Universidade Federal de Mato Grosso do Sul
Centro de Ciências Exatas e Tecnologia
Departamento de Computação e Estatística

“Tópicos em Algoritmos Paralelos e Distribuídos”

Circuitos de Euler em Grafos

Claudia Nasu
cnasu@dct.ufms.br

Apresentação

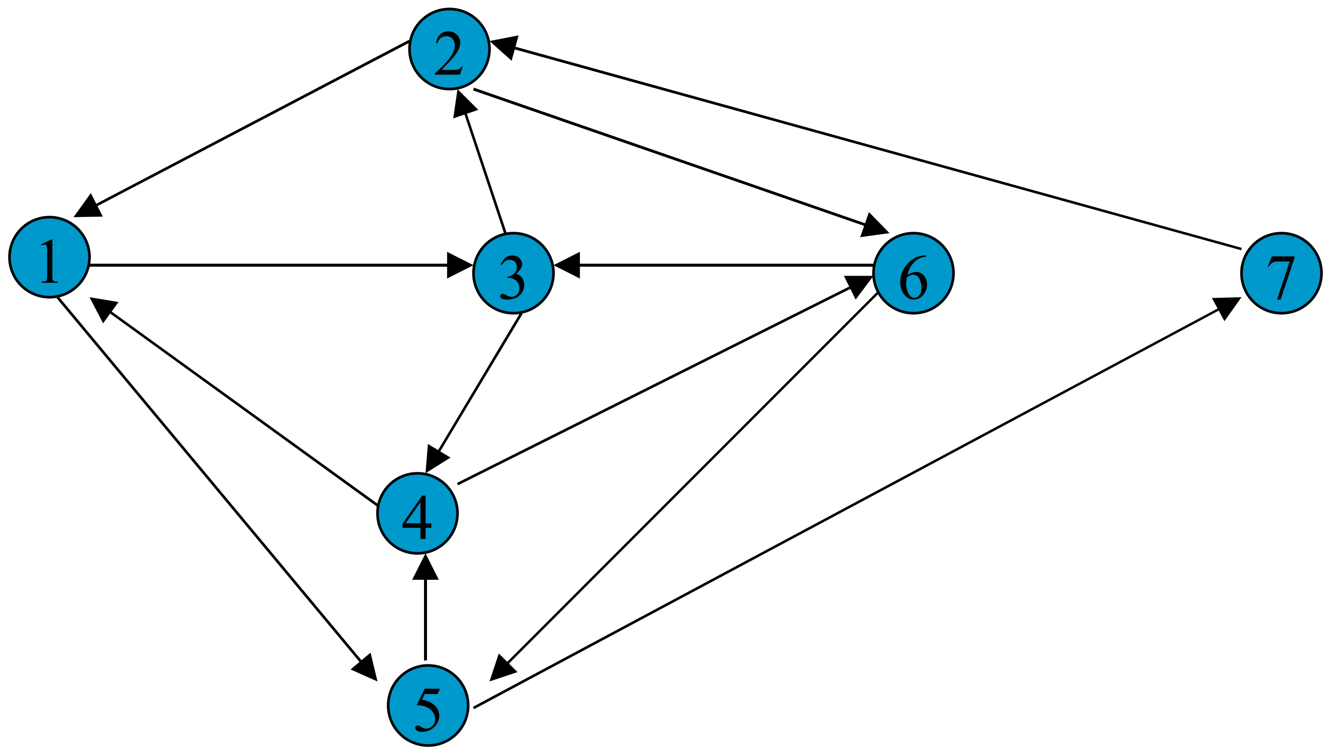
- Introdução
- Algoritmo
- Um exemplo

Introdução

- Um circuito Euleriano é um ciclo que passa por cada aresta do grafo exatamente uma vez.
- Teorema: um grafo conexo dirigido contém um circuito Euleriano, se e somente se, para cada vértice *v*, *indegree*(*v*) = *outdegree*(*v*).

Introdução (cont.)

