Chair of Combinatorial Optimization
RWTH Aachen University

Bachelor Thesis in Computer Science
March 2026

Min-max and Min-max Regret Spanning Trees under Discrete and Interval Uncertainty

Submitted by
Archit Dhama
Street + Number
Postal Code + City
Matriculation No.: 396580
archit.dhama@rwth-aachen.de

First Examiner: Prof. Dr. Christina Büsing
Chair of Combinatorial Optimization
RWTH Aachen University

Second Examiner: Prof. Dr. Arie M. C. A. Koster
Chair II of Mathematics
RWTH Aachen University

Abstract

This thesis is a self-contained primer on robust spanning trees under cost uncertainty. We formalize and contrast the *Min-Max* and *Min-Max Regret* objectives for discrete scenario sets and interval uncertainty, prove the foundational tools used later (fundamental cut/cycle, MST optimality via cut/cycle criteria), and present a curated synthesis of reported complexity and approximability results. A running micro-graph and a compact TikZ gallery illustrate how robust solutions differ from nominal MSTs.

Contents

Chapter 1

Introduction

1.1 Motivation and Problem Statement

Designing cost–efficient networks is a recurring task in transportation, telecommunications, energy, and logistics. When edge costs are known, the *minimum spanning tree* (MST) is a classical and well-understood baseline: it connects all vertices with minimum total cost. In practice, however, costs are seldom known precisely at design time. They depend on estimates, market prices, delays, outages, or policy changes, which may shift before the network is built or operated. A tree that is optimal for a single nominal estimate can therefore become unexpectedly expensive when the realized costs differ.

This thesis studies spanning trees when edge costs are *uncertain*. We consider two common ways of representing uncertainty that already cover many practical situations: (i) a *discrete set of scenarios*, each a full cost vector; and (ii) *interval ranges* for each edge, forming a Cartesian product of bounds. Given such an uncertainty description, there are (at least) two natural design goals. The first is *Min–Max*: choose a tree that keeps its *worst-case total cost* as small as possible across all scenarios. The second is *Min–Max Regret*: choose a tree whose cost is, in the worst case, close to the *scenario-optimal* MST; that is, we hedge not against absolute cost but against how far we fall short of the best we could have done if the scenario had been known in advance. Both perspectives formalize robustness, yet they prioritize different risks: Min–Max is absolute (guard the total), Regret is relative (guard the performance gap).

Even at this conceptual level, two ingredients are crucial. First, we need a clean and self-contained foundation for MST reasoning—cuts, cycles, and the fundamental exchange arguments—so that all later robust models rest on solid ground. Second, when costs lie in intervals, we must understand *where* worst cases live: Do we have to consider all points in a box, or do extremal cases suffice (e.g., bounds on selected edges)? Answering such questions allows us to reason about robustness without enumerating infinitely many possibilities.

We focus on spanning trees under discrete- and interval-uncertainty and study Min–Max and Min–Max Regret objectives; precise models and notation follow in Chapter 2. We also fix basic complexity and approximation vocabulary early on, enabling concise statements in later chapters.

1.2 Objectives and Scope

This thesis aims to provide a self-contained, correctness-first primer on robust spanning trees that carries the reader from classical MST reasoning to two robust objectives—*Min–Max* and *Min–Max Regret*—under two standard uncertainty styles (discrete scenarios and intervals). The emphasis is on precise modeling, short but complete foundational proofs, and a curated synthesis of established complexity and approximation results, complemented by small worked examples.

