Universidade Federal de Mato Grosso do Sul
Departamento de Computação e Estatística

"Tópicos em Algoritmos Paralelos e Distribuídos"

Circuitos de Euler em Paralelo

Claudia Nasu
cnasu@dct.ufms.br

Apresentação

- Preliminares
- Algoritmo
- Um exemplo

Preliminares

- Um circuito de Euler é um ciclo que passa por cada aresta do grafo exatamente uma vez.

- Um grafo euleriano *G* pode ser decomposto em um conjunto de circuitos disjuntos em aresta *C*, chamado *partição de Euler* de *G*.

Algoritmo

- Cáceres, Deo, Sastry e Szwarcfiter
- Modelo: PRAM CREW
- Complexidade:
  - Tempo: $O(log^2\ n)$
  - Processadores: $O(m/log\ m)$
- Pré-requisitos:
  - Ordenação inteira
  - List ranking ótimo

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, representado pelos vetores EDGE e SUC

- O algoritmo utiliza uma estrutura denominada **suporte *(strut)***.
- O suporte é utilizado para identificar diretamente os vértices em que uma operação denominada **costura *(stitch)*** deve ser aplicada.
- Esta operação une dois ou mais circuitos disjuntos em arestas, que passam por um determinado vértice, em um único circuito.

Algoritmo

Passo 1:

- Encontrar uma partição de Euler *C* para *G:*
  - Ordenar os vetores EDGE e SUC lexicograficamente.

EDGE:

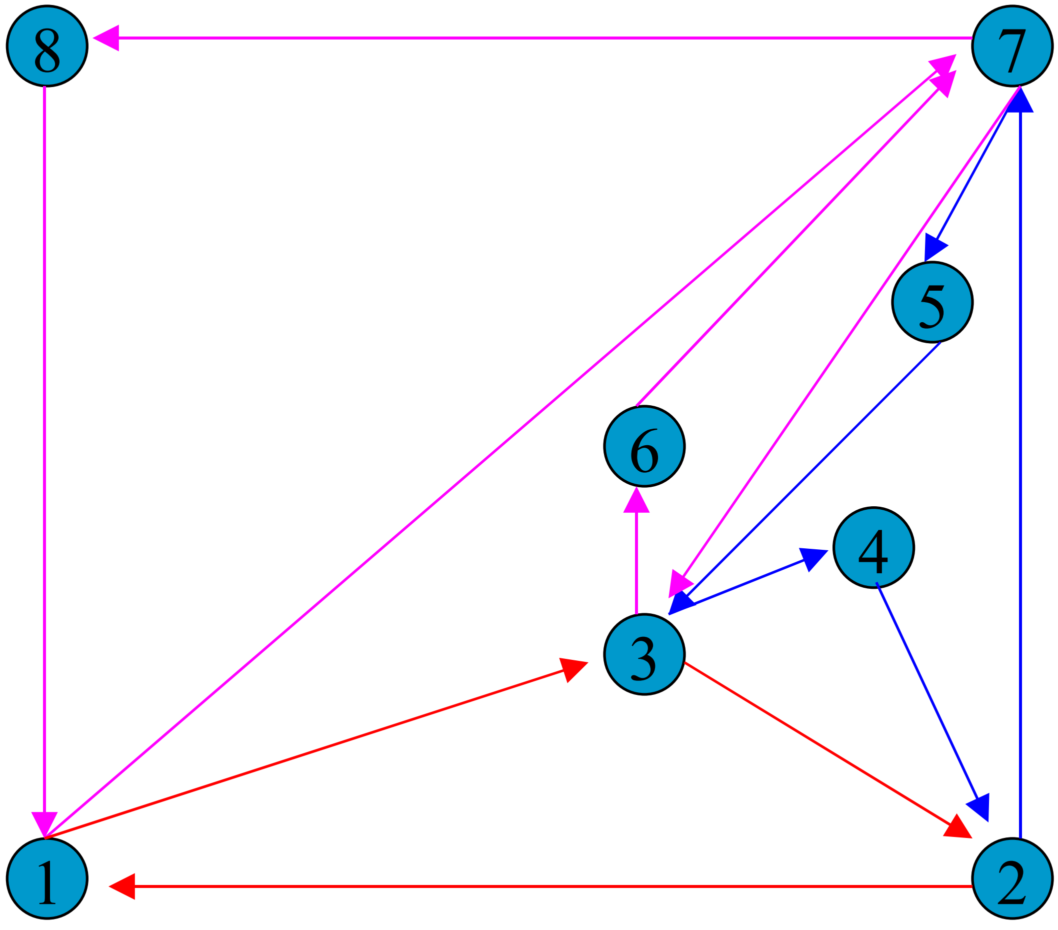
(2,1)	(8,1)	(3,2)	(4,2)	(1,3)	(5,3)	(7,3)	(3,4)	(7,5)	(3,6)	(1,7)	(2,7)	(6,7)	(7,8)

