Dynamic Matchings in Convex Bipartite Graphs

Gerth Stølting Brodal
University of Aarhus

Loukas Georgiadis
Hewlett-Packard Laboratories

Kristoffer Arnsfelt Hansen
University of Chicago

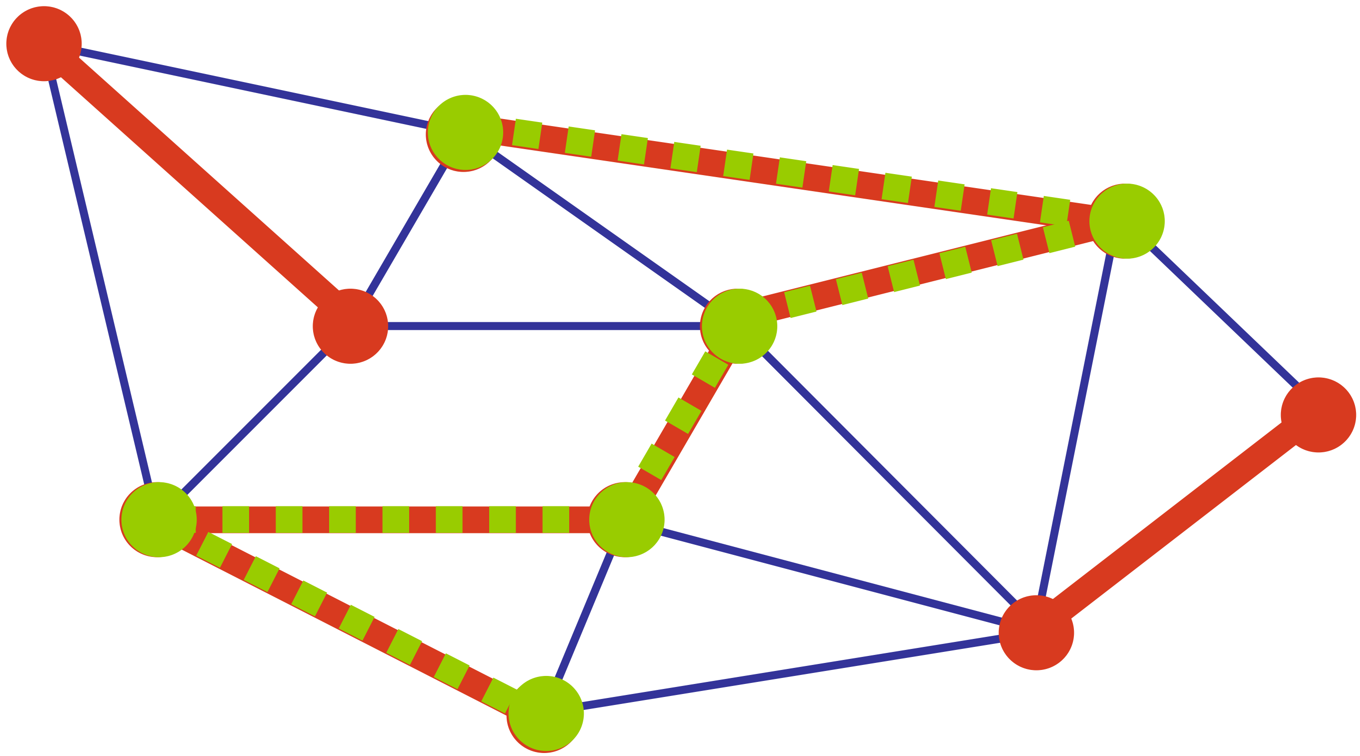
Irit Katriel
Brown University

MFCS 2007, Hotel Růže, Český Krumlov, Czech Republic, August 26-31, 2007

Outline of Talk

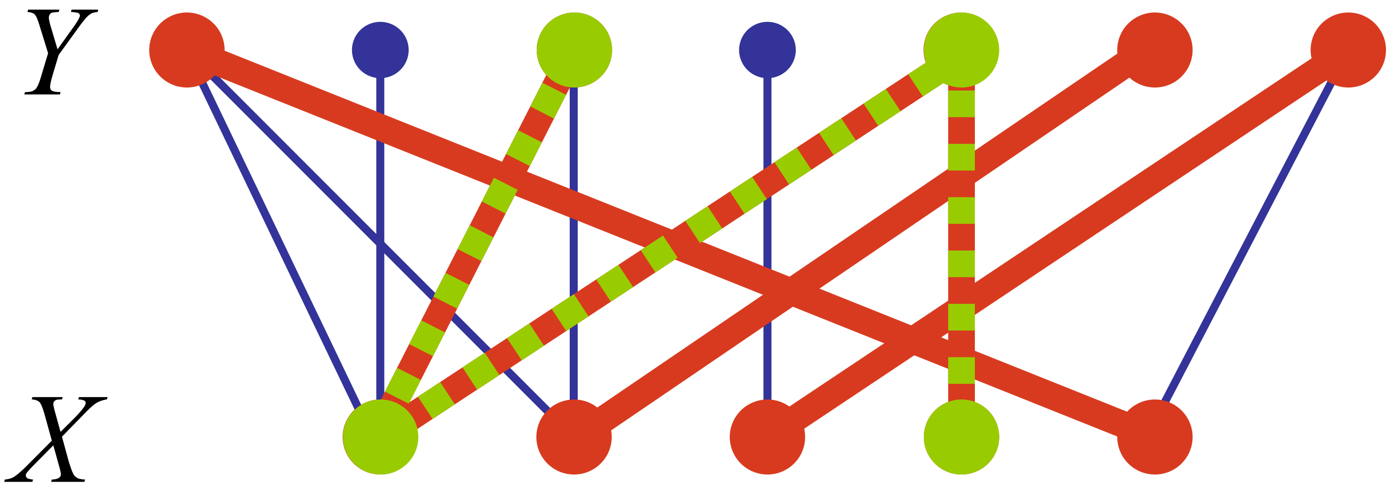
- Definitions: Graphs and matchings
- Definitions: Convex bipartite graphs
- Glovers algorithm for convex bipartite graphs
- Definitions: Matchings in dynamic convex bipartite graphs
- **Result**
- Ingredients of the solution
- Conclusion

