

A crucial ingredient for the design and analysis of approximation algorithms is a technique to obtain an upper bound (for maximization problems) or a lower bound (for minimization problems).

Therefore **Linear Programs** or **Integer Linear Programs** play a vital role in the design of many approximation algorithms.

A crucial ingredient for the design and analysis of approximation algorithms is a technique to obtain an upper bound (for maximization problems) or a lower bound (for minimization problems).

Therefore **Linear Programs** or **Integer Linear Programs** play a vital role in the design of many approximation algorithms.

## Definition 65

An **Integer Linear Program** or **Integer Program** is a Linear Program in which all variables are required to be integral.

## Definition 66

A **Mixed Integer Program** is a Linear Program in which a subset of the variables are required to be integral.

### Definition 65

An **Integer Linear Program** or **Integer Program** is a Linear Program in which all variables are required to be integral.

### Definition 66

A **Mixed Integer Program** is a Linear Program in which a subset of the variables are required to be integral.

Many important combinatorial optimization problems can be formulated in the form of an Integer Program.

Note that solving Integer Programs in general is NP-complete!

Many important combinatorial optimization problems can be formulated in the form of an Integer Program.

**Note that solving Integer Programs in general is NP-complete!**

# Set Cover

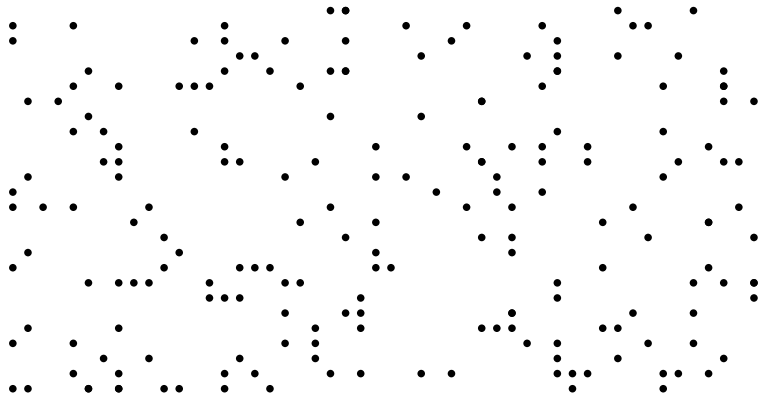
Given a ground set  $U$ , a collection of subsets  $S_1, \dots, S_k \subseteq U$ , where the  $i$ -th subset  $S_i$  has weight/cost  $w_i$ . Find a collection  $I \subseteq \{1, \dots, k\}$  such that

$$\forall u \in U \exists i \in I: u \in S_i \text{ (every element is covered)}$$

and

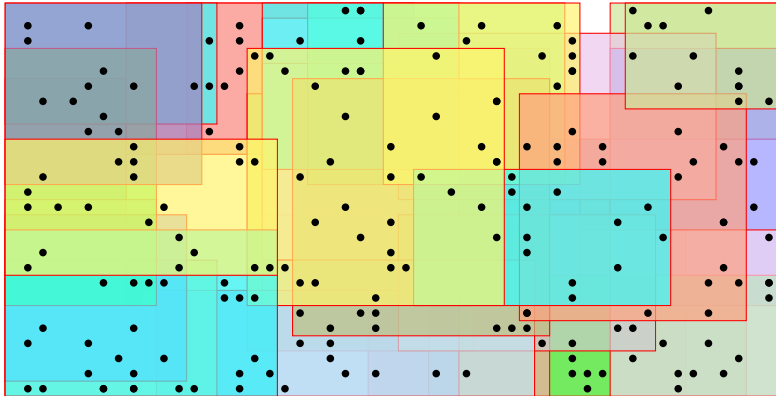
$$\sum_{i \in I} w_i \text{ is minimized.}$$

