FAIR AND RELIABLE RECONNECTIONS FOR TEMPORARY DISRUPTIONS IN ELECTRIC DISTRIBUTION NETWORKS USING SUBMODULARITY

CYRUS HETTLE*, SWATI GUPTA*, AND DANIEL K. MOLZAHN*

Abstract. The economic cost of power outages has been estimated to be between $35 billion and $50 billion annually in the United States. We analyze and advance methodologies for electrical distribution network reconfiguration in order to quickly restore power, an important lever for improving reliability. For short-term network disruptions, we give a novel model for reconfiguration by considering switches that can be sequentially closed upon detection of a network anomaly upstream. The order in which switches close has an impact on reliability metrics such as SAIDI, which are the basis for regulator-imposed financial performance incentives. We introduce the "Minimum Reconnection Time (MRT)" problem, which aims to find an optimal switch ordering that minimizes the cost of outages and show that it generalizes metrics like SAIDI. We first show that MRT is a special case of the well-known minimum linear ordering problem from submodular optimization literature and that MRT is also NP-hard. We show how to approximate MRT using recent kernel-based randomized rounding approaches, and in turn improve the state-of-the-art for a broad class of MLOP instances. Finally, we note that the choice of reliability metric can result in significant differences for low and medium voltage buses. We therefore consider optimizing multiple metrics simultaneously using local search methods that reconfigure the system's base tree to optimize service disruptions, reconnection times, and energy losses. We computationally validate our reconfiguration methods on the NREL SMART-DS Greensboro synthetic urban-suburban network and show significant improvement in reliability metrics.

1. Introduction. 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 [10, 80, 45, 46]. Recent research in monetizing the economic costs of power outages provides estimates between $35 billion and $50 billion annually in the United States [44]. 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 [67], and 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 [55]. These values are orders of magnitude larger than the typical costs for electricity (around 10.5 cents per kilowatt-hour in the United States [74]), reflecting the importance of providing reliable supplies of electricity.

Since reliability is so valuable, electric industry groups recognize utilities based on their performance with respect to reliability metrics such as the System Average Interruption Duration Index (SAIDI), e.g., the ReliabilityOne Award [59] 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 [8]. 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, with similar incentives in Great Britain, Italy, etc. [31, 39, 37].

Reducing outage time is an important way to improve a network's reliability metrics. After an isolated fault occurs in a system, causing an outage downstream

*Georgia Institute of Technology, {chettle, swatig, molzahn}@gatech.edu

in the network, it is crucial to quickly reconfigure the network to reroute power to the affected areas. Reconfiguration can potentially be executed long before the fault is manually repaired, reducing the outage duration for many customers. 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, such as the location of a fault that is causing a power outage, is communicated to a central hub (e.g., substation or control center) which computes an appropriate switching configuration that is then communicated to each switch. In contrast, *decentralized* approaches have local controllers that operate solely based on local information. Solving a reconfiguration problem centrally with full information and control is simpler, but has speed, robustness, and security disadvantages [78, 43]. For these reasons, protection and reconfiguration systems are moving towards an adaptive 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 [57, 11, 56, 53, 75, 66, 17].

To help realize this future paradigm, in this paper we propose a decentralized approach for automatically reconfiguring radial distribution networks to quickly restore power after a fault and consequent power interruption. We next discuss a summary of our contributions in (1) proposing a new model for faults and the protection system, (2) showing the hardness of finding the best reconnection order of switches, (3) developing new approximation guarantees for reliability metrics, (4) balancing reliability metrics and energy losses associated with the initial radial tree using local-search for fair service to different populations and (5) evaluating our approach on the NREL SMART-DS Greensboro dataset.

1. Modeling faults and the protection system. In Section 3, we describe our graph model and novel protection system. We assume a network is given as a graph $G = (V, E)$, with an initial radial tree $T \subseteq E$. We employ lines that are not in the base radial network, $S = E \setminus T$, each with a normally open switch that can detect an abnormal voltage difference on its endpoints signifying a downstream outage. We say that a switch $s \in S$ *covers* $e \in T$ if $T - e + s$ is connected. To ensure that the reconnected network remains radial (i.e., a tree) in order to satisfy short-circuit current requirements and facilitate coordination of the protection system [47, 79], we require that exactly one switch closes after a fault. Since our methods are decentralized and intended for settings that lack communication between switches, we assign an ordering of the switches in which each waits a distinct amount of time after detecting a voltage difference on its terminal buses before closing. Changes in voltage after one switch is closed will be detected by other switches with longer waiting times, which will then stay open. This reconnection order is periodically communicated to the switches in advance of the occurrence of a fault (e.g., based on estimated probability of failure of the radial lines), and the switches operate in a decentralized fashion thereafter. Thus, this approach does not require low-latency communications.

Any ordering of switches suffices for our system model, but the order in which switches are programmed to automatically close, on detecting a voltage difference, can have a significant impact on the delay that customers face. We analyze reconfiguration methods which aim to optimize various performance metrics while avoiding the need for centralized communications and computations. We use two metrics to characterize the severity of service disruptions, since different metrics capture disproportionate impacts on different parts of the network, and therefore, different demographics who are typically situated in clusters or strongly correlated with zipcode locations [13, 32]:

(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 a given time period.

Given an initial radial network, finding an optimal ordering of switches that minimizes either R-Time or SAIDI can be written as an instance of a new problem Minimum Reconnection Time or MRT: given edge weights $b(e)$ for each $e \in T$, find an ordering of the switches $S = E \setminus T$ such that the total b-weighted coverage time of $e \in T$ is minimized:

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

Setting $b(e) = p(e)$ or $b(e) = p(e)f(e)$, where $p(e)$ is the probability an edge fails in the network and $f(e)$ is the demand downstream of edge e, we can write R-Time and SAIDI as instances of MRT.

2. NP-hardness of MRT: Having formulated network reliability as MRT, we first show in Section 4 that this minimum reconnection time problem is in fact NP-hard using a reduction from the *tree augmentation problem* (TAP). In TAP, we are given a graph $G = (V, E)$ with spanning tree $T \subseteq E$, and must select a minimum-cardinality subset F of E such that $T \cup F$ is 2-edge-connected [12].

Theorem 4.4. *The tree augmentation problem can be reduced in polynomial time to the* Min Reconnection Time *problem, and the latter problem is therefore NP-hard.*

Given an instance $G = (V, E)$ and T of the tree augmentation problem, we modify the instance by adding edge-disjoint triangles $T_i = \{\{v, v_{i,1}\}, \{v_{i,1}, v_{i,2}\}, \{v_{i,2}, v\}\}$ (for $i \in [|V| \cdot |E|]$) to G which each share one specific vertex v in T. We define the MRT base tree to be the tree T' obtained by adding two edges from each newly added triangle, i.e., $T' = T \cup \bigcup_i (T_i \setminus \{v_{i,2}, v\})$. We solve MRT on this modified instance, giving each edge in the original tree weight $b(e) = 1$ and each newly-added edge weight $b(e) = 1/2$. The solution σ will cover all the newly-added edges after covering all the edges in the original tree. We show that because there are a large number of new edges, σ will cover the original tree using as few edges as possible, and those edges form a solution to the original TAP instance.

3. Connections to Submodular Optimization. Next, we show in Section 5 that the MRT problem has an interesting connection to the field of submodular optimization. A function $f : 2^V \longrightarrow \mathbb{R}$ on the power set of a ground set V is said to be *submodular* if

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B), \quad \forall A, B \subseteq V. \tag{1.1}$$

Submodular functions are a discrete analog of convexity, and capture the property of diminishing returns ($f(S \cup \{e\}) - f(S) \geq f(T \cup \{e\}) - f(T)$ for $S \subseteq T, e \notin T$). Their negatives are referred to as supermodular functions (with the inequality in (1.1) reversed). In optimization and computer science, both these functions are often found in applications related to coverage and scheduling [28, 19, 21], network topology [58], selecting a small set of scenarios to analyze the behavior of renewable energy sources in a stochastic model [77], selecting generators to minimize low-frequency phase oscillations [50], and partitioning a system into isolated islands to avoid cascading failures [49].

We show that MRT is in fact a special case of the *minimum sum set cover* (MSSC) problem, which has been studied as a linear ordering problem on supermodular functions [18, 33]. Let $H = (V, E)$ be a hypergraph, where each hyperedge $e \in E$ is a subset of the vertex set V. Then, MSSC seeks to find a linear ordering σ of vertices V so that the total waiting time for each hyperedge to be covered, i.e. $\sum_e \min_{v \in e} \sigma(v)$, is minimized. It is known that MSSC is NP-hard, and there is a tight polynomial-time 4-approximation algorithm achieved by a simple greedy algorithm [18]. We note that NP-hardness of MSSC does not imply the NP-hardness of MRT, since MRT is a special case of MSSC (Lemma 5.1).

4. Improved guarantees for Min Sum Set Cover and MRT. Since MRT is a special case of MSSC, a natural question is to ask if the greedy algorithm's approximation factor of 4 is also tight for MRT. To explore this further, in Section 5 we view MRT as an MSSC instance by considering a hypergraph over the set of switches, so that $H = (S, E_H)$, where each hyperedge $e_H \in E_H$ corresponds to the set of switches that covers edge $e \in T$. Suppose the maximum coverage of any edge e in the radial network T is c (i.e., at most c switches cover any tree edge), we give an improved approximation factor of $(2c/(c+1))^2$ for MRT instances. Note that this ratio asymptotically approaches 4 as $c \to \infty$, but for simpler networks as seen in practice it can be much closer to 2.