Exemplo:



Algoritmo

- Atallah e Vishkin
- Modelo: PRAM CREW
- Complexidade:
  - Tempo: $O(log^2 n)$
  - Processadores: $O(m)$
- Pré-requisitos:
  - Euler-tour em árvores
  - Árvore geradora

Algoritmo

- Entrada:
  - Grafo dirigido $G = (V,E)$, onde $V = \{1,2,\ldots, n\}$, Euleriano.
  - Lista das arestas do grafo, armazenada no vetor EDGE de dimensão $m = |E|$
- Saída:
  - um circuito Euleriano de G

Algoritmo

Passo 1:

- Ordenação dos elementos de EDGE:
  - Dadas duas arestas (*i*, *j*) e (*k*, *l*), então, (*i*, *j*) < (*k*, *l*) se *j* < *l* ou (*j* = *l* e *i* < *k*).

(2,1)	(4,1)	(3,2)	(7,2)	(1,3)	(6,3)	(3,4)	(5,4)	(1,5)	(6,5)	(2,6)	(4,6)	(5,7)

$j = l$

$j < l$

$i < k$

Algoritmo

Passo 1 (cont.):

- Ordenação dos elementos de SUCESSOR:
  - Dadas duas arestas (i, j) e (k, l), então, $(i, j) < (k, l)$ se $i < k$ ou ($i = k$ e $j < l$).

(1,3)	**(1,5)**	**(2,1)**	**(2,6)**	**(3,2)**	**(3,4)**	**(4,1)**	**(4,6)**	**(5,4)**	**(5,7)**	**(6,3)**	**(6,5)**	**(7,2)**

$i = k$

$j < l$

$i < k$

- Como a ordenação é lexicográfica, o número de arestas da primeira ordenação chegando ao vértice *v* é igual ao número de arestas saindo deste vértice na segunda ordenação.
- Com isto, a *i*-ésima aresta *(u,v)* da primeira ordenação terá como correspondente a *i*-ésima aresta da segunda ordenação *(v,w)*.

Algoritmo

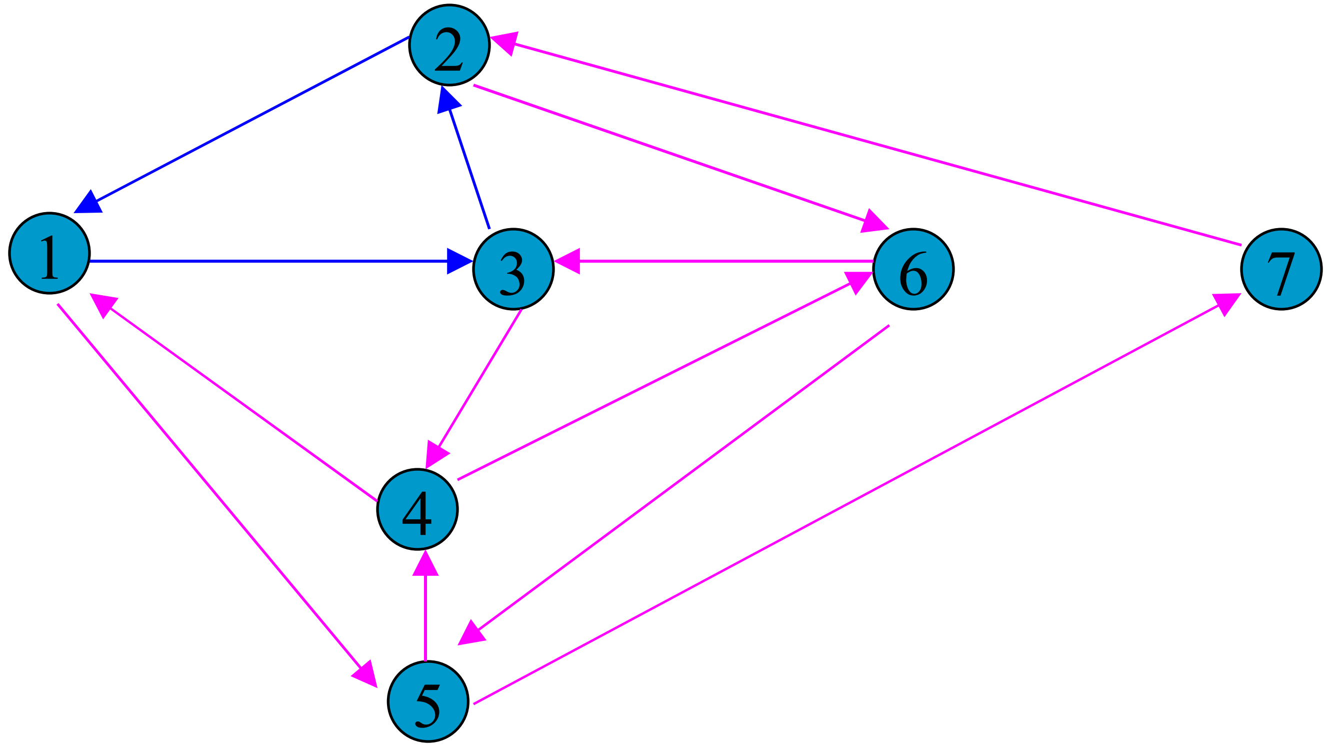
Passo 1 (cont.):

