Emparelhamento Máximo

André da Cunha Ribeiro

Professor de Informática IFGoiano - Rio Verde
Doutor em Engenharia de Sistemas e Computação COPPE - UFRJ

Apresentação

1. Emparelhamento;
2. Caminhos aumentantes;
3. Resultados principais.

Emparelhamento

Definição

Um **emparelhamento** (ou matching) em um grafo $G = (V, E)$ é um conjunto de arestas $M \subseteq E(G)$ tal que quaisquer duas arestas não compartilham um vértice. Os vértices u, v são ditos **saturados** por M. Os vértices que não são incidentes em nenhuma aresta de M são ditos **insaturados**.

Emparelhamento - exemplo

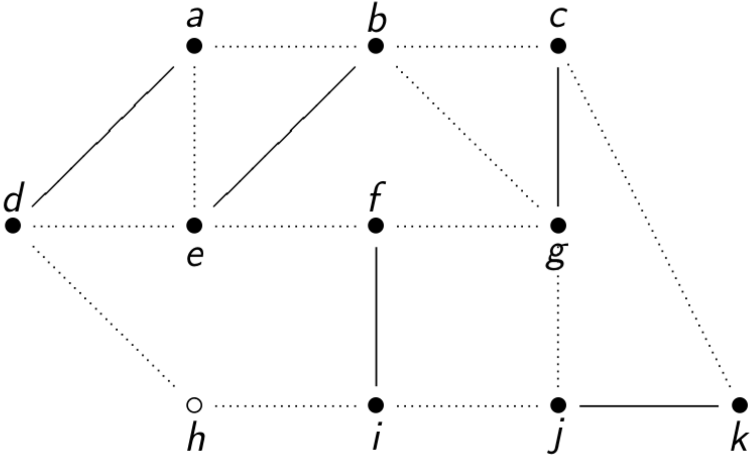


Figura: $M(G) = (ad, be, fi, cg, jk)$

Emparelhamento Máximo e Maximal

Definição

Um emparelhamento M é **maximal** se não é possível adicionar arestas a $E(M)$, e é **máximo** se for um emparelhamento maximal que contém o maior número de arestas. Todo emparelhamento máximo é maximal, porém nem todo maximal é máximo.

Emparelhamento - exemplo

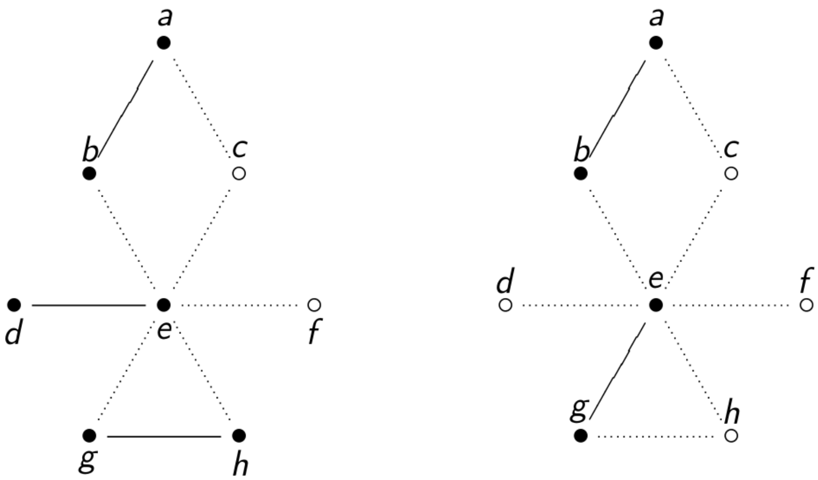
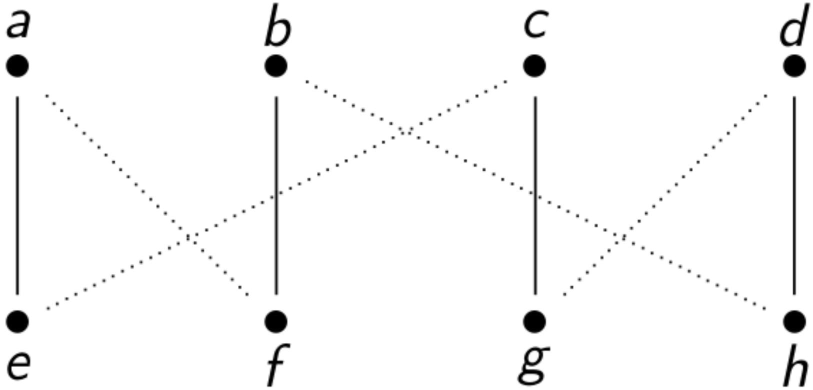


Figura: $M(G_a) = (ab, de, gh)$ - máximo e $M(G_b) = (ab, ge)$ - maximal

Emparelhamento Perfeito

Definição

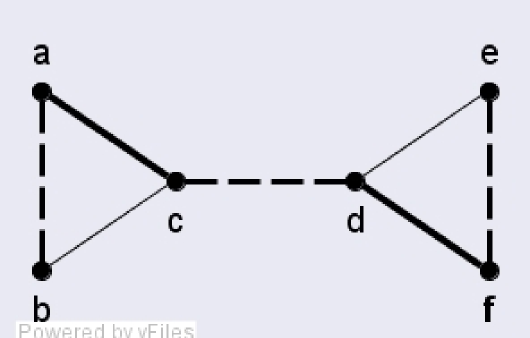
Um emparelhamento M é **perfeito** se satura todos os vértices de G.



Emparelhamento

Um **emparelhamento** no grafo é um conjunto de arestas, tal que, quaisquer duas arestas nesse conjunto não possuem vértices em comum.

- Emparelhamento Maximal
- Emparelhamento Máximo
- Emparelhamento Perfeito



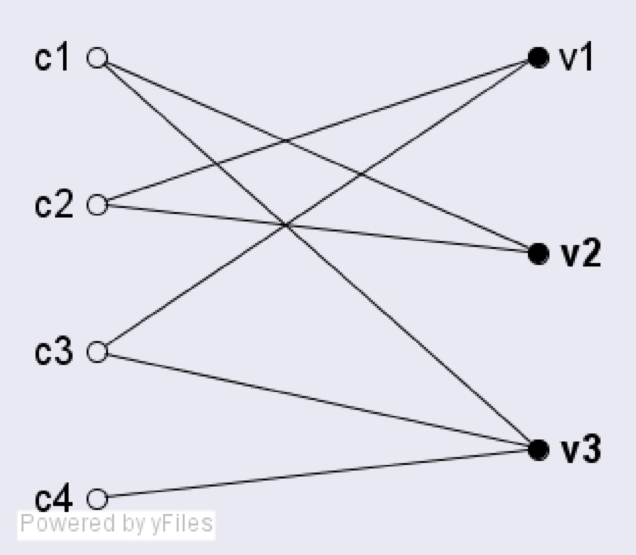
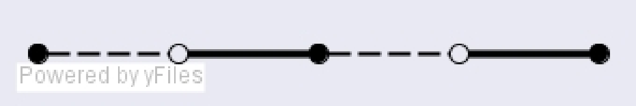


Figura: Aplicação de um emparelhamento máximo

Caminhos aumentantes

Definições

Seja M um emparelhamento qualquer Um **caminho** M**-alternante** é um caminho simples que alterna arestas em M e arestas que não estão em M.



Caminhos aumentantes

Definições

Seja M um emparelhamento qualquer. Um **caminho M-aumentante** é um caminho M-alternante onde os extremos não são saturados por M.

Caminhos aumentantes

Definição

1. Dado um emparelhamento qualquer M, um caminho M-**alternante** é um caminho que alterna arestas em M e arestas que não estão em M.
2. Um caminho M-**aumentante** é um caminho M-alternante onde os vértices finais não são saturados por M.
3. Logo, se existir um caminho P em G que seja M-aumentante, é possível criar um novo emparelhamento M', onde $|M'| = |M| + 1$, e $M' = \{(uv)|(uv) \in P \wedge (uv) \notin M\}$.

Caminhos aumentantes

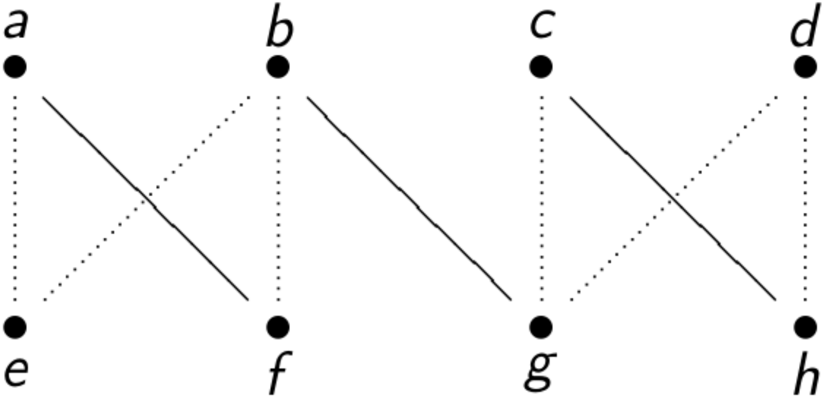


Figura: $CaminhoAumentante = (ea, \textbf{af}, fb, \textbf{bg}, gc, \textbf{ch}, hd); |M| = 3$

Caminhos alternantes

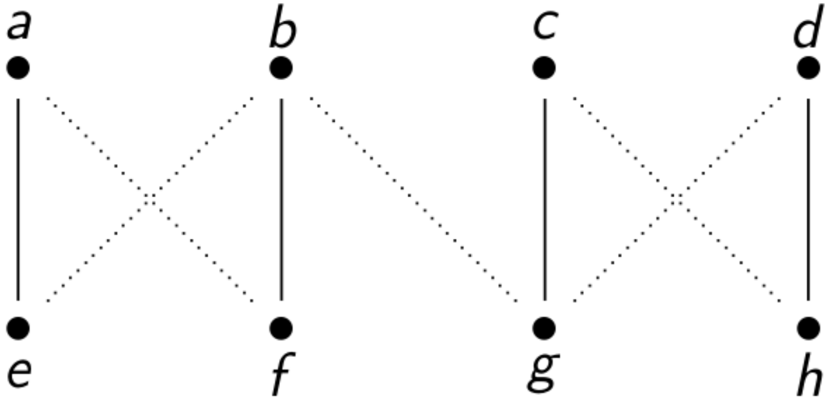


Figura: $CaminhoAlternante = (\textbf{ea}, af, \textbf{fb}, bg, \textbf{gc}, ch, \textbf{hd}); |M| = 4$

Sejam A e B dois conjuntos

A **diferença simétrica** entre esses conjuntos é denotado por $A \oplus B = (A - B) \cup (B - A)$.

