Universidade Federal de Alfenas

Algoritmos em Grafos

Aula 17 – Fluxo Máximo: Emparelhamento Bipartido Máximo

Prof. Humberto César Brandão de Oliveira
humberto@bcc.unifal-mg.edu.br

Relembrando...

- Aula passada:
  - Método de Ford-Fulkerson

FORD-FULKERSON(G, s, t)

  inicializar fluxo *f* com 0

  enquanto existir um caminho aumentante *p* em *G*

    ampliar fluxo *f* ao longo do caminho *p*

  fim enquanto

  retornar fluxo máximo e o vetor *f*

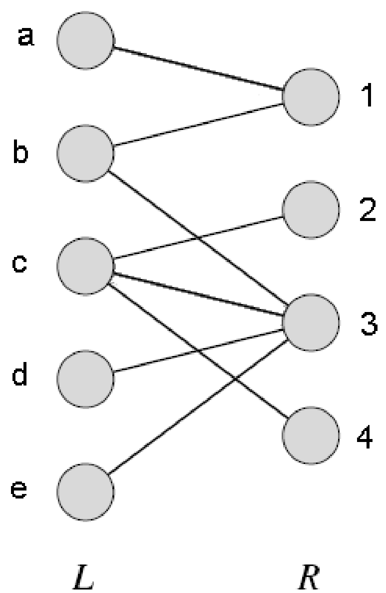
fim.

Conceito:
Grafo Bipartido

- Em um grafo bipartido *G=(V,A)*, o conjunto de vértices *V* é dividido em dois subconjuntos:
  - *V1* e *V2*;

- Nenhum vértice *V1* é adjacente de outro vértice do próprio conjunto *V1*;

- O mesmo vale para o conjunto *V2*.



Emparelhamento Bipartido Máximo

- Imagine que você possui:
  - L máquinas;
  - R tarefas;

- E deseja processar a máxima quantidade de tarefas... Mas cada tarefa não pode ser processada por todas as máquinas. Cada tarefa pode ser processada por um subconjunto de L.

Emparelhamento Bipartido Máximo

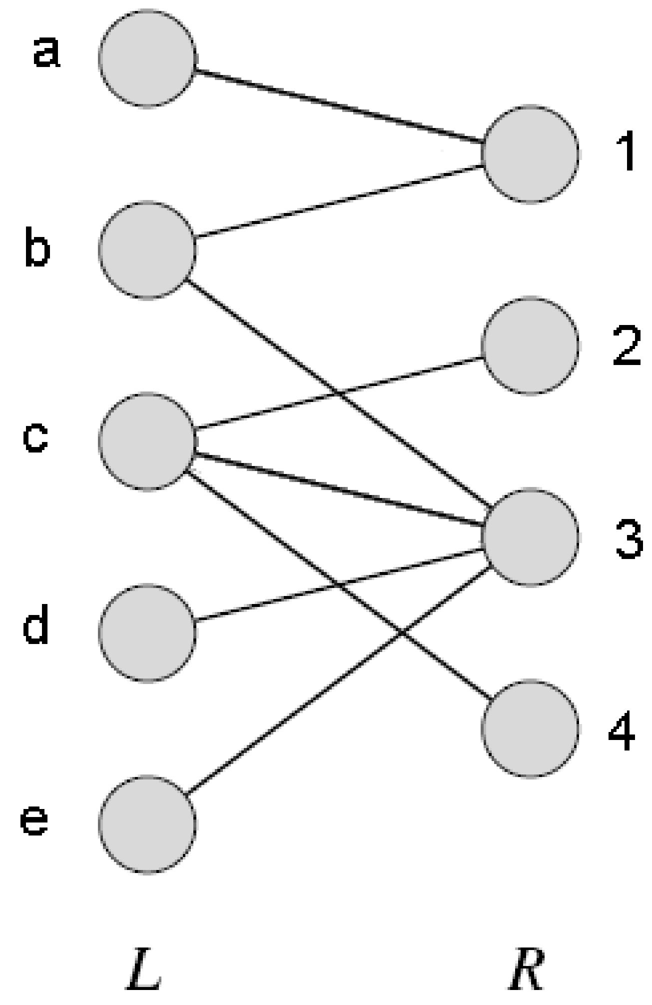
- L máquinas;
- R tarefas;

