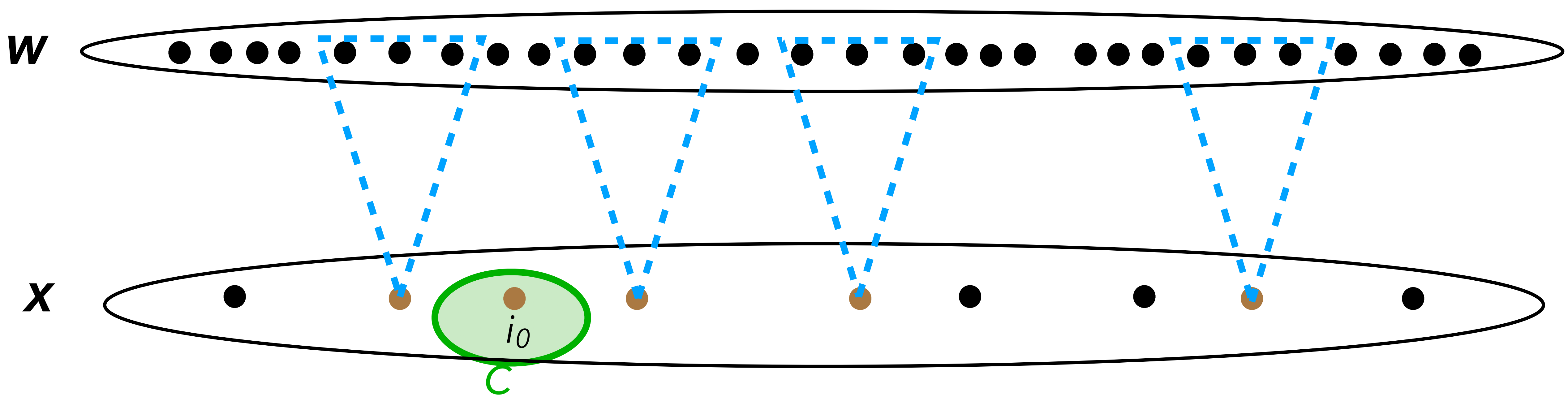
During a phase: Build an *augmenting tree*. Swap sets of hyperedges in the tree to find more space.

End of a phase: Produce new hypermatching covering $S' \in \mathcal{I}$, where $|S'|=|S|$. Larger matching.

Proof: Augmenting tree

Input: $S \in \mathcal{I}$, $i_0 \ni S$, matching $M \subseteq \mathcal{E}_{(\frac{1}{3}-\frac{\delta}{3}-\varepsilon)T}$ covering $S \setminus i_0$, layer index ℓ.

Discovered nodes $C = \{i_0\}$, add edges $A = \emptyset$, blocking edges $B = \emptyset$, matching



Proof: Augmenting tree

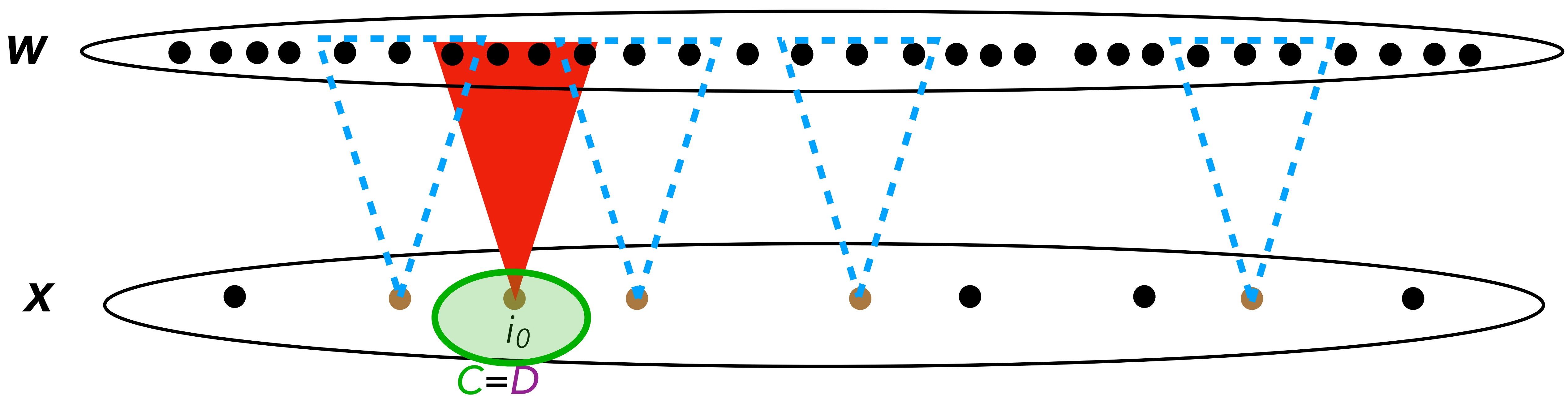
Input: ***S*** $\in \mathcal{I}$, $i_0 \ni S$, matching $M \subseteq \mathcal{E}_{(\frac{1}{3}-\frac{\delta}{3}-\varepsilon)T}$ covering ***S*** $\setminus i_0$, layer index ℓ.

