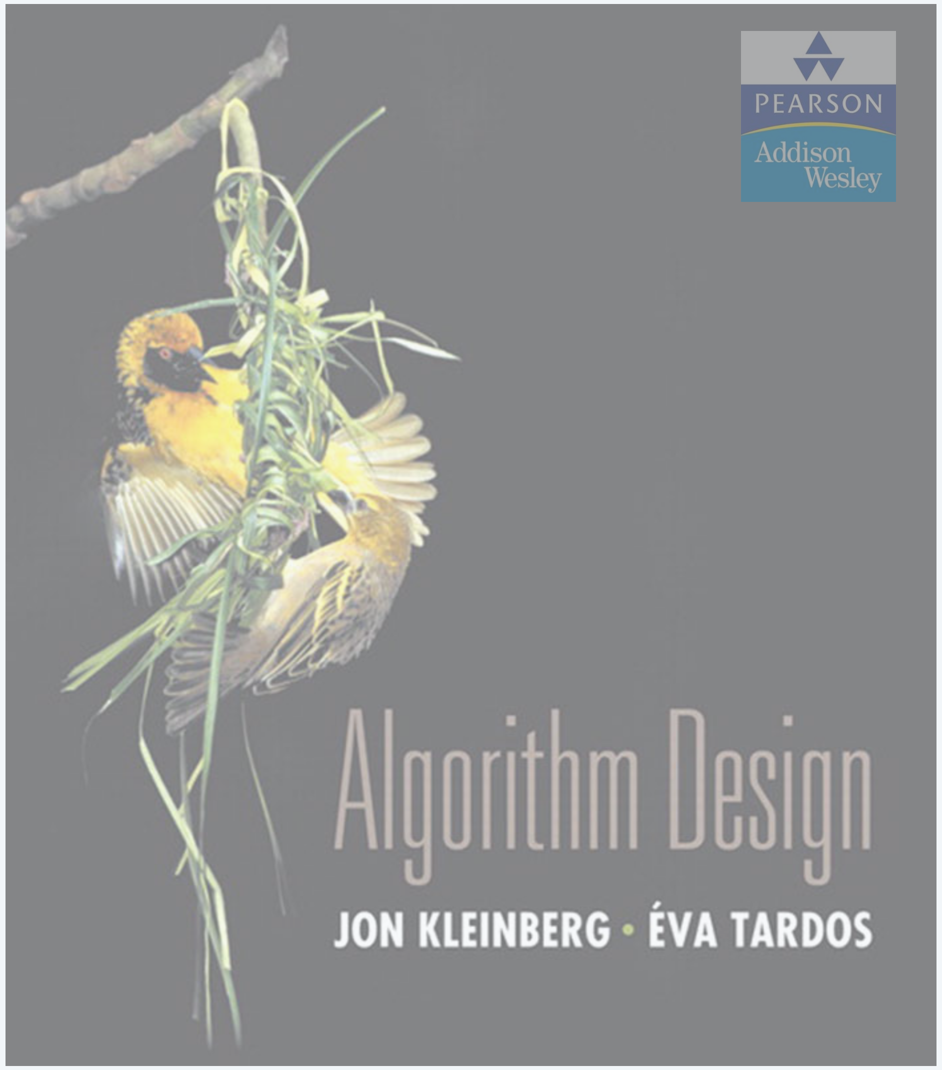


SECTION 11.7

11. APPROXIMATION ALGORITHMS

- *load balancing*
- *center selection*
- *pricing method: weighted vertex cover*
- *LP rounding: weighted vertex cover*
- *generalized load balancing*
- *knapsack problem*

Generalized load balancing

Input. Set of m machines M; set of n jobs J.

- Job $j \in J$ must run contiguously on an authorized machine in $M_j \subseteq M$.
- Job $j \in J$ has processing time t_j.
- Each machine can process at most one job at a time.

Def. Let J_i be the subset of jobs assigned to machine i.
The load of machine i is $L_i = \Sigma_{j \in J_i} t_j$.

