

MINIMUM SPANNING TREE AS AN INTEGER LINEAR PROGRAM

CSCI 4113/6101

INSTRUCTOR: NORBERT ZEH

SEPTEMBER 9, 2025

As we have seen, the single-source shortest paths (SSSP) problem is a problem easily modelled as a linear program. The SSSP problem and the minimum spanning tree (MST) problem seem to have a lot in common: if you compare Dijkstra's and Prim's algorithms for these two problems, then you will notice that they are virtually identical, the only difference being the choice of the priorities for vertices currently in the priority queue. It seems reasonable to expect then that we can develop an LP formulation for the MST problem much like the LP formulation for the SSSP problem we developed in the previous topic. Unfortunately, this is not the case.

An MST of an edge-weighted graph is easily modelled as an *integer* linear program (ILP) though, which is a linear program with the added constraint that variables must be given integer values. The natural ILP formulations of the MST problem have exponentially many constraints. As we discuss in the next topic, this is not really a problem, as there are algorithms that can solve LPs with exponentially many constraints in polynomial time provided the LPs satisfy some conditions that are satisfied by the MST problem. The restriction that the variables can only be assigned integer values is a much bigger problem. As we will discuss in the next topic, even deciding whether an ILP has a feasible solution at all, let alone finding an optimal solution, is NP-hard. Therefore, the ILP formulations of the MST problem we discuss here are not very useful from an algorithmic point of view. We discuss them to introduce integer linear programming using a familiar problem as an example. Later in this course, we will use ILPs to model NP-hard problems. For those problems, the ILP formulation will offer useful insights to obtain approximation algorithms or efficient exponential-time algorithms.

Once again, let us start by reviewing the definition of a minimum spanning tree. Given an edge-weighted connected undirected graph (G, w) , a **minimum spanning tree** (MST) of (G, w) is a tree $T = (V, E')$ with $E' \subseteq E$ and such that every tree $T' = (V, E'')$ with $E'' \subseteq E$ satisfies $w(E'') \geq w(E')$ (see Fig. 1).

PROBLEM 1 (Minimum spanning tree, MST). *For an edge-weighted graph (G, w) , compute a spanning tree T of G of minimum weight.*

As already mentioned, an **integer linear program** (ILP) is just an ordinary linear program with the added constraint that a solution is feasible if it satisfies all the constraints of the ILP and all variables are given integer values. An optimal solution is once again a solution with maximum or minimum objective function value among all feasible solutions.

1 TWO NATURAL ILP FORMULATIONS OF THE MST PROBLEM

Since the MST problem is to choose the edge set E' of the MST from among the edges in E , it is natural to represent the set E' as a set of variables $\{x_e \mid e \in E\}$ such that $x_e = 1$ if $e \in E'$ and $x_e = 0$ if $e \notin E'$. In other words, we impose the constraints that $x_e \in \{0, 1\}$, for all $e \in E$, that is, x_e is an integer between 0 and 1.

The weight of the MST $T = (V, E')$ is then

$$w(T) = \sum_{e \in E} w_e x_e.$$

This is the objective function to be minimized. The constraints of the ILP need to enforce that $T = (V, E')$ is connected and acyclic. As Exer. 2 asks you to show, a graph $T = (V, E')$ is a tree if and only if it has $n - 1$ edges and contains no cycles. Since $T \subseteq G$, any cycle T might contain is a cycle of G . To ensure that T contains no cycles then, all we have to ensure is that T misses at least one edge from every cycle in G . This gives us our first ILP formulation of the MST problem:

$$\begin{aligned} & \text{Minimize } \sum_{e \in E} w_e x_e \\ & \text{s.t. } \sum_{e \in E} x_e = n - 1 \\ & \quad \sum_{e \in C} x_e \leq |C| - 1 \quad \forall \text{ cycle } C \text{ in } G \\ & \quad x_e \in \{0, 1\} \quad \forall e \in E. \end{aligned} \tag{1}$$

Prop. 3 below proves that this is a correct ILP formulation of the MST problem. To prove this proposition, we will use the following lemma, which we will use in many correctness proofs of ILP formulations.

LEMMA 2. *Let P be an (integer) linear program with objective function f , let Q be some problem with objective function g , and let ϕ be a surjective mapping from solutions of P to solutions of Q . Assume further that this mapping ϕ satisfies the following two conditions:*

- (i) *A solution $\hat{x}$ of P is feasible if and only if $\phi(\hat{x})$ is a feasible solution of Q .*
- (ii) *For any two solutions $\hat{x}$ and $\tilde{x}$ of P , we have $f(\hat{x}) \leq f(\tilde{x})$ if and only if $g(\phi(\hat{x})) \leq g(\phi(\tilde{x}))$.*

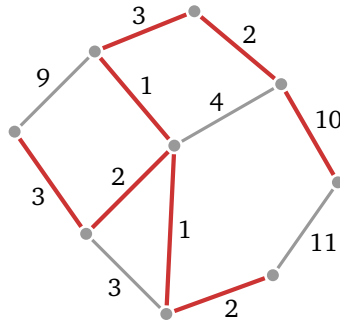


Figure 1: A minimum spanning tree (red) of an edge-weighted graph.