Matchings in General Graphs



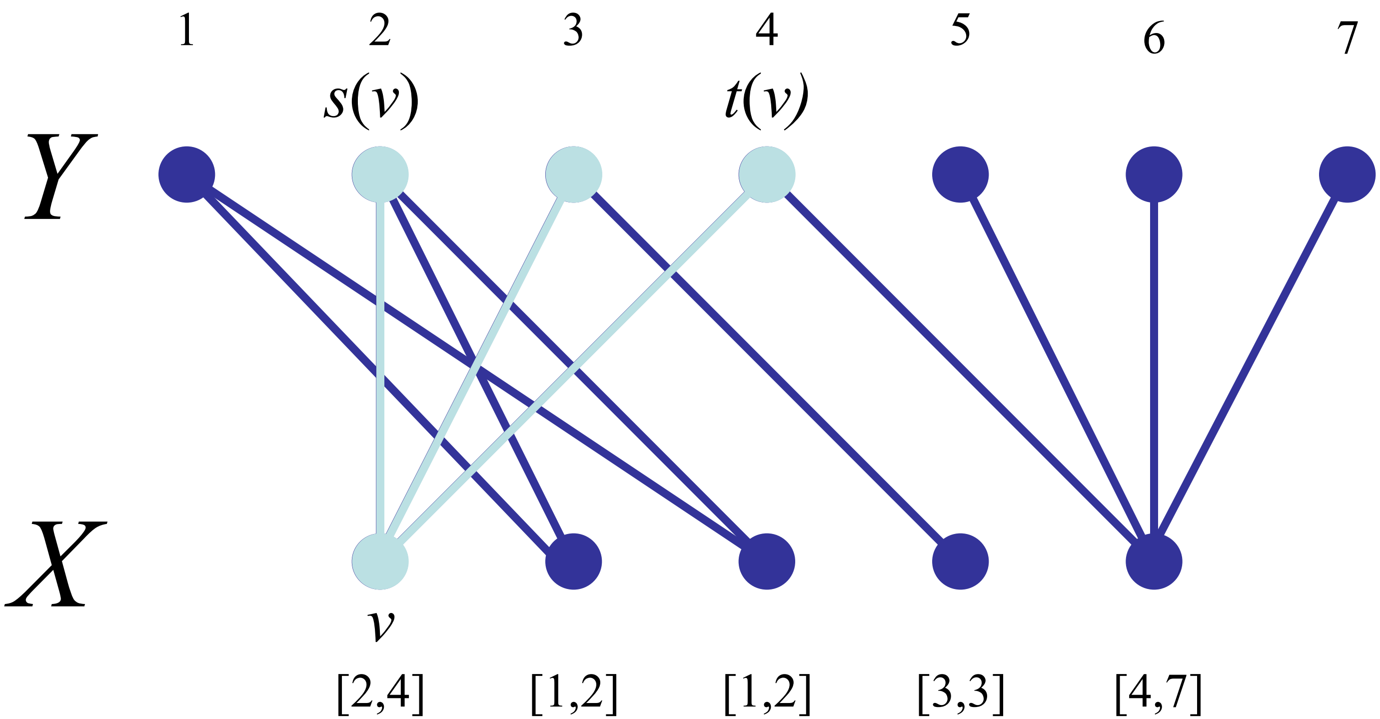
Deterministic $O(E\sqrt{V})$	Micali, Vaziani '80
Randomized $O(V^{2.476})$	Mucha, Sankowski '04