Def. The makespan is the maximum load on any machine $= \max_i L_i$.

Generalized load balancing. Assign each job to an authorized machine to minimize makespan.

Generalized load balancing: integer linear program and relaxation

ILP formulation. x_{ij} = time machine i spends processing job j.

$$
\begin{array}{llllll}
(IP) & \min & L & & & \\
& \text{s. t.} & \sum_i x_{ij} & = & t_j & \text{for all } j \in J \\
& & \sum_j x_{ij} & \leq & L & \text{for all } i \in M \\
& & x_{ij} & \in & \{0,\ t_j\} & \text{for all } j \in J \text{ and } i \in M_j \\
& & x_{ij} & = & 0 & \text{for all } j \in J \text{ and } i \notin M_j
\end{array}
$$

LP relaxation.

$$
\begin{array}{llllll}
(LP) & \min & L & & & \\
& \text{s. t.} & \sum_i x_{ij} & = & t_j & \text{for all } j \in J \\
& & \sum_j x_{ij} & \leq & L & \text{for all } i \in M \\
& & x_{ij} & \geq & 0 & \text{for all } j \in J \text{ and } i \in M_j \\
& & x_{ij} & = & 0 & \text{for all } j \in J \text{ and } i \notin M_j
\end{array}
$$

Generalized load balancing: lower bounds

Lemma 1. The optimal makespan $L^* \geq \max_j t_j$.

Pf. Some machine must process the most time-consuming job. ▪

Lemma 2. Let L be optimal value to the LP. Then, optimal makespan $L^* \geq L$.

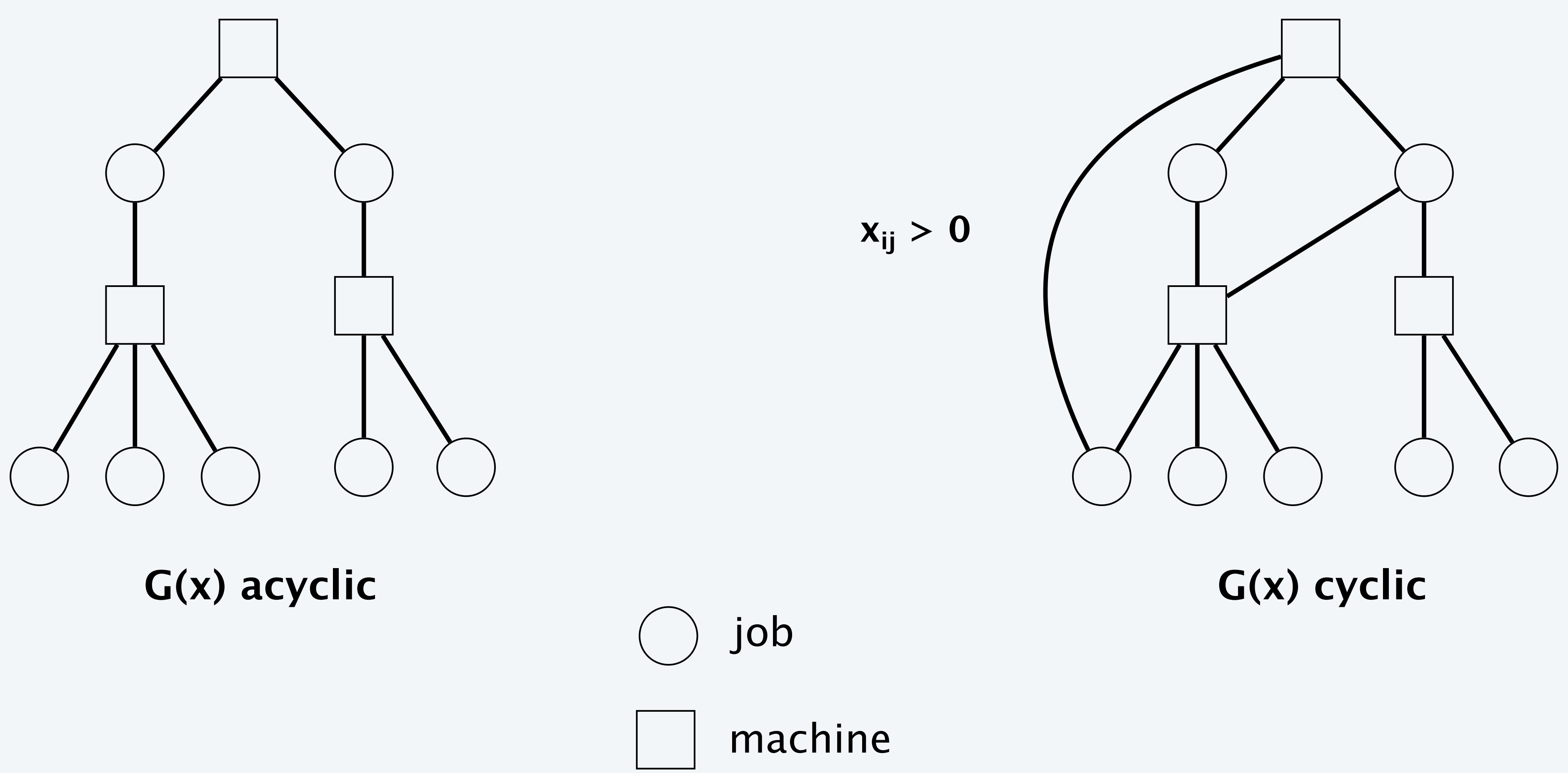
Pf. LP has fewer constraints than ILP formulation. ▪