THEOREM 5.4. *Let $G = (V, E)$ be the given network, let $T \subseteq E$ be a spanning tree on G with edge weights $b(e) \in \mathbb{R}_+$ for all $e \in T$. Let the set of switches be $S = E \setminus T$, and let $c = \max_{e \in T} |\{s \in S : T - e + s \text{ is connected}\}|$ be the maximum coverage of any edge in the tree. Then, if $c \geq 2$, there is a polynomial-time $(\frac{2c}{c+1})^2$-approximation algorithm for the MRT problem on G.*

To achieve this improvement in the approximation factor, we use *fractional kernels* within a kernel-based randomized rounding approach [62, 27], which may be of independent interest. In particular, we consider c-uniform instances of MSSC and show the improvement in the approximation for these. A kernel-based randomized rounding approach involves solving a linear programming relaxation to find x^*, "spreading" this fractional solution using a kernel K, and then subsequently rounding Kx^* appropriately. In a recent work, Bansal et al. [4] obtained a 4-approximation for MSSC using $K = 2/t \cdot \mathbb{1}[t \geq t']$, and a 16/9-approximation for the minimum sum vertex cover problem (MSVC) using $K = 4\frac{t'(t'+1)}{t(t+1)(t+2)} \cdot \mathbb{1}[t \geq t']$. They showed that the latter bound is tight for the special case of solving MSSC on 2-uniform hypergraphs, that is, each hyperedge has exactly two vertices. In contrast, to attain tight guarantees for MSSC on c-uniform hypergraphs, we considered a new kernel of the form $K(t, t') = \frac{2c}{c+1} \frac{t'^{2/(c-1)}}{\sum_{i=1}^{t} i^{2/(c-1)}} \cdot \mathbb{1}[t \geq t']$, and showed the following approximation bound for its performance. Note that this ratio is better than 4 for all instances of MSSC.

THEOREM 5.2. *Let $H = (V, E)$ be a c-uniform hypergraph, where $c \geq 3$. Then applying α-point rounding with kernel*

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

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

Our proposed kernels depend on the value of c, and like the kernels used in [4], are based on discrete analogs of negative power functions. These kernels spread the weight of relaxed LP solutions more aggressively, which is necessary to ensure that small fractional contributions in weight (which are more frequent for small c) contribute to a good tentative schedule, but that the kernel does not create so much more total weight that the gap between the tentative schedule and final order becomes large. To be able to spread the weights in the solution x^* at the correct rates, our kernels must include fractional powers of the times t, whereas the kernels of [4] use only integral powers. This complicates the analysis of the kernel-based weight-spreading process. In particular, the analysis of the gap between the LP objective and the cost of the tentative schedule uses discrete calculus-like manipulations of the kernels, and requires close bounds on their sums. These are significantly more intricate when dealing with the fractional powers of t used in our kernels. We further show that the $(2c/(c+1))^2$ approximation ratio is tight (Theorem 5.5) for c-uniform MSSC instances, with respect to the natural LP relaxation.

THEOREM 5.5. *Let $c \geq 2$. Then the integrality gap of the LP for MSSC on c-uniform hypergraphs is at least $(\frac{2c}{c+1})^2$.*

To show this gap, we construct a c-uniform instance consisting of the disjoint union of a large number of complete c-uniform hypergraphs (i.e., each subset of c vertices comprises a hyperedge) of varying sizes. The LP relaxation can efficiently cover each hypergraph by assigning each vertex weight $1/c$, thus covering each complete c-uniform hypergraph on n vertices in time n/c. In contrast, any integral solution must select all but $(c-1)$ vertices in each of the complete hypergraphs, which takes $n-c+1$ time steps. As the size of the instance grows, the gap between the costs of these solutions approaches $(\frac{2c}{c+1})^2$.

5. Balancing multiple netrics of reliability and energy losses. Given $G = (V, E)$, the choice of starting spanning tree T can have a large impact on the reconnection metrics. For example, Figure 1 shows two different spanning trees on the 7-vertex wheel graph. The set of switches for each spanning tree has a unique optimal reconnection order (up to graph isomorphism) that minimizes both SAIDI and R-TIME. The optimal order 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 order 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 (for electricity distribution networks, these losses can be as large 10% of the total energy loss) [42, 26]. In Figure 1, energy losses are 6 in the "spoke" (left) and 91 in the "wheel" (right), where we assume each bus has uniform demand of one power unit all lines have uniform resistances, and losses are proportional to the square of the power flow through each line.

To optimize the radial tree, in Section 6 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 losses. The local search uses a *branch exchange* method, which proposes a new spanning tree at each step by replacing a tree edge with a non-tree edge. It then evaluates the new spanning tree on the SAIDI, R-TIME, and energy loss metrics, and accepts it if it has better objective value than the current tree. To efficiently estimate SAIDI and R-TIME, we take the best of the greedy MRT solution and approximations using the randomized rounding algorithm c-uniform MSSC (for instance-specific c).

Obtaining good approximate solutions to SAIDI and R-TIME critically relies on the *coverage* information of the network, i.e., which switches cover which edges.

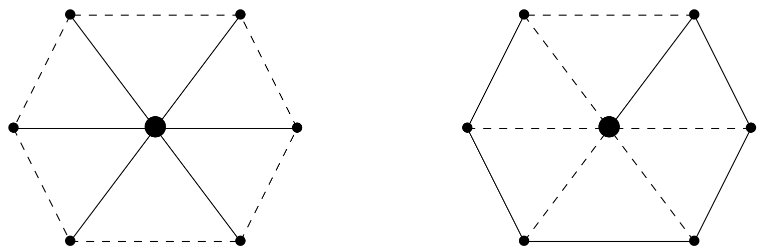

FIG. 1. *Two spanning trees on the same network. Solid edges are in T, dashed edges are in $E \setminus T$, the source r is the largest vertex, and all vertex weights and edge failure probabilities are equal.*

However, updates to the network structure due to local search significantly slow down the overall process. Indeed, a naive approach to update network connectivity would take $O(|V|^2|E-T|)$ time at each step. For tractability of local search, we show that only the coverage of certain edges needs to be updated in each step and we give an efficient way to detect these. This reduces the update time by a factor of $\Omega(\min\{|E-T|, |V|/d\})$, where d is the length of the longest cycle of G, significantly speeding up the local search.

LEMMA 6.1. *Let $G = (V, E)$ be a graph and let $T \subseteq E$ be a spanning tree of G. Let $e \in T$, $s_e \in E - T$ be an edge and a switch such that $T' = T - e + s_e$. Then, all covering data for T also hold for the new spanning tree T', except the following pairs (f, s):*
(i) f is in the unique cycle C_e in $T + s_e$ and the unique cycle C_s in $T + s$ contains e.
(ii) 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 demonstrate this, we show that if we make two simultaneous branch exchange steps, the tree remains connected as long as one of the removed edges does not intersect with the cycle created by adding the switch in the other branch exchange. We prove that this condition is satisfied as long as neither (i) nor (ii) holds and use it to show that $T - f + s$ is connected if and only if $T' - f + s$ is connected.

6. Experimental validation. In Section 7, we test our methods on the NREL SMART-DS Greensboro synthetic urban-suburban network. This network contains 18 distinct radial networks, each with its own substation, and a total of 145052 buses. We show that by contracting some edges with low flow, we can greatly reduce the computational complexity of the problem with minimal impact on solution quality. The synthetic network has an insufficient number of switches to be able to achieve good coverage in most areas, and we therefore give a modified greedy method for determining good locations for augmenting the system with a modest number of additional switches. This method selects switches to construct one-by-one based on the weight $b(e) = p(e)f(e)$ of the edges they would cover. However, instead of setting the weight of a covered edge to zero, we halve it each time it is covered. Thus, the resulting set of additional switches balances small cardinality with increased redundancy and flexibility, making it suitable for local search computations.

Adding enough switches to protect against failures almost everywhere in the network is very expensive, but we show that excellent coverage, prioritizing the heaviest-load edges, can be achieved by adding relatively few switches. Indeed, on average over 90% of the weight $b(e) = p(e)f(e)$ in the SAIDI metric can be covered by adding a number of switches equal to 0.19% of the number in the spanning tree. We demonstrate that after adding switches, the local search branch exchange method can significantly improve outage and energy metrics, with up to an 95% reduction in the product of R-Time, SAIDI, and energy loss and nearly a 60% improvement on average.

2. Related work. 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. Distribution network reconfiguration has been a widely studied problem in power systems engineering since the 1970s [54] and remains an active research topic [56, 81, 83, 76, 82, 22, 69, 35, 16]. Many types of algorithms have been applied to this problem, including, e.g., traditional optimization methods such as linear programming [69] and mixed-integer programming [7, 20], heuristic decision strategies [36, 83], machine learning methods [82, 22, 83, 35], approximation algorithms [26], and metaheuristic techniques [82, 76, 16]. See [56] for a 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) [14, 73, 65]. There are a variety of self-healing approaches using decentralized schemes where all agents are on the same level of the network. Li et al. isolate faults onto "islands" and use alternate paths to restore service. The method proposed by Torres et al. employs an internal timer for each switch [72]. Industrial implementations of network reconfiguration include, for instance, ADMS products provided by Schweitzer Engineering Laboratories [68], Advanced Control Systems, Inc. [1], General Electric [23], S&C Electric [34], etc. Case studies and discussions on a range of practical implementation issues are provided in [14, 73, 51].

