Faster Fully Dynamic Matching With Small Approximation Ratios

Aaron Bernstein

Cliff Stein

Columbia University

Static Maximum Matching

- **Unweighted n-vertex, m-edge** graph G
- Goal: find a maximum cardinality matching (MCM) in G
- Exact algorithms:
  - $O(mn^{1/2})$ [Micali,Vazirani FOCS 1980]
  - $O(MM(n)) \sim O(n^{2.36})$ [Sankowski FOCS 2004]
- Exact algorithms for bipartite graphs:
  - $O(mn^{1/2})$ [Hopcroft-Karp, Sicomp, 1973]
  - $\tilde{O}(m^{10/7})$ [Madry FOCS 2013].
- Approximate algorithms:
  - $(1 + \varepsilon)$-approximation in time $\tilde{O}(m/\varepsilon)$

Dynamic Exact MCM

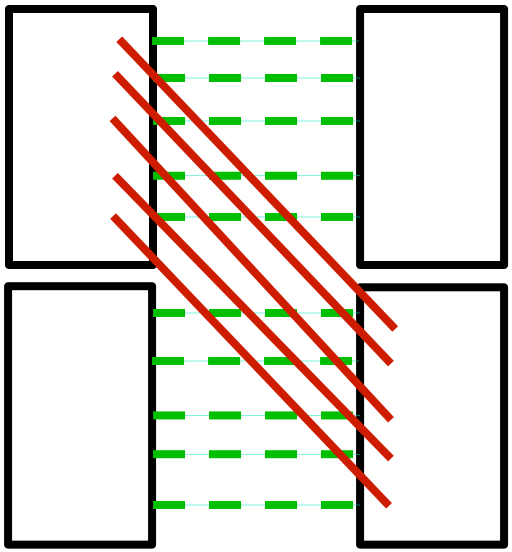
- Dynamic – edges are inserted and deleted, we maintain a matching
- Trivial:
  - Recompute - $O(mn^{1/2})$ or $O(MM(n))$ per update
  - Find augmenting path - $O(m)$ per update
- Sankowski's algorithm:
  - $O(n^{1.495})$ randomized [SODA 2007].
- What about dynamic **approximate** MCM?
- Note: The "adwords" problem is a different dynamic matching problem

Dynamic Approximate MCM

- Variants:
  - Randomized vs. deterministic
  - Worst-case vs. amortized
  - Bipartite vs. non-bipartite
- Two very different problems:
  - 2-or-worse approximation.
  - Better-than-2 approximation.

Maximal Matching

- Maximal: can't add an edge to the matching
- Maximal ≠ Maximum
- Maximal matching is 2-approximate.
- Maximal matching purely local
  - If x and y are free, can always match x and y, without considering rest of graph.
- 1.99 approximation not purely local.



- Maximum has size 10
- Maximal has size 5

Dynamic Approximate MCM (before us)

- 2-or-worse Approximation:
  - $O(\log(n))$ update, 2-approximation, randomized [Baswana, Gupta, Sen, FOCS 2011]
- Better-than-2 approximation:
  - $O(m^{1/2})$ update, $(1+\varepsilon)$-approximation. [Gupta, Peng, FOCS 2013]
- Deterministic Algorithms:
  - $O(m^{1/2})$ update, $(1+\varepsilon)$-approximation [Gupta, Peng]
  - $O(m^{1/3})$ update, $(3+\varepsilon)$-approximation [Bhattacharya, Henzinger, Italiano, SODA 2015]
- Our Goals:
  - faster better-than-2 approximation
  - faster deterministic algorithm.

Our results

- **Recall State of the art:**
  - 2-approx., O(log n) update, randomized
  - (1+ε)-approx., $O(m^{1/2})$ update, deterministic
  - (3+ε)-approx., $O(m^{1/3})$ update, deterministic
- **Our results:**
  - (3/2+ε)-approx., $O(m^{1/4})$ update, deterministic
  - Same bounds (different algorithm and analysis for bipartite (ICALP 2015) and general (SODA 2016) graphs
  - Fastest better-than-2 approximation.
  - Fastest deterministic algorithm for **any** constant approx.
  - Also: better results for small arboricity graphs (see papers)

Subsequent to our results

- **Recall our results:**
  - $(3/2+\varepsilon)$-approx., $O(m^{1/4})$ update, deterministic
  - Same bounds (different algorithm and analysis for bipartite (ICALP 2015) and general (SODA 2016) graphs
  - Fastest better-than-2 approximation.
  - Fastest deterministic algorithm for **any** constant approx.
  - Also: better results for small arboricity graphs
- **Simultaneous:**
  - Even better results for small arboricity: [Peleg, Solomon **SODA 2016**]
- **Subsequent:** [Bhattacharya, Henzinger, Nanongkai, STOC 2016]
  - $O(\text{polylog } n)$ update, $(2+\varepsilon)$-approximation
  - $O(n^{\varepsilon})$ update, <2-approximation of value in bipartite graphs

