Fair and Reliable Reconnections for Temporary Disruptions in Electric Distribution Networks using Submodularity

Cyrus Hettle[1], Swati Gupta[1], and Daniel Molzahn[1]

[1]Georgia Institute of Technology
{chettle,swatig,molzahn}@gatech.edu

Abstract

We analyze a distributed approach for automatically reconfiguring distribution systems into an operational radial network after a fault occurs by creating an ordering in which switches automatically close upon detection of a downstream fault. The switches' reconnection ordering significantly impacts the expected time to reconnect under normal disruptions and thus affects reliability metrics such as SAIDI and CAIDI, which are the basis for regulator-imposed financial incentives for performance.

We model the problem of finding a switch reconnection ordering that minimizes SAIDI and the expected reconnection time as Minimum Reconnection Time (MRT), which we show is a special case of the well-known minimum linear ordering problem [29] from the submodular optimization literature, and in particular the Min Sum Set Cover problem (MSSC) [19]. We prove that MRT is also NP-hard.

We generalize the kernel-based rounding approaches of Bansal et al. [5] for Min Sum Vertex Cover to give tight approximation guarantees for MSSC on c-uniform hypergraphs for all c. For all instances of MSSC, our methods have a strictly better approximation ratio guarantee than the best possible methods for general MSSC.

Finally, we consider optimizing multiple metrics simultaneously using local search methods that also reconfigure the system's base tree to ensure fairness in service disruptions and reconnection times and reduce energy loss. We computationally validate our approach on the NREL SMART-DS Greensboro synthetic urban-suburban network. We evaluate the performance of our reconfiguration methods and show significant reductions compared to single-metric-based optimizations.

1 Introduction

Electric power systems are transitioning from their traditional static centralized paradigm towards future operation that is increasingly adaptive, dynamic, and distributed [55, 18]. As part of this transition, the protection systems which prevent damage from faults and the network reconfiguration systems which restore the supply of power are becoming more advanced. Protection and reconfiguration systems are moving towards an adaptive, dynamic, and distributed paradigm where components of the distribution system work together to detect and characterize failures and then restore the supply of power as quickly as possible [49, 12, 48, 45, 62]. In an effort to help realize this future paradigm, this paper analyzes decentralized approaches for automatically reconfiguring distribution networks to quickly restore power after a fault.

Improvements in distribution network reconfiguration are motivated by the high societal cost of electricity interruptions. Despite investments from utility companies in improving reliability, service disruptions remain challenging [11, 65, 41, 42]. Recent research in monetizing the economic costs of power outages provides estimates between \$35 billion and \$50 billion annually in the United States [40]. Related research studies on the Value of Lost Load (VOLL) give estimates ranging from several dollars per kilowatt-hour to several hundred dollars per kilowatt-hour, depending on the underlying methodology, type of customer, and assumptions [56]. Industry uses similar values in their operation and planning activities; for instance, the Midcontinent Independent System Operator (MISO), which is one of the largest power system operators in the United States, uses a value of lost load equal to \$35 per kilowatt-hour [47]. These values, which are orders of magnitude larger than the typical costs for electricity (around 10.5 cents per kilowatt-hour in the United States [60]), underscore the importance of providing reliable supplies of electricity.

Analyzing and advancing methodologies for network reconfiguration in order to more quickly restore power is an important topic for improving the reliability of electric power systems. Accordingly, this paper develops a combinatorial model to characterize and improve the performance of reconfiguration systems for electric distribution networks. We aim to reconfigure the network topology after a fault occurs in order to optimize various performance criteria. Dating back to the 1970s [46], distribution network reconfiguration is a widely studied problem in power systems engineering and remains an active research topic [48, 66, 68, 63, 67, 21, 58, 32, 17]. The goal of distribution network reconfiguration is to optimize one or more performance criteria (typically related to service reliability and network losses) while maintaining an operable distribution network configuration. Many types of algorithms have been applied to this problem, including, e.g., traditional optimization methods such as linear programming [58] and mixed-integer programming [8, 20], heuristic decision strategies [33, 68], machine learning methods [67, 21, 68, 32], approximation algorithms [25], and metaheuristic techniques [67, 63, 17]. See [48] for a recent survey of the network reconfiguration literature.

In industrial applications, optimal network reconfiguration is classified as a Fault Location, Isolation, and Service Restoration (FLISR) application that aims to embed "self-healing" features within advanced distribution management systems (ADMS) [15, 59, 54]. Industrial implementations of network reconfiguration include, for instance, ADMS products provided by Schweitzer Engineering Laboratories [57], Advanced Control Systems, Inc. [1], General Electric [22], S&C Electric [31], etc. Case studies and discussions on a range of practical implementation issues are provided in [15, 59, 43].

Much of the existing work in the academic literature focuses on *centralized* approaches for distribution network reconfiguration. In centralized approaches, information regarding distribution system operations is communicated to a central location (e.g., substation or control center) which computes an appropriate switching configuration that is then communicated to each switch. In contrast, *distributed* approaches involve limited data exchange between local controllers, and *decentralized* approaches have local controllers that operate solely based on local information. While capable of identifying high-quality solutions to the network reconfiguration problem, centralized approaches require significant amounts of data and communication infrastructure, can be slow to execute, and introduce single points of failure and cybersecurity concerns [30]. In contrast, distributed and decentralized approach can operate more quickly, do not have a single point of failure, and can have advantages in terms of cybersecurity, but may not always reach the optimal solution computed by a centralized approach. A mix of centralized, distributed, and decentralized approaches are commonly applied in practice. Distributed and decentralized approaches are often used in settings where fast restoration times are critical or to back up centralized approaches [59].

In this paper, we focus on a decentralized setting where individual switches restore power without requiring real-time central coordination. To this end, we represent the distribution system by a graph $G = (V, E)$ with power demands $w(v)$ associated with each vertex $v \in V$, a source vertex $r \in V$ to model the substation, and an initial configuration as a spanning tree T (i.e., a radial network).

Region	New England	Middle Atlantic	East North Central	West North Central	South Atlantic
Average SAIDI	275	229	281	208	233
Region	Mountain	Pacific	East South Central	West South Central	US Total
Average SAIDI	122	149	218	220	214

Table 1: Average annual SAIDI values (in minutes) for regions of the United States from 2013 to 2015.

We assume that each edge e in the tree T has a given probability of failure $p(e)$. Non-tree edges $E \setminus T$ model switches that can operate in a decentralized manner to connect disconnected parts in a network following an edge failure. Physically, we assume that the switches can detect a voltage difference on either end and close when this event happens. We say that each switch covers the tree edges on the unique path between the switch's endpoints. To ensure that the reconnected graph is still radial (i.e., a tree), we seek an ordering in which the switches can automatically activate.

The methods analyzed in this paper for performing this reconfiguration aim to optimize various performance metrics while avoiding the need for centralized communications and computations. We show that any ordering suffices for our system model, but can result in different expected reconnection times. We refer to the problem of minimizing this expectation as the Minimum Reconnection Time problem, or MRT. We show that MRT is a special case of the well-known minimum linear ordering problem (MLOP) from the submodular optimization literature [29], and in particular of the Min Sum Set Cover (MSSC) problem [19]. Since it is a special case, the hardness of MSSC does not directly imply the hardness of MRT, which we show still holds using a reduction from the so-called tree augmentation problem [13].

Minimizing average reconnection time, however, does not capture disproportionate impact on different parts of the network, and therefore, different demographics who are typically situated in clusters or strongly correlated with zipcode locations [14]. We use two metrics to characterize the severity of service disruptions [28]:

(i) Reconnection time (R-Time): expected time between an edge failure and the reconnection of the network.

(ii) SAIDI: average expected total outage duration for a customer over the time period.

Utility companies and regulatory agencies closely monitor these reliability metrics and use them to assess utility performance during the rate cases that determine utility revenues. While there are variations due to exogenous differences between locations (e.g., more or less frequent severe weather), utility practices strongly influence the values of these service reliability metrics. Thus, utilities compare these service metrics to judge performance across the industry, especially within similar geographic regions [4, 27]. Using EIA Form 861 data [61], Table 1 from [40] shows values for SAIDI averaged across various regions of the United States.

Further illustrating their importance, electric industry groups award utilities based on their performance with respect to these reliability metrics (e.g., the ReliabilityOne Award [50] and the American Public Power Association's Certificate of Excellence Award [3]). Moreover, some regulators implement explicit financial incentives or penalties for utilities to improve upon these metrics. As one example, the California Public Utilities Commission set a target SAIDI value of 60 minutes for San Diego Gas & Electric [9]. Overperforming or underperforming this target beyond ± 2 minutes was rewarded or penalized by $375,000 per minute up to a maximum of $3 million annually. Financial incentives also vary across regions and countries. For instance, in Great Britain, both the Interruption Incentive Scheme and Guaranteed Standards requirements give distribution utilities financial incentives to reduce SAIDI. Similarly, the Italian Regulatory Authority for Electricity and

Gas sets benchmarks for SAIDI and penalizes distribution utilities based on the difference between these benchmarks and their performance [26]. Further discussion of performance-based approaches to regulation in the utility industry is available in e.g., [36, 34].

In this work, we model R-Time and SAIDI using weighted instances of the Minimum Reconnection Time problem. Other commonly used metrics such as CAIDI (the average expected duration of an outage, weighted by the number of customers affected) can also be optimized for using these methods. Our first set of theoretical results deal with providing approximation guarantees for settings where the goal is to minimize R-Time or SAIDI for a given radial network. In particular, we extend a recent kernel-based rounding approach of Bansal et al. [5] for MSSC to find approximation ratios dependent on the number of switches that can reconnect each tree edge. However, in practice, the interplay of these metrics becomes important due to the financial incentives faced by utility companies and the impact on service level for customers.

We further find that the choice of starting tree can also have a large impact on the reconnection metrics. For instance, Figure 1 shows two different spanning trees on the 7-vertex wheel graph. Each spanning tree has a unique optimal permutation (up to graph isomorphism) that minimizes both of SAIDI and R-Time. The optimal permutation of the "spoke" network on the left achieves objective value $\frac{3}{2}$ for both R-Time and SAIDI, while in the "wheel" network on the right, the optimal permutation achieves objective value 1 for R-Time and $\frac{7}{2}$ for SAIDI. Moreover, the choice of the initial tree also plays a significant role towards determining the energy losses in the network [39, 25], although this objective cannot be modeled as a special case of MRT.

To optimize the radial configurations for distribution systems, we propose using local search techniques over the set of spanning trees using a multi-criteria objective that involves all three metrics of interest: SAIDI, R-Time, and energy. The idea of local search goes back to Calinescu et al. [10] for maximizing monotone submodular functions. Our objectives here instead require us to minimize a *product* of sums of supermodular functions over changing network topologies, thus making the analysis harder for local search. For our numerical tests, we apply local search to the Greensboro, NC urban-suburban synthetic network developed by NREL's SMART-DS project [52]. However, each iteration of the local search is extremely slow due to the large size of the network. Therefore, we give more efficient methods for updating the connectivity in the network, reducing the running time of local search. Moreover, in each iteration of local search, solving an integer program (IP) to find the exact minimum reconnection times or SAIDI (each an NP-hard problem) is expensive. We finally show that the approximation algorithms developed in the first part of this work provide reasonable proxies for making progress in the local search algorithm.

We summarize our key contributions below:

(i) **Modeling Faults and Protection Systems:** We give a novel model to capture single-occurrence faults in an electricity distribution network. We formulate the *minimum reconnection time* (MRT) problem for finding a reconnection order in this model which minimizes delays in service restoration and hence can improve performance metrics such as SAIDI and CAIDI which are the basis of performance incentives for utility companies. (Section 2)

(ii) **Connections to submodular literature and NP-hardness:** We show that MRT is a special case of the well-studied minimum sum set cover (MSSC) problem [19]. We show that this restriction of MSSC is also NP-hard by using a reduction from the tree augmentation problem [13]. (Section 3)

(iii) **Approximation algorithms:** The best known approximation for the MSSC problem is factor 4, achieved by a simple greedy algorithm, and a recent result by Bansal et al. [5] gives an α-point rounding approach which also a 4-approximation for MSSC. We extend the latter

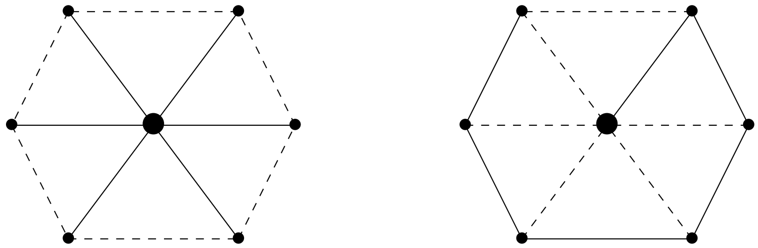

Figure 1: Two spanning trees on the same network. Solid edges are in T, dashed edges are in $E \setminus T$, the source r is the largest vertex, and all vertex weights and edge failure probabilities are equal. The optimal reconnection order of the "spoke" network (left) achieves an objective value of $\frac{3}{2}$ for both R-Time and SAIDI; in the "wheel" network (right), the optimal reconnection order achieves an objective value of 1 for R-Time and $\frac{7}{2}$ for SAIDI. Energy is also different: 6 in the "spoke" and 91 in the "wheel".

approach to show a better, and tight, approximation ratio of $(2c/(c+1))^2$ for MSSC on c-uniform hypergraphs, generalizing the $c = 2$ case shown by Bansal et al. This corresponds to MRT instances in which each edge in the given radial network is covered by at most c switches. (Section 4)

(iv) **Local search for multi-objective minimization:** To find radial networks that can achieve reasonable R-Time, SAIDI, and energy losses simultaneously, we use local search over spanning trees using branch exchange steps, where one tree edge is replaced by a switch. A single local search step may check dozens or hundreds of possible branch exchanges, and for each possible exchange, the connectivity information needed for computing reconnection orders changes. Over networks of practical interest, existing methods for detecting cycles are very slow. We give a method for updating connectivity which achieves a factor of $\Omega(\min\{|E-T|, |V|/d\})$ speedup, where d is the length of the longest cycle in G.[1] (Section 5)

(v) **Computations:** Finally, we computationally validate our approaches on networks generated using the Greensboro, NC dataset from the NREL SMART-DS project [52]. We first show that the performance of the alpha-based rounding algorithms changes favorably as c is decreased. Moreover, in practice, we propose taking the best of the greedy and the α-point rounding solutions, since our theoretical results are worst-case bounds and the greedy algorithm often performs favorably. We next show that local search with approximate R-Time and SAIDI computations can in fact find network topologies that achieve close to a factor of 80% reduction in R-Time, 70% reduction in SAIDI, and 35% in energy. (Section 6)

2 Problem Formulation and Modeling

2.1 Modeling the network

We model an electric distribution network as a graph $G = (V, E)$, where the vertices V represent the buses and the edges E are distribution lines connecting the buses. We consider radial distribution networks with a single power supply at a substation $r \in V$, which is the root of a spanning tree T

[1] That is, our methods improve running time by at least a constant multiple of $\min\{|E-T|, |V|/d\}$ asymptotically as the size of the network increases.

on G. Let $S = E \setminus T$ denote the set of edges not in the spanning tree. There is also a power demand $w(v)$ for each vertex $v \in V$ and an electrical resistance $R(e)$ for each edge $e \in E$, which depends on the physical properties of the corresponding distribution line.