Objectives. We pursue the following concrete objectives:

- **Foundational bedrock.** Present the graph/MST basics used later (cuts, cycles, fundamental exchange arguments), with complete proofs and a tiny running example (Chapter 2).
- **Algorithmic preliminaries (complexity & approximation).** A concise primer (decision vs. optimization; P/NP; reductions; weak/strong NP-hardness; approximation ratios; PTAS/FPTAS; APX) so that results in later chapters can be stated tersely; this primer will live in Chapter 2.
- **Unified modeling.** Formalize *Min–Max* and *Min–Max Regret* for spanning trees under (a) *discrete* uncertainty and (b) *interval* uncertainty, using consistent notation across chapters (Chapters 3 and 4).
- **Extremal reasoning for intervals.** State and use short extremal lemmas that identify where worst cases occur within interval boxes, enabling analysis without enumerating infinitely many realizations.
- **Curated landscape.** Collect and organize established results on computational complexity and known approximation (or inapproximability) guarantees for the above models.
- **Illustrative micro-examples.** Use a single small graph throughout to compare behaviors of the models and to make definitions tangible.
- **Synthesis and outlook.** Summarize models and results in a one-page table and outline concise takeaways and avenues for future work (Chapters 5 and 6).

Scope and boundaries. We deliberately restrict attention to the following setting:

- **Problem class.** Undirected, connected graphs with real-valued edge costs; solutions are spanning trees.
- **Uncertainty descriptions.** (i) *Discrete* sets of cost scenarios, i.e., a finite set $\mathcal{U} = \{c^{(1)}, \ldots, c^{(K)}\}$; (ii) *Interval* uncertainty given by per-edge bounds forming a Cartesian product $\prod_e [\,\ell_e, u_e\,]$.
- **Objectives.** *Min–Max* (minimize worst-case total cost) and *Min–Max Regret* (minimize worst-case gap to the scenario-optimal MST). Formal definitions appear in Chapter 2 and are used in Chapters 3 and 4.

A detailed per-chapter roadmap is given in Section 1.4.

1.3 Contributions

This section states what the reader *gets* from this document—concrete deliverables and value added beyond a standard overview.

- **Foundations package (Chapter 2).** A self-contained write-up of the MST toolkit (cut/cycle criteria, fundamental exchange arguments) with complete proofs, a consistent notation bank, and one reusable micro-graph figure.
- **Compact complexity/approximation primer.** A one-page glossary of the terms we use for algorithmic results, included in Chapter 2 to keep Chapters 3–4 focused on model specifics.
- **Interval extremal toolkit.** A compact set of lemmas pinpointing where worst cases occur in interval boxes, with short proofs and guidance on how they are used later for Min–Max and Regret analyses.
- **Normalized notation and cross-referencing.** One consistent symbol set ($\mathcal{U}, \mathcal{T}, \mathrm{MST}((\cdot))$) and a clear cross-reference policy (via `\Cref`) applied throughout, so readers can navigate definitions/results without ambiguity.
- **Organized literature map.** A tabulated map of established complexity and approximation/inapproximability results, indexed by objective (Min–Max / Regret), uncertainty (discrete / interval), and scenario parameter K where relevant.
- **Synthesis artifacts (Chapter 5).** (i) A one-page summary table (models $\times$ uncertainty $\times$ complexity/approx); (ii) a small example gallery (shared micro-graph) illustrating behaviors across objectives.
- **Optional full proof (Appendix A).** One representative hardness proof reproduced in full for pedagogical completeness.
- **Reproducible build and readable sources.** Clean Route A LaTeX build (biblatex/biber, cleveref, TikZ) and a *readable* Markdown export of chapters to facilitate review.

1.4 Thesis Roadmap

This document proceeds from foundations to robust models and, finally, to a concise synthesis.

1. **Chapter 2 (Foundations).** We fix notation, prove the MST toolkit (Fundamental Cut Lemma; cycle/cut criteria), introduce a tiny running graph, and include a short primer on complexity and approximation to support the statements used in later chapters.

2. **Chapter 3 (Min–Max Spanning Tree).** We formalize Min–Max for spanning trees under discrete scenarios and interval uncertainty, and we state the minimal interval extremal facts needed to reason about worst cases. A small worked example illustrates the model.

3. **Chapter 4 (Min–Max Regret Spanning Tree).** We formalize regret and its discrete/interval variants, relate regret to scenario-optimal MSTs, and use short extremal lemmas for intervals. A worked example mirrors Chapter 3 for direct comparison.

4. **Chapter 5 (Synthesis & Example Gallery).** We summarize the established landscape in a one-page table (model $\times$ uncertainty; complexity; (in)approximability; notes on K) and gather a few TikZ micro-examples that highlight behavioral contrasts.