Discovered nodes ***C*** = $\{i_0\}$, add edges ***A*** = ∅, blocking edges ***B***=∅, matching

Repeat until termination

1. Find candidate add edges in $\mathcal{E}_{(\frac{1}{3}-\frac{\delta}{3}-\frac{\varepsilon}{2})T}$ that are:

(a) Disjoint to resources in ***A*** and ***B***, (b) cover ***D*** $\subseteq X$, with $(S \setminus$ ***C***$) \cup D$ in $\mathcal{I}$, (c) |***D***| ≥ Ω_ε(|***C***|).



Input: $S \in \mathcal{I}$, $i_0 \ni S$, matching $M \subseteq \mathcal{E}_{(\frac{1}{3}-\frac{\delta}{3}-\varepsilon)T}$ covering $S \setminus i_0$, layer index ℓ.

Discovered nodes $C = \{i_0\}$, add edges $A = \varnothing$, blocking edges $B = \varnothing$, matching

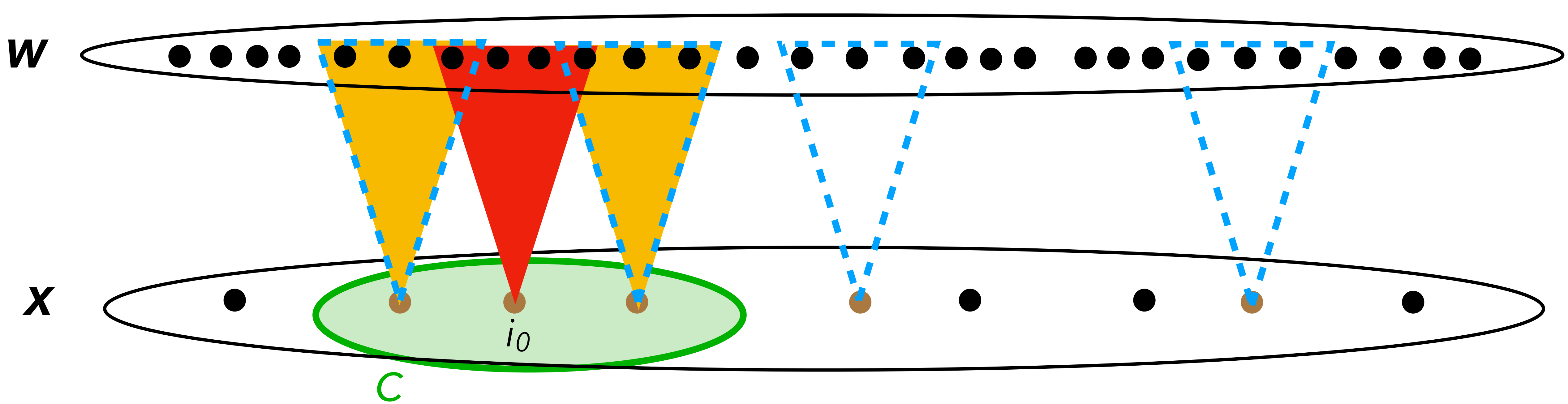
Repeat until termination

1. Find candidate add edges in $\mathcal{E}_{(\frac{1}{3}-\frac{\delta}{3}-\frac{\varepsilon}{2})T}$ that are:

(a) Disjoint to resources in A and B, (b) cover $D \subseteq X$, with $(S \setminus C) \cup D$ in $\mathcal{I}$,

(c) $|D| \geq \Omega_\varepsilon(|C|)$.

2. **If** add edges intersect $\Omega_\varepsilon(|C|)$ edges of matching:
Add intersected matching edges to B, update A and C, and layer index $\ell+1$



Input: $S \in \mathcal{I}$, $i_0 \ni S$, matching $M \subseteq \mathcal{E}_{(\frac{1}{3}-\frac{\delta}{3}-\varepsilon)T}$ covering $S \setminus i_0$, layer index ℓ.