# Set Cover

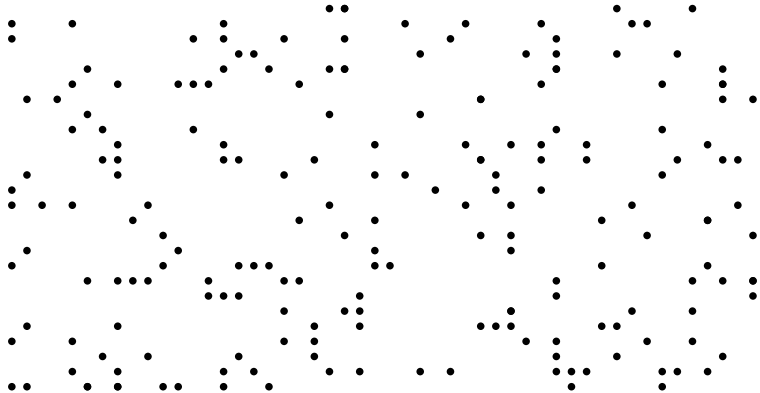




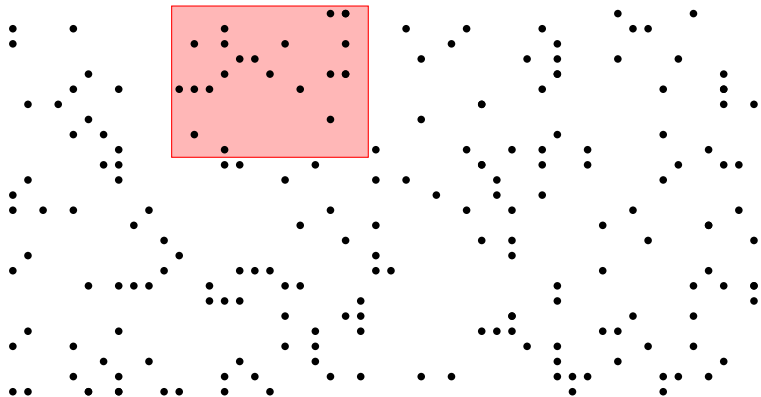
# Set Cover



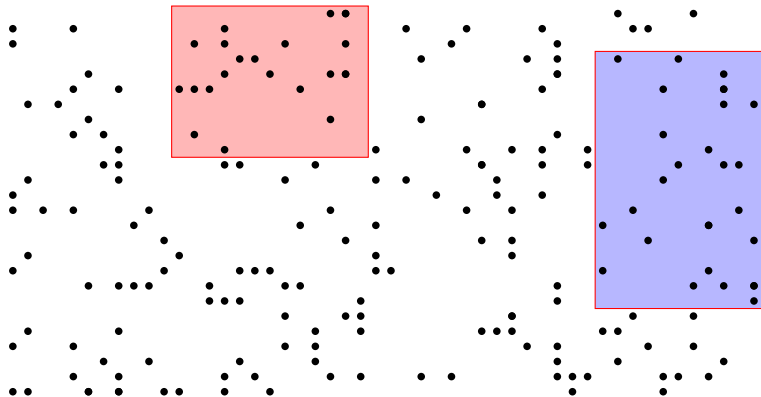
# Set Cover



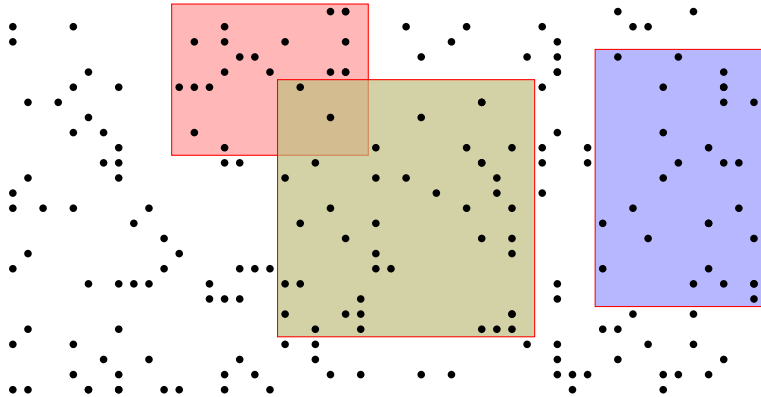
# Set Cover



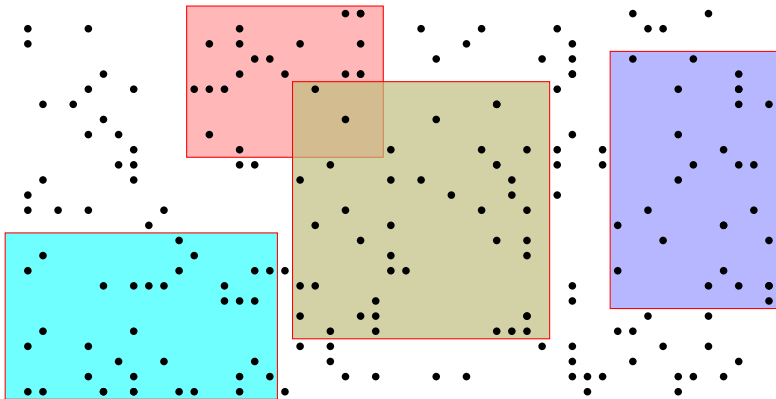
# Set Cover



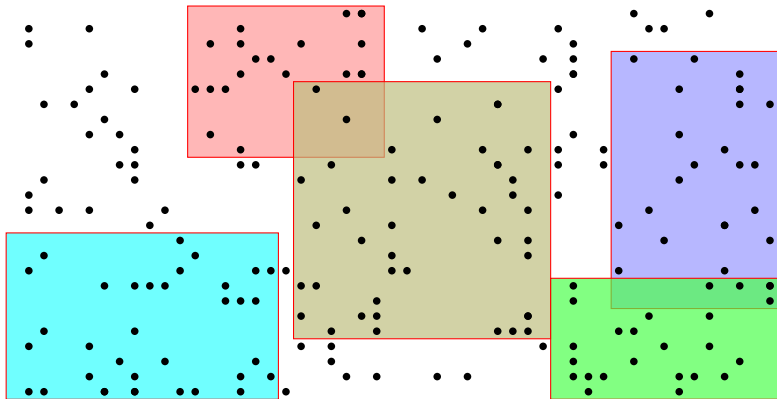