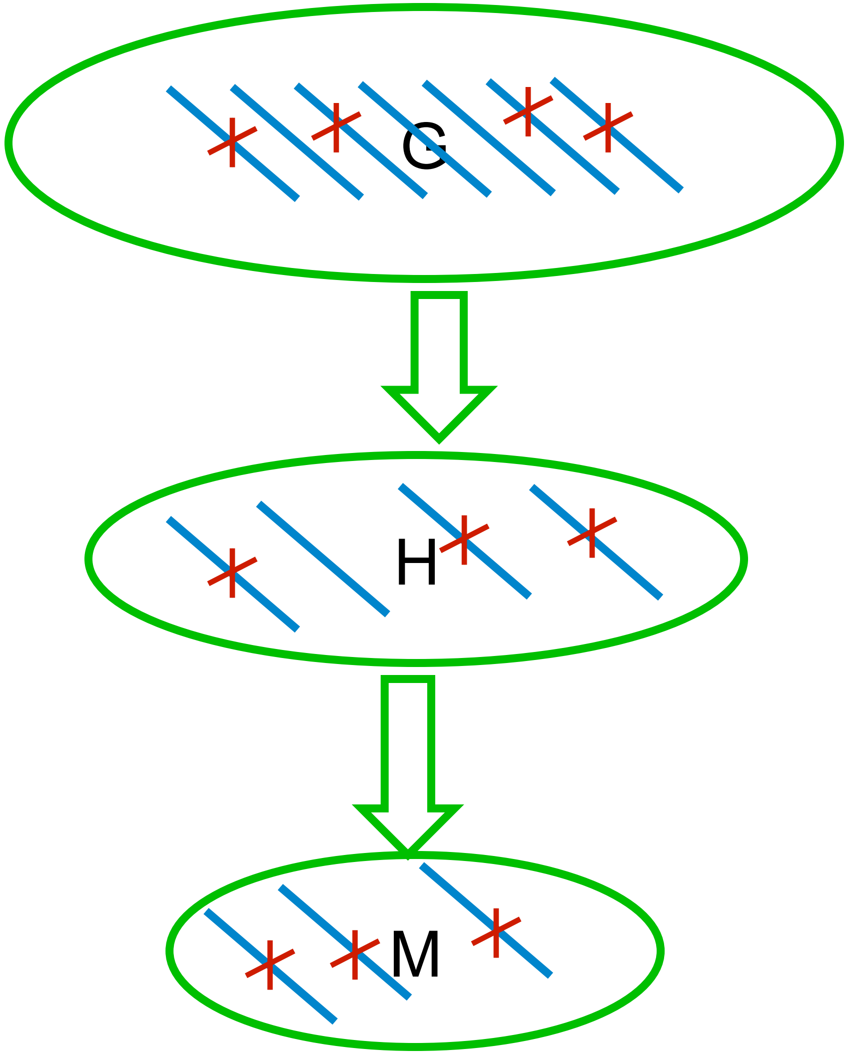
Characterizing Approximate Matchings

- **Exact matching:** matching M contains no augmenting paths → M is maximum
- **2-approximate matching:** matching M is maximal → M is 2-approximate
  - **Maximal:** for every edge (u,v), u or v (or both) is matched
- **Better-than-2 approximation:** if all augmenting paths w.r.t M have length > (2k-1) then M is a (1+1/k)-approx. matching.
  - All path lengths > 1 → 2-approximation
  - All path lengths > 3 → 3/2-approximation.
  - All path lengths > 1/ε → (1+O(ε))-approximation.
  - Unlike maximality, cannot be expressed in terms of local constraints on each edge

Characterizing Approximate Matchings

- **Ideally show:** A matching M that satisfies some set of local constraints is a (1+ε)-approximate matching (or at least better-than-2 approximate).
  - We were unable to do this even for bipartite graphs.
- **Our results:** use local constraints to characterize a **small subgraph** H of G such that H is guaranteed to contain a large matching. (used in previous work, but with different small subgraph, typically fractional or b-matching.)
- Recall a b-matching is a subgraph where each vertex has degree at most b

Matching-Preserving Subgraph



- **What we want from H:**
  - 1. M is easy to maintain in H
  - 2. Large matching in H is relatively large w.r.t to G
  - 3. Easy to maintain H dynamically.

Edge Degree Constrained Subgraph (EDCS)

- **Defn:** $\mu(H)$ = size of maximal matching in H.
- **Defn:** $\delta_H(u)$ = degree of u in H.

- A subgraph **H** of **G** is an **EDCS(B)** if it has the following properties (B is parameter of our choosing):
  - **1.** For each edge (u,v) in H,

$$\delta_H(u) + \delta_H(v) \le B$$

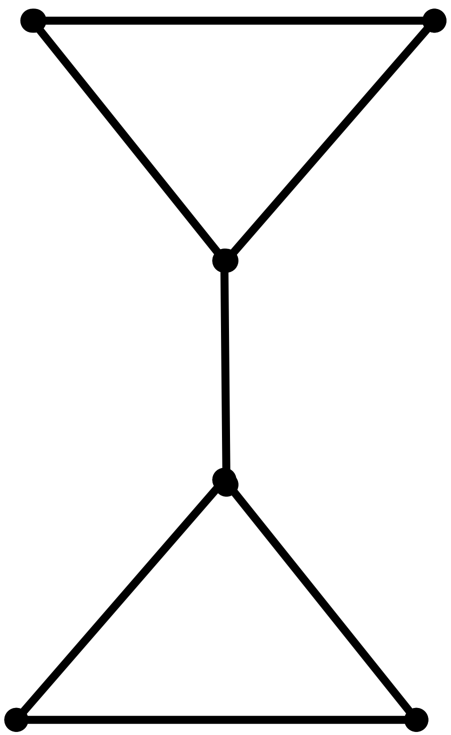
  - **2.** For each edge (u,v) in G\H,

$$\delta_H(u) + \delta_H(v) \ge B(1- \varepsilon^2)$$

Example of EDCS

- Example of EDCS(3):
  - **1.** For each edge (u,v) in H, $\delta_H(u) + \delta_H(v) \leq 3$
  - **2.** For each edge (u,v) in G\H, $\delta_H(u) + \delta_H(v) \geq 2$

$\mu(G) = 3$

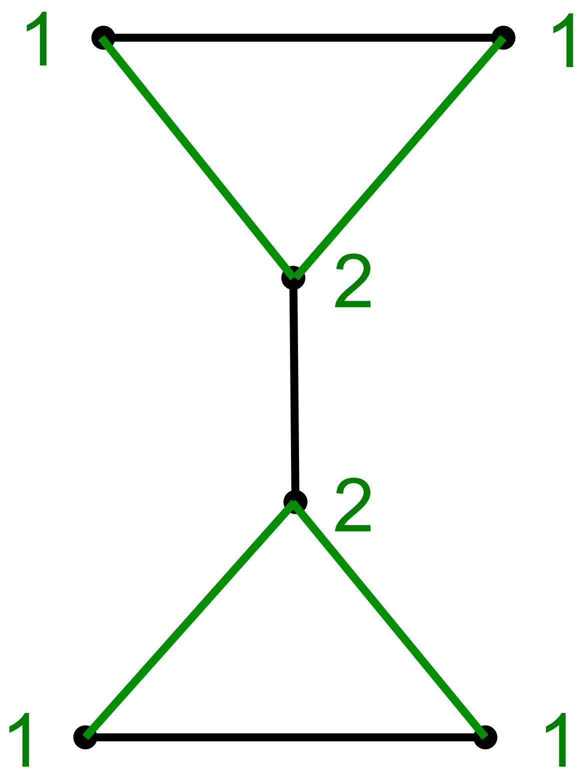


Example of EDCS

- Example of EDCS(3):
  - **1.** For each edge (u,v) in H, $\delta_H(u) + \delta_H(v) \leq 3$
  - **2.** For each edge (u,v) in $G \backslash H$, $\delta_H(u) + \delta_H(v) \geq 2$