For all $e \in E$, we let $p(e)$ be the expected number of times that edge $e \in T$ will fail in a fixed time period (interchangeably, we think of $p(e)$ as the probability of failure.) Let $d(e)$ be the set of vertices downstream of e: i.e., those vertices v such that the unique path in T between s and v includes e) and $f(e)$ be the total weight of the vertices in $d(e)$. Let $t(e)$ be the time to restore power when e fails. Note that p, d, and w are functions of the network configuration, while t also depends on the methods used to restore power in the network. We next discuss the model for faults and the protection system.

We note that this graph-based modeling approach is a simplified representation of the distribution network used in, for instance, power flow studies that additionally require such information as the line reactances, reactive demands, capacitor susceptances, etc. [37]. While precluding details such as reactive power injections and voltages, this modeling approach enables application of the graph theoretic concepts that are the basis for this paper's theoretical analyses. Similar graph-based models are used in other power system analyses, such as [2, 51, 39, 38, 25].

2.2 Modeling faults and the protection system

Each edge in E has a switch which can be opened or closed to allow or stop current flow, and some edges have breakers which will trip and stop flow in the event of a nearby fault. In general, not every edge will have a breaker present. However, we can preprocess the graph by contracting each set of connected edges in $E(0)$ that has only one breaker to a single edge.[2] At any time t, let $E(t) \subseteq E$ be the set of edges across which current is allowed to flow, i.e., those edges whose switches are closed. Then $E(t)$ will always be a forest and the vertices receiving power are precisely those in the same component as the power source. Each substation supplies power to its connected component, which is in turn operated radially [64, 37]. At $t = 0$, $E(t)$ is the spanning tree T since all edges are operational and all vertices are in the same component as r. If failures occur at time $t > 0$, $E(t)$ could become the disconnected union of multiple trees.

Suppose a fault occurs in edge e at time 0. To protect the system against this fault, a breaker on edges e will trip to isolate the fault. Consequently any vertices downstream of the downstream breaker(s) will be without power until the fault is repaired. However, closing a switch on an edge not currently in use (i.e., in S) could allow current to flow through that edge to one of the disconnected vertices from another vertex that is still energized. We would like to do so to temporarily restore power to as many vertices as possible as quickly as possible to maximize service availability, while still ensuring that no current will flow to the component of $T \setminus \{e\}$ containing the faulted edge, thus continuing to isolate the fault.

For instance, consider the network in Figure 2. Vertex r is the source. Edges in T are solid and edges in S are dashed. Suppose a fault occurs on edge e. Breakers will trip, isolating the fault, and vertices v_2, v_3, v_4, and v_5 will be without power since they are no longer in the same component of $E(t)$ as the source vertex. If the switch on s_1 or the switch on s_2 is closed, then $E(t)$ is again a tree and power will be restored to all vertices. Closing the switch on s_3 would not restore power, since vertex v_1 is upstream of the fault. Closing switches on both s_1 and s_2 would create a cycle in $E(t)$.

In our model, choosing which edge(s) to close switches on is not a difficult task if we have perfect information of the status of the network (such as which breakers have tripped) and can open and close switches centrally. To do so, we choose any edge in S that connects a vertex with power to a

[2] The fault probability of this edge is the sum of the fault probabilities of the contracted edges. The metrics depend on the weight of each individual vertex, but this can be approximated by the total weight on the contracted vertex.

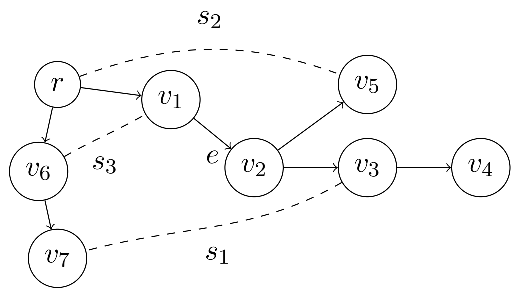

Figure 2: An example of a network comprising a spanning tree (solid edges) and switches (dashed edges.)

vertex that is without power and not in the component of the fault and close the switch on that edge. Repeat this process, closing switches on edges in S until no such edges exist. This will restore power to as many vertices as possible.

However, performing this operation centrally can be challenging and slow. Instead, we consider a process that can run locally in a decentralized manner. That is, every switch has a relay which reads the voltages of its incident vertices. Based on that information, the relay can open or close the switch automatically, possibly with a timed delay.

Once the breakers trip, every vertex receiving power will have positive voltage, and the others will have zero voltage. We assume that each switch can detect whether each of its terminal vertices have positive voltage. The switch also has a map of G, but does not know $E(t)$ for $t > 0$ or which breakers are currently tripped.

Suppose that a single edge fails, every edge has a breaker, and we are guaranteed that the breakers closest to the fault do trip (operate as intended). Then all vertices upstream of the fault have power, and the fault is isolated from all vertices downstream of the fault. Therefore, we may safely restore power anywhere. Include in the network a priority time $t(e)$ for each edge e. Ensure that these priority times are sufficiently separated so that for any edges e_1, e_2 with $t(e_1) < t(e_2)$, any changes in the network (faults, breakers tripping, switches being opened and closed, detecting the voltage of a vertex) that occur before time $t(e_1)$ will be completed before time $t(e_2)$.

Consider the following algorithm. When a switch $s \in S$ detects that one of its terminals has positive voltage and the other does not, it waits for preassigned length of time $t(s)$. If after that time the voltages of its terminals are unchanged, the relay closes the switch on that edge. Hence power will be restored to the component of $E(t)$ containing the zero voltage vertex and will not be sent across the fault. Moreover, by the assumption on $t(s)$, it cannot be the case that multiple switches close in such quick succession that a cycle forms in $E(t)$. Note that edge s will detect a difference in voltages (and potentially try to close) if and only if e, the edge that failed, is in the unique cycle in $T \cup \{s\}$; in other words, $T \setminus \{e\} \cup \{s\}$ is a tree. In this case, we say that s *covers* e.

For simplicity, henceforth we let $\varepsilon > 0$ be such that if $|t(s_1) - t(s_2)| \geq \varepsilon$ for all $s_1 \neq s_2 \in S$, the above condition will be satisfied; then set all times $t(s)$ to be distinct integer multiples of ε. Equivalently, we may consider a permutation σ of S, where switch σ_k waits time $k\varepsilon$. Finally for all $e \in T$, set $t(e) = \min_{s \in S : s \text{ covers } e} t(s)$. Any valid function $t : S \to [\varepsilon, \infty)$ will yield an assignment of values for $t(s)$ and eventually restore power to the maximum possible number of vertices, but we would also like to do so as quickly as possible. There are several different metrics for measuring the speed of a reconnection method.

2.3 Definition of metrics

We consider three metrics on networks and reconnection methods. The System Average Interruption Duration Index (SAIDI) is a standard metric that measures average outage time, weighted by the number of consumers (vertex weights) affected by each outage:

$$\text{SAIDI} = \frac{\sum_{v\in V} w(v) \sum_{e:v\in d(e)} p(e)t(e)}{\sum_{v\in V} w(v)} = \frac{\sum_{e\in T} f(e)p(e)t(e)}{\sum_{v\in V} w(v)}.$$

We next consider the reconnection time R-Time metric, which measures the expected length of time of an outage and the reconnection of the network, without considering the demand at each bus:

$$\text{R-Time} = \frac{\sum_{e\in T} p(e)t(e)}{\sum_{e\in T} p(e)}.$$

Finally, we consider the energy loss in the network due to the line resistances, which can be a substantial operational cost. If we assume that power consumption is proportional to vertex weight, by Ohm's law the energy loss in any edge $e \in T$ is $R(e)f(e)^2$. (Note that we do not consider the additional non-linear complications of AC power flow.) Then the total energy loss in the network is the quantity

$$\text{Energy} = \sum_{e\in T} R(e)f(e)^2.$$

SAIDI and R-Time can be viewed as special instances of a new problem: Min Reconnection Time (MRT), which aims to find an ordering of switches that minimizes a weighted total waiting time for the edges of T to be reconnected. That is, given a distribution network $G = (V, E)$, a spanning tree T over V and a set of switches $S = E \setminus T$, edge weights on the tree edges $T \to \mathbb{R}_+$, the MRT problem is to find an ordering σ of switches that minimizes the weighted reconnection time:

$$\text{(MRT):} \quad \min_{\sigma:S\to[|S|]} \sum_{e\in T} b(e) \min_{s \text{ covers } e} \sigma(s) \tag{1}$$

The tree edge weights $b(e)$ are chosen to reflect our choice of instance and metric. If we set $b(e) = p(e)$, the resulting objective is the expected amount of time to power restoral after an outage (R-Time), while if we set $b(e) = p(e)f(e)$, we obtain expected duration of time that an average customer is without power, i.e., SAIDI (up to the constant factor of the denominator of those objectives.) Therefore, when we discuss methods or results for MRT, these methods apply to the specific problems of finding a reconnection order to minimize either of these metrics.

3 Connections to the Minimum Linear Ordering Problem

In this section, we show that MRT is a special case of the Min Sum Set Cover Problem, which in turn is a special case of the Minimum Linear Ordering Problem on supermodular functions. We also show that MRT is NP-hard using a reduction from the Tree Augmentation Problem.

3.1 Notation and Preliminaries

We start with some preliminaries needed to review this section. A set function $f : 2^V \longrightarrow \mathbb{R}$ on a ground set V is supermodular if for all $A, B \subseteq V$, $f(A\cup B)+f(A\cap B) \geq f(A)+f(B)$. For example, the function $f(A) = |A|^2$ is supermodular. A hypergraph $H = (V, E)$ is a pair of a vertex set V and a set of hyperedges E, where each hyperedge $e \in E$ is a nonempty subset (of any size) of the vertex set V. This generalizes the notation of graphs $G = (V, E)$ where each edge $e \in E$ is a subset of V containing exactly two vertices, i.e., $e = (u, v)$ for $u, v \in V$. A hypergraph is said to be c-uniform if each hyperedge contains exactly c vertices. Further, we denote orderings of any finite set V by $\sigma : V \to V$, where $\sigma(i)$ gives the position of element $i \in V$.

In this section, we show that the MRT is an instance of the Minimum Linear Ordering Problem (MLOP) on supermodular functions, and in particular the Min Sum Set Cover (MSSC) problem introduced in [19]. We first define the general form of the MLOP problem, with respect to any set function f:

Definition 1. *Given a ground set S and a function $f : 2^S \longrightarrow \mathbb{R}_+$, the MLOP is to find a linear ordering σ of S that minimizes $\sum_{i=1}^{|S|} f(\{s \in S : \sigma(s) \leq i\})$.*

To see how MRT can be viewed as a special case of MLOP, consider a given spanning tree T of the underlying network $G = (V, E)$. The goal is to find an ordering of the ground set of switches $S = E \setminus T$ which minimizes the time for edges in T to be covered. For each switch s, define U_s to be the set of tree edges covered by s. Given a set of switches $W \subseteq S$ then $f(W)$ is defined to be the cardinality of the set of tree edges not covered by W, i.e., $f(W) = |T \setminus \cup_{s \in W} U_s|$. This function can be shown to be supermodular. At each time period $t = i$, any edge which is not covered by the first $i - 1$ switches adds 1 unit to the objective function. As it is in MRT, the value of the set function f can be weighted by $b(e)$, thus recovering R-Time or SAIDI as the objective as desired. Any ordering that minimizes the above mentioned instance of MLOP also minimizes MRT.[3]

Another instance of the MLOP is the Min Sum Set Cover (MSSC) problem, which given a set of subsets $S_1, \ldots, S_k \subseteq S$, seeks to find an ordering of the S_i such that the total waiting time for each element of the ground set S to be covered is minimized. As an instance of MLOP, this problem uses the function $f(U) = |S \setminus \cup_{i \in U} S_i|$ for $U \subseteq V$, which again can be shown to be supermodular. We may view this instance using the hyperedges of a hypergraph $H = (V, E)$, where each hyperedge $e \in E$ is a nonempty subset (of any size) of the vertex set V. Taking the ground set to be the edge set E, the MSSC problem can then be formally stated as:

Definition 2. *The* MSSC *problem, given a hypergraph $H = (V, E)$, is to find a linear ordering σ of vertices V so that the total waiting time for each edge to be covered is minimized, i.e., $\sum_e \min_{v \in e} \sigma(v)$ is minimized.*

Again, note that one can write an instance of MRT as an instance of MSSC by taking H to be the hypergraph with a vertex for each switch $s \in S = E \setminus T$ and a hyperedge $\{s \in S : T - e + f \text{ is connected}\}$ for each tree edge $e \in T$. However, MRT instances form a strict subset of MSSC instances, as we show in the lemma below:

Lemma 1. *There exist instances of* MSSC *which are not instances of* MRT*, so* MRT *is in fact a special case of* MSSC*.*

For instance, let $H = (\{1, 2, 3, 4\}, \{\{1, 2, 3\}, \{1, 4\}, \{2, 4\}, \{3, 4\}\})$. Then in any corresponding instance of MRT, for a set of switches S corresponding to the hyperedges of H to exist, tree edges e_1, e_2, e_3 must lie on a path, with a fourth tree edge e_4 incident to each of them. However, this is impossible. Since MRT is a strict subset of MSSC, showing the NP-hardness of MRT requires additional argument. It also may not hold in special instances of MRT, for instance if G is planar.[4]

[3] There is an additive factor of $|S|$ in the objective that is not accounted for in the mentioned MLOP reduction. However, this does not change the optimal solution.

[4] For example, even though finding the maximum weighted cut in a graph is NP-hard [24], it is solvable in polynomial time when restricted to planar instances [16].