5. **Chapter 6 (Conclusion & Outlook).** We extract the main takeaways, note limitations, and briefly indicate natural extensions (e.g., budgeted-Γ uncertainty or polyhedral sidebars) that lie beyond our core scope.

Appendices. Appendix A (optional) contains one representative hardness/inapproximability proof rendered in our notation. Appendix B lists all symbols in a compact table for quick reference. Appendix C holds auxiliary figures or expanded examples referenced from the main text.

Chapter 2

Foundations

2.1 Graphs, Trees, and Notation

AWU 2.1.1 (Core objects; purpose). Fix a finite, undirected, simple (no loops or parallel edges) graph $G = (V, E)$. Unless stated otherwise, G is connected. For $X \subseteq V$, the *cut* induced by X is $\delta(X) := \{\{u, v\} \in E : u \in X, v \in V \setminus X\}$. A *path* is a sequence of distinct vertices whose consecutive pairs form edges; a *cycle* is a closed path with at least three vertices. A subgraph $T = (V, E(T))$ is a *spanning tree* if it is connected and acyclic. We denote by $\mathcal{T}$ the set of all spanning trees of G. When an edge-weight (cost) vector $c \in \mathbb{R}^E_{\geq 0}$ is fixed, the cost of a tree T under c is $c(T) := \sum_{e \in E(T)} c_e$. The minimum spanning tree (MST) value under c is $\mathrm{MST}(()c) := \min_{T \in \mathcal{T}} c(T)$. All symbols in this section are consistent with the global notation file and will be referenced throughout the thesis.

AWU 2.1.2 (Fundamental structures; purpose). Fix a spanning tree $T \in \mathcal{T}$. For any non-tree edge $f \in E \setminus E(T)$, the graph $T \cup \{f\}$ contains a unique simple cycle; this is the *fundamental cycle* of f with respect to T and is denoted by C_f. For any tree edge $e \in E(T)$, the graph $T \setminus \{e\}$ has exactly two connected components; let $X_e \subset V$ be the vertex set of one such component (the choice is arbitrary but fixed). The set $\delta(X_e)$ is the *fundamental cut* of e with respect to T. By construction, every edge in $\delta(X_e)$ has one endpoint in X_e and the other in $V \setminus X_e$. Fundamental cycles and fundamental cuts are well-defined with respect to T and will be used to formulate and prove optimality criteria for MSTs.

AWU 2.1.3 (Tree basics; purpose). Let T be a spanning tree on G. Then: (i) between any two vertices of T there is a unique simple path; (ii) $|E(T)| = |V| - 1$; (iii) removing any tree edge disconnects T into exactly two components. These facts will be used to justify fundamental cuts/cycles.

Lemma 1 (Fundamental Cycle Lemma)**.** *Let $T \in \mathcal{T}$ and let $f \in E \setminus E(T)$ be a non-tree edge with endpoints u, v. Then the graph $T \cup \{f\}$ contains a unique simple cycle, namely C_f, obtained by adding f to the unique u-v path in T.*

Proof. By Section 2.1, there is a unique simple path in T between u and v. Adding $f = \{u, v\}$ closes exactly one simple cycle consisting of this path plus f. No other cycle can appear because T is acyclic and f was the only added edge. □

2.2 Fundamental Cut Lemma

Lemma 2 (Fundamental Cut Lemma). *Let $T \in \mathcal{T}$ be a spanning tree and let $e \in E(T)$. Let $X_e \subset V$ be a component vertex set of $T \setminus \{e\}$, and let $\delta(X_e)$ be its fundamental cut. Then $E(T) \cap \delta(X_e) = \{e\}$; that is, e is the unique tree edge crossing its fundamental cut.*

Proof. Deleting e from T yields two connected components with vertex sets X_e and $V \setminus X_e$. Any remaining edge of T lies entirely inside one of these components (otherwise T would still be connected after deleting e, contradicting that T is a tree). Hence no edge of $E(T) \setminus \{e\}$ has endpoints in distinct components, i.e., no such edge lies in $\delta(X_e)$. On the other hand, e itself has one endpoint in X_e and the other in $V \setminus X_e$ by construction, so $e \in \delta(X_e)$. Therefore $E(T) \cap \delta(X_e) = \{e\}$. $\square$

Remark 1. The set $\delta(X_e)$ and the property in Lemma 2 depend only on T and the choice of the tree edge e. This lemma will be cited in the proofs of the MST cut and cycle criteria (Theorems 1 to 3). Equal edge weights are permitted; uniqueness in the lemma refers to set membership in $E(T)$, not to minimum or maximum weights.