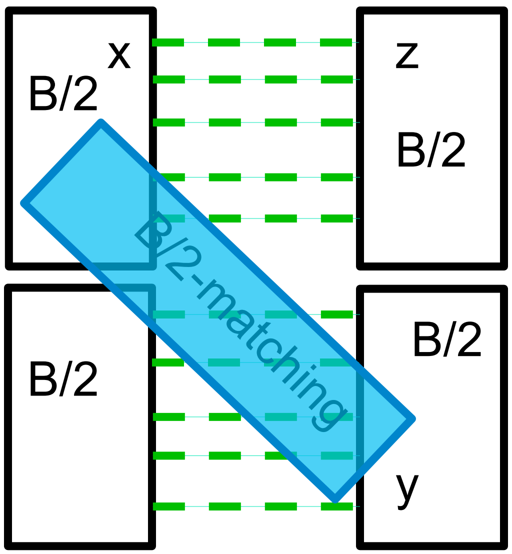
$\mu(G) = 3$

$\mu(H) = 2$



Example of a non-EDCS

- Example of EDCS(B):
  - **1.** For each edge (u,v) in H, $\delta_H(u) + \delta_H(v) \leq B$
  - **2.** For each edge (u,v) in $G \backslash H$, $\delta_H(u) + \delta_H(v) \geq B(1-\varepsilon^2)$



$\delta_H(x) + \delta_H(z) = B/2 + 1$

$\delta_H(x) + \delta_H(y) = B$

- B/2-matching (complete bipartiite graph) only contains a 2-approximation to the maximum matching
- B/2-matching is not an EDCS
- Our “local” algorithm will recognize this

What do we need to do?

1. **Prove the main theorem:** Let H be an EDCS(B) of a graph G. Then: $\mu(H) \geq (2/3-\varepsilon)\,\mu(G)$
   - only true if $B > 4/\varepsilon^2$
   - H has bounded degree B
2. **Algorithm problem 1:** Show how to maintain an EDCS(B) quickly in a dynamic graph
3. **Algorithm problem 2**: Maintain a $(1+\varepsilon)$ approx. matching M in H
   - Easy because H has bounded degree [Gupta, Peng, FOCS 2013]

3/2-approximation follows

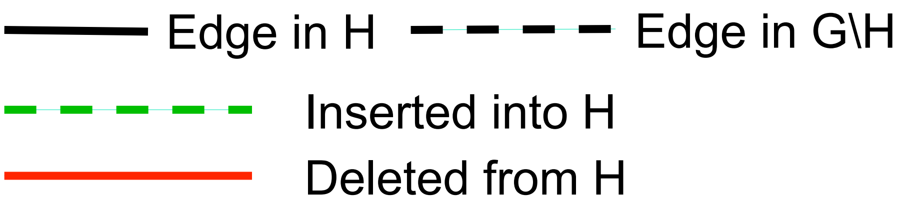
- $\text{Size}(M) \geq (1-\varepsilon)\mu(H) \geq (2/3-\varepsilon)(1-\varepsilon)\mu(G) = (2/3 - O(\varepsilon))\ \mu(G)$

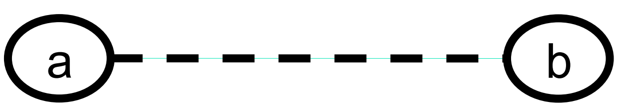
- Now, let's briefly talk about maintaining an EDCS, and then the main theorem

Maintaining an EDCS (bipartite graph)

- EDCS(B):
  - 1. For each edge (u,v) in H, $\delta_H(u) + \delta_H(v) \leq B$
  - 2. For each edge (u,v) in G\H,

    $\delta_H(u) + \delta_H(v) \geq B(1 - \varepsilon^2)$





Maintaining an EDCS (bipartite graph)

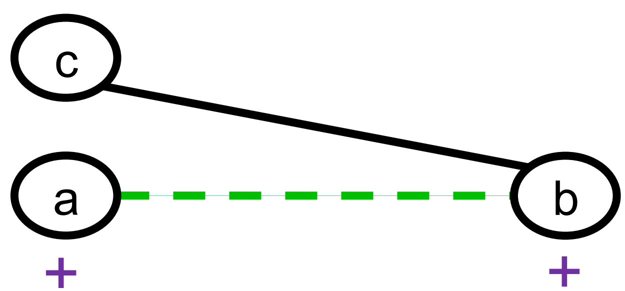
- EDCS(B:
  - 1. For each edge (u,v) in H, $\delta_H(u) + \delta_H(v) \le B$
  - 2. For each edge (u,v) in G/H,

    $\delta_H(u) + \delta_H(v) \ge B(1- \varepsilon^2)$

Edge in H — Edge in G\H

Inserted into H

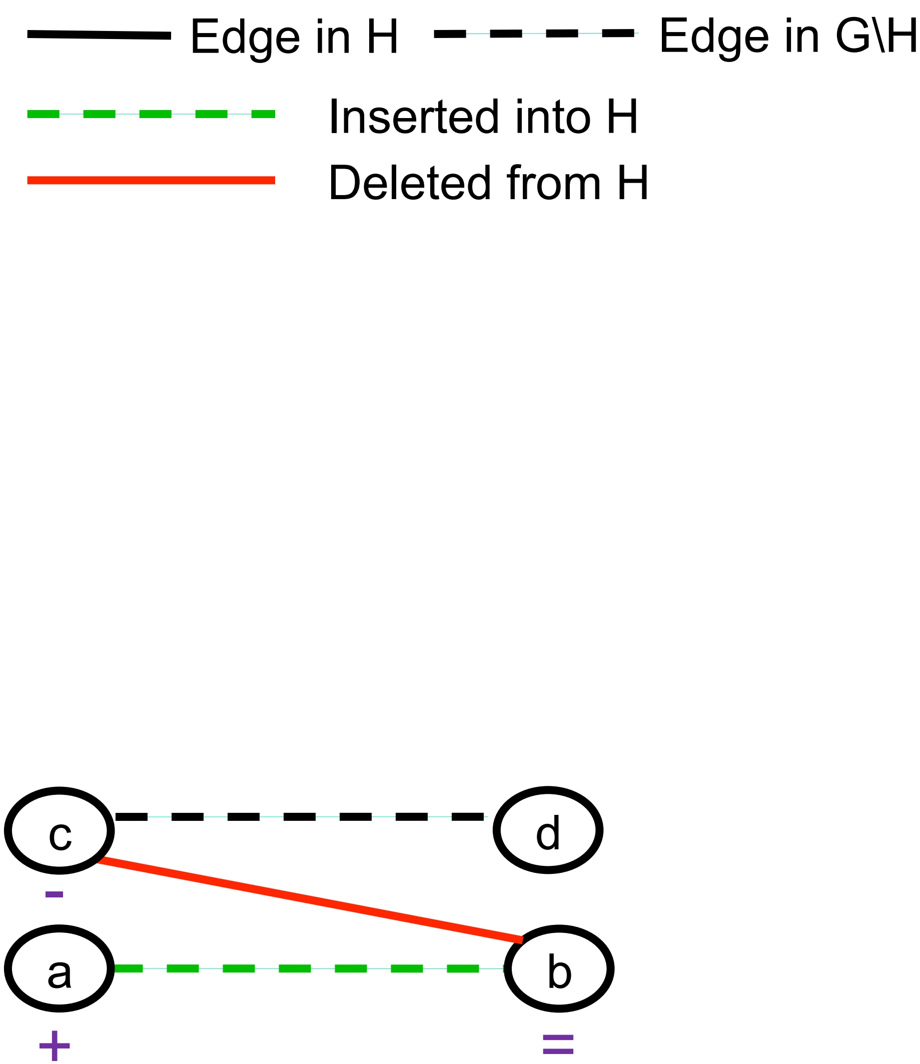
Deleted from H



Maintaining an EDCS (bipartite graph)