Generalized load balancing: structure of LP solution

Lemma 3. Let x be solution to LP. Let $G(x)$ be the graph with an edge between machine i and job j if $x_{ij} > 0$. Then $G(x)$ is acyclic.

can transform x into another LP solution where G(x) is acyclic if LP solver doesn't return such an x

Pf. (deferred)



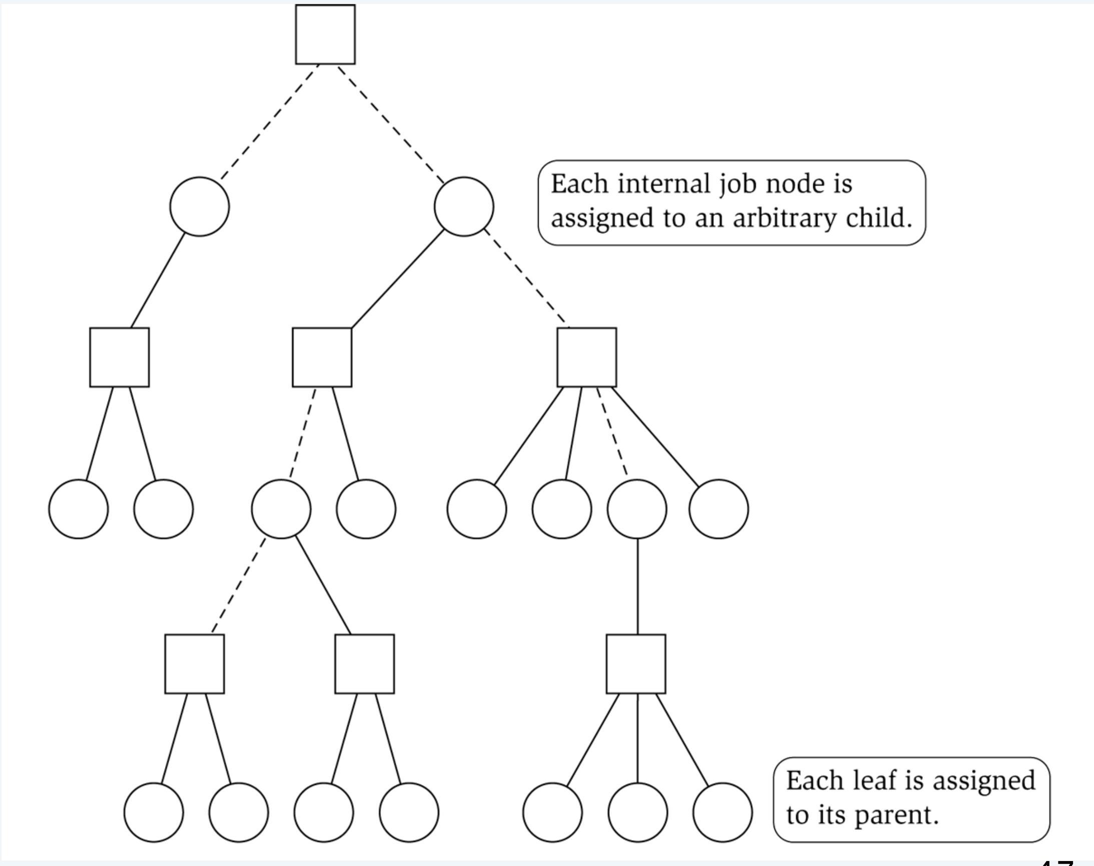
Generalized load balancing: rounding

Rounded solution. Find LP solution x where $G(x)$ is a forest. Root forest $G(x)$ at some arbitrary machine node r.