Teorema (Berge 1957)

Um emparelhamento M tem cardinalidade máxima em G se, e somente se, G não possui caminho M-aumentante.

Método utilizado na maioria dos algoritmos

Algoritmo 1: MétodoBerge

```
início
    seja M um emparelhamento em G
    enquanto existir um caminho M-aumentante P em relação a M faça
        construa um emparelhamento M′ = M ⊕ P
        M = M′
    fim
fim
```

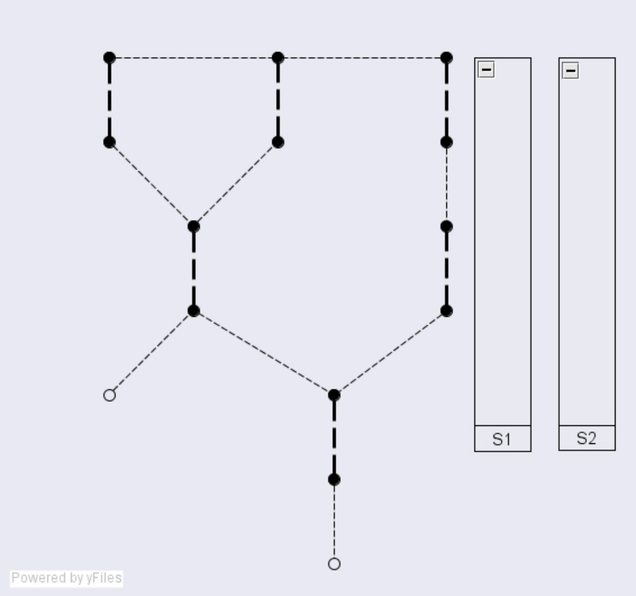
Principais algoritmos

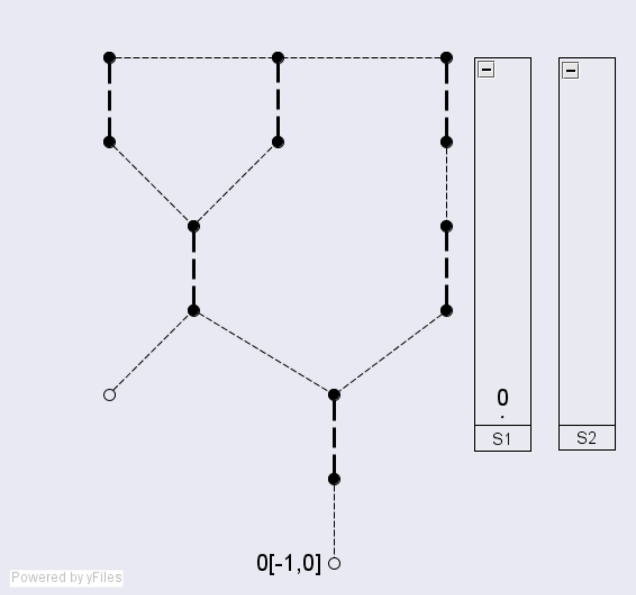
Referência	Limite tempo
1. Edmonds (1965)	n^4
2. Witzgall and Zahn (1965)	n^3
3. Kameda and Munro (1974)	nm
4. Micali and Vazirani (1980)	$m\sqrt{n}$

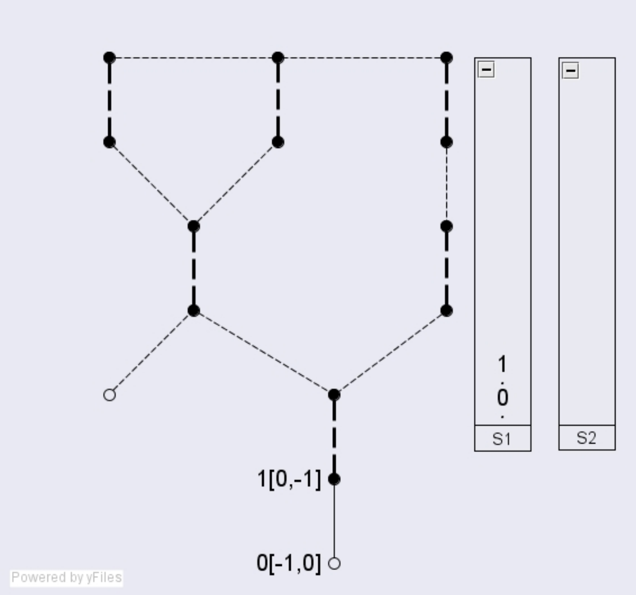
Tabela: Complexidade dos Algoritmos de Emparelhamento Máximo

Algoritmo 2: EmparelhamentoMáximo

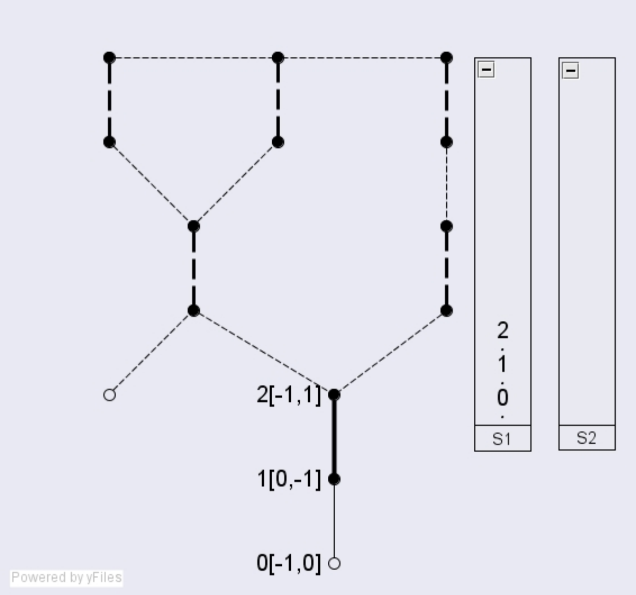
início
$M \leftarrow \emptyset$ **para cada** *vértice* $r \in V$ **faça**
se *vértice* r *não é saturado* **então**
$S_1 \leftarrow \emptyset$ $S_2 \leftarrow \emptyset$
$NUM \leftarrow \emptyset$
$p_o \leftarrow \emptyset$ $p_e \leftarrow \emptyset$
se *caminho-aumentante* **então**
$M \leftarrow$ aumentar-emparelhamento
fim
fim
fim
$M^* \leftarrow M$
fim