SUC:

(1,3)	(1,7)	(2,1)	(2,7)	(3,2)	(3,4)	(3,6)	(4,2)	(5,3)	(6,7)	(7,3)	(7,5)	(7,8)	(8,1)

Algoritmo

Exemplo:



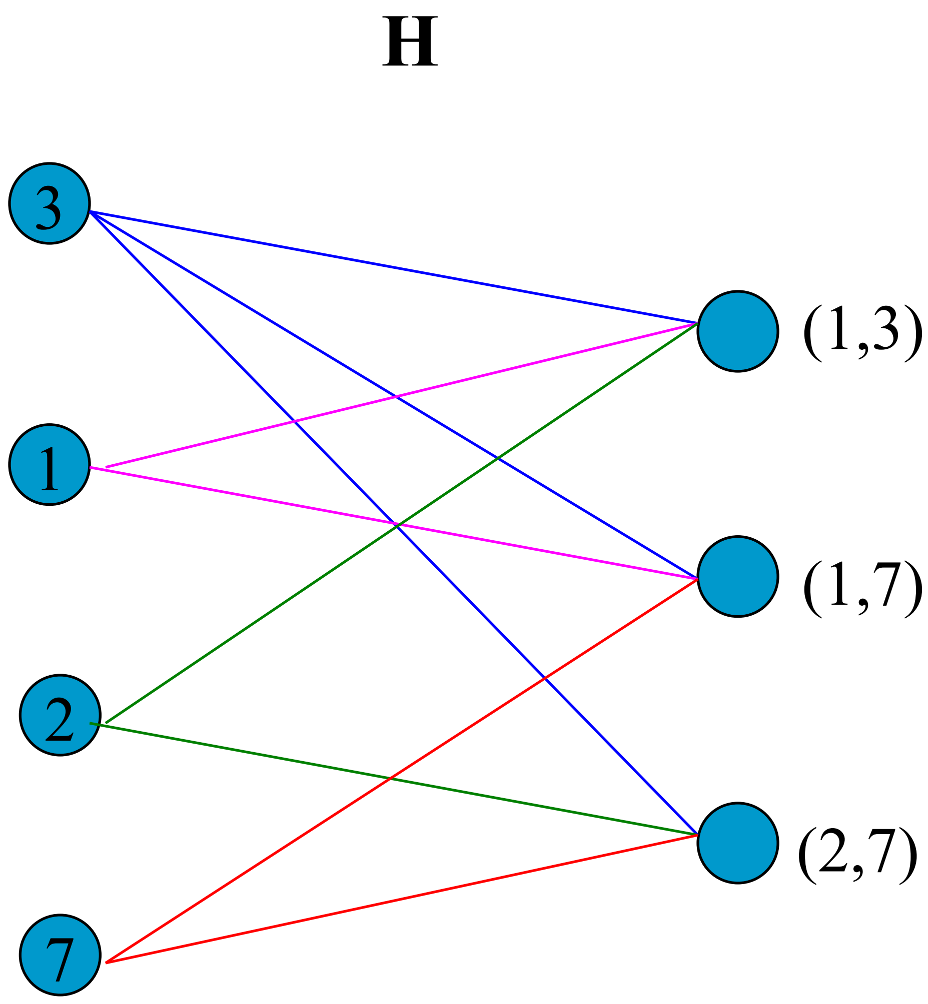
Algoritmo

Passo 2:

- Construa um grafo bipartido $H = (V_1, C', E_1)$, tal que:
  - $V_1 \subseteq V$ um subconjunto formado por vértices que possuam mais que um circuito de C passando por ele.
  - C' é o conjunto de vértices que possui um correspondência um-para-um com os circuitos de C.
  - E_1 as arestas conectam um vértice $v_i \in V_1$ com os vértices em C' que correspondem a circuitos que passam por v_i.