Matchings in Bipartite Graphs

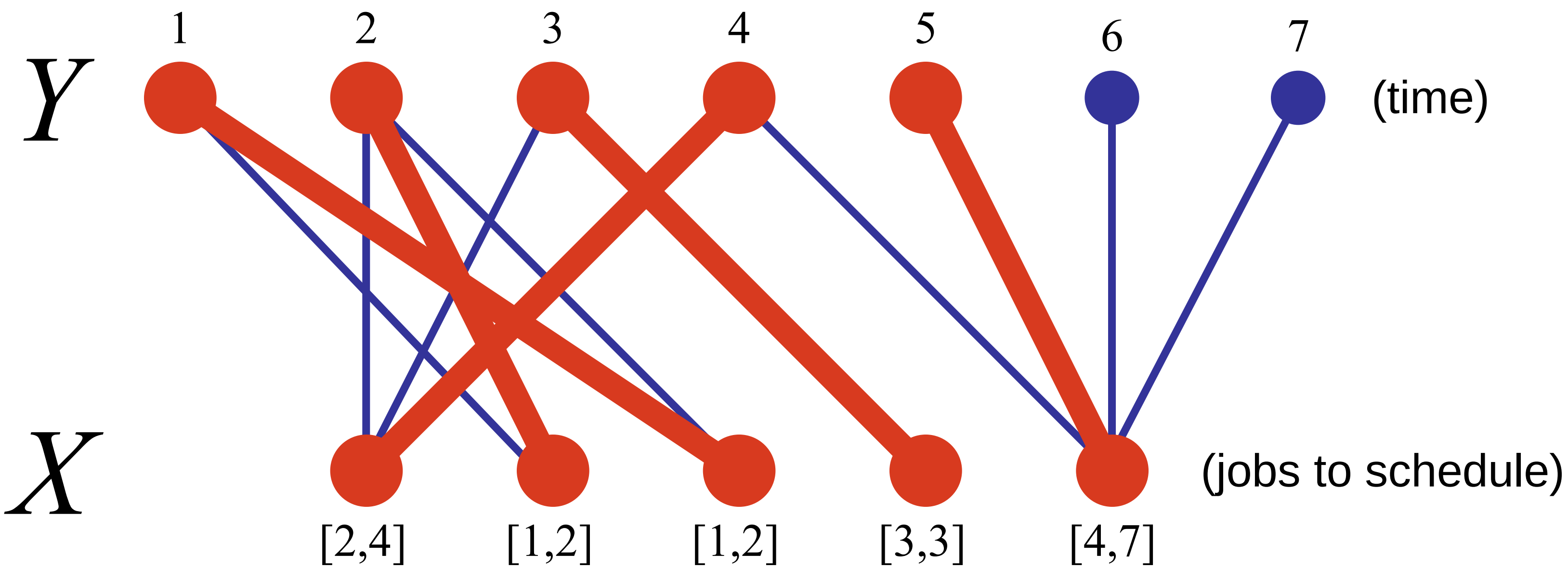


Deterministic $O(E\sqrt{V})$	Hopcroft, Karp ’73
Randomized $O(V^{2.476})$	Mucha, Sankowski ’04