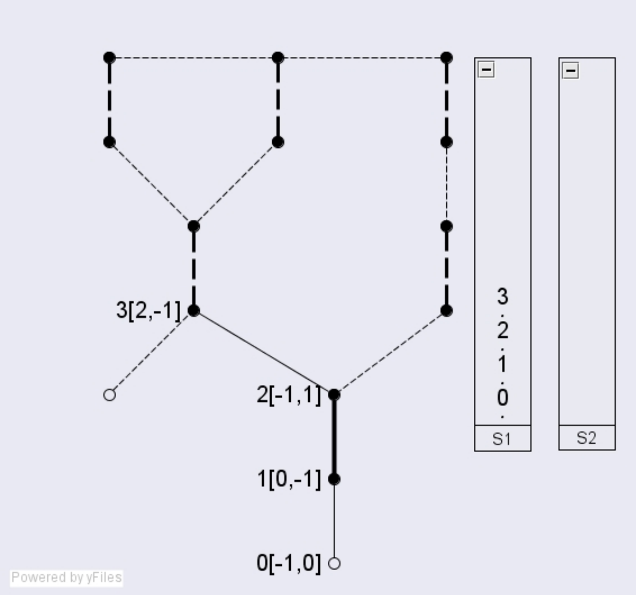

S1
S2
Powered by yFiles

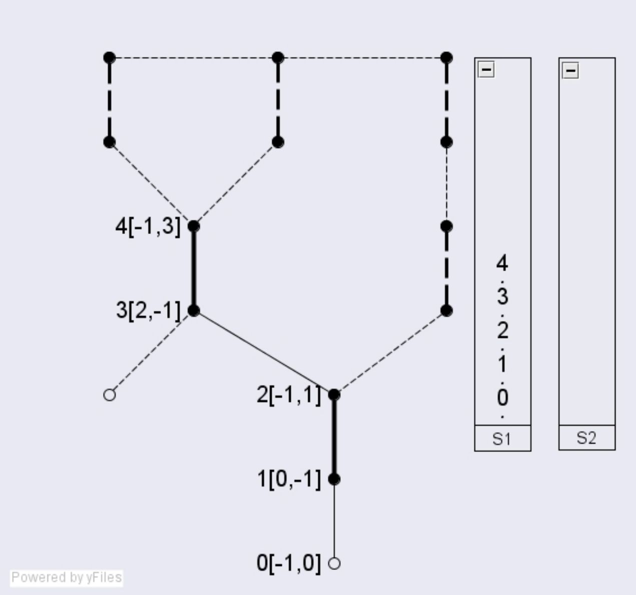

0
S1
S2
0[-1,0]
Powered by yFiles

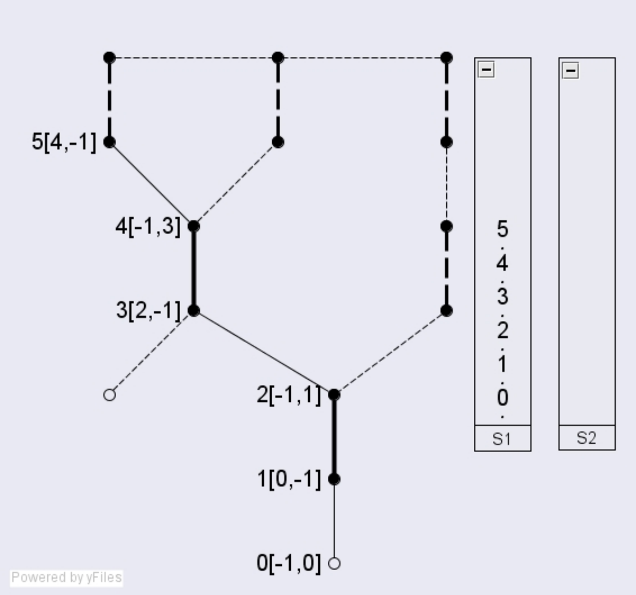


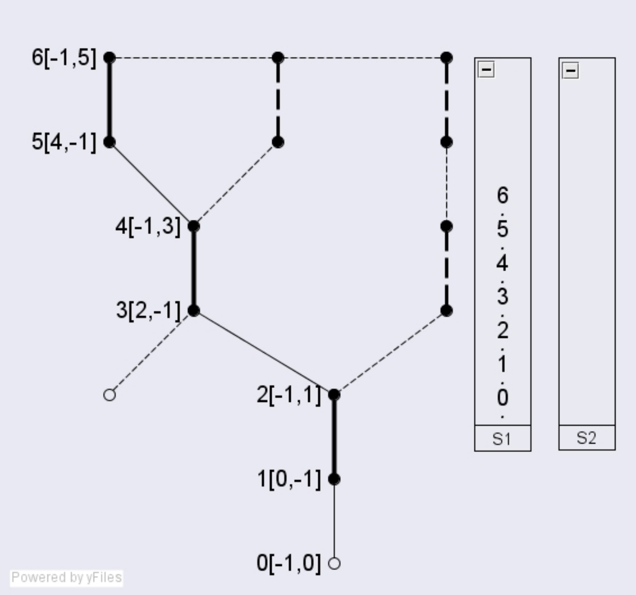
1
0
S1
S2
1[0,-1]
0[-1,0]
Powered by yFiles

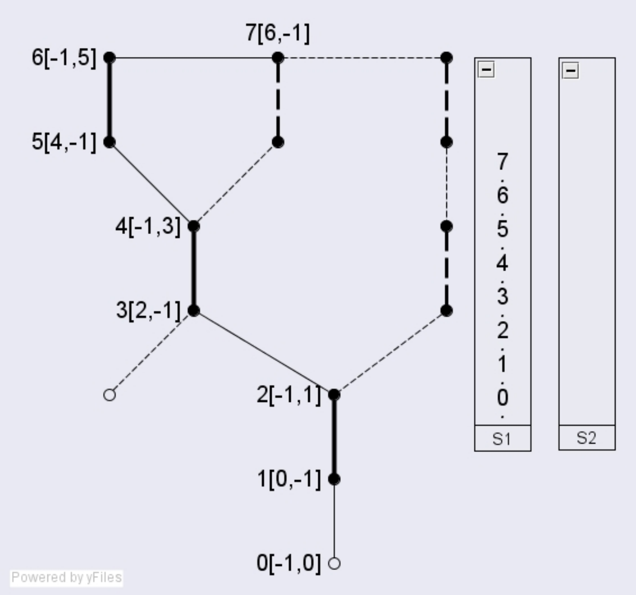

2[-1,1]
1[0,-1]
0[-1,0]
2
1
0
S1
S2
Powered by yFiles

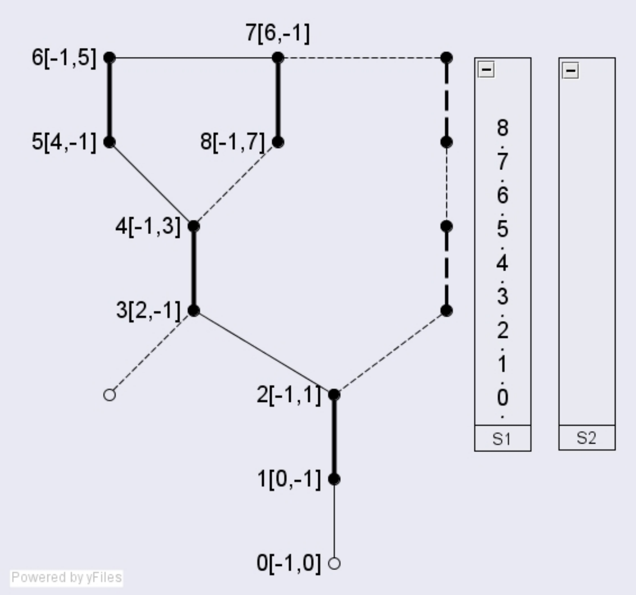

3[2,-1]
2[-1,1]
1[0,-1]
0[-1,0]
3
2
1
0
S1
S2
Powered by yFiles

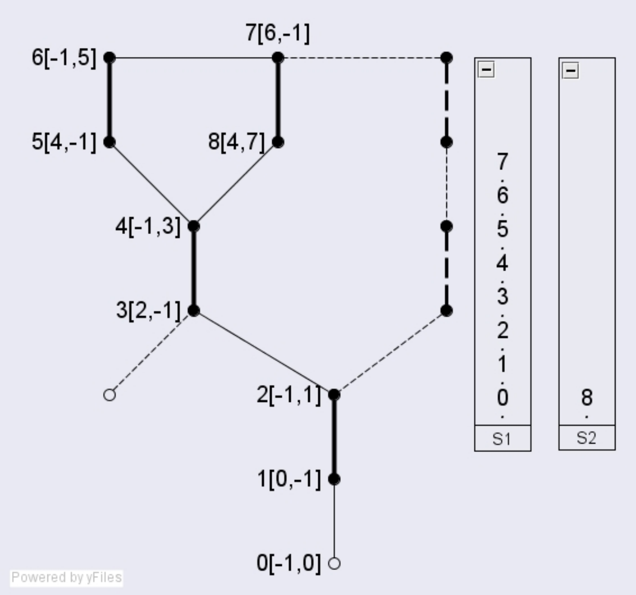

4[-1,3]
3[2,-1]
2[-1,1]
1[0,-1]
0[-1,0]
4
3
2
1
0
S1
S2
Powered by yFiles

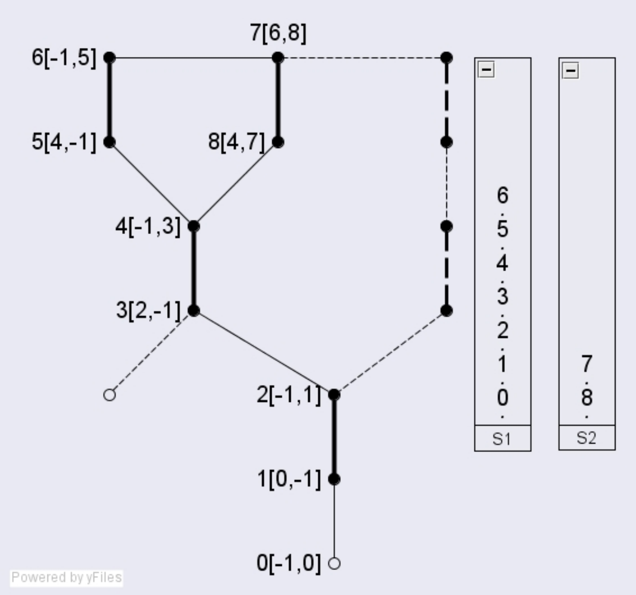

5[4,-1]
4[-1,3]
3[2,-1]
2[-1,1]
1[0,-1]
0[-1,0]
5 . 4 . 3 . 2 . 1 . 0 .
S1
S2
Powered by yFiles

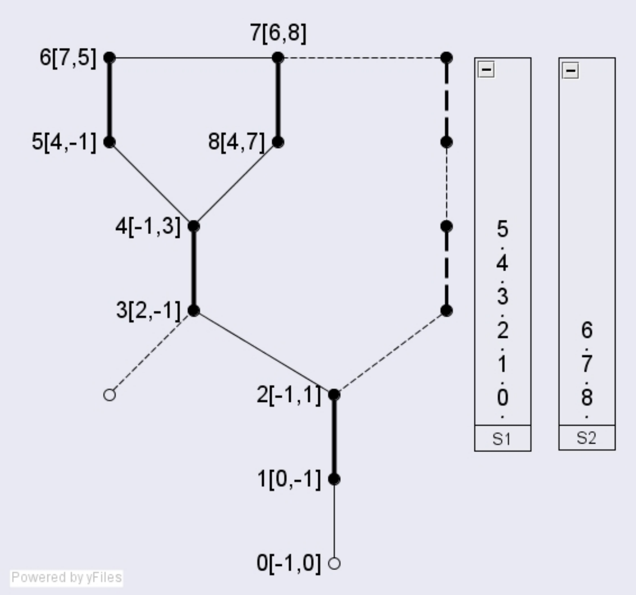

6[-1,5]
5[4,-1]
4[-1,3]
3[2,-1]
2[-1,1]
1[0,-1]
0[-1,0]
6
5
4
3
2
1
0
S1
S2
Powered by yFiles