2.3 MST Optimality: Cycle and Cut Criteria

Remark 2 (Safe/unsafe edges and ties). *Cut property (safe edges).* For any cut, every cheapest edge crossing it is safe to add to some MST. *Cycle property (unsafe edges).* On any cycle, every edge strictly heavier than the minimum-cost edge on that cycle is unsafe for every MST. With ties, statements are understood as "one of the cheapest" or "no strictly cheaper alternative": our theorems use inequalities like $c_e \le c_{e'}$ to cover equal-cost cases. These formulations justify the one-paragraph correctness arguments for Kruskal and Prim in Section 2.4.

Theorem 1 (Cycle Criterion: Necessity). *Let T be an MST under edge costs c. For every non-tree edge $f \in E \setminus E(T)$ with fundamental cycle C_f, every tree edge $e \in E(T) \cap C_f$ satisfies $c_e \le c_f$. In words: no edge of T on C_f is more expensive than f.*

Proof. Suppose there exists $f \notin E(T)$ and $e \in C_f \cap E(T)$ with $c_e > c_f$. Then $T' := T \setminus \{e\} \cup \{f\}$ is a spanning tree (swapping along the fundamental cycle). Its cost is $c(T') = c(T) - c_e + c_f < c(T)$, contradicting optimality of T. The exchange is valid because adding f creates exactly one cycle C_f, and removing any tree edge on that cycle restores acyclicity. $\square$

Theorem 2 (Cut Criterion: Necessity). *Let T be an MST under edge costs c. For every tree edge $e \in E(T)$ with fundamental cut $\delta(X_e)$, we have $c_e \le c_{e'}$ for all $e' \in \delta(X_e)$. Equivalently, e is among the cheapest edges crossing its fundamental cut.*

Proof. Assume there exists $e' \in \delta(X_e)$ with $c_{e'} < c_e$. By Lemma 2, $e' \notin E(T)$. Then $T' := T \setminus \{e\} \cup \{e'\}$ is a spanning tree whose cost is strictly smaller than $c(T)$, a contradiction. $\square$

Theorem 3 (MST Criteria Equivalence). *A spanning tree T is an MST under costs c if and only if it satisfies either of the following equivalent properties:*

1. *(Cut property) For every $e \in E(T)$, e is among the cheapest edges in its fundamental cut $\delta(X_e)$.*

2. *(Cycle property) For every $f \in E \setminus E(T)$ and every $e \in C_f \cap E(T)$, $c_e \leq c_f$.*

Proof. The forward implications are Theorems 1 and 2. For the converse, assume the cut property holds and suppose, for contradiction, that there exists a spanning tree T^* with $c(T^*) < c(T)$. Consider the symmetric difference $E(T) \triangle E(T^*)$; pick $e \in E(T) \setminus E(T^*)$ that lies on a path to reduce $c(T)$. The unique cut $\delta(X_e)$ separates the components of $T \setminus \{e\}$; because T^* is connected, it contains some edge $e' \in \delta(X_e) \cap E(T^*)$. By the cut property, $c_e \leq c_{e'}$. Swapping e for e' in T does not increase cost; repeating yields a tree no more expensive than T^*, contradicting minimality of T^*. The cycle-property-based converse is analogous via exchanges along fundamental cycles. □

Proposition 1 (Distinct Weights Imply Uniqueness). *If all edge costs are pairwise distinct, the MST is unique.*