Convex Bipartite Graphs

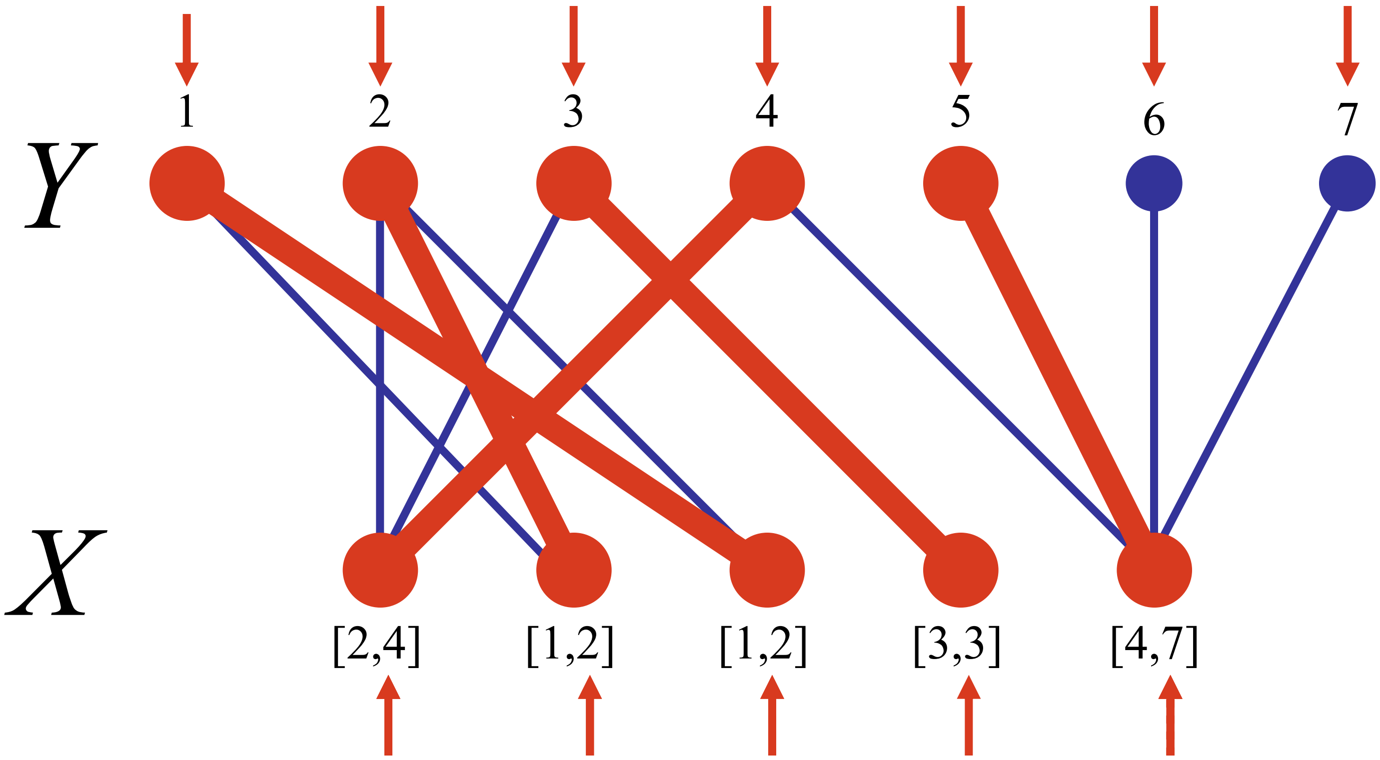


Matchings in Convex Bipartite Graphs



Deterministic $O(X+Y)$	Gabow, Tarjan '85
Deterministic $O(X)$	Steiner, Yeomans '96

Matchings in Convex Bipartite Graphs: Glover's Greedy Algorithm



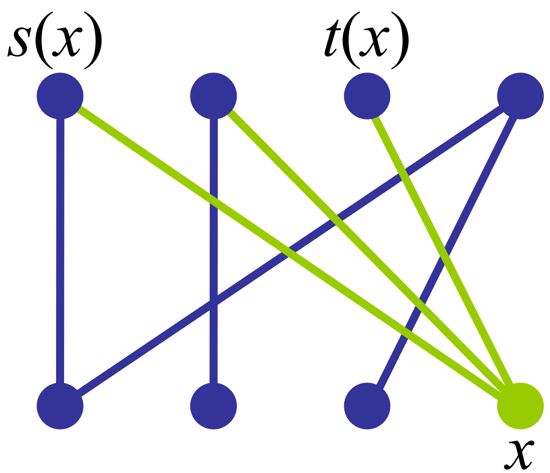
For the current time y maintain a priority queue of the $t(v)$ where $y \in [s(v),t(v)]$

Dynamic Matchings in Convex Bipartite Graphs

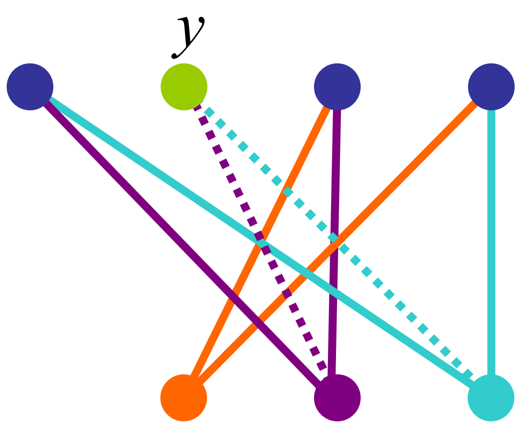
Observation

$O(1)$ edge changes can change $\Omega(X)$ edges in the matching but only $O(1)$ nodes in the matching need to change

Dynamic Convex Bipartite Graphs: Updates



Insert(x,[$s(x)$,$t(x)$])
Delete(x)