Then a solution $\hat{x}$ of P is an optimal solution of P if and only if $\phi(\hat{x})$ is an optimal solution of Q .

Proof. We assume w.l.o.g. that P and Q are minimization problems. The case when they are both maximization problems is analogous.

Consider an arbitrary feasible solution $\hat{x}$ of P . By (i), $\phi(\hat{x})$ is a feasible solution of Q . To prove that $\hat{x}$ is an optimal solution of P if and only if $\phi(\hat{x})$ is an optimal solution of Q , we prove the contrapositive: $\hat{x}$ is not an optimal solution of P if and only if $\phi(\hat{x})$ is not an optimal solution of Q .

If $\hat{x}$ is not an optimal solution of P , then there exists another feasible solution $\tilde{x}$ of P such that $f(\tilde{x}) < f(\hat{x})$. By (i), $\phi(\tilde{x})$ is a feasible solution of Q . By (ii), $f(\tilde{x}) < f(\hat{x})$ implies that $g(\phi(\tilde{x})) < g(\phi(\hat{x}))$. Therefore, $\phi(\hat{x})$ is not an optimal solution of Q .

Conversely, if $\phi(\hat{x})$ is not an optimal solution of Q , then there exists another feasible solution S of Q such that $g(S) < g(\phi(\hat{x}))$. Since ϕ is surjective, there exists a solution $\tilde{x}$ of P such that $S = \phi(\tilde{x})$. By (i), $\tilde{x}$ is a feasible solution of P . By (ii), $g(\phi(\tilde{x})) < g(\phi(\hat{x}))$ implies that $f(\tilde{x}) < f(\hat{x})$. Therefore, $\hat{x}$ is not an optimal solution of P . $\square$

PROPOSITION 3. *If $\hat{x}$ is an optimal solution of the ILP (1), then the graph $T = (V, E')$, where $E' = \{e \in E \mid \hat{x}_e = 1\}$, is a minimum spanning tree of G .*

Proof. We consider an edge set $E' \subseteq E$ a feasible solution of the MST problem if $T = (V, E')$ is a spanning tree, and an optimal solution if T is a *minimum* spanning tree.

The mapping from solutions $\hat{x}$ of (1) to edge sets $E' = \{e \in E \mid \hat{x}_e = 1\}$ is surjective because every subset $E' \subseteq E$ can be obtained in this fashion from the solution $\hat{x}$ of (1) defined as such that

$$\hat{x}_e = \begin{cases} 1 & \text{if } e \in E' \\ 0 & \text{otherwise.} \end{cases}$$

Moreover, this mapping satisfies the two conditions of Lem. 2:

(i) A solution $\hat{x}$ of (1) is feasible if and only if the corresponding set of edges E' contains $n - 1$ edges and misses at least one edge from every cycle in G . By Exer. 2, this is the case if and only if $T = (V, E')$ is a spanning tree of G .

(ii) The weight of the edge set E' corresponding to a solution $\hat{x}$ of (1) is

$$w(E') = \sum_{e \in E'} w_e = \sum_{e \in E} w_e \hat{x}_e.$$

This implies that for any two solutions $\hat{x}$ and $\tilde{x}$ of (1) and their corresponding edge sets E' and E'' , we have $w(E') \leq w(E'')$ if and only if $\sum_{e \in E} w_e \hat{x}_e \leq \sum_{e \in E} w_e \tilde{x}_e$.

Since the mapping from solutions of (1) to subsets of edges of G satisfies the conditions of Lem. 2, every optimal solution of (1) has the property that the corresponding edge set E' defines a minimum spanning tree $T = (V, E')$ of G . $\square$

A second formulation of the MST problem is also based on the characterization of a tree given in Exer. 2: to be a spanning tree of G , T must have $n - 1$ edges and must be connected. What we need to figure out is how to express that T is connected. To do so, we need the concept of a cut in a graph.

A **cut** in a graph $G = (V, E)$ is a partition of V into two non-empty subsets S and $V \setminus S$, represented by specifying the set $\emptyset \subset S \subset V$ (see Fig. 2). An edge **crosses** the cut S if it has exactly one endpoint in S .

The following lemma uses cuts to characterize connected graphs.

LEMMA 4. *A graph $G = (V, E)$ is connected if and only if every cut in G is crossed by at least one edge in E .*

Proof. We prove the contrapositive: G is disconnected if and only if there exists a cut that is not crossed by any edge.