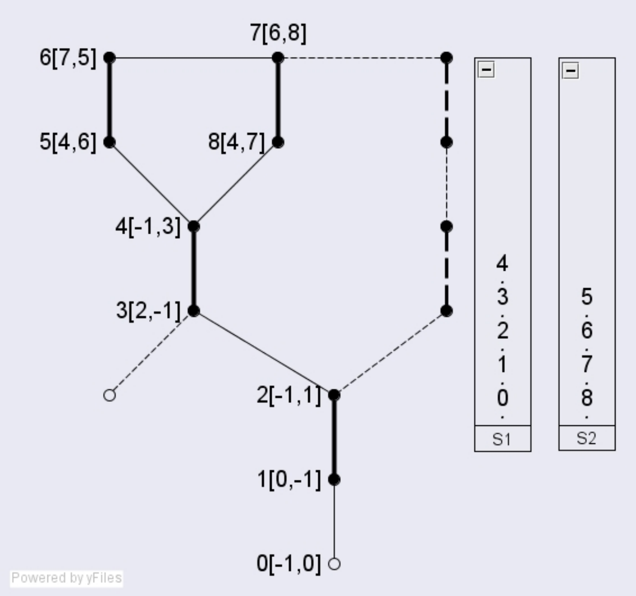

7[6,-1]
6[-1,5]
5[4,-1]
4[-1,3]
3[2,-1]
2[-1,1]
1[0,-1]
0[-1,0]
7
6
5
4
3
2
1
0
S1
S2
Powered by yFiles

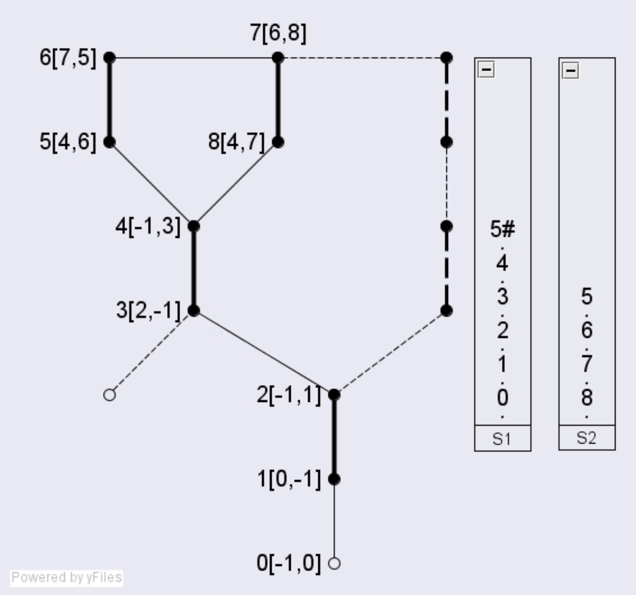

7[6,-1]
6[-1,5]
5[4,-1]
8[-1,7]
4[-1,3]
3[2,-1]
2[-1,1]
1[0,-1]
0[-1,0]
8
7
6
5
4
3
2
1
0
S1
S2
Powered by yFiles

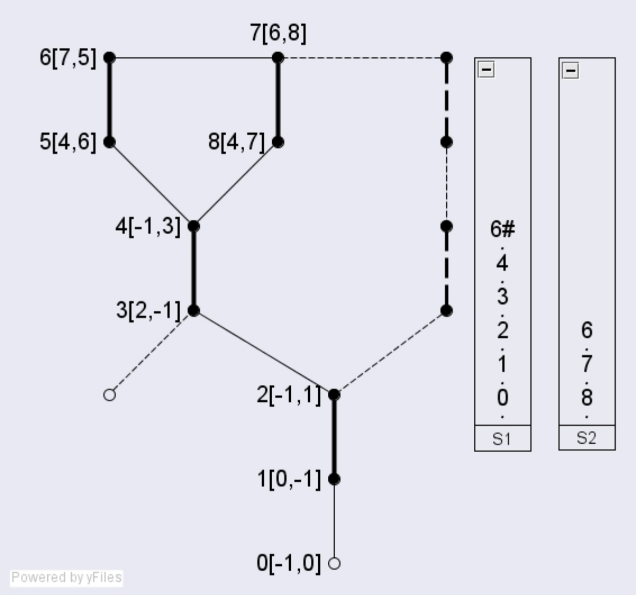

7[6,-1]
6[-1,5]
5[4,-1]
8[4,7]
4[-1,3]
3[2,-1]
2[-1,1]
1[0,-1]
0[-1,0]
7
6
5
4
3
2
1
0
S1
8
S2
Powered by yFiles

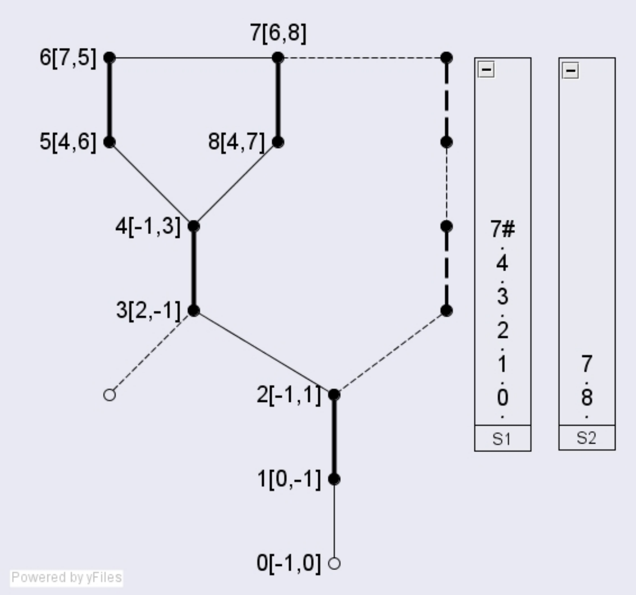

7[6,8]
6[-1,5]
5[4,-1]
8[4,7]
4[-1,3]
3[2,-1]
2[-1,1]
1[0,-1]
0[-1,0]
6
5
4
3
2
1
0
S1
7
8
S2
Powered by yFiles

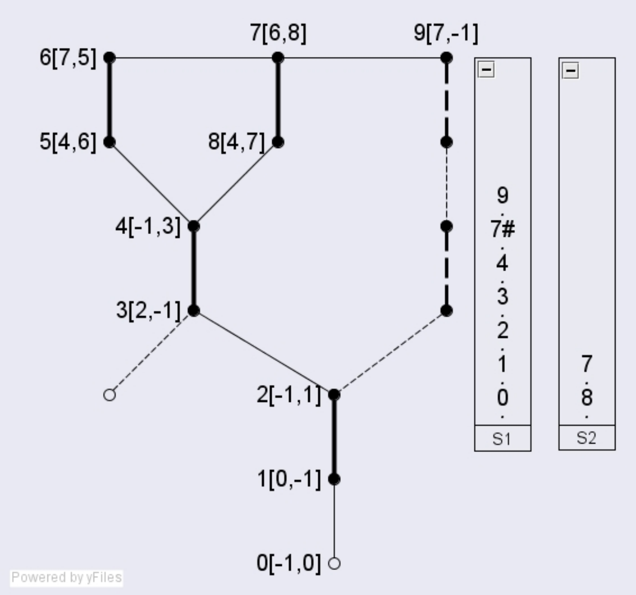

7[6,8]
6[7,5]
5[4,-1]
8[4,7]
4[-1,3]
3[2,-1]
2[-1,1]
1[0,-1]
0[-1,0]
5
4
3
2
1
0
S1
6
7
8
S2
Powered by yFiles

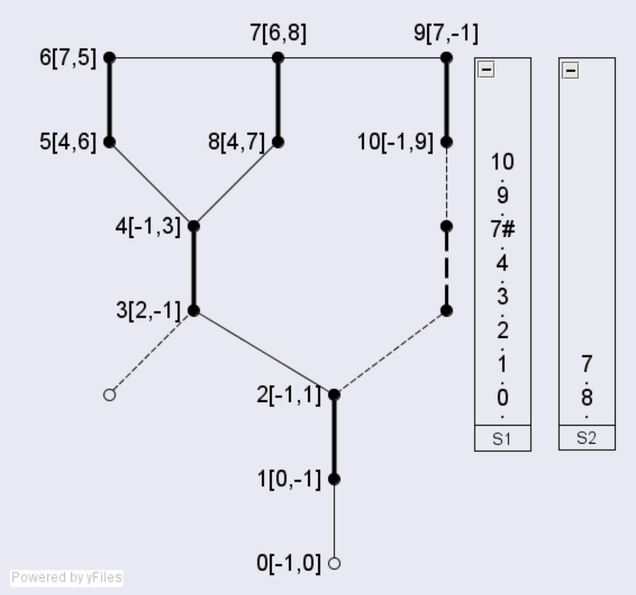

7[6,8]
6[7,5]
5[4,6]
8[4,7]
4[-1,3]
3[2,-1]
2[-1,1]
1[0,-1]
0[-1,0]
4
3
2
1
0
S1
5
6
7
8
S2
Powered by yFiles