Possível alocação das máquinas:

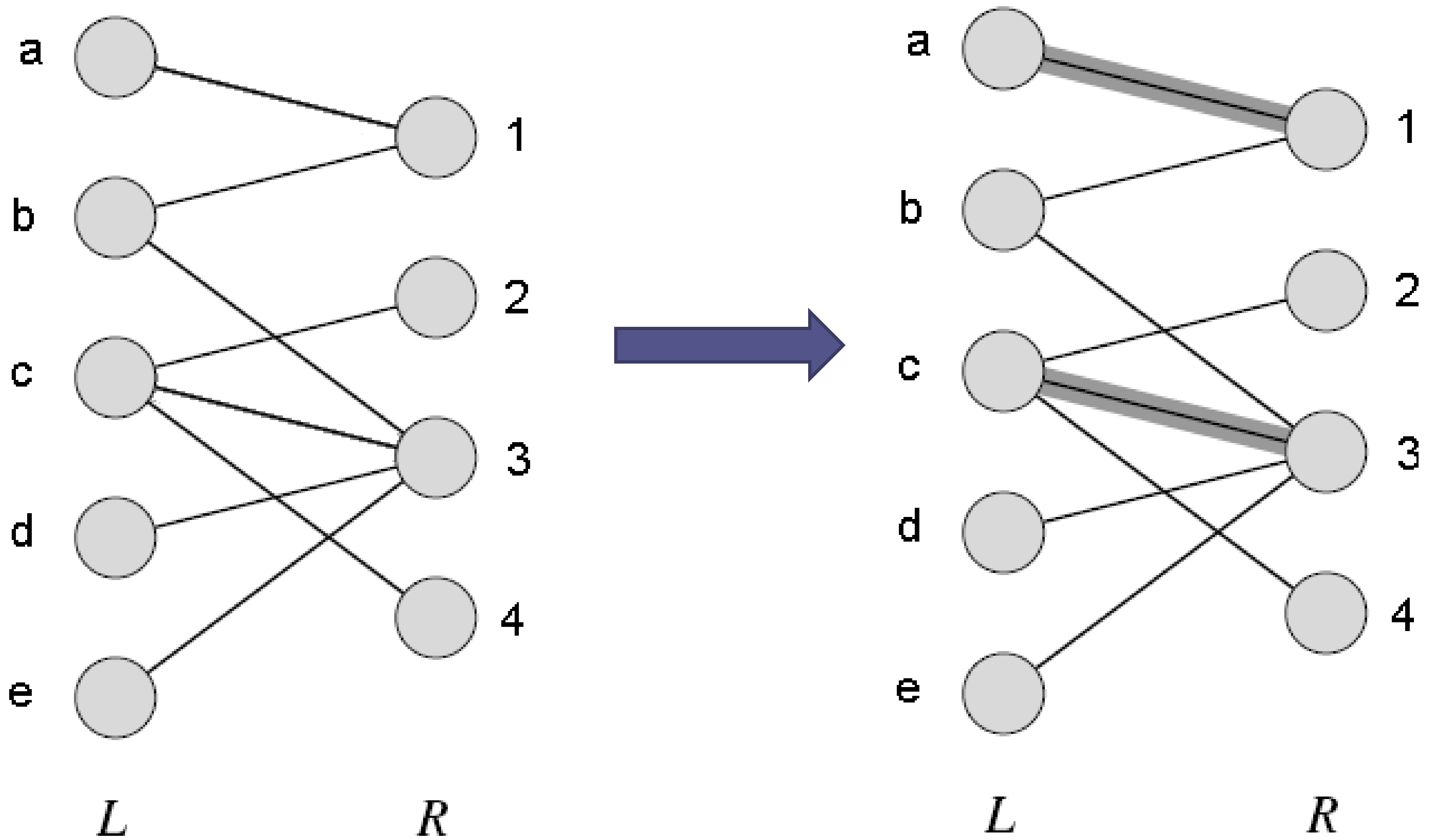
c → 3

a → 1

É possível alocar outra máquina respeitando as restrições?