Discovered nodes $C = \{i_0\}$, add edges $A = \varnothing$, blocking edges $B = \varnothing$, matching

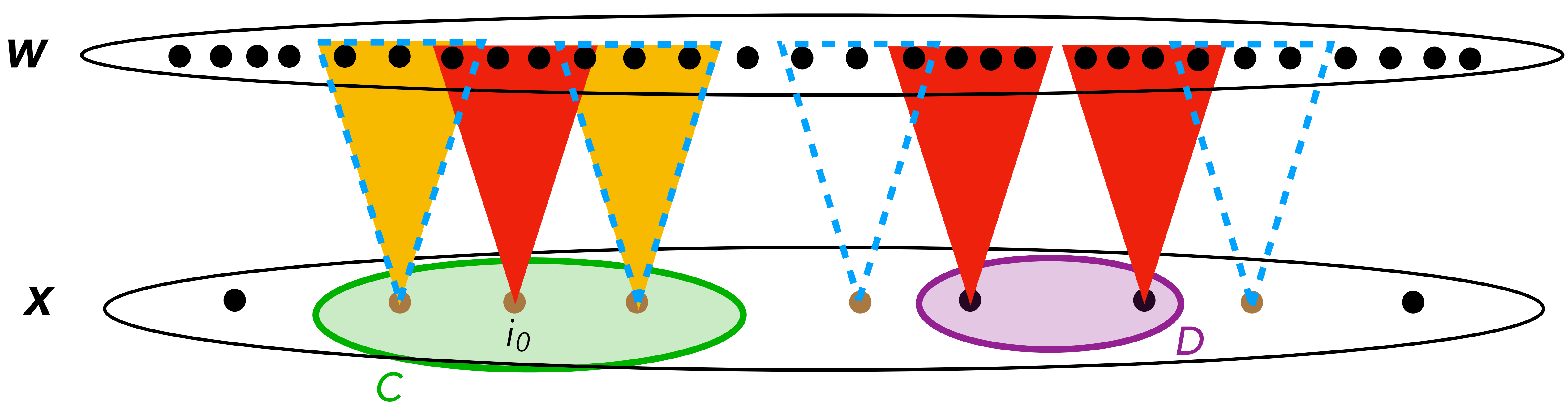
Repeat until termination

1. Find candidate add edges in $\mathcal{E}_{(\frac{1}{3}-\frac{\delta}{3}-\frac{\varepsilon}{2})T}$ that are:

(a) Disjoint to resources in A and B, (b) cover $D \subseteq X$, with $(S \setminus C) \cup D$ in $\mathcal{I}$, (c) $|D| \geq \Omega_\varepsilon(|C|)$.

2. **If** add edges intersect $\Omega_\varepsilon(|C|)$ edges of matching:

Add intersected matching edges to B, update A and C, and layer index $\ell+1$



Input: $S \in \mathcal{I}$, $i_0 \ni S$, matching $M \subseteq \mathcal{E}_{(\frac{1}{3}-\frac{\delta}{3}-\varepsilon)T}$ covering $S \setminus i_0$, layer index ℓ.

Discovered nodes $C = \{i_0\}$, add edges $A = \varnothing$, blocking edges $B = \varnothing$, matching

Repeat until termination

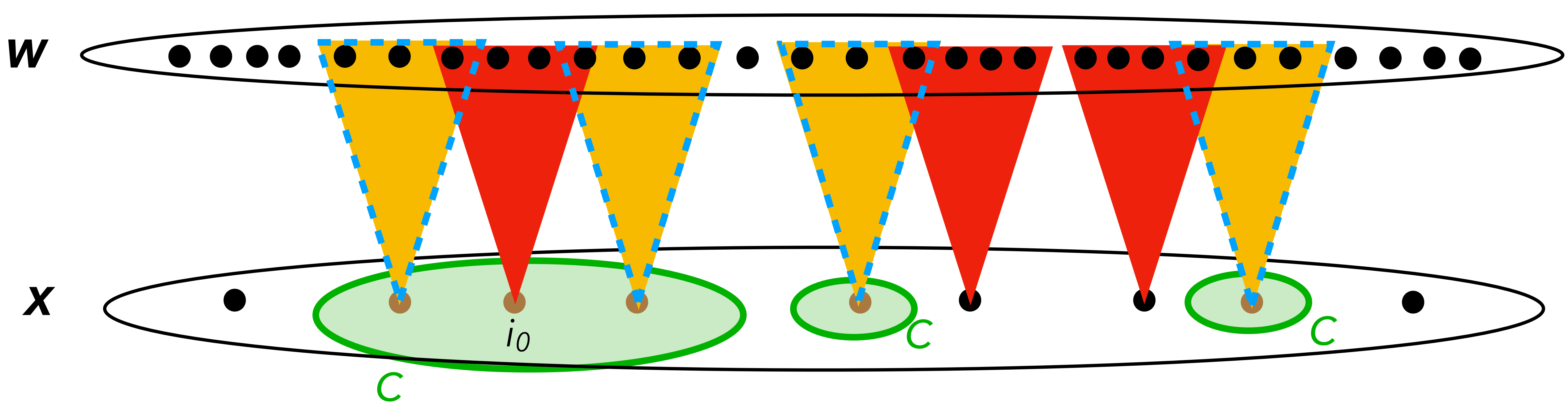
1. Find candidate add edges in $\mathcal{E}_{(\frac{1}{3}-\frac{\delta}{3}-\frac{\varepsilon}{2})T}$ that are:

(a) Disjoint to resources in A and B, (b) cover $D \subseteq X$, with $(S \setminus C) \cup D$ in $\mathcal{I}$,

(c) $|D| \geq \Omega_\varepsilon(|C|)$.