# Set Cover



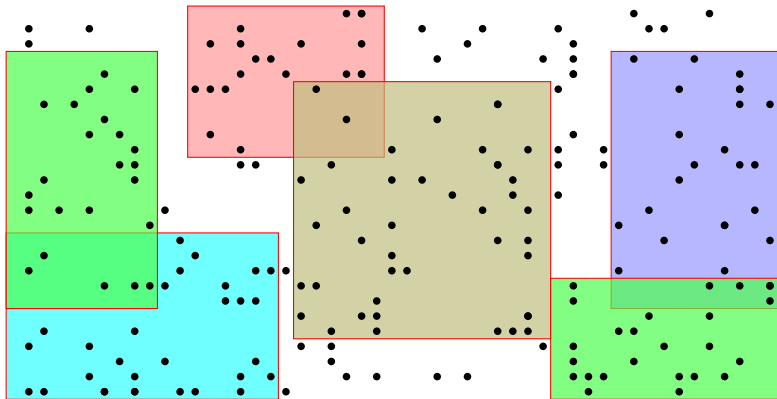
# Set Cover



# Set Cover

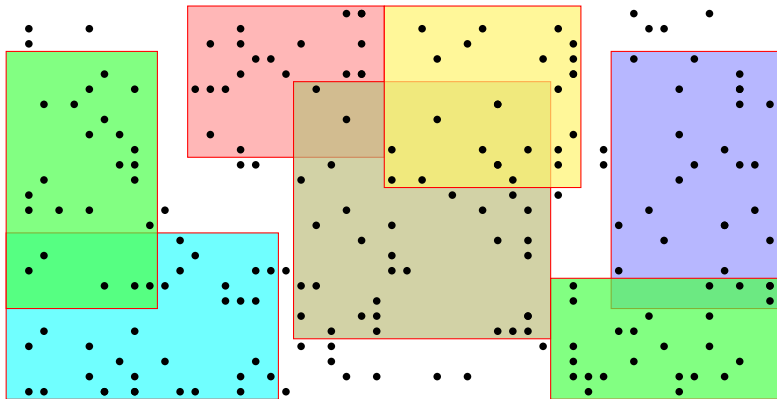


# Set Cover

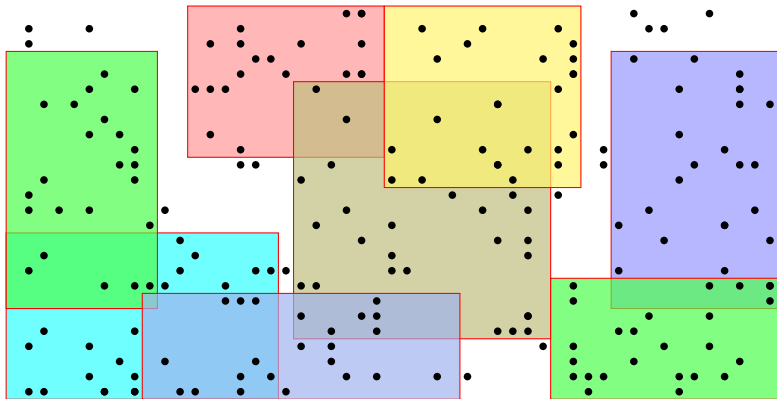




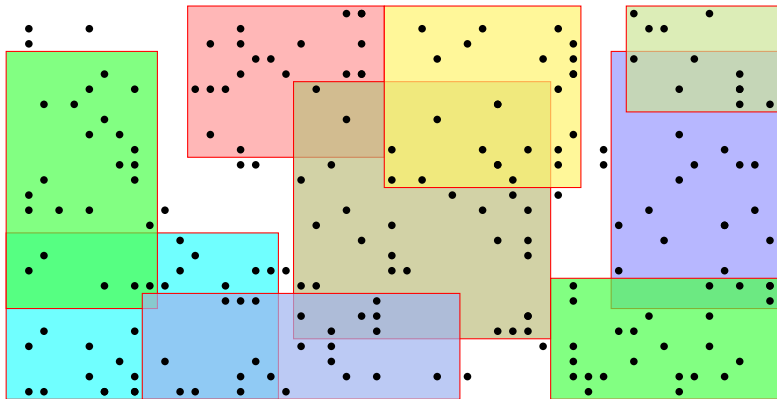
# Set Cover



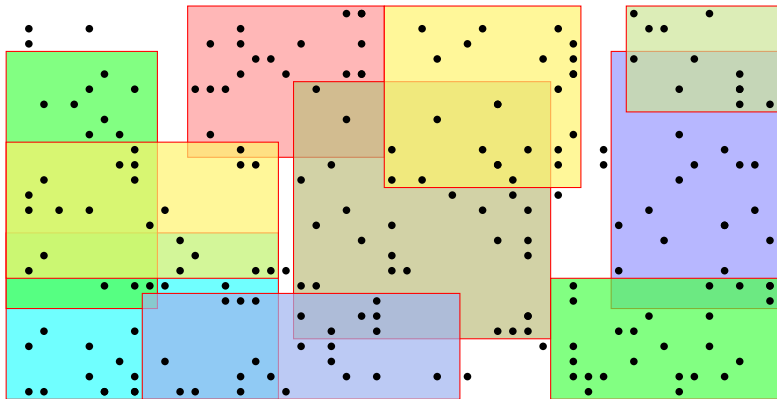
# Set Cover



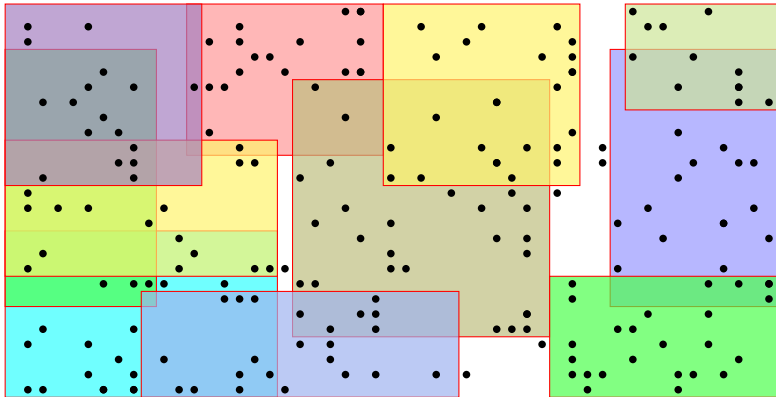