- EDCS(B):
  - 1. For each edge (u,v) in H, $\delta_H(u) + \delta_H(v) \leq B$
  - 2. For each edge (u,v) in G/H,

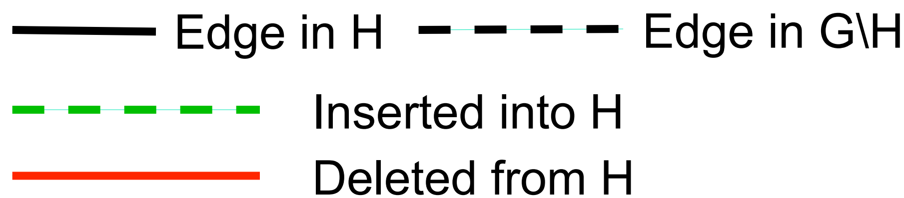
    $\delta_H(u) + \delta_H(v) \geq B(1- \varepsilon^2)$

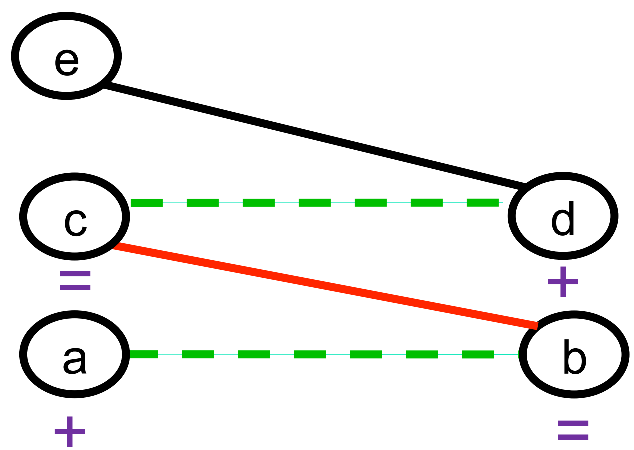


Maintaining an EDCS (bipartite graph)

- EDCS(B):
  - 1. For each edge (u,v) in H, $\delta_H(u) + \delta_H(v) \leq B$
  - 2. For each edge (u,v) in G/H,

    $\delta_H(u) + \delta_H(v) \geq B(1- \varepsilon^2)$

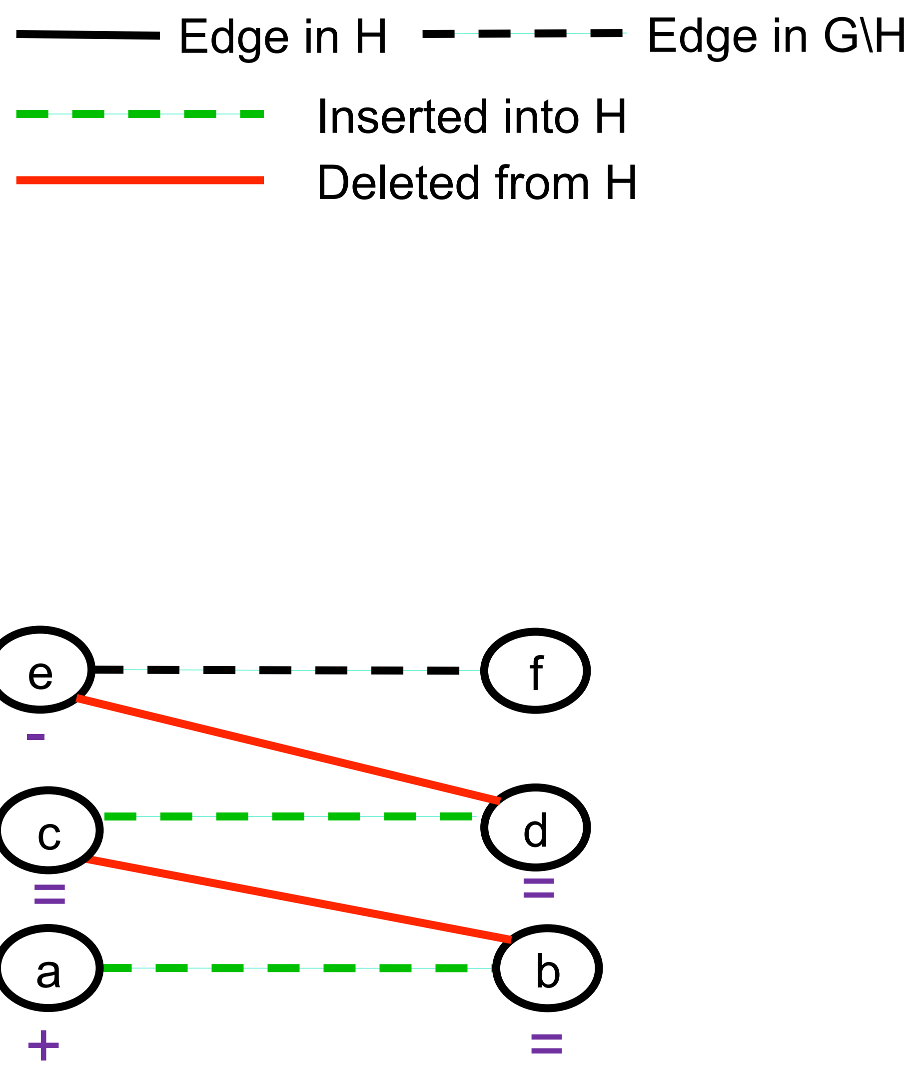




Maintaining an EDCS (bipartite graph)

- EDCS(B):
  - 1. For each edge (u,v) in H, $\delta_H(u) + \delta_H(v) \le B$
  - 2. For each edge (u,v) in G/H,