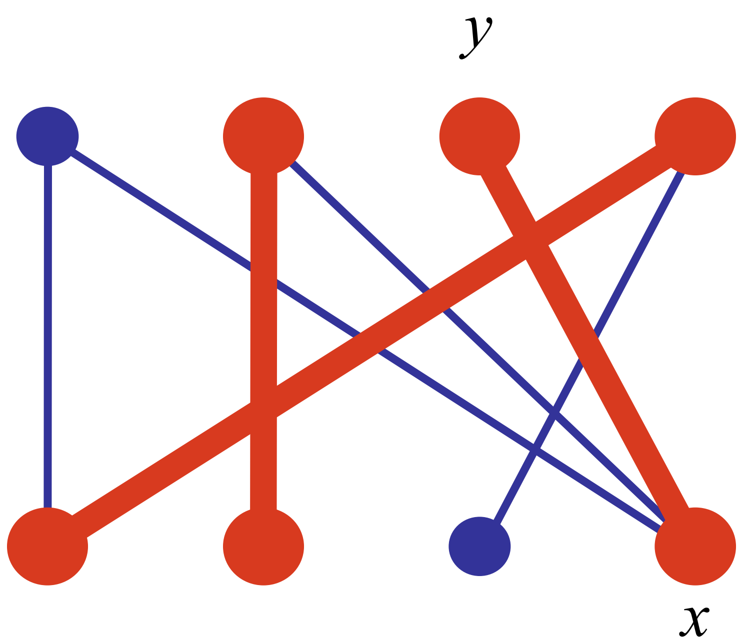


Insert(y)
Delete(y)

Note: Cannot delete/insert a y that equals $s(x)$ or $t(x)$ for some x

Observation: Updates preserve convexity

Dynamic Convex Bipartite Graphs: Queries



Matched?(x) Matched?(y)
Mate(x) Mate(y)

Result

Updates	$O(\log^2 X)$ amortized
Matched?	$O(1)$ worst-case
Mate	$O(\sqrt{X} \cdot \log^2 X)$ amortized