3.2 NP-hardness of Minimum Reconnection Time

To show the NP-hardness of the minimum reconnection time problem (MRT), we show a reduction from another NP-hard problem called the Tree Augmentation problem (TAP) [13]. Given $G = (V, E)$ and a spanning tree $T \subseteq E$ on V, the objective of TAP is to find a minimum-cardinality set of edges $F \subseteq E$ such that $T \cup F$ is 2-edge-connected. Similarly, a waiting time assignment t (a solution to MRT) is guaranteed to restore power by time k if the subgraph of G containing the edges of $T \cup \{s \in S : t(s) \leq k\varepsilon\}$ creates a 2-edge-connected graph. Note that MRT can be viewed as a timed version of TAP, much as MSSC is the timed version of ordinary Set Cover, where the latter seeks to find the minimum number of subsets to cover a given set of elements [35].

Using a reduction from TAP, we show that the MRT problem is NP-hard. The idea is to take an instance of TAP, modify the graph in the instance by adding polynomially many short paths, and show that the solution to the MRT problem on the resulting network yields a solution to TAP on the original network. Therefore, if MRT can be solved in polynomial time, this would imply a solution of TAP in polynomial time, thereby contradicting its NP-hardness.

Theorem 1. *The* Tree Augmentation Problem *can be reduced in polynomial time to the* Min Reconnection Time *problem, and the latter problem is therefore NP-hard.*

Proof. Let $G = (V, E)$ and T be an instance of tree augmentation. Let $|V| = n, |E| = m$. Let OBJ be the objective value for the problem MRT of finding an edge order on this network (with $w(v) = 1$ for all $v \in V$ and $p(e) = 1$ for all $e \in T$) that minimizes expected R-Time. Then OBJ is positive and at most $\frac{n-1}{m} \cdot \frac{m(m+1)}{2} = \frac{(n-1)(m+1)}{2}$, with that value being attained only if in the best solution each of the m edges covers an equal number of the edges in T.

Let v be an arbitrary vertex in T and let T' be the tree given by adding the $\frac{n-1}{m} \cdot \frac{m(m+1)}{2}$ 2-paths $\{\{\{v, v_{1,1}\}, \{v_{1,1}, v_{1,2}\}\}, \ldots, \{\{v, v_{mn,1}\}, \{v_{mn,1}, v_{mn,2}\}\}\}$ to T, where $v_{i,j} \notin T$ for all i, j. Give every edge e in $T' \setminus T$ failure probability $p(e) = 1/2$. Let G' be the graph given by $\left(V \cup \bigcup_{i \in \{1,\ldots,mn\}, j \in 1,2} v_{i,j}, E \cup \{\{v, v_{1,2}\}, \ldots, \{v, v_{mn,2}\}\}\right)$.

Let t be any solution to MRT on this network, with $E(0) = T'$. Note that the newly added edges $\{\{v, v_{1,2}\}, \ldots, \{v, v_{mn,2}\}\}$ are the only edges that cover $T' \setminus T$. Furthermore, since these edges each cover tree edges of total failure probability 1, without loss of generality we may assume they are the last mn edges that contribute a positive value to the objective taken in the solution to MRT. Let the edges chosen before those mn edges be $e_1, \ldots, e_k$, and let C_i be the unique cycle in $T \cup \{e_i\}$. Then the objective value of the solution is

$$\begin{aligned} OBJ(\sigma) &= \sum_{i=1}^{k} i \sum_{e \in (C_i \setminus \{e_i\}) \setminus \bigcup_{j=1}^{i-1} (C_i \setminus \{e_i\})} 1 + \sum_{\ell=k+1}^{k+mn} 1 \\ &= \sum_{i=1}^{k} i \sum_{e \in (C_i \setminus \{e_i\}) \setminus \bigcup_{j=1}^{i-1} (C_i \setminus \{e_i\})} 1 + mn \cdot k + \frac{mn(mn+1)}{2} \end{aligned}$$

and since taking edges $e_1, \ldots, e_k$ in that order yields a solution to MRT on G and T with objective value $\sum_{i=1}^{k} i \sum_{e \in (C_i \setminus \{e_i\}) \setminus \bigcup_{j=1}^{i-1} (C_i \setminus \{e_i\})} 1$, we have that

$$mn \cdot k + \frac{mn(mn+1)}{2} \leq OBJ(\sigma) \leq \frac{(n-1)(m+1)}{2} + mn \cdot k + \frac{mn(mn+1)}{2}.$$

Since $\frac{(n-1)(m+1)}{2} < mn$, any other solution to MRT on G' and T' that uses edges $f_1, \ldots, f_{k^*} \subseteq E$ to cover T, where $k^* < k$, will have an objective value less than $OBJ(\sigma)$. Therefore the optimum

Algorithm 1 (Greedy) Greedy algorithm for MRT [19]

Input: Network $G = (V, E)$; spanning tree T over V; probabilities of failure $p(e)$ for each edge $e \in T$.
Output: An ordering $\sigma_1, \ldots, \sigma_m$ of $S = E \setminus T$.
Initialize: Set of uncovered edges $U = E$.
for i from 1 to $|S|$ **do**
  Set $\sigma_i = \text{argmax}_{s \in S \setminus \{\sigma_1, \ldots, \sigma_{i-1}\}} \sum_{e \in U : s \text{ covers } e} b(e)$.
  Set $U = U \setminus \{e \in U : \sigma_i \text{ covers } e\}$
return $\sigma_1, \ldots, \sigma_m$

solution to MRT on G' and T' must use the fewest edges $\{f_1, \ldots, f_{k^*}\}$ possible to cover T, and therefore $\{f_1, \ldots, f_{k^*}\}$ is an optimum solution to Tree Augmentation on G and T. Since G' and T' have size polynomial in the size of G and T, this gives a reduction of Tree Augmentation to MRT, so MRT is also NP-hard. $\square$

4 Kernel-based rounding for c-uniform MRT instances

Since an instance of MRT is an instance of MSSC, one may apply the greedy algorithm for MSSC [19]. For each i from 1 to $|S|$, the greedy algorithm chooses

$$\sigma_i = \underset{s \in S}{\text{argmax}} \sum_{\substack{e \in T \\ s \text{ covers } e \\ \sigma_j \text{ does not cover } e, \forall j < i}} b(e),$$

that is, σ_i is the edge in S which covers the uncovered tree edges with greatest total weight. Pseudocode appears in Algorithm 1. In [19], Feige et al. showed that the greedy algorithm is a 4-approximation for general MSSC, and that this bound is tight.

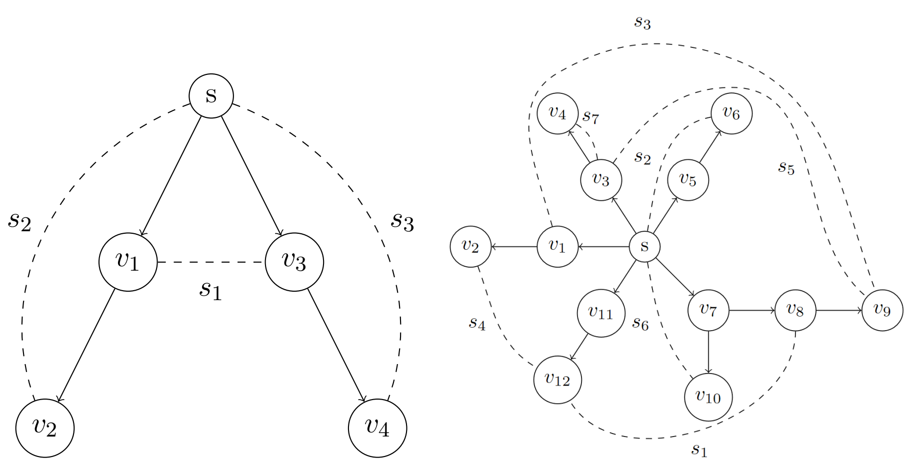

Figure 3: Two networks where the greedy algorithm for MRT does not achieve the optimum value.

However, for MRT we construct instances with a provable gap of 4/3. We suspect that the greedy algorithm achieves a strictly less than 4 approximation ratio on MRT instances.

Lemma 2. *There exist instances of MRT where the greedy algorithm does not find an optimal solution.*

Example 1: Consider the leftmost network in Figure 3. Edges in T are shown solid, while edges in S are dashed. Let $\varepsilon = 1$ and let $p(e) = 1$ for all $e \in T$. Then an optimal solution to MRT on this is to take the edges of S in order $\sigma = (s_2, s_3, s_1)$, yielding objective value 3/2. However, the greedy algorithm, breaking ties lexicographically, will take $\sigma = (s_1, s_2, s_3)$, yielding objective value 7/4, a multiplicative gap of 7/6.
Example 2: Another instance where the greedy algorithm is not optimal for MRT is shown on the right in Figure 3. Again, let $\varepsilon = 1$ and let $p(e) = 1$ for all $e \in T$. Then an optimal solution to MRT on this is to take $\sigma = (s_4, s_5, s_2, s_6, s_7, s_1, s_3)$, yielding objective value 27/12. However, the greedy algorithm, breaking ties lexicographically, will take $\sigma = (s_1, s_2, s_3, s_4, s_5, s_6, s_7)$, yielding an objective value of 36/12, a 4/3 multiplicative gap to the optimum. Note that in this network, the edge e between v_3 and v_4 is covered only by edge s_7, so any optimal ordering on S can take s_7 as the last edge σ_i that covers an edge not covered by $\sigma_1, \ldots, \sigma_{i-1}$. In this sense, the existence of e does not make solving the MRT problem any more difficult. However, if e is removed from G, the multiplicative gap actually shrinks, to 29/22.

We next show how to improve the approximation ratio for Min Sum Set Cover (including the minimum reconnection time problem) for a broad range of instances by extending a recent technique of Bansal et al. [5] that uses a kernel-based rounding approach. They transform a solution of an LP relaxation to minimum sum set cover (MSSC) to an integral order by applying a specially chosen kernel and applying randomized rounding to the kernelized point. They also show that this method gives another 4-approximation for the general MSSC problem and a 16/9-approximation for the Min Sum Vertex Cover problem, in which each edge has two vertices. In the parlance of MRT, this is equivalent to each tree edge being covered by at most two switches.

We extend their results to c-uniform hypergraphs for all $c \geq 3$, a condition which is meaningful in our electricity distribution network setting. By analogy to the zones of protection used in typical protection system designs where three-zone impedance relays are often used [6, 23], we expect that tree edges are usually covered by a small number of switches.

Our approximation algorithm can always be used in general, for any instance of MRT, and the guarantees we show bridge the gap between the approximation ratio of $16/9 \approx 1.778$ (for MSVC) and 4 (for the general case) as c increases. To reflect the more general setting, for the remainder of this section we use the hypergraph formulation of MSSC, defined above.

The α-point rounding method begins by solving an LP relaxation of an integer program formulation of MSSC. The formulation we give differs from that of [5], since we found this to be computationally more efficient, but otherwise the formulation does not affect the algorithm.[5]

$$
\begin{aligned}
\text{minimize} \quad & \sum_{t,e} t b(e) u_{e,t} \\
\text{subject to} \quad & \sum_{v:C_v \ni e} x_{v,t} - u_{e,t} \geq 0 \qquad \forall t \text{ time periods}, e \text{ edges}, \\
& \sum_{t} u_{e,t} = 1 \qquad \forall e \text{ edges}, \\
& \sum_{v} x_{v,t} = 1 \qquad \forall t \text{ time periods}, \\
& \text{all } u_{e,t}, x_{v,t} \geq 0 \text{ for vertices } s, \text{ time } t \text{ and edges } e, \text{ (binary in IP formulation).}
\end{aligned}
$$

[5] Our LP is polynomial size and uses the same variables $x_{v,t}$ to denote whether vertex v is chosen at time step t, as in Bansal et al. [5].

Algorithm 2 KERNEL α-POINT ROUNDING FOR MSSC [5]

Input: Hypergraph $H = (V, E)$; kernel $K(t, t')$.
Output: An ordering $\sigma_1, \dots, \sigma_{|V|}$ of V.
$x \leftarrow$ Optimal fractional solution for MRT.
$z_{s,t} \leftarrow \sum_{t'} K(t, t') x_{v,t'}$ $\forall v \in V, 1 \le t \le |V|$
for $v \in V$ **do**
    Sample $\alpha_v \sim [0, 1]$.
    $\tau_v \leftarrow$ the earliest time t for which $\sum_{t' \le t} z_{v,t'} \ge \alpha_v$
return an ordering σ of V, scheduling vertices according to τ and breaking ties at random.

Here each vertex $v \in V$ covers an associated set of edges C_v. In the IP, indices $t \in [|V|]$ denote the index in the selection order σ, so that $\sigma_i = v$ if and only if $x_{v,t} = 1$. Variables $u_{e,t}$ denote the time t at which edge e is covered. Each edge e must be covered at some time t, and at each time t, at most one vertex can be chosen.

Having solved the LP relaxation, which can be done in polynomial time, we apply a kernel $K(t, t')$ to the solution x to obtain new weights z. Applying an α-point randomized rounding, we obtain a provisional schedule of selection times τ for all vertices v, then convert this list to our desired ordering σ by taking the vertices in the order in which they appear in τ, breaking ties at random. Pseudocode for this algorithm appears in Algorithm 2.

In the 2-uniform hypergraph case (Min Sum Vertex Cover), Bansal et al. showed that α-point rounding is a 16/9-approximation by using the kernel

$$4 \frac{t'(t'+1)}{t(t+1)(t+2)} \cdot \mathbb{1}[t \ge t'].$$

Moreover, they show that the kernel $2/t \cdot \mathbb{1}[t \ge t']$ gives a 4-approximation for general MSSC, which is best possible. We show that one can interpolate between these two cases using a different family of kernels. For $c \ge 3$, we show that α-point rounding with the kernel

$$K(t, t') = \frac{2c}{c+1} \frac{t'^{2/(c-1)}}{\sum_{i=1}^{t} i^{2/(c-1)}} \cdot \mathbb{1}[t \ge t'].$$

is a $(\frac{2c}{c+1})^2$ approximation for Min Sum Set Cover on c-uniform hypergraphs. Note that as $c \to \infty$, $K(t, t')$ converges to $2/t \cdot \mathbb{1}[t \ge t']$, the kernel used by Bansal et al. for the general case of MSSC. At $c = 2$, the kernels are similar, with the same constant factor and a quadratic in t' divided by a cubic in t, but the two polynomials use slightly different discretizations of t'^2 and t^3. In addition, the second part of our analysis requires that $c - 1 \ge 2$.

The overall structure of the analysis is borrowed from Bansal et al.'s proof that yields a 16/9 approximation for Min Sum Vertex Cover. However, there are additional challenges due to the new kernel, particularly in the analysis of the gap in expected cover time between the LP objective and scheduled time τ. In particular, our proof of lower bounds on the total cumulative weight assigned to the vertices of an edge e requires additional convex analysis, including two more general inequalities (lemmas 3 and 6).

