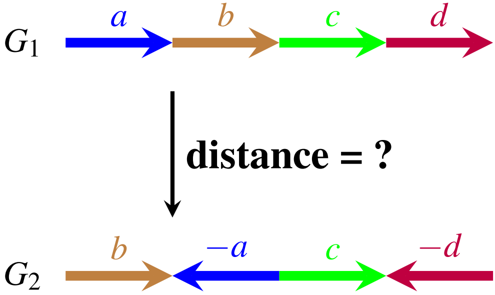
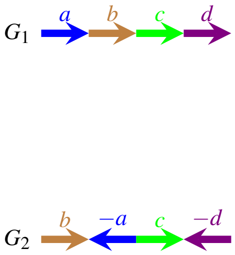
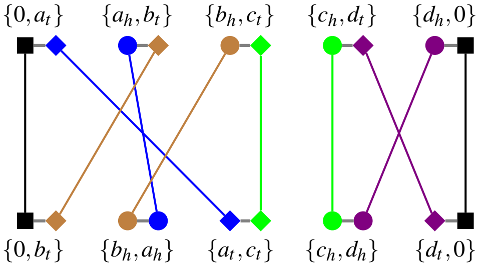
Genome Rearrangement Distance

- The minimum number of operations to transform G_1 into G_2

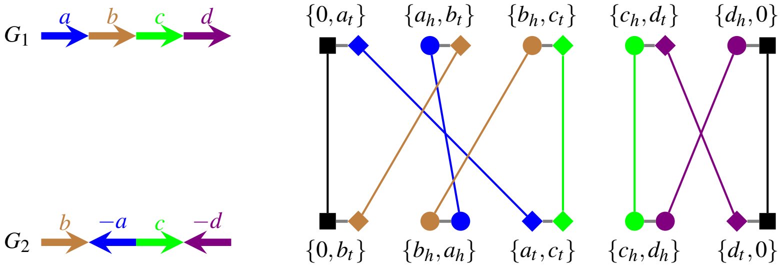


Adjacency Graph



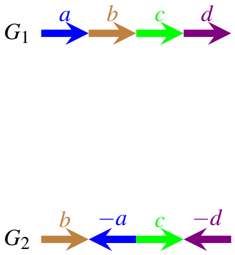


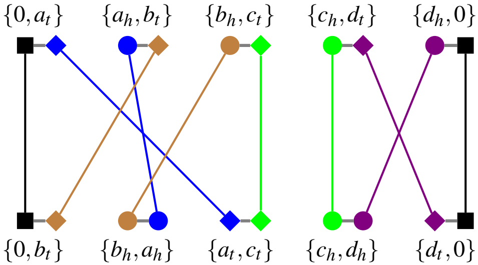
Adjacency Graph



- DCJ distance = (#adjacencies) − (#cycles).

Adjacency Graph

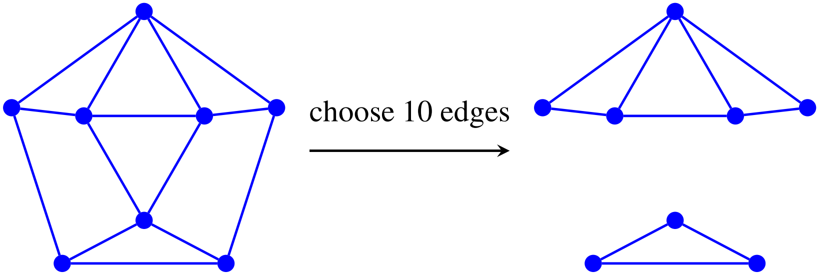




- DCJ distance = (#adjacencies) − (#cycles).
- To minimize DCJ distance, we need to compute a decomposition of the corresponding adjacency graph with maximized number of cycles.

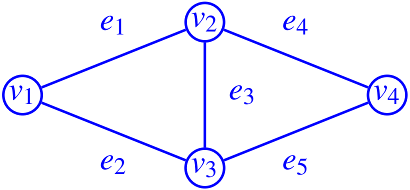
Problem Statement

Problem: given an undirected graph $G = (V, E)$, to choose k edges and remove others, such that the number of connected components in the remaining graph is maximized.
(**Formulate this problem as an ILP**.)



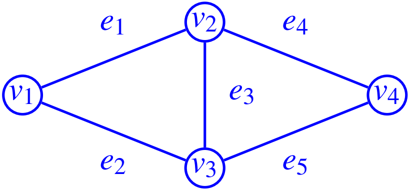
ILP Formulation

Consider the following example with $k = 3$.



ILP Formulation

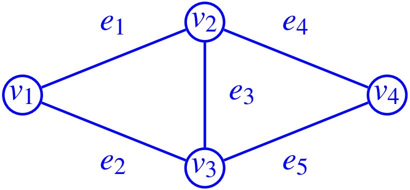
Consider the following example with $k = 3$.