	$O(\min\{k \cdot \log^2 X, X \cdot \log X\})$ worst-case
Space	$O(Y + X \cdot \log X)$

k = # updates since the last Mate query for the same node

Related Work

Perfect matchings in general graphs can be maintained in time $O(V^{1.495})$ per edge update

Sankowski '04

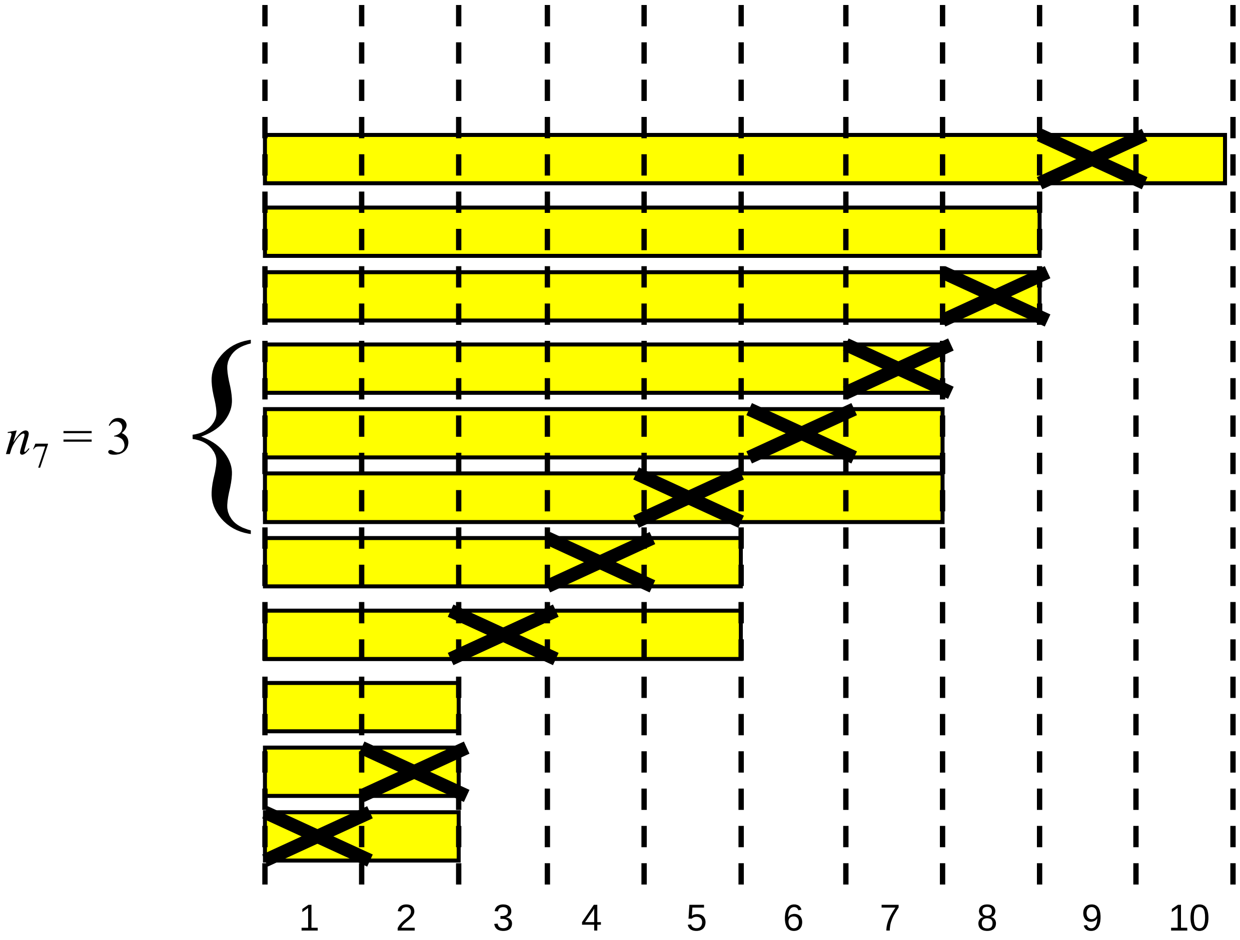
Size of maximum matchings in general graphs can be maintained in time $O(V^{1.495})$ per edge update