Theorem 2. *Let $H = (V, E)$ be a c-uniform hypergraph, where $c \ge 3$. Then applying α-point rounding with kernel*

$$K(t, t') = \frac{2c}{c+1} \frac{t'^{2/(c-1)}}{\sum_{i=1}^{t} i^{2/(c-1)}} \cdot \mathbb{1}[t \ge t'],$$

is a polynomial-time $(\frac{2c}{c+1})^2$ approximation for the Min Sum Set Cover problem on H.

Proof. As defined in the algorithm, let $z_{s,t} = \sum_{t'} K(t,t')x_{s,t'}$ be the weight assigned for any vertex s and time period t after rounding. Fix a hyperedge $e \in T$. Let $e = \{v_1, \dots, v_c\}$. Let $z_{e,t} = \sum_{i=1}^{c} z_{v_i,t}$ be the total weight of vertices covering e at exactly time t, after applying the kernel K to the LP solution x, and let $z_{e,<t} = \sum_{t'<t} z_{e,t'}$. Let $q_t(e) = \mathbb{P}[e \text{ is not scheduled before time } t \text{ in } \tau]$, and define $p_t(e) = ((1 - z_{e,<t}/c)^c)_+$, which we will show is an upper bound on the former probability.

Define the cost of covering $e \in T$ in the LP ($c_x(e)$), a bound on the time e is scheduled in the tentative order τ ($c_z(e)$), and the cost of covering e in the final order σ ($c_\sigma(e)$) as follows:

(i) LP cost: $c_x(e) = \sum_t (1 - \sum_{t'<t} \sum_{i=1}^{c} x_{v_i,t'})_+$, where $(\cdot)_+ = \max\{0, \cdot\}$.

(ii) Upper bound on expected schedule time in τ: $c_z(e) = \sum_t p_t(e)$.

(iii) Cost in σ: $c_\sigma(e) = \mathbb{E}[\min\{t : \sigma_t \in \{v_1, \dots, v_c\}\}]$.

Note that the LP objective is $\sum_{e\in T} c_x(e)$ and the objective value of the ordering returned by α-point rounding is $\sum_{e\in T} c_\sigma(e)$. Therefore, it suffices to show that $(\frac{2c}{c+1})^2 c_x(e) \geq c_\sigma(e)$. To do so, we separately show that $(\frac{2c}{c+1})c_x(e) \geq c_z(e)$ and $(\frac{2c}{c+1})c_z(e) \geq c_\sigma(e)$. We begin with the latter.

Part A. We would like to show first that $(\frac{2c}{c+1})c_z(e) \geq c_\sigma(e)$. The following argument can be generalized to a variety of kernels for α-point rounding.[6] We apply it to our choice of kernel with $\beta = \frac{2c}{c+1}$.

1. **For the kernel $K(t',t) = \frac{2c}{c+1} \frac{t'^{2/(c-1)}}{\sum_{i=1}^{t} i^{2/(c-1)}} \cdot \mathbb{1}[t \geq t']$, $\sum_v z_{v,t} \leq 2c/(c+1) = \beta$.**

   We have that for any t,

$$\begin{aligned}\sum_v z_{v,t} &= \sum_v \sum_{t'} K(t,t') x_{v,t} = \sum_{t'} K(t,t') \sum_v x_{v,t} \\ &\leq \sum_{t'} K(t,t') = \frac{2c}{c+1} \sum_{t'=1}^{t} \frac{t'^{2/(c-1)}}{\sum_{i=1}^{t} i^{2/(c-1)}} = \frac{2c}{c+1} \frac{\sum_{t'=1}^{t} t'^{2/(c-1)}}{\sum_{i=1}^{t} i^{2/(c-1)}} = \frac{2c}{c+1}.\end{aligned}$$

2. **We next show that $p_t(e) = ((1 - z_{e,<t}/c)^c)_+$ upper-bounds $q_t(e)$ and is 0 for sufficiently large t.**

   For all v, we have that

$$\begin{aligned}\sum_t z_{v,t} &= \sum_t \sum_{t'} K(t,t') x_{v,t'} \\ &= \sum_{t'} x_{v,t'} \sum_t K(t,t') \\ &= \sum_{t'} x_{v,t'} \sum_{t\geq t'} \frac{2c}{c+1} \frac{t'^{2/(c-1)}}{\sum_{i=1}^{t} i^{2/(c-1)}} \\ &= \sum_{t'} x_{v,t'} \frac{2c}{c+1} t'^{2/(c-1)} \sum_{t\geq t'} \frac{1}{\sum_{i=1}^{t} i^{2/(c-1)}} \\ &> \sum_{t'} x_{v,t'} \frac{2c}{c+1} t'^{2/(c-1)} \sum_{t\geq t'} \frac{1}{\int_0^{t+1} i^{2/(c-1)} di}\end{aligned}$$

[6] In particular, claim 1 requires that for some $\beta \geq 1$, $\sum_{t'} K(t,t') \leq \beta$ for all t; claim 2 requires that for all $t' \geq 1$, $\sum_t K(t,t') \geq 1$; and claims 3, 4, and 5 do not otherwise depend on K.

$$
\begin{aligned}
&= \sum_{t'} x_{v,t'} \frac{2c}{c+1} t'^{2/(c-1)} \sum_{t \geq t'} \frac{c+1}{c-1} (t+1)^{-(c+1)/(c-1)} \\
&> \sum_{t'} x_{v,t'} \frac{2c}{c-1} t'^{2/(c-1)} \int_{t'+1}^{\infty} (t+1)^{-(c+1)/(c-1)} dt \\
&= \sum_{t'} x_{v,t'} \frac{2c}{c-1} t'^{2/(c-1)} \frac{c-1}{2} (t'+1)^{-2/(c-1)} \\
&= \sum_{t'} x_{v,t'} \cdot c \left(\frac{t'}{t'+1} \right)^{2/(c-1)} \\
&\geq \sum_{t'} x_{v,t'} \cdot 3(1/2)^1 \\
&> \sum_{t'} x_{v,t'} \cdot 1 \\
&= 1.
\end{aligned}
$$

where the last equality follows since without loss of generality we may assume that in the solution of the linear program for MSSC, each vertex v is selected with total weight 1 across all time steps.

For all v, since $\sum_{t=1}^{\infty} z_{v,t} > 1$, for sufficiently large t we have that $z_{v,<t} = \sum_{i=1}^{t-1} z_{v,i} \geq 1$. Therefore, $z_{e,<t} = \sum_{i=1}^{c} z_{v_i,<t} \geq c$ for sufficiently large t, and so $p_t(e) = 0$ for sufficiently large t.

Moreover, by the arithmetic mean-geometric mean inequality, we have that $q_t(e)$ is bounded above by $p_t(e)$, since

$$
q_t(e) = \prod_{i=1}^{c} (1 - z_{v_i,<t}) \leq \left(1 - \sum_{i=1}^{c} z_{v_i,<t}/c \right)_+^c = p_t(e).
$$

3. **Upper bound the expected cost of e in the final ordering σ:** We claim that

$$
\mathbb{E}[c_\sigma(e)] \leq \sum_t (q_t(e) - q_{t+1}(e))(1 + \beta(t-1) - z_{e,<t} + (\beta - z_{e,t})/2). \tag{2}
$$

Each term of this sum corresponds to the event that e is scheduled at exactly time t in τ. This happens with probability $q_t(e) - q_{t+1}(e)$. The quantity $S(t) = 1 + \beta(t-1) - z_{e,<t} + (\beta - z_{e,t})/2$ is an upper bound on the expected number of vertices selected in σ before and including the one that covers e. By Claim 1, in this quantity:

- 1 corresponds to the vertex which covers e.
- $\beta(t-1) - z_{e,<t}$ is an upper bound on the expected number of vertices scheduled in τ before e is covered. Indeed, this quantity is $\sum_{v \notin e} z_{v,<t} \leq \beta(t-1) - z_{e,<t}$.
- In expectation, at most β vertices are scheduled at exactly time t in τ, of which $(\beta - z_{e,t})$ do not cover e. In expectation, at most half of these vertices will be scheduled in σ before e is covered, yielding the bound of $(\beta - z_{e,t})/2$, since ties are broken at random.

$S(t)$ is non-decreasing with respect to t. For each t, replacing $q_t(e)$ with $p_t(e)$ increases the right-hand side of Equation (2) by $(p(t) - q(t))(S(t+1) - S(t)) > 0$. Therefore, since $p_t(e)$ is positive for a finite number of t (Claim 2), replacing all $q_t(e)$ with the corresponding $p_t(e)$ can only increase the sum, and therefore

$$\mathbb{E}[c_\sigma(e)] \le \sum_t (p_t(e) - p_{t+1}(e))(1 + \beta(t-1) - z_{e,<t} + (\beta - z_{e,t})/2).$$

Note that $\mathbb{E}[c_z(e)] = \sum_t p_t(e) = \sum_t t(p_t(e) - p_{t+1}(e))$, so $\mathbb{E}[c_\sigma(e)] \le \beta\, \mathbb{E}[c_z(e)]$ is implied by each of the following inequalities. We prove the last of these in Claim 4.

$$\begin{aligned}
\sum_t (p_t(e) - p_{t+1}(e))(1 + \beta(t-1) - z_{e,<t} + (\beta - z_{e,t})/2) &\le \beta \sum_t t(p_t(e) - p_{t+1}(e)) \\
\sum_t (p_t(e) - p_{t+1}(e))(1 - z_{e,<t} - z_{e,t}/2) &\le \beta/2 \sum_t (p_t(e) - p_{t+1}(e)) \\
(1 - \beta/2) \sum_t (p_t(e) - p_{t+1}(e)) &\le \sum_t (p_t(e) - p_{t+1}(e))(z_{e,<t} + z_{e,t}/2) \\
1 - \beta/2 &\le \sum_t (p_t(e) - p_{t+1}(e))(z_{e,<t} + z_{e,t}/2) \\
1/2 &\le \sum_t (z_{e,t}/2)(p_t(e) + p_{t+1}(e)) \\
1 &\le \sum_t z_{e,t}(p_t(e) + p_{t+1}(e)).
\end{aligned}$$

4. **Lower-bounding $\sum_t z_{e,t}(p_t(e) + p_{t+1}(e))$.**

   Note that $p_t(e)$ is a convex function of $z_{e,<t}$, as it is 0 when $z_{e,<t} \ge c$. Therefore, for all $a < b$ we have that $(b-a)(p_t(a) + p_t(b)) \ge 2\int_a^b p_t(u)\, du$. In particular, for all t we have that

$$z_{e,t}(p_t(e) + p_{t+1}(e)) = (z_{e,t+1} - z_{e,t})(p_t(e) + p_{t+1}(e)) \ge 2 \int_{z_{e,<t}}^{z_{e,<t+1}} p_u(e)\, du$$

   and so

$$\sum_t z_{e,t}(p_t(e) + p_{t+1}(e)) \ge 2 \sum_t \int_{z_{e,<t}}^{z_{e,<t+1}} p_u(e)\, du = 2\int_0^c (1 - u/c)^c\, du = 2c/(c+1) > 1.$$

5. **Final claim:** $(2c/(c+1))c_z(e) \ge c_\sigma(e)$

   Claim 3 holds using $\beta = 2c/(c+1)$ (claim 1). Applying claim 4, we have that $(2c/(c+1))c_z(e) \ge c_\sigma(e)$ as desired.

Part B. We would now like to show that $(2c/(c+1))c_x(e) \ge c_z(e)$.
For all vertices v, we have that

$$z_{v,<t} = \sum_{t'<t} z_{v,t'} = \sum_{t'<t} \sum_{t''<t'} K(t', t'') x_{v,t''} = \sum_{t''<t} x_{v,t''} \sum_{t'=t''}^{t-1} K(t', t'').$$

To analyze this sum, we use the following lemma, which we prove in Appendix A.

Lemma 3. *For $t \in \mathbb{N}$ and $0 < p \le 1$, $\sum_{i=1}^t i^p \le \frac{p}{p+1} \frac{t^p(t+1)^p}{(t+1)^p - t^p}$.*

Examining the inner sum $\sum_{q=t''}^{t-1} K(q,t'')$ more closely and applying Lemma 3 with $p = 2/(c-1)$, we have that

$$\begin{aligned}
\sum_{q'=t''}^{t-1} K(q,t'') &= \frac{2c}{c+1} t''^{2/(c-1)} \sum_{q'=t''}^{t-1} \frac{1}{\sum_{i=1}^{q} i^{2/(c-1)}} \\
&\geq \frac{2c}{c+1} t''^{2/(c-1)} \sum_{q'=t''}^{t-1} \frac{c+1}{2} \left(\frac{1}{q^{2/(c-1)}} - \frac{1}{(q+1)^{2/(c-1)}} \right) \\
&= \frac{2c}{c+1} t''^{2/(c-1)} \frac{c+1}{2} \left(\frac{1}{t''^{2/(c-1)}} - \frac{1}{t^{2/(c-1)}} \right) \\
&= c \left(1 - \frac{t''^{2/(c-1)}}{t^{2/(c-1)}} \right).
\end{aligned}$$

Thus

$$z_{v,<t} = \sum_{t''<t} x_{v,t''} \sum_{t'=t''}^{t-1} K(t',t'') \geq \sum_{t''<t} c \left(1 - \frac{t''^{2/(c-1)}}{t^{2/(c-1)}} \right) x_{v,t''}$$

and so

$$\begin{aligned}
c_z(e) = \sum_t p_t(e) &= \sum_t \left(1 - \sum_{i=1}^{c} z_{v_i,<t}/c \right)_+^c \\
&\leq \sum_t \left(1 - \sum_{t'<t} \left(1 - \frac{t'^{2/(c-1)}}{t^{2/(c-1)}} \right) \sum_{i=1}^{c} x_{v_i,t'} \right)_+^c .
\end{aligned}$$

We now modify the LP solution x in order to bound $c_z(e)$. These modifications do not decrease our lower bound $c_z(e)$ and do not increase the LP cost $c_x(e)$. First, decrease x as necessary so that $\sum_{i=1}^{c} x_{v_i,<t} = \sum_{t'<t} \sum_{i=1}^{c} x_{v_i,t'} \leq 1$ for all t. Second, for all t, replace each $x_{v_i,t}$ by their average value $a_t = \sum_{i=1}^{c} x_{v_i,t}/c$. As $c_z(e)$ depends only on the total $\sum_{i=1}^{c} x_{v_i,t}$ and not the distribution of weight between these vertices, this does not change $c_z(e)$. Then our goal is to show that

$$c_z(e) \leq \sum_t \left(1 - \sum_{t'<t} \left(1 - \frac{t'^{2/(c-1)}}{t^{2/(c-1)}} \right) \sum_{i=1}^{c} x_{v_i,t'} \right)_+^c \leq \frac{2c}{c+1} \sum_t \left(1 - \sum_{t'<t} a_{t'} \right)^c = \frac{2c}{c+1} c_x(e)$$

for all nonnegative a with $||a||_1 = 1$.