2. **If** add edges intersect $\Omega_\varepsilon(|C|)$ edges of matching:

Add intersected matching edges to B, update A and C, and layer index $\ell+1$



Input: **S** $\in \mathcal{I}$, $i_0 \ni S$, matching $M \subseteq \mathcal{E}_{(\frac{1}{3}-\frac{\delta}{3}-\varepsilon)T}$ covering **S** $\setminus i_0$, layer index ℓ.

Discovered nodes **C** = $\{i_0\}$, add edges **A** = $\varnothing$, blocking edges **B**=$\varnothing$, matching

Repeat until termination

1. Find candidate add edges in $\mathcal{E}_{(\frac{1}{3}-\frac{\delta}{3}-\frac{\varepsilon}{2})T}$ that are:

(a) Disjoint to resources in **A** and **B**, (b) cover **D** $\subseteq X$, with $(S \setminus \mathbf{C}) \cup D$ in $\mathcal{I}$, (c) $|\mathbf{D}| \geq \Omega_\varepsilon(|\mathbf{C}|)$.

2. **If** add edges intersect $\Omega_\varepsilon(|\mathbf{C}|)$ edges of matching:

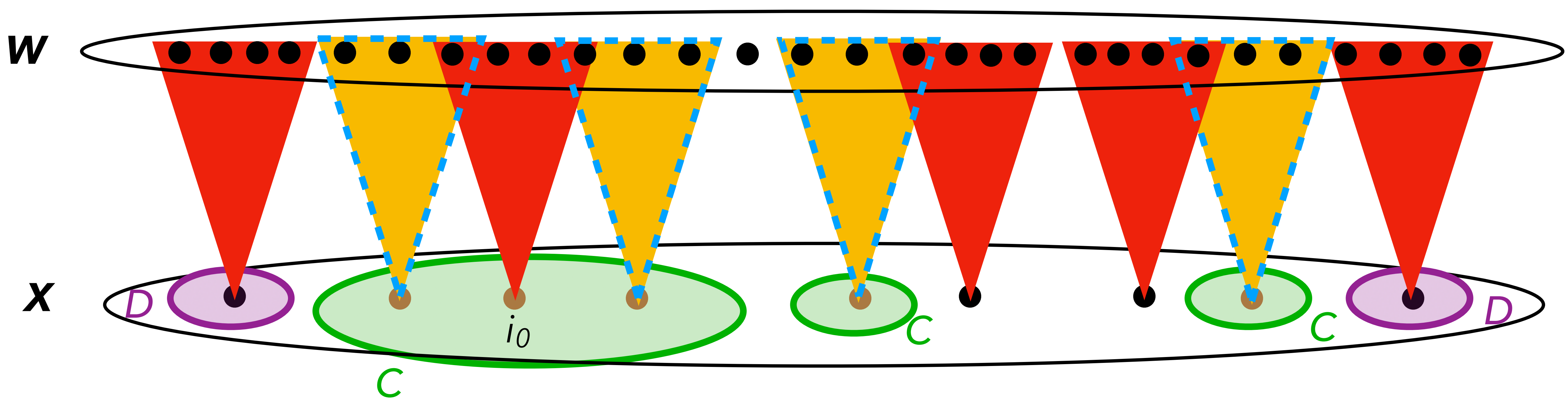
Add intersected matching edges to **B**, update **A** and **C**, and layer index $\ell+1$

3. **Otherwise** $\Omega_\varepsilon(|\mathbf{C}|)$ of add edges have resources summing to value

> (1/3-δ/3-ε)T free from matching and add edges:

If add edge covers i_1 with $S \cup \{i_1\}$ in $\mathcal{I}$, **END**.

Swap C^* from layer ℓ^* in matching. Update $S= S\setminus C^* \cup D^*$, $\ell=\ell^*$, **B**, **A**, **C**



Input: ***S*** $\in \mathcal{I}$, $i_0 \ni S$, matching $M \subseteq \mathcal{E}_{(\frac{1}{3}-\frac{\delta}{3}-\varepsilon)T}$ covering ***S*** $\setminus i_0$, layer index ℓ.

Discovered nodes ***C*** $= \{i_0\}$, add edges ***A*** $= \varnothing$, blocking edges ***B*** $=\varnothing$, matching

Repeat until termination

1. Find candidate add edges in $\mathcal{E}_{(\frac{1}{3}-\frac{\delta}{3}-\frac{\varepsilon}{2})T}$ that are:

(a) Disjoint to resources in ***A*** and ***B***, (b) cover ***D*** $\subseteq X$, with $(S \setminus$ ***C***$) \cup D$ in $\mathcal{I}$,

(c) $|$***D***$| \geq \Omega_\varepsilon(|$***C***$|)$.