The Min Sum Set Cover problem was introduced by Feige et al. in 2002 [18] and has seen application in such diverse fields as search procedures [30], password security [84], and network stream query processing [71]. An α-point rounding method was used by Bansal et al. in their approximation algorithms for MSSC and min sum vertex cover, as well as by Happach and Schulz in algorithms for generalizations of MSSC [4, 29]. The technique is often used in approximation algorithms to convert a preemptive schedule into a non-preemptive one by scheduling each task for the time at which a random α-fraction of the job has been completed. Frequently, a preemptive schedule is generated by solving a relaxation of a linear programming or other formulation, and α-point rounding is used to convert the preemptive schedule into a feasible solution. It was introduced by Hall et al. and Phillips et al. [27, 63] in 1997, and has been used extensively in later scheduling work; see [70] for a survey.

3. Problem formulation and modeling. In this section, we describe our graph-based model for distribution networks and a decentralized method for restoring power. This method involves selecting an order of the waiting times for each switch, and we give a problem formulation for choosing an order to minimize outage metrics including SAIDI.

3.1. 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. One vertex $r \in V$ denotes the substation, and is the root of a spanning tree T on G representing the initial radial configuration. Let $S = E \setminus T$ denote the set of edges (switches) not in the spanning tree.

We note that this graph-based modeling approach is a simplified representation of the distribution network compared to the model used in, for instance, power flow studies that additionally involve information about the line reactances, reactive demands, capacitor susceptances, etc. [40]. 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, 60, 41], including in the context of network reconfiguration [42, 26, 48].

3.2. 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, if necessary we can preprocess the graph by contracting each set of connected edges in T that has only one breaker to a single edge.[1] Each substation supplies power to its connected component, which is in turn operated radially [47, 79, 40]. At any time t, the set of edges across which current is allowed to flow will be a forest, and the vertices receiving power are those in the same component as the power source r. Initially, all edges are operational and so all vertices are in the same component as r. If failures occur, the set of edges across which current is allowed to flow could become a disconnected union of multiple trees.

When a fault occurs in edge e, 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 s 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. This is feasible if and only if e is in the unique cycle in $T + e$; in other words, $T - e + s$ is a tree. In this case, we say that s *covers* e.

To maximize service availability, we would like to close such a switch to temporarily restore power to as many vertices as possible as quickly as possible. However, we must ensure that no current will flow to the component of $T - e$ containing the faulted edge, thus continuing to isolate the fault.

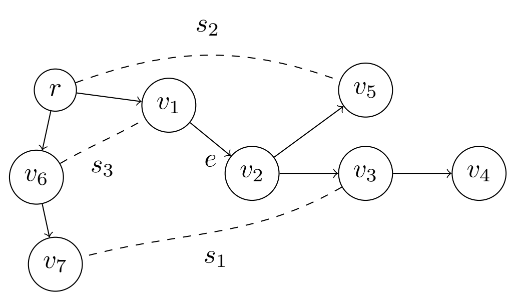

FIG. 2. *An example of a network comprising a spanning tree (solid edges) and switches (dashed edges.)*

[1]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.

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 as the source vertex. If the switch on s_1 or the switch on s_2 is closed, then power would be restored. 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 fault.

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 single edge in S that connects a vertex with power to a vertex that is without power and not in the component of the fault and close the switch on that edge.

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 the current status of the network.

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.

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 a 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 containing the zero voltage vertex and will not attempt to energize the faulted component.

If the priority times are sufficiently separated, then 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 $t(e_2)$. This ensures that multiple switches cannot close in such quick succession that a cycle forms.

For simplicity, we let $\varepsilon > 0$ be a network-dependent constant 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. Then it suffices to choose a permutation $\sigma : S \to |S|$ and set $t(s) = \sigma(s)\varepsilon$.

Any valid function ordering σ will yield an assignment of values for $t(s)$ and eventually safely restore power to the maximum possible number of vertices, but we would like to do so as quickly as possible. There are several different metrics for measuring the effectiveness of a reconnection method, which we discuss next.

3.3. Definition of metrics. We consider three metrics on networks and reconnection methods, which depend on data about the network. Each vertex $v \in V$ has a power demand $w(v)$ and each edge $e \in E$ has an electrical resistance $r(e)$, which depends on the physical properties of the corresponding distribution line. For all $e \in E$, 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 let $f(e)$ be the total weight of the vertices in $D(e)$. Let $t(e) = \min_{s \in S: s \text{ covers } e} t(s)$ be the time needed 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.

The System Average Interruption Duration Index (SAIDI) is a standard metric that measures average outage time, weighted by the demand of each vertex $w(v)$ (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 $r(e)f(e)^2$. (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:

$$\text{(MRT):} \quad \min_{\sigma: S \to [|S|]} \sum_{e \in T} b(e) \min_{s \text{ covers } e} \sigma(s). \tag{3.1}$$

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 the 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.

4. 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.

4.1. 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 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 [18]. We first define the general form of MLOP with respect to any set function f:

Definition 4.1. *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.[2]

Another instance of MLOP is the Min Sum Set Cover (MSSC) problem, which given a set of subsets $S_1, \dots, S_k \subseteq S$, 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 4.2. *The* MSSC *problem, given a hypergraph $H = (V, E)$, is to find a linear ordering σ of the 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 4.3. *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

[2]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.

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.[3]

4.2. 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) [12]. 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. Compare this to MRT, in which an ordering σ (a solution to MRT) is guaranteed to restore power by time k if the subgraph of G containing the edges of $T \cup \{s \in S : \sigma(s) \le k\}$ is a 2-edge-connected graph. Thus MRT can be viewed as a timed version of TAP, much as MSSC is the timed version of ordinary Set Cover: the latter seeks to find the minimum *number* of subsets to cover a given set of elements [38].

Theorem 4.4. *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 $G = (V, E)$ with spanning tree T be an instance of TAP, and let $n = |V|, m = |E|$.

We modify the instance to create a new graph $G' = (V', E')$ with a new spanning tree T'. Add the $2mn$ vertices $v_{i,j}$, where $i \in [mn], j \in [2]$, setting $V' = V \cup \{v_{1,1}, v_{1,2}, \dots, v_{mn,1}, v_{mn,2}\}$. Select an arbitrary vertex $v \in V$, and let $P_T = \{\{v, v_{1,1}\}, \{v_{1,1}, v_{1,2}\}, \dots, \{v, v_{mn,1}\}, \{v_{mn,1}, v_{mn,2}\}\}$ be a set of edges which constitute mn disjoint 2-paths. Let $P_S = \{\{v, v_{1,2}\}, \dots, \{v, v_{mn,2}\}\}$ be a set of mn edges, and note that for $1 \le i \le mn$, $\{v, v_{i,2}\}$ covers the edges $\{v, v_{1,1}\}, \{v_{1,1}, v_{1,2}\}$. Finally, let $E' = E \cup P_T \cup P_S$, let $T' = T \cup P_T$, and give all edges $e \in T$ weight $b(e) = 1$ and all edges $e \in P_T$ weight $b(e) = 1/2$.

Let σ be an optimum solution to MRT on $G' = (V', E')$ with tree T'. Note that the newly added switches $\{v, v_{1,2}\}, \dots, \{v, v_{mn,2}\}$ are the only switches that cover edges in P_T. Furthermore, these switches each cover tree edges of total weight 1, and since all edges in T have weight 1, without loss of generality we may assume that the switches in P_S are the last mn switches in σ that contribute a positive quantity to the objective $OBJ(\sigma)$ of σ as a solution to MRT. Let the switches chosen before those mn switches be $s_1, \dots, s_k = \sigma^{-1}(1), \dots, \sigma^{-1}(k)$, and for $1 \le i \le k$, let $S_i = \{e \in T : s_i \text{ covers } e\}$. Then the objective value of σ is

$$\begin{aligned} OBJ(\sigma) &= \sum_{i=1}^{k} i \left| S_i \setminus \bigcup_{j=1}^{i-1} S_j \right| + \sum_{\ell=k+1}^{k+mn} 1 \\ &= \sum_{i=1}^{k} i \left| S_i \setminus \bigcup_{j=1}^{i-1} S_j \right| + mn \cdot k + \frac{mn(mn+1)}{2}. \end{aligned}$$

Suppose there is a solution $F = \{f_1, \dots, f_{k^*}\} \subseteq S$ to Tree Augmentation on the original instance G, T such that $k^* < k$. We construct another solution σ^* to MRT on G' and T' as follows.

[3] For example, even though finding the maximum weighted cut in a graph is NP-hard [25], it is solvable in polynomial time when restricted to planar instances [15].