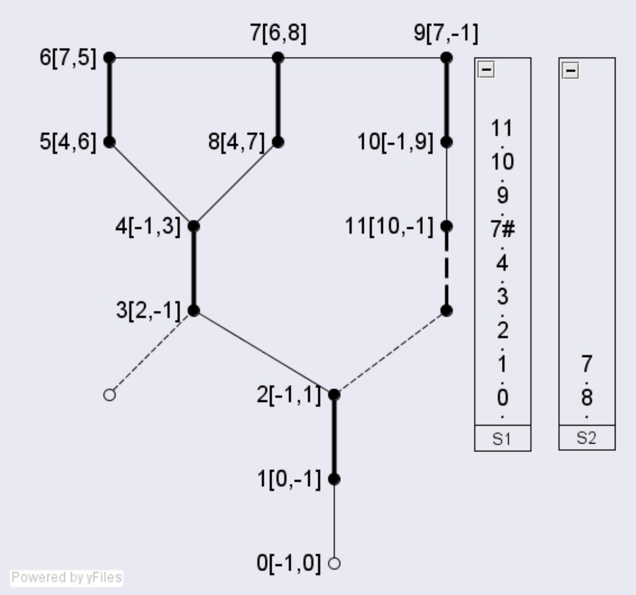

7[6,8]
6[7,5]
5[4,6]
8[4,7]
4[-1,3]
3[2,-1]
2[-1,1]
1[0,-1]
0[-1,0]
5#
4
3
2
1
0
S1
5
6
7
8
S2
Powered by yFiles


7[6,8]
6[7,5]
5[4,6]
8[4,7]
4[-1,3]
3[2,-1]
2[-1,1]
1[0,-1]
0[-1,0]
6#
4
3
2
1
0
S1
6
7
8
S2
Powered by yFiles

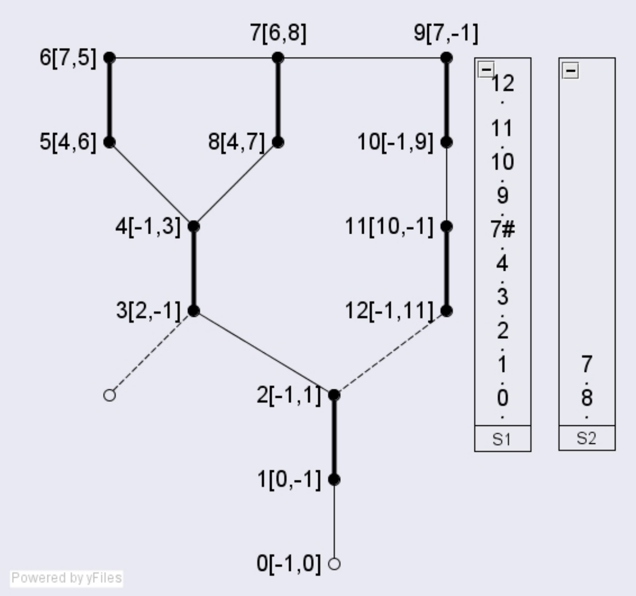

7[6,8]
6[7,5]
5[4,6]
8[4,7]
4[-1,3]
3[2,-1]
2[-1,1]
1[0,-1]
0[-1,0]
7#
4
3
2
1
0
S1
7
8
S2
Powered by yFiles

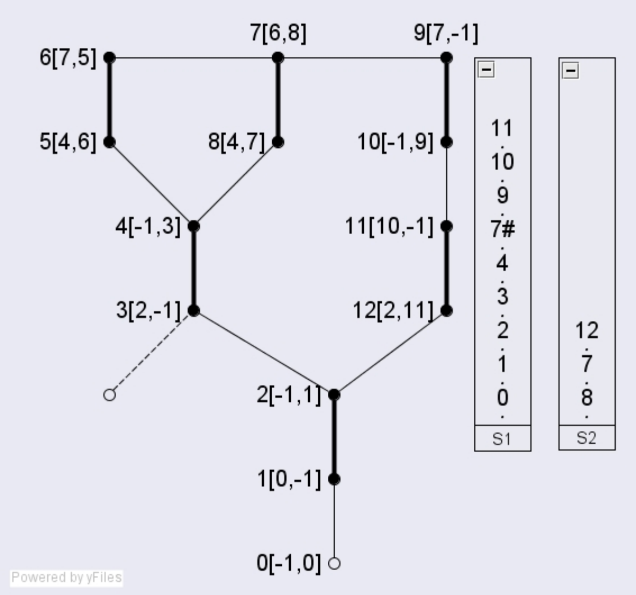

7[6,8]
9[7,-1]
6[7,5]
5[4,6]
8[4,7]
4[-1,3]
3[2,-1]
2[-1,1]
1[0,-1]
0[-1,0]
9
7#
4
3
2
1
0
S1
7
8
S2
Powered by yFiles


6[7,5]
7[6,8]
9[7,-1]
5[4,6]
8[4,7]
10[-1,9]
4[-1,3]
3[2,-1]
2[-1,1]
1[0,-1]
0[-1,0]
10
9
7#
4
3
2
1
0
S1
7
8
S2
Powered by yFiles

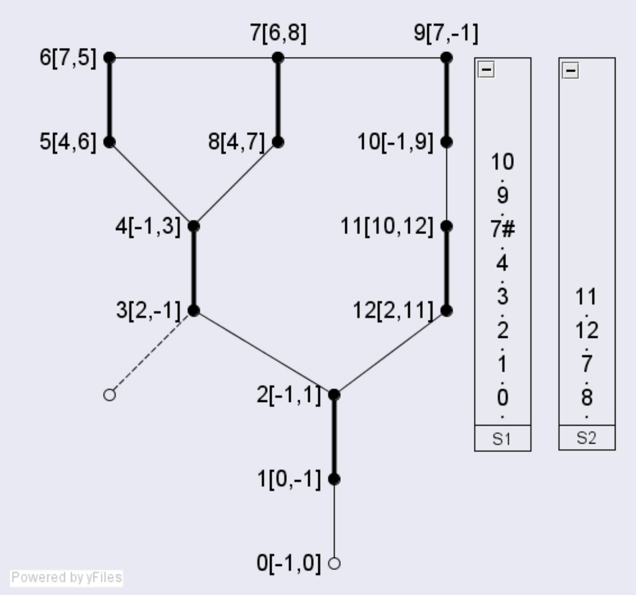


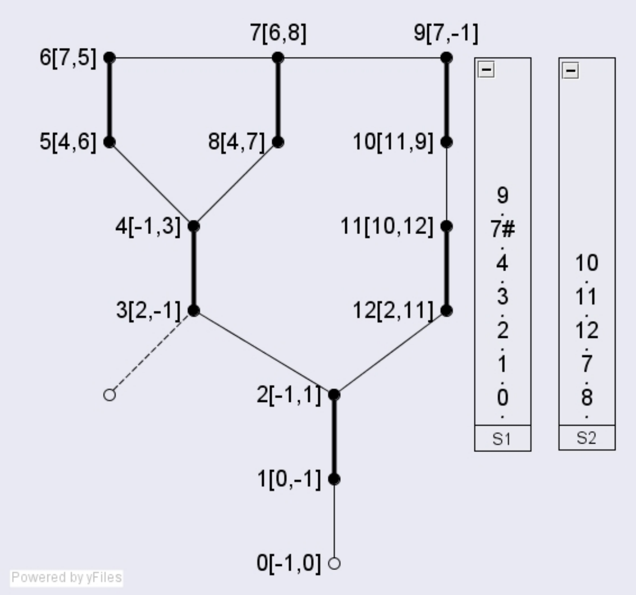
7[6,8]
9[7,-1]
6[7,5]
5[4,6]
8[4,7]
10[-1,9]
4[-1,3]
11[10,-1]
3[2,-1]
2[-1,1]
1[0,-1]
0[-1,0]
11
10
9
7#
4
3
2
1
0
S1
7
8
S2
Powered by yFiles

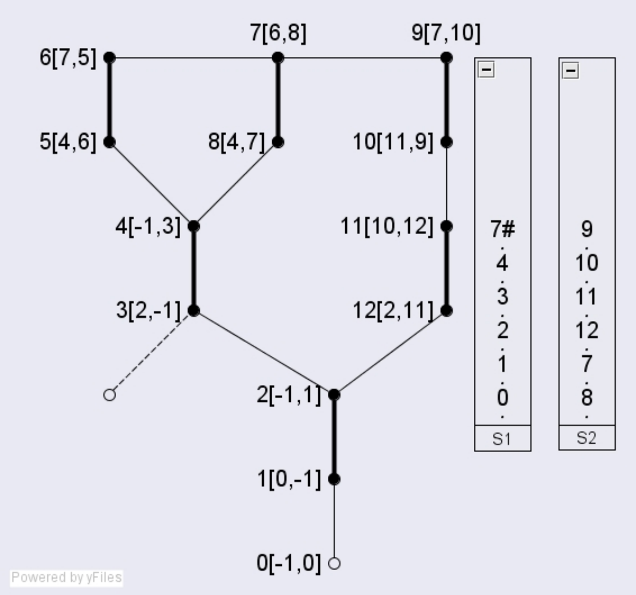

7[6,8]
9[7,-1]
6[7,5]
5[4,6]
8[4,7]
10[-1,9]
4[-1,3]
11[10,-1]
3[2,-1]
12[-1,11]
2[-1,1]
1[0,-1]
0[-1,0]
12
11
10
9
7#
4
3
2
1
0
S1
7
8
S2
Powered by yFiles



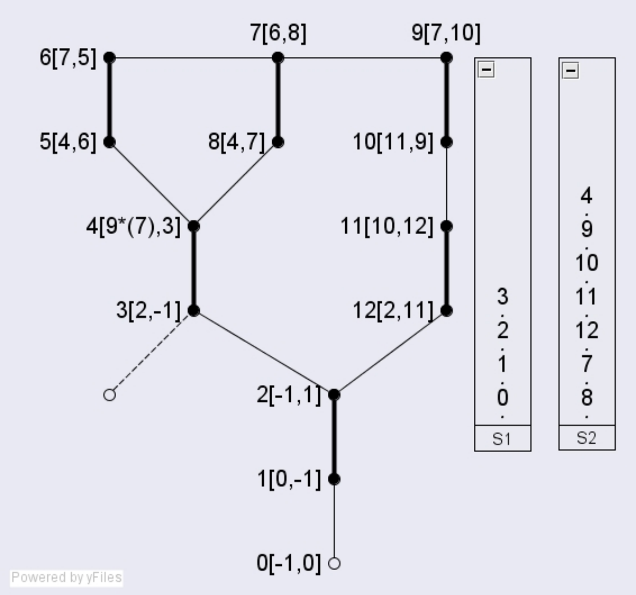
7[6,8]
9[7,-1]
6[7,5]
5[4,6]
8[4,7]
10[-1,9]
4[-1,3]
11[10,-1]
3[2,-1]
12[2,11]
2[-1,1]
1[0,-1]
0[-1,0]
11
10
9
7#
4
3
2
1
0
S1
12
7
8
S2
Powered by yFiles