Emparelhamento Bipartido Máximo

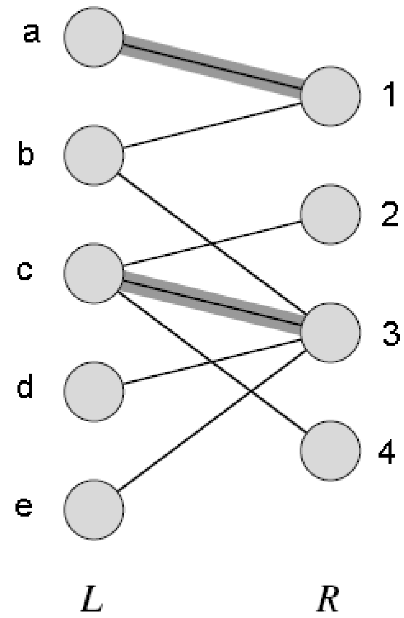


Emparelhamento Bipartido Máximo

- Formalizando o Emparelhamento:
  - Dado um grafo não orientado $G=(V,A)$, um emparelhamento é um subconjunto de arestas $M \subseteq A$ tais que, para todos os vértices $v \in A$, no máximo uma aresta de M é incidente sobre v.
    - Ou seja: Uma tarefa não pode ser atendida por duas máquinas e uma máquina não pode atender duas tarefas.

Emparelhamento Bipartido Máximo

- Conceito:
  - Dizemos que um vértice $v \in V$ é correspondido pelo emparelhamento M, se alguma aresta de M é incidente sobre v.



Emparelhamento Bipartido Máximo

- **Formalizando o Emparelhamento máximo:**
  - Um **emparelhamento máximo é um emparelhamento de cardinalidade máxima**, ou seja, um emparelhamento M tal que, para qualquer emparelhamento M´, temos:

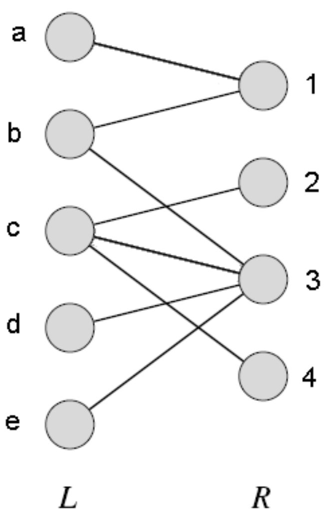
$$|M| \geq |M'|$$

Emparelhamento Bipartido Máximo

- A princípio vamos focar o emparelhamento em grafos bipartidos.
  - Supomos que o conjunto de vértices pode ser particionado:
  - Onde L e R são disjuntos:

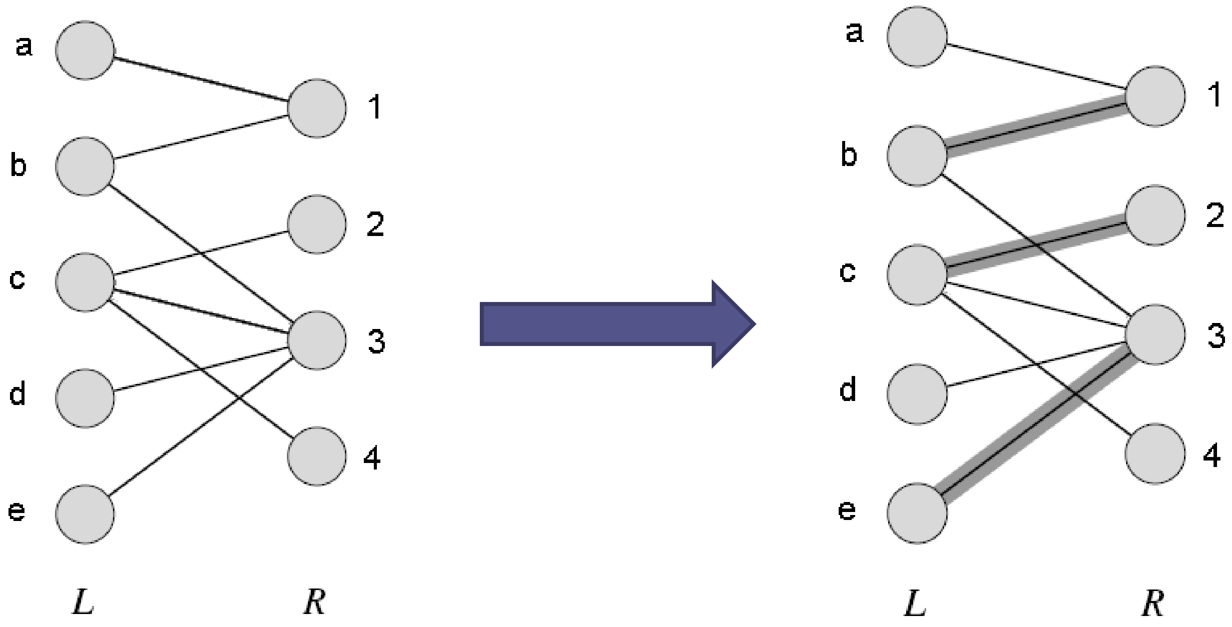
$$V = L \cup R \qquad\qquad L \cap R = \{\}$$