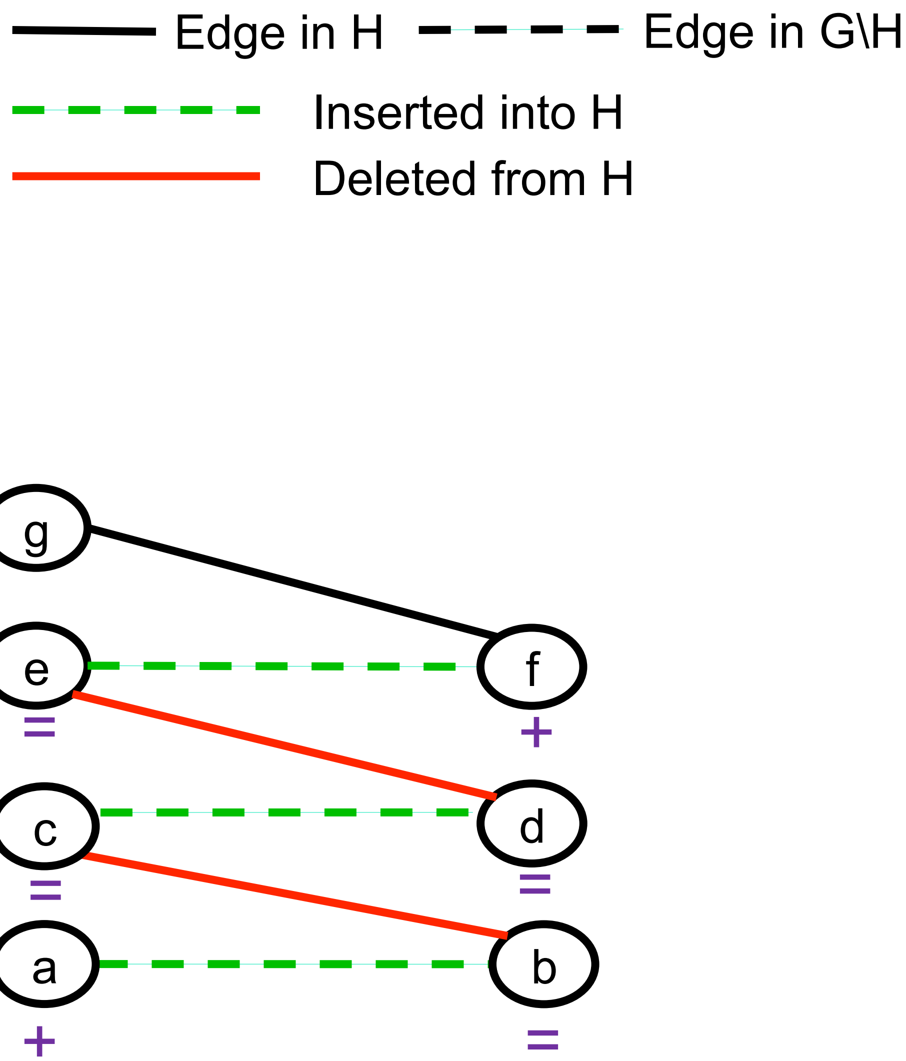
    $\delta_H(u) + \delta_H(v) \ge B(1- \varepsilon^2)$



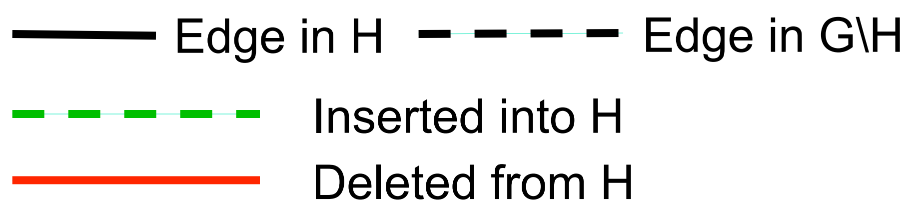
Maintaining an EDCS (bipartite graph)

- EDCS(B):
  - 1. For each edge (u,v) in H, $\delta_H(u) + \delta_H(v) \leq B$
  - 2. For each edge (u,v) in G/H,

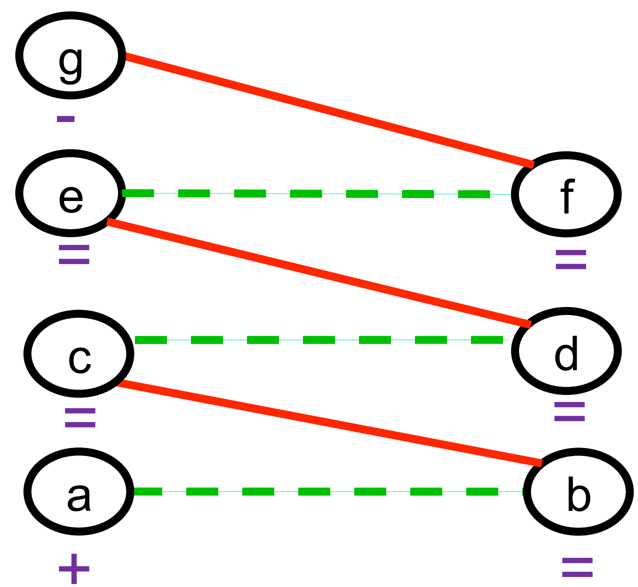
    $\delta_H(u) + \delta_H(v) \geq B(1- \varepsilon^2)$



Maintaining an EDCS (bipartite graph)

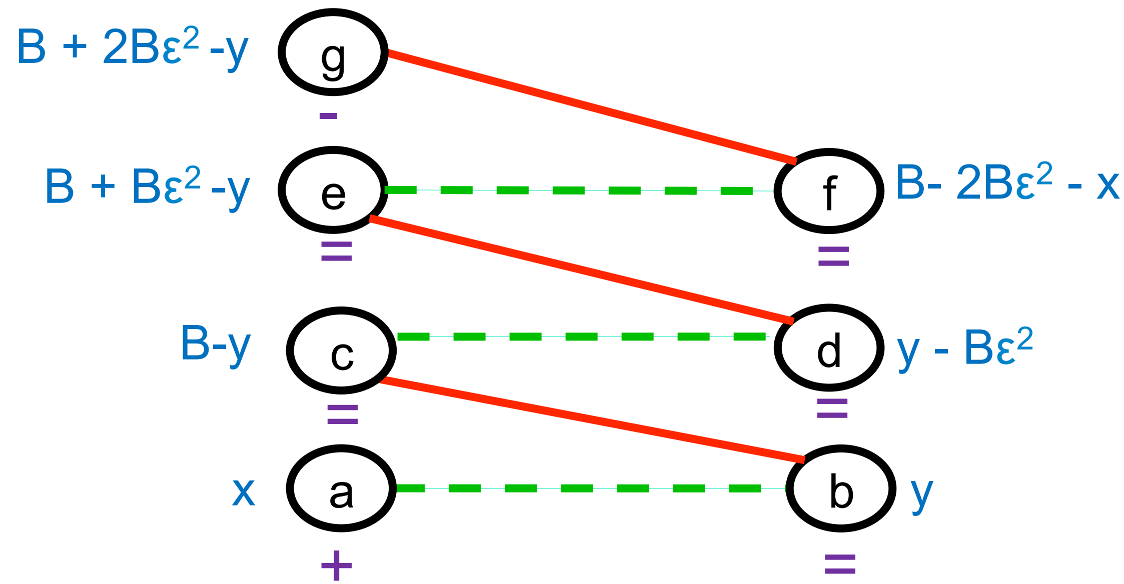


- EDCS(B, λ):
  - 1. For each edge (u,v) in H, $\delta_H(u) + \delta_H(v) \leq B$
  - 2. For each edge (u,v) in G/H, $\delta_H(u) + \delta_H(v) \geq B(1- \varepsilon^2)$
- Can show: only need $O(\varepsilon^{-2})$ rebalances
- Difference from augmenting path: **Any** sequences of rebalances will work and be short.
  - No “backtracking”