# Set Cover



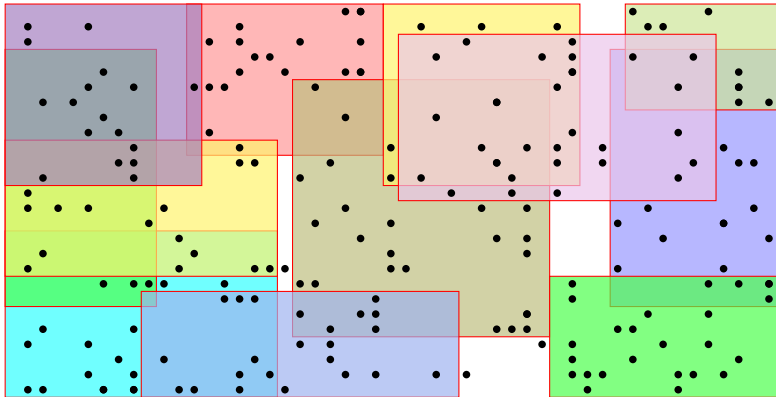
# Set Cover



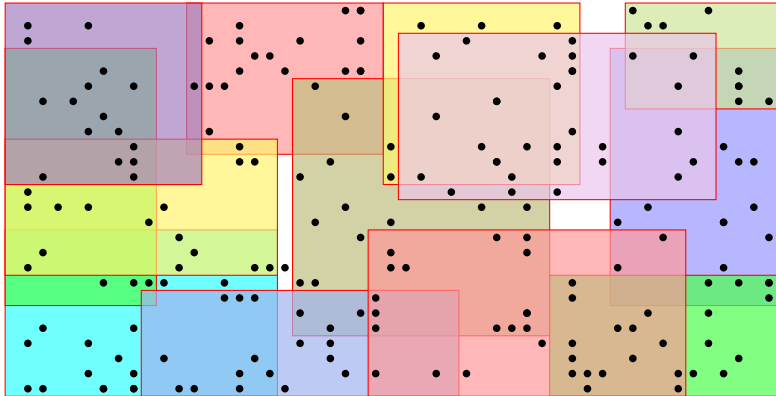
# Set Cover



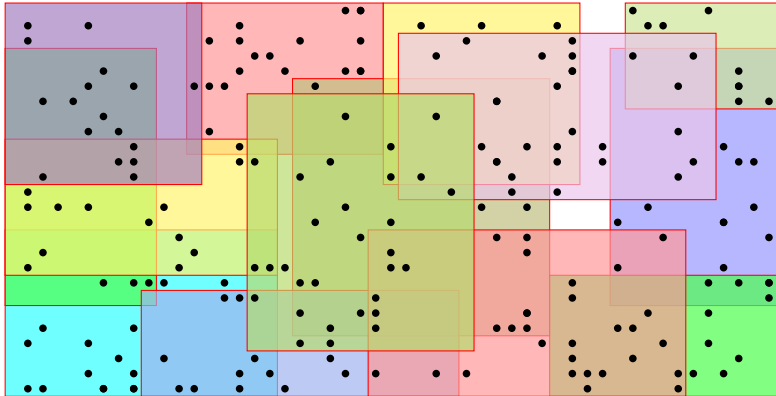
# Set Cover



# Set Cover



# Set Cover





# IP-Formulation of Set Cover

$$\begin{array}{llll} \min & & \sum_i w_i x_i & \\ \text{s.t.} & \forall u \in U & \sum_{i:u \in S_i} x_i & \geq 1 \\ & \forall i \in \{1, \dots, k\} & x_i & \geq 0 \\ & \forall i \in \{1, \dots, k\} & x_i & \text{integral} \end{array}$$

# Vertex Cover

Given a graph  $G = (V, E)$  and a weight  $w_v$  for every node. Find a vertex subset  $S \subseteq V$  of minimum weight such that every edge is incident to at least one vertex in  $S$ .