If G is disconnected, then choose an arbitrary vertex s , and let S be the set of all vertices reachable from s . Since G is disconnected, we have $S \subset V$. Since $s \in S$, we have $S \neq \emptyset$. Thus, S is a cut. If this cut were crossed by an edge (u, v) , then we would have $u \in S$ and $v \notin S$. However, the endpoints of every edge in G are either both reachable from s or neither is. Thus, S is not crossed by any edge.

If there is a cut S that is not crossed by any edge, then let $u \in S$ and $v \in V \setminus S$. Since $\emptyset \subset S \subset V$, two such vertices exist. If there existed a path $P = \langle x_0, \dots, x_k \rangle$ from u to v in G , then note that $x_0 = u \in S$ and $x_k = v \notin S$. Thus, there would exist an edge $(x_i, x_{i+1}) \in P$ such that $x_i \in S$ and $x_{i+1} \notin S$. Since no such edge exists, there is no path from u to v in G , that is, G is disconnected. $\square$

Based on this characterization of a connected graph, we obtain the following ILP formulation of the MST problem:

$$\begin{aligned}
 & \text{Minimize } \sum_{e \in E} w_e x_e \\
 \text{s.t. } & \sum_{e \in E} x_e = n - 1 \\
 & \sum_{e \text{ crosses } S} x_e \geq 1 \quad \forall \text{ cut } S \text{ in } G \\
 & x_e \in \{0, 1\} \quad \forall e \in E.
 \end{aligned} \tag{2}$$

PROPOSITION 5. *If $\hat{x}$ is an optimal solution of the ILP (2), then the graph $T = (V, E')$, where $E' = \{e \in E \mid \hat{x}_e = 1\}$, is a minimum spanning tree of G .*

Proof. As in the proof of Prop. 3, we consider a subset of edges $E' \subseteq E$ a feasible solution of the MST problem if $T = (V, E')$ is a spanning tree of G , and an optimal solution if T is a *minimum* spanning tree.

As observed in the proof of Prop. 3, the mapping from solutions $\hat{x}$ of (2) to subsets $E' = \{e \in E \mid \hat{x}_e = 1\}$ is surjective and since the ILPs (1) and (2) have the same objective function, satisfies condition (ii) of Lem. 2.

A solution $\hat{x}$ of (2) is feasible if and only if the corresponding edge set E' contains $n - 1$ edges and every cut $\emptyset \subset S \subset V$ is crossed by at least one of the edges in E' . By Lem. 4, this is the case if and only if E' contains $n - 1$ edges and the graph $T = (V, E')$ is connected. By Exer. 2, this is the case if and only if T is a spanning tree of G . Thus, the mapping also satisfies condition (i) of Lem. 2. Therefore, a solution

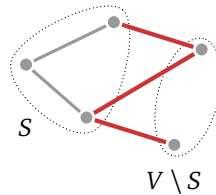


Figure 2: A cut $\emptyset \subset S \subset V$. The red edges cross the cut.

$\hat{x}$ of (2) is optimal if and only if the corresponding edge set E' defines an MST $T = (V, E')$ of G . $\square$

2 EXERCISES

EXERCISE 1. This exercise aims to give you an impression of the power of integer linear programming, compared to ordinary linear programming. In a linear program, it is difficult to express that some variable x_i should be either between 1 and 3 or between 5 and 10 or between 20 and 100; what it isn't allowed to do is to be less than 1 or between 3 and 5 or between 10 and 20 or greater than 100. In contrast, such constraints are easy to express in an integer linear program. This exercise asks you to figure out how. Consider a finite set of integers $\{a_1, b_1, a_2, b_2, \dots, a_k, b_k\}$ with $a_1 \leq b_1 < a_2 \leq b_2 < \dots < a_k \leq b_k$. Express the constraint that $x_i \in \{a_1, \dots, b_1\} \cup \{a_2, \dots, b_2\} \cup \dots \cup \{a_k, \dots, b_k\}$ using only the tools available to you in an integer linear program: linear constraints plus the requirement that all variables have integer values.

Hint: One way to do this is by introducing k helper variables $z_1, \dots, z_k$. You should ensure that each is either 0 or 1 and that exactly one of them is 1. You should then be able to express the condition that if $z_j = 1$, then $a_j \leq x_i \leq b_j$.

EXERCISE 2. Prove that if a graph G on n vertices satisfies two of the following conditions, then it also satisfies the third:

- (a) G has $n - 1$ edges.
- (b) G contains no cycle.
- (c) G is connected.

Prove that any graph that satisfies these three conditions is a tree.

EXERCISE 3. Prove that the MST of an edge-weighted graph (G, w) is unique if $w_{e_1} \neq w_{e_2}$, for all $e_1, e_2 \in E$.

EXERCISE 4. Prove the following claim, often referred to as the **Cut Theorem**: Let (G, w) be an edge-weighted graph and let $\emptyset \subset C \subset V$ be an arbitrary cut of G . Then every MST of (G, w) contains an edge e that crosses the cut C and such that $w_e = \min\{w_f \mid f \text{ crosses } C\}$.