Bipartite running time

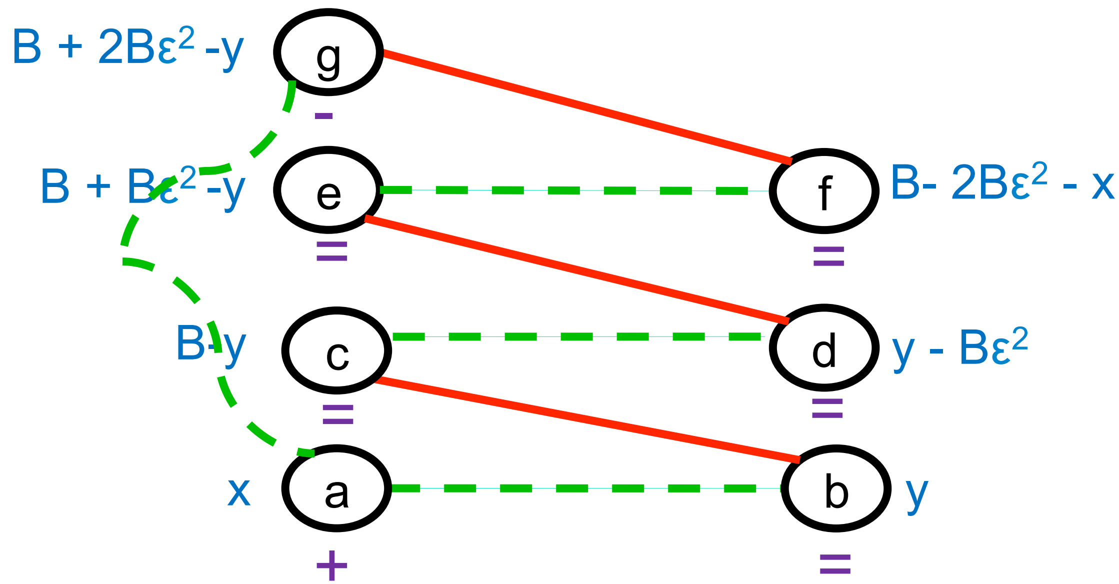
- EDCS(B):
  - 1. For each edge (u,v) in H, $\delta_H(u) + \delta_H(v) \leq B$
  - 2. For each edge (u,v) in G\H, $\delta_H(u) + \delta_H(v) \geq B(1-\varepsilon^2)$



- Only $O(\varepsilon^{-2})$ rebalances per edge change in G
- Easy: Rebalance a node in time O(degree) = O(n)
- Can do: Rebalance in time O(arboricity) (dynamic orientation++)
- General Graphs: arboricity = $m^{1/2}$
- Lazy Rebalancing: buffer of $B\varepsilon^{-2}$ between EDCS properties
- Resulting worst case update time: $O(m^{1/2}\varepsilon^{-2}/B)$

NonBipartite running time

- EDCS(B, λ):
  - 1. For each edge (u,v) in H, $\delta_H(u) + \delta_H(v) \le B$
  - 2. For each edge (u,v) in G\H, $\delta_H(u) + \delta_H(v) \ge B(1-\varepsilon^2)$

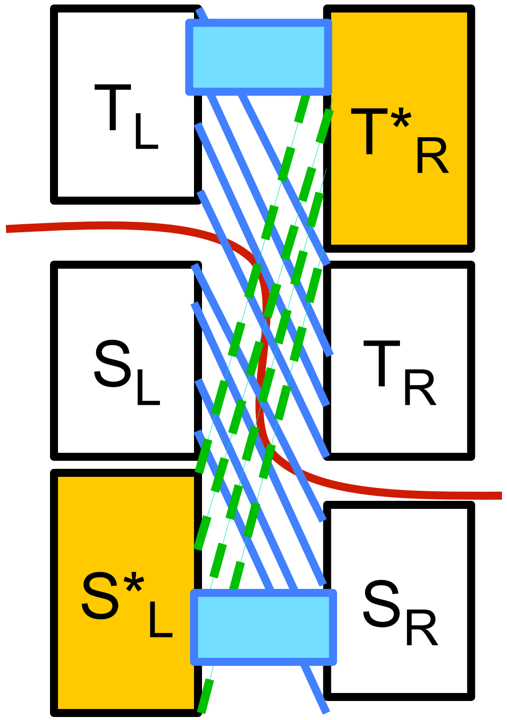


- Can't show $O(\varepsilon^{-2})$ rebalances per edge change in G, but can show a simliar amortized bound, using
  - Potential function
  - Dynamic orientation
  - Lots of data structures
- Implicitly dealing with blossoms

Showing an EDCS contains a large matching

Proof sketch for bipartite graphs

- The residual graph defined by the maximum matching in H (n vertices per side).
- Contradiction Assumption: $\mu(H) < (2/3-\varepsilon)n$



- H contains no edges crossing cut.
- $|S^*_L| = |T^*_R| \geq (n/3)(1+\varepsilon)$
- Assume: $|S_L|=|S_R|=|T_L|=|T_R| < n/3$
- $\geq (n/3)(1+\varepsilon)$ disjoint crossing edges in G\H
- Assume: crossing edges go from S^*_L to T^*_R
- **Then:** average degree in H of $|S^*_L| \cup |T^*_R| \geq B/2(1-\varepsilon^2) \sim B/2$ (Prop. 2)
- Can't fit degree of S^*_L in smaller S_R (Prop. 1)

- **1.** For edge (u,v) in H, $\delta(u)+\delta(v) \leq B$
- **2.** For edge (u,v) in G\H, $\delta(u)+\delta(v) \geq B(1-\varepsilon^2)$