2. **If** add edges intersect $\Omega_\varepsilon(|$***C***$|)$ edges of matching:

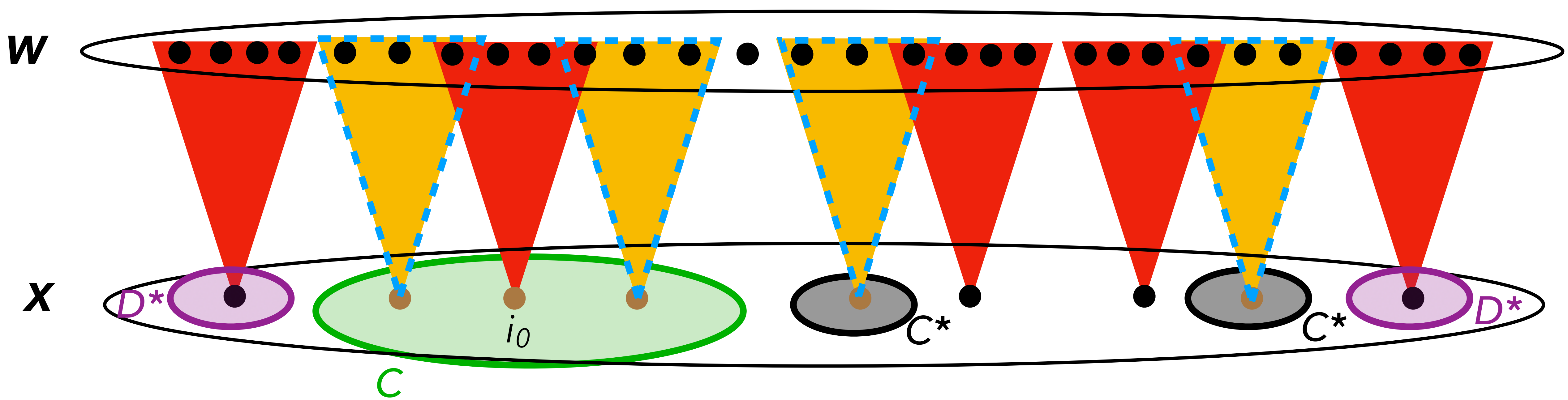
Add intersected matching edges to ***B***, update ***A*** and ***C***, and layer index $\ell+1$

3. **Otherwise** $\Omega_\varepsilon(|$**C**$|)$ of add edges have resources summing to value

> (1/3-δ/3-ε)T free from matching and add edges:

If add edge covers i_1 with $S \cup \{i_1\}$ in $\mathcal{I}$, **END**.

Swap C^* from layer ℓ^* in matching. Update $S= S \setminus C^* \cup D^*$, $\ell=\ell^*$, ***B***, ***A***, ***C***



Input: ***S*** $\in \mathcal{I}$, $i_0 \ni S$, matching $M \subseteq \mathcal{E}_{(\frac{1}{3}-\frac{\delta}{3}-\varepsilon)T}$ covering ***S*** $\setminus i_0$, layer index ℓ.

Discovered nodes ***C*** = $\{i_0\}$, add edges ***A*** = ∅, blocking edges ***B***=∅, matching

Repeat until termination

1. Find candidate add edges in $\mathcal{E}_{(\frac{1}{3}-\frac{\delta}{3}-\frac{\varepsilon}{2})T}$ that are:

(a) Disjoint to resources in ***A*** and ***B***, (b) cover ***D*** $\subseteq X$, with $(S \setminus$ ***C***$) \cup D$ in $\mathcal{I}$,

(c) $|$***D***$| \geq \Omega_\varepsilon(|$***C***$|)$.

2. **If** add edges intersect $\Omega_\varepsilon(|$***C***$|)$ edges of matching:

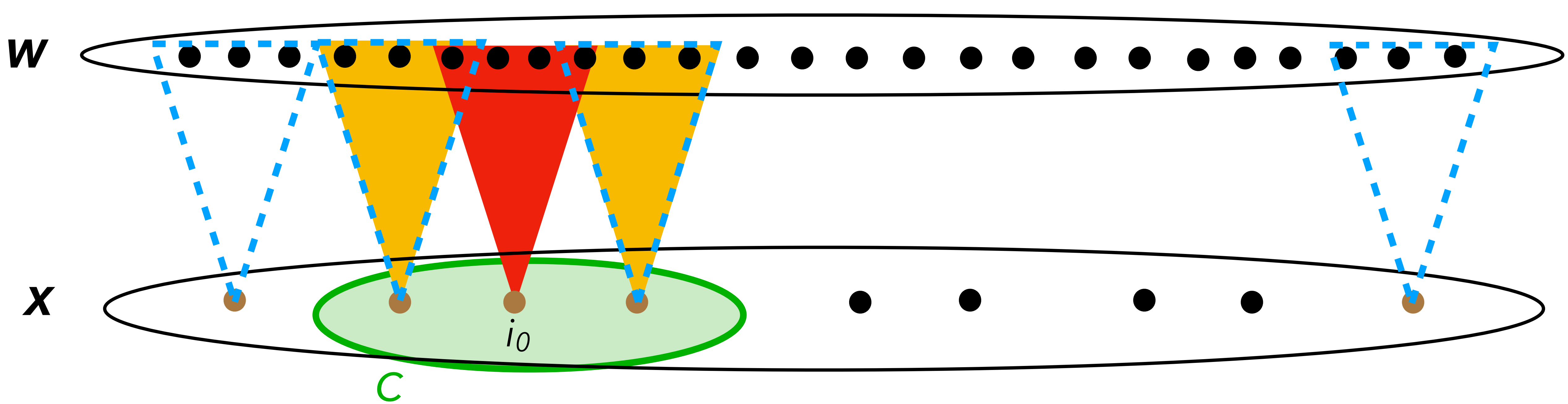
Add intersected matching edges to ***B***, update ***A*** and ***C***, and layer index $\ell+1$

3. **Otherwise** $\Omega_\varepsilon(|$***C***$|)$ of add edges have resources summing to value

> (1/3-δ/3-ε)T free from matching and add edges:

If add edge covers i_1 with $S \cup \{i_1\}$ in $\mathcal{I}$, **END**.

Swap C^* from layer ℓ^* in matching. Update $S = S \setminus C^* \cup D^*$, $\ell=\ell^*$, ***B***, ***A***, ***C***

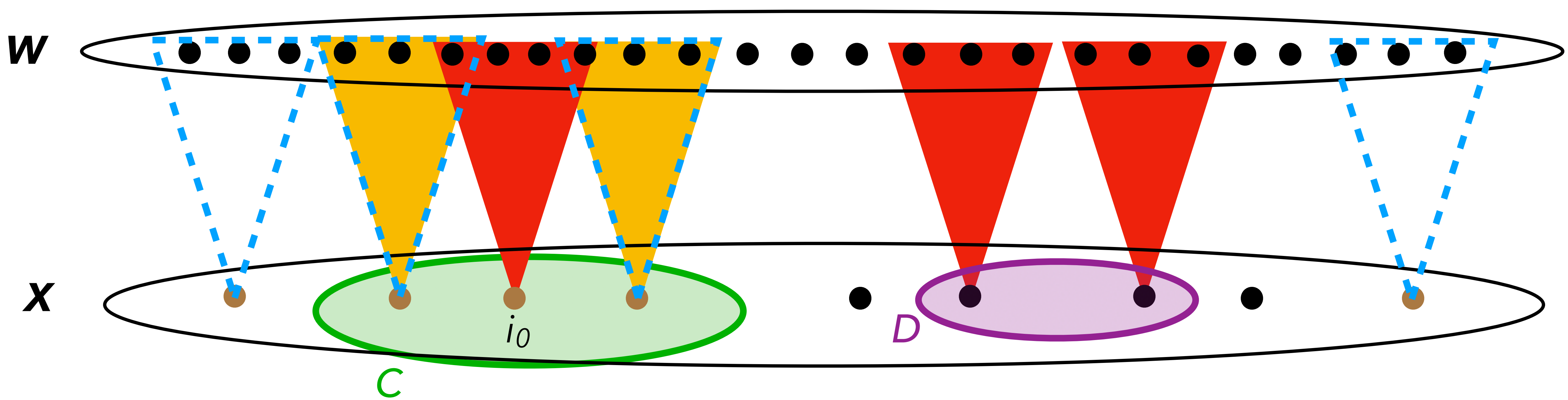


Input: $S \in \mathcal{I}$, $i_0 \ni S$, matching $M \subseteq \mathcal{E}_{(\frac{1}{3}-\frac{\delta}{3}-\varepsilon)T}$ covering $S \setminus i_0$, layer index ℓ.

Discovered nodes $C = \{i_0\}$, add edges $A = \varnothing$, blocking edges $B = \varnothing$, matching

Repeat until termination

1. Find candidate add edges in $\mathcal{E}_{(\frac{1}{3}-\frac{\delta}{3}-\frac{\varepsilon}{2})T}$ that are:
   (a) Disjoint to resources in A and B, (b) cover $D \subseteq X$, with $(S \setminus C) \cup D$ in $\mathcal{I}$,
   (c) $|D| \geq \Omega_\varepsilon(|C|)$.
2. **If** add edges intersect $\Omega_\varepsilon(|C|)$ edges of matching:
   Add intersected matching edges to B, update A and C, and layer index $\ell+1$
3. **Otherwise** $\Omega_\varepsilon(|C|)$ of add edges have resources summing to value
   > (1/3-δ/3-ε)T free from matching and add edges:
   If add edge covers i_1 with $S \cup \{i_1\}$ in $\mathcal{I}$, **END**.
   Swap C^* from layer ℓ^* in matching. Update $S = S \setminus C^* \cup D^*$, $\ell = \ell^*$, B, A, C



Input: $S \in \mathcal{I}$, $i_0 \ni S$, matching $M \subseteq \mathcal{E}_{(\frac{1}{3}-\frac{\delta}{3}-\varepsilon)T}$ covering $S \setminus i_0$, layer index ℓ.