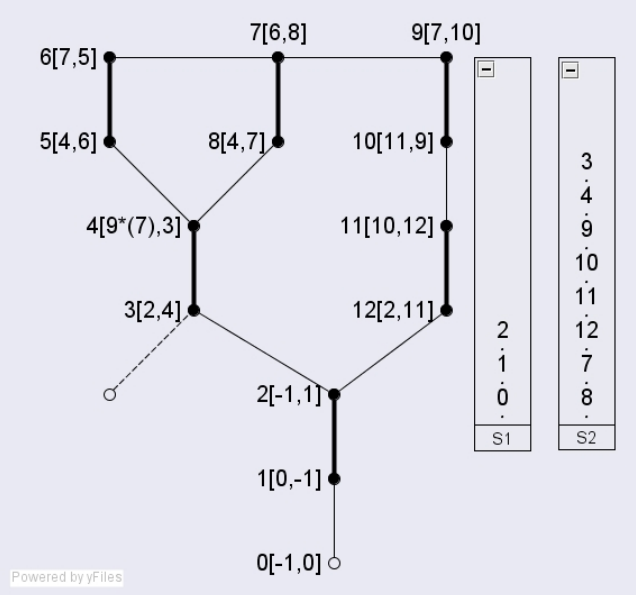

7[6,8]
9[7,-1]
6[7,5]
5[4,6]
8[4,7]
10[-1,9]
4[-1,3]
11[10,12]
3[2,-1]
12[2,11]
2[-1,1]
1[0,-1]
0[-1,0]
10
9
7#
4
3
2
1
0
S1
11
12
7
8
S2
Powered by yFiles

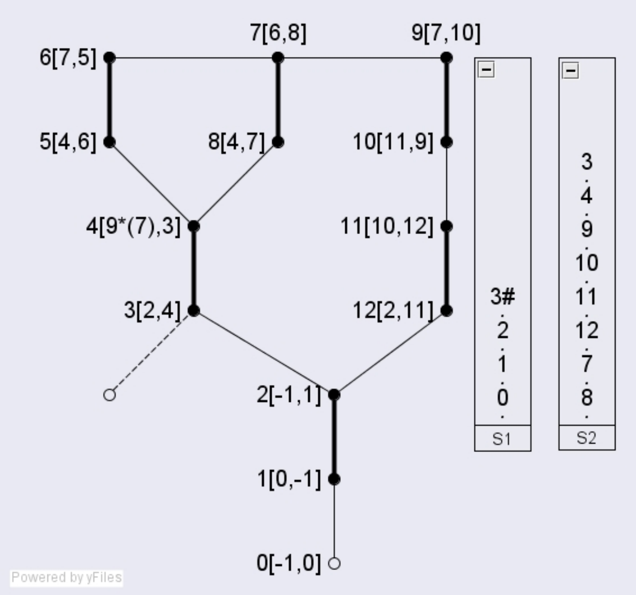

7[6,8]
9[7,-1]
6[7,5]
5[4,6]
8[4,7]
10[11,9]
4[-1,3]
11[10,12]
3[2,-1]
12[2,11]
2[-1,1]
1[0,-1]
0[-1,0]
9
7#
4
3
2
1
0
S1
10
11
12
7
8
S2
Powered by yFiles

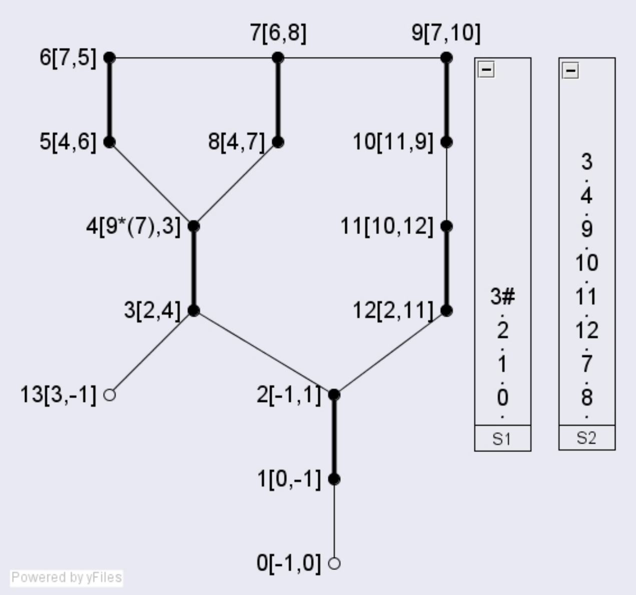


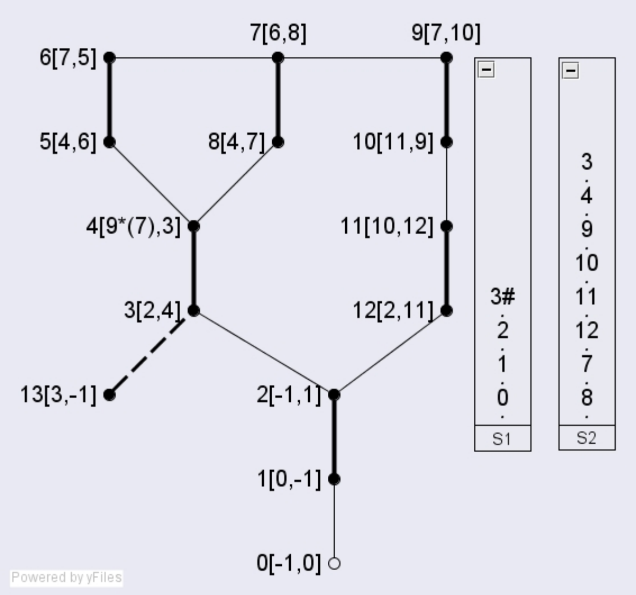
7[6,8]
9[7,10]
6[7,5]
5[4,6]
8[4,7]
10[11,9]
4[-1,3]
11[10,12]
3[2,-1]
12[2,11]
2[-1,1]
1[0,-1]
0[-1,0]
7#
4
3
2
1
0
S1
9
10
11
12
7
8
S2
Powered by yFiles

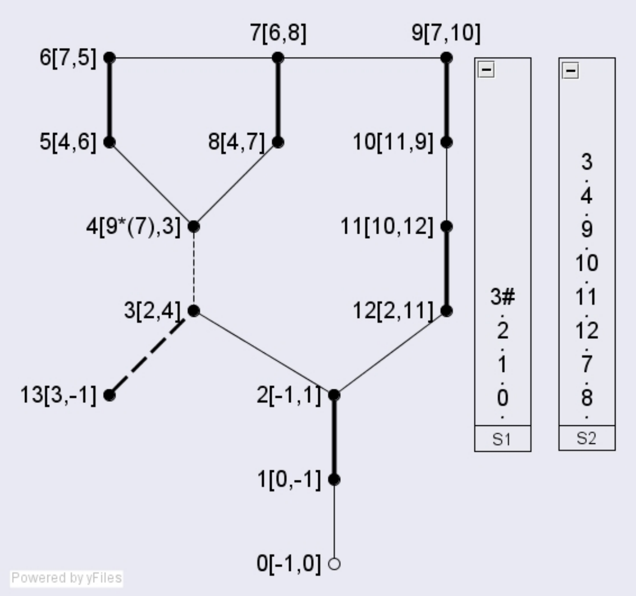

7[6,8]
9[7,10]
6[7,5]
5[4,6]
8[4,7]
10[11,9]
4[9*(7),3]
11[10,12]
3[2,-1]
12[2,11]
2[-1,1]
1[0,-1]
0[-1,0]
3
2
1
0
S1
4
9
10
11
12
7
8
S2
Powered by yFiles


7[6,8]
9[7,10]
6[7,5]
5[4,6]
8[4,7]
10[11,9]
4[9*(7),3]
11[10,12]
3[2,4]
12[2,11]
2[-1,1]
1[0,-1]
0[-1,0]
2
1
0
S1
3
4
9
10
11
12
7
8
S2
Powered by yFiles

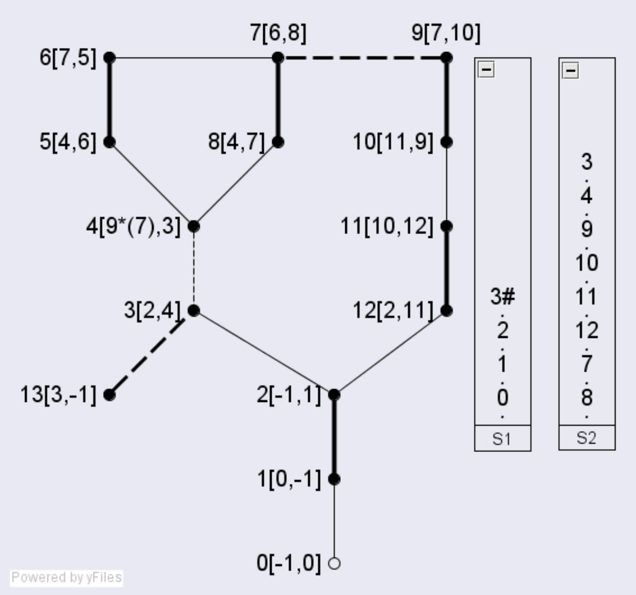

7[6,8]
9[7,10]
6[7,5]
5[4,6]
8[4,7]
10[11,9]
4[9*(7),3]
11[10,12]
3[2,4]
12[2,11]
2[-1,1]
1[0,-1]
0[-1,0]
3#
2
1
0
S1
3
4
9
10
11
12
7
8
S2
Powered by yFiles