Algoritmo

Passo 2:



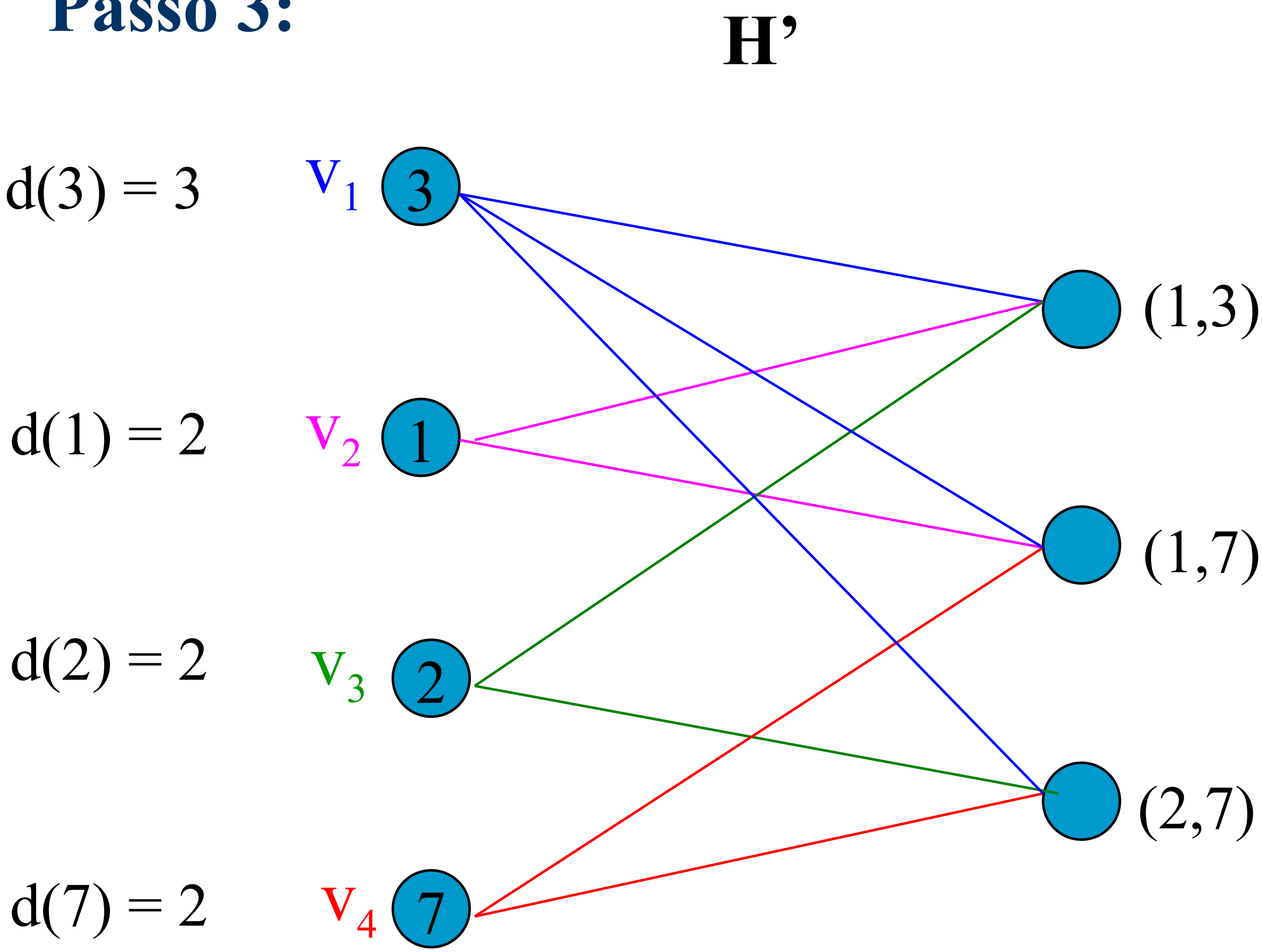
Algoritmo

Passo 3:

- Rotular os vértices de H :
  - Seja v um vértice no subgrafo H' de H.
  - $d_{H'}(v)$ indica o grau de v em H'.
  - Ordene os vértices de V_1 em ordem não-crescente de seus graus e rotule como v_1, v_2, ..., $v_{|H'|}$.