- If job j is a leaf node, assign j to its parent machine i.
- If job j is not a leaf node, assign j to any one of its children.

Lemma 4. Rounded solution only assigns jobs to authorized machines.

Pf. If job j is assigned to machine i, then $x_{ij} > 0$. LP solution can only assign positive value to authorized machines. ▪



Generalized load balancing: analysis

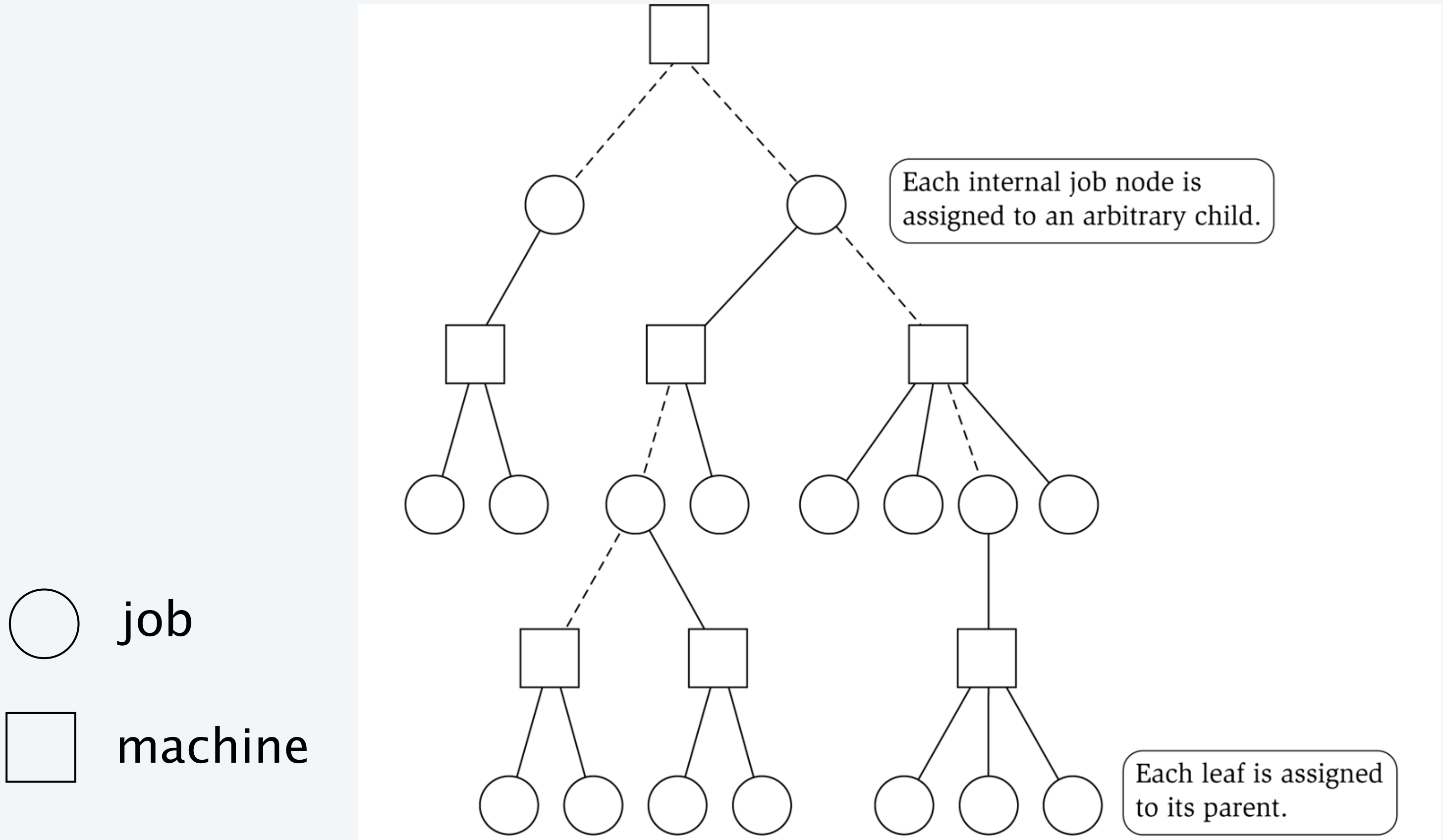
Lemma 5. If job j is a leaf node and machine $i = parent(j)$, then $x_{ij} = t_j$.