Main Theorem in Nonbipartite Graphs

- **Defn:** a subgraph **H** of **G** is an **EDCS(B)** if:
  - **1.** For each edge (u,v) in H, $\delta_H(u) + \delta_H(v) \leq B$
  - **2.** For each edge (u,v) in G\H, $\delta_H(u) + \delta_H(v) \geq B(1-\varepsilon^2)$
- **Main Thm:** if H is an EDCS(B), $\mu(H) \geq (2/3-\varepsilon)\mu(G)$
- **Idea:** explicitly construct a large **fractional matching** in H
  - Let val(u,v) be the fractional value on edge (u,v) in a fractional matching
  - val(u,v) will depend on $\delta_H(u)$ and $\delta_H(v)$
- Simplifies several proofs for bipartite case
- **Problem:** integrality gap in nonbipartite graphs

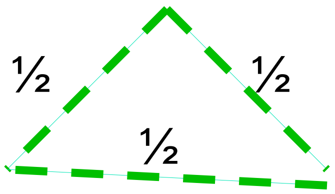
Fractional Matching

- **Dfn:** let $\mu_f(G)$ be the size of maximum fractional matching in G
- **Bipartite Graphs:** $\mu_f(G) = \mu(G)$
- **Nonbipartite Graphs:** $\mu(G) \geq (2/3)\mu_f(G)$
- **Can show:** if H is an EDCS(B), then

$$\mu_f(H) \geq (2/3 - \varepsilon)\mu(G)$$

- **Not good enough:**

$$\mu(H) \geq (2/3)\mu_f(H) \geq (4/9 - \varepsilon)\mu(G)$$



α-Restricted Fractional Matching

- **Dfn:** Let val(u,v) be the fractional value on edge (u,v) in a fractional matching
- A fractional matching is **α-restricted** ($\alpha<1$) if for every edge (u,v) either $val(u,v) = 1$ or $val(u,v) \le \alpha$
- **Dfn:** let $\mu^{\alpha}_{f}(G)$ be the maximum value of an α-restricted fractional matching in G
- **Classic Thm:**

$$\mu(G) \ge (2/3)\mu_f(G)$$

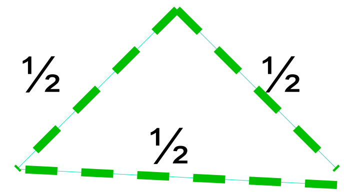
- **Our Thm:**

$$\mu(G) \ge (\alpha/(\alpha+1))\ \mu^{\alpha}_{f}(G)$$

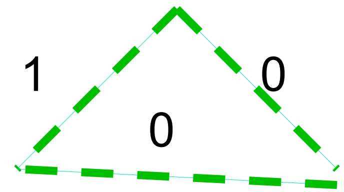
- **Corollary:**

$$\mu(G) \ge (1-\varepsilon)\ \mu^{\varepsilon}_{f}(G)$$

α=1/2



α=1/3



EDCS contains a large ε-restricted matching

- **Main Thm:** If a subgraph H is an EDCS(B):

$$\mu(H) \geq (2/3-\varepsilon)\ \mu(G)$$

- **Main Lemma:** If a subgraph H is an EDCS(B):

$$\mu^{\varepsilon}_{f}(H) \geq (2/3-\varepsilon)\ \mu(G)$$

- **Main Lemma → Main Theorem:**

$$\mu(H) \geq (1-\varepsilon)\ \mu^{\varepsilon}_{f}(H) \geq (2/3-\varepsilon)(1-\varepsilon)\mu(G) = (2/3-O(\varepsilon))\mu(G)$$

Explicit Construction

- Given some subgraph H that is an EDCS(B)
  - **1.** For each edge (u,v) in H, $\delta_H(u) + \delta_H(v) \leq B$
  - **2.** For each edge (u,v) in G\H, $\delta_H(u) + \delta_H(v) \geq B(1-\varepsilon^2)$
- **Want to show:** can construct an ε-restricted fractional matching M_H of H such that:

$$\mathrm{val}(M_H) \geq (2/3-\varepsilon)\mu(G)$$

- Very involved proof. For this presentation will make many simplifying assumptions.