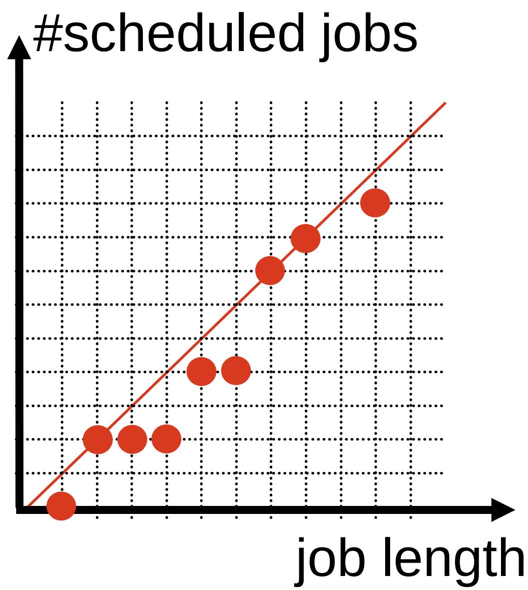
Sankowski '07

Ingredients of Our Solution

- Special data structure for the case $s(x)=1$ for all x
- Dynamic version of Dekel-Sahni's parallel algorithm (divide-and-conquer)
- Mate queries are handled by lazy construction of the matching

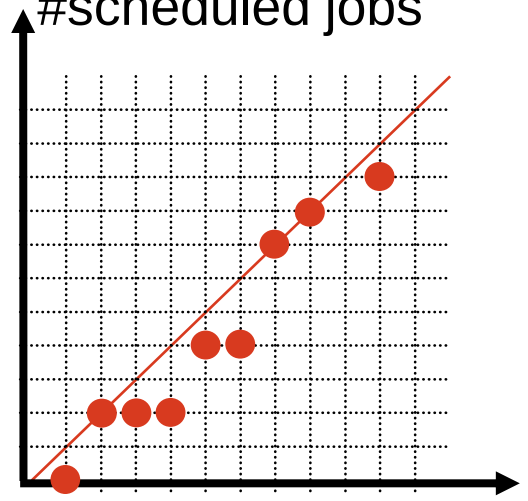
Case: $s(x)=1$ for all x
(Jobscheduling view)