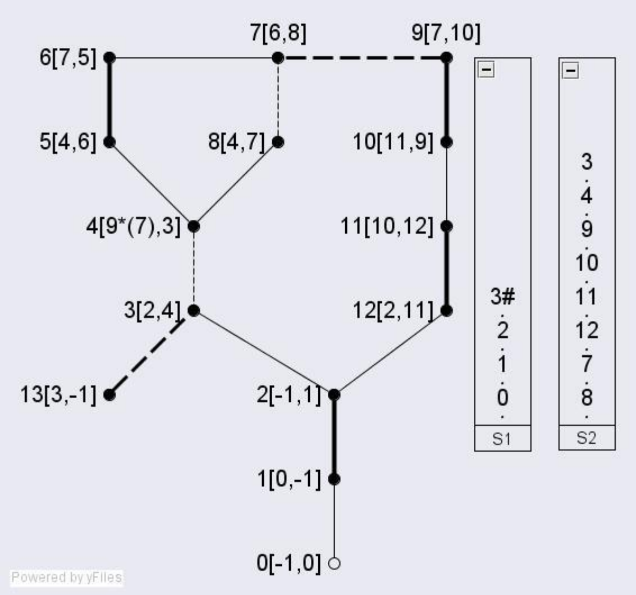

7[6,8]
9[7,10]
6[7,5]
5[4,6]
8[4,7]
10[11,9]
4[9*(7),3]
11[10,12]
3[2,4]
12[2,11]
13[3,-1]
2[-1,1]
1[0,-1]
0[-1,0]
3#
2
1
0
S1
3
4
9
10
11
12
7
8
S2
Powered by yFiles

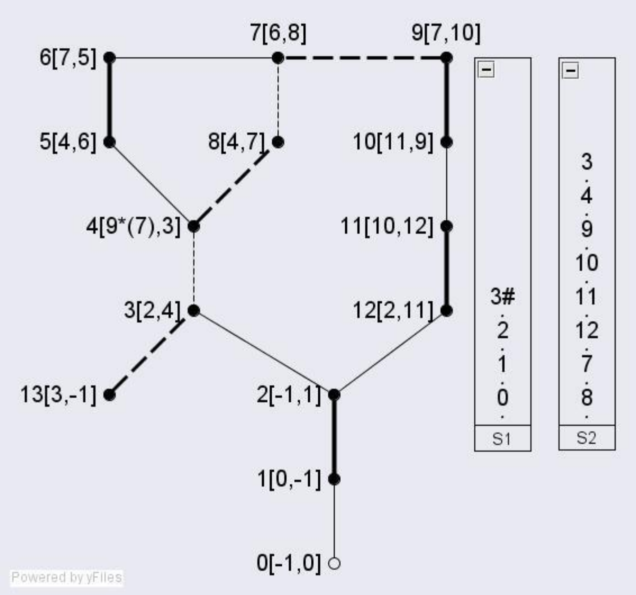

7[6,8]
9[7,10]
6[7,5]
5[4,6]
8[4,7]
10[11,9]
4[9*(7),3]
11[10,12]
3[2,4]
12[2,11]
13[3,-1]
2[-1,1]
1[0,-1]
0[-1,0]
3#
2
1
0
S1
3
4
9
10
11
12
7
8
S2
Powered by yFiles

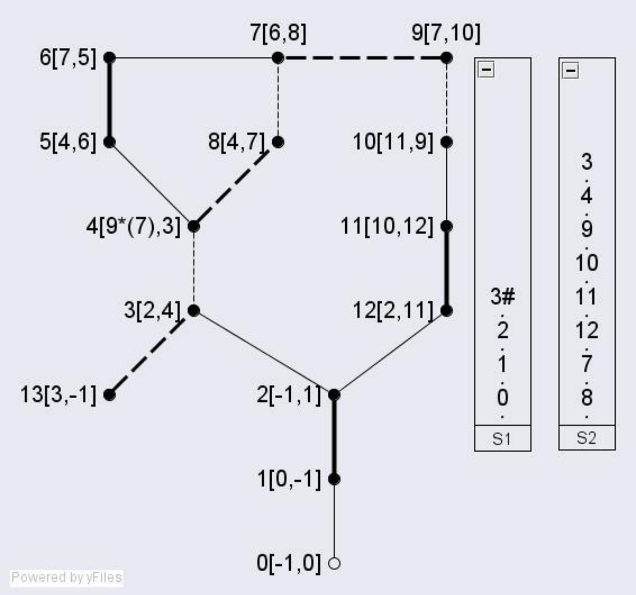


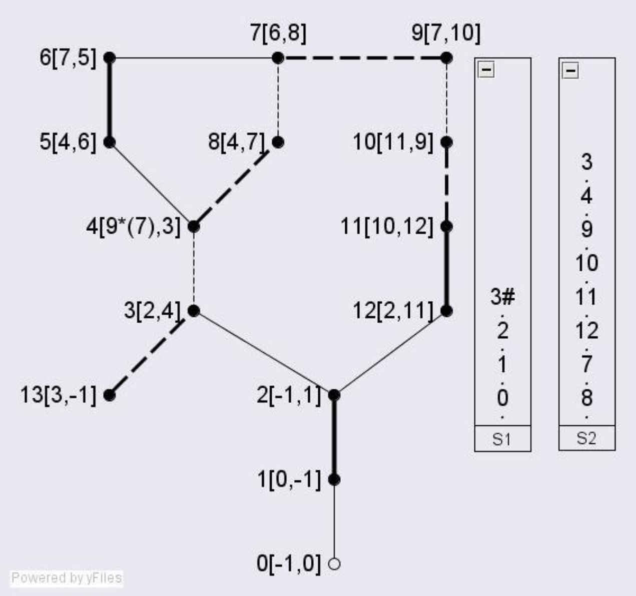
7[6,8]
9[7,10]
6[7,5]
5[4,6]
8[4,7]
10[11,9]
4[9*(7),3]
11[10,12]
3[2,4]
12[2,11]
13[3,-1]
2[-1,1]
1[0,-1]
0[-1,0]
3#
2
1
0
S1
3
4
9
10
11
12
7
8
S2
Powered by yFiles

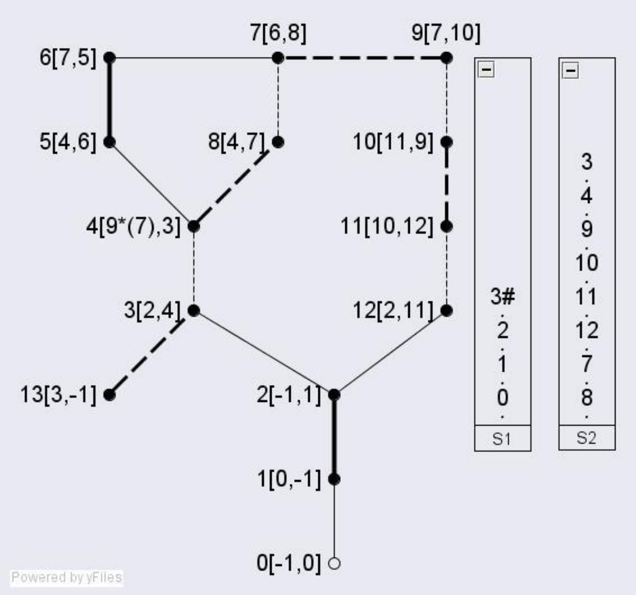

6[7,5]
7[6,8]
9[7,10]
5[4,6]
8[4,7]
10[11,9]
4[9*(7),3]
11[10,12]
3[2,4]
12[2,11]
13[3,-1]
2[-1,1]
1[0,-1]
0[-1,0]
3#
2
1
0
S1
3
4
9
10
11
12
7
8
S2
Powered by yFiles

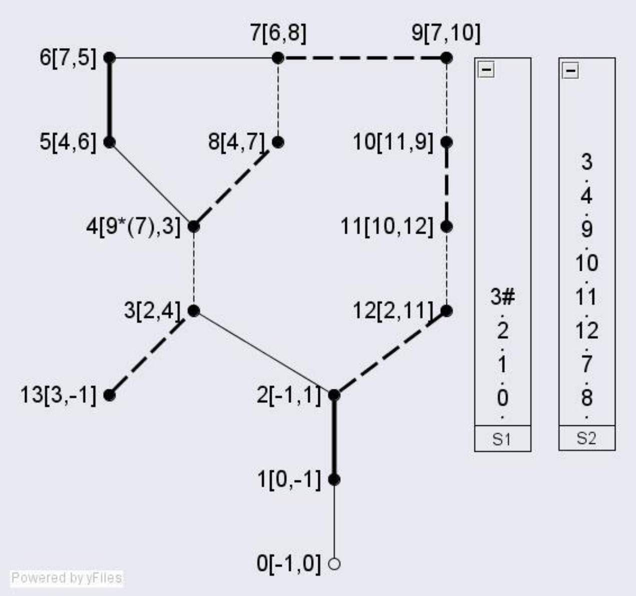

7[6,8]
9[7,10]
6[7,5]
5[4,6]
8[4,7]
10[11,9]
4[9*(7),3]
11[10,12]
3[2,4]
12[2,11]
13[3,-1]
2[-1,1]
1[0,-1]
0[-1,0]
3#
2
1
0
S1
3
4
9
10
11
12
7
8
S2
Powered by yFiles