For $1 \leq i \leq k^*$, let $F_i = \{e \in T : f_i \text{ covers } e\}$. Without loss of generality, assume that $\left|F_i \setminus \bigcup_{j=1}^{i-1} F_j\right| \geq \left|F_{i+1} \setminus \bigcup_{j=1}^{i} F_j\right|$ for all $i \in [k^* - 1]$ (if not, we can permute the indices of the f_i to make this so). Let σ^* be a permutation of $E' \setminus T'$ such that

$$\sigma^{*-1}(i) = \begin{cases} f_i, & \text{if } i \leq k^* \\ \{v, v_{i-k^*,2}\}, & \text{if } k^* + 1 \leq i \leq k^* + mn. \end{cases}$$

Then since $\sum_{i=1}^{k^*} \left|F_i \setminus \bigcup_{j=1}^{i-1} F_j\right| = \left|\bigcup_{i=1}^{k^*} F_j\right| = |T| = n - 1$, we have that

$$\sum_{i=1}^{k^*} i \left|F_i \setminus \bigcup_{j=1}^{i-1} F_j\right| \leq \sum_{i=1}^{k^*} i \frac{n-1}{k^*} = \frac{k^*(k^*+1)}{2} \frac{n-1}{k^*} = \frac{(k^*+1)(n-1)}{2} \leq mn.$$

Therefore,

$$\begin{aligned} OBJ(\sigma^*) = \sum_{i=1}^{k^*} i \left|F_i \setminus \bigcup_{j=1}^{i-1} F_j\right| + \sum_{\ell=k^*+1}^{k^*+mn} 1 &\leq mn + mn \cdot k^* + \frac{mn(mn+1)}{2} \\ &\leq mn \cdot k + \frac{mn(mn+1)}{2} < OBJ(\sigma), \end{aligned}$$

which would contradict the optimality of σ. Therefore, there is no such solution $\{f_1, \ldots, f_{k^*}\}$ to Tree Augmentation on G and T using $k^* < k$ edges, and so $\{s_1, \ldots, s_k\}$ is an optimum solution to the original Tree Augmentation instance. Since G' and T' have size polynomial in the size of G and T, this gives a polynomial-time reduction of Tree Augmentation to MRT, and thus MRT is also NP-hard. ∎

5. Approximation algorithms for MRT. We next discuss methods for solving MSSC and MRT with provable approximation guarantees. First, we discuss the greedy algorithm for general MSSC and show that it does not always give an optimal solution to MRT. Second, we give new kernels and new analysis for the alpha-point rounding method for MSSC introduced by Bansal et al. [4]. We show a $(2c/(c+1))^2$ approximation ratio for MSSC on c-uniform graphs for $c \geq 2$ (corresponding to MRT instances where at most c switches cover each edge) and demonstrate that this guarantee is tight with respect to the natural LP relaxation.

5.1. The greedy algorithm. Since an instance of MRT is an instance of MSSC, one may apply the greedy algorithm for MSSC [18]. For each i from 1 to $|S|$, the greedy algorithm chooses

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

that is, σ_i is the edge in S which covers the uncovered tree edges with greatest total weight. Pseudocode appears in Algorithm 5.1. In [18], Feige et al. showed that the greedy algorithm is a 4-approximation for general MSSC, and that this bound is tight.

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

Algorithm 5.1 (Greedy) Greedy algorithm for MRT [18]

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 $\sigma_i = \operatorname{argmax}_{s \in S \setminus \{\sigma_1, \ldots, \sigma_{i-1}\}} \sum_{e \in U : s \text{ covers } e} b(e)$.
    Set $U = U \setminus \{e \in U : \sigma_i \text{ covers } e\}$
return $\sigma_1, \ldots, \sigma_m$

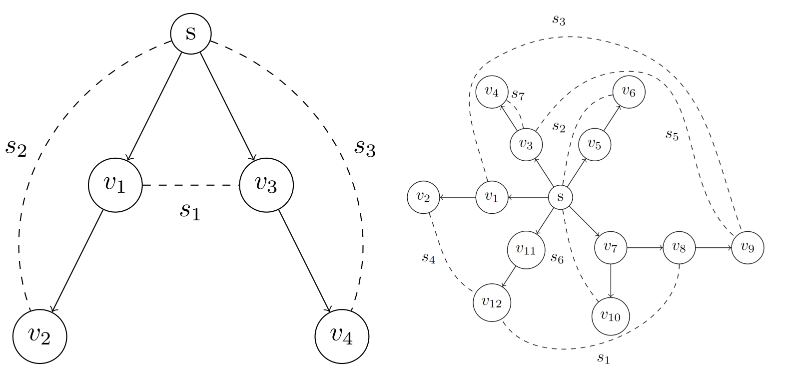

Fig. 3. *Two networks where the greedy algorithm for MRT does not achieve the optimum value.*

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 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.

5.2. Kernel-based rounding for c-uniform MRT instances. 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 called kernel-based alpha-point rounding [4]. The idea is to consider a solution to an LP relaxation to minimum sum set cover (MSSC) and transform it to an fractional order by applying a specially chosen kernel. Randomized rounding of the kernelized point then produces an approximation. Though the technique is quite general, the choice of kernel and analysis to bound the approximation factor can vary significantly from instance to instance. The kernels used in Bansal et al.'s work provide a 4-approximation for the general MSSC and a 16/9-approximation for the Minimum 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. These kernels, however, can only guarantee a 4-approximation for general MRT instances.

In this work, we introduce novel kernel functions and new analysis that helps improve the best-known approximation guarantees for general MSSC instances. Given an MRT instance with every switch covering at most c edges of the underlying operational radial network, we first convert such an instance to a minimum sum set cover (MSSC) instance on a c-regular hypergraph. For these instances, our new kernels interpolate the above mentioned approximation factors from $16/9 \approx 1.778$ to 4 dependent on c. Therefore, these kernels strictly improve the state-of-the-art factors for MSSC instances.

By analogy to the zones of protection used in typical protection system designs where three-zone impedance relays are often used [5, 24], 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. To reflect the more general setting, for the remainder of this section we use the hypergraph formulation of MSSC, defined above.

The kernel-based 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 [4], since we found this to be computationally more efficient, but otherwise the formulation does not affect the algorithm[4].

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

If the variables are binaries, then 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 5.2.

In the 2-uniform hypergraph case (Min Sum Vertex Cover), Bansal et al. showed

[4] 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. [4].

Algorithm 5.2 KERNEL α-POINT ROUNDING FOR MSSC [4]

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

that α-point rounding is a 16/9-approximation by using the kernel

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

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

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

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 $2/t \cdot \mathbb{1}[t \geq t']$, 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 $c - 1 \geq 2$.

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 the scheduled time τ. In this part, we show that this gap is upper-bounded by the solution to a optimization problem which achieves its maximum at the vertex of a simplex and thus can be written explicitly. 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 5.3 and A.1).

THEOREM 5.2. *Let $H = (V, E)$ be a c-uniform hypergraph, where $c \geq 3$. Then applying α-point rounding with kernel*

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

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 $e = \{v_1, \ldots, v_c\}$. 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 $q_t(e) = \mathbb{P}[e$ is not scheduled before time t 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.[5] We apply it to our choice of kernel with $\beta = \frac{2c}{c+1}$.

Claim 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,

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

Claim 2. The quantity $p_t(e) = ((1 - z_{e,<t}/c)^c)_+$ upper-bounds $q_t(e)$ and is 0 for sufficiently large t.

For all v, we have that

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

[5] 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.

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

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

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

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

$$
\mathbb{E}[c_\sigma(e)] \leq \sum_t (q_t(e) - q_{t+1}(e))(1 + \beta(t-1) - z_{e,<t} + (\beta - z_{e,t})/2). \tag{5.1}
$$

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 first 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, that 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 since ties are broken at random, yielding the bound of $(\beta - z_{e,t})/2$.

$S(t)$ 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 (5.1) 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

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

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

$$\sum_t (p_t(e) - p_{t+1}(e))(1 + \beta(t-1) - z_{e,<t} + (\beta - z_{e,t})/2) \leq \beta \sum_t t(p_t(e) - p_{t+1}(e))$$
$$\sum_t (p_t(e) - p_{t+1}(e))(1 - z_{e,<t} - z_{e,t}/2) \leq \beta/2 \sum_t (p_t(e) - p_{t+1}(e))$$
$$(1-\beta/2)\sum_t (p_t(e) - p_{t+1}(e)) \leq \sum_t (p_t(e) - p_{t+1}(e))(z_{e,<t} + z_{e,t}/2)$$
$$1 - \beta/2 \leq \sum_t (p_t(e) - p_{t+1}(e))(z_{e,<t} + z_{e,t}/2)$$
$$1/2 \leq \sum_t (z_{e,t}/2)(p_t(e) + p_{t+1}(e))$$
$$1 \leq \sum_t z_{e,t}(p_t(e) + p_{t+1}(e)).$$

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

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

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

and so

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

Claim 5. Finally, we claim that $(2c/(c+1))c_z(e) \geq c_\sigma(e)$.

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

Part B. We would now like to show that $(2c/(c+1))c_x(e) \geq c_z(e)$.

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

Examining the inner sum $\sum_{q=t''}^{t-1} K(q, t'')$ more closely and applying Lemma 5.3 with $p = 2/(c-1)$, we have that

$$\sum_{q'=t''}^{t-1} K(q, t'') = \frac{2c}{c+1} t''^{2/(c-1)} \sum_{q'=t''}^{t-1} \frac{1}{\sum_{i=1}^q i^{2/(c-1)}}$$

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

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

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

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 [4], 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

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

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

Therefore, $c_z(e) \leq 2c/(c+1)c_x(e)$ as desired. ∎

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.

Theorem 5.4. *Let $G = (V, E)$ be the given network, let $T \subseteq E$ be a spanning tree on G with edge weights $b(e) \in \mathbb{R}_+$ for all $e \in T$. Let the set of switches be $S = E \setminus T$, and let $c = \max_{e\in T} |\{s \in S : T - e + s \text{ is connected}\}|$ be the maximum coverage of any edge in the tree. Then, if $c \geq 2$, there is a polynomial-time $(\frac{2c}{c+1})^2$-approximation algorithm for the MRT problem on G.*

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 5.2, or Theorem 22 of [4] 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. ∎

We further show that this approximation ratio is tight with respect to the natural LP relaxation. The following result generalizes Lemmas 26 and 27 of [4] on the integrality gap for MSVC.

