In all problems considered in this paper, the input is the same: An undirected connected graph $G = [V; E]$ with edge weights $w : E \rightarrow R$. We will assume that $G$ is a complete graph on $V$ without loss.

In the Robust Spanning Tree problem, the scenario is as follows: First one selects a spanning tree $T$ in the graph $G$; then an adversary removes a node [an edge] from $T$. We want to select the tree maximize the sum of the weights of the remaining edges.

In the Adaptive Spanning Tree problem, the scenario is modified as follows: First one selects a spanning tree $T$ in the graph $G$; then an adversary removes a node [an edge] from $T$. The first person now adds edges from those not selected or removed so as to connect the components of the tree that remain. We want to minimize the total weight of the original tree plus the patch up cost.

There is an alternative way to define the Adaptive Spanning tree problem in the following manner: First one selects a spanning tree $T_1$ in the graph $G$; then an adversary removes one (or more) node(s) [edge(s)] from $T_1$. Now the first person selects another spanning tree $T_2$ in the remaining subgraph. We want to minimize the sum of weights of edges in the set $T_1 \cup T_2$.

One of the questions we consider is: are the above two formulations equivalent?

0.0.1 Robust Optimization with edge(s) removal

It is "clear" that the adversary will remove the edge(s) with largest weight. This problem is the following problem for the case where one edge is removed:

$$\max_{T \in \mathcal{T}} [w(T) - \max_{e \in T} w(e)]$$

where $w(T) = \sum_{e \in T} w(e)$ and $\mathcal{T}$ is the set of all spanning trees in $G$. The following lemma solves the first player’s problem:

**Lemma 1** Any maximum weight spanning tree in $G$ is optimal.

**Proof.** Let $T_1$ be a maximum weight spanning tree and let $T_2$ be any other spanning tree of $G$. Let the weights of edges in both trees be ordered in decreasing order and let the vectors of weights so ordered be $w(T_1) = [w_1 \geq w_2 \geq \ldots \geq w_m]$ and $w(T_2) = [w'_1 \geq w'_2 \geq \ldots \geq w'_m]$. It is well known that $w(T_1) \geq w(T_2)$ [see
FF]. Hence, removal of the largest $k$ edges from each tree yields a better solution in case for $T_1$ for all values of $k \leq m$.

**Remark 2** This result holds when we consider an arbitrary matroid.

**Remark 3** But the situation is quite different even when one node is removed. [There is no concept equivalent to a node in general matroid.]

### 0.0.2 Robust Optimization with node removal

Here is an example to show now the solution may not be a maximum weight spanning tree.

Weights of red edges is 1 and those of black edges is $1 + \epsilon$ where $0 < \epsilon$. Maximum weight spanning tree consists of black edges with total weight equal to $4(1 + \epsilon)$; but if the central node is removed we will be left with a value of 0 for the remaining edges. Consider a spanning tree consisting of three red edges and one black edge forming a hamiltonian path. The weight is $4 + \epsilon$; maximum loss can be at most $2 + \epsilon$ and hence the weight of remaining edges is $2 > 0$. Hence best solution is not the maximum weight spanning tree for this problem. Moreover, this problem includeds as a special case the problem of finding a spanning tree in an undirected graph whose maximum degree at a node is minimum and this problem is known to be NP-hard since it generalizes the hamiltonian path problem. We show this now.

Let $G$ be a graph with all edge weights equal to 1. For any spanning tree, it should be clear that the adversary will remove the node with the maximum degree resulting in a loss equal to the degree of that node. Hence the tree that is
optimal for the robust problem will be that which minimizes the largest degree. Hence the result follows.

Hence we need to look for approximation algorithms for this problem along the lines of finding approximation algorithms for minimum degree spanning tree problem and its generalizations.

0.0.3 Adaptive Optimization with one edge removal and its patching:

Here is the adaptive optimization problem (using the first form) with one edge removal formulated:

\[
\min_{T \in \Upsilon} \{ w(T) + \max_{e \in T} \min_{f \neq e : \{T - e + f\} \in \Upsilon} w(f) \}
\]

**Theorem 4** Let \( T_1 \) be a minimum weight spanning tree of \( G \). Then \( T_1 \) is optimal for the above problem and hence:

\[
w(T_1) + \max_{e \in T_1} \min_{f \neq e : \{T_1 - e + f\} \in \Upsilon} w(f) = \min_{T \in \Upsilon} \{ w(T) + \max_{e \in T} \min_{f \neq e : \{T - e + f\} \in \Upsilon} w(f) \}
\]

We need a few results before proving this theorem some of which are well known.

**Lemma 5** Let \( T_1 \) be a minimum weight spanning tree. Let \([S, \bar{S}]\) be any cut that partitions the set of nodes \( V \) of \( G \) into two nonempty disjoint sets. Let \( w \) be the minimum weight of any edge that crosses this cut. Then, among the edges of \( T_1 \) that cross this cut, at least one of them has a weight equal to \( w \).