Discovered nodes $C = \{i_0\}$, add edges $A = \varnothing$, blocking edges $B = \varnothing$, matching

Repeat until termination

1. Find candidate add edges in $\mathcal{E}_{(\frac{1}{3}-\frac{\delta}{3}-\frac{\varepsilon}{2})T}$ that are:

(a) Disjoint to resources in A and B, (b) cover $D \subseteq X$, with $(S \setminus C) \cup D$ in $\mathcal{I}$, (c) $|D| \geq \Omega_\varepsilon(|C|)$.

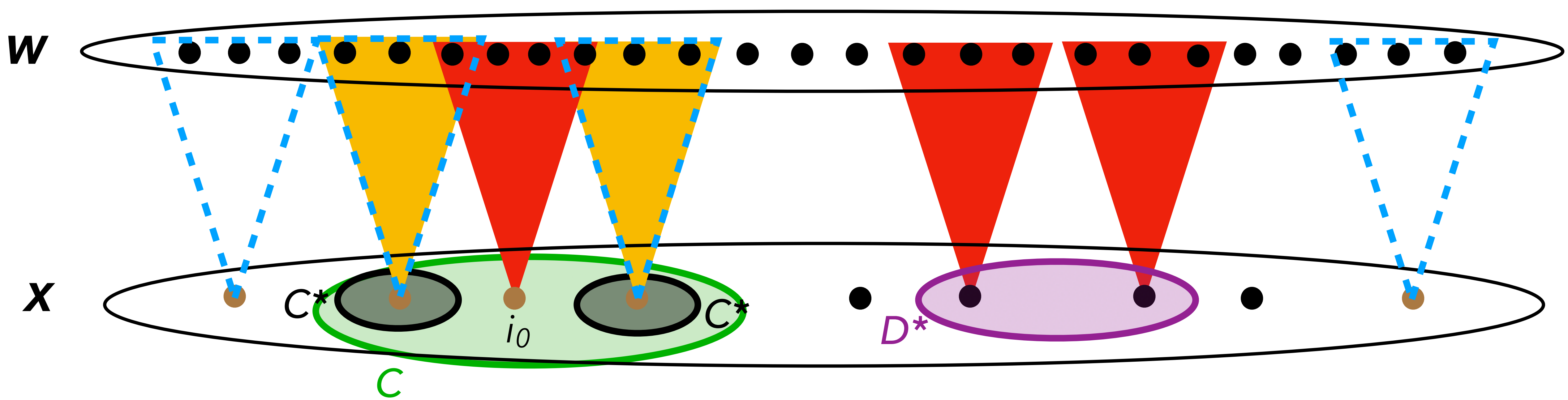
2. **If** add edges intersect $\Omega_\varepsilon(|C|)$ edges of matching:

Add intersected matching edges to B, update A and C, and layer index $\ell+1$

3. **Otherwise** $\Omega_\varepsilon(|C|)$ of add edges have resources summing to value > (1/3-δ/3-ε)T free from matching and add edges:

If add edge covers i_1 with $S \cup \{i_1\}$ in $\mathcal{I}$, **END**.

Swap C^* from layer ℓ^* in matching. Update $S = S \setminus C^* \cup D^*$, $\ell = \ell^*$, B, A, C