- Os vetores EDGE e SUCESSOR definem juntos um conjunto de ciclos (arestas disjuntas). Em qualquer ciclo, a aresta seguinte à aresta armazenada em EDGE(*i*) está em SUCESSOR(*i*).

- P(*i*) aponta para SUCESSOR(*j*), onde *j* é o endereço em EDGE da aresta armazenada em SUCESSOR(*i*).

(2,1)	(4,1)	(3,2)	(7,2)	(1,3)	(6,3)	(3,4)	(5,4)	(1,5)	(6,5)	(2,6)	(4,6)	(5,7)

(1,3)	(1,5)	(2,1)	(2,6)	(3,2)	(3,4)	(4,1)	(4,6)	(5,4)	(5,7)	(6,3)	(6,5)	(7,2)



Algoritmo

Passo 2:

- CYCLEREP(*i*) armazena a aresta representante do ciclo ao qual EDGE(i) pertence.

for all i, $1 \leq i \leq m$ **in parallel do** CYCLEREP(*i*) $\leftarrow$ SUCESSOR(i)
  repeat log m **times**
    for $1 \leq i \leq m$ **in parallel do**
      CYCLEREP(*i*) $\leftarrow$ min{CYCLEREP(*i*), CYCLEREP(P(*i*))}
      P(i) $\leftarrow$ P(P(i))

(1,3)	**(1,5)**	**(1,3)**	**(1,5)**	**(1,3)**	**(1,5)**	**(1,5)**	**(1,5)**	**(1,5)**	**(1,5)**	**(1,5)**	**(1,5)**	**(1,5)**

Algoritmo

Passo 2 (cont.):

- Grafo bipartido G' (V', E'):
  - $V' = V \cup C$, onde C denota o conjunto de arestas representando os ciclos.
  - $E' = \{(u,v) \mid u \in V,\ v \in C$ e u está no circuito representado por $v\}$ }

for all i, $1 \leq i \leq m$ **in parallel do**

    for EDGE(i) = (u,v) **do**

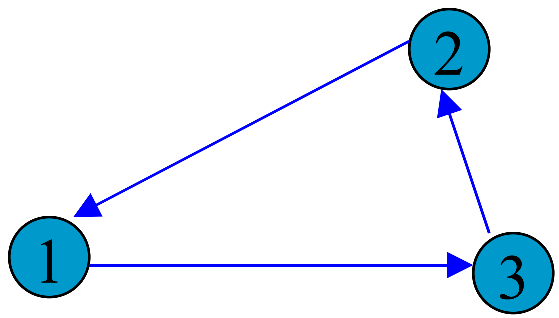
        EDGE' (2i - 1) $\leftarrow$ (u, CYCLEREP(i))

        EDGE'(2i) $\leftarrow$ (v,CYCLEREP(i))

Algoritmo

Passo 2 (cont.):

(2,1)	(4,1)	(3,2)	(7,2)	(1,3)	(6,3)	(3,4)	(5,4)	(1,5)	(6,5)	(6,6)	(4,6)	(5,7)



(2,(1,3)) **(1,(1,3))** **(4,(1,5))** **(1,(1,5))** **(3,(1,3))** **(2,(1,3))** **(7,(1,5))** **(2,(1,5))**