Algoritmo

Passo 3:



Algoritmo

Passo 4:

- Defina um strut (suporte) em V_1:

  – uma strut S é uma floresta geradora de H tal que a cada $c'_i \in C'$ incide, em S, exatamente uma aresta de E_1, e (v_i, c'_i) é uma aresta de S somente se (v_k, c'_i) não for uma aresta de H, para qualquer $v_k \in V_1$, $k < j$.

Algoritmo

Passo 4:

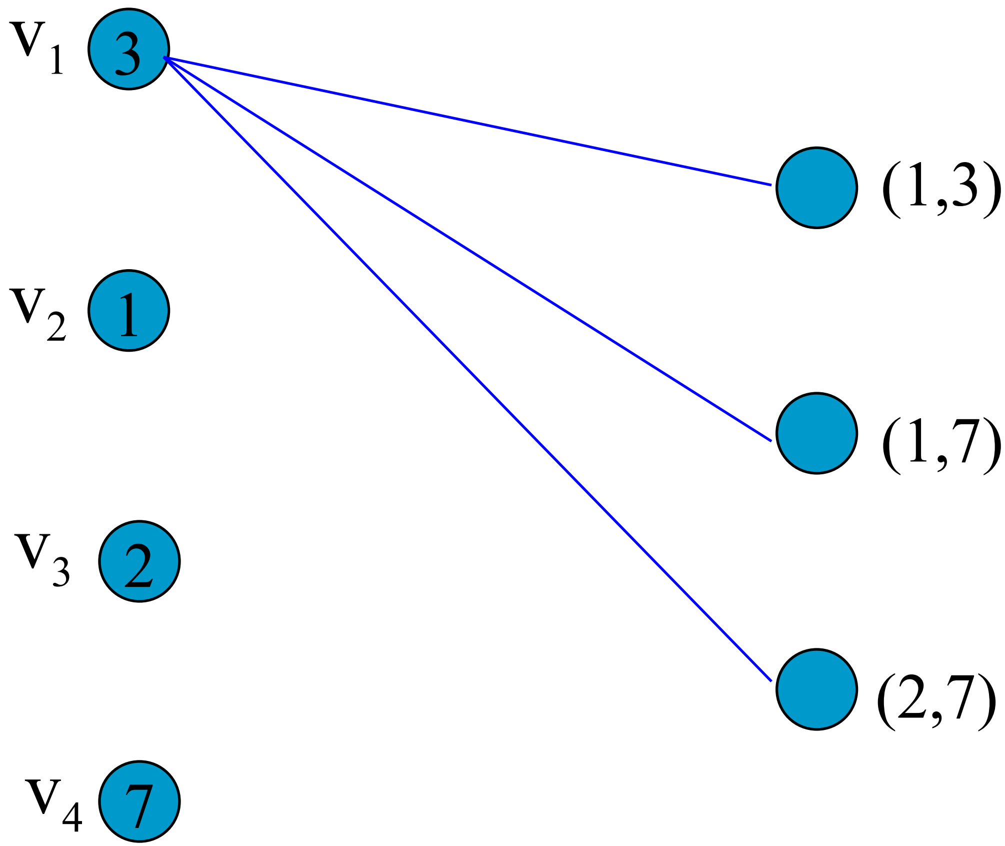
S

$d(3) = 3$

$d(1) = 0$

$d(2) = 0$

$d(7) = 0$



Algoritmo

Passo 4:

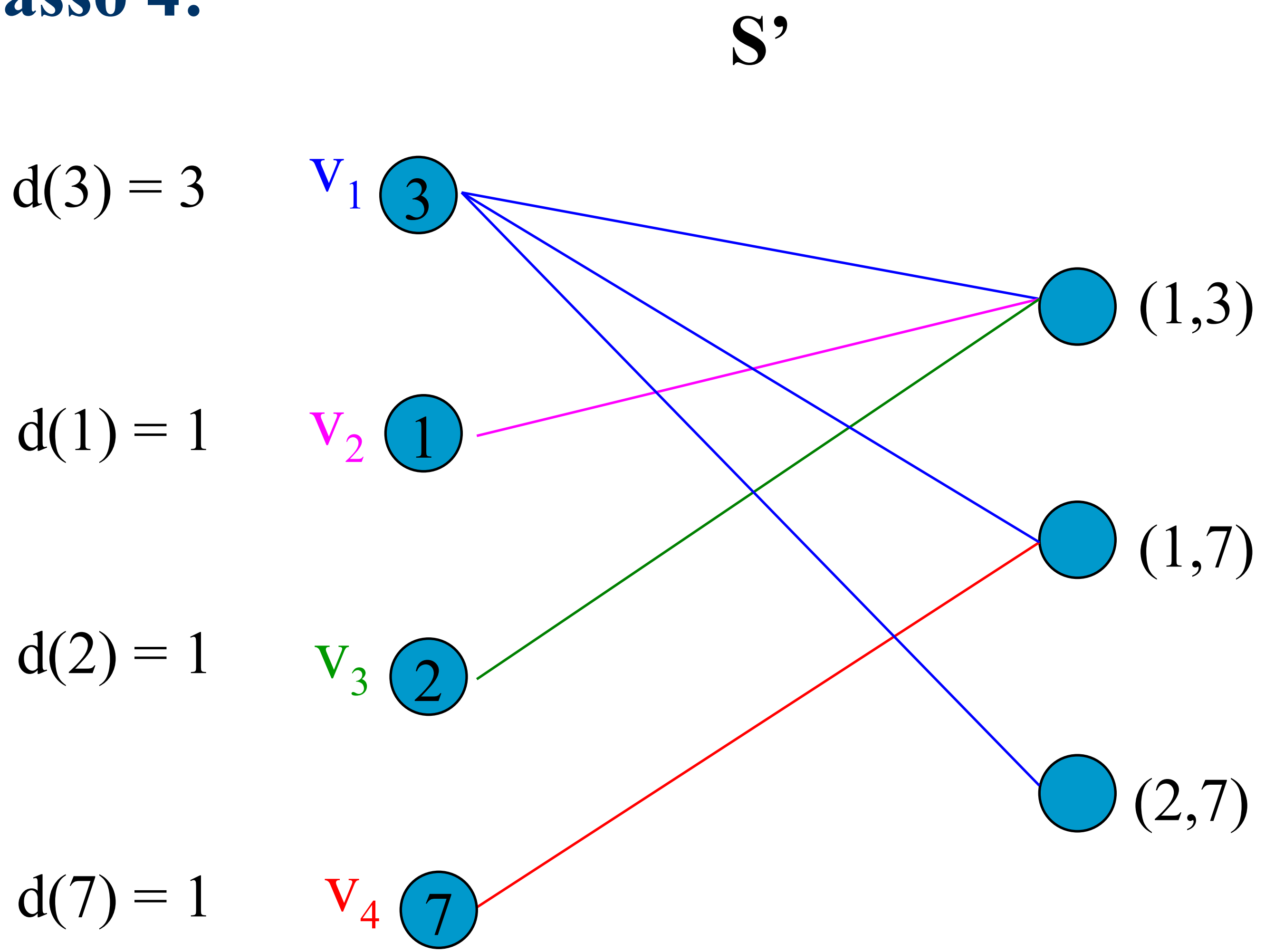
– Um vértice $v \in V_1$ é chamado de vértice zero-diferença em S se $d_H(v) - d_S(v) = 0$.

– Defina $S' = S$. Para cada vértice de V_1 não-zero-diferença em H, acrescente uma aresta de H em S'.

– Defina $d_{S'}(v)$ o grau de um vértice v em S'.

Algoritmo

Passo 4:



Algoritmo

Passo 4:

- Construa o vetor STITCH que contém todas as arestas que devem ser "costuradas".

- Considere todas as arestas (v_j, c'_k) de S' tal que $d_{S'}(v_j) > 1$:

  STITCH $\leftarrow$ arestas que entram em v_j e pertencem ao circuito C_k.

STITCH = {(1,3), (7,3), (5,3)}

Algoritmo

Passo 5:

- Aplicar a operação stitch:

  for cada aresta em STITCH **do in parallel**

  SUC[STITCH[i]] $\leftarrow$ SUC[STITCH[(i + 1) mod k]]

  onde k é d_S[STITCH[i].vertice2]

- No exemplo temos:

  SUC[(1,3)] $\leftarrow$ SUC[(7,3)]

  SUC[(7,3)] $\leftarrow$ SUC[(5,3)]

  SUC[(5,3)] $\leftarrow$ SUC[(1,3)]

Algoritmo

Passo 5:

EDGE:

(2,1)	(8,1)	(3,2)	(4,2)	(1,3)	(5,3)	(7,3)	(3,4)	(7,5)	(3,6)	(1,7)	(2,7)	(6,7)	(7,8)