  - Além disso, todas as arestas de A passam entre L e R.



Emparelhamento Bipartido Máximo

- O problema é:
  - Como encontrar um emparelhamento bipartido máximo?



Emparelhamento Bipartido Máximo

- Podemos usar o algoritmo de Ford-Fulkerson para encontrar um emparelhamento máximo em um grafo bipartido não orientado.

- Devemos transformar um problema no outro:
  - O problema do emparelhamento máximo no problema de fluxo máximo;

Emparelhamento Bipartido Máximo

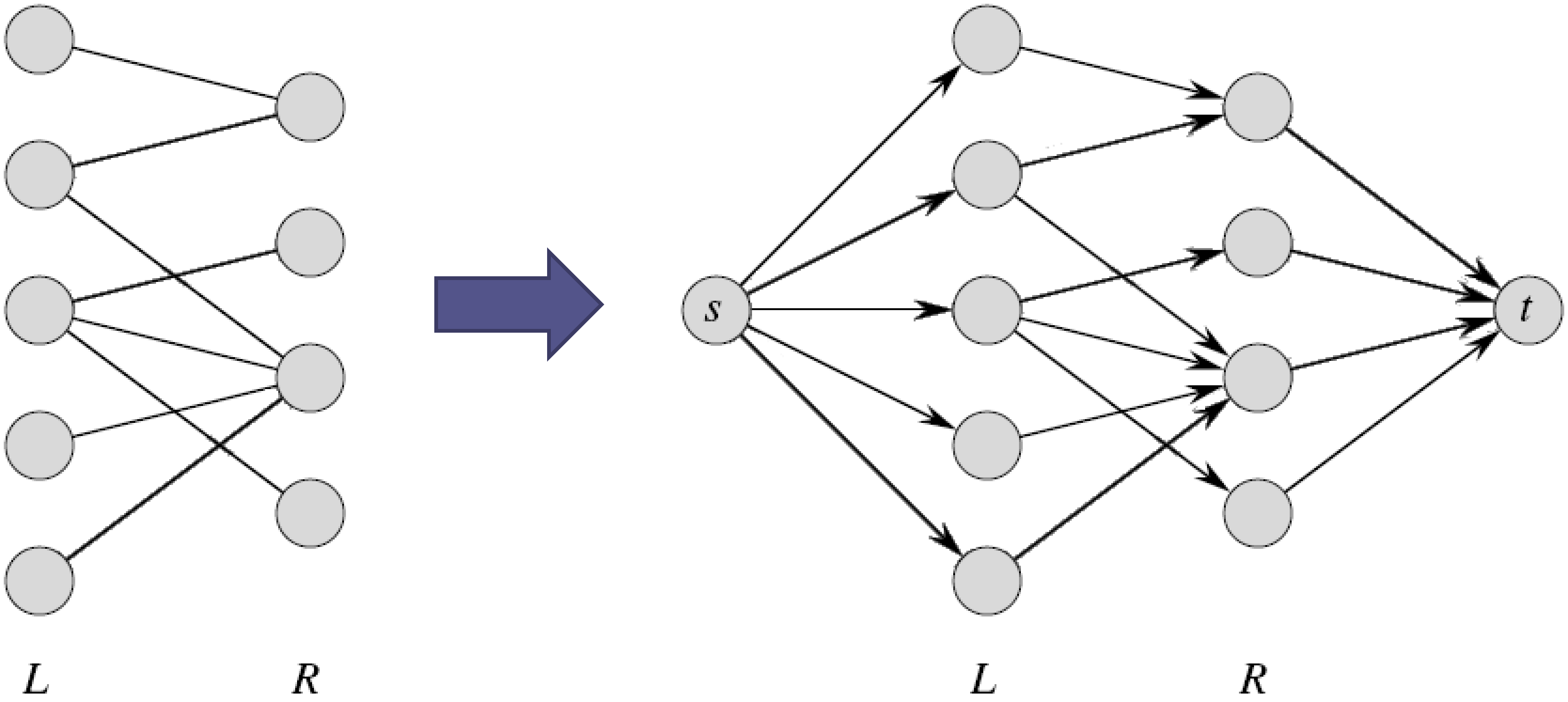
- A partir de $G=(V,A)$, onde V é dividido em duas partições disjuntas (L e R), definimos um fluxo em rede $G'=(V',A')$