(1,(1,3)) **(3,(1,5))** **...**

Algoritmo

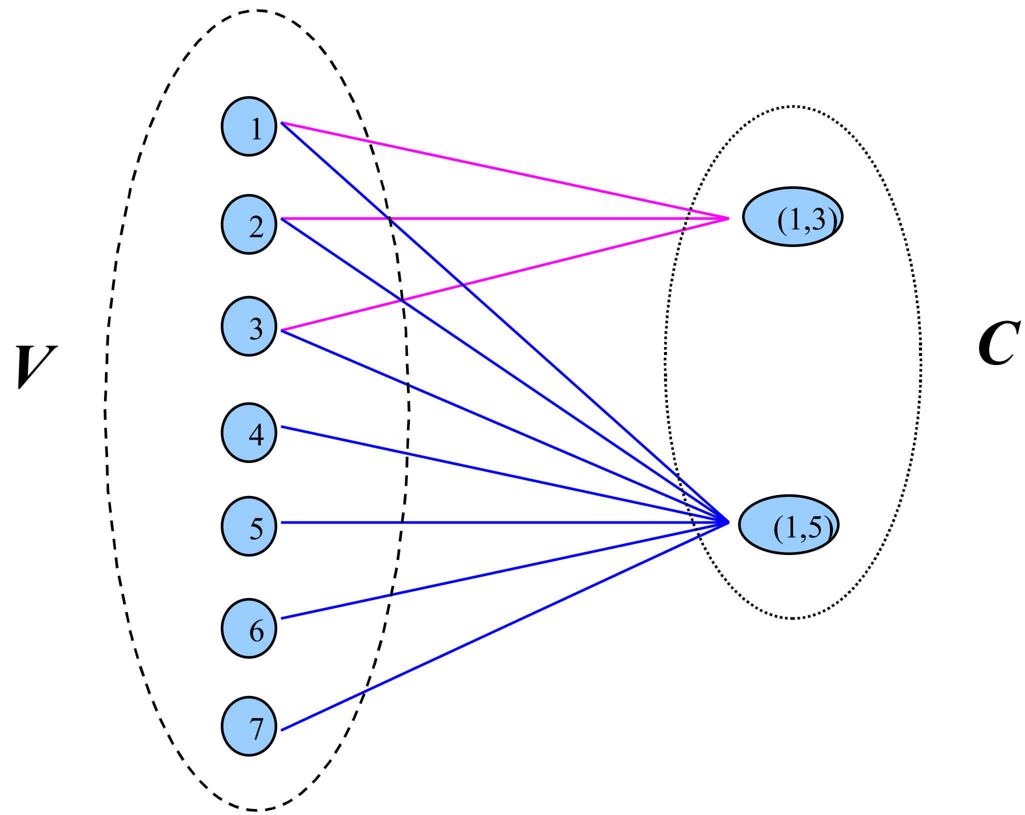
Passo 2 (cont.):

- Problema: as arestas aparecem pelo menos duas vezes no vetor, pois duas arestas consecutivas em um determinado ciclo tem um vértice em comum.
- Solução: ordenar as arestas e eliminar as cópias.

Algoritmo

Passo 2 (cont.):

- Grafo bipartite *G*':



Algoritmo

Passo 3:

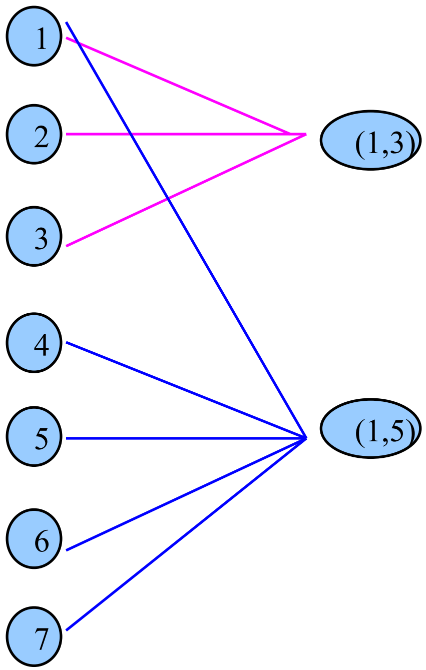
- Determina uma árvore geradora *T* de *G*'.

- *T*' é o grafo obtido de *T* substituindo-se cada aresta (*i*,*j*) por duas arestas direcionadas e anti-paralelas (*i*,*j*) e (*j*,*i*).