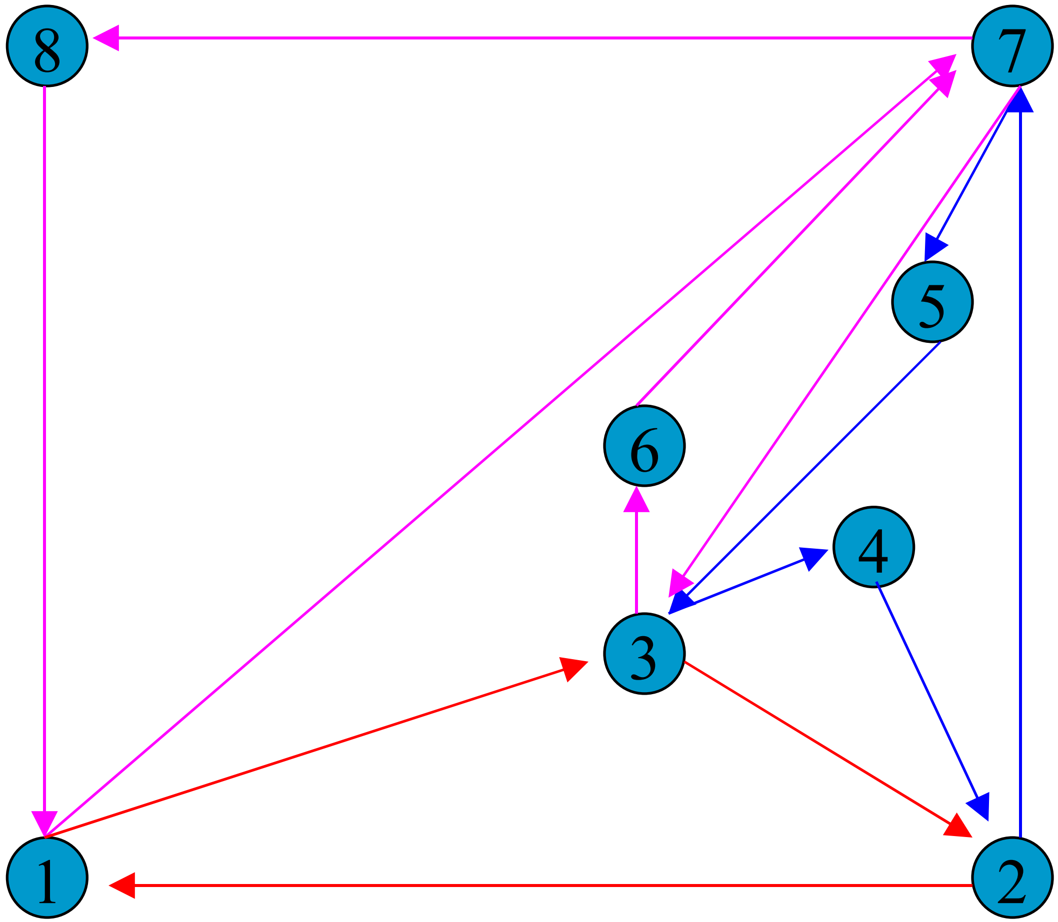
SUC:

(1,3)	(1,7)	(2,1)	(2,7)	(3,2)	(3,4)	(3,6)	(4,2)	(5,3)	(6,7)	(7,3)	(7,5)	(7,8)	(8,1)

(3,6) (3,2) (3,4)

Algoritmo

Passo 5:



Algoritmo

Passo 6:

- Usando os vetores EDGE e SUC obtidos ao final do passo 5, aplicar os passos 2 a 5 até que um circuito de Euler para *G* seja encontrado.

Resultados

- Seja $H=(V_1, C', E1)$ um grafo bipartido e S um suporte em V_1 para H. Seja H' o grafo obtido, a partir de H, adicionando a S precisamente uma aresta de E_1-S incidente em cada vértice não zero-diferença de V_1. H' é acíclico. Além disso, se V_1 contém exatamente um vértice zero-diferença, então H' é uma árvore geradora de H.

- O número de circuitos disjuntos em arestas restantes depois do passo 5 é, no máximo, o número de circuitos disjuntos em arestas na iteração anterior dividido por dois.

Complexidade

- O algoritmo utiliza o modelo PRAM CREW e possui complexidade de tempo $O(\log^2 n)$ utilizando $O(m/\log m)$ processadores.