Note that for each t, the corresponding summand $(1 - d \cdot a)^c$ is positive and a convex function of a, since the vector d given by $(d)_{t'} = t'/t \cdot \mathbb{1}[t' < t]$ satisfies $||d||_\infty \leq 1$. Therefore $c_z(e)$ is a convex function of a. Moreover, $c_x(e)$ is a linear function of a. Therefore, by Fact 8 of [5], the quotient $c_z(e)/c_x(e)$ is maximized at an extreme point a of the positive simplex, i.e. for some u, $a_u = 1$ and $a_{u'} = 0$ for $u' \neq u$. At this point a, we have that $2c/(c+1)c_x(e) = 2c/(c+1)u$ and

$$\begin{aligned}
c_z(e) &= \sum_t \left(1 - \sum_{t'<t} \left(1 - \frac{t'^{2/(c-1)}}{t^{2/(c-1)}} \right) a_{t'} \right)^c \\
&= \sum_{t=1}^{u} \left(1 - \sum_{t'<t} \left(1 - \frac{t'^{2/(c-1)}}{t^{2/(c-1)}} \right) a_{t'} \right)^c + \sum_{t>u} \left(1 - \sum_{t'<t} \left(1 - \frac{t'^{2/(c-1)}}{t^{2/(c-1)}} \right) a_{t'} \right)^c \\
&= u + \sum_{t>u} \left(1 - \left(1 - \frac{u^{2/(c-1)}}{t^{2/(c-1)}} \right) \right)^c
\end{aligned}$$

$$
\begin{aligned}
&\leq u + \int_u^\infty (u/t)^{2c/(c-1)} dt \\
&= u + \frac{1}{2c/(c-1) - 1} u \\
&= u + \lim_{b\to\infty} \left(-\frac{c-1}{c+1} u^{2c/(c-1)} t^{-(c+1)/(c-1)} \right) \Big|_u^b \\
&= u + \frac{c-1}{c+1} u^{2c/(c-1)} u^{-(c+1)/(c-1)} \\
&= 2c/(c+1)u
\end{aligned}
$$

so $c_z(e) \leq 2c/(c+1)c_x(e)$ as desired. $\square$

We show that the approximation guarantee for c-uniform hypergraphs extend to all hypergraphs H in which each edge contains at most c vertices. In the MRT problem, this corresponds to networks in which each tree edge is covered by at most c switches.

Corollary 1. *Let $H = (V, E)$ be a hypergraph, each of whose edges contains at most c vertices. Then there is a polynomial-time $(\frac{2c}{c+1})^2$-approximation algorithm for the Min Sum Set Cover problem on H.*

Proof. For each edge e with fewer than c vertices, modify H by adding additional dummy vertices belonging only to e, until e has exactly c vertices. Solve the LP on the resulting c-uniform hypergraph H'. Note that the objective value of the LP on H' is the same as the objective value of the LP on the original hypergraph H, since assigning all weight given to a dummy vertex at a given time to a non-dummy vertex which covers the same edge cannot increase the LP objective.

Apply α-point rounding with kernel

$$
K(t, t') = \begin{cases} 4\frac{t'(t'+1)}{t(t+1)(t+2)} \cdot \mathbb{1}[t \geq t'], c = 2 \\ \frac{2c}{c+1} \frac{t'^{2/(c-1)}}{\sum_{i=1}^t i^{2/(c-1)}} \cdot \mathbb{1}[t \geq t'], c \geq 3 \end{cases}
$$

to the LP solution on H'. By Theorem 2, or Theorem 22 of [5] if $c = 2$, the expected objective value of the solution σ' produced by α-point rounding is at most $(\frac{2c}{c+1})^2$ times the LP objective.

Finally, convert σ' into an ordering σ of the vertices of H as follows. For each dummy vertex in σ', exchange its place in the ordering with a non-dummy vertex later in the ordering which covers the same edge. When no more such exchanges can be made, remove the dummy vertex from the ordering. This procedure cannot increase the time at which each edge e is covered, and hence cannot increase the objective value. Therefore, the expected objective value of the resulting ordering σ of the vertex set V is at most $(\frac{2c}{c+1})^2$ times the LP objective, as desired. $\square$

Next, we show that this approximation ratio is tight. The following result generalizes Lemmas 26 and 27 of [5] on the integrality gap for MSVC.

Theorem 3. *Let $c \geq 2$. Then the integrality gap of the LP for MSSC on c-uniform hypergraphs is at least $(\frac{2c}{c+1})^2$.*

Proof. For a suitable $k = \omega(1)$ and $i = 1, \ldots, k$, let $n_i = Ni^{-\alpha}$ where $\alpha = 2/(c+1) + \varepsilon$, with ε approaching 0 and N large enough that n_i can be rounded without affecting the solution. Let H be the hypergraph consisting of disjoint copies of the complete c-uniform hypergraphs K_i on n_i vertices.

We first upper-bound the LP cost of MSSC on H. Setting

$$x_{v,t} = \begin{cases} \frac{1}{n_i}, \text{ if } v \in K_i \text{ and } \sum_{j<i} \frac{n_j}{c} < t \leq \sum_{j\leq i} \frac{n_j}{c} \\ 0, \text{ otherwise.} \end{cases}$$

gives a feasible solution. In this solution, each edge e in K_i is completely covered by time $(n_1 + \cdots + n_i)/c$, and so the LP objective is at most

$$\begin{aligned} \sum_{i=1}^{k} \binom{n_i}{c} \frac{n_1 + \cdots + n_i}{c} &\leq \sum_{i=1}^{k} \frac{N^c}{c!} i^{-c\alpha} \frac{N}{c} \sum_{j=1}^{i} j^{-\alpha} \\ &\leq \sum_{i=1}^{k} \frac{N^{c+1}}{c!c} i^{-c\alpha} \frac{i^{1-\alpha}}{1-\alpha} \\ &\leq \frac{N^{c+1}}{c!c(1-\alpha)} \sum_{i=1}^{k} i^{1-(c+1)\alpha} = \frac{N^{c+1}}{c!c(1-\alpha)} \sum_{i=1}^{k} i^{-1-(c+1)\varepsilon}. \end{aligned}$$

As we make ε arbitrarily small, $\sum_{i=1}^{k} i^{-1-(c+1)\varepsilon} \leq 1 + \int_1^\infty x^{1-e\varepsilon}\, dx = 1 + 1/(c+1)\varepsilon$. Thus, the objective of the LP solution is at most

$$\frac{N^{c+1}}{c!c(1-\alpha)} \sum_{i=1}^{k} i^{-1-(c+1)\varepsilon} \leq \frac{N^{c+1}}{c!c(\frac{c-1}{c+1} - \varepsilon)} \frac{1}{(c+1)\varepsilon}.$$

This gives a bound of approximately $N^{c+1}/c!c(c-1)\varepsilon$ for the LP solution for sufficiently small ε and large k.

Next, we show that the greedy algorithm gives an optimum solution to MSSC on H, and lower-bound the cost of this solution.

In any solution σ to MSSC, the jth vertex selected from K_i covers $\binom{n_i-j+1}{c}$ hyperedges that had not been covered before. Therefore, the order of vertices chosen within each K_i does not matter, and permuting the order in which vertices from each K_i are chosen relative to each other does not change the number of hyperedges those vertices cover. Thus the greedy algorithm, which at each time step picks a vertex from one of the K_i with the greatest number of unpicked vertices, produces an optimal solution.

For any $1 \leq i \leq k-1$, consider the $i(n_i - n_{i+1})$ time steps during which the greedy algorithm selects vertices from $K_1, \ldots, K_i$ until n_{i+1} vertices remain in each. During each of these time steps, there are $\binom{n_{i+1}}{c}$ uncovered edges in each of $K_1, \ldots, K_i$, and $\binom{n_j}{c}$ uncovered edges in each K_j for $j \geq i+1$. Therefore, the objective cost is at least

$$\sum_{i=1}^{k-1} i(n_i - n_{i+1}) \left(i \cdot \binom{n_{i+1}}{c} + \sum_{j=i+1}^{k} \binom{n_j}{c} \right).$$

Since $\binom{n_i}{c} \approx n_i^c/c!$, we have that

$$i \cdot \binom{n_{i+1}}{c} + \sum_{j=i+1}^{k} \binom{n_j}{c} \approx \frac{N^c}{c!} \left(i^{1-c\alpha} + \int_i^k j^{-c\alpha}\, dj \right) = \frac{N^c}{c!} \left(i^{1-c\alpha} + \frac{1}{c\alpha - 1}(i^{1-c\alpha} - k^{1-c\alpha}) \right).$$

Using this and $n_i - n_{i+1} \approx N\alpha i^{-1-\alpha}$, the objective is at least

$$\frac{N^{c+1}}{c!} \sum_{i=1}^{k} i \cdot \alpha i^{-1-\alpha} \left(\frac{c\alpha}{c\alpha - 1} i^{1-c\alpha} + \frac{1}{c\alpha - 1} \right) = \frac{N^{c+1}}{c!} \left(\sum_{i=1}^{k} \frac{c\alpha^2}{c\alpha - 1} i^{1-(c+1)\alpha} + \frac{\alpha k^{1-c\alpha} \sum_{i=1}^{k} i^{-\alpha}}{c\alpha - 1} \right).$$

In this expression, as k increases and ε decreases, the first term goes to

$$\frac{N^{c+1}}{(c-1)!}\frac{\alpha^2}{c\alpha-1}\frac{1}{(c+1)\alpha-2}\approx\frac{N^{c+1}}{(c-1)!}\frac{2^2}{(c+1)^2\frac{c-1}{c+1}}\frac{1}{(c+1)\varepsilon}=\frac{4N^{c+1}}{(c-1)!(c-1)\varepsilon(c+1)^2}.$$

Combining the two objective bounds yields the desired integrality gap of

$$\frac{\frac{4N^{c+1}}{(c-1)!(c-1)\varepsilon(c+1)^2}}{\frac{N^{c+1}}{c!c(c-1)\varepsilon}}=\left(\frac{2c}{c+1}\right)^2. \qquad \square$$

Note that this ratio matches the approximation guarantee of Theorem 2, and approaches the general approximation ratio of 4 as $c\to\infty$.

Our results in this section are worst-case. Computationally, we find that the greedy algorithm and the α-point rounding approaches are competitive (the latter outperforms the former for some instances on synthetic networks). We include computationally observed approximation factors in Section 6.3.

5 Fair Multi-objective Minimization

Different choices of the initial spanning tree $T\subseteq E$ and reconnection order σ on $S=E\setminus T$ can have widely diverging performance on R-Time and SAIDI, as well as energy costs dependent on the choice of the tree (see, for example, the system in Figure 1). In this section, we give a local search method for choosing a spanning tree $T\subseteq E$, including bookkeeping techniques to speed up this method.

5.1 Branch exchange local search

Since we would like to simultaneously minimize all the three objectives as best as possible to decrease loss of energy and improve service reliability across all customers, our goal is to find a single choice of T and σ that achieves strong performance on all three objectives. This becomes a multi-objective optimization problem, which we model as a composite product objective

$$\text{SAIDI}\cdot\text{R-TIME}\cdot\text{Energy}=\frac{\sum_{v\in V}w(v)\sum_{e:v\in d(e)}p(e)t(e)}{\sum_{v\in V}w(v)}\cdot\frac{\sum_{e\in T}p(e)t(e)}{\sum_{e\in T}p(e)}\cdot\sum_{e\in T}R(e)f(e)^2,$$

by using a well-known local search heuristic, called *branch exchange*, on spanning trees.

If the objective function is chosen to be linear in the weights $b(e)$ of the edges, for instance if it is a linear combination of the R-Time and SAIDI objectives, then it is possible to find an optimal spanning tree and reconnection order using an integer programming formulation (see Appendix B.) However, this program is large and solving it is intractable for larger networks.

The branch exchange search, Algorithm 3, takes as input a network G with initial spanning tree T_0 and an objective function $F(T,\sigma)$, which can be a combination of multiple metrics on T (such as energy) and σ (such as R-Time and SAIDI). In each iteration, the local search tests a random pair $(e,s)\in T\times S$ that is feasible (i.e., $T'=T-e+s$ is still connected.) This step, exchanging tree and non-tree edges, gives the algorithm its name. The algorithm then generates an order σ' on the new set of switches $S+e-s$ and compares the new objective value $F(T',\sigma')$ to the current value $F(T,\sigma)$. If the objective is improved, the branch exchange is accepted; otherwise, another random pair (e,s) is sampled and this process is repeated. The algorithm stops when no feasible branch exchange improves the objective value.

Algorithm 3 Branch exchange local search

Input: Network $G = (V, E)$; initial spanning tree T_0 over V and order σ_0 over $S = E \setminus T$; probabilities of failure $p(e)$ for each edge $e \in T_0$; vertex weights $w(v)$ for each $v \in V$; objective function $F(T, \sigma)$.
Output: A spanning tree T over V; a reconnection order σ over $E - T$.
$T \leftarrow T_0$, $\sigma \leftarrow \sigma_0$, $P \leftarrow T \times (E - T)$
while $|P| > 1$ **do**
    $(e, s) \sim P$
    $P \leftarrow P \setminus \{(e, s)\}$
    $T' \leftarrow T - e + s$
    Find a reconnection order σ' using greedy or other algorithms for metrics used in F.
    if $F(T', \sigma') < F(T, \sigma)$ **then**
        $T \leftarrow T', P \leftarrow T \times (E - T)$
return T, σ

Note that the component of the composite product objective that depends on σ', SAIDI $\cdot$ R-Time, can no longer be written an instance of MRT. Therefore, we use a version of the greedy algorithm to choose σ', where at each time step i we choose the switch s which achieves the greatest marginal increase in the product of the weights $p(e)f(e)$ and $p(e)$ used for SAIDI and R-Time, respectively.

5.2 Efficient updates

Given a network $G = (V, E)$, a substantial part of the branch exchange local search for finding a tree T that achieves strong performance for metrics modelled using MRT is updating the connectivity information, i.e. for which switches s and tree edges e the tree $T - e + s$ remains connected. During the local search procedure, T and $S = E \setminus T$ will evolve, requiring connectivity updates at each step. Indeed, updating this information, which is input to the greedy algorithms for MRT, is significantly more computationally expensive than executing those algorithms, computing energy costs, and comparing the objective values to evaluate whether to accept a branch exchange. This information is also useful in filtering and efficiently choosing (at random) feasible branch exchange pairs (e, s).