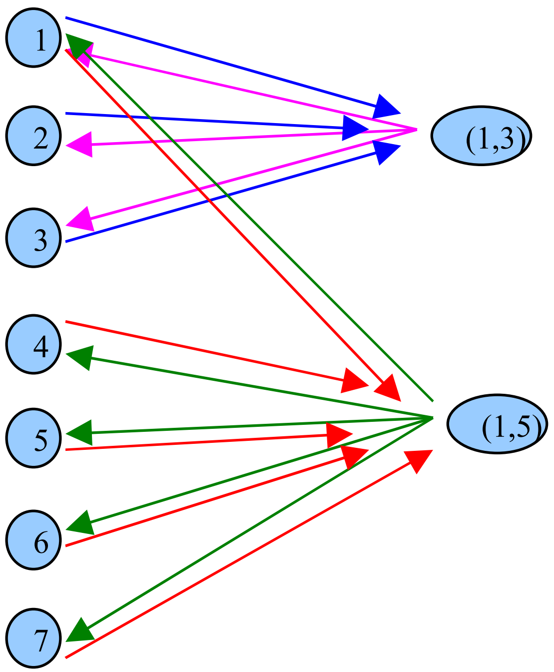
- Determina um circuito euleriano de *T*' utilizando a técnica de Euler-tour em árvores.

Algoritmo

Passo 3 (cont.):



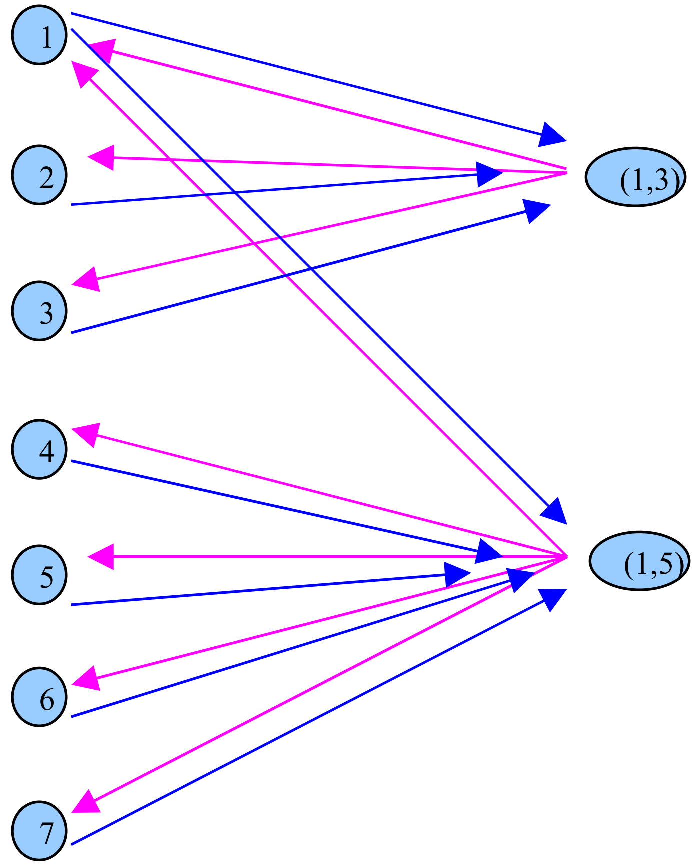
T



T'

Algoritmo

Passo 3 (cont.):



Algoritmo

Passo 4:

- Determina um ciclo *L* cujas arestas se alternam entre arestas de *T'* e arestas de *G*.
- Propriedade de *L*: as arestas de *G* e de *T'* aparecem em *L* na ordem de um circuito Euleriano em *G* e um circuito Euleriano em *T'*.
- Definir uma ordem circular para cada $w \in C$.
- Considere *w* de grau *d* em *T*. Seja (v_0,w), (v_1,w), ..., (v_{d-1},w) vértices adjacentes a *w* em *T, onde* v_0, v_1, ..., v_{d-1}, são vértices reais.

Algoritmo

Passo 4 (cont.):

- Modificar os vetores EDGE e SUCESSOR de maneira que descrevam *L*.
- Adicionar a EDGE as arestas de *T'* e, para cada $w \in C$ e $0 \leq \alpha \leq d$-1:

  $\text{SUCESSOR}(v_\alpha, w) \leftarrow (v_\alpha, j_\alpha)$
- Para as arestas de *G*:

  $\text{SUCESSOR}(i_\alpha, v_\alpha) \leftarrow (w_\alpha, v_\alpha)$
- Para as aresta do tipo (w, v_α), em *T* considere v_α adjacente a $w_0, w_1, \ldots, w_{d-1}$:

  $\text{SUCESSOR}(w_i, v_\alpha) \leftarrow (v_\alpha, w_{i+1(\text{mod } d)})$

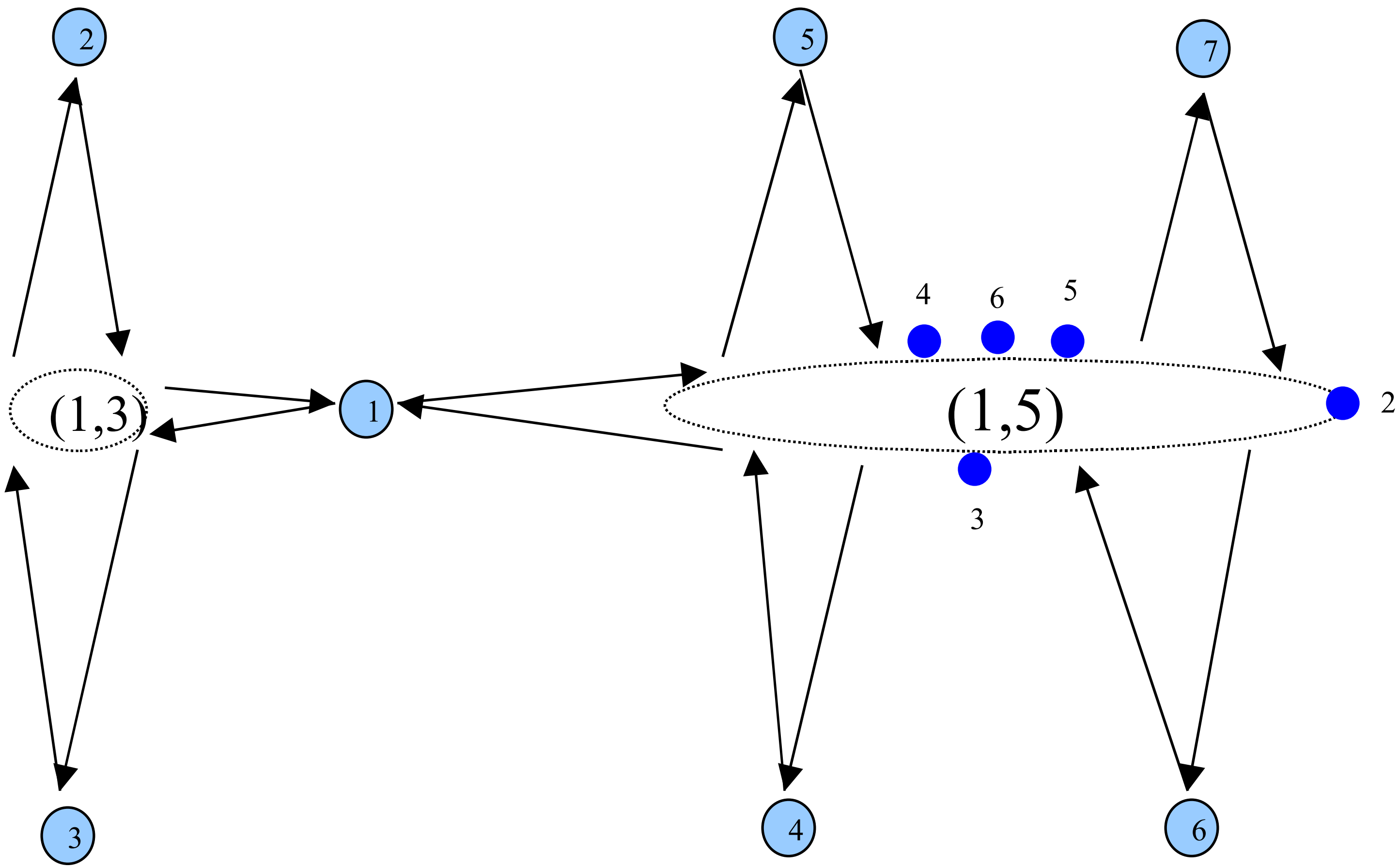
Algoritmo

Passo 4 (cont.):

- O ciclo *L* é descrito por EDGE e SUCESSOR.
- Segue que, *L* conterá as arestas de *T'* na ordem de um circuito Euleriano.
- Agora, cada aresta w $\in$ T' é expandindo para o circuito definido por SUCESSOR.

Algoritmo

Passo 4 (cont.):



Algoritmo

Passo 5:

- O algoritmo descarta de L as arestas de T', obtendo, assim, o circuito Euleriano de G.