Pf.

- Since i is a leaf, $x_{ij} = 0$ for all $j \neq parent(i)$.
- LP constraint guarantees $\Sigma_i \, x_{ij} = t_j$. ▪

Lemma 6. At most one non-leaf job is assigned to a machine.

Pf. The only possible non-leaf job assigned to machine i is $parent(i)$. ▪



Generalized load balancing: analysis

Theorem. Rounded solution is a 2-approximation.

Pf.

- Let $J(i)$ be the jobs assigned to machine i.
- By LEMMA 6, the load L_i on machine i has two components:
  - leaf nodes:

$$\sum_{\substack{j \in J(i) \\ j \text{ is a leaf}}} t_j \overset{\text{Lemma 5}}{=} \sum_{\substack{j \in J(i) \\ j \text{ is a leaf}}} x_{ij} \le \sum_{j \in J} x_{ij} \overset{\text{LP}}{\le} \underbrace{L}_{\text{optimal value of LP}} \overset{\text{Lemma 2 (LP is a relaxation)}}{\le} L^*$$

  - parent:

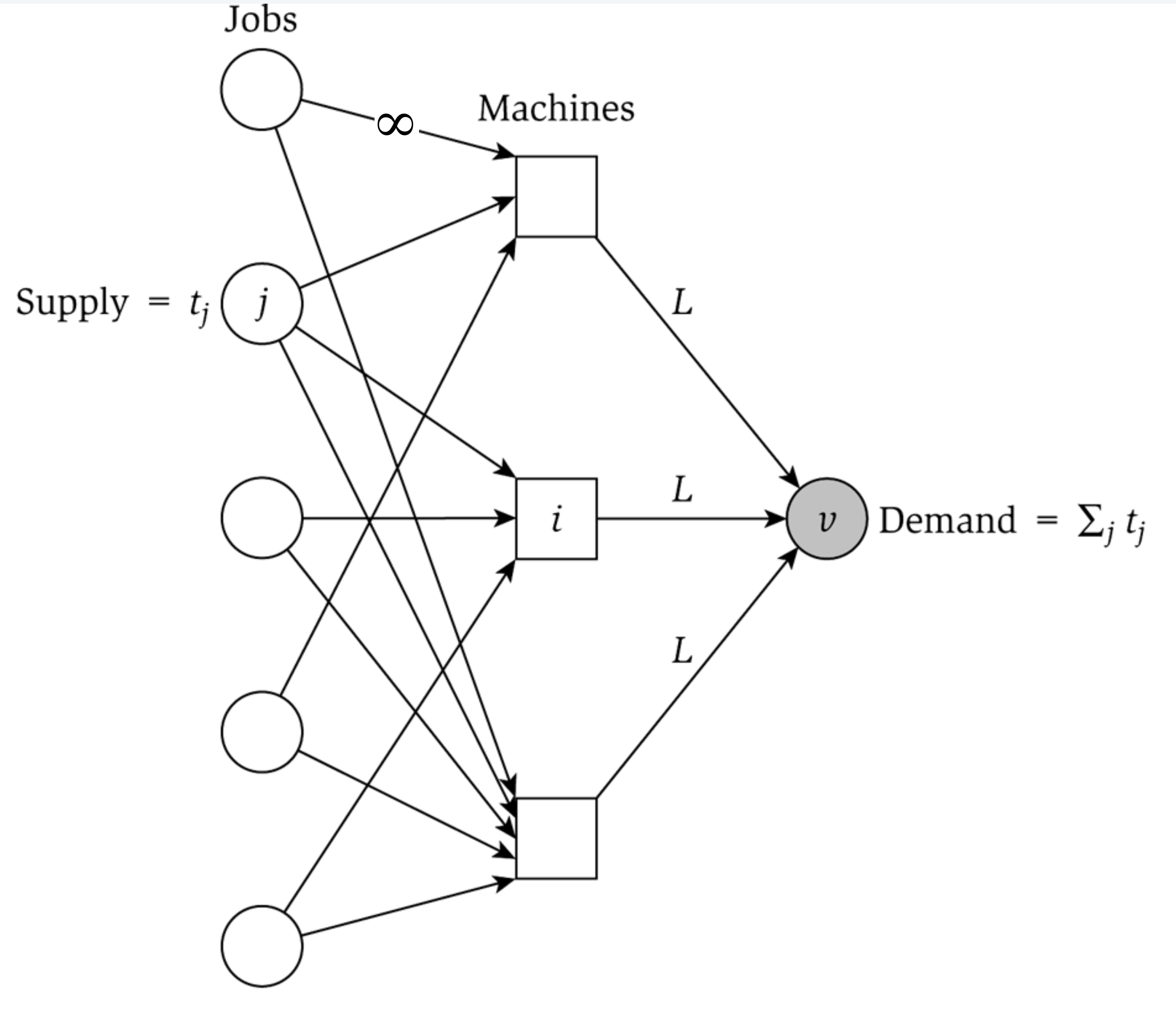
$$t_{\text{parent}(i)} \overset{\text{Lemma 1}}{\le} L^*$$

- Thus, the overall load $L_i \le 2L^*$. ▪

Generalized load balancing: flow formulation

Flow formulation of LP.

$$\begin{array}{llll}
\sum_i x_{ij} & = & t_j & \text{for all } j \in J \\
\sum_j x_{ij} & \leq & L & \text{for all } i \in M \\
x_{ij} & \geq & 0 & \text{for all } j \in J \text{ and } i \in M_j \\
x_{ij} & = & 0 & \text{for all } j \in J \text{ and } i \notin M_j
\end{array}$$