**Proof.** We give a proof for the sake of completeness (although this is probably a well known result). Suppose the result is not true. Let \( e \) be a minimum weight edge crossing the cut and by supposition, \( e \notin T_1 \). Adding \( e \) to \( T_1 \) creates a unique cycle \( C \) of which \( e \) is a part. Since \( e \in C \) crosses the cut, there is another edge \( f \) that also crosses the cut but whose edge weight \( w(f) > w(e) \) by supposition. Consider the spanning tree \( T_2 = T_1 - e + f \). It should be clear that \( w(T_2) < w(T_1) \) and this contradicts the assumption that \( T_1 \) is a minimum weight spanning tree of \( G \). ■

**Proof.** (of Main Theorem): Suppose the claim in the theorem is not true. Let \( T_2 \neq T_1 \) be a spanning tree that satisfies the relation:

\[
w(T_2) + \max_{e \in T_2} \min_{f \neq e : \{T_2 - e + f\} \in \Upsilon} w(f) = \min_{T \in \Upsilon} \{ w(T) + \max_{e \in T} \min_{f \neq e : \{T - e + f\} \in \Upsilon} w(f) \}
\]

Let \((u, v) = e_1 \in T_2 \setminus T_1 \). Consider \( T_2 \setminus e_1 \); this consists of two subtrees one of which contains an edge incident at \( u \) and the other contains an edge incident at \( v \). Let these two subtrees (that are part of \( T_2 \)) be denoted respectively by \( T_2^u \) and \( T_2^v \). Let

\[w(f_1) = \min_{f \neq e_1 : \{T_2 - e_1 + f\} \in \Upsilon} w(f)\]

We will call the edge \( f_1 \) as a minimum weight patching edge for \( e_1 \) with respect to tree \( T_2 \) and denote \( f_1 \) by \( \text{patch}_{T_2}(e_1) \). Similar remarks apply to any edge of any spanning tree.
**Remark 6** Minimum Weight Patching edges may not be unique but their weights are equal.

**Proof.** (continued): The edge $f_1$ connects the subtrees $T_u^w$ and $T_v^w$ in place of the edge $e_1$. The choice would be some minimum weight edge not in $T_2$ that crosses the cut with $S = \{\text{set of nodes in } T_2^w\}$. Among the edges of $T_1$ that cross this cut [and there are such edges none of which is $e_1$], we select one with minimum weight. This also happens to be minimum weight edge that crosses this cut by above lemma. Hence $w(f_1) \leq w(e_1)$. Consider the tree $T_3 = T_2 - e_1 + f_1$. This tree $T_3$ has one more edge in common with tree $T_1$ than tree $T_2$. Weight of the minimum weight patching edge for $f_1$ with respect to tree $T_3$ can not be larger than $w(e_1)$ since $e_1$ is a candidate for patching for $f_1$ with respect to tree $T_3$. Moreover,

$$w(T_3) + w(\text{patch}_{T_3}(f_1)) \leq w(T_3) + w(e_1) = w(T_2) + w(f_1) = w(T_2) + w(\text{patch}_{T_2}(e_1))$$

Now if we show that

$$w(T_3) + w(\text{patch}_{T_3}(p)) \leq \max\{w(T_3) + w(e_1); w(T_2) + w(\text{patch}_{T_2}(p))\} \quad \forall p \in T_2 \setminus e_1$$

then it would show that $T_3$ is no worse than $T_2$ for the adaptive spanning tree problem. By repeating this process the theroem would be proved since $T_3$ has one more common edge with tree $T_1$. Without loss we will show this for some edge $p$ in $T_2^u$. Removing $p$ from $T_2^u$ breaks it into two parts $T_2^u(1)$ and $T_2^u(2)$. See diagram below:

![Diagram](image-url)
There are two cases to consider: (i) one end of $f_1$ is in $T_u^2(1)$ [this is what is shown in the above diagram]; (ii) one end of $f_1$ is in $T_u^2(2)$ (this is shown in diagram below). In the first case, when we remove edge $p$ from $T_3$ we get two subtrees that are respectively $T_u^2(1)$ and $T_u^2(1) + T_v^2 + f_1$. In the second case we get two subtrees that are respectively $T_u^2(1)$ and $T_u^2(2) + f_1 + T_v^2$.

If $patch_{T_2}(p) = q$ does not cross the cut $(S, \bar{S})$ with $S = \{\text{set of nodes in } T_u^2\}$, then $q$ is still a candidate for $patch_{T_3}(p)$ and hence

$$w(T_3) + w(patch_{T_3}(p)) \leq \max[w(T_3) + w(e_1); w(T_2) + w(patch_{T_2}(p))]$$

continues to hold. Now we assume that $patch_{T_2}(p) = q$ crosses the cut $(S, \bar{S})$. Consider the two cases separately:

Case (i): In this case the edge $e_1$ is a candidate for $patch_{T_3}(p)$ and hence

$$w(T_3) + w(patch_{T_3}(p)) \leq \max[w(T_3) + w(e_1); w(T_2) + w(patch_{T_2}(p))]$$

case (ii): See diagram below:

In this case $patch_{T_2}(p)$ is still a candidate for $patch_{T_3}(p)$ and hence

$$w(T_3) + w(patch_{T_3}(p)) \leq \max[w(T_3) + w(e_1); w(T_2) + w(patch_{T_2}(p))]$$

This completes the proof of the main theorem for one edge removal. ■
0.0.4 Adaptive Optimization with one node removal:

Let $G$ be a graph with all edge weights equal to 1. For any spanning tree, it should be clear that the adversary will remove the node with the maximum degree resulting in additional cost of patching equal to the degree of that node. Hence the tree that is optimal for the adaptive optimization problem will be that which minimizes the largest degree. Hence the result follows that this problem is also NP-hard.