In this section, we give a description of those pairs of tree edges and switches whose connectivity data might change after a branch exchange update. Updating only these pairs improves the computational efficiency of this step significantly: by a factor of $\Omega(\min\{|E - T|, |V|/d\})$, where d is the circumference, or length of a longest cycle in G.

Consider a branch exchange update on $T \subseteq E$. This update replaces T with $T' = T - e + s_e$, where $e \in T$, $s_e \in E - T$, and T' is connected. That is, it exchanges the tree edge e for the switch s_e, yielding the new spanning tree T'. To compute a reconnection order on the new set of switches $E - T'$, whether by the greedy algorithm or an integer program, it is necessary to know the *covering data* of G and T', i.e. for which pairs (f, s) of edges $f \in T'$, $s \in E - T'$ the graph $T' - f + s$ is connected. If we have already computed the covering data for G and T, to update this list for T' it suffices to consider only those pairs (f, s) excluded by the following lemma.

Lemma 4. *Let $G = (V, E)$ be a graph, let $T \subseteq E$ be a spanning tree of G. Consider an edge $e \in T$ that is deleted, and some edge $s_e \in E - T$ which reconnects it into another spanning tree $T' = T - e + s_e$. Then, all covering data for T also hold for the new spanning tree T', except the following pairs (f, s):*

1. *f is in the unique cycle C_e in $T + s_e$ and the unique cycle C_s in $T + s$ contains e.*
2. *f is in the unique cycle C'_e in $T' + e$ and the unique cycle C'_s in $T' + s$ contains s_e.*

In other words, $T - f + s$ is connected if and only if $T' - f + s$ is connected, for all other such pairs of edges (f, s).

To prove this result, we show a second lemma:

Lemma 5. *Let $G = (V, E)$ be a graph and let $T \subseteq E$ be a spanning tree of G. Let $s_e, s_f \in S = E - T$ and let C_e, C_f be the unique cycles in $T + s_e, T + s_f$, respectively. Let $e \in C_e \cap T$ and $f \in C_f \cap T$. Then if $e \notin C_f$ or $f \notin C_e$, $T - e - f + s_e + s_f$ is connected.*

Proof. By symmetry, it suffices to show the result when $e \notin C_f$.

By hypothesis, $T - e + s_e$ is a spanning tree on G. Let u and v be the two vertices incident to f. Then $T - e - f + s_e$ has two connected components, one containing u and one containing v. Therefore, it suffices to show that there is a path between u and v in $T - e - f + s_e + s_f$. Indeed, $C_f - f$ is such a path, since $e \notin C_f$. Thus $T - e_f + s_e + s_f$ is connected. □

Proof of Lemma 4. Suppose neither of the conditions holds.

If $T - f + s$ is connected, $f \in C_s$. Since condition 1 does not hold, either $e \notin C_s$ or $f \notin C_e$. Therefore, by Lemma 5, $T - e - f + s_e, s = T' - f + s$ is connected.

If $T' - f + s$ is connected, $f \in C'_s$. Since condition 2 does not hold, either $f \notin C'_e$ or $s_e \notin C'_s$. Therefore, by Lemma 5, $T' - s_e, f + e, s_f = T - f + s$ is connected.

Therefore $T - f + s$ is connected if and only if $T' - f + s$ is connected. □

In general, computing the connectivity of $T - f + s$ for all pairs (f, s) of edges $f \in T$, $s \in E - T$ takes $O(|V|^2|E - T|)$ time, for instance by performing a depth-first search on each of the $|V||E - T|$ graphs $T - f + s$. Let d be the circumference (length of longest cycle) of G. Then by updating connectivity only for those pairs excluded in Lemma 4, the time needed to update connectivity for $T' = T - e + s_e$ is reduced to

$$\begin{aligned} O(|V|(|E - T| + |V| + |C_e||E - T| + |C'_e||E - T|) &= O(|V|(|V| + d|E - T|)) \\ &= O(|V|^2 + d|V||E - T|). \end{aligned}$$

Therefore, updating only pairs (f, s) as described in Lemma 4, we can save an overall factor of $\Omega(\min\{|E - T|, |V|/d\})$ runtime in each step of the local search procedure.

6 Computations

In this section, we test our methods for improving reconnection metrics on the Greensboro, NC urban-suburban synthetic network from the NREL SMART-DS project. We describe methods for reducing the size of large instances, improving tractability, and give experimental results using greedy and α-point rounding methods for MRT and our local search methods.

6.1 Greensboro Dataset

The NREL SMART-DS synthetic network covers most of the Greensboro area, including residential, industrial, and other areas, and contains 145052 buses [52]. The network also contains demand data for each bus, which we treat as the vertex weights $w(v)$, and lines between buses, whose coordinates are also given. We take the failure probability $p(e)$ of each line e to be proportional to the straight-line distance between its endpoints, as in practice, failure rates tend to be proportional to length [7]. The network contains 19 distinct connected components. Of these, 18 correspond to radial distribution networks, each with its own substation, while the other links the substations. We apply our methods to each of the 18 distribution networks, treating them as entirely separate and independent networks.

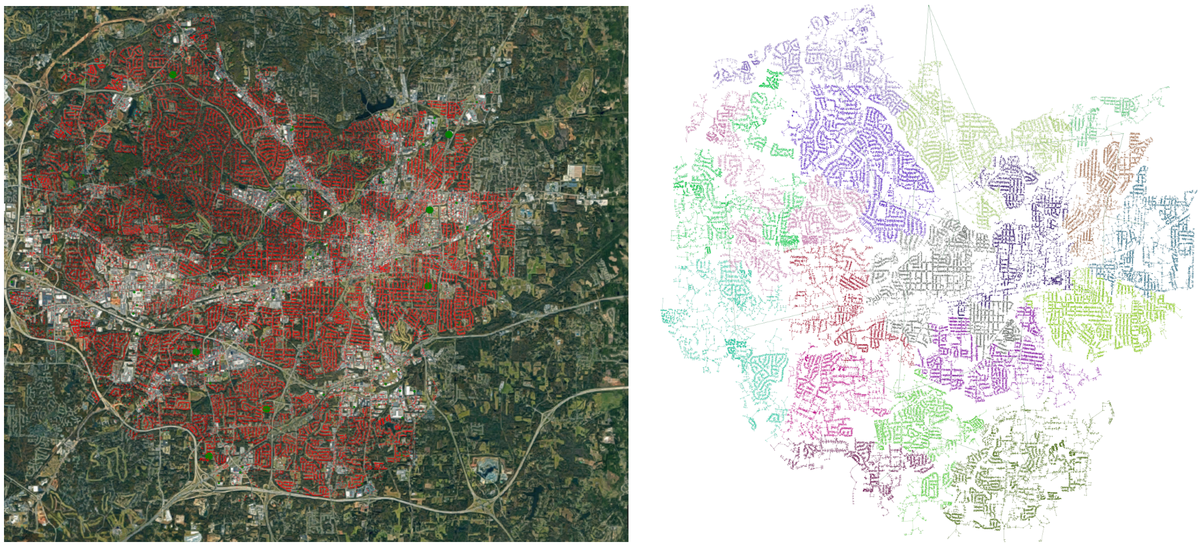

Figure 4: (left) Buses shown on a satellite image of Greensboro, NC. (right) Each color is a different connected component in the graph where all the switches (black edges) are removed.

The left image in Figure 4 shows the locations of the buses, plotted as red dots, on a satellite image of Greensboro. On the right, each of the 19 components are plotted in a distinct color, showing how they partition the city.

Components may have over 10000 vertices each. Before performing local search or other computationally intensive procedures on a component, we may wish to pre-process the network to simplify and reduce its size while retaining key information about its structure. We employ two reduction steps in this pre-processing:

1. Let v be a vertex of degree 2 in T with neighbors u and w. Then v may be removed. Add an edge between u and w, with half its weight $w(v)$ added to the demands of each of u and w.

2. Let v be a vertex of degree 1 in T with neighbor u. Then if $w(v)$ is less than some chosen threshold W, contract the edge uv in T and add $w(v)$ to $w(u)$.

These steps eliminate bridge vertices, as well as leaves corresponding to buses with low weight. Applying these steps repeatedly significantly reduces the size of the network. We give precise pseudocode in Algorithm 4.

6.2 Adding switches

The synthetic network has only 134 switches (non-spanning tree edges) across its 18 distribution networks. To increase coverage, we add additional switches, aiming to maximize coverage while minimizing the number and length of the switches. To do so, we use a two-step procedure for each contracted component. First, we generate a large set of candidate switches. To do so, we repeatedly sample random pairs of buses in the network. For each pair (u, v), we add a switch between them if the edges covered by (u, v) have total weight exceeding a given threshold (which may depend on the component) and either the distance between u and v is at most 1000m or v is one of the ten closest vertices to u. We stop when each edge is covered by at least two potential switches.

Once we have obtained a candidate set R of switches s, each of length $\ell(s)$ meters, we solve the following integer program to choose a set of switches $S \subseteq R$ that obtains good coverage while

Algorithm 4 (TREE CONTRACTION) Reducing tree size

Input: Spanning tree T over a vertex set V; probabilities of failure $p(e)$ for each edge $e \in T$; weights $w(v)$ for each $v \in V$, weight threshold W.
Output: A tree T'.
Initialize: $T' = T$.
while a vertex was deleted in the previous loop **do**
    for $v \in V$ **do**
        if $\deg_{T'}(v) = 1$ and $w(v) < W$ **then**
            Add $w(v)$ to $w(u)$, where u is v's neighbor.
            Add $p(uv) \cdot w(v)/w(u)$ to $p(uw)$, where w is u's parent. Delete v.
        if $\deg_{T'}(v) = 2$ **then**
            Add $w(v)/2$ to $w(u)$ and $w(v)$, where u and w are v's neighbors.
            Add edge uw with failure probability $p(uw) = p(uv) + p(vw)$. Delete v.
return T'.

being as inexpensive as possible to construct. We may also impose a cost on the upper bound c of the number of switches that cover each edge, guaranteeing c-uniformity. As before, for all $s \in R$, $C_s = \{e \in E : T - e + s \text{ is connected}\}$.

$$\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{s \in R} (\ell[s] + 1000)x_s + 100c & \\
\text{subject to} & \displaystyle\sum_{s : C_s \ni e} x_s \geq 1 & \forall e \text{ tree edges}, \\
& \displaystyle\sum_{s : C_s \ni e} x_s \leq c & \forall e \text{ tree edges}, \\
& \text{all } x_s \text{ binary}, c \geq 0. &
\end{array}$$

6.3 Approximations for MRT

Next, we find reconnection orders for R-TIME and SAIDI on each of the 18 component networks. To do so, we compare three different approaches for each metric:

- greedy algorithm
- α-point rounding on linear program solution
- integer program

The first two of these are polynomial time, with greedy being extremely computational inexpensive (hence our use of it in our local search algorithms) and solving the linear program taking the majority of the time for α-point rounding. On many of the components, solving the integer program was of comparable speed to α-point rounding, but on some it took much longer. Moreover, in our experiments testing α-point rounding on random synthetic instances of c-uniform MSSC, which were not MRT instances, the integer program took hours to solve even on instances with only a few hundred hyperedges.

#	Vert.	Leaves	Switches	**R-Time*** **(obj)**	SAIDI		R-Time		**SAIDI*** **(obj)**
1	8538	4643	172	11074.78	12228.49	(10.4%)	3234.76	(15.5%)	3735.90
2	7286	3834	190	9997.17	10768.01	(7.7%)	2992.30	(10.3%)	3301.31
3	6238	3247	191	12094.07	13237.28	(9.5%)	3496.99	(15.9%)	4053.05
4	10953	6021	246	16145.83	17350.67	(7.5%)	4684.22	(16.6%)	5463.90
5	5914	3156	159	9635.66	10550.89	(9.5%)	2428.36	(12.9%)	2741.28
6	13010	6814	290	14666.74	16303.69	(11.2%)	4209.04	(42.4%)	5993.68
7	8529	4773	157	12789.89	14248.85	(11.4%)	4788.10	(43.4%)	6865.31
8	6950	3748	124	9313.83	10087.89	(8.3%)	3699.10	(15.6%)	4274.54
9	8147	4415	203	12739.60	13778.00	(8.2%)	3705.09	(9.4%)	4052.61
10	5901	3119	148	7585.66	8110.98	(6.9%)	2620.81	(12.7%)	2954.22
11	7006	3899	116	10751.39	11685.46	(8.7%)	5363.95	(35.0%)	7243.42
12	8396	4402	213	14156.85	15228.94	(7.6%)	4690.84	(22.9%)	5765.44
13	2427	1388	61	5514.82	6105.68	(10.7%)	3482.84	(39.7%)	4865.76
14	9841	4951	384	15402.82	16820.65	(9.2%)	5329.88	(22.3%)	6520.59
15	9319	5035	134	9323.08	9879.60	(6.0%)	2627.71	(8.5%)	2851.67
16	11536	5862	185	15318.78	16154.60	(5.5%)	4074.36	(17.5%)	4789.30
17	11031	5793	251	13788.25	14916.54	(8.2%)	4188.35	(10.9%)	4645.18
18	4008	2229	57	5633.63	6141.12	(9.0%)	4506.09	(19.3%)	5376.74

Table 2: Graphical data and reconnection metrics for the 18 distribution networks. The Switches column contains the number of switches added by the integer program to select switches from the candidate set. For each metric, the percentage given is the relative increase in objective value from using the greedy algorithm to optimize for the other metric. Metrics are given in units of meters $\cdot\, \varepsilon$, where the length of each line e is proportional to its probability of failure $p(e)$ [7].

Table 6.3 gives statistics on the 18 distribution networks and the reconnection metrics after the greedy algorithms are applied to minimize R-Time and SAIDI, respectively. For each objective, we compute the values of both reconnection metrics. Note that in both cases, the choice of objective matters substantially. The value of R-Time is on average nearly 10% greater when SAIDI is the objective than when R-Time is the objective, and the value of SAIDI is up to 43.4% greater when greedy attempts to minimize R-Time than when its objective is SAIDI.

Figure 5 compares the performance of the three methods, expressed as a percentage difference relative to the LP objective. For each method, the solid line shows the average performance across the 18 components, and the shaded region shows the range of results. For α-point rounding, 250 sets of samples of rounding points α_s were taken for each component and each value of c in $K(t,t') = 2c/(c-1)(t(t+1)^{2/(c-1)})$, and the sample yielding the best objective value was chosen. Note that α-point rounding performs best for small values of c where the kernel decays more aggressively, resulting in a solution that tends to be closer to the LP solution, even when some edges in the tree are covered by many switches and the applicable approximation guarantees do not hold. Overall, both polynomial-time methods are competitive, with the greedy method sometimes obtaining the optimal IP solution and sometimes being outperformed by α-point rounding.