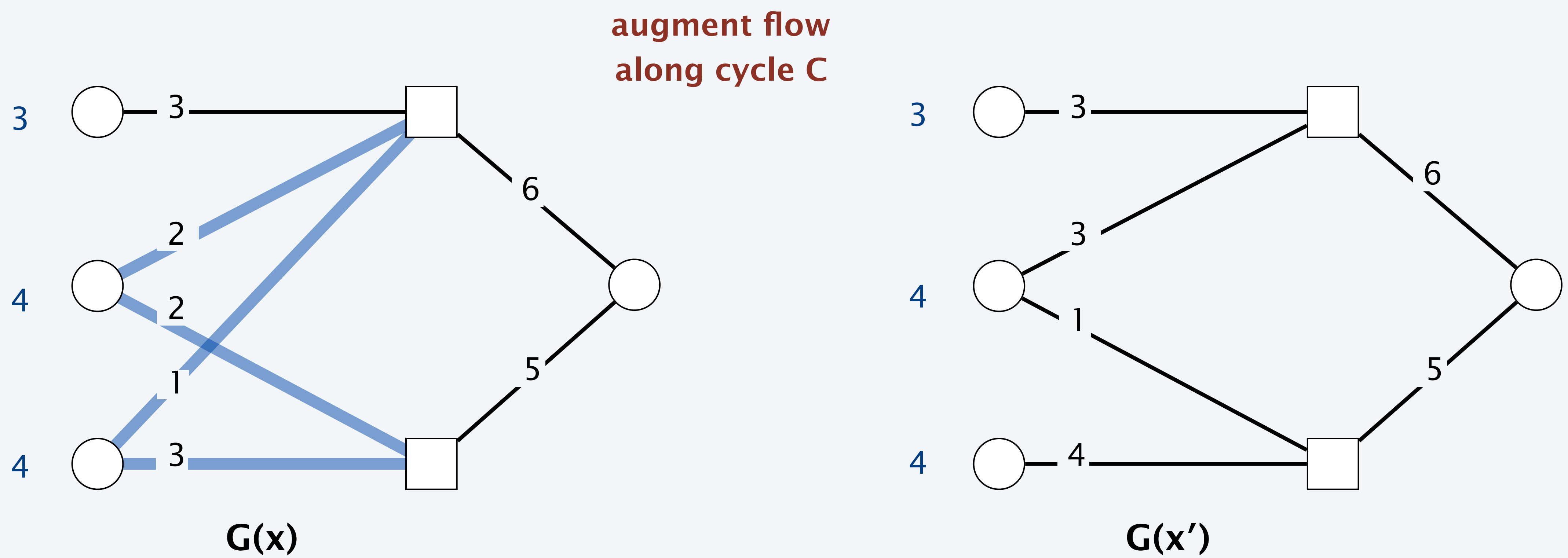
Observation. Solution to feasible flow problem with value L are in 1-to-1 correspondence with LP solutions of value L.

Generalized load balancing: structure of solution

Lemma 3. Let (x, L) be solution to LP. Let $G(x)$ be the graph with an edge from machine i to job j if $x_{ij} > 0$. We can find another solution (x', L) such that $G(x')$ is acyclic.

Pf. Let C be a cycle in $G(x)$.

- Augment flow along the cycle C. ← flow conservation maintained
- At least one edge from C is removed (and none are added).
- Repeat until $G(x')$ is acyclic. ▪



Conclusions

Running time. The bottleneck operation in our 2-approximation is solving one LP with $mn+1$ variables.

Remark. Can solve LP using flow techniques on a graph with $m+n+1$ nodes: given L, find feasible flow if it exists. Binary search to find L^*.

Extensions: unrelated parallel machines. [Lenstra–Shmoys–Tardos 1990]

- Job j takes t_{ij} time if processed on machine i.
- 2-approximation algorithm via LP rounding.
- If **P** ≠ **NP**, then no no ρ-approximation exists for any $\rho < 3/2$.

Mathematical Programming 46 (1990) 259–271 259
North-Holland

APPROXIMATION ALGORITHMS FOR SCHEDULING UNRELATED PARALLEL MACHINES

Jan Karel LENSTRA
Eindhoven University of Technology, Eindhoven, The Netherlands, and Centre for Mathematics and Computer Science, Amsterdam, The Netherlands

David B. SHMOYS and Éva TARDOS
Cornell University, Ithaca, NY, USA