Proof Sketch of main theorem

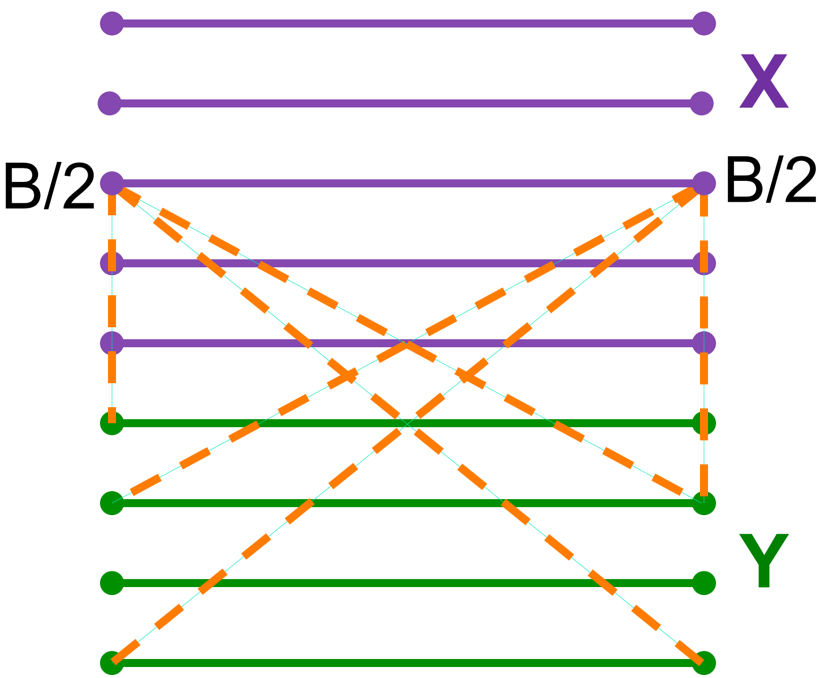
- **Simplified** EDCS constraints

  (u,v) in H: $\delta_H(u) + \delta_H(v) \le B$

  (u,v) in G\H: $\delta_H(u) + \delta_H(v) \ge B$

Maximum matching M_G

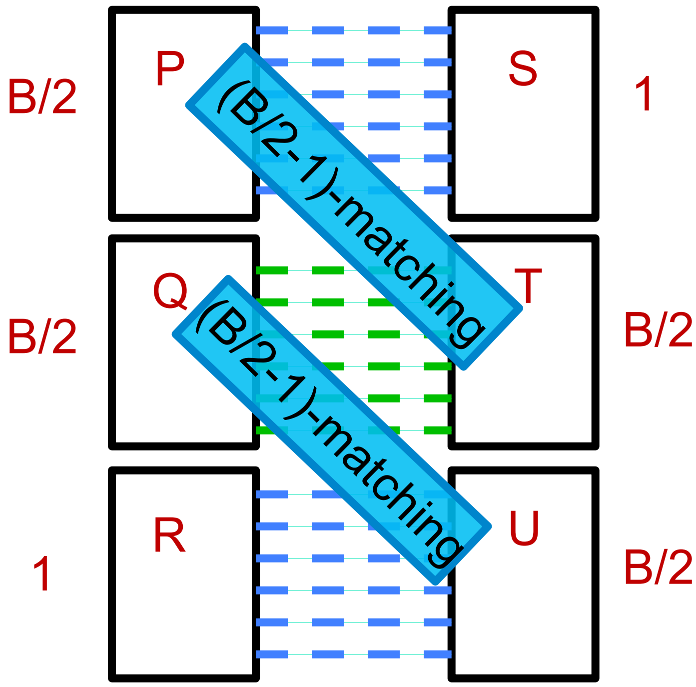
Note: $|X| + |Y| = 2\mu(G)$



- **Big Assumption:** if edge (u,v) in X-X: $\delta_H(u) = \delta_H(v) = B/2$
- **Medium Assumption:** All edges in H incident to Y
- **Fractional Matching**: for each edge (u,v) in X-Y, val(u,v) = 2/B
  - Legality: all relevant vertices have degree at most B/2
- **Total value:** $|X|(B/2)(2/B) = |X|$
- **Case 1:** $|X| \ge 2\mu(G)/3$
- **Case 2:** $|Y| \ge 4\mu(G)/3$. At least $2\mu(G)/3$ edges in Y-Y
- **Actual proof:** mix two cases. Probabilistic method.

EDCS is only a 3/2-approximation.

- Unweighted EDCS(B):
  - **1.** For each edge (u,v) in H, $\delta_H(u) + \delta_H(v) \leq B$
  - **2.** For each edge (u,v) in G\H, $\delta_H(u) + \delta_H(v) \geq B(1- \varepsilon^2)$

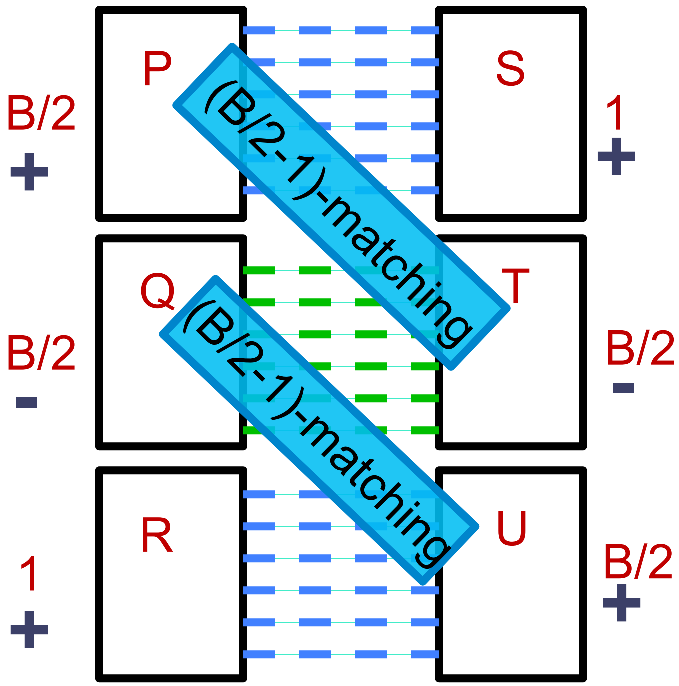


- Blue edges = In H
- Green edges = In G\H
- $\mu(H) = 2n/3 = 2/3\ \mu(G)$
- **Note:** edges between P-S and R-U have edge degree **<<** $B(1- \varepsilon^2)$
- H is a valid EDCS only because constraint 2 only applies to edges in G\H

Weighted EDCS

- Unweighted EDCS(B):
  - **1.** For each edge (u,v) in H, $\delta_H(u) + \delta_H(v) \leq B$
  - **2.** For each edge (u,v) in G\H, $\delta_H(u) + \delta_H(v) \geq B(1- \varepsilon^2)$
- **Definition:** G^B = G, but with B copies of each edge
- Weighted EDCS(B):
  - Equivalent to Unweighted EDCS(B) on G^B
  - Degree $\delta_H(u)$ now takes into account multiplicity.
  - Multiplicities can be ignored when computing the matching M in H because in M max degree = 2.
- Constraint 2 in a weighted EDCS:
  - Some copy of edge (u,v) is always in G\H
  - So: for each edge (u,v) in **G**, $\delta_H(u) + \delta_H(v) \geq B(1- \varepsilon^2)$

Unweighted VS. Weighted



- Blue edges = In H
- Green edges = In G\H
- Unweighted EDCS:
  - **1.** For edge (u,v) in H, $\delta(u)+\delta(v) \leq B$
  - **2.** For edge (u,v) in G\H, $\delta(u)+\delta(v) \geq B(1-\varepsilon^2)$
- Weighted EDCS:
  - Each edge has multiplicity B
  - **1.** For edge (u,v) in H, $\delta(u)+\delta(v) \leq B$ (same)
  - **2.** For edge (u,v) in G, $\delta(u)+\delta(v) \geq B(1-\varepsilon^2)$

- Unweighted EDCS: $\mu(H) = 2n/3$.
- Weighted EDCS: P-S and R-U edges violate Prop. 2.
  - Must add weight to P-S and R-U edges.
  - Must then remove weight from P-T and Q-U edges (Prop. 1)
  - Must now add weight to Q-T edges (Prop. 2).

Weighted EDCS

- Can show that in a weighted EDCS satisfies $\mu(H) \geq (1- \varepsilon)\ \mu(G)$
- Can only maintain a weighted EDCS efficiently in low arboricity bipartite graphs
- Intuitively, with a weighted EDCS, deleting 1 edge in G can deleted a large amount of weight in H.

Review

- Main Result: $O(m^{1/4})$ update time, $(3/2-\varepsilon)$-approx.
  - Fastest better than 2-approximation
  - Fastest deterministic algorithm for any constant approx.
- EDCS(B): subgraph H such that
  - **1.** For each edge (u,v) in H, $\delta_H(u) + \delta_H(v) \leq B$
  - **2.** For each edge (u,v) in $G \backslash H$, $\delta_H(u) + \delta_H(v) \geq B(1- \varepsilon^2)$
- **Main Theorem:** $\mu(H) \geq (3/2-\varepsilon)\ \mu(G)$
  - Techniques very different for bipartite and nonbipartite case
- Lots of dynamic graph and graph orientation details to achieve running time

Open Problems

- Can we maintain a $(1+\varepsilon)$ approximation in update time $O(m^{1/2-c})$ update time for any fixed $c > 0$?
- How well can we approximate matching in polylog update time
- Very interesting even on bipartite graphs, with randomization and amortization.
- Weighted EDCS may be helpful