6.4 Local search for multi-objective optimization

In the branch exchange Algorithm 3 for local search, for any potential exchange of $e \in T$ and $s \in S$, the energy cost is immediately known. Computing SAIDI·R-Time requires specifying a reconnection order σ. To do this efficiently, we again use a greedy approach, where at each time step i we choose the switch s which achieves the greatest marginal increase

$$\sum_{e \in U \text{ covered by } s} w(e)p(e) \cdot \sum_{e \text{ covered by } \{\sigma_1,\ldots,\sigma_{i-1}\}} p(e) + \sum_{e \in U \text{ covered by } s} p(e) \cdot \sum_{e \text{ covered by } \{\sigma_1,\ldots,\sigma_{i-1}\}} w(e)p(e)$$

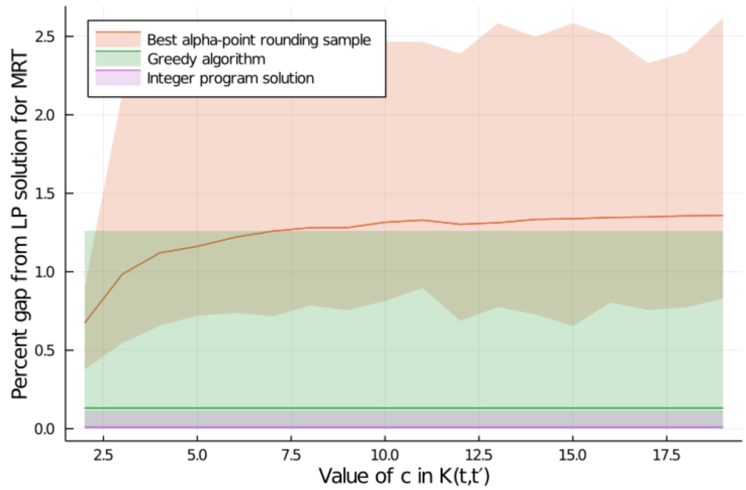

Figure 5: Performance of α-point rounding, the greedy algorithm, and an integer program for minimizing R-TIME over the 18 components of the Greensboro network, compared to the linear program objective. For each metric, the solid line denotes the average performance of the corresponding method over the 18 components, while the band ranges between the highest and lowest observed percentage differences.

in the objective SAIDI·R-TIME, where U is the set of edges not covered by $\sigma_1, \ldots, \sigma_{i-1}$.

Figure 6 shows the results of performing the branch exchange local search on Component 7 on the Greensboro network. The local search ended after 251 branch exchanges, stopping at a tree T with no feasible switches that would improve the product objective. All three objectives improved substantially during the local search, with energy decreasing by 36%, SAIDI by 70%, R-TIME by 81%, and their product by 93%.

Since the possible branch exchanges are sampled randomly, the results of the local search can vary across multiple runs. Figure 6 shows the results of 50 different applications of local search to component 7, each stopping after 100 exchanges. For each step, the mean of each metric, across all 50 runs, is plotted in the band between the 10th and 90th percentile values. Note that energy is the most volatile metric, as one branch exchange step can greatly affect the total energy loss in the network.

7 Conclusion

We analyzed a distributed approach for automatically reconfiguring distribution systems into an operational radial network after a fault occurs by creating an ordering in which switches automatically close upon detection of a downstream fault. The switches' reconnection ordering significantly impacts the expected time to reconnect under normal disruptions and thus affects reliability metrics such as SAIDI and CAIDI, which are the basis for regulator-imposed financial incentives for performance.

We modeled these metrics using a new problem called Minimum Reconnection Time (MRT), which we showed is a special case of the well-known minimum linear ordering problem. We showed that MRT is NP-hard and generalized the kernel-based rounding approaches of Bansal et al. to interpolate approximation guarantees from 9/4 to 4 dependent on the coverage of each tree edge. Finally, using local search, we optimized multiple metrics simultaneously on the NREL SMART-DS Greensboro dataset, thereby giving a proof of concept that branch exchange might be implemented

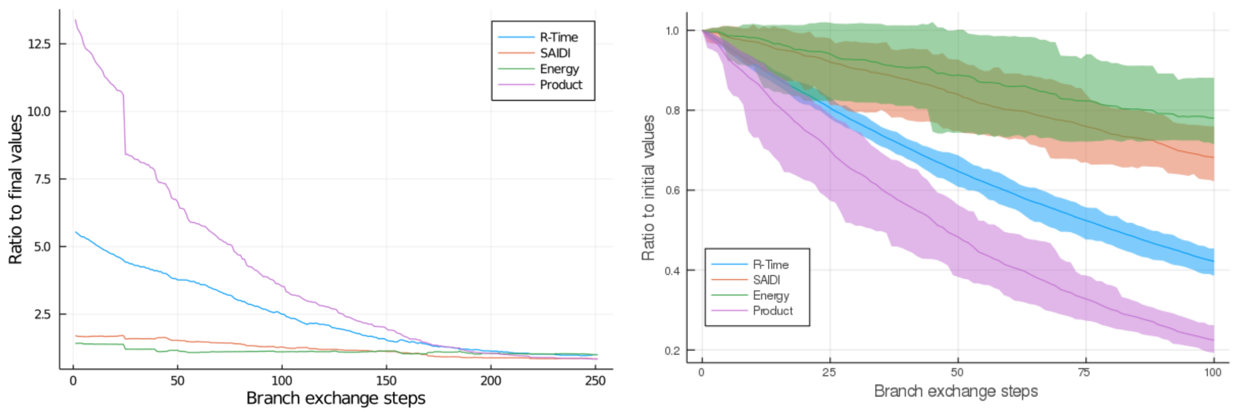

Figure 6: Left: R-Time, SAIDI, energy, and product objectives for Greensboro Component 7 during the branch exchange local search. Objectives are plotted as a ratio of their value after each step to their value when the algorithm stopped, after 251 exchanges. Right: Average values and confidence intervals (first to last deciles) for each objective, taken over 50 applications of the branch exchange local search. Objectives are plotted as a ratio of their value to the initial values.

efficiently using proxies for MRT subproblems.

Our work raises interesting theoretical and practical open questions. First, it is open if the greedy algorithm for MLOP attains a worst-case ratio better than 4-factor, as the experiments suggest. Second, local search is currently well-understood for maximization of submodular functions under matroid base constraints [10], but our problem here requires analysis of local search for minimizing the *product* of supermodular functions over the matroid base constraint. This seems to be a non-trivial extension, and we leave this as an open question. Lastly, we hope that our work provides a starting point for analyzing multiple simultaneous failures in the distribution network, a situation that might be interesting for major disruptions in practice.

References

[1] Advanced Control Systems, Inc.: PRISM Advanced Distribution Management System. `https://www.acspower.com/wp-content/uploads/dlm_uploads/2019/02/060120-PRISM-ADMS.pdf` (2020)

[2] Albert, R., Albert, I., Nakarado, G.L.: Structural vulnerability of the North American power grid. Physical Review E **69**, 025103 (2004)

[3] American Public Power Association: APPA recognizes member utilities for reliability efforts. `https://www.publicpower.org/periodical/article/appa-recognizes-member-utilities-reliability-efforts` (2020)

[4] American Public Power Association Reliability Team: APPA sample utility 2019 annual benchmarking report. `https://www.publicpower.org/system/files/documents/Sample_2019eRTAnnualBenchmarkingReport.pdf` (2020)

[5] Bansal, N., Batra, J., Farhadi, M., Tetali, P.: Improved approximations for min sum vertex cover and generalized min sum set cover. In: Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 998–1005. SIAM (2021)

[6] Blackburn, J.L., Domin, T.J.: Protective Relaying: Principles and Applications. CRC press (2015)

[7] Bolacell, G.S., Venturini, L.F., da Rosa, M.A., Issicaba, D.: Evaluating short circuit indices in an integrated assessment of distribution system adequacy and power quality. Electric Power Systems Research **189**, 106657 (2020)

[8] Borghetti, A.: A mixed-integer linear programming approach for the computation of the minimum-losses radial configuration of electrical distribution networks. IEEE Transactions on Power Systems **27**(3), 1264–1273 (2012)

[9] California Public Utilities Commission: Decision on the general rate cases of San Diego Gas & Electric Company and Southern California Gas Company and the proposed settlements. `https://docs.cpuc.ca.gov/PublishedDocs/Published/G000/M164/K606/164606603.pdf` (2016)

[10] Calinescu, G., Chekuri, C., Pal, M., Vondrák, J.: Maximizing a monotone submodular function subject to a matroid constraint. SIAM Journal on Computing **40**(6), 1740–1766 (2011)

[11] Caswell, H.C., Forte, V.J., Fraser, J.C., Pahwa, A., Short, T., Thatcher, M., Werner, V.G.: Weather normalization of reliability indices. IEEE Transactions on Power Delivery **26**(2), 1273–1279 (2011)

[12] Chen, C., Wang, J., Qiu, F., Zhao, D.: Resilient distribution system by microgrids formation after natural disasters. IEEE Transactions on Smart Grid **7**(2), 958–966 (2016)

[13] Cohen, N., Nutov, Z.: A (1+ln2)-approximation algorithm for minimum-cost 2-edge-connectivity augmentation of trees with constant radius. Theoretical Computer Science **489-490**, 67 – 74 (2013)

[14] Datta, A., Fredrikson, M., Ko, G., Mardziel, P., Sen, S.: Use privacy in data-driven systems: Theory and experiments with machine learnt programs. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 1193–1210 (2017)

[15] Department of Energy, Electricity Delivery & Energy Reliability: Fault location, isolation, and service restoration technologies reduce outage impact and duration. Tech. rep., US Department of Energy (2014). `https://www.smartgrid.gov/files/B5draftreport-12-18-2014.pd`

[16] Dorfman, Y., Orlova, G.: Finding the maximal cut in a graph. Engineering Cybernetics **10**, 502–506 (1972)

[17] Elattar, E.E., Shaheen, A.M., El-Sayed, A.M., El-Sehiemy, R.A., Ginidi, A.R.: Optimal operation of automated distribution networks based-MRFO algorithm. IEEE Access **9**, 19586–19601 (2021)

[18] Fang, X., Misra, S., Xue, G., Yang, D.: Smart grid – The new and improved power grid: A survey. IEEE Communications Surveys & Tutorials **14**(4), 944–980 (2012)

[19] Feige, U., Lovász, L., Tetali, P.: Approximating min-sum set cover. In: K. Jansen, S. Leonardi, V. Vazirani (eds.) Approximation Algorithms for Combinatorial Optimization, pp. 94–107. Springer Berlin Heidelberg, Berlin, Heidelberg (2002)

[20] Franco, J.F., Rider, M.J., Lavorato, M., Romero, R.: A mixed-integer LP model for the reconfiguration of radial electric distribution systems considering distributed generation. Electric Power Systems Research **97**, 51–60 (2013)

[21] Gao, Y., Wang, W., Shi, J., Yu, N.: Batch-constrained reinforcement learning for dynamic distribution network reconfiguration. IEEE Transactions on Smart Grid **11**(6), 5357–5369 (2020)

[22] GE Digital Energy: GENe DMS. `https://www.gegridsolutions.com/products/brochures/uos/genedms.pdf` (2012)

[23] Glover, J.D., Sarma, M.S., Overbye, T.: Power System Analysis & Design. Cengage Learning (2017)

[24] Goemans, M.X., Williamson, D.P.: Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. Journal of the ACM (JACM) **42**(6), 1115–1145 (1995)

[25] Gupta, S., Khodabakhsh, A., Mortagy, H., Nikolova, E.: Electrical flows over spanning trees. Mathematical Programming, Series B pp. 1–41 (2021)

[26] Hesmondhalgh, S., Zarakas, W., Brown, T.: Approaches to setting electric distribution reliability standards and outcomes. Tech. rep., The Brattle Group (2012)

[27] IEEE PES Distribution Reliability Working Group: PES DRWG benchmark results. `https://cmte.ieee.org/pes-drwg/benchmarking/` (2020)

[28] IEEE Power and Energy Society, Transmission and Distribution Committee: IEEE guide for electric power distribution reliability indices (2004)

[29] Iwata, S., Tetali, P., Tripathi, P.: Approximating minimum linear ordering problems. In: Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, pp. 206–217. Springer (2012)

[30] Jaatun, M.G., Gaup Moe, M.E., Nordbø, P.E.: Cyber security considerations for self-healing smart grid networks. In: International Conference on Cyber Security and Protection of Digital Services (Cyber Security), pp. 1–7 (2018)

[31] Jayantilal, A., McCarthy, C.A.: Reducing Customer Outages–Smart Distribution through Layered Intelligence. `https://www.sandc.com/en/solutions/self-healing-grids/` (2012). S&C Electric Company

[32] Ji, X., Yin, Z., Zhang, Y., Xu, B., liu, Q.: Real-time autonomous dynamic reconfiguration based on deep learning algorithm for distribution network. Electric Power Systems Research **195**, 107132 (2021)

[33] Ji-Yuan Fan, Lan Zhang, McDonald, J.D.: Distribution network reconfiguration: Single loop optimization. IEEE Transactions on Power Systems **11**(3), 1643–1647 (1996)

[34] Joskow, P.L.: Incentive Regulation in Theory and Practice: Electricity Distribution and Transmission Networks, pp. 291–344. University of Chicago Press (2014)

[35] Karp, R.M.: Reducibility among combinatorial problems. In: Complexity of computer computations, pp. 85–103. Springer (1972)

[36] Kaufmann, L., Getachew, L., Makos, M., Rich, J.: System reliability regulation: A jurisdictional survey. `https://www.oeb.ca/oeb/_Documents/EB-2010-0249/PEG_OEB_Service_Quality_Report.pdf` (2010)

[37] Kersting, W.H.: Distribution System Modeling and Analysis, 4 edn. CRC Press (2017)

[38] Kersulis, J.A., Hiskens, I.A., Coffrin, C., Molzahn, D.K.: Topological graph metrics for detecting grid anomalies and improving algorithms. In: 20th Power Systems Computation Conference (PSCC). Dublin, Ireland (2018)

[39] Khodabakhsh, A., Yang, G., Basu, S., Nikolova, E., Caramanis, M., Lianeas, T., Pountourakis, E.: A submodular approach for electricity distribution network reconfiguration. In: 51st Hawaii International Conference on Systems Science (HICSS), pp. 2717–2726 (2017)

[40] LaCommare, K.H., Eto, J.H., Dunn, L.N., Sohn, M.D.: Improving the estimated cost of sustained power interruptions to electricity customers. Energy **153**, 1038–1047 (2018)

[41] Larsen, P.H., LaCommare, K.H., Eto, J.H., Sweeney, J.L.: Assessing changes in the reliability of the U.S. electric power system. Tech. Rep. LBNL-188741, Lawrence Berkeley National Laboratory (2015)