Proof. Under distinct costs, the minimum edge on any cut is unique, hence Theorem 3(1) selects a unique edge at each step of any cut-respecting construction, resulting in a unique MST. Alternatively, if two different MSTs existed, their symmetric difference would contain a cycle with two different maximum-cost edges, contradicting distinctness. □

2.4 Algorithmic Anchors: Kruskal and Prim

Remark 3 (Kruskal). Kruskal's algorithm scans edges in nondecreasing cost order and adds an edge if and only if it does not create a cycle with the edges already chosen. Correctness follows from the cut property: when the algorithm considers an edge e that connects two components of the current forest, e is the cheapest edge crossing the cut induced by these components; therefore it is safe to include. The process terminates with a spanning tree after selecting $|V| - 1$ edges.

Remark 4 (Prim). Prim's algorithm grows a single tree from an arbitrary start vertex by repeatedly adding the cheapest-cost edge leaving the current tree to a new vertex. By the cut property, the chosen edge is safe at each step (it is cheapest across the cut defined by the current tree and its complement). After $|V| - 1$ iterations the result is a spanning tree whose cost is minimum.

2.5 Uncertainty Sets and Robust Criteria (Definitions Only)

Definition 1 (Uncertainty Sets). *We consider two cost-uncertainty models. (i)* Discrete scenarios: *a finite set $\mathcal{U} = \{c^{(1)}, \ldots, c^{(K)}\} \subset \mathbb{R}^E_{\geq 0}$ of edge-cost vectors. (ii)* Interval boxes: *for each $e \in E$, a closed interval $[\ell_e, u_e] \subseteq \mathbb{R}_{\geq 0}$, so the admissible costs form the Cartesian product $\prod_{e \in E}[\ell_e, u_e]$. For any c in the respective uncertainty set, the cost of a spanning tree T is $c(T) = \sum_{e \in E(T)} c_e$, and the scenario-specific MST value is $\mathrm{MST}(()c) = \min_{T \in \mathcal{T}} c(T)$. For a scenario index $u \in \{1, \ldots, K\}$, we write $c^{(u)}$ for the corresponding cost vector with components $c^{(u)}_e$.*

Definition 2 (Robust Objectives)**.** *Given an uncertainty set $\mathcal{U}$: (i) The* Min–Max Spanning Tree *problem seeks $T \in \mathcal{T}$ minimizing the worst-case cost* $\max_{c \in \mathcal{U}} c(T)$*. (ii) The* Min–Max Regret Spanning Tree *problem seeks $T \in \mathcal{T}$ minimizing the worst-case regret* $\max_{c \in \mathcal{U}} \Big(c(T) - \mathrm{MST}(()c)\Big)$*. In the interval model, the maxima are taken over all $c \in \prod_{e \in E}[\ell_e, u_e]$. These are the only robust notions used later; structural results and examples are deferred to Chapters 3 and 4.*

2.6 Algorithmic Preliminaries

AWU 2.6.1 (Complexity glossary). An *optimization problem* associates a cost to each feasible solution and seeks a minimum (or maximum). A related *decision version* asks if there exists a solution of cost at most a given threshold. Class **P** contains decision problems solvable in polynomial time; class **NP** contains problems verifiable in polynomial time. A problem is *NP-hard* if every problem in **NP** reduces to it via a polynomial-time many-one reduction; it is *NP-complete* if it is both in **NP** and NP-hard. An NP-hard problem is *weakly* NP-hard if it admits a pseudo-polynomial-time algorithm; otherwise it is *strongly* NP-hard. A running time is *pseudo-polynomial* if it is polynomial in the numeric value of inputs (e.g., cost magnitudes) rather than only in input length.

AWU 2.6.2 (Approximation glossary). For a minimization problem with optimal value OPT, an algorithm is an *α-approximation* if it returns a solution of cost at most $\alpha \cdot \mathrm{OPT}$ for all inputs (ratio $\alpha \geq 1$). A *PTAS* (polynomial-time approximation scheme) is a family of algorithms that, for any fixed $\varepsilon > 0$, yields a $(1+\varepsilon)$-approximation in time polynomial in the input size (but possibly exponential in $1/\varepsilon$); an *FPTAS* is polynomial in both input size and $1/\varepsilon$. Class **APX** contains problems admitting some constant-factor approximation; a problem is *APX-hard* if it is at least as hard to approximate as every problem in **APX** under appropriate approximation-preserving reductions (e.g., L-reductions). Inapproximability statements are typically conditioned on $\mathbf{P} \neq \mathbf{NP}$.