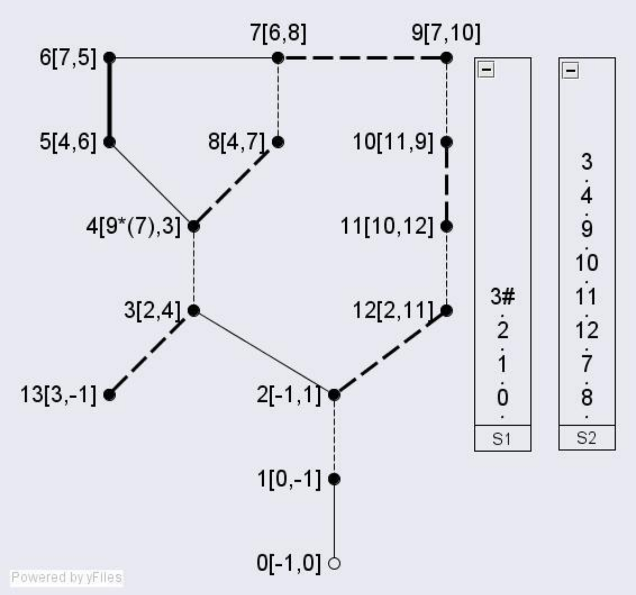

7[6,8]
9[7,10]
6[7,5]
5[4,6]
8[4,7]
10[11,9]
4[9*(7),3]
11[10,12]
3[2,4]
12[2,11]
13[3,-1]
2[-1,1]
1[0,-1]
0[-1,0]
3#
2
1
0
S1
3
4
9
10
11
12
7
8
S2
Powered by yFiles

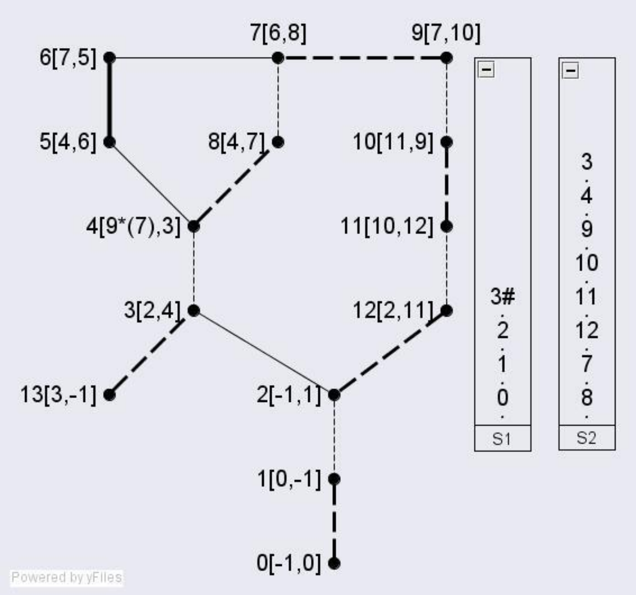

7[6,8]
9[7,10]
6[7,5]
5[4,6]
8[4,7]
10[11,9]
4[9*(7),3]
11[10,12]
3[2,4]
12[2,11]
13[3,-1]
2[-1,1]
1[0,-1]
0[-1,0]
3#
2
1
0
S1
3
4
9
10
11
12
7
8
S2
Powered by yFiles


7[6,8]
9[7,10]
6[7,5]
5[4,6]
8[4,7]
10[11,9]
4[9*(7),3]
11[10,12]
3[2,4]
12[2,11]
13[3,-1]
2[-1,1]
1[0,-1]
0[-1,0]
3#
2
1
0
S1
3
4
9
10
11
12
7
8
S2
Powered by yFiles


7[6,8]
9[7,10]
6[7,5]
5[4,6]
8[4,7]
10[11,9]
4[9*(7),3]
11[10,12]
3[2,4]
12[2,11]
13[3,-1]
2[-1,1]
1[0,-1]
0[-1,0]
3#
2
1
0
S1
3
4
9
10
11
12
7
8
S2
Powered by yFiles


7[6,8]
9[7,10]
6[7,5]
5[4,6]
8[4,7]
10[11,9]
4[9*(7),3]
11[10,12]
3[2,4]
12[2,11]
13[3,-1]
2[-1,1]
1[0,-1]
0[-1,0]
3#
2
1
0
S1
3
4
9
10
11
12
7
8
S2
Powered by yFiles


7[6,8]
9[7,10]
6[7,5]
5[4,6]
8[4,7]
10[11,9]
4[9*(7),3]
11[10,12]
3[2,4]
12[2,11]
13[3,-1]
2[-1,1]
1[0,-1]
0[-1,0]
3#
2
1
0
S1
3
4
9
10
11
12
7
8
S2
Powered by yFiles


7[6,8]
9[7,10]
6[7,5]
5[4,6]
8[4,7]
10[11,9]
4[9*(7),3]
11[10,12]
3[2,4]
12[2,11]
13[3,-1]
2[-1,1]
1[0,-1]
0[-1,0]
3#
2
1
0
S1
3
4
9
10
11
12
7
8
S2
Powered by yFiles

Resultados

Teorema Kameda e Munro 1974

Um emparelhamento máximo pode ser encontrado no tempo $c_1nm + c_2$, onde c_1 e c_2 são constantes.

Teorema 2 (Hall)

Seja G um grafo bipartido com bipartição (X, Y). Então G admite um emparelhamento que satura todos os vértices de X se e somente se $|N(S)| \geq |S|$ para todo $S \subseteq X$.

Notação

A vizinhança de um vértice v, escrita $N(v)$, é o conjunto de todos os vértices adjacentes a v. $N(S) = \bigcup_{v \in S} N(v)$.

Resultados

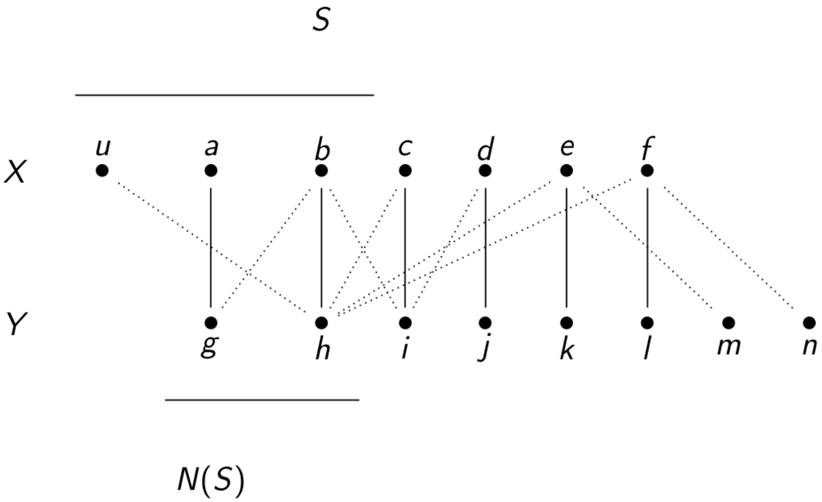


Figura: Aplicação do Teorema de Hall

Corolário 1

Se G é um grafo bipartido k-regular com $k > 0$, então G tem um emparelhamento perfeito.

Demonstração.

Seja G um grafo bipartido k-regular com bipartição (X, Y). Desde que G é k-regular, temos que $k|X| = |E| = k|Y|$ e desde que $k > 0$, $|X| = |Y|$. Seja $S \subseteq X$ e denote por E_1 e E_2 os conjuntos de arestas incidentes à S e a $N(S)$, respectivamente. Pela definição de $N(S)$, temos que $E_1 \subseteq E_2$ e

$$k|N(S)| = |E_2| \geq |E_1| = k|S|.$$

Logo $|N(S)| \geq |S|$. Pelo Teorema 2, temos que G tem um emparelhamento M saturando todos os vértices de X . Portanto, M é um emparelhamento perfeito, desde que $|X| = |Y|$.

Resultados

Observação:

Para algum emparelhamento M e alguma cobertura K temos que

$$|M| \leq |K|.$$

Teorema 3

Seja M um emparelhamento e K uma cobertura tal que $|M| = |K|$. Então M é um emparelhamento máximo e K é uma cobertura mínima.

Teorema 4

Em um grafo bipartido o número de arestas em um emparelhamento máximo é igual o número de vértices em uma cobertura mínima.

Resultados

Definição 1

Uma componente de um grafo é **ímpar/par** se possui um número ímpar/par de vértices. Denotaremos por $i(G)$ o número de componentes ímpares de um grafo G.

Teorema 5 (Tutte, 1947)

G tem um emparelhamento perfeito se e somente se $i(G - S) \leq |S|$, para todo $S \subseteq V(G)$

Referências

- CORMEN, T H. et al. Algoritmos: teoria e prática. Rio de Janeiro: Editora Campus. 3ª Edição. 2012.
- GOLDBARG, E. et al. Grafos - Conceitos, Algoritmos e Aplicações. Elsevier - Campus. 1ª Edição. 2012.
- NETTO, P O B. Grafos - Teorias, Modelos, Algoritmos. Blucher. 5ª Edição. 2012.
- GIBBONS, A. Algorithmic Graph Theory. Cambridge University Press. 1994.
- NETTO, P O B. Grafos - Introdução e Prática. Edgard Blucher. 1ª Edição. 2009.
- SZWARCFITER, J L. Grafos e Algoritmos Computacionais. Editora tem Campus.1984.
- WEST, D. Introduction to Graph Theory. Prentice Hall. 2000.
- YELENN, J; GROSS, J. Graph Theory and Its Applications. CRC Press. 1998.

Muito Obrigado!!!

Contato: andre.cunha@ifgoiano.edu.br