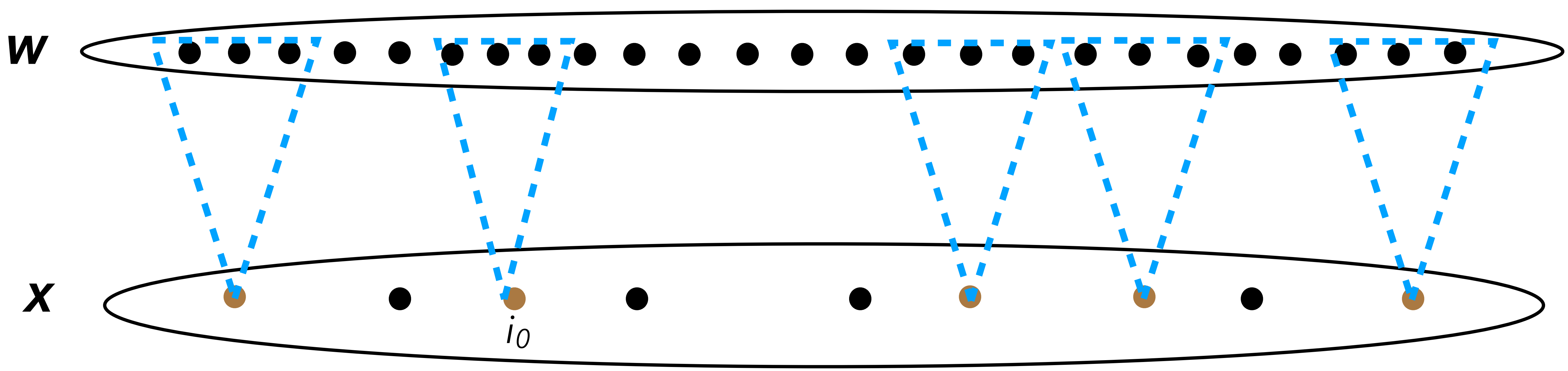


Input: $S \in \mathcal{I}$, $i_0 \ni S$, matching $M \subseteq \mathcal{E}_{(\frac{1}{3}-\frac{\delta}{3}-\varepsilon)T}$ covering $S \setminus i_0$, layer index ℓ.

Discovered nodes $C = \{i_0\}$, add edges $A = \emptyset$, blocking edges $B = \emptyset$, matching

Repeat until termination

1. Find candidate add edges in $\mathcal{E}_{(\frac{1}{3}-\frac{\delta}{3}-\frac{\varepsilon}{2})T}$ that are:
   (a) Disjoint to resources in A and B, (b) cover $D \subseteq X$, with $(S \setminus C) \cup D$ in $\mathcal{I}$,
   (c) $|D| \geq \Omega_\varepsilon(|C|)$.
2. **If** add edges intersect $\Omega_\varepsilon(|C|)$ edges of matching:
   Add intersected matching edges to B, update A and C, and layer index $\ell+1$
3. **Otherwise** $\Omega_\varepsilon(|C|)$ of add edges have resources summing to value > (1/3-δ/3-ε)T free from matching and add edges:
   If add edge covers i_1 with $S \cup \{i_1\}$ in $\mathcal{I}$, END.
   Swap C^* from layer ℓ^* in matching. Update $S = S \setminus C^* \cup D^*$, $\ell = \ell^*$, B, A, C



Augmenting tree: termination

Augmenting tree: termination

Define *signature vector* to show algorithm terminates quickly:

$s = \{s_1, s_2, \cdots, s_\ell, \infty\}, \qquad s_j = O(\log($**# blocking edges by layer j**$))$

Augmenting tree: termination

Define *signature vector* to show algorithm terminates quickly:

$s = \{s_1, s_2, \cdots, s_\ell, \infty\}, \qquad s_j = O(\log(\textbf{\# blocking edges by layer j}))$

s decreases lexicographically after each iteration and # of signature vectors is polynomial in $n = |X|+|W|$.

=> poly many iterations

Open Problems

Open Problems

p_{ij} in $\{0, p_j\}$

Santa Claus (Restricted Max Min Fair Allocation)

Approximation factor between 2 and 4

Integrality gap of our new LP between 2 and 4

Open Problems

p_{ij} in $\{0, p_j\}$

Santa Claus (Restricted Max Min Fair Allocation)

Approximation factor between 2 and 4

Integrality gap of our new LP between 2 and 4

Arbitrary p_{ij}

Unrestricted Max Min Fair Allocation

NP-hard to approximate within factor < 2 (like restricted)

$O(\log^{10} n)$-approximation in quasi-polynomial time
[Chakrabarty, Chuzhoy, Khanna '09]

CLP has root *n* gap

Open Problems

p_{ij} in $\{0, p_j\}$

Santa Claus (Restricted Max Min Fair Allocation)

Approximation factor between 2 and 4

Integrality gap of our new LP between 2 and 4

Arbitrary p_{ij}

Unrestricted Max Min Fair Allocation

NP-hard to approximate within factor < 2 (like restricted)

$O(\log^{10} n)$-approximation in quasi-polynomial time [Chakrabarty, Chuzhoy, Khanna '09]

CLP has root n gap

Other uses for Matroid Max-Min Fair Allocation?

Open Problems

p_{ij} in $\{0, p_j\}$

Santa Claus (Restricted Max Min Fair Allocation)

Approximation factor between 2 and 4

Integrality gap of our new LP between 2 and 4

Arbitrary p_{ij}

Unrestricted Max Min Fair Allocation

NP-hard to approximate within factor < 2 (like restricted)

O(log^{10}n)-approximation in quasi-polynomial time
[Chakrabarty, Chuzhoy, Khanna '09]

CLP has root *n* gap

Other uses for Matroid Max-Min Fair Allocation?

Thanks!