AWU 2.6.3 (Proof patterns). We rely on three recurring proof patterns. (i) *Exchange arguments* on trees: modify a tree along a fundamental cut or cycle to obtain a cheaper tree, contradicting optimality (used in Section 2.3). (ii) *Extremal arguments* for interval uncertainty: identify a scenario in the uncertainty box that maximizes a given linear functional, avoiding enumeration (details appear in Chapters 3–4). (iii) *Reductions*: classical NP-hardness via many-one reductions and approximation lower bounds via gap-preserving reductions; full proofs are cited or deferred to Appendix A if reproduced.

2.7 Micro-Graph and Notation Pointer

Remark 5. All symbols introduced in Sections 2.1, 2.5 and 2.6 are summarized in Appendix B (Notation Table). We use `\Cref` for all internal cross-references.

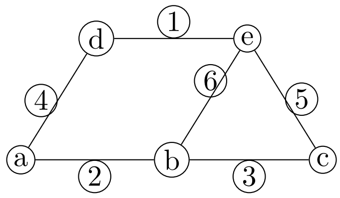

Figure 2.1: Reused micro-graph for illustrating fundamental cuts/cycles.

Chapter 3

Min-Max Spanning Tree

3.1 Models

3.1.1 Discrete Scenarios

$$\min_{T \in \mathcal{T}} \max_{u \in \mathcal{U}} \sum_{e \in T} c_e^{(u)}$$

3.1.2 Interval Uncertainty

3.2 Short Properties Used Later

Lemma 3 (Interval extremal cost for a fixed tree)**.** *For interval costs, the worst-case cost of a fixed tree is attained by setting each chosen edge to its upper bound.*

Proof. □

3.3 Reported Complexity and Approximability

3.3.1 Discrete: dependence on the number of scenarios K

3.3.2 Intervals: NP-hardness

3.4 Modeling Patterns / Formulations (Pointers)

3.5 Worked Micro-Graph Example

Chapter 4

Min-Max Regret Spanning Tree

4.1 Regret Definition and Objective

4.2 Models

4.2.1 Discrete Scenarios

4.2.2 Interval Uncertainty

4.3 Short Properties Used Later

Lemma 4 (Interval extremal regret for a fixed tree)**.** *For interval costs, the worst-case regret of a fixed tree is attained at bound assignments.*

Proof. □

4.4 Reported Complexity and Approximability

4.4.1 Discrete K: constant vs. unbounded

4.4.2 Intervals: NP-hardness; 2-approximation

4.5 Modeling Pointers / Reformulations

4.6 Worked Micro-Graph Example

4.7 Pointer to Appendix A (Representative Full Proof)

Chapter 5

Synthesis and Example Gallery

Chapter 6

Conclusion and Outlook

6.1 Key Takeaways

6.2 Limitations

6.3 Outlook

Appendix A

Notation Table

Symbol	Meaning
$G = (V, E)$	Undirected connected graph
c_e	Edge cost (nominal)
$\delta(X)$	Cut induced by $X \subseteq V$
$\mathcal{T}$	Set of spanning trees
C_e	Fundamental cycle of non-tree edge e w.r.t. T
X_e	Vertex side for the fundamental cut of $e \in E(T)$
$\mathcal{U}$	Uncertainty set of scenarios
$c_e^{(u)}$	Cost of e in scenario u
$\text{Regret}(T, u)$	Scenario regret

Affidavit / Eidesstattliche Versicherung

I hereby declare that I have written this thesis independently and that I have not used any sources or aids other than those indicated. All passages which are quoted or closely paraphrased from other works are marked as such. This thesis has not been submitted in the same or a substantially similar form to any other examination board.

Place, Date Signature