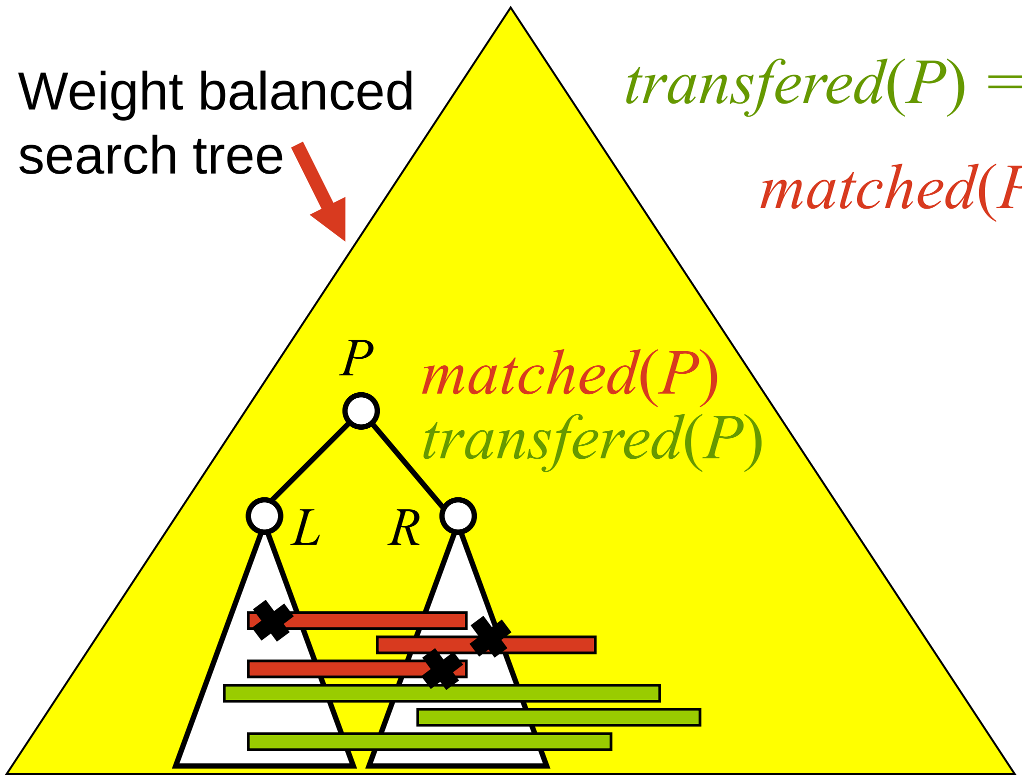


Case: $s(x)=1$ for all x
(Jobscheduling view)

- Store vertical distance from each • to diagonal
- Insert job length i :
  - Find first $j \geq i$ with • on diagonal
  - Decrement distance by one for $i..j\text{-}1$
- $O(\log n)$ time using augmented binary search tree



Dynamizing Dekel-Sahni



$y_1\ y_2\ y_3\ y_4$ .....

$transfered(P) = transfered(R) \cup transfered'(R)$

$matched(P) = matched(L) \cup matched'(R)$

$transfered'(R)$ and $matched'(R)$ are computed from $transfered(L)$ and $matched(R)$ assuming all starting times equal $\min(R)$

Updates: $O(1)$ changes at each level, i.e. total $O(\log^2 n)$ time

Conclusion

Dynamic Matchings in Convex Bipartite Graphs

Updates	$O(\log^2 X)$ amortized	
Matched?	$O(1)$ worst-case	
Mate	$O(\sqrt{X} \cdot \log^2 X)$ amortized	
	$O(\min\{k \cdot \log^2 X, X \cdot \log X\})$ worst-case	
Space	$O(Y + X \cdot \log X)$	

Open problems...