Theorem 5.5. *Let $c \geq 2$. Then the integrality gap of the LP for MSSC on c-uniform hypergraphs is at least $(\frac{2c}{c+1})^2$.*

We include a detailed proof in Appendix B. The key idea of the proof is to construct a c-uniform instance out of the union of disjoint complete c-uniform hypergraphs of varying sizes. These complete hypergraphs can be efficiently covered by a fractional LP solution, which completely covers each complete hypergraph in turn, moving from largest to smallest. In contrast, each complete hypergraph requires many time periods to be covered by an optimal integral solution, which can be obtained by the greedy algorithm. The greedy algorithm will select vertices from the largest complete hypergraph until it has as many vertices remaining as the second-largest, then alternately select vertices from both until they have as many vertices remaining as the third-largest, etc. By constructing an instance from increasingly large numbers of increasingly large complete c-uniform hypergraphs, the ratio between the objective values of these two solutions approaches $(2c/(c+1))^2$.

Note that this ratio matches the approximation guarantee of Theorem 5.2, and approaches the general approximation ratio of 4 as $c \to \infty$.

We finally note that 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 7.3.

6. Balancing multiple reliability and energy loss metrics. 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). Moreover, as we discuss in Section 7.1, the choice of optimization metric will affect the expected reconnection times of various types of buses (e.g. in residential or commercial area) differently.

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. The idea of local search for maximizing monotone submodular functions goes back to Calinescu et al. [9]. Our objectives require us to minimize a *product* of sums of supermodular functions over changing network topologies, thus making the analysis of the overall local search process more difficult.

6.1. 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

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

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 C.) However, this program is large and solving it is intractable for larger networks.

The branch exchange search, Algorithm 6.1, 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.,

Algorithm 6.1 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

$T' = T - e + s$ 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.

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.

6.2. Efficient updates. Given a network $G = (V, E)$, a substantial portion of the running time of the branch exchange local search Algorithm 6.1 is updating the connectivity information, i.e. for which pairs of 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 6.1. *Let $G = (V, E)$ be a graph and let $T \subseteq E$ be a spanning tree of G.*

Let $e \in T$, $s_e \in E - T$ be an edge and a switch such that $T' = T - e + s_e$. Then, all covering data for T also hold for the new spanning tree T', except the following pairs (f, s):
(i) f is in the unique cycle C_e in $T + s_e$ and the unique cycle C_s in $T + s$ contains e.
(ii) 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 6.2. *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 6.1. Suppose neither of the conditions holds.
If $T - f + s$ is connected, $f \in C_s$. Since condition (i) does not hold, either $e \notin C_s$ or $f \notin C_e$. Therefore, by Lemma 6.2, $T - e - f + s_e + s = T' - f + s$ is connected.
If $T' - f + s$ is connected, $f \in C'_s$. Since condition (ii) does not hold, either $f \notin C'_e$ or $s_e \notin C'_s$. Therefore, by Lemma 6.2, $T' - s_e - f + e + s = T - f + s$ 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 6.1, 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 6.1, we can save an overall factor of $\Omega(\min\{|E - T|, |V|/d\})$ runtime in each step of the local search procedure.

7. 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.

7.1. Greensboro dataset. The NREL SMART-DS synthetic network covers most of the Greensboro area, including residential, industrial, and other areas, and contains 145052 buses [61]. 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

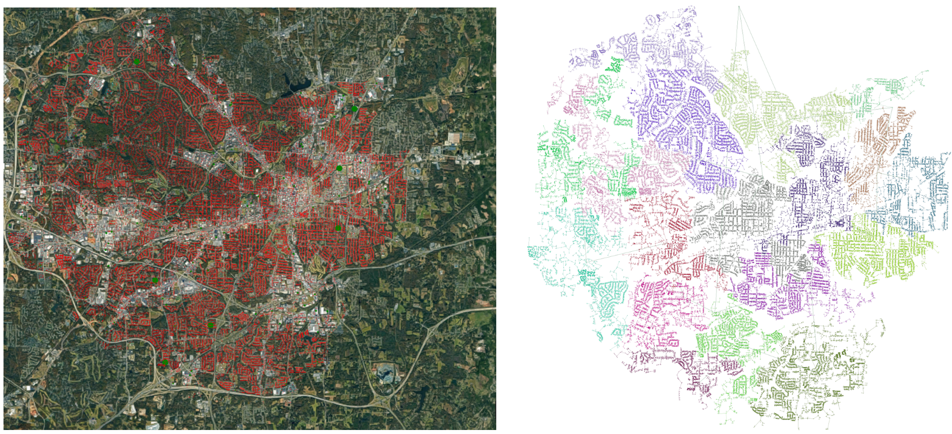

FIG. 4. *(left) Buses shown on a satellite image of Greensboro, NC. (right) Each color is a different connected component in the graph where all the switches (black edges) are removed.*

proportional to length [6]. 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. 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. We give precise pseudocode in Algorithm 7.1. Repeatedly applying these steps significantly reduces the size of the network. Using a threshold of $W = 10$ kW on the Greensboro network, the resulting contracted trees have an average of 17% of the the number of nodes of the original trees. These simplifications primarily consist of contracting leaves corresponding to low-demand buses in residential areas, and do not significantly change the network topology.

7.2. Adding switches. The synthetic network has only 108 switches (non-spanning tree edges) across its 18 distribution networks. To increase coverage, we introduce additional switches, aiming to maximize coverage while minimizing the number and length of the added switches. To do so, we use a two-step procedure for each contracted component.

First, we generate a large set R of candidate switches. We specify a maximum length ℓ. For each pair (u, v), we add the switch s between them if the distance between u and v is at most ℓ and the set of edges covered by s is maximal with respect to inclusion over all such edges.

Algorithm 7.1 (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'.

Once we have obtained a candidate set of switches s, each of length at most ℓ meters, we use a greedy algorithm to choose a set of switches $S \subseteq R$ that obtains good coverage while being as inexpensive as possible to construct. Define the *exposure* of each edge $e \in T$ to be $x(e) = f(e)p(e)$. This quantity, the product of flow and failure probability, corresponds to the coefficient of the reconnection time $t(e)$ in the definition of SAIDI. We then apply a greedy algorithm, which picks switches from R one-by-one to create S. At each step, it picks the switch s that maximizes

$$\sum_{e \text{ covered by } s} 2^{-|\{t \in S : t \text{ covers } e\}|} x(e),$$

so that each time an edge is covered its exposure is halved. The algorithm terminates when each edge that can be covered has been covered.

Note that the additional switches in S are simply *options*. In practice, the coverage gains from adding switches decrease quickly and would soon become outweighed by the associated building and maintenance costs. For most components, excellent coverage (over 90% of total exposure) can be achieved by setting $\ell = 1000$m and adding at most 20 switches from S. Figure 5 shows how coverage improves as the number of switches added increases, and how this depends on the choice of ℓ.

Since they are not all needed to ensure good coverage, the set of switches S has some redundancy, due to the halving of exposures in the greedy algorithm. We make this choice in order to allow for greater flexibility when choosing a reconnection order and reconfiguring the network.

7.3. Approximations for MRT. Having computed additional switch sets S for each of the 18 component networks, using a maximum length of 1000m, we examine reconnection orders for R-Time and SAIDI. To do so, we compare three different approaches for solving MRT:

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

The first two of these are polynomial time, with greedy being extremely computationally 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 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.

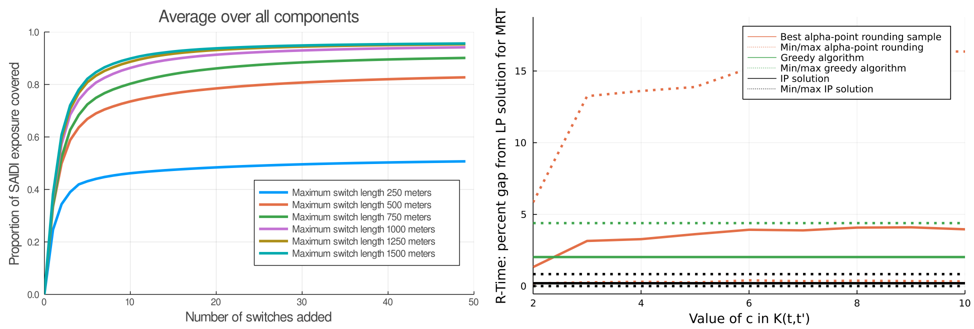

Fig. 5. *Left: The coverage obtained by the switches added by the greedy algorithm for switch addition. For each component, we measure coverage by the fraction* $\sum_{e \text{ covered by } s\in S} x(e)/\sum_{e\in T} x(e)$ *of the total exposure of the tree which is covered by* S*. For different maximum lengths* $\ell = 250,\ 500,\ 750,\ 1000,\ 1250,\ 1500m$*, we compare the average of this fraction over all 18 components. By adding a number of switches equal to an average of 0.18% of the total number of tree edges in each component, we are able to cover 90% of SAIDI exposure. Right: 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 dotted lines indicate the highest and lowest observed percentage differences.*

Table 1

Graphical data and reconnection metrics for the 18 distribution networks. The switches (Sw.) column contains the number of preexisting switches in the component, as well as the number added by the modified greedy algorithm which selects switches from the candidate set with maximum length 1000m. For each metric, the percentage given is the relative increase in objective value (the starred metric) from using the greedy algorithm to optimize for the other metric. R-Time *is given in units of meters* $\cdot\, \varepsilon$*, where the length of each line* e *is proportional to its probability of failure* $p(e)$ *[6].* SAIDI *is given in units of 1000* $(\text{meters} \cdot \varepsilon \cdot kW)$*.*

#	Vert.	Sw.	New	SAIDI*	R-Time	SAIDI	R-Time*	SAIDI gap	R-Time gap
1	8529	4	47	39.0	120.67	41.35	114.1	6.01%	5.75%
2	8396	7	45	32.24	114.54	39.68	108.34	23.07%	5.72%
3	13010	7	49	43.18	110.52	43.43	94.72	0.56%	16.68%
4	9841	15	48	66.82	99.96	75.92	94.48	13.61%	5.8%
5	9319	1	46	27.25	113.82	30.64	97.07	12.45%	17.26%
6	6238	8	47	64.37	172.25	64.75	134.08	0.59%	28.47%
7	8538	3	45	47.27	130.91	55.32	113.02	17.01%	15.83%
8	8147	8	46	51.59	137.96	62.66	121.76	21.45%	13.3%
9	7286	8	42	87.6	120.33	167.51	96.07	91.21%	25.25%
10	5901	5	47	45.88	138.51	67.73	124.93	47.6%	10.87%
11	11031	8	47	52.87	113.93	57.87	99.52	9.44%	14.48%
12	11536	6	49	19.3	91.39	20.63	86.36	6.86%	5.83%
13	2427	3	43	67.45	187.84	86.09	156.84	27.63%	19.77%
14	6950	3	42	23.28	84.32	24.01	78.47	3.15%	7.45%
15	5914	4	46	47.2	160.84	56.68	149.98	20.08%	7.24%
16	7006	2	43	29.86	169.19	31.83	161.11	6.57%	5.01%
17	4008	2	46	24.99	212.38	31.41	199.7	25.69%	6.35%
18	10953	14	46	65.05	125.62	79.44	106.44	22.13%	18.02%

Table 1 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 SAIDI is on average more than 50% greater when greedy attempts to minimize

R-Time than when its objective is SAIDI, and the value of R-Time is on average more than 14% greater when SAIDI is the objective than when R-Time is the objective.

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 dotted lines show the best and worst performance over all components. For α-point rounding, 500 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, usually within a few percentage points of the optimum, with the greedy method sometimes obtaining the optimal solution and sometimes being outperformed by α-point rounding.

7.4. Choice of optimization metric. Of the 145052 buses in the network, 63 run at a medium voltage, with a demand of at least 700 kW, while the rest run at a low voltage and have a demand of at most 350 kW. The medium voltage buses correspond to commercial and industrial areas, while most low voltage buses occur in residential areas. Choosing a reconnection order in each component to minimize SAIDI will tend to prioritize quickly restoring power after the failure of tree edges that supply power to high-demand medium voltage buses, whereas minimizing MRT will tend to prioritize quickly restoring power to the more numerous low voltage buses. Indeed, optimizing for SAIDI produces an 8% longer expected reconnection time for low voltage buses compared to optimizing for R-Time (expected reconnection time for low voltage buses is 2.70 units, when optimizing for R-Time, and 2.91 units when optimizing for SAIDI), but a 33% reduction in expected reconnection time for medium voltage buses (expected reconnection time for medium voltage buses is 3.86 units when optimizing for R-Time, and 2.57 units when optimizing for SAIDI).

7.5. Local search for multi-objective optimization. In the branch exchange Algorithm 6.1 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

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

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

Figure 6 (left) shows the results of 25 different applications of local search to each component of the Greensboro network, each stopping after 100 exchanges or when a local optimum was reached. For each component, we take the local search run achieving the best product, expressing the value of each objective (SAIDI, R-Time, energy, and the product of all three) at each step as a ratio to the initial values on that component. Finally, we plot the average of these ratios over time for all 18 components.

Since the possible branch exchanges are sampled randomly, the results of the local search can vary across multiple runs. Figure 6 (right) shows the results of 50

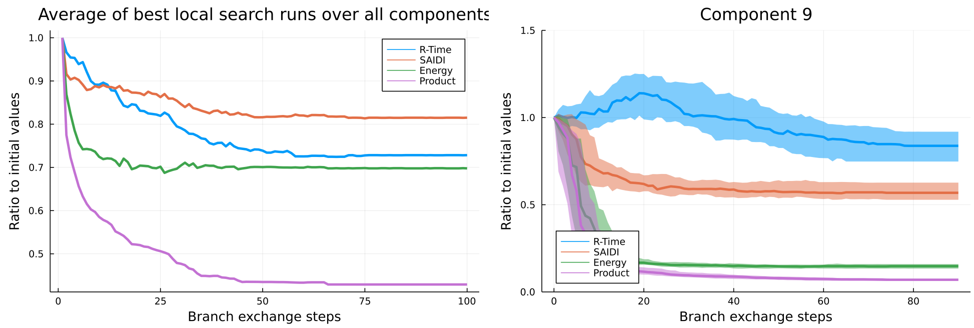

Fig. 6. R-Time, *SAIDI, energy, and product objectives for Greensboro components during the branch exchange local search. Objectives are plotted as a ratio of their value after each step to the initial values. Left: Average best values for each objective, taken over 25 applications of the branch exchange local search on all components. Right: Median values and percentiles (first to last deciles) for each objective, taken over 50 applications of the branch exchange local search on the component 9. Since the local search is on the "product", even though the product decreases, one can see an increase in each of the terms, as shown in the plot. In component 9, we see a significant improvement in energy and smaller improvements in* R-Time *and* SAIDI. *This component highlights the large improvements that can be made by local search, as well as how* R-Time *and* SAIDI *often behave differently.*

different applications of local search to component 9 of the Greensboro network, each stopping after 90 exchanges. For each step, the median of each metric across all 50 runs is plotted in the band between the 10th and 90th percentile values. All three objectives improved substantially during the local search, with energy losses decreasing by 85%, SAIDI by 43%, R-Time by 17%, and their product by 93%. Similar plots for local search on other components are included in Appendix D. Note that while the performance of local search varies from component to component, there are no obvious trends relating the features of the networks for various components to the local search results.

8. Conclusion. We analyzed a decentralized self-healing approach for automatically reconfiguring a distribution system 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 choice of the switches' waiting times, which requires small intervals between each switch's assigned times to avoid creating cycles in the network, is equivalent to choosing a reconnection ordering. The choice of this ordering significantly impacts the expected time to reconnect under normal disruptions and thus affects reliability metrics such as SAIDI, which are the basis for regulator-imposed financial incentives for performance.

We modeled SAIDI and expected reconnection time R-Time 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 efficiently using proxies for MRT subproblems.

We demonstrated that the choice of reconnection metric during the optimization process has a large impact on the behavior of the reconnection order. Optimizing for SAIDI resulted in a 33% reduction in reconnection time for the medium voltage buses that occur in commercial and industrial areas, while optimizing for R-Time reduced reconnection time in the more numerous but lower-demand low voltage buses. Depending on the priorities of the network designers, incorporating both metrics into the optimization process, as we do in our branch exchange local search, may be appropriate.

Our work raises interesting theoretical and practical open questions. First, it is open if the greedy algorithm for MRT attains a worst-case ratio better than 4-factor, as our experiments suggest. Second, local search is currently well-understood for maximization of submodular functions under matroid base constraints [9], 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] 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)

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

[6] 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)

[7] 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)

[8] 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)

[9] 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)

[10] 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)

[11] 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)

[12] 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)

[13] 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)

[14] 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

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

[16] 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)

[17] 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)

[18] 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)

[19] Fokkink, R., Lidbetter, T., Végh, L.A.: On submodular search and machine scheduling. Mathematics of Operations Research **44**(4), 1431–1449 (2019)

[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] Fujito, T.: Approximation algorithms for submodular set cover with applications. IEICE Transactions on Information and Systems **83**(3), 480–487 (2000)

[22] 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)

[23] GE Digital Energy: GENe DMS. https://www.gegridsolutions.com/products/brochures/uos/genedms.pdf (2012)
[24] Glover, J.D., Sarma, M.S., Overbye, T.: Power System Analysis & Design. Cengage Learning (2017)
[25] 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)
[26] Gupta, S., Khodabakhsh, A., Mortagy, H., Nikolova, E.: Electrical flows over spanning trees. Mathematical Programming, Series B pp. 1–41 (2021)
[27] Hall, L.A., Schulz, A.S., Shmoys, D.B., Wein, J.: Scheduling to minimize average completion time: Off-line and on-line approximation algorithms. Mathematics of Operations Research **22**(3), 513–544 (1997)
[28] Happach, F.: Min-sum set cover, OR-scheduling, and related problems. Ph.D. thesis, Technische Universität München (2020)
[29] Happach, F., Schulz, A.S.: Approximation algorithms and LP relaxations for scheduling problems related to min-sum set cover. arXiv preprint arXiv:2001.07011 (2020)
[30] Hermans, B., Leus, R., Matuschke, J.: Exact and approximation algorithms for the expanding search problem. INFORMS Journal on Computing (2021)
[31] Hesmondhalgh, S., Zarakas, W., Brown, T.: Approaches to setting electric distribution reliability standards and outcomes. Tech. rep., The Brattle Group (2012)
[32] IEEE Power and Energy Society, Transmission and Distribution Committee: IEEE guide for electric power distribution reliability indices (2004)
[33] 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)
[34] 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
[35] 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)
[36] Ji-Yuan Fan, Lan Zhang, McDonald, J.D.: Distribution network reconfiguration: Single loop optimization. IEEE Transactions on Power Systems **11**(3), 1643–1647 (1996)
[37] Joskow, P.L.: Incentive Regulation in Theory and Practice: Electricity Distribution and Transmission Networks, pp. 291–344. University of Chicago Press (2014)
[38] Karp, R.M.: Reducibility among combinatorial problems. In: Complexity of computer computations, pp. 85–103. Springer (1972)
[39] 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)
[40] Kersting, W.H.: Distribution System Modeling and Analysis, 4 edn. CRC Press (2017)
[41] 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)
[42] 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)
[43] Kroposki, B., Bernstein, A., King, J., Vaidhynathan, D., Zhou, X., Chang, C.Y., Dall'Anese, E.: Autonomous energy grids: Controlling the future grid with large amounts of distributed energy resources. IEEE Power and Energy Magazine **18**(6), 37–46 (2020)
[44] 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)
[45] 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)
[46] 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)
[47] Lavorato, M., Franco, J.F., Rider, M.J., Romero, R.: Imposing radiality constraints in distribution system optimization problems. IEEE Transactions on Power Systems **27**(1), 172–180 (2012)
[48] Li, W., Li, Y., Chen, C., Tan, Y., Cao, Y., Zhang, M., Peng, Y., Chen, S.: A full decentralized multi-agent service restoration for distribution network with dgs. IEEE Transactions on

Smart Grid **11**(2), 1100–1111 (2020)
[49] Liu, Z., Clark, A., Lee, P., Bushnell, L., Kirschen, D., Poovendran, R.: A submodular optimization approach to controlled islanding under cascading failure. In: Proceedings of the 8th International Conference on Cyber-Physical Systems, pp. 187–196 (2017)
[50] Liu, Z., Clark, A., Lee, P., Bushnell, L., Kirschen, D., Poovendran, R.: Submodular optimization for voltage control. IEEE Transactions on Power Systems **33**(1), 502–513 (2017)
[51] 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)
[52] 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)
[53] 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)
[54] 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)
[55] 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
[56] Mishra, S., Das, D., Paul, S.: A comprehensive review on power distribution network reconfiguration. Energy Systems **8**(2), 227–284 (2017)
[57] 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)
[58] Narayanan, H.: Submodular functions and electrical networks, vol. 54. Elsevier (1997)
[59] PA Consulting: ReliabilityOne Awards: Recognizing outstanding reliability in North American utilities. https://www.paconsulting.com/events/2020/reliabilityone-2020/ (2020)
[60] 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)
[61] 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)
[62] Phillips, C., Stein, C., Wein, J.: Minimizing average completion time in the presence of release dates. Mathematical Programming **82**(1), 199–223 (1998)
[63] Phillips, C.A., Stein, C., Torng, E., Wein, J.: Optimal time-critical scheduling via resource augmentation. In: Proceedings of the twenty-ninth annual ACM Symposium on Theory of Computing, pp. 140–149 (1997)
[64] Pinelis, I.: Answer to "A geometric mean form of the Hermite-Hadamard inequality, for negative powers". mathoverflow.net/questions/388875 (2021)
[65] 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)
[66] 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)
[67] 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)
[68] Schweitzer Engineering Laboratories: SEL Distribution Network Automation. https://cms-cdn.selinc.com/assets/Literature/Product%20Literature/Flyers/DNA_PF00273.pdf?v=20170530-182047 (2017)
[69] 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)
[70] Skutella, M.: List scheduling in order of α-points on a single machine. In: Efficient Approximation and Online Algorithms, pp. 250–291. Springer (2006)
[71] Srivastava, U., Munagala, K., Widom, J.: Operator placement for in-network stream query processing. In: Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, pp. 250–258 (2005)
[72] Torres, B.S., Ferreira, L.R., Aoki, A.R.: Distributed intelligent system for self-healing in smart grids. IEEE Transactions on Power Delivery **33**(5), 2394–2403 (2018)
[73] Uluski, R.: Optimal network reconfiguration for electric distribution systems. Tech. Rep.

1022005, EPRI (2012). https://www.epri.com/research/products/1022005
[74] U.S. Energy Information Administration: 2018 state electricity profiles (2019). URL https://www.eia.gov/electricity/state/
[75] 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)
[76] 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)
[77] Wang, Y., Liu, Y., Kirschen, D.S.: Scenario reduction with submodular optimization. IEEE Transactions on Power Systems **32**(3), 2479–2480 (2016)
[78] Wang, Y., Wang, S., Wu, L.: Distributed optimization approaches for emerging power systems operation: A review. Electric Power Systems Research **144**, 127–135 (2017)
[79] 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)
[80] Ward, D.M.: The effect of weather on grid systems and the reliability of electricity supply. Climatic Change **121**, 103–113 (2013)
[81] Xu, J., Zhang, T., Du, Y., Zhang, W., Yang, T., Qiu, J.: Islanding and dynamic reconfiguration for resilience enhancement of active distribution systems. Electric Power Systems Research **189**, 106749 (2020)
[82] 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)
[83] 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)
[84] Zhang, Y., Monrose, F., Reiter, M.K.: The security of modern password expiration: An algorithmic framework and empirical analysis. In: Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS '10, p. 176–186 (2010)

Appendix A. Proof of Lemma 5.3.

Lemma 5.3 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

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

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 [64].[6]

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

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

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 $g(v,q) = v^{q+1} - 1 - (q+1)(v-1)v^{q/2}$,

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

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. ∎

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

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

Therefore $f(t+1) \geq f(t) + 1/p$, and from Equation A.1 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. ∎

[6] 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.

Appendix B. Proof of Theorem 5.5.

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

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

Appendix C. 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:

$$\begin{aligned}
\text{minimize} \quad & \sum_{e,t} t \cdot b(e) x_{et} \\
\text{subject to} \quad & \sum_{e \subset C} s_e \le |C| - 1, \quad \forall \text{ cuts } C, \\
& \sum_{e} s_e = n - 1, \\
& \sum_{t} x_{et} = s_e, \quad \forall e, \\
& x_{et} \le \sum_{f} c_{eft} \quad \forall e, t, \\
& c_{eft} \le y_{ft}, \quad \forall e, f, t, \\
& y_{et} \le 1 - s_e, \quad \forall e, t, \\
& \sum_{e} y_{et} = 1, \quad \forall t, \\
& c_{eft} \le \sum_{g \text{ across } C} s_g - s_e, \quad \forall e, f, t, \text{cuts } C \text{ crossed by } e \text{ but not } f,
\end{aligned}$$

$$w(e) \geq |C|\left(1 - \sum_{f \text{ crossing } C} s_f - s_e\right), \quad \forall e, \text{cuts } C \text{ not containing the source vertex}$$

all variables binary except $w(e) \geq 0$

This exponential-size formulation is intractable for large graphs, so we adapt the $O(mn)$-size formulation due to Martin [52]. 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.

$$\begin{aligned}
\text{minimize} \quad & \sum_{e,t} t \cdot b(e) x_{et} \\
\text{subject to} \quad & \sum_{e} s_e = n - 1, \\
& s_e = a_{k,i,j} + a_{k,j,i}, \quad \forall e \in E, 1 \leq k \leq n, \\
& \sum_{j} a_{k,i,j} \leq 1, \quad \forall 1 \leq i, k \leq n, \\
& a_{k,k,j} = 0, \quad \forall 1 \leq j, k \leq n, \\
& \sum_{t} x_{et} = s_e, \quad \forall e, \\
& x_{et} \leq \sum_{f} c_{eft} \quad \forall e, t, \\
& c_{eft} \leq y_{ft}, \quad \forall e, f, t, \\
& y_{et} \leq 1 - s_e, \quad \forall e, t, \\
& \sum_{e} y_{et} = 1, \quad \forall t, \\
& c_{eft} \leq r_{ef}, \quad \forall e, f, t, \\
& r_{ef} \leq 1 - s_e, \quad \forall e, f \\
& \sum_{g} s_{e,f,g} = n - 1, \quad \forall e, f \\
& s_{e,f,e} = 0, \quad \forall e, f \\
& s_{e,f,f} = 0, \quad \forall e, f \\
& s_{e,f,g} = b_{e,f,k,i,j} + b_{e,f,k,j,i}, \quad \forall e, f \in E, g \in E \setminus \{e\}, 1 \leq k \leq n, \\
& \sum_{j} b_{e,f,k,i,j} \leq 1 + n(1 - r_{ef}), \quad \forall e, f, \in E, 1 \leq i, k \leq n, \\
& b_{e,f,k,k,j} = 0, \quad \forall e, f \in E, 1 \leq j, k \leq n, \\
& \text{all variables binary}
\end{aligned}$$

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.

Appendix D. Plots for local search on other Greensboro components.

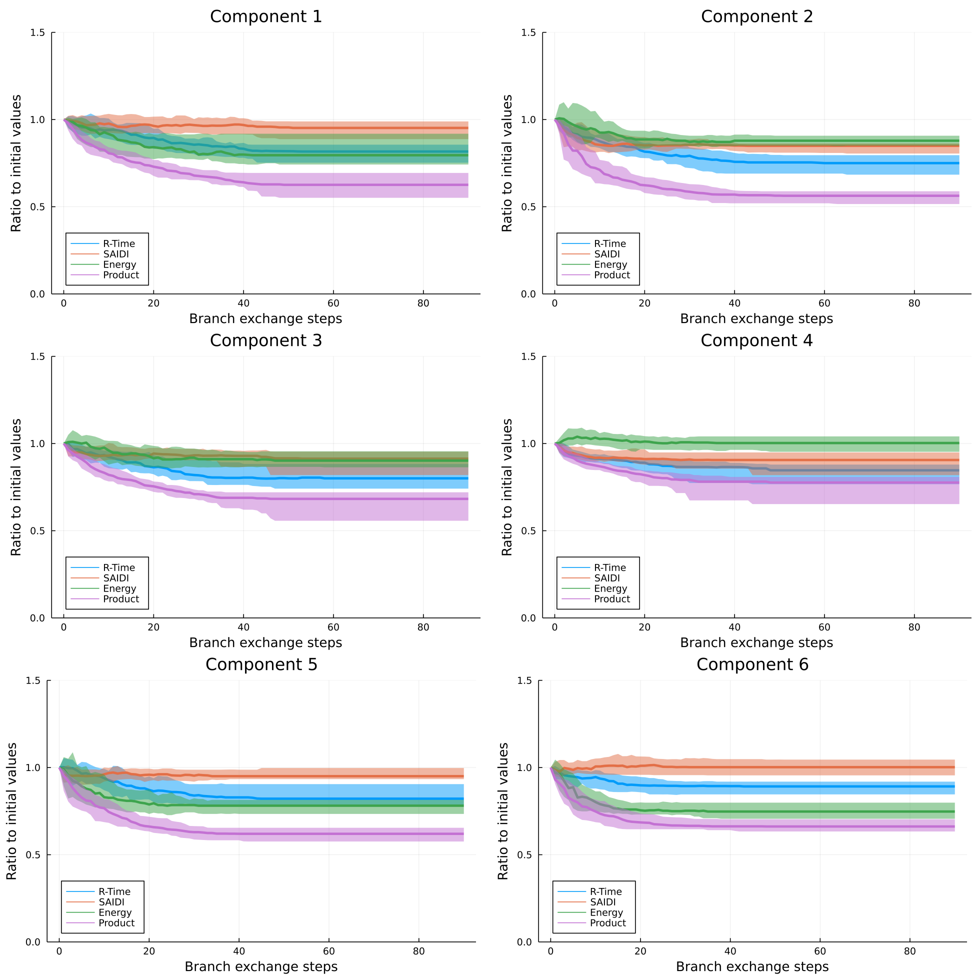

Fig. 7. R-Time, *SAIDI, energy, and product objectives for Greensboro components 1-6 during the branch exchange local search. Objectives are plotted as a ratio of their value after each step to the initial values, and each plot shows the median values and percentiles (first to last deciles) for each objective, taken over 50 applications of the branch exchange local search on the component.*

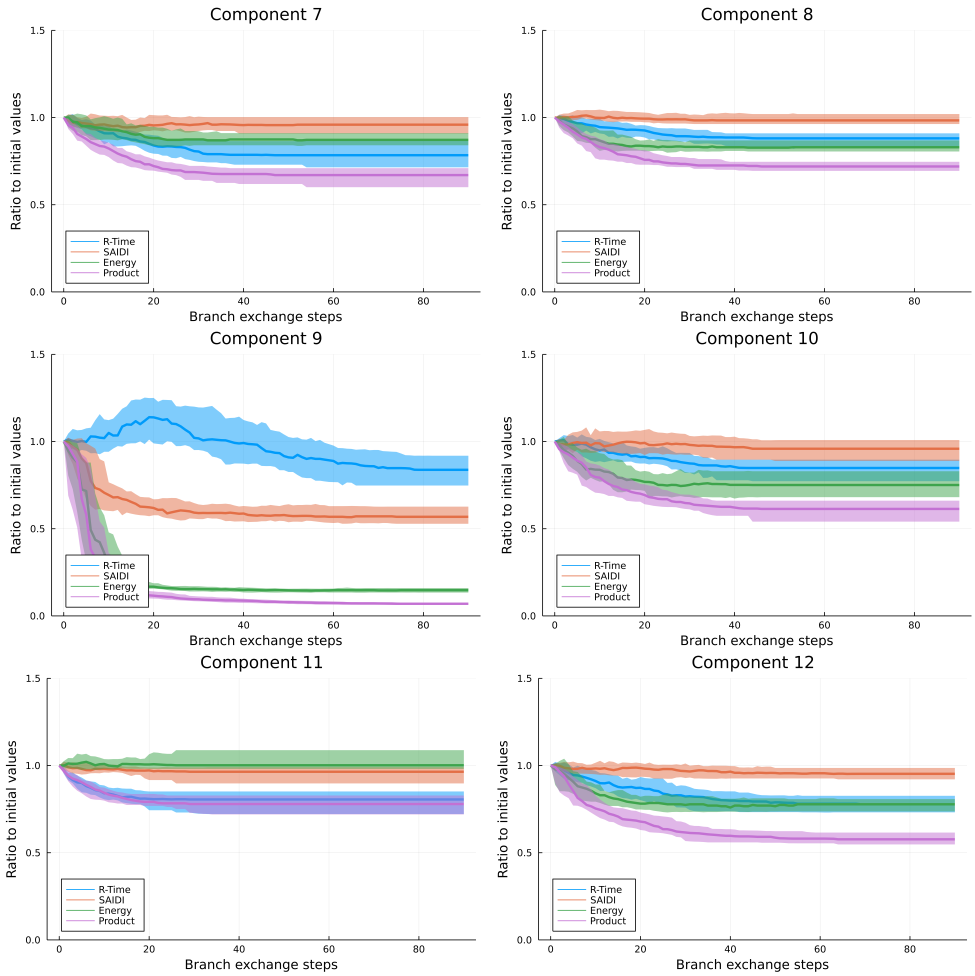

Fig. 8. R-Time, *SAIDI, energy, and product objectives for Greensboro components 7-12 during the branch exchange local search. Objectives are plotted as a ratio of their value after each step to the initial values, and each plot shows the median values and percentiles (first to last deciles) for each objective, taken over 50 applications of the branch exchange local search on the component.*

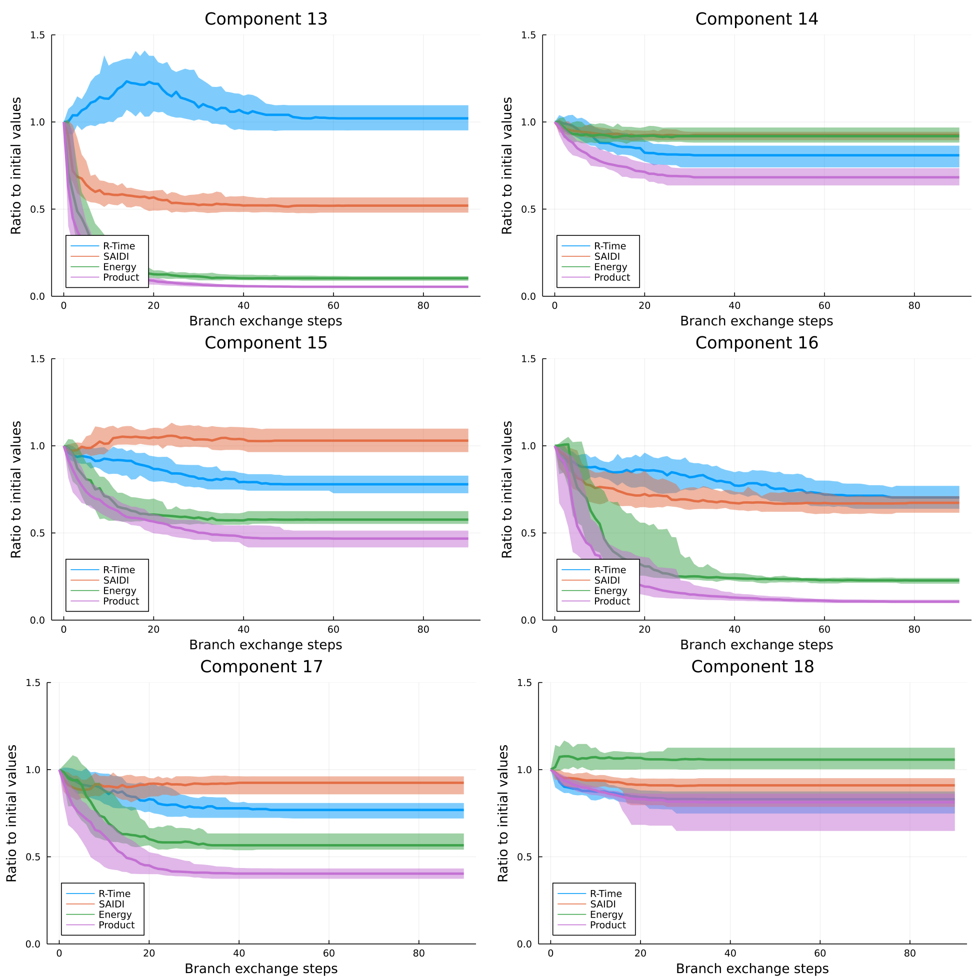

FIG. 9. R-TIME, *SAIDI, energy, and product objectives for Greensboro components 13-18 during the branch exchange local search. Objectives are plotted as a ratio of their value after each step to the initial values, and each plot shows the median values and percentiles (first to last deciles) for each objective, taken over 50 applications of the branch exchange local search on the component.*