$$V' = V \cup \{s,t\}$$

$$\begin{aligned} A' = &\{(s,u) : u \in L\} \\ &\cup \{(u,v) : u \in L, v \in R, (u,v) \in A\} \\ &\cup \{(v,t) : v \in R\} \end{aligned}$$

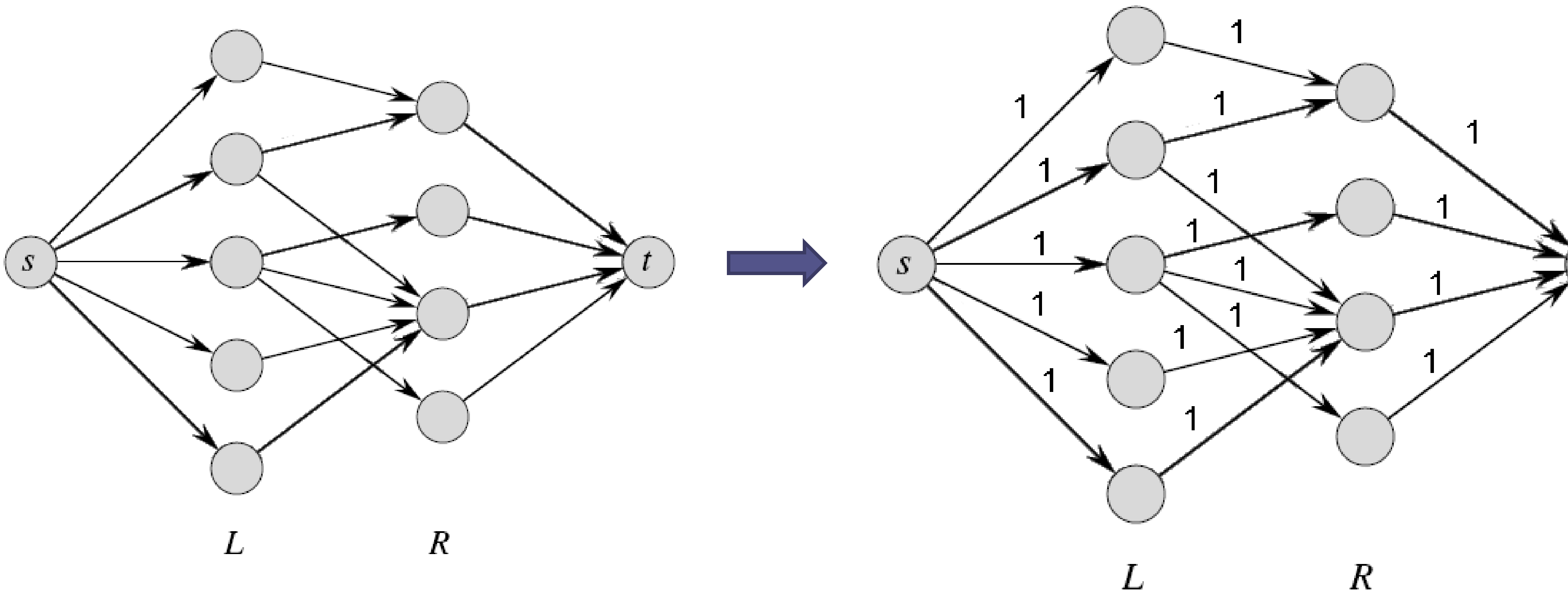
Emparelhamento Bipartido Máximo

$$V' = V \cup \{s,t\} \quad A' = \{(s,u) : u \in L\}$$
$$\cup \{(u,v) : u \in L, v \in R, (u,v) \in A\}$$
$$\cup \{(v,t) : v \in R\}$$