[42] Larsen, P.H., Lawson, M., LaCommare, K.H., Eto, J.H.: Severe weather, utility spending, and the long-term reliability of the U.S. power system. Energy **198**, 117387 (2020)

[43] Marcelino Madrigal and Robert Uluski and Kwawu Mensan Gaba: Practical guidance for defining a smart grid modernization strategy, the case of distribution, Revised edition. Tech. rep., World Bank Publications (2017)

[44] Martin, R.: A sharp polynomial size linear programming formulation of the minimum spanning tree problem. Graduate School of Business, University of Chicago, Chicago, IL (1986)

[45] Meliopoulos, A.P.S., Cokkinides, G.J., Myrda, P., Liu, Y., Fan, R., Sun, L., Huang, R., Tan, Z.: Dynamic state estimation-based protection: Status and promise. IEEE Transactions on Power Delivery **32**(1), 320–330 (2017)

[46] Merlin, A., Black, H.: Search for a minimal-loss operating spanning tree configuration in an urban power distribution system. In: 5th Power Systems Computation Conference (PSCC). Cambridge, UK (1975)

[47] Midcontinent Independent System Operator: Schedule 28: Demand curves for operating reserve, regulating and spinning reserve, and regulating reserve, up ramp capability, down ramp capability, and market-wide short-term reserve. https://cdn.misoenergy.org/Schedule

[48] Mishra, S., Das, D., Paul, S.: A comprehensive review on power distribution network reconfiguration. Energy Systems **8**(2), 227–284 (2017)

[49] Moxley, R., Fodero, K.: High-speed distribution protection made easy: Communications-assisted protection schemes for distribution applications. In: 18th International Conference and Exhibition on Electricity Distribution (CIRED), pp. 1–16 (2005)

[50] PA Consulting: ReliabilityOne Awards: Recognizing outstanding reliability in North American utilities. `https://www.paconsulting.com/events/2020/reliabilityone-2020/` (2020)

[51] Pagani, G.A., Aiello, M.: The power grid as a complex network: A survey. Physica A: Statistical Mechanics and its Applications **392**(11), 2688–2700 (2013)

[52] Palmintier, B., Elgindy, T., Mateo, C., Postigo, F., Gómez, T., de Cuadra, F., Martinez, P.D.: Experiences developing large-scale synthetic U.S.-style distribution test systems. Electric Power Systems Research **190**, 106665 (2021)

[53] Pinelis, I.: Answer to "A geometric mean form of the Hermite-Hadamard inequality, for negative powers". mathoverflow.net/questions/388875 (2021)

[54] Poudel, S., Sharma, P., Dubey, A., Schneider, K.P.: Advanced FLISR with intentional islanding operations in an ADMS environment using GridAPPS-D. IEEE Access **8**, 113766–113778 (2020)

[55] Sakis Meliopoulos, A.P., Cokkinides, G., Huang, R., Farantatos, E., Choi, S., Lee, Y., Yu, X.: Smart grid technologies for autonomous operation and control. IEEE Transactions on Smart Grid **2**(1), 1–10 (2011)

[56] Schröder, T., Kuckshinrichs, W.: Value of lost load: An efficient economic indicator for power supply security? A literature review. Frontiers in Energy Research **3**, 55 (2015)

[57] Schweitzer Engineering Laboratories: SEL Distribution Network Automation. `https://cms-cdn.selinc.com/assets/Literature/Product%20Literature/Flyers/DNA_PF00273.pdf?v=20170530-182047` (2017)

[58] Shi, Q., Li, F., Olama, M., Dong, J., Xue, Y., Starke, M., Winstead, C., Kuruganti, T.: Network reconfiguration and distributed energy resource scheduling for improved distribution system resilience. International Journal of Electrical Power & Energy Systems **124**, 106355 (2021)

[59] Uluski, R.: Optimal network reconfiguration for electric distribution systems. Tech. Rep. 1022005, EPRI (2012). `https://www.epri.com/research/products/1022005`

[60] U.S. Energy Information Administration: 2018 state electricity profiles (2019). URL `https://www.eia.gov/electricity/state/`

[61] U.S. Energy Information Adminstration: Annual electric power industry report, Form EIA-861. `https://www.eia.gov/electricity/data/eia861/` (2020)

[62] van der Walt, H.L., Bansal, R.C., Naidoo, R.: PV based distributed generation power system protection: A review. Renewable Energy Focus **24**, 33–40 (2018)

[63] Wang, J., Wang, W., Yuan, Z., Wang, H., Wu, J.: A chaos disturbed beetle antennae search algorithm for a multiobjective distribution network reconfiguration considering the variation of load and DG. IEEE Access **8**, 97392–97407 (2020)

[64] Wang, Y., Xu, Y., Li, J., He, J., Wang, X.: On the radiality constraints for distribution system restoration and reconfiguration problems. IEEE Transactions on Power Systems **35**(4), 3294–3296 (2020)

[65] Ward, D.M.: The effect of weather on grid systems and the reliability of electricity supply. Climatic Change **121**, 103–113 (2013)

[66] Xu, J., Zhang, T., Du, Y., Zhang, W., Yang, T., Qiu, J.: Islanding and dynamic reconfiguration for resilience enhancement of active distribution systems. In: 21st Power Systems Computation Conference (PSCC) (2020)

[67] Yin, Z., Ji, X., Zhang, Y., Liu, Q., Bai, X.: Data-driven approach for real-time distribution network reconfiguration. IET Generation, Transmission, & Distribution **14**(13), 2450–2463 (2020)

[68] Zhan, J., Liu, W., Chung, C.Y., Yang, J.: Switch opening and exchange method for stochastic distribution network reconfiguration. IEEE Transactions on Smart Grid **11**(4), 2995–3007 (2020)

A Proof of Lemma 3

Lemma 3 For $t \in \mathbb{N}$ and $0 < p \leq 1$, $\sum_{i=1}^{t} i^p \leq \frac{p}{p+1} \frac{t^p (t+1)^p}{(t+1)^p - t^p}$.

Proof. We proceed by induction on t. The claim holds for $t = 1$ since $2^p - 1 \leq p$ and therefore

$$\sum_{i=1}^{t} i^p = 1 \leq \frac{p \cdot 2^p}{(p+1)(2^p - 1)} = \frac{p}{p+1} \frac{t^p (t+1)^p}{(t+1)^p - t^p}.$$

Now suppose $\sum_{i=1}^{t} i^p \leq \frac{p}{p+1} \frac{t^p (t+1)^p}{(t+1)^p - t^p}$ holds for some $t \geq 1$. Then

$$\begin{aligned}
\sum_{i=1}^{t+1} i^p &\leq \frac{p}{p+1} \frac{t^p (t+1)^p}{(t+1)^p - t^p} + (t+1)^p \\
&= \frac{p}{p+1} \frac{t^p (t+1)^p + \frac{p+1}{p}(t+1)^{2p} - \frac{p+1}{p}(t+1)^p t^p}{(t+1)^p - t^p} \\
&= \frac{p}{p+1} (t+1)^p \frac{\frac{p+1}{p}(t+1)^p - \frac{1}{p} t^p}{(t+1)^p - t^p} \\
&= \frac{p}{p+1} (t+1)^p \left(\frac{(t+1)^p}{(t+1)^p - t^p} + \frac{1}{p} \right). \qquad (3)
\end{aligned}$$

The derivative of $f(t) = \frac{(t+1)^p}{(t+1)^p - t^p}$ is

$$f'(t) = p \frac{(t+1)^p t^{p-1} - (t+1)^{p-1} t^p}{((t+1)^p - t^p)^2} = p \frac{(t+1)^p t^{p-1} - (t+1)^{p-1} t^p}{\left(p \int_t^{t+1} x^{p-1} dx \right)^2}.$$

To bound the denominator of the derivative, we use the following lemma, due to Pinelis [53].[7]

Lemma 6. *Let $q \in (-1, 0]$, $v \geq 1$, and $u \in (0, v)$. Then*

$$\int_u^v x^q dx \leq (v-u) u^{q/2} v^{q/2}.$$

[7] This lemma is similar to the bound of $(v-u)(u^q + v^q)/2$ given by the Hermite-Hadamard inequality on convex functions, but with the arithmetic mean replaced by the geometric mean.

Proof. Note that the desired inequality is homogenous of degree $q+1$ in u and v. Therefore, by rescaling u and v simultaneously, we may assume without loss of generality that $u=1$. Then the desired inequality is equivalent to

$$(q+1)\left(\int_u^v x^q dx - (v-u)u^{q/2}v^{q/2}\right) = v^{q+1} - 1 - (q+1)(v-1)v^{q/2} \leq 0.$$

We next show that for all q, the derivative with respect to v of the expression $g(v,q) = v^{q+1} - 1 - (q+1)(v-1)v^{q/2}$,

$$\frac{d}{dv}g(v,q) = (q+1)v^q - (q+1)\left(v^{q/2} + \frac{q}{2}(v-1)v^{q/2-1}\right) = \frac{q+1}{2}v^{q/2-1}(2v^{q/2+1} - 2v - q(v-1)),$$

is negative.

As a function of q for any fixed v, $h_v(q) = 2v^{q/2+1} - 2v - q(v-1)$ is convex. Moreover, $h_v(-1) = -(\sqrt{v}-1)^2 < 0$ and $h_v(0) = 0$. Therefore h_v is negative for all $q \in (-1, 0]$, and since $\frac{q+1}{2}v^{q/2-1} \geq 0$, the derivative $\frac{d}{dv}g(v,q)$ is negative as desired. For all q and $v=1$, $g(v,q)=0$, so since $g(v,q)$ is decreasing in v, $g(v,q) \leq 0$ for all q and v. This proves the desired inequality. $\square$

For all $u > 0$, applying Lemma 6 with $q = p-1$ and $v = u+1$ yields

$$f'(u) = p\frac{(u+1)^p u^{p-1} - (u+1)^{p-1}u^p}{\left(p\int_u^{u+1} x^{p-1}dx\right)^2} \geq p\frac{(u+1)^p u^{p-1} - (u+1)^{p-1}u^p}{p^2(u+1)^{p-1}u^{p-1}} = \frac{(u+1)-u}{p} = \frac{1}{p}.$$

Therefore $f(t+1) \geq f(t) + 1/p$, and from Equation 3 we have that

$$\begin{aligned}
\sum_{i=1}^{t+1} i^p &\leq \frac{p}{p+1}(t+1)^p\left(\frac{(t+1)^p}{(t+1)^p - t^p} + \frac{1}{p}\right) \\
&= \frac{p}{p+1}(t+1)^p(f(t) + 1/p) \\
&\leq \frac{p}{p+1}(t+1)^p f(t+1) \\
&= \frac{p}{p+1}\frac{(t+1)^p(t+2)^p}{(t+2)^p - (t+1)^p}
\end{aligned}$$

which completes the induction. $\square$

B Integer Programming Formulations

As above, let $b(e)$ be the weights used in the objective for MRT. Then the problem of finding an optimum spanning tree $T \subseteq E$ and switch reconnection order σ on $S = E \setminus T$ can be formulated as an integer program:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{e,t} t \cdot b(e) x_{et} \\
\text{subject to} \quad & \sum_{e \subset C} s_e \le |C| - 1, \quad \forall \text{ cuts } C, \\
& \sum_e s_e = n - 1, \\
& \sum_t x_{et} = s_e, \quad \forall e, \\
& x_{et} \le \sum_f c_{eft} \quad \forall e, t, \\
& c_{eft} \le y_{ft}, \quad \forall e, f, t, \\
& y_{et} \le 1 - s_e, \quad \forall e, t, \\
& \sum_e y_{et} = 1, \quad \forall t, \\
& c_{eft} \le \sum_{g \text{ across } C} s_g - s_e, \quad \forall e, f, t, \text{cuts } C \text{ crossed by } e \text{ but not } f, \\
& w(e) \ge |C| \left(1 - \sum_{f \text{ crossing } C} s_f - s_e \right), \quad \forall e, \text{cuts } C \text{ not containing the source vertex} \\
& \text{all variables binary except } w(e) \ge 0
\end{aligned}
$$

This exponential-size formulation is intractable for large graphs, so we adapt the $O(mn)$-size formulation due to Martin [44]. This yields an $O(mn^2(m-n))$-size formulation. Note that if the graph is planar or otherwise low genus, even smaller formulations for the spanning tree polytope and hence for this IP exist.

$$
\begin{aligned}
\text{minimize} \quad & \sum_{e,t} t \cdot b(e) x_{et} \\
\text{subject to} \quad & \sum_e s_e = n - 1, \\
& s_e = a_{k,i,j} + a_{k,j,i}, \quad \forall e \in E, 1 \le k \le n, \\
& \sum_j a_{k,i,j} \le 1, \quad \forall 1 \le i, k \le n, \\
& a_{k,k,j} = 0, \quad \forall 1 \le j, k \le n, \\
& \sum_t x_{et} = s_e, \quad \forall e, \\
& x_{et} \le \sum_f c_{eft} \quad \forall e, t, \\
& c_{eft} \le y_{ft}, \quad \forall e, f, t, \\
& y_{et} \le 1 - s_e, \quad \forall e, t, \\
& \sum_e y_{et} = 1, \quad \forall t,
\end{aligned}
$$

$$
\begin{aligned}
& c_{eft} \leq r_{ef}, \quad \forall e, f, t, \\
& r_{ef} \leq 1 - s_e, \quad \forall e, f \\
& \sum_g s_{e,f,g} = n - 1, \quad \forall e, f \\
& s_{e,f,e} = 0, \quad \forall e, f \\
& s_{e,f,f} = 0, \quad \forall e, f \\
& s_{e,f,g} = b_{e,f,k,i,j} + b_{e,f,k,j,i}, \quad \forall e, f \in E, g \in E \setminus \{e\}, 1 \leq k \leq n, \\
& \sum_j b_{e,f,k,i,j} \leq 1 + n(1 - r_{ef}), \quad \forall e, f, \in E, 1 \leq i, k \leq n, \\
& b_{e,f,k,k,j} = 0, \quad \forall e, f \in E, 1 \leq j, k \leq n, \\
& \text{all variables binary}
\end{aligned}
$$

Here r_{ef} denotes that if edge e is deleted and switch f is closed, the graph will remain connected and $b_{e,f,k,i,j}$ denotes the orientation of the edge between i and j in the graph rooted at vertex k where edge e is deleted and switch f is closed. Here is one IP which encodes the MRT problem with constant failure probabilities and a fixed network. Here $f \in S$ and each switch f covers an associated set of tree edges C_f. Indices $t \in [|S|]$ denote waiting times (with $\varepsilon = 1$) or reconnection order.