

Iloglu, Suzan; Albert, Laura A.

## Article

# An integrated network design and scheduling problem for network recovery and emergency response

Operations Research Perspectives

**Provided in Cooperation with:**

Elsevier

*Suggested Citation:* Iloglu, Suzan; Albert, Laura A. (2018) : An integrated network design and scheduling problem for network recovery and emergency response, Operations Research Perspectives, ISSN 2214-7160, Elsevier, Amsterdam, Vol. 5, pp. 218-231, <https://doi.org/10.1016/j.orp.2018.08.001>

This Version is available at:

<https://hdl.handle.net/10419/246351>

### Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

### Terms of use:

*Documents in EconStor may be saved and copied for your personal and scholarly purposes.*

*You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.*

*If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.*



<https://creativecommons.org/licenses/by-nc-nd/4.0/>



# An integrated network design and scheduling problem for network recovery and emergency response



Suzan Iloglu, Laura A. Albert\*

Department of Industrial Engineering & Systems Engineering, University of Wisconsin-Madison, 1513 University Avenue, Madison, WI 53706, United States

## ARTICLE INFO

### Keywords:

OR in disaster relief  
P-median problem  
Network restoration  
Scheduling on parallel servers  
Emergency response

## ABSTRACT

Effective recovery and restoration of infrastructure systems play a crucial role in recovery after disasters. This issue is particularly critical when delivering time-sensitive services and commodities. Damage to infrastructure can lead to disruptions and diminished capacity to respond to emergencies. We model the interdependencies between infrastructure systems and service providers as a network model, where emergency responders deliver critical services while network recovery crews repair damage to critical infrastructure. We present a novel extension to the P-median problem, where the objective is to minimize the cumulative weighted distance between the emergency responders and the calls for service over the time horizon by coordinating the activities of two types of service providers. We locate emergency responders (facilities) on a network over a finite time horizon while network recovery crews install arcs. The installation part of the models is modeled as a scheduling problem with identical parallel servers (the repair crews), where an arc can be used by the emergency responders when installation is completed. We propose Lagrangian relaxation formulations of the models, which we solve using subgradient optimization. A feasible solution is obtained using the Lagrangian relaxation, which provides an upper bound to the original models. We test our models with both real-world data and data sets from Beasley's OR Library to demonstrate the effectiveness of the algorithm in solving large-scale models. The results give insight into the optimal schedule for restoring critical arcs in a network when delivering critical services and commodities after a disruptive event.

## 1. Introduction

Emergency services play an integral role in responding to emergencies during and immediately after a disaster strikes to reduce both human suffering and property loss. At these times, delivering time-sensitive services and commodities is critically important for minimizing the risk to human life and health. For example, in 2012, Hurricane Sandy caused an estimated \$65 billion in damage to the East Coast of the United States including damaging road and transportation infrastructure. The Tappan Zee Bridge, the Bayonne Bridge, the Hugh L. Carey Tunnel, and the Queens-Midtown Tunnel were all seriously damaged by Hurricane Sandy [1]. As a result of damage to these bridges and other transportation infrastructure, the affected regions experienced increased congestion and commute times that lasted for days after Hurricane Sandy [2]. The damage resulted in severe consequences. For example, a fire that occurred during the hurricane in Breezy Point, a neighborhood in the New York City borough of Queens, destroyed 126 houses. Serious damage to the roads made it difficult for responders to reach the houses in a timely manner, allowing the fire to

spread and ultimately destroy more houses [3]. Also, Hurricane Sandy created 15 million cubic yards of debris due to the strong winds and heavy rains. The debris blocked roads, tunnels and transportation corridors in the effected areas and made the trips longer [4].

Damage to critical infrastructure can make the delivery of emergency services more difficult and increase response time during the recovery period. For example, in many disasters, damaged roads may have reduced capacity or may be impassable due to debris. Early stages of disaster recovery necessitate repairing this damage to critical infrastructure at the same time as critical services are delivered, thus aiding emergency response efforts. This paper studies how to coordinate these recovery efforts. We do so by introducing and analyzing integrated network design and scheduling models that determine how two types of service responders should work together to guide the restoration and recovery of infrastructure such as roads so that emergency services can be better delivered during the recovery period.

Few papers in the literature focus on interdependent infrastructure and network recovery to quantify the performance of the dependent infrastructures over a time horizon or planning period. However, there

\* Corresponding author.

E-mail addresses: [iloglu@wisc.edu](mailto:iloglu@wisc.edu) (S. Iloglu), [laura@enr.wisc.edu](mailto:laura@enr.wisc.edu) (L.A. Albert).

are some notable exceptions. Several papers formulate and solve network restoration problems in which infrastructure systems with critical system services are modeled as network flows [5]. Nurre et al. [6] introduce a new integrated network design and scheduling problem (INDS) that allows recovery crews to install nodes and arcs in the network to maximize the cumulative weighted flow through the network over a time horizon. This model can be used for short-term restoration and disaster preparedness activities. The installation components of our models have similar constraints to the model proposed by Nurre et al. [6].

Cavdaroglu et al. [7] propose an extension to INDS that also includes interdependencies between infrastructure systems. Instead of studying the performance of the system, the researchers focus on how quickly services are restored over the time horizon. Sharkey et al. [8] extend the interdependent layered network model of Lee et al. [5,9] to measure the performance of a system over the restoration period. They build upon the work of Cavdaroglu et al. [7] to consider damage scenarios that require restoration decisions for all infrastructure. Nurre and Sharkey [10] analyze 12 different INDS problems and show that all INDS problems are at least NP-hard. A related paper by Gutfraind et al. [11] introduces the neighbor-aided network installation problem as a discrete optimization problem to minimize the total cost of recovering a network. The authors propose a simple rule for recovering basic infrastructure networks by choosing the most accessible damaged network nodes in every iteration. Baxter et al. [12] introduce an incremental network design problem with shortest paths that focuses on network maintenance instead of network expansion in order to minimize the total cost over the planning grid. Likewise, Engel et al. [13] propose a theoretical framework to the incremental network design problem with minimum spanning trees. Duque et al. [14] introduce the Network Repair Crew Scheduling and Routing Problem (NRCSR). They optimize the accessibility of demands by scheduling and routing a single recovery crew to repair roads starting from a single depot. Averbakh and Pereira [15] introduce the Flowtime Network Construction Problem (FNCP) and the weighted version of this problem (FNCP-W). They schedule the repairing of all unavailable vertices by a single recovery crew with the objective to minimize the total recovery time of the vertices.

We study network recovery in P-median model variations. Several other studies address related issues of recovery, reliability, and vulnerability in P-median models. Wang et al. [16] consider simultaneously opening new facilities and closing existing facilities with the objective of minimizing the total weighted travel distance for customers. They develop greedy interchange, Tabu search, and Lagrangian relaxation heuristics for the model. Reliability is a related issue in P-median models, where network damage is modeled as facilities that are sometimes unavailable for service. Snyder and Daskin [17] present location models that minimize cost, including the expected transportation cost of facility failure, with the goal of choosing facility locations that are concurrently reliable and inexpensive under traditional objective functions. They present reliability models for the P-median and uncapacitated fixed charge location problems, and they propose a Lagrangian relaxation algorithm to solve the models. Cui et al. [18] propose a reliability facility location design model that extends the work of Snyder and Daskin [17] to consider site-dependent failure probabilities as opposed to identical failure probabilities for all locations. O’Hanley et al. [19] study the unreliable P-median problem by considering site-dependent failure probabilities. They introduce a technique to linearize site-dependent failure probabilities of the facilities. A limitation of these papers is that they consider independent facility failures rather than cascading failures, which limits their applicability and do not address network restoration.

Another stream of papers relevant to disasters and large-scale emergencies studies vulnerability and protection strategies in location models [20,21]. Church et al. [22] introduce the  $r$ -interdiction median (RIM) problem and the  $r$ -interdiction covering (RIC) problem to identify

the most critical facilities in the systems. These papers address the problem of identifying network protection strategies. The RIM model eliminates  $r$  facilities to maximize the cumulative weighted distance, and the RIC model maximizes the amount of demand no longer covered after  $r$  facilities are removed. Losada et al. [23] propose a bilevel mixed integer linear program to optimize resilience of the system against worst-case losses. They account for recovery time of facilities and multiple disruption probabilities over time in the model. Çelik et al. [24] consider the Stochastic Debris Clearance Problem (SDCP), which finds a sequence of roads to clear with the objective of maximizing satisfied relief demand. They model SDCP using partially observable Markov decision processes (POMDPs) that consider stochasticity in the debris (demand).

In this study, we make the following contributions:

1. We propose a new P-median problem variation that studies the interdependency between two types of service providers: network recovery crews who install arcs in the network and emergency response crews who are located at available facility locations where they deliver essential services. We schedule the installation of arcs over a finite time horizon by network recovery crews. Emergency responders can use these arcs once installation is complete to serve demand in demand points. The goal is to minimize the weighted cumulative distance between the emergency responders and the demand points over the time horizon. An extension to this model approximately models path-based arc installations between demand and facility locations. The proposed models are novel in that they coordinate the activities of two types of service providers, whose restoration activities are interdependent.
2. We introduce Lagrangian relaxation techniques to efficiently solve the models. We formulate and solve the Lagrangian relaxation dual problems using subgradient optimization, which yields a lower bound to the optimal objective function value. We propose heuristics to obtain a feasible solution to the models and an upper bound to the optimal objective function value using the Lagrangian relaxation.
3. We conduct extensive computational studies to demonstrate how these algorithms improve the time to solve the original models, and we discuss key insights gained from solving the models. The model solutions shed light on critical components of a network whose restoration can aid emergency response efforts.

The remainder of the paper is organized as follows. Section 2 provides the mathematical formulation of our models, applies the Lagrangian relaxation method to our models, and shows how to solve the relaxed models using subgradient optimization. It also provides a heuristic for finding feasible solutions to our models using the Lagrangian relaxation solution. Section 3 reports computational results and discusses the practical insights obtained from the results. The models and analysis shed light on the network components that should be prioritized during network recovery. Section 4 provides concluding remarks.

## 2. Model formulation

In this section, we introduce integrated network design and restoration models for recovering a network while providing emergency response services, and we formulate the models as integer programming models. The goal of the models is to coordinate the activities of two types of crews: emergency responders and repair crews. This is done by locating emergency responders and scheduling repair crews over a finite time horizon. At each time period, the models locate  $P$  emergency responders at open facilities in the network. The models also assign demand to open stations using available arcs in the network. This is accomplished using a multi-period P-median model, where locations are selected over time while allowing disrupted network components to

be installed by network recovery crews over a time horizon. Some arcs are initially available and can be used to assign demand to open stations at any time period, and other arcs must be installed by the repair crews over a finite time horizon and can be used upon installation. To guide the decisions made over the time horizon, the objective is to minimize the cumulative weighted distance between the emergency responders and the demands over the time horizon. We first introduce the base model, and then we introduce an extension to this model.

2.1. Integrated restoration and location problem (IRLP)

We start with a connected network  $G = (I, A)$  where  $I$  is the set of demand nodes and  $A$  is the set of initially operational arcs in the network. Let  $J \subset I$  be the set of potential facility locations and  $A'$  consist of all disrupted arcs that are not initially available in the network. Each arc  $(i, j) \in A'$  has an associated integral installation time  $p_{ij}$  and can be installed by one of the  $K$  identical recovery crews over a finite horizon of  $T$  time periods, which we model as a parallel machine scheduling sub-problem [25]. Once installation is completed, an arc can be used by the emergency responders to serve demand beginning in that time period. Each demand node  $i \in I$  has demand  $w_{it}$  in time period  $t = 1, \dots, T$ , and every arc  $(i, j) \in A \cup A'$  has distance  $d_{ij}$ . In each time period, we locate emergency responders at  $P$  of the  $|J|$  potential facilities, thus making the facility open, and we assign one of these open facilities to each demand point  $i \in I$ . The emergency responders can be relocated in each time period. The objective is to minimize the cumulative demand-weighted distance between the demand points and facility locations over the time horizon  $T$ .

The model has two parts: (1) the design part locates the emergency responders at the available facilities and (2) the recovery part schedules the installation of some of the arcs. There are two types of decision variables that correspond to these two model parts. All decision variables are binary variables.

The design decision variables are

- $x_{ijt} = 1$  if arc  $(i, j) \in A \cup A'$  is used to assign an emergency responder at facility  $j$  to demand node  $i$  at time  $t$  and 0 otherwise, for  $t = 1, \dots, T$ , and
- $y_{jt} = 1$  if an emergency responder is located at  $j \in J$  at time  $t$  and 0 otherwise, for  $t = 1, \dots, T$ .

The recovery decision variables are

- $\beta_{ijt} = 1$  if arc  $(i, j) \in A'$  is operational at time  $t$  and 0 otherwise, for  $t = 1, \dots, T$ , and
- $\alpha_{kijt} = 1$  if network recovery crew  $k$  completes the installation of arc  $(i, j) \in A'$  at time  $t$  and 0 otherwise, for  $t = 1, \dots, T$ .

We formulate our model as an integer programming model as follows.

$$Z = \min \sum_{t=1}^T \sum_{(i,j) \in A \cup A'} w_{it} d_{ij} x_{ijt} \tag{2.1}$$

$$\text{s. t. } \sum_{j:(i,j) \in A \cup A'} x_{ijt} = 1 \quad \text{for } i \in I, t = 1, \dots, T \tag{2.2}$$

$$\sum_{j \in J} y_{jt} \leq P \quad \text{for } t = 1, \dots, T \tag{2.3}$$

$$x_{ijt} - y_{jt} \leq 0 \quad \text{for } (i, j) \in A \cup A', t = 1, \dots, T \tag{2.4}$$

$$x_{ijt} \in \{0, 1\} \quad \text{for } (i, j) \in A \cup A', t = 1, \dots, T \tag{2.5}$$

$$y_{jt} \in \{0, 1\} \quad \text{for } j \in J, t = 1, \dots, T \tag{2.6}$$

$$x_{ijt} \leq \beta_{ijt} \quad \text{for } (i, j) \in A', t = 1, \dots, T \tag{2.7}$$

$$\sum_{(i,j) \in A'} \sum_{s=t}^{\min\{T, t+p_{ij}-1\}} \alpha_{kij s} \leq 1 \quad \text{for } k = 1, \dots, K, t = 1, \dots, T \tag{2.8}$$

$$\beta_{ijt} - \sum_{s=1}^t \sum_{k=1}^K \alpha_{kij s} \leq 0 \quad \text{for } (i, j) \in A', t = 1, \dots, T \tag{2.9}$$

$$\sum_{t=1}^{p_{ij}-1} \beta_{ijt} = 0 \quad \text{for } (i, j) \in A' \tag{2.10}$$

$$\sum_{k=1}^K \sum_{t=1}^{p_{ij}-1} \alpha_{kij t} = 0 \quad \text{for } (i, j) \in A' \tag{2.11}$$

$$\alpha_{kijt} \in \{0, 1\}, \beta_{ijt} \in \{0, 1\} \quad \text{for } (i, j) \in A', k = 1, \dots, K, t = 1, \dots, T \tag{2.12}$$

The objective (2.1) is to minimize the cumulative weighted distance between the demands and the closest open facilities over the time horizon. Constraint set (2.2) requires each demand node to be served by one emergency responder at each time period using the initial or installed arcs. Constraint set (2.3) locates at most  $P$  emergency responders at available facilities in the network in each time period. Constraint set (2.4) ensures that only open facilities serve each demand node. Constraint sets (2.5) and (2.6) require the design decision variables to be binary. Constraint set (2.7) ensures that the installable arcs in  $A'$  can only be used to assign demand nodes to facilities if the arcs have been installed prior to the time they are used. This set of constraints links the design unit with the recovery unit. Constraint sets (2.8)–(2.12) represent the network recovery unit of the model. Constraint set (2.8) enforces the condition that at most one arc is installed by each recovery crew in each time period, where  $\alpha_{kijt} = 1$  means recovery crew  $k$  begins installation of arc  $(i, j) \in A'$  at time  $t - p_{ij} + 1$  and completes installation at time  $t$ . This arc can then be used by the emergency responders starting from time  $t$ . Constraint set (2.9) requires arc  $(i, j)$  to be operational since  $\beta_{ijt} = 1$  can only be set to one after installation is complete. Constraint sets (2.10) and (2.11) ensure that  $\beta_{ijt}$  and  $\alpha_{kijt}$  cannot be set to one before the processing time for arc  $(i, j)$ . Constraint set (2.12) requires the recovery decision variables to be binary.

We assume that all emergency responders locations  $j \in J$  are connected to each other so that emergency responders can change their current positions at any time period with no cost by essentially moving in the network using available arcs. As discussed in Averbakh and Pereira [15], traveling speed in initially available components is faster than installation speed so it is reasonable enough to have no cost associated with changing locations. Gendreau et al. [26] consider no cost associated with relocation in their maximal expected coverage relocation problem (MECRP) formulation for emergency vehicles, however, the model contains an upper bound on the number of waiting site changes to limit relocation. This assumption is practical for our models in two ways; first, emergency responders can change locations easily and make other stations operational if they are responding to a call for service on the network, and then move to another station upon completion. Second, as we discuss later, emergency responders do not change locations often even there is no cost for doing so, and hence, this assumption is not overly restrictive. Emergency responders could be fire fighters, paramedics, and first responders, and emergency locations could be fire and rescue stations as well as temporary facilities such as schools, libraries, and other public buildings, which can be used as community health service facilities after disasters. Hence, relocating emergency responders depending on the demand and recovery process could help to reach out more people. In the case when it is not practical for emergency responders to change locations, we can simply reformulate our model as a static emergency responder location problem

by restricting the location variables  $y_{jt}$  to be identical for all time periods by using  $y_j$  for all  $j \in J$  in (2.3), (2.4), and (2.6). In addition, we can consider capacitated facilities to capture the availability of the emergency responders by adding a capacity constraint for each potential facility location  $j \in J$  at each time period as follows:

$$\sum_{i \in I} w_{it} x_{ijt} \leq Q_j y_{jt} \text{ for } j \in J, t = 1, 2, \dots, T,$$

where  $Q_j$  represents the capacity of the facility location  $j \in J$ .

IRLP has two parts. The design part consists of constraint sets (2.2) – (2.6) and the recovery part consists of constraint sets (2.8) – (2.12). These two parts are linked by constraint set (2.7). Hence, Lagrangian relaxation allows us to split the model into two parts. In the next section, we discuss how to apply Lagrangian relaxation to this set of constraints.

### 2.1.1. Lagrangian relaxation

Lagrangian relaxation is one of the most commonly used methods for hard problems in combinatorial optimization. By dualizing the constraints that make the problem “hard”, the Lagrangian problem becomes an easier problem [27]. In IRLP, constraint (2.7) links the design decision variables with the restoration decision variables. Applying Lagrangian relaxation to this set of constraints is advantageous in several ways. First, doing so decomposes it into two parts (the design part and the recovery part), which are included in the objective with a penalty term. Therefore, the relaxation is easier to solve than the original model. Second, the Lagrangian relaxation problem solution offers a lower bound on the optimal solution to the original problem. Third, a feasible solution to the original model can be constructed from the Lagrangian relaxation problem solution, which provides an upper bound to the original problem. We use this upper bound as a cut in the branch and bound tree when solving the original model, thereby reducing the computation time needed to find an optimal solution.

Now, we present the Lagrangian relaxation problem formulation on constraint set (2.7). Let  $u_{ijt}$  be Lagrangian multiplier for  $(i, j) \in A'$ ,  $t = 1, \dots, T$ , then we define Lagrangian relaxation problem as following:

$$L(u) = \min \sum_{t=1}^T \sum_{(i,j) \in A \cup A'} w_{it} d_{ij} x_{ijt} + \sum_{t=1}^T \sum_{(i,j) \in A'} u_{ijt} [x_{ijt} - \beta_{ijt}]$$

or we can rewrite  $L(u)$  as

$$L(u) = \min \sum_{t=1}^T \sum_{(i,j) \in A \cup A'} (w_{it} d_{ij} + u_{ijt}) x_{ijt} - \sum_{t=1}^T \sum_{(i,j) \in A'} u_{ijt} \beta_{ijt}$$

s. t. (2.2) – (2.6) and (2.8) – (2.12)

The Lagrangian relaxation dual is

$$\max_{u \geq 0} L(u)$$

The Lagrangian relaxation dual remains NP-hard since the P-median problem is embedded in it. However, in practice we can quickly solve the Lagrangian relaxation dual problem using subgradient algorithm. We discuss this next.

### 2.1.2. Subgradient algorithm

Subgradient optimization is used to solve the Lagrangian relaxation problem and identify a lower bound to the original model by iteratively adjusting the Lagrangian multipliers  $u^n$  in iteration  $n$ . Starting with the given inputs, initial Lagrangian multiplier  $u^0$ , initial upper bound  $L^*$ , and initial decreasing adaption parameter  $\theta^0$ , each iteration of the subgradient algorithm updates the previous Lagrangian multiplier with the step size in the positive subgradient direction. The algorithm checks if the new bound is better than the previous one. We start with  $\theta^0 = 2$ , as suggested in [28]. If there is no improvement in the bound after more than  $N^*$  iterations, we reduce  $\theta$  by a factor of 1/2. If the change in the

Lagrangian multiplier is not smaller than  $\epsilon$  after  $N$  iterations, the algorithm terminates and returns  $L_{best}$  which is the best lower bound found by the subgradient algorithm. We note that the choices for  $N$  and  $N^*$  values are problem specific. We use  $N = 100$  and  $N^* = 2$ , and our solution procedure is not sensitive to these values. We first solve the linear programming relaxation of IRLP, and we use the “shadow prices” for the dual variables associated with constraint set (2.7) as the initial value for Lagrangian multiplier  $u_{ijt}^0$ . Using this initial value of  $u^0$ , we optimize the Lagrangian relaxation dual problem using the subgradient algorithm as in [29].

### 2.1.3. Lagrangian heuristic

We now present a heuristic that constructs a feasible solution for IRLP using the Lagrangian relaxation problem solution. Let  $\bar{\beta}$ ,  $\bar{y}$  and  $\bar{\alpha}$  denote a feasible solution for the Lagrangian relaxation problem. Our aim is to find  $\bar{x}$  values that are feasible for constraint set (2.7). Let  $S_1$  denote the triplet  $(i, j, t)$ , where the arc  $(i, j) \in A'$  is operational at time  $t$  and the location  $j$  is open at time  $t$ , defined as  $S_1 = \{(i, j, t) \mid \bar{\beta}_{ijt} = 1 \text{ and } \bar{y}_{jt} = 1, (i, j) \in A'\}$ . Let  $S_2$  denote the triplet  $(i, j, t)$ , where the facility  $j$  is open for arc  $(i, j) \in A$ , which are initially in the network as  $S_2 = \{(i, j, t) \mid \bar{y}_{jt} = 1, (i, j) \in A\}$ .

To construct feasible solutions for each  $t = 1, \dots, T$  and for each  $i \in I$ , we set  $\bar{x}_{ij^*t} = 1$  for  $j^*$  that solves

$$j^* \in \arg \min_{j \in J} \left\{ \min_{(i,j,t) \in S_1} w_{it} d_{ij}, \min_{(i,j,t) \in S_2} w_{it} d_{ij} \right\}$$

and set  $\bar{x}_{ijt} = 0$  for all other  $(i, j) \in A \cup A'$  and  $t = 1, \dots, T$ . This assigns the demand node at  $i$  to the closest open facility  $j$ . The solution  $\bar{\beta}$ ,  $\bar{x}$ ,  $\bar{y}$  and  $\bar{\alpha}$  is feasible since the  $\bar{\beta}$ ,  $\bar{y}$  and  $\bar{\alpha}$  values obtained from the Lagrangian relaxation solution satisfy all constraints except the Lagrangian relaxation constraint set (2.7). With this assignment of  $\bar{x}$ , constraint set (2.7) is satisfied. Finding this feasible solution from the Lagrangian relaxation for IRLP gives us an upper bound for the IRLP. This upper bound is used as a cut value in the branch and bound tree of the original problem to reduce the tree size and improve computational time.

IRLP allows us to shorten direct arcs between demand points and emergency responder locations by installing arcs without considering other possible disrupted paths between same demand points and emergency responder locations. To capture this, we include multiple parallel versions of each disrupted arc in the network to allow installation of an arc to be completed in multiple ways. In the next section, we introduce a component based integrated restoration and location problem that allows us to include new installation structures in the model to approximately include these additional features in the model.

## 2.2. Component based IRLP (c-IRLP)

In this section, we present the component based IRLP (c-IRLP), an extension of IRLP that considers multiple parallel versions of any installable arc to capture the component-wise benefits of installed arcs. Instead of a single arc connecting  $i$  to  $j$  in  $A'$  as in IRLP, we introduce several *components* between  $i$  to  $j$ , each of which corresponds to a different road on a path that can be repaired. In the c-IRLP, if we use location  $j$  to serve demand at  $i$  in a network, then we could possibly do a quick repair of a component to decrease its distance and then repair another component later to decrease its distance even more by repairing another component in the path. Therefore, we allow for multiple parallel versions of the arc  $(i, j)$  for each of its disrupted components, each with a different cost and processing time. In addition, the c-IRLP allows a single repair of a component in the network to benefit multiple arcs in  $A'$  if multiple arcs share a common component such as a road.

The c-IRLP starts with the same connected network  $G = (I, A)$  as in

IRLP. Different from IRLP, we define a new set  $C'$  as an installable component set and we install components instead of arcs. Each component is associated with one or more arcs that all share the component. To capture all arcs that benefit from installation of a component, we define set  $AC(c)$  for each  $c \in C'$  that consists of arcs  $(i, j) \in A'$  whose distances are decreased after installing component  $c \in C'$ . Note that  $A' = \cup_{c \in C'} AC(c)$  is the set of arcs installed through components.

Each component  $c \in C'$  has an associated integral installation time  $p_c$  and can be installed by one of the  $K$  identical recovery crews over a finite horizon of  $T$  time periods. Once installation of a component  $c$  is complete, arcs in  $AC(c)$  associated with component  $c$  can be used by emergency responders to serve demand at demand points.

As in IRLP, each demand node  $i \in I$  has demand  $w_{it}$  in time period  $t = 1, \dots, T$  and in each time period, we locate emergency responders at  $P$  of the  $|J|$  potential facilities. In this model we have multiple parallel versions of the arc each with different cost, therefore, we consider the distance between demand  $i$  and emergency responder location  $j$  using component  $c$  as  $d_{cij}$ . We also define  $d_{0ij}$  as the initial distance between  $i$  and  $j$  before any installation for arcs  $(i, j) \in A$ . Similar to IRLP, the emergency responders can be relocated in each time period. The objective is to minimize cumulative demand-weighted distance between the demand points and facility locations over the time horizon  $T$ .

The model has two parts as in IRLP, and all decision variables are binary variables. In this model, we redefine the decision variables  $x, \beta$ , and  $\alpha$ . The variable  $y$  is the same as in IRPL.

The design decision variables are

- $\tilde{x}_{cijt} = 1$  if arc  $(i, j) \in A \cup A'$  is used to assign an emergency responder at facility  $j$  to demand node  $i$  at time  $t$  by using installed component  $c$  and 0 otherwise, for  $t = 1, \dots, T$ ,
- $\tilde{x}_{0ijt} = 1$  if arc  $(i, j) \in A$  is used to assign an emergency responder at facility  $j$  at time  $t$  by using initially available arc  $(i, j) \in A$  and 0 otherwise, for  $t = 1, \dots, T$  and
- $y_{jt} = 1$  if an emergency responder is located at  $j \in J$  at time  $t$  and 0 otherwise, for  $t = 1, \dots, T$ .

The recovery decision variables are

- $\tilde{\beta}_{ct} = 1$  if component  $c \in C'$  is operational at time  $t$  and 0 otherwise, for  $t = 1, \dots, T$ , and
- $\tilde{\alpha}_{kct} = 1$  if network recovery crew  $k$  completes the installation of component  $c \in C'$  at time  $t$  and 0 otherwise, for  $t = 1, \dots, T$ .

We formulate our model as an integer programming model as follows.

$$Z = \min \sum_{t=1}^T \sum_{c \in C'} \sum_{(i,j) \in AC(c)} w_{it} d_{cij} \tilde{x}_{cijt} + \sum_{t=1}^T \sum_{(i,j) \in A} w_{it} d_{0ij} \tilde{x}_{0ijt} \quad (2.13)$$

$$\text{s. t. } \sum_{c \in C'} \sum_{j: (i,j) \in AC(c)} \tilde{x}_{cijt} + \sum_{j: (i,j) \in A} \tilde{x}_{0ijt} = 1 \quad \text{for } i \in I, t = 1, \dots, T \quad (2.14)$$

$$\sum_{j \in J} y_{jt} \leq P \quad \text{for } t = 1, \dots, T \quad (2.15)$$

$$\tilde{x}_{cijt} - y_{jt} \leq 0 \quad \text{for } c \in C', (i, j) \in AC(c), t = 1, \dots, T \quad (2.16)$$

$$\tilde{x}_{0ijt} - y_{jt} \leq 0 \quad \text{for } (i, j) \in A, t = 1, \dots, T \quad (2.17)$$

$$\tilde{x}_{cijt} \in \{0, 1\} \quad \text{for } c \in C', (i, j) \in AC(c), t = 1, \dots, T \quad (2.18)$$

$$\tilde{x}_{0ijt} \in \{0, 1\} \quad \text{for } (i, j) \in A, t = 1, \dots, T \quad (2.19)$$

$$y_{jt} \in \{0, 1\} \quad \text{for } j \in J, t = 1, \dots, T \quad (2.20)$$

$$\tilde{x}_{cijt} \leq \tilde{\beta}_{ct} \quad \text{for } c \in C', (i, j) \in AC(c), t = 1, \dots, T \quad (2.21)$$

$$\sum_{c \in C'} \sum_{s=t}^{\min\{T, t+p_c-1\}} \tilde{\alpha}_{kcs} \leq 1 \quad \text{for } k = 1, \dots, K, t = 1, \dots, T \quad (2.22)$$

$$\tilde{\beta}_{ct} - \sum_{s=1}^t \sum_{k=1}^K \tilde{\alpha}_{kcs} \leq 0 \quad \text{for } c \in C', t = 1, \dots, T \quad (2.23)$$

$$\sum_{t=1}^{p_c-1} \tilde{\beta}_{ct} = 0 \quad \text{for } c \in C' \quad (2.24)$$

$$\sum_{k=1}^K \sum_{t=1}^{p_c-1} \tilde{\alpha}_{kct} = 0 \quad \text{for } c \in C' \quad (2.25)$$

$$\tilde{\alpha}_{kct} \in \{0, 1\}, \tilde{\beta}_{ct} \in \{0, 1\} \quad \text{for } c \in C', k = 1, \dots, K, t = 1, \dots, T \quad (2.26)$$

Constraint sets (2.13)–(2.20) and (2.22)–(2.26) are analogous to constraint sets (2.1)–(2.6) and (2.8)–(2.12) in IRLP, respectively. The objective (2.13) is to minimize the cumulative weighted distance between the demands and the closest open facilities using available arcs after component installations or initial arcs in the network over the time horizon. Constraint set (2.14) requires each demand node to be served by one emergency responder at each time period using the initial arcs or arcs that become available after installation of components. The most important change is in (2.21), which ensures that component  $c$  in  $C'$  can only be used to assign demand node  $i$  to facility  $j$  if the component  $c$  has been installed prior to the time period and the installation of component  $c$  shortens the distance between  $i$  and  $j$  (i.e.,  $(i, j) \in AC(c)$ ). Constraint set (2.21) links the design part with the recovery part as in IRLP.

As shown in Appendix A, unit processing times that are identical for all components in  $C'$  (i.e., without loss of generality  $p_c = 1$  for all  $c \in C'$ ) allow us to remove many variables and constraints, which drastically simplifies the integer programming formulation. In this simplified version of the model, we select  $K$  components to install at each time period instead of assigning component installation jobs to individual recovery crews. This simplification results in decreased computational times, even for experiments with long time horizons and large recovery crews.

As in IRLP, the Lagrangian relaxation allows us to split the c-IRLP model into two parts by relaxing constraint set (2.21). In the next section, we discuss how we need to adjust the Lagrangian relaxation algorithm we present in Section 2.1.1 to solve the c-IRLP.

### 2.2.1. Lagrangian relaxation and subgradient algorithm

Now, we present the Lagrangian relaxation problem formulation on constraint set (2.21). Let  $u_{cijt}$  be Lagrangian multiplier for  $c \in C', (i, j) \in AC(c), t = 1, \dots, T$ , analogous to Lagrangian relaxation in Section 2.1.1. Therefore, we briefly highlight new differences.

After simplifying the relaxed formulation, we can write the Lagrangian relaxation problem  $L(u)$  as

$$L(u) = \min \sum_{t=1}^T \sum_{c \in C'} \sum_{(i,j) \in AC(c)} (w_{it} d_{cij} + u_{cijt}) \tilde{x}_{cijt} + \sum_{t=1}^T \sum_{(i,j) \in A} w_{it} d_{0ij} \tilde{x}_{0ijt} - \sum_{t=1}^T \sum_{c \in C'} \sum_{(i,j) \in AC(c)} u_{cijt} \tilde{\beta}_{ct} \quad \text{s. t. } (2.14) - (2.20) \text{ and } (2.22) - (2.26)$$

The Lagrangian relaxation dual is

$$\max_{u \geq 0} L(u)$$

We solve the Lagrangian relaxation dual problem using a subgradient algorithm as for IRLP. Algorithm 1 can be applied to the c-IRLP with two minor modifications. Line 1 in the Algorithm 1 must change to

```

1: Initialize:
   An initial value  $u^0 \geq 0$  (assign as the shadow price associated with the Lagrangian relaxation constraint in the linear programming relaxation)
    $\theta^0 := 2$ 
    $L_{best} = -\infty$ 
    $L^* = \sum_{t=1}^T \sum_{(i,j) \in A} w_{it} d_{ij}$ 
2: Subgradient iterations
3: for  $n := 0, \dots, N$  do
4:   Solve for  $L(u^n)$  which yields  $x$  and  $\beta$ 
5:    $\gamma_{ijt}^n := x_{ijt}^n - \beta_{ijt}^n$ 
6:    $t^n := \theta^n (L^* - L(u^n)) / \|\gamma^n\|^2$  step size
7:    $u^{n+1} := \max\{0, u^n + t^n \gamma^n\}$ 
8:   if  $\|u^{n+1} - u^n\| < \epsilon$  then
9:     Stop
10:  end if
11:   $L_{best} = \max(L_{best}, L(u^n))$ 
12:  if  $L_{best}$  does not improve after  $N^*$  iterations then
13:     $\theta^{n+1} := \theta^n / 2$ 
14:  else
15:     $\theta^{n+1} := \theta^n$ 
16:  end if
17:   $n := n + 1$ 
18: end for
19: return  $L_{best}$ 

```

**Algorithm 1.** Subgradient optimization algorithm for IRLP.

$L^* = \sum_{t=1}^T \sum_{(i,j) \in A} w_{it} d_{0ij}$  and line 5 must change to  $\gamma_{cijt}^n := \bar{x}_{cijt}^n - \bar{\beta}_{cjt}^n$ . The remaining parts of the subgradient algorithm in Section 2.1.2 remain the same.

### 2.2.2. Lagrangian and linear relaxation heuristic

We now present a heuristic that constructs a feasible solution for the c-IRLP. We do so by obtaining a feasible solution to the Lagrangian relaxation of the c-IRLP, and enhancing the feasible solution using the linear relaxation solution of the c-IRLP.

Danna et al. [30] uses an approach to obtain a feasible solution to a mixed integer programming model better than the current incumbent solution by enhancing the solution using a linear relaxation solution. We use a similar approach. We can obtain an incumbent solution using the Lagrangian relaxation, however, it does not satisfy the constraint set (2.21). Contrarily, the linear relaxation solution satisfies all of the constraint sets except the integrality constraints. We combine these two solutions to find a feasible solution that satisfies all of the constraint sets by first identifying the variables that take the same values in the Lagrangian and linear relaxation solutions, and fixing these variables, thus forming a partial solution. Let  $\bar{x}, \bar{y}$  denote values of the design decision variables for the Lagrangian relaxation problem solution, and let  $\bar{\bar{x}}, \bar{\bar{y}}$  denote values of the design decision variables for the linear relaxation problem solution. Then we define sets  $S_1 = \{(c, i, j, t): \bar{x}_{cijt} = \bar{\bar{x}}_{cijt}\}$  and  $S_2 = \{(j, t): \bar{y}_j = \bar{\bar{y}}_j\}$ , which denote the values of  $\bar{x}$  and  $\bar{y}$  variables that have the same values in the Lagrangian and linear relaxation solutions, respectively. Note that the values of  $\bar{x}$  and  $\bar{y}$  are integer in the set  $S_1$  and  $S_2$ . We fix the values of the  $\bar{x}$  and  $\bar{y}$  variables in the c-IRLP for the indexes in the sets  $S_1$  and  $S_2$ , respectively. Then, we solve the c-IRLP that yields a solution that is feasible for the constraint sets (2.13)–(2.26) and is an upper bound for the optimal objective function. Even though we solve an NP-complete problem, this method can be computationally efficient since many of the variables are fixed in this procedure. For example, in our experiments in the following section, we fix approximately 80 to 90 percent of  $\bar{x}$  variables and 5 to 10 percent of  $\bar{y}$  variables that have the same values in the Lagrangian and linear relaxation

solutions. In addition, we use sets  $S_1$  and  $S_2$  to set starting values for the variables  $x$  and  $y$ , respectively, in the standard implementation. Using the partial solution as a starting point helps the solver obtain an incumbent solution faster than without setting the starting point. Hence, this approach can improve the computation time compare to the computation time of the model using standard implementation.

## 3. Computational results

In this section, we present and analyze computational results using both real-world data representing road infrastructure and emergency medical calls for Hanover County, Virginia, United States and data sets from Beasley's OR Library [31].

### 3.1. Computational results for IRLP

Computations for IRLP were performed on a computer with a 2.6 GHz Intel Core 5 Duo Processor with 8GB of RAM. We used CPLEX 12.6.2.0 to solve the integer programming model that was coded in GAMS. Each experiment was run with a 3600-s time limit.

The Hanover County data has  $|I| = 125$  demand nodes and  $|J| = 16$  fire and rescue stations (available facilities). We assume that our available facilities and demand nodes are located in the center of each grid point. Since disasters are rare, the emergency calls for service observed after a single disaster may be insufficient to represent the anticipated demand for all demand points. Therefore, we use data aggregated over a 31 month period to estimate post-disaster demand, and we assume that the demand that would occur in a short period of time after a disaster is proportional to the aggregated demand data over a long period. This assumption appears to be reasonable based on the data we collected after Hurricane Irene in 2011. We use real distances,  $d_{ij}$ , in miles, from facility location  $j$  to call location  $i$  in the disrupted arc set  $|A'| = 325$ , where  $d_{ij} \leq 6$  and in the initial connected network arc set  $A$ , where  $d_{ij} > 6$ . In addition, lengthened versions of the arcs in  $A'$  are contained in the initial connected network arc set  $A$ , where  $|A| = 2000$ .

Processing times,  $p_{ij}$ , are integers randomly generated from a discrete  $U[1, 3]$  distribution. We allow  $P = 8$  available facilities to be open in each time period. Since our model considers short-term repair of the road network components, we consider the time horizon as a day and vary the number of time periods  $T$  and the number of network recovery crews  $K$ .

We also test the model and algorithm with two data sets, each consisting of 100 demand nodes from Beasley’s OR Library uncapacitated P-median data sets (Pmed1 and Pmed3). The data sets provide a graph, which we use as the initial graph  $G = (I, A)$ , the number of open facilities  $P$ , and arc distances associated with each arc. In both data sets, we set  $J = I$  and  $w_{it}$  are randomly generated from a continuous uniform  $U(0, 1)$  distribution for each demand node  $i \in I$ . We assume  $w_{it}$  stays the same at each time period  $t = 1, \dots, T$  for each demand node  $i \in I$ . The installable arc set  $A'$  contains the three shortest arcs for each demand point. The distances of the arcs in  $A'$  are equal to the values in the original data set from Beasley’s OR Library. Lengthened versions of these arcs are contained in the set of initially available arcs  $A$ . The distances of the lengthened arcs in  $A$  are increased by half of their original distances from Beasley’s data for each arc that also appears in  $A'$ . The distances of the remaining arcs in  $A$  are not disrupted, and therefore they are equal to the original distances given in Beasley data. Processing times  $p_{ij}$  for each arc  $(i, j) \in A'$  are randomly generated from a discrete uniform  $U[1, 2]$  distribution. We consider each time horizon as a day and vary the time horizon  $T$  and the number of network recovery crews  $K$  in the computational experiments as in the Hanover County data experiments.

Tables 1 and 2 summarize the objective values and running times for the Hanover County data set and for the data sets from Beasley’s OR Library, respectively. The  $|A|$ ,  $|A'|$  and  $|J|$  columns report the number of arcs initially in the network, number of installable arcs, and the number of facility locations, while the columns labeled  $K$  and  $T$  report the number of recovery crews and the length of the time horizon. The “Lagrangian Relaxation (LR) Lower Bound” column reports the lower bound values obtained from the Lagrangian relaxation and the running time in seconds in parentheses. The LR lower bounds are better than the lower bounds obtained from the linear programming relaxations for all experiments in Tables 1 and 2. The Lagrangian heuristic identifies upper bounds for the optimal values, which are shown in the “Lagrangian Relaxation (LR) Upper Bound” column. The column “Optimal

Solution Using LR Upper Bound” reports the objective function value and the cumulative computation time in parentheses. These solutions are found using the Lagrangian heuristic upper bound as a cut value in the branch and bound tree of the original model. With this improvement, all instances in Table 1 were solved to optimality within the 3600-s limit, and the optimality gap of almost all instances decreased within the 3600-s limit in Table 2. The last column entitled “Optimal Solution Without Using Upper Bound” reports the optimal objective function value and the computation time in parentheses when using CPLEX without the Lagrangian heuristic upper bound. The computation time in the column “Optimal Solution Using LR Upper Bound” includes the CPLEX time to solve the linear relaxation, to obtain the upper bound using Lagrangian relaxation, and to solve the model. The computation time in the column “Optimal Solution Without Using Upper Bound” only includes the time to calculate the optimal solution for the original model.

The choice of step size in the Lagrangian relaxation is important, since the convergence of the algorithm depends on the step size. We observe that the Polyak step size formula  $t_k = \frac{\theta^k(L^* - L(u^k))}{\|y^k\|^2}$  [32] results in a fast convergence in the subgradient algorithm for computational experiments for the Hanover County and Beasley’s OR Library’s data sets.

In Table 1, ten of the fifteen experiments solve to optimality within the 3600-s limit without using the upper bound, while in Table 2, nine of the eighteen experiments solve optimally within the 3600-s limit without using the upper bound. As we can observe in the Tables 1 and 2, CPLEX requires more time to find an optimal solution over the longer time horizons, which is not surprising, since the number of decision variables and the number of constraints increase with the length of the time horizon  $T$ . Total times using different methods are comparable. Using the upper bound as a cut value in the branch and bound tree improves the computation times for thirteen of the fifteen and eight of the eighteen experiments (shown in boldface) in the two rightmost columns in Tables 1 and 2, respectively. In addition, using the upper bound as a cut value in the branch and bound tree decreases the optimality gap for six out of eighteen experiments (shown in boldface) in Table 2. Also, in the experiment with  $T = 10$  and  $K = 3$  for the Pmed3 data set, the resulting best integer feasible solution using the upper bound as a cut value in the branch and bound tree is better than the best integer feasible solution without using upper bound.

Table 1

Improvements in the optimal solutions for IRLP using Lagrangian upper bounds for the Hanover County data. Computational time, in seconds, is shown in the parentheses. “Optimal Solution Using LR Upper Bound” reports the objective function value and the computation time using CPLEX Lagrangian heuristic upper bound in the branch and bound tree and “Optimal Solution Without Using Upper Bound” reports the optimal objective function value and the computation time when using CPLEX without the Lagrangian heuristic upper bound.

$ A $	$ A' $	$ I $	$ J $	$P$	$K$	$T$	Linear Relaxation Lower Bound	Lagrangian Relaxation(LR) Lower Bound	Lagrangian Relaxation(LR) Upper Bound	Optimal Solution Upper Bound Using LR	Optimal Solution Upper Bound Without Using LR
2000	325	125	16	8	3	10	39.42 (4)	42.43 (9)	49.48 (14)	45.19 (25)	45.19 (16)
2000	325	125	16	8	4	10	38.75 (3)	41.81 (12)	49.06 (16)	44.44 (49)	44.44 (35)
2000	325	125	16	8	5	10	38.13 (4)	41.32 (14)	49.11 (17)	43.79 (57)	43.79 (83)
2000	325	125	16	8	3	15	57.59 (5)	62.17 (30)	73.69 (29)	66.07 (224)	66.07 (674)
2000	325	125	16	8	4	15	56.33 (11)	61.35 (48)	79.58 (56)	64.76 (198)	64.76 (967)
2000	325	125	16	8	5	15	55.25 (16)	60.15 (31)	72.42 (39)	63.66 (568)	63.66 (1006)
2000	325	125	16	8	3	20	75.06 (17)	81.49 (33)	97.16 (41)	86.30 (272)	86.30 (1946)
2000	325	125	16	8	4	20	73.17 (24)	79.90 (46)	97.00 (55)	84.38 (466)	84.38 (1946)
2000	325	125	16	8	5	20	71.57 (39)	78.71 (57)	96.71 (68)	82.81 (548)	82.81 (793)
2000	325	125	16	8	3	24	88.63 (30)	96.36 (56)	117.11 (67)	102.12 (552)	102.12 (1965)
2000	325	125	16	8	4	24	86.21 (50)	94.24 (78)	117.61 (88)	99.70 (372)	99.75 (3600) (Gap 0.06%)
2000	325	125	16	8	5	24	84.19 (77)	92.84 (105)	115.00 (120)	97.85 (464)	97.95 (3600) (Gap 0.17%)
2000	325	125	16	8	3	30	108.43 (67)	118.46 (95)	143.62 (107)	125.27 (2001)	125.99 (3600) (Gap 0.6%)
2000	325	125	16	8	4	30	105.19 (152)	115.86 (194)	143.88 (209)	122.27 (865)	124.00 (3600) (Gap 1.58%)
2000	325	125	16	8	5	30	102.50 (230)	113.90 (261)	144.05 (289)	119.99 (1228)	120.00 (3600) (Gap 0.02%)



**Table 2**

Improvements in the optimal solutions for IRLP using Lagrangian upper bounds for Beasley’s data. Computational time, in seconds, is shown in the parentheses. “Optimal Solution Using LR Upper Bound” reports the objective function value and the computation time using CPLEX Lagrangian heuristic upper bound in the branch and bound tree and “Optimal Solution Without Using Upper Bound” reports the optimal objective function value and the computation time when using CPLEX without the Lagrangian heuristic upper bound.

	$ A $	$ A' $	$ I $	$ J $	$P$	$K$	$T$	Linear Relaxation Lower Bound	Lagrangian Relaxation Lower Bound	Lagrangian Relaxation Upper Bound	Optimal Solution Using LR Upper Bound	Optimal Solution Using LR Upper Bound	
Pmed1	10,000	300	100	100	5	3	10	28,434.87 (157)	29,500.21 (209)	31,960.94 (213)	30,244.43 (327)	30,244.43 (1195)	
							4	27,890.37 (173)	29,259.65 (201)	32,060.34 (205)	29,979.35 (769)	29,979.35 (887)	
							5	27,484.74 (171)	29,183.47 (231)	31,660.91 (236)	29,793.31 (785)	29,793.31 (562)	
							3	15	41,577.88 (421)	43,805.66 (605)	47,719.55 (611)	44,853.09 (1086)	44,853.09 (3171)
							4	15	40,843.32 (301)	43,682.06 (346)	47,693.86 (353)	44,509.49 (1237)	44,509.49 (2040)
							5	15	40,384.06 (245)	43,651.92 (348)	48,254.89 (514)	44,296.61 (1244)	44,296.61 (1332)
							3	20	54,504.287 (591)	58,289.95 (737)	63,433.44 (624)	59,403.17 (3600)	59,443.37 (3600)
												(Gap 0.05%)	(Gap 0.18%)
							4	20	53,733.75 (477)	58,205.75 (606)	64,102.98 (612)	59,011.63 (2636)	59,024.50 (3600)
												(Gap 0.07%)	(Gap 0.07%)
Pmed3	10,000	300	100	100	10	3	10	19,870.12 (71)	20,772.03 (89)	22,988.81 (91)	21,514.02 (1260)	21,514.02 (1381)	
							4	10	19,356.51 (73)	20,519.53 (94)	23,155.06 (98)	21,325.4 (3420)	21,325.4 (2878)
							5	10	18,971.67 (85)	20,397.49 (97)	23,147.13 (100)	21,199.18 (3600)	21,206.93 (3600)
												(Gap 0.48%)	(Gap 0.45%)
							3	15	28,777.47 (359)	30,686.42 (580)	34,663.63 (586)	31,947.38 (3600)	32,155.81 (3600)
												(Gap 0.05%)	(Gap 1.19%)
							4	15	28,065.82 (344)	30,531.11 (396)	34,770.12 (402)	31,948.37 (3600)	31,779.40 (3600)
												(Gap 1.49%)	(Gap 0.97%)
							5	15	27,603.22 (254)	30,488.25 (299)	34,774.21 (305)	31,308.17 (3600)	31,309.97 (3600)
												(Gap 0.12%)	(Gap 0.14%)
3	20	37,456.98 (475)	40,710.83 (588)	46,627.35 (595)	43,303.03 (3600)	47,968.59 (3600)							
					(Gap 3.09%)	(Gap 12.5%)							
4	20	36,679.15 (338)	40,623.7 (466)	46,216.43 (472)	42,070.64 (3600)	42,081.40 (3600)							
					(Gap 1.15%)	(Gap 1.15%)							
5	20	36,214.41 (254)	40,617.78 (388)	46,249.88 (394)	41,438.53 (3600)	41,470.72 (3600)							
					(Gap 0.06%)	(Gap 0.29%)							

Tables 1 and 2 report algorithm running times and objective function values. We can further examine the solutions produced by the models. To do so we represent the objective function value as the sum of the objective function value recorded in each time period, which captures the minimum cumulative weighted distance

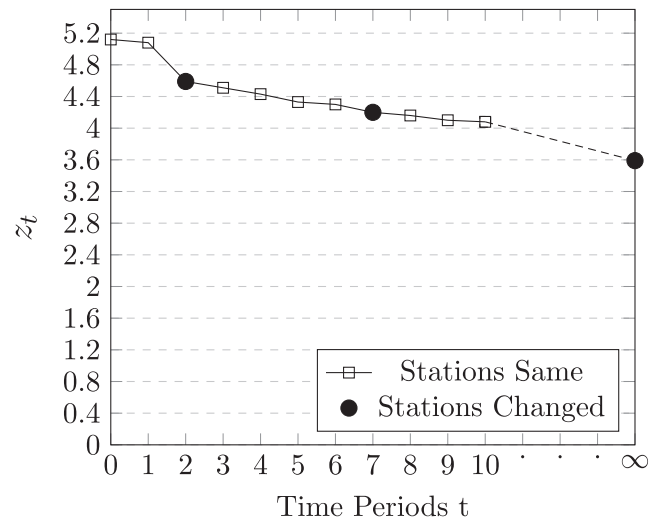
$$z_t = \sum_{(i,j) \in A \cup A'} w_{ij} d_{ij} x_{ijt}, \text{ for } t = 1, \dots, T$$

$$\text{and } Z = \sum_{t=1}^T z_t.$$

We compare our results to two static comparison cases that consider a single time period time  $T = 0$  and  $T = \infty$ . To do so we solve the canonical P-median problem with network  $G = (I, A)$  for  $T = 0$  and with  $G = (I, A \cup A')$  for  $T = \infty$ , which reflect the network when no arcs and all arcs are installed, respectively. We report the objective function values, in Fig. 1 and the locations of the open facilities in Table 3. This allows us to compare emergency responder locations in the model solutions to those prior to recovery efforts in damaged networks ( $T = 0$ ) and in networks that are fully functional ( $T = \infty$ ).

Fig. 1 shows the values of  $z_t$  over the time horizon for the experiment  $K = 5$  and  $T = 10$  using the Hanover County data set. The objective function value  $z_t$  starts with highest value at  $t = 0$  (refers to a single time period  $T = 0$ ) since no arcs are installed yet, and it decreases over the time horizon as the network recovery crews finish installing arcs from  $A'$ , which are then used by the emergency responders to decrease the average distance. The best possible single period objective function value occurs at  $t = \infty$  when all arcs are installed.

Table 3 shows locations of the open facilities over the time horizon for the instance with  $K = 5$  and  $T = 10$ . The locations of open facilities change two times during this experiment, which are also indicated in Fig. 1. In time period 2, Facility 1 closes and Facility 15 opens, and in time period 7, Facility 8 closes and Facility 9 opens. By allowing the model to change the location of some of the emergency responders, the average distance between nodes and emergency responders is further reduced between time periods. Hence, changing emergency responder



**Fig. 1.** Objective function value for IRLP accrued in each time period for  $K = 5$  and  $T = 10$  for the Hanover County data set, where  $t = 0$  represents the objective value without any repair and  $t = \infty$  represents the objective value after all arcs are installed.

locations improves emergency responders response times during the recovery phase. We observed at most two changes in the locations of the emergency responders in all of the computational experiments reported in Table 1 even there is no cost for doing so. In some of our experiments not reported in this study, we observed up