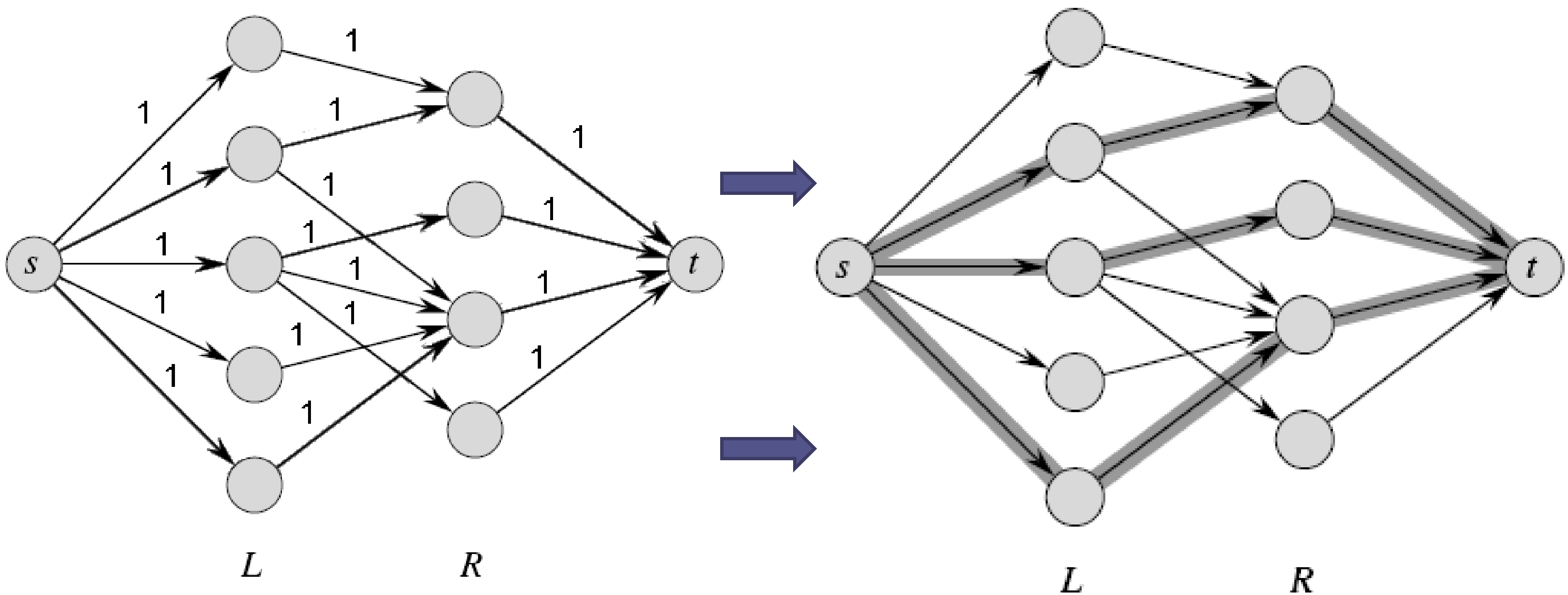
Emparelhamento Bipartido Máximo

- Para finalizar, consideramos que cada aresta possui capacidade igual a 1.



Emparelhamento Bipartido Máximo

- Agora é só aplicar o algoritmo de fluxo máximo para descobrir o emparelhamento bipartido máximo.



Emparelhamento Bipartido Máximo

- Detalhe importante:
  - Se a função de capacidade c utiliza apenas valores inteiros, então o fluxo máximo f produzido pelo método de Ford-Fulkerson apresenta a propriedade de que $|f|$ é inteiro.
  - Esta condição é necessária no emparelhamento:
    - Uma máquina não pode atender meia tarefa.

Bibliografia

- CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; (2002). Algoritmos – Teoria e Prática. Tradução da 2ª edição americana. Rio de Janeiro. Editora Campus.
  - 26.3 – Emparelhamento Bipartido Máximo

- ZIVIANI, N. (2007). Projeto e Algoritmos com implementações em Java e C++. São Paulo. Editora Thomson;