# IP-Formulation of Vertex Cover

$$\begin{array}{ll} \min & \sum_{v \in V} w_v x_v \\ \text{s.t.} & \forall e = (i, j) \in E \quad x_i + x_j \geq 1 \\ & \forall v \in V \quad x_v \in \{0, 1\} \end{array}$$

# Maximum Weighted Matching

Given a graph  $G = (V, E)$ , and a weight  $w_e$  for every edge  $e \in E$ . Find a subset of edges of maximum weight such that no vertex is incident to more than one edge.

$$\begin{array}{ll} \max & \sum_{e \in E} w_e x_e \\ \text{s.t.} & \forall v \in V \quad \sum_{e: v \in e} x_e \leq 1 \\ & \forall e \in E \quad x_e \in \{0, 1\} \end{array}$$

# Maximum Weighted Matching

Given a graph  $G = (V, E)$ , and a weight  $w_e$  for every edge  $e \in E$ . Find a subset of edges of maximum weight such that no vertex is incident to more than one edge.

$$\begin{array}{ll} \max & \sum_{e \in E} w_e x_e \\ \text{s.t.} & \forall v \in V \quad \sum_{e: v \in e} x_e \leq 1 \\ & \forall e \in E \quad x_e \in \{0, 1\} \end{array}$$

# Maximum Independent Set

Given a graph  $G = (V, E)$ , and a weight  $w_v$  for every node  $v \in V$ . Find a subset  $S \subseteq V$  of nodes of maximum weight such that no two vertices in  $S$  are adjacent.

$$\begin{array}{ll} \max & \sum_{v \in V} w_v x_v \\ \text{s.t.} & \forall e = (i, j) \in E \quad x_i + x_j \leq 1 \\ & \forall v \in V \quad x_v \in \{0, 1\} \end{array}$$

# Maximum Independent Set

Given a graph  $G = (V, E)$ , and a weight  $w_v$  for every node  $v \in V$ . Find a subset  $S \subseteq V$  of nodes of maximum weight such that no two vertices in  $S$  are adjacent.

$$\begin{array}{ll} \max & \sum_{v \in V} w_v x_v \\ \text{s.t.} & \forall e = (i, j) \in E \quad x_i + x_j \leq 1 \\ & \forall v \in V \quad x_v \in \{0, 1\} \end{array}$$

# Knapsack

Given a set of items  $\{1, \dots, n\}$ , where the  $i$ -th item has weight  $w_i$  and profit  $p_i$ , and given a threshold  $K$ . Find a subset  $I \subseteq \{1, \dots, n\}$  of items of total weight at most  $K$  such that the profit is maximized.

$$\begin{array}{ll} \max & \sum_{i=1}^n p_i x_i \\ \text{s.t.} & \sum_{i=1}^n w_i x_i \leq K \\ & \forall i \in \{1, \dots, n\} \quad x_i \in \{0, 1\} \end{array}$$



# Knapsack

Given a set of items  $\{1, \dots, n\}$ , where the  $i$ -th item has weight  $w_i$  and profit  $p_i$ , and given a threshold  $K$ . Find a subset  $I \subseteq \{1, \dots, n\}$  of items of total weight at most  $K$  such that the profit is maximized.

$$\begin{array}{ll} \max & \sum_{i=1}^n p_i x_i \\ \text{s.t.} & \sum_{i=1}^n w_i x_i \leq K \\ & \forall i \in \{1, \dots, n\} \quad x_i \in \{0, 1\} \end{array}$$

## Definition 67

A linear program LP is a **relaxation** of an integer program IP if any feasible solution for IP is also feasible for LP and if the objective values of these solutions are identical in both programs.

We obtain a relaxation for all examples by writing  $x_i \in [0, 1]$  instead of  $x_i \in \{0, 1\}$ .

## Definition 67

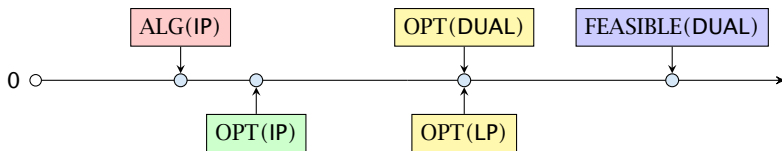
A linear program LP is a **relaxation** of an integer program IP if any feasible solution for IP is also feasible for LP and if the objective values of these solutions are identical in both programs.

We obtain a relaxation for all examples by writing  $x_i \in [0, 1]$  instead of  $x_i \in \{0, 1\}$ .

By solving a relaxation we obtain an upper bound for a maximization problem and a lower bound for a minimization problem.

# Relations

## Maximization Problems:



## Minimization Problems:

