

Scheduling Post-Disaster Repairs in Electricity Distribution Networks

Yushi Tan, Feng Qiu, Arindam K. Das, Daniel S. Kirschen,
Payman Arabshahi, Jianhui Wang

Abstract

Natural disasters, such as hurricanes, earthquakes and large wind or ice storms, typically require the repair of a large number of components in electricity distribution networks. Since power cannot be restored before these repairs have been completed, optimally scheduling the available crews to minimize the cumulative duration of the customer interruptions reduces the harm done to the affected community. Considering the radial network structure of the distribution system, this repair and restoration process can be modeled as a job scheduling problem with soft precedence constraints. As a benchmark, we first formulate this problem as a time-indexed ILP with valid inequalities. Three practical methods are then proposed to solve the problem: (i) an LP-based list scheduling algorithm, (ii) a single to multi-crew repair schedule conversion algorithm, and (iii) a scheduling algorithm based on ρ -factors which can be interpreted as Component Importance Measures. We show that the first two algorithms are 4 and $(2 - \frac{1}{m})$ approximations respectively. We also prove that the latter two algorithms are equivalent. Numerical results validate the effectiveness of the proposed methods.

Keywords

Electricity distribution network, Disaster management, Infrastructure resilience, Scheduling with soft precedence constraints, Time-indexed integer programming, LP-based list scheduling, Conversion algorithm

1 Introduction

Natural disasters, such as Hurricane Sandy in November 2012, the Christchurch Earthquake of February 2011 or the June 2012 Mid-Atlantic and Midwest Derecho, caused major damage to the electricity distribution networks and deprived homes and businesses of electricity for prolonged periods. Such power outages carry heavy social and economic costs. Estimates of the annual cost of power outages caused by severe weather between 2003 and 2012 range from \$18 billion to \$33 billion on average [8]. Physical damage to grid components must be repaired before power can be restored. [4] [16]. Hurricanes often cause storm surges that flood substations and corrode metal, electrical components and wiring [28]. Earthquake can trigger ground liquefaction that damage buried cables and dislodge transformers [13]. Wind and ice storms bring down trees, breaking

overhead cables and utility poles [11]. As the duration of an outage increases, its economic and social costs rise exponentially. See [31] [22] for discussions of the impacts of natural disasters on power grids and [30] [23] for its impact on other infrastructures.

It is important to distinguish the distribution repair and restoration problem discussed in this paper from the blackout restoration problem and the service restoration problem. Blackouts are large scale power outages (such as the 2003 Northeast US and Canada blackout) caused by an instability in the power generation and the high voltage transmission systems. This instability is triggered by an electrical fault or failure and is amplified by a cascade of component disconnections. Restoring power in the aftermath of a blackout is a different scheduling problem because most system components are not damaged and only need to be re-energized. See [1] [2] for a discussion of the blackout restoration problem and [27] for a mixed-integer programming approach for solving this problem. On the other hand, service restoration focuses on re-energizing a part of the local, low voltage distribution grid that has been automatically disconnected following a fault on a single component or a very small number of components. This can usually be done by isolating the faulted components and re-energizing the healthy parts of the network using switching actions. The service restoration problem thus involves finding the optimal set of switching actions. The repair of the faulted component is usually assumed to be taking place at a later time and is not considered in the optimization model. Several approaches have been proposed for the optimization of service restoration such as heuristics [29] [10], knowledge based systems [15], and dynamic programming [18].

Unlike the outages caused by system instabilities or localized faults, outages caused by natural disasters require the repair of numerous components in the distribution grid before consumers can be reconnected. The research described in this paper therefore aims to schedule the repair of a significant number of damaged components, so that the distribution network can be progressively re-energized in a way that minimizes the cumulative harm over the total restoration horizon. Fast algorithms are needed to solve this problem because it must be solved immediately after the disaster and may need to be re-solved multiple times as more detailed information about the damage becomes available. Relatively few papers address this problem. Coffrin and Van Hentenryck [7] propose a technique to co-optimize the sequence of repairs, the load pick-ups and the generation dispatch. However, the sequencing of repair does not consider the fact that more than one repair crew could work at the same time. Nurre et al. [17] formulate an integrated network design and scheduling (INDS) problem with multiple crews, which focuses on selecting a set of nodes and edges for installation in general infrastructure systems and scheduling them on work groups. They also propose a heuristic dispatch rule based on network flows and scheduling. The rest of the paper is organized as follows. In Section 2, we define the problem of optimally scheduling multiple repair crews in a radial electricity distribution network after a natural disaster, and show that this problem is at least strongly \mathcal{NP} -hard. In Section 3, we formulate the post-disaster repair problem as an integer linear programming (ILP) using a multi-commodity flow model, analyze its complexity, and present a family of valid inequalities. Subsequently, we propose three polynomial time algorithms based on adaptations of known algorithms in parallel machine scheduling theory, and provide performance bounds on their worst-case performance. A list scheduling algorithm based on LP relaxation of the ILP model is discussed in Section 4; an algorithm which converts the optimal single crew repair sequence to a multi-crew repair sequence

is presented in Section 5; and a heuristic dispatch rule based on ρ -factors, shown to be equivalent to the conversion algorithm, is addressed in Section 6. In Section 7, we apply these methods to several standard test models of distribution networks. Section 8 draws conclusions.

2 Problem formulation

A distribution network can be represented by a graph G with the set of nodes N and the set of edges (a.k.a, lines) L . We assume that the network topology G is radial, which is a valid assumption for most electricity distribution networks. Let $S \subset N$ represent the set of source nodes which are initially energized and $D = N \setminus S$ represent the set of sink nodes where consumers are located. An edge in G represents a distribution feeder or some other connecting component. Severe weather can damage these components, resulting in a widespread disruption of power supply to the consumers. Let L^D and $L^I = L \setminus L^D$ denote the sets of damaged and intact edges, respectively. Without any loss of generality, we assume that there is only one source node in G . If an edge is damaged, all downstream nodes lose power due to lack of electrical connectivity. Therefore, our goal is to find a schedule by which the damaged lines should be repaired such that the aggregate harm due to loss of electric power is minimized. We define this harm as follows:

$$\sum_{n \in N} w_n T_n, \quad (1)$$

where w_n is a positive quantity that captures the importance of the load at node n and T_n is the time required to restore power at node n (or the energization time of node n). Of course, the energization time of the source nodes is 0, i.e., $T_n = 0$, $\forall n \in S$. The concept of node energization time is critical and merits further elaboration. Consider, for example, a simple 3-node chain graph, $i \rightarrow j \rightarrow k$, where node i is the source. Also, assume that the repair times of the lines $i \rightarrow j$ and $j \rightarrow k$ are 5 and 10 time units respectively. Suppose that we have only one repair crew and that line $j \rightarrow k$ is repaired first, followed by $i \rightarrow j$. If the repair process starts at time $t = 0$, both nodes j and k can be energized at time $t = 15$ because node k cannot be energized until a path has been repaired from the source all the way to k .

The importance of a node depends on the amount of load connected to it as well as the type of load served. For example, re-energizing a hospital would receive a higher priority than a similar amount of residential load. These priority factors would need to be assigned by the utility companies and their determination is outside the scope of this paper. In this paper, we consider both the case where a single crew must carry all the repairs and the case where multiple crews work simultaneously and independently on the repair of separate lines. Each line $l \in L$ is assumed to have a capacity \bar{f}_l and each damaged line $l \in L^D$ requires a repair time p_l which is determined by the extent of damage and the location of l . We assume that all repair times are integers and that it would take every crew the same amount of time to repair the same damaged line. We also assume that crew travel times are minimal and can be either ignored or factored into the component repair times. Instead of a rigorous power flow model, we model network connectivity using a simple network flow model, i.e., as long as a sink node is connected to the source, we assume that all the load on this node can be supplied without violating any line capacity constraint. For simplicity, we treat the three-phase distribution network as if it

were a single phase system. Our analysis could be extended to a three-phase system using a multi-commodity flow model, as in [34].

We construct two simplified directed radial graphs to model the effect that the topology of the distribution network has on scheduling. The first graph, G' , is called the ‘damaged component graph’. All nodes in G that are connected by intact edges are merged into a supernode in G' . The set of edges in G' is the set of damaged lines in G , L^D . From a computational standpoint, the nodes of G' can be obtained by treating the edges in G as undirected, deleting the damaged edges/lines, and finding all the connected components of the resulting graph. The set of nodes in each such connected component represents a (super)node in G' . The edges in G' can then be placed straightforwardly by keeping track of which nodes in G are mapped to a particular node in G' . The direction to these edges can be assigned from a knowledge of the direction of the power flow prior to the disaster. The second graph, P , called a ‘soft precedence constraint graph’, is formally defined in Section 4.2. The nodes in this graph are the damaged lines in G . An edge exists between two nodes if they share the same node in G' . Computationally, the precedence constraints embodied in P can be obtained by finding the shortest path (in a radial network, however, there is a unique path between any pair of nodes) from the source to each leaf node in G and retaining only the damaged lines along each such path. Consider, for example, the IEEE 13-node test feeder [12] shown in Fig. 1. Assume that there are four damaged lines, 650 – 632, 632 – 645, 684 – 611 and 671 – 692. The corresponding G' and P are shown in Fig. 2. Following the procedure discussed above and assuming that node 650 is the source, it can be verified that the precedence constraints are: (i) (650 – 632) \rightarrow (632 – 645), using the path from 650 to 646, (ii) (650 – 632) \rightarrow (684 – 611), using the path from 650 to 611, and (iii) (650 – 632) \rightarrow (671 – 692), using the path from 650 to 675. While these constraints can be concatenated into one tree, as shown in Fig. 2 (b), it is quite possible to end up with multiple disjoint trees (forest). For example, if the damaged lines were 632 – 645, 645 – 646, 671 – 684, 684 – 611 and 684 – 652 instead, the precedence graph would constitute of two disjoint trees, i.e., $P = [\mathcal{T}_1, \mathcal{T}_2]$, where $\mathcal{T}_1 = [(632 - 645) \rightarrow (645 - 646)]$ and $\mathcal{T}_2 = [(671 - 684) \rightarrow (684 - 652); (671 - 684) \rightarrow (684 - 611)]$. If 645 – 646 is deleted from the set of damaged lines, $P = \mathcal{T}_2$ effectively.

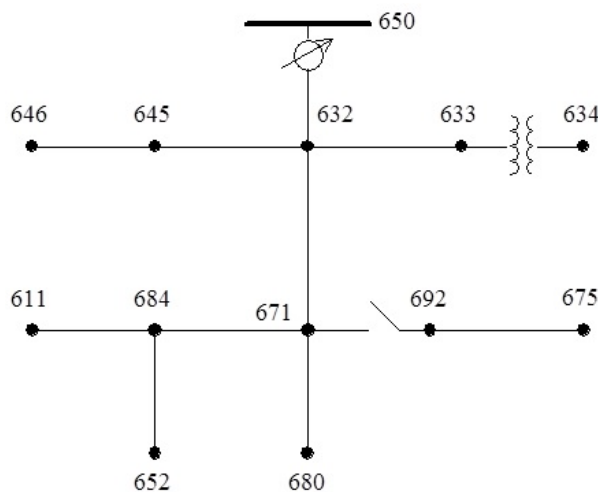


Figure 1: IEEE 13 Node Test Feeder

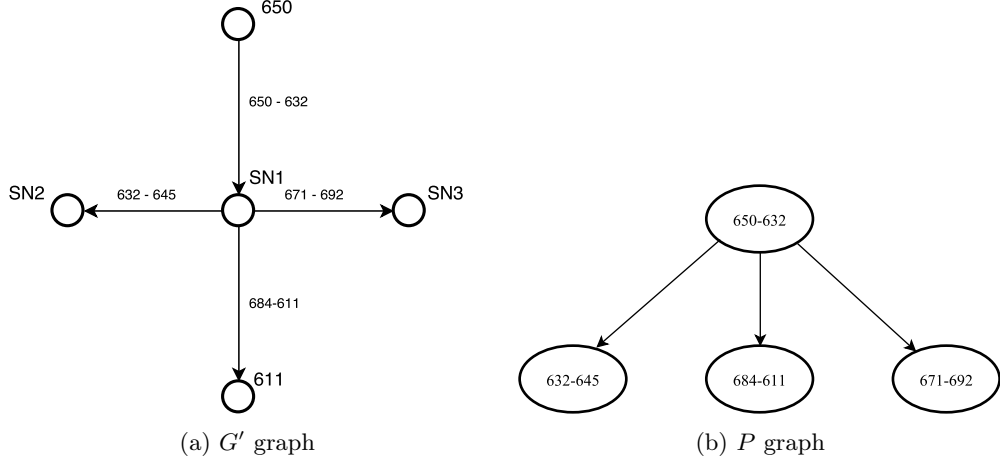


Figure 2: (a) The damaged component graph, G' , obtained from Fig. 1, assuming that the damaged lines are 650 – 632, 632 – 645, 684 – 611 and 671 – 692. In this graph, the supernode SN_1 comprises of the nodes $\{632, 633, 634, 671, 680, 684, 652\}$, $SN_2 = \{645, 646\}$, and $SN_3 = \{692, 675\}$. The set of edges in this graph is the set of damaged lines. (b) The corresponding soft precedence graph, P . An edge exists between the nodes (650 – 632) and (684 – 611) because they share the same node, SN_1 , in G' . As this graph shows, line (650, 632) must be repaired first, allowing for node 632 to be energized. The next three lines that need to be repaired (in any order, since there aren't any precedence constraints among them) are the leaf nodes in the graph, before power can be restored to nodes 645/646, 611 and 692/675.

2.1 Complexity Analysis

In this section, we study the computational complexity of the multi-crew repair scheduling problem and show that it is at least strongly \mathcal{NP} -hard. We first state the following theorem [19] [6]:

Theorem 1. *The job scheduling problem, where the weighted completion time is to be minimized using P identical parallel machines, denoted by $P \parallel \sum_j w_j C_j$, is strongly \mathcal{NP} -hard.*

Proof. We show that a specific instance of the multi-crew repair problem can be reduced to the scheduling problem $P \parallel \sum_j w_j C_j$. Consider a star network where the source (the ‘hub’ node, say, a substation) is directly connected to n sink nodes (consumers), with w_j denoting the weight of node j , $j = 1, \dots, n$. Suppose the n edges between the source and the sinks are all damaged and m repair crews are dispatched, $m < n$. As illustrated in the paragraph immediately following eqn. 1, generally speaking, a particular node, say k , cannot be deemed to be energized as soon as the preceding line, say $j \rightarrow k$, has been repaired. This is due to the electrical continuity constraint which requires that a ‘repaired path’ exist from any source to node k before power can be restored to this node. In the case of a star network, however, this constraint is redundant because any of the sink nodes can be energized as soon as the edge connecting it to the source/hub node has been repaired. Since the energization times of the sink nodes are now independent of the availability of ‘paths’ from the source, as also the energization times of other nodes, minimization of the total weighted completion time in this problem instance is analogous to $P \parallel \sum_j w_j C_j$, with $P = m$, and is therefore strongly \mathcal{NP} -hard. Consequently, the general multi-crew repair scheduling problem is at least strongly \mathcal{NP} -hard. \square

3 Integer Linear Programming (ILP) formulation

We model the job sequencing using time-indexed decision variables, x_l^t , where $x_l^t = 1$ if line l is being repaired by a crew at time period t . Variable y_l^t denotes the repair status of line l where $y_l^t = 1$ if the repair is done by the end of time period $t - 1$ and ready to energize at time period t . Finally, $u_i^t = 1$ if node i is energized at time period t . Let T denote the time horizon for the restoration efforts. Although we cannot know T exactly until the problem is solved, a conservative estimate should work. Since $T_i = \sum_{t=1}^T (1 - u_i^t)$ by discretization, the objective function of eqn. 1 can be rewritten as:

$$\text{minimize} \quad \sum_{t=1}^T \sum_{i \in N} w_i (1 - u_i^t) \quad (2)$$

This problem is to be solved subject to two sets of constraints: (i) repair constraints and (ii) network flow constraints, which are discussed next. We mention in passing that the above time-indexed (ILP) formulation provides a strong relaxation of the original problem [17] and allows for modeling of different scheduling objectives without changing the structure of the model and the underlying algorithm.

3.1 Repair Constraints

Repair constraints model the behavior of repair crews and how they affect the status of the damaged lines and the sink nodes that must be re-energized. The three constraints below are used to initialize the binary status variables y_l^t and u_i^t . Eqn. 3 forces $y_l^t = 0$ for all lines which are damaged initially (i.e., at time $t = 0$) while eqn. 4 sets $y_l^t = 1$ for all lines which are intact. Eqn. 5 forces the status of all source nodes, which are initially energized, to be equal to 1 for all time periods.

$$y_l^1 = 0, \quad \forall l \in L^D \quad (3)$$

$$y_l^t = 1, \quad \forall l \in L^I, \quad \forall t \in [1, T] \quad (4)$$

$$u_i^t = 1, \quad \forall i \in S, \quad \forall t \in [1, T] \quad (5)$$

where T is the restoration time horizon. The next set of constraints is associated with the binary variables x_l^t . Eqn. 6 constrains the maximum number of crews working on damaged lines at any time period t to be equal to m , where m is the number of crews available.

$$\sum_{l \in L^D} x_l^t \leq m, \quad \forall t \in [1, T] \quad (6)$$

Observe that, compared to the formulation in [17], there are no crew indices in our model. Since these indices are completely arbitrary, the number of feasible solutions can increase in crew indexed formulations, leading to enhanced computation time. For example, consider the simple network $i \rightarrow j \rightarrow k \rightarrow l$, where node i is the source and all edges require a repair time of 5 time units. If 2 crews are available, suppose the optimal repair schedule is: ‘assign team 1 to $i \rightarrow j$ at time $t = 0$, team 2 to $j \rightarrow k$ at $t = 0$, and team 1 to $k \rightarrow l$ at $t = 5$. Clearly, one possible equivalent solution conveying the same repair schedule and yielding the same cost, is:

‘assign team 2 to $i \rightarrow j$ at $t = 0$, team 1 to $j \rightarrow k$ at $t = 0$, and team 1 to $k \rightarrow l$ at $t = 5$ ’. In general, formulations without explicit crew indices may lead to a reduction in the size of the feasible solution set. Although the optimal repair sequences obtained from such formulations do not natively produce the work assignments to the different crews, this is not an issue in practice because operators can choose to let a crew work on a line until the job is complete and assign the next repair job in the sequence to the next available crew (the first m jobs in the optimal repair schedule can be assigned arbitrarily to the m crews).

Finally, constraint eqn. 7 formalizes the relationship between variables x_l^t and y_l^t . It mandates that y_l^t cannot be set to 1 unless at least p_l number of x_l^τ 's, $\tau \in [1, t - 1]$, are equal to 1, where p_l is the repair time of line l .

$$y_l^t \leq \frac{1}{p_l} \sum_{\tau=1}^{t-1} x_l^\tau, \quad \forall l \in L^D, \quad \forall t \in [1, T] \quad (7)$$

While we do not explicitly require that a crew may not leave its current job unfinished and take up a different job, it is obvious that such a scenario cannot be part of an optimal repair schedule.

3.2 Network flow constraints

As mentioned previously, electrical continuity dictates that a sink node can be energized only if there is a ‘repaired path’ from any source to that node. We use a modified form of standard flow equations to impose this continuity requirement. Specifically, we require that the flows, originating from the source nodes (eqn. 8), travel through lines which have already been repaired (eqn. 9). Once a sink node receives a flow, it can be energized (eqn. 10).

$$\sum_{l \in \delta_G^-(i)} f_l^t \geq 0, \quad \forall t \in [1, T], \quad \forall i \in S \quad (8)$$

$$-M \times y_l^t \leq f_l^t \leq M \times y_l^t, \quad \forall t \in [1, T], \quad \forall l \in L \quad (9)$$

$$u_i^t \leq \sum_{l \in \delta_G^+(i)} f_l^t - \sum_{l \in \delta_G^-(i)} f_l^t, \quad \forall t \in [1, T], \quad \forall i \in D \quad (10)$$

In eqn. 9, M is a suitably large constant, which, in practice, can be set equal to the number of sink nodes, $M = |D|$. In eqn. 10, $\delta_G^+(i)$ and $\delta_G^-(i)$ denote the sets of lines on which power flows into and out of node i in G respectively.

3.3 Valid inequalities

Valid inequalities typically reduce the computing time and strengthen the bounds provided by the LP relaxation of an ILP formulation. We present the following shortest repair time path inequalities, which resemble the ones in [17]. A node i cannot be energized until all the lines between the source s and node i are repaired. Since the lower bound to finish all the associated repairs is $\lfloor S RTP_i / m \rfloor$, where m denotes the number of crews available and $S RTP_i$ denotes the shortest repair time path between s and i , the following inequality is valid:

$$\sum_{t=1}^{\lfloor S RTP_i / m \rfloor - 1} u_i^t = 0, \quad \forall i \in N \quad (11)$$

To summarize, the multi-crew distribution system post-disaster repair problem can be formulated as:

$$\begin{aligned} & \text{minimize} && \text{eqn. 2} \\ & \text{subject to} && \text{eqns. 3} \sim \text{11} \end{aligned} \tag{12}$$

4 List scheduling algorithms based on linear relaxation

A majority of the approximation algorithms used for scheduling is derived from linear relaxations of ILP models, based on the scheduling polyhedra of completion vectors developed in [20] and [24]. We briefly restate the definition of scheduling polyhedra and then introduce a linear relaxation based list scheduling algorithm followed by a worst case analysis of the algorithm.

4.1 Scheduling polyhedron

Let N denote a set of jobs to be scheduled, the ground set. For any $A \subset N$, define:

$$g(A) = \frac{1}{2} \left(\sum_{j \in A} p_j \right)^2 + \frac{1}{2} \sum_{j \in A} p_j^2 \tag{13}$$

Theorem 2 (Wolsey, 1985 [32]; Queyranne, 1993 [20]). *The supermodular polyhedron $Q = \{C \in \mathbb{R}^N : \sum_{j \in A} p_j C_j \geq g(A) \ \forall A \subset N\}$ is the convex hull of the completion time vectors in general single machine scheduling.*

Similarly, define:

$$f(A) = \frac{1}{2m} \left(\sum_{j \in A} p_j \right)^2 + \frac{1}{2} \sum_{j \in A} p_j^2 \tag{14}$$

where m denotes the number of identical machines (analogous to the number of repair crews in our context).

Theorem 3 (Schulz, 1996 [24]). *The supermodular polyhedron $Q = \{C \in \mathbb{R}^N : \sum_{j \in A} p_j C_j \geq f(A) \ \forall A \subset N\}$ is the convex hull of the completion time vectors in general parallel machine scheduling.*

4.2 Linear relaxation of scheduling with soft precedence constraints

A substantial body of research exists on scheduling with precedence constraints. In general, precedence constraints mandate that a job cannot start until its predecessor is complete. Such hard precedence constraints, however, are not applicable in post-disaster restoration. While it is true that a sink node in an electrical network cannot be energized unless there is an intact path (i.e., all damaged lines along that path have already been repaired) from the source (feeder) to this sink node, this does not mean that multiple lines on some path from the source to the sink cannot be repaired concurrently. Instead of explicit continuity constraints discussed in Section 3.2, we adopt a set of soft precedence constraints in this section to ensure that a set of

lines, necessary to establish an intact path from the source, is repaired prior to the energization of any sink node. As mentioned previously, we assume that the distribution network can be modeled as a directed radial graph, which is usually the case in practice.

We keep track of two separate time vectors: the completion times of line repairs, denoted by C_l 's, and the energization times of nodes, denoted by E_n 's. While we have so far associated the term ‘energization time’ with nodes in the given network topology, G , we now show that it is also possible to define energization times on the lines. Consider the example in Fig. 2. The precedence graph, P , requires that the line 650 – 632 be repaired prior to the line 671 – 692. If this (soft) precedence constraint is met, as soon as the line 671 – 692 is repaired, it can be energized, or equivalently, all nodes in SN_3 (nodes 692 and 675) in the damaged component graph, G' , can be deemed to be energized. The energization time of the line 671 – 692 is therefore identical to the energization times of nodes 692 and 675. Before generalizing the above example, we need to define some notations. Given a directed edge l , let $h(l)$ and $t(l)$ denote the head and tail node of l . Let $l = h(l) \rightarrow t(l)$ be any edge in the damaged component graph G' . Provided the soft precedence constraints are met, it is easy to see that $E_l = E_{t(l)}$, where E_l is the energization time of line l and $E_{t(l)}$ is the energization time of the node $t(l)$ in G . Analogously, the weight of node $t(l)$, $w_{t(l)}$, can be interpreted as a weight on the line l , w_l . The soft precedence constraint between two lines i and j , $i \prec j$, indicates that line j cannot be energized unless line i is energized, in which case, $E_j \geq E_i$ must be true. Since the objective of the post-disaster repair and restoration is to minimize the harm, quantified as the weighted energization time, we propose the following LP relaxation:

$$\underset{C, E}{\text{minimize}} \quad \sum_{j \in L^D} w_j E_j \quad (15)$$

$$\text{subject to} \quad C_j \geq p_j, \quad \forall j \in L^D \quad (16)$$

$$E_j \geq C_j, \quad \forall j \in L^D \quad (17)$$

$$E_j \geq E_i, \quad \forall (i \rightarrow j) \in P \quad (18)$$

$$\sum_{j \in A} p_j C_j \geq \frac{1}{2m} \left(\sum_{j \in A} p_j \right)^2 + \frac{1}{2} \sum_{j \in A} p_j^2, \quad \forall A \subset L^D \quad (19)$$

where P is the soft precedence graph discussed in Section 2 (see also Fig. 2). Eqn. 16 constrains the completion time of any damaged line to be lower bounded by its repair time, eqn. 17 ensures that any line cannot be energized until it has been repaired, eqn. 18 models the soft precedence constraints, and eqn. 19 characterizes the scheduling polyhedron.

The above formulation can be simplified a bit, by recognizing that the C_j 's are redundant intermediate variables. Combining eqns. 17 and 19, we have:

$$\sum_{j \in A} p_j E_j \geq \sum_{j \in A} p_j C_j \geq \frac{1}{2m} \left(\sum_{j \in A} p_j \right)^2 + \frac{1}{2} \sum_{j \in A} p_j^2, \quad \forall A \subset L^D \quad (20)$$

which indicates that the vector of E_j 's is in the same scheduling polyhedron as the vector of

C_j 's. After some simple algebra, the LP-relaxation can be reduced to:

$$\text{minimize}_E \quad \sum_{j \in L^D} w_j E_j \quad (21)$$

$$\text{subject to} \quad E_j \geq p_j, \forall j \in L^D \quad (22)$$

$$E_j \geq E_i, \forall (i \rightarrow j) \in P \quad (23)$$

$$\sum_{j \in A} p_j E_j \geq \frac{1}{2m} \left(\sum_{j \in A} p_j \right)^2 + \frac{1}{2} \sum_{j \in A} p_j^2, \quad \forall A \subset L^D \quad (24)$$

We note that although there are exponentially many constraints in the above model, the separation problem for these inequalities can be solved in polynomial time using the ellipsoid method [24].

4.3 LP-based approximation algorithm

List scheduling algorithms, which are among the simplest and most commonly used approximate solution methods for parallel machine scheduling problems [21], assign the job at the top of a priority list to whichever machine is idle first. An LP relaxation provides a good insight into the priorities of jobs and has been widely applied to scheduling with hard precedence constraints. We adopt a similar approach in this paper. Algorithm 1, based on a sorted list of the LP midpoints, summarizes our proposed approach. We now develop an approximation bound for Algorithm 1.

Algorithm 1 Algorithm for single/multiple crew repair scheduling in distribution networks, based on LP midpoints

Let E^{LP} denote any feasible solution to the constraint eqns. 22 - 24. Define the LP mid points to be $M_j^{LP} := E_j^{LP} - p_j/2$, $\forall j \in L^D$. Create a job priority list by sorting the M_j^{LP} 's in an ascending order. Whenever a crew is free, assign to it the next job from the priority list. The first m jobs in the list are assigned arbitrarily to the m crews.

Lemma 1. *Let S_j^H and C_j^H denote the start time and completion time respectively of some line j in the schedule constructed by Algorithm 1. Then:*

$$S_j^H \leq 2M_j^{LP}, \quad \forall j \in L^D \quad (25)$$

$$C_j^H \leq 2E_j^{LP}, \quad \forall j \in L^D \quad (26)$$

Proof. Define $M := [M_j^{LP} : j = 1, 2, \dots, |L^D|]$. Let \tilde{M} denote M sorted in ascending order, \tilde{I}_j denote the position of some line $j \in L^D$ in \tilde{M} , and $\{k : \tilde{I}_k \leq \tilde{I}_j, k \neq j\} := R$ denote the set of jobs whose LP midpoints are upper bounded by M_j^{LP} . First, we claim that $S_j^H \leq \frac{1}{m} \sum_{i \in R} p_i$. To see why, split the set R into m subsets, corresponding to the schedules of the m crews, i.e., $R = \bigcup_{k=1}^m R^k$. Since job j is assigned to the first idle crew and repairs commence immediately, we have:

$$S_j^H = \min \left\{ \sum_{i \in R^k} p_i : k = 1, 2, \dots, m \right\} \leq \frac{1}{m} \sum_{k=1}^m \sum_{i \in R^k} p_i = \frac{1}{m} \sum_{i \in R} p_i, \quad (27)$$

where the inequality follows from the fact that the minimum of a set of positive numbers is upper bounded by the mean. Next, noting that $M_j^{LP} = E_j^{LP} - p_j/2$, we rewrite eqn. 24 as follows:

$$\sum_{j \in A} p_j M_j^{LP} \geq \frac{1}{2m} \left(\sum_{j \in A} p_j \right)^2, \quad \forall A \subset L^D \quad (28)$$

Now, letting $A = R$, we have:

$$\left(\sum_{i \in R} p_i \right) M_j^{LP} \geq \sum_{i \in R} p_i M_i^{LP} \geq \frac{1}{2m} \left(\sum_{i \in R} p_i \right)^2, \quad (29)$$

where the first inequality follows from the fact that $M_j^{LP} \geq M_i^{LP}$ for any $i \in R$. Combining eqns. 27 and 29, it follows that $S_j^H \leq 2M_j^{LP}$. Consequently, $C_j^H = S_j^H + p_j \leq 2M_j^{LP} + p_j = 2E_j^{LP}$. \square

Lemma 2. Let $D_j^H := E_j^H - C_j^H$ denote the delay in energization of line j after repairs have been completed based on a schedule constructed by Algorithm 1. Then:

$$D_j^H \leq 2E_j^{LP}, \quad \forall j \in L^D \quad (30)$$

Proof. Let $i \preceq j$ denote that job i is a predecessor of j in the precedence graph P . First, we observe that $E_j^H = \max_{i \preceq j} C_i^H$, since l will be energized as soon as repairs on l and all its predecessors are complete. Now, using Lemma 1, we have:

$$C_i^H - C_j^H \leq C_i^H \leq 2E_i^{LP} \leq 2E_j^{LP}, \quad \forall i \preceq j, \quad (31)$$

and consequently,

$$D_j^H = E_j^H - C_j^H = \max_{i \preceq j} C_i^H - C_j^H \leq 2E_j^{LP} \quad (32)$$

\square

Theorem 4. Algorithm 1 is a 4-approximation.

Proof. Noting that $D_j^H := E_j^H - C_j^H$, and using eqns. 26 and 30, we have:

$$E_j^H \leq 4E_j^{LP} \quad (33)$$

Let E_j^* denote the energization time of line j in the optimal schedule. Then, with E_j^{LP} being the solution of the linear relaxation,

$$\sum_{j \in L^D} w_j E_j^{LP} \leq \sum_{j \in L^D} w_j E_j^* \quad (34)$$

Finally, from eqns. 33 and 34, we have:

$$\sum_{j \in L^D} w_j E_j^H \leq 4 \sum_{j \in L^D} w_j E_j^{LP} \leq 4 \sum_{j \in L^D} w_j E_j^* \quad (35)$$

The lemma follows. \square

5 An algorithm for converting the optimal single crew repair sequence to a multi-crew schedule

In practice, many utilities schedule repairs using a priority list [33], which leaves much scope for improvement. We analyze the repair and restoration process as it would be done with a single crew because this provides important insights into the general structure of the multi-crew scheduling problem. Subsequently, we provide an algorithm for converting the single crew repair sequence to a multi-crew schedule and analyze its worst case performance.

5.1 Single crew restoration in distribution networks

We show that this problem is equivalent to $1 \mid \text{outtree} \mid \sum_j w_j C_j$, which stands for scheduling to minimize the total weighted completion time of N jobs with a single machine under ‘outtree’ precedence constraints. Outtree precedence constraints require that each job may have at most one predecessor. Given the manner in which we derive the soft precedence (see Section 2), it is easy to see that the precedence graph P will indeed follow outtree precedence requirements, i.e. each node in P will have at most one predecessor, as long as the network topology G does not have any cycles.

Lemma 3. *Given one repair crew, the optimal schedule in a radial distribution system must follow outtree precedence constraints in the soft precedence constraint graph P .*

Proof. Given one repair crew, each schedule can be represented by a sequence of damaged lines. Let $i - j$ and $j - k$ be two damaged lines such that the node (j, k) is the immediate successor of node (i, j) in the soft precedence graph P . Let π be the optimal sequence and π' another sequence derived from π by swapping $i - j$ and $j - k$. Denote the energization times of nodes j and k in π by E_j and E_k respectively. Similarly, let E'_j and E'_k denote the energization times of nodes j and k in π' . Define $f := \sum_{n \in N} w_n E_n$.

Since node k cannot be energized unless node j is energized and until the line between it and its immediate predecessor is repaired, we have $E'_k = E'_j$ in π' and $E_k > E_j$ in π . Comparing π and π' , we see that node k is energized at the same time, i.e., $E'_k = E_k$, and therefore, $E'_j > E_j$. Thus:

$$\begin{aligned} f(\pi') - f(\pi) &= (w_j E'_j + w_k E'_k) - (w_j E_j + w_k E_k) \\ &= w_j (E'_j - E_j) + w_k (E'_k - E_k) > 0 \end{aligned} \tag{36}$$

Therefore, any job swap that violates the outtree precedence constraints will strictly increase the objective function. Consequently, the optimal sequence must follow these constraints. \square

It follows immediately from Lemma 3 that:

Lemma 4. *Single crew repair and restoration scheduling in distribution networks is equivalent to $1 \mid \text{outtree} \mid \sum_j w_j C_j$, where the outtree precedences are given in the soft precedence constraint graph P .*

5.2 Recursive scheduling algorithm for single crew restoration scheduling

As shown above, the single crew repair problem in distribution networks is equivalent to $1 \mid \text{outtree} \mid \sum_{n \in N} w_n C_n$, for which an optimal algorithm exists [3]. We will briefly discuss this

algorithm and the reasoning behind it. Details and proofs can be found in [6]. Let $J^D \subseteq L^D$ denote any subset of damaged lines. Define:

$$w(J^D) := \sum_{j \in J^D} w_j, \quad p(J^D) := \sum_{j \in J^D} p_j, \quad q(J^D) := \frac{w(J^D)}{p(J^D)}$$

Obviously, for some $j \in J^D$, the above definition reduces to $q(j) = \frac{w_j}{p_j}$. Then, as shown in [6]:

Lemma 5. *Let π be the optimal sequence and I^D and J^D represent two sets of damaged lines in π , to be processed consecutively, such that I^D is scheduled immediately before J^D . If I^D and J^D are disjoint and there is no precedence constraint between any $i \in I^D$ and any $j \in J^D$, then $q(I^D) \geq q(J^D)$.*

Algorithm 2, adapted from [6] with a change of notation, finds the optimal repair sequence by recursively merging the nodes in the soft precedence graph P . The input to this algorithm is the precedence graph P . Let $N(P) = \{1, 2, \dots, |N(P)|\}$ denote the set of nodes in P (representing the set of damaged lines, L^D), with node 1 being the designated root. The predecessor of any node $n \in P$ is denoted by $\text{pred}(n)$. Lines 1 – 7 initialize different variables. In particular, we note that the predecessor of the root is arbitrarily initialized to be 0 and its weight is initialized to $-\infty$ to ensure that the root node is the first job in the optimal repair sequence. Broadly speaking, at each iteration, a node $j \in N(P)$ (m could also be a group of nodes) is chosen to be merged into its immediate predecessor $i \in N(P)$ if $q(j)$ is the largest. The algorithm terminates when all nodes have been merged into the root. Upon termination, the optimal single crew repair sequence can be recovered from the predecessor vector and the element $A(1)$, which indicates the last job finished. For example, consider an arbitrary 5-job scheduling problem where $A(1) = 3$ and the predecessor vector is $\text{pred} = [0, 1, 5, 2, 4]$. The optimal repair sequence is therefore: $\{\text{pred}(2) = 1 \rightarrow \text{pred}(4) = 2 \rightarrow \text{pred}(5) = 4 \rightarrow \text{pred}(3) = 5 \rightarrow A(1) = 3\}$. An illustrative example is shown in Fig. 3. It is shown in Section 4.3.1 of [6] that the algorithm can compute the optimal sequence in $O(n \log n)$ -time.

We conclude this section by noting that Algorithm 2 requires the precedence graph P to have a defined root. However, as illustrated in Section 2, it is quite possible for P to be a forest, i.e., a set of disjoint trees. In such a situation, P can be modified by introducing a dummy root node with a repair time of 0 and inserting directed edges from this dummy root to the roots of each individual tree in the forest. This fictitious root will be the first job in the repair sequence returned by the algorithm, which can then be stripped off.

5.3 Conversion algorithm and an approximation bound

A greedy procedure for converting the optimal single crew sequence to a multiple crew schedule is given in Algorithm 3. We now prove that it is a $(2 - \frac{1}{m})$ approximation algorithm. We start with two lemmas that provide lower bounds on the minimal harm for an m -crew schedule, in terms of the minimal harms for single crew and ∞ -crew schedules. Let $H^{1,*}$, $H^{m,*}$ and $H^{\infty,*}$ denote the minimal harms when the number of repair crews is 1, some arbitrary m ($2 \leq m < \infty$), and ∞ respectively. In the proofs below, we will again associate the term ‘energization time’ with lines, which, as discussed in Section 4.2, is equivalent to energization times of nodes, provided the soft precedence constraints are met.

Algorithm 2 Optimal algorithm for single crew repair and restoration in distribution networks. The input to this algorithm is the precedence graph P . Let $N(P) = \{1, 2, \dots, |N(P)|\}$ denote the set of nodes in P , with node 1 being the designated root. The predecessor of any node $n \in P$ is denoted by $\text{pred}(n)$. Upon termination, the optimal repair sequence can be unraveled from the predecessor vector $\{\text{pred}(n) : n = 1, 2, \dots, |N(P)|\}$, as discussed in the text.

```

1:  $w(1) \leftarrow -\infty$ ;    $\text{pred}(1) \leftarrow 0$ ;
2: for  $n = 1$  to  $|N(P)|$  do
3:    $A(n) \leftarrow n$ ;    $B_n \leftarrow \{n\}$ ;    $q(n) \leftarrow w(n)/p(n)$ ;
4: end for
5: for  $n = 2$  to  $|N(P)|$  do
6:    $\text{pred}(n) \leftarrow \text{parent of } n \text{ in } P$ ;
7: end for
8:  $\text{nodeSet} \leftarrow \{1, 2, \dots, |N(P)|\}$ ;
9: while  $\text{nodeSet} \neq \{1\}$  do
10:  Find  $j \in \text{nodeSet}$  such that  $q(j)$  is largest;   % ties can be broken arbitrarily
11:  Find  $i$  such that  $\text{pred}(j) \in B_i$ ,  $i = 1, 2, \dots, |N(P)|$ ;
12:   $w(i) \leftarrow w(i) + w(j)$ ;
13:   $p(i) \leftarrow p(i) + p(j)$ ;
14:   $q(i) \leftarrow w(i)/p(i)$ ;
15:   $\text{pred}(j) \leftarrow A(i)$ ;
16:   $A(i) \leftarrow A(j)$ ;
17:   $B_i \leftarrow \{B_i, B_j\}$ ;   % ‘,’ denotes concatenation
18:   $\text{nodeSet} \leftarrow \text{nodeSet} \setminus \{j\}$ ;
19: end while

```

Algorithm 3 Algorithm for converting the optimal single crew schedule to an m -crew schedule

Treat the optimal single crew repair sequence as a priority list, and, whenever a crew is free, assign to it the next job from the list. The first m jobs in the single crew repair sequence are assigned arbitrarily to the m crews.

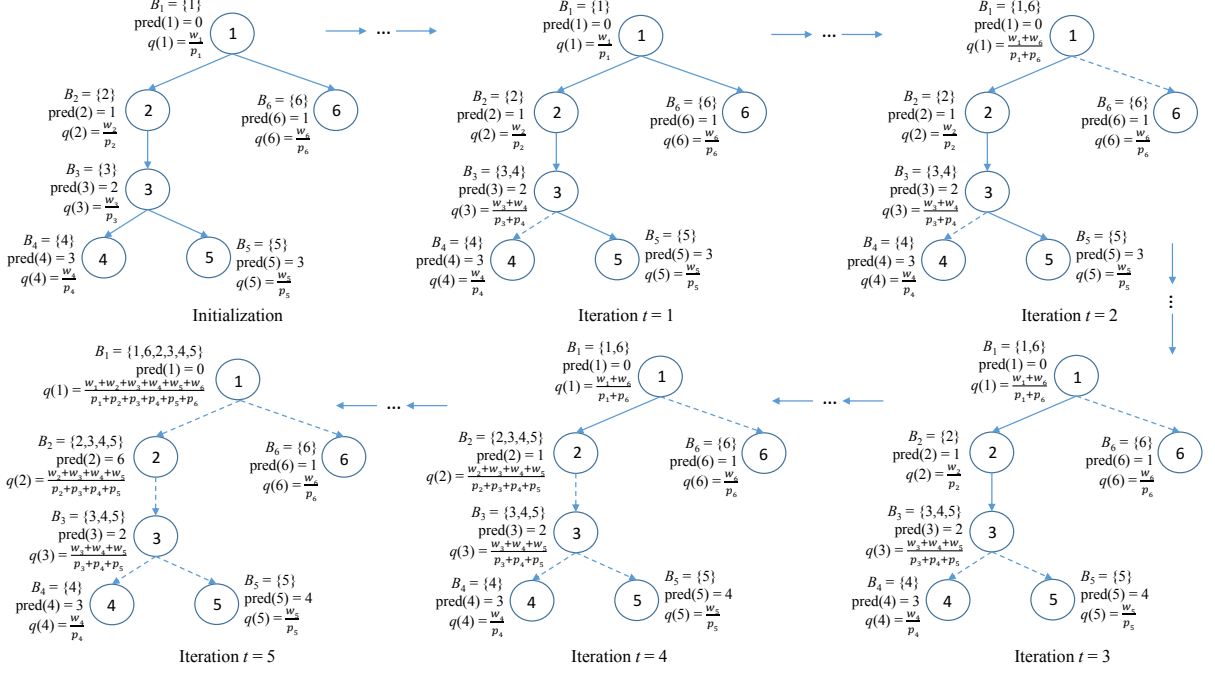


Figure 3: An example illustrating Algorithm 2. The dotted lines indicate which node is chosen for merger at each iteration. For example, node 4 is merged into node 3 at iteration 1. The optimal repair sequence for this example is $[1, 6, 2, 3, 4, 5]$.

Lemma 6. $H^{m,*} \geq \frac{1}{m} H^{1,*}$

Proof. Given an arbitrary m -crew schedule S^m with harm H^m , we first construct a 1-crew repair sequence, S^1 . We do so by sorting the energization times of the damaged lines in S^m in ascending order and assigning the corresponding sorted sequence of lines to S^1 . Ties, if any, are broken according to precedence constraints or arbitrarily if there is none. By construction, for any two damaged lines i and j with precedence constraint $i \prec j$, the completion time of line i must be strictly smaller than the completion time of line j in S^1 , i.e., $C_i^1 < C_j^1$. Additionally, $C_i^1 = E_i^1$ because the completion and energization times of lines are identical for a 1-crew repair sequence which meets the soft precedence constraints in P .

Next, we claim that $E_i^1 \leq mE_i^m$, where E_i^1 and E_i^m are the energization times of line i in S^1 and S^m respectively. In order to prove it, we first observe that:

$$E_i^1 = C_i^1 = \sum_{\{j: E_j^m \leq E_i^m\}} p_j \leq \sum_{\{j: C_j^m \leq E_i^m\}} p_j, \quad (37)$$

where the second equality follows from the manner we constructed S^1 from S^m and the inequality follows from the fact that $C_j^m \leq E_j^m \Rightarrow \{j: E_j^m \leq E_i^m\} \subseteq \{j: C_j^m \leq E_i^m\}$ for any m -crew schedule. In other words, the number of lines that have been energized before line i is energized is a subset of the number of lines on which repairs have been completed before line i is energized. Next, we split the set $\{j: C_j^m \leq E_i^m\} := R$ (say) into m subsets, corresponding to the schedules of the m crews in S^m , i.e., $R = \bigcup_{k=1}^m R^k$, where R^k is a subset of the jobs in R that appear in the k^{th} crew's schedule. It is obvious that the sum of the repair times of the lines in each R^k

can be no greater than E_i^m . Therefore,

$$E_i^1 \leq \sum_{\{j: C_j^m \leq E_i^m\}} p_j := \sum_{j \in R} p_j = \sum_{k=1}^m \left(\sum_{j \in R^k} p_j \right) \leq m E_i^m \quad (38)$$

Proceeding with the optimal m -crew schedule $S^{m,*}$ instead of an arbitrary one, it is easy to see that $E_i^1 \leq m E_i^{m,*}$, where $E_i^{m,*}$ is the energization time of line i in $S^{m,*}$. The lemma then follows straightforwardly.

$$H^{m,*} = \sum_{i \in L^D} w_i E_i^{m,*} \geq \sum_{i \in L^D} w_i \frac{1}{m} E_i^1 = \frac{1}{m} \sum_{i \in L^D} w_i E_i^1 = \frac{1}{m} H^1 \geq \frac{1}{m} H^{1,*} \quad (39)$$

□

Before stating the next lemma, we provide an example which illustrates some of the ideas in the proof of the previous lemma. Consider the graph $G = (a - b - c - d - e)$, where node a is the source. From left to right, the lines are numbered 1, 2, 3, and 4, with repair times 10, 40, 20 and 30 respectively. Assume that all lines are damaged and $m = 2$. Suppose $S^2 = [(\text{crew-1}) : 1, 3; (\text{crew-2}) : 2, 4]$. The energization and completion times of the four lines are: (i) $C_1^2 = E_1^2 = 10$, (ii) $C_2^2 = 40$, $E_2^2 = 40$, (iii) $C_3^2 = 30$, $E_3^2 = 40$, and (iv) $C_4^2 = E_4^2 = 70$. Notice that even though line 3 ($c - d$) is completed at time $t = 30$, it can only be energized at time $t = 40$ since that's when repairs on line 2 ($b - c$) are completed. In fact, repairs on two lines ($a - b$ and $c - d$) have been completed before time $t = E_2^2 = 40$, but only one ($a - b$) has been energized. The precedence graph for this example is $P = (a - b) \rightarrow (b - c) \rightarrow (c - d) \rightarrow (d - e)$. Sorting the energization times in S^2 in ascending order, the 1-crew sequence is: $S^1 = [1, 2, 3, 4]$, where we used P to break a tie between lines 2 and 3. It can be verified that the completion and energization times of the lines are identical in S^1 .

Lemma 7. $H^{m,*} \geq H^{\infty,*}$

Proof. This is intuitive, since the harm is minimized when the number of repair crews is at least equal to the number of damaged lines. In the ∞ -crew case, every job can be assigned to one crew. For any damaged line $j \in L^D$, $C_j^\infty = p_j$ and $E_j^\infty = \max_{i \preceq j} C_i^\infty = \max_{i \preceq j} p_i$. Also, $C_j^{m,*} \geq p_j = C_j^\infty$ and $E_j^{m,*} = \max_{i \preceq j} C_i^{m,*} \geq \max_{i \preceq j} p_i = E_j^\infty$. Therefore:

$$H^{m,*} = \sum_{j \in L^D} w_j E_j^{m,*} \geq \sum_{j \in L^D} w_j E_j^\infty = H^{\infty,*} \quad (40)$$

□

Theorem 5. Let E_j^m be the energization time of line j after the conversion algorithm is applied to the optimal single crew repair schedule. Then, $\forall j \in L^D$, $E_j^m \leq \frac{1}{m} E_j^{1,*} + \frac{m-1}{m} E_j^{\infty,*}$.

Proof. Let S_j^m and C_j^m denote respectively the start and energization times of some line $j \in L^D$ in the m -crew repair schedule, S^m , obtained by applying the conversion algorithm to the optimal 1-crew sequence, $S^{1,*}$. Also, let \mathcal{I}_j denote the position of line j in $S^{1,*}$ and $\{k : \mathcal{I}_k < \mathcal{I}_j\} := R$ (say) denote the set of all lines completed before j in $S^{1,*}$. First, we claim that: $S_j^m \leq \frac{1}{m} \sum_{i \in R} p_i$.

A proof can be constructed by following the approach taken in the proof of Lemma 1 and is therefore omitted. Now:

$$C_j^m = S_j^m + p_j \quad (41)$$

$$\leq \frac{1}{m} \sum_{i \in R} p_i + p_j \quad (42)$$

$$= \frac{1}{m} \sum_{i \in R \cup j} p_i + \frac{m-1}{m} p_j \quad (43)$$

$$= \frac{1}{m} C_j^{1,*} + \frac{m-1}{m} p_j \quad (44)$$

and

$$E_j^m = \max_{i \preceq j} C_i^m \quad (45)$$

$$\leq \max_{i \preceq j} \frac{1}{m} C_i^{1,*} + \max_{i \preceq j} \frac{m-1}{m} p_i \quad (46)$$

$$= \frac{1}{m} C_j^{1,*} + \frac{m-1}{m} \max_{i \preceq j} p_i \quad (47)$$

$$= \frac{1}{m} E_j^{1,*} + \frac{m-1}{m} E_j^{\infty,*} \quad (48)$$

□

Theorem 6. *The conversion algorithm is a $(2 - \frac{1}{m})$ -approximation.*

Proof.

$$H^m = \sum_{j \in L^D} w_j E_j^m \quad (49)$$

$$\leq \sum_{j \in L^D} w_j \left(\frac{1}{m} E_j^{1,*} + \frac{m-1}{m} E_j^{\infty,*} \right) \quad \dots \text{ using Theorem 5} \quad (50)$$

$$= \frac{1}{m} \sum_{j \in L^D} w_j E_j^{1,*} + \frac{m-1}{m} \sum_{j \in L^D} w_j E_j^{\infty,*} \quad (51)$$

$$= \frac{1}{m} H^{1,*} + \frac{m-1}{m} H^{\infty,*} \quad (52)$$

$$\leq \frac{1}{m} (m H^{m,*}) + \frac{m-1}{m} H^{m,*} \quad \dots \text{ using Lemmas 6 - 7} \quad (53)$$

$$= \left(2 - \frac{1}{m} \right) H^{m,*} \quad (54)$$

□

6 Heuristic dispatch rule

In this section, we develop a multi-crew restoration algorithm, proven to be equivalent to the conversion algorithm discussed in the previous section, but from a different perspective. In the process, we define a parameter, $\rho(l)$, $\forall l \in L^D$, which can be interpreted as a ‘component importance measure’ (CIM) in the context of reliability engineering. Towards that goal, we revisit the single crew repair problem, in conjunction with the algorithm proposed in [9].

Let S_l denote the set of all trees rooted at node l in P and $s_l^* \in S_l$ denote the minimal subtree which satisfies:

$$\rho(l) := \frac{\sum_{j \in N(s_l^*)} w_j}{\sum_{j \in N(s_l^*)} p_j} = \max_{s_l \in S_l} \left(\frac{\sum_{j \in N(s_l)} w_j}{\sum_{j \in N(s_l)} p_j} \right), \quad (55)$$

where $N(s_l)$ is the set of nodes in s_l . We define the ratio on the left-hand side of the equality in eqn. 55 to be the ρ -factor of line l , denoted by $\rho(l)$. According to Smith's ratio rule [26], jobs are sequenced in descending order of the ratio w_l/p_l , which indicates that jobs with a larger weight and a smaller repair time have a higher priority. The parameter $\rho(l)$, which is a generalization of the ratio w_l/p_l , characterizes the repair priority of some damaged line l in terms of its own importance as well as the importance of all its successor nodes in P , and can be interpreted as a component importance measure for line l . We refer to the tree s_l^* as the minimal ρ -maximal tree rooted at l , which resembles the minimal ρ -maximal initial sets discussed in [25]. An important characteristic of the ρ -factors is that they can be used to solve the single crew repair scheduling problem optimally, as stated in Algorithm 4 below, adopted from [9].

Algorithm 4 Algorithm for single crew repair scheduling in distribution networks

Whenever the crew is free, say at time t , select among the remaining candidate lines the one with the highest ρ -factor. The candidate set comprises all lines which are connected to the set of energized nodes at time t .

It has been proven in [3] that Algorithms 2 and 4 are equivalent. We now show that the ρ -factors needed in Algorithm 4 can be calculated as a byproduct of Algorithm 2, thereby reinforcing the equivalence of the two algorithms.

Lemma 8. *There exists an optimal schedule such that all lines in the minimal ρ -maximal tree are repaired without interruption.*

A rigorous proof can be constructed following the technique used to prove Lemma 3 in [25], although the two lemmas are not exactly identical. Instead, we provide a more intuitive explanation. We will prove Lemma 8 by contradiction. Assume that the optimal sequence within some s_l^* is $S^* = [l, \dots, 1]$. Suppose that this sequence is interrupted by some other line r which has no precedence constraints between itself and any other node in s_l^* , i.e., the new sequence is $S_1 = [l, \dots, k+1, \{r\}, k, \dots, 1]$. We compare this sequence with two other sequences, $S_2 = [\{r\}, l, \dots, 1]$, and $S_3 = [1, \dots, l, \{r\}]$. Let us assume that S_1 is better than S_2 , in the sense that $\rho_1(l) > \rho_2(l)$, where:

$$\rho_1(l) = \frac{w_l + \dots + w_{k+1} + w_r + w_{k-1} + \dots + w_1}{p_l + \dots + p_{k+1} + p_r + p_{k-1} + \dots + p_1} \quad \text{and} \quad \rho_2(l) = \frac{w_l + w_{l-1} + \dots + w_1}{p_l + p_{l-1} + \dots + p_1}$$

are the ρ factors of node l in S_1 and S_2 respectively. It can then be shown that:

$$\rho_1(l) > \rho_2(l) \Rightarrow \frac{w_r}{p_r} > \frac{w_l + w_{l-1} + \dots + w_1}{p_l + p_{l-1} + \dots + p_1} > \frac{w_l + w_{l-1} + \dots + w_{k+1}}{p_l + p_{l-1} + \dots + p_{k+1}},$$

where the last inequality follows from the definition of the ρ -factor applied to node l in S^* . Therefore,

$$\frac{w_r}{p_r} > \frac{w_l + w_{l-1} + \dots + w_{k+1}}{p_l + p_{l-1} + \dots + p_{k+1}},$$

which implies that S_2 is better than S_1 (see Lemma 5), a contradiction. The same reasoning holds if we assume that S_1 is better than S_3 . In other words, either S_2 or S_3 is always better than S_1 , thereby establishing that all lines in the minimal ρ -maximal tree must be repaired without interruption in an optimal schedule. The rigorous proof should consider the case when the optimal sequence is interrupted multiple times. By comparing Lemma 8 and Algorithm 2, we have the following result.

Theorem 7. *The largest $q(j)$ found in line 10 of Algorithm 2 is the ρ -factor of j , $j \in L^D$, and B_j is the set of nodes in the minimal ρ -maximal tree rooted at j .*

Proof. We prove this theorem by induction on the iterations of the loop. On the very first iteration, the line $j \in L^D$ with the largest w_j/p_j ratio is chosen, and trivially, this ratio is equal to $\rho(j)$. Suppose the theorem holds for iterations 1 through t . When analyzing iteration $t + 1$, we make the following claims.

- Claim 1: In any iteration, $q(j)$ cannot decrease after node k is merged into node j . If $q(j)$ were to decrease, it must be true that $q(j) > q(k)$ before the merger, which means that k could not have been chosen for merger into j in the first place.
- Claim 2: Deleting a node or a group of nodes to B_j cannot increase $q(j)$. Referring to Algorithm 2, let B_j^t denote the set of nodes in B_j at some iteration t . Pick some node $k \in B_j^t$, where k is a successor of j in the precedence graph P . Let A_k^t be any subset of B_k^t , such that $k \in A_k^t$. Suppose A_k^t is deleted from the set B_j^t . For obvious reasons, we need to only address the case where such a deletion does not violate the precedence constraints in P (i.e., if k is removed, all its successors in P which also appear in B_k^t must also be removed). Let $q^t(j)$ and $\tilde{q}^t(j)$ denote the ρ -factors of j before and after the deletion operation. Clearly,

$$q^t(j) = \frac{\sum_{i \in B_j^t} w_i}{\sum_{i \in B_j^t} p_i} \quad \text{and} \quad \tilde{q}^t(j) = \frac{\sum_{i \in \{B_j^t \setminus A_k^t\}} w_i}{\sum_{i \in \{B_j^t \setminus A_k^t\}} p_i}.$$

Since the B_j 's represent an ordered sequence of nodes (see line 17 in Algorithm 2), deletion of the set A_k^t from B_j^t implies that the last node in the residual sequence is the first node which appears to the left of k in B_j^t . We first consider the case when this residual sequence comprises j and at least one other node, i.e., there is at least one other node between j and k in B_j^t . Assume that $\tilde{q}^t(j) > q^t(j)$. If this is true, it can be shown that:

$$\frac{\sum_{i \in \{B_j^t \setminus A_k^t\}} w_i}{\sum_{i \in \{B_j^t \setminus A_k^t\}} p_i} > \frac{\sum_{i \in A_k^t} w_i}{\sum_{i \in A_k^t} p_i},$$

implying that the algorithm should have first recursively merged all successors of node j which are in the set $\{B_j^t \setminus A_k^t\}$ into j , before merging the successors of k in A_k^t into k . This is a contradiction of the optimality of Algorithm 2, and therefore, it must be true that $q^t(j) > \tilde{q}^t(j)$. If the residual sequence comprises j only (i.e., j is immediately to the left of k in B_j^t), $q^t(j) > \tilde{q}^t(j) = \frac{w_j}{p_j}$, by Claim 1.

- Claim 3: Adding a node or a group of nodes to B_j cannot increase $q(j)$. Note that not all nodes are viable candidates for addition. Since the precedence graph is a directed tree and the algorithm works up toward the root, the only node(s) which need to be considered are successors of j in P which are not in B_j^t . Let A_j^t be a subset of the set of successor nodes of j in P , such that $A_j^t \cap B_j^t = \emptyset$. Suppose we add the nodes in A_j^t to B_j^t , without violating any precedence constraint. Let $q^t(j)$ and $\tilde{q}^t(j)$ denote the ρ -factors of j before and after the addition operation, i.e.,

$$q^t(j) = \frac{\sum_{i \in B_j^t} w_i}{\sum_{i \in B_j^t} p_i} \quad \text{and} \quad \tilde{q}^t(j) = \frac{\sum_{i \in \{B_j^t \cup A_j^t\}} w_i}{\sum_{i \in \{B_j^t \cup A_j^t\}} p_i}.$$

Following a similar reasoning as in Claim 2, it can be shown that $q^t(j) > \tilde{q}^t(j)$.

Combining Claims 2 and 3, we see that $q(j) = \rho(j)$ in iteration $t + 1$, and therefore B_j is essentially the set of nodes in the minimal ρ -maximal tree rooted at j . This completes the induction process. \square

Algorithm 4 can be extended straightforwardly to accommodate multiple crews. However, in this case, it could happen that the number of lines that are connected to energized nodes is smaller than the number of repair crews. To cope with this issue, we also consider the lines which are connected to the lines currently being repaired, as described in Algorithm 5 below. However, it turns out, as in Theorem 8, that the candidate sets in both algorithms are actually identical.

Algorithm 5 Algorithm for multi-crew repair scheduling in distribution networks

Whenever a crew is free, say at time t , select among the remaining candidate lines the one with the highest ρ -factor. The candidate set consists of all the lines that are connected to already energized nodes, as well as the lines that are being repaired at time t .

Theorem 8. *Algorithm 5 is equivalent to Algorithm 3 discussed in Section 5.*

Proof. We first show that Algorithms 4 and 5 select the same job at each time step, as long as ties, if any, are broken identically. This does not mean, however, that their schedules are identical since Algorithm 4 is applicable to a single crew while Algorithm 5 is applicable for multiple crews. The proof is by induction on the order of lines being selected. In iteration 1, it is obvious that Algorithms 4 and 5 choose the same line for repair. Suppose this is also the case for iterations 2 to $t - 1$, with the lines chosen for repair being l_1, l_2, l_3, \dots , and l_{t-1} respectively. Then, in iteration t , the set of candidate lines for both algorithms is the set of immediate successors of the supernode $\{l_1, l_2, \dots, l_{t-1}\}$. Since $q(j) = \rho(j)$ by Theorem 7, both algorithms will choose job j in iteration t , thereby completing the induction process. Then Algorithm 5 converts the sequence of Algorithm 4 into a multi-crew schedule and Algorithm 4 is equivalent to Algorithm 2, therefore Algorithm 5 is equivalent to the conversion algorithm. \square

Finally, we would like to mention that a more general method based on parametric minimum cuts in an associated directed precedence graph is presented in [14]. This procedure also allows for online computation of the repair schedule with periodic updates of the ρ -factors.

7 Case Studies

In this section, we apply our proposed methods to three IEEE standard test feeders of different sizes. We consider the worst case, where all lines are assumed to be damaged. In each case, the importance factor w of each node is a random number between 0 and 1, with the exception of a randomly selected extremely important node with $w = 5$. The repair times are uniformly distributed on integers from 1 to 10. We compare the performances of the three methods, with computational time being of critical concern since restoration activities, in the event of a disaster, typically need to be performed in real time or near real time. All experiments were performed on a desktop with a 3.10 GHz Intel Xeon processor and 16 GB RAM. The ILP formulation was solved using Julia for Mathematical Programming with Gurobi 6.0.

7.1 IEEE 13-Node Test Feeder

The first case study is performed on the IEEE 13 Node Test Feeder shown in Fig. 1, assuming that the number of repair crews is $m = 2$. Since this distribution network is small, an optimal solution could be obtained by solving the ILP model. We ran 1000 experiments in order to compare the performances of the two heuristic algorithms w.r.t the ILP formulation.

Fig. 4 shows the density plots of optimality gaps of LP-based list scheduling algorithm (LP) and the conversion algorithm (CA), along with the better solution from the two (EN). Fig. 4a shows the optimality gaps when all repair times are integers. The density plot in this case is cut off at 0 since the ILP solves the problem optimally. Non-integer repair times can be scaled up arbitrarily close to integer values, but at the cost of reduced computational efficiency of the ILP. Therefore, in the second case, we perturbed the integer valued repair times by ± 0.1 , which represents a reasonable compromise between computational accuracy and efficiency. The optimality gaps in this case are shown in Fig. 4b. In this case, we solved the ILP using rounded off repair times, but the cost function was computed using the (sub-optimal) schedules provided by the ILP model and the actual non-integer repair times. This is why the heuristic algorithms sometimes outperform the ILP model, as is evident from Fig. 4b. In both cases, the two heuristic algorithms can solve most of the instances with an optimality gap of less than 10%. Comparing the two methods, we see that the conversion algorithm (CA) has a smaller mean optimality gap, a thinner tail, and a better worst case performance. However, this does not mean that the conversion algorithm is universally superior. In approximately 34% of the problem instances, we have found that the LP-based list scheduling algorithm yields a solution which is no worse than the one provided by the conversion algorithm.

7.2 IEEE 123-Node Test Feeder

Next, we ran our algorithms on one instance of the IEEE 123-Node Test Feeder [12] with $m = 5$. Since solving such problems to optimality using the ILP requires a prohibitively large computing time, we allocated a time budget of one hour. As shown in Table 1, both LP and HA were able to find a better solution than the ILP, at a fraction of the computing time.

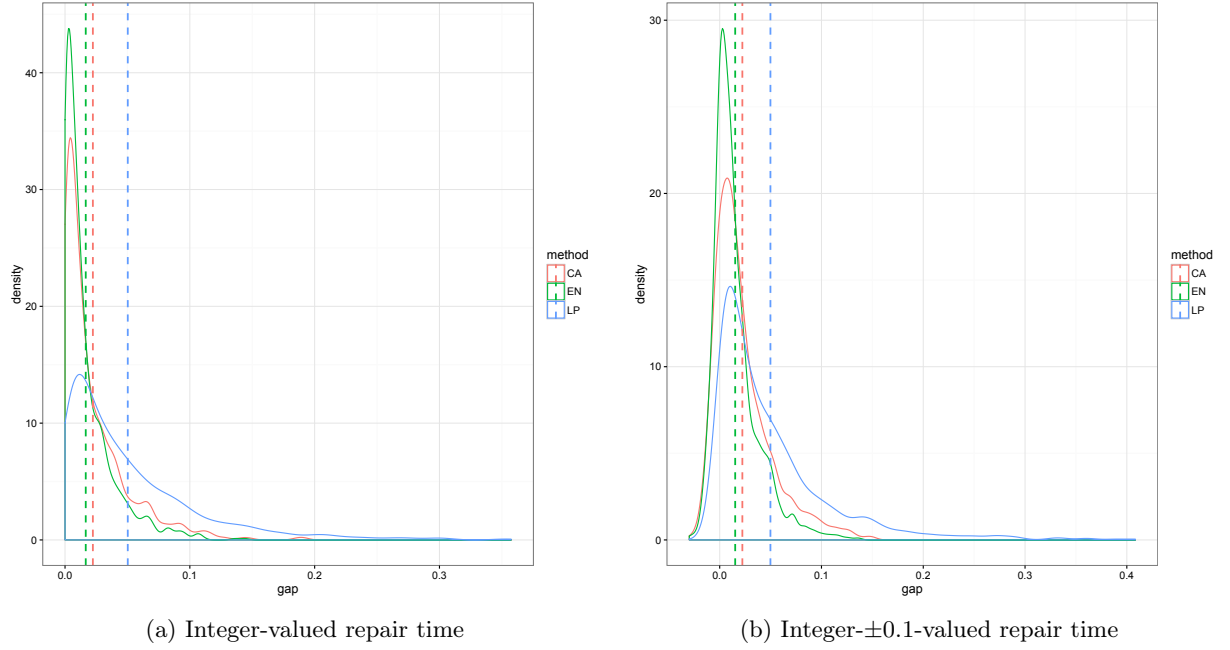


Figure 4: Density plot of optimality gap with means

| | Harm | Time(s) |
|----------------------|----------------------|---------|
| ILP | 3.0788×10^3 | 3600 |
| Conversion Algorithm | 2.2751×10^3 | <1s |
| Linear Relaxation | 2.3127×10^3 | 24s |

Table 1: Performance comparison for the IEEE 123-node test feeder

7.3 IEEE 8500-Node Test Feeder

Finally, we tested the two heuristic algorithms on one instance of the IEEE 8500-Node Test Feeder medium voltage subsystem [5] containing roughly 2500 lines, with $m = 10$. We did not attempt to solve the ILP model in this case and it took about 3 days to solve its linear relaxation (which is reasonable since we used the ellipsoid method to solve the LP with exponentially many constraints). The conversion algorithm actually solved the instance in 28s.

7.4 Discussion

From the three test cases above, we conclude that the ILP model would not be very useful for scheduling repairs and restoration in real time or near real time, except for very small problems. Even though it can be slow for large problems, the LP-based list scheduling algorithm can serve as an useful secondary tool for moderately sized problems. The conversion algorithm appears to have the best overall performance by far, in terms of solution quality and computing time.

8 Conclusion

In this paper, we investigated the problem of post-disaster repair and restoration in electricity distribution networks. We first proposed an ILP formulation which, although useful for benchmarking purposes, is feasible in practice only for small scale networks due to the immense computational time required to solve it to optimality or even near optimality. We then presented three heuristic algorithms. The first method, based on LP-relaxation of the ILP model, is proven to be a 4-approximation algorithm. The second method converts the optimal single crew schedule, solvable in polynomial time, to an arbitrary m -crew schedule with a proven performance bound of $(2 - \frac{1}{m})$. The third method, based on ρ -factors which can be interpreted as component importance measures, is shown to be equivalent to the conversion algorithm. Simulations conducted on three IEEE standard networks indicate that the conversion algorithm provides very good results and is computationally efficient, making it suitable for real time implementation. The LP-based algorithm, while not as efficient, can still be used for small and medium scale problems.

Although we have focused on electricity distribution networks, the heuristic algorithms can also be applied to any infrastructure network with a radial structure (e.g., water distribution networks). Future work includes development of efficient algorithms with proven approximation bounds which can be applied to arbitrary network topologies (e.g., meshed networks). While we have ignored transportation times between repair sites in this paper, this will be addressed in a subsequent paper. In fact, when repair jobs are relatively few and minor, but the repair sites are widely spread out geographically, optimal schedules are likely to be heavily influenced by the transportation times involved instead of the repair times. Finally, many distribution networks contain switches that are normally open. These switches can be closed to restore power to some nodes from a different source. Doing so obviously reduces the aggregate harm. We intend to address this issue in the future.

References

- [1] M. M. Adibi and L. H. Fink. Power system restoration planning. *IEEE Transactions on Power Systems*, 9(1):22–28, Feb 1994.
- [2] M. M. Adibi and L. H. Fink. Overcoming restoration challenges associated with major power system disturbances - restoration from cascading failures. *IEEE Power and Energy Magazine*, 4(5):68–77, Sept 2006.
- [3] D Adolphson and T Ch Hu. Optimal linear ordering. *SIAM Journal on Applied Mathematics*, 25(3):403–423, 1973.
- [4] The GridWise Alliance. Improving electric grid reliability and resilience: Lessons learned from Superstorm Sandy and other extreme events. Technical report, July 2013.
- [5] R. F. Artritt and R. C. Dugan. The ieee 8500-node test feeder. In *IEEE PES T D 2010*, pages 1–6, April 2010.

- [6] Peter Brucker. *Scheduling algorithms*, volume 3. Springer, 2007.
- [7] Carleton Coffrin and Pascal Van Hentenryck. Transmission system restoration: Co-optimization of repairs, load pickups, and generation dispatch. In *Power Systems Computation Conference (PSCC), 2014*, pages 1–8. IEEE, 2014.
- [8] Executive Office of the President. Economic benefits of increasing electric grid resilience to weather outages. Technical report, 2013.
- [9] WA Horn. Single-machine job sequencing with treelike precedence ordering and linear delay penalties. *SIAM Journal on Applied Mathematics*, 23(2):189–202, 1972.
- [10] Yunhe Hou, Chen-Ching Liu, Kai Sun, Pei Zhang, Shanshan Liu, and Dean Mizumura. Computation of milestones for decision support during system restoration. In *2011 IEEE Power and Energy Society General Meeting*, pages 1–10. IEEE, 2011.
- [11] Infrastructure Security and Energy Restoration, Office of Electricity Delivery and Energy Reliability, U.S. Department of Energy. A review of power outages and restoration following the June 2012 Derecho. Technical report, Aug 2012.
- [12] W. H. Kersting. Radial distribution test feeders. In *2001 IEEE Power Engineering Society Winter Meeting. Conference Proceedings (Cat. No.01CH37194)*, volume 2, pages 908–912 vol.2, 2001.
- [13] Alexis Kwasinski, John Eiding, Alex Tang, and Christophe Tundo-Bornarel. Performance of electric power systems in the 2010 - 2011 Christchurch, New Zealand, Earthquake Sequence. *Earthquake Spectra*, 30(1):205–230, 2014.
- [14] Eugene L Lawler. Sequencing jobs to minimize total weighted completion time subject to precedence constraints. *Annals of Discrete Mathematics*, 2:75–90, 1978.
- [15] T-K Ma, C-C Liu, M-S Tsai, R Rogers, SL Muchlinski, and J Dodge. Operational experience and maintenance of online expert system for customer restoration and fault testing. *IEEE Transactions on Power Systems*, 7(2):835–842, 1992.
- [16] NERC. Hurricane Sandy event analysis report. Technical report, January 2014.
- [17] Sarah G Nurre, Burak Cavdaroglu, John E Mitchell, Thomas C Sharkey, and William A Wallace. Restoring infrastructure systems: An integrated network design and scheduling (INDS) problem. *European Journal of Operational Research*, 223(3):794–806, 2012.
- [18] Raúl Pérez-Guerrero, Gerald Thomas Heydt, Nevida J Jack, Brian K Keel, and Armindo R Castellano. Optimal restoration of distribution systems using dynamic programming. *IEEE Transactions on Power Delivery*, 23(3):1589–1596, 2008.
- [19] Michael L Pinedo. *Scheduling: theory, algorithms, and systems*. Springer Science & Business Media, 2012.
- [20] Maurice Queyranne. Structure of a simple scheduling polyhedron. *Mathematical Programming*, 58(1-3):263–285, 1993.

- [21] Maurice Queyranne and Andreas S Schulz. Approximation bounds for a general class of precedence constrained parallel machine scheduling problems. *SIAM Journal on Computing*, 35(5):1241–1253, 2006.
- [22] D. A. Reed, K. C. Kapur, and R. D. Christie. Methodology for Assessing the Resilience of Networked Infrastructure. *IEEE Systems Journal*, 3(2):174–180, June 2009.
- [23] Andrei M. Reinhorn, Michel Bruneau, and Gian Paolo Cimellaro. Framework for analytical quantification of disaster resilience. *Engineering Structures*, 32(11):3639–3649, November 2010.
- [24] Andreas S Schulz et al. *Polytopes and scheduling*. PhD thesis, Citeseer, 1996.
- [25] Jeffrey B Sidney. Decomposition algorithms for single-machine sequencing with precedence relations and deferral costs. *Operations Research*, 23(2):283–298, 1975.
- [26] Wayne E Smith. Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, 3(1-2):59–66, 1956.
- [27] Wei Sun, Chen-Ching Liu, and Li Zhang. Optimal generator start-up strategy for bulk power system restoration. *IEEE Transactions on Power Systems*, 26(3):1357–1366, 2011.
- [28] The City of New York. A stronger, more resilient New York. Technical report, Jun 2013.
- [29] Sakae Toune, Hiroyuki Fudo, Takamu Genji, Yoshikazu Fukuyama, and Yosuke Nakanishi. Comparative study of modern heuristic algorithms to service restoration in distribution systems. *IEEE Transactions on Power Delivery*, 17(1):173–181, 2002.
- [30] Eric D. Vugrin, Drake E. Warren, and Mark A. Ehlen. A resilience assessment framework for infrastructure and economic systems: Quantitative and qualitative resilience analysis of petrochemical supply chains to a hurricane. *Process Safety Progress*, 30(3):280–290, September 2011.
- [31] Y. Wang, C. Chen, J. Wang, and R. Baldick. Research on resilience of power systems under natural disasters - a review. *Power Systems, IEEE Transactions on*, PP(99):1–10, 2015.
- [32] LA Wolsey. Mixed integer programming formulations for production planning and scheduling problems. Invited talk at the 12th International Symposium on Mathematical Programming. *Massachusetts Institute of Technology, Cambridge, MA*, 1985.
- [33] Ningxiong Xu, Seth D Guikema, Rachel A Davidson, Linda K Nozick, Zehra Çağnan, and Kabeh Vaziri. Optimizing scheduling of post-earthquake electric power restoration tasks. *Earthquake engineering & structural dynamics*, 36(2):265–284, 2007.
- [34] Emre Yamangil, Russell Bent, and Scott Backhaus. Designing resilient electrical distribution grids. *Proceedings of the 29th Conference on Artificial Intelligence, Austin, Texas*, 2015.