- For each edge e_i, we use a binary variable x_i to indicate whether e_i is chosen. We use the following constraint to guarantee exactly k edges are chosen:

$$x_1 + x_2 + x_3 + x_4 + x_5 = 3$$

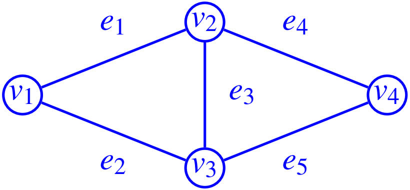
ILP Formulation



- To count the number of connected components, for vertex v_j, $1 \leq j \leq |V|$, we use a variable y_j to indicate the **label** of v_j, and set **distinct** upper bounds for all the labels:

$$1 \leq y_1 \leq 1$$
$$1 \leq y_2 \leq 2$$
$$1 \leq y_3 \leq 3$$
$$1 \leq y_4 \leq 4$$

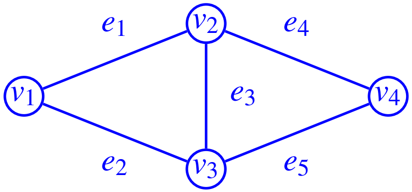
ILP Formulation



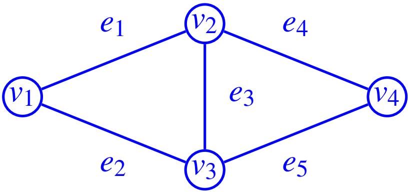
- We guarantee that if an edge is chosen, then its two adjacent vertices have the same label:

$$y_1 \le y_2 + 1 \cdot (1 - x_1);\ \ y_2 \le y_1 + 2 \cdot (1 - x_1) \qquad \text{(for } e_1\text{)}$$
$$y_1 \le y_3 + 1 \cdot (1 - x_2);\ \ y_3 \le y_1 + 3 \cdot (1 - x_2) \qquad \text{(for } e_2\text{)}$$
$$y_2 \le y_3 + 2 \cdot (1 - x_3);\ \ y_3 \le y_2 + 3 \cdot (1 - x_3) \qquad \text{(for } e_3\text{)}$$
$$y_2 \le y_4 + 2 \cdot (1 - x_4);\ \ y_4 \le y_2 + 4 \cdot (1 - x_4) \qquad \text{(for } e_4\text{)}$$
$$y_3 \le y_4 + 3 \cdot (1 - x_5);\ \ y_4 \le y_3 + 4 \cdot (1 - x_5) \qquad \text{(for } e_5\text{)}$$

ILP Formulation

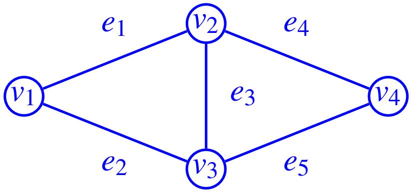


ILP Formulation



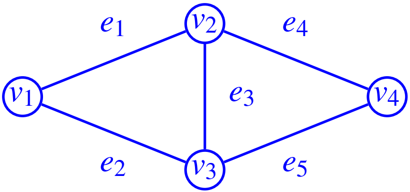
- The equality can propagate along the chosen edges. Thus, in the remaining graph all vertices in the same connected component have the same label.

ILP Formulation

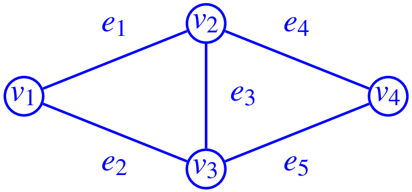


- The equality can propagate along the chosen edges. Thus, in the remaining graph all vertices in the same connected component have the same label.
- Since all vertices have distinct upper bounds, in each connected component, at most one vertex can reach its upper bound. Thus, we can use the number of vertices whose upper bound is reached, to count the number of connected components.

ILP Formulation



ILP Formulation

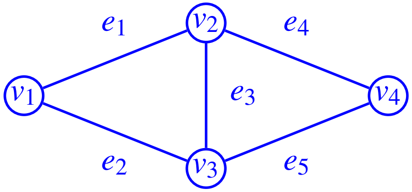


- We use a binary variable z_j to indicate whether the label of v_j reaches its upper bound:

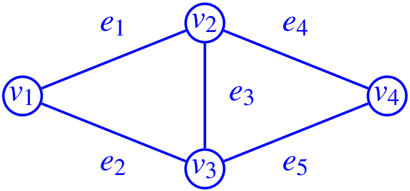
$$1 \cdot z_1 \le y_1$$
$$2 \cdot z_2 \le y_2$$
$$3 \cdot z_3 \le y_3$$
$$4 \cdot z_4 \le y_4$$

We can verify that, $z_j = 1$ only if $y_j = j$, i.e., the label of v_j reaches its upper bound.

ILP Formulation



ILP Formulation



- The objective function of the ILP formulation can be set to maximize the number of vertices whose upper bound can be reached:

$$\max z_1 + z_2 + z_3 + z_4$$