1 Terminology

Directed tree: A directed graph that would be a tree if we ingored the directions of the edges.

Arborescence: A directed tree with a "root" such that every node of it has a unique path to it from the root. There's always exactly one root.

Spanning subgraph of $D(V, A)$: A subgraph $D(V, B)$ such that $B \subseteq A$.

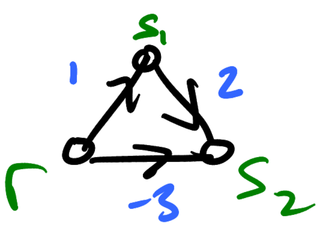
Branching system of $D(G, A)$ with k specified root nodes: A spanning subgraph of D created by edge-disjoint arborescences rooted at the k root nodes. Edge-disjoint means none of these arborescences share any edges. Every node has at most k incoming edges (at most one from each arborescence).

Optimal branching system of a weighted digraph with k specified root nodes: the branching system with the lowest total weight.

2 Examples:

2.1 3-nodes

2.1.1 1 root

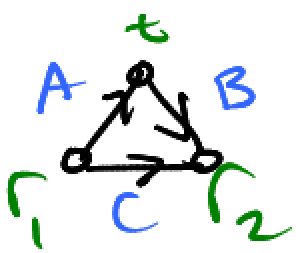


The branching systems are:

$$B_1 = \{r, s_1, s_2\}, c(B_1) = 0 \tag{1}$$
$$B_2 = \{rs_2, rs_1\}, c(B_2) = -2 \tag{2}$$
$$B_3 = \{rs_1, s_1s_2\}, c(B_3) = 3 \tag{3}$$
$$B_4 = \{rs_1, s_2\}, c(B_4) = 1 \tag{4}$$
$$B_5 = \{rs_2, s_1\}, c(B_5) = -3. \tag{5}$$

B_5 is the optimal branching system!

2.1.2 2 roots



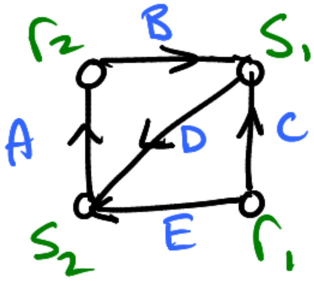
$$c\left(\begin{bmatrix}A\\B\\C\end{bmatrix}\right) = \begin{bmatrix}1\\2\\-3\end{bmatrix}$$

The branching systems are:

$$B_1 = \{r, r_2, t\}, c(B_1) = 0 \tag{6}$$
$$B_2 = \{A, r_2\}, c(B_2) = 1 \tag{7}$$

B_1 is the optimal branching system! If r_2 wasn't a root then we would also be allowed $B_3 = \{A, C\}$ and $B_4 = \{A, B\}$, leading to B_3 being the OBS.

2.2 4-nodes:

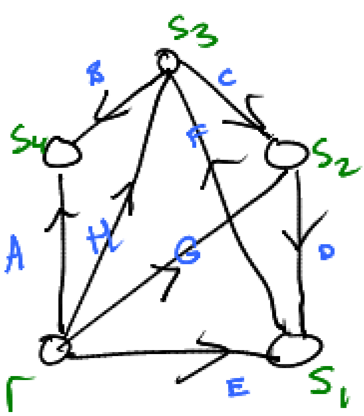


$$C\begin{pmatrix} A \\ B \\ C \\ D \\ E \end{pmatrix} = \begin{bmatrix} 1 \\ 2 \\ -3 \\ -1 \\ 6 \end{bmatrix}$$

The branching systems are:

$$B_1 = \{r_1, r_2, s_1, s_2\}, c(B_1) = 0 \tag{8}$$
$$B_2 = \{C, E, B\}, c(B_2) = \tag{9}$$
$$B_3 = \{CD, r_2\}, c(B_3) = \tag{10}$$
$$B_4 = \{C, BD\}, c(B_4) = \tag{11}$$
$$B_5 = \{C, r_2, s_2\}, c(B_5) = \tag{12}$$
$$B_6 = \{E, r_2, s_1\}, c(B_6) = \tag{13}$$
$$B_7 = \{BD, r_1\}, c(B_7) = \tag{14}$$
$$B_8 = \{B, r_1, r_2\}, c(B_8) = \tag{15}$$
$$B_9 = \{BD, E\}, c(B_9) = \tag{16}$$
$$B_{10} = \{BD, C\}, c(B_{10}) = \tag{17}$$
$$B_{11} = \{BD, E, C\}, c(B_{11}) = \tag{18}$$
$$B_{12} = \{E, CD, B\}, c(B_{12}) = \tag{19}$$
$$B_{13} = \{B, C, s_2\}, c(B_{13}) = \tag{20}$$
$$B_{14} = \{E, CD, r_2\}, c(B_{14}) = \tag{21}$$

2.3 5-nodes:



$$C\begin{pmatrix} A \\ B \\ C \\ D \\ E \\ F \\ G \\ H \end{pmatrix} = \begin{bmatrix} 1 \\ -2 \\ 3 \\ 4 \\ 4 \\ -3 \\ 2 \\ 1 \end{bmatrix}$$

The branching systems are:

3 Algorithm

Finding an OBS entails finding a min-cost set D of edges which is both a basis of matroid M_1 and of M_2 (optimum matroid intersection problem).

- $M_1 = (E, J_1)$ in which J_1 has no edges entering node r, and at most (at most? but corollary said "exactly") k edges entering each other node of the graph.
- $M_2 = (E, J_2)$ in which J_2 can be partitioned into at most k forests.

So all we have to do is (after converting the weighted directed graph into a weighted incidence matrix):

- Find forests in the graph, by using Gaussian elimination in GF(2) to find sets of columns which are linearly independent. This gives us J_2.
- Construct J_1 (but don't necessarily store all of its elements).
- Use weighted matroid intersection to find the $J \in J_1 \cap J_2$ with the smallest total weight. Only consider elements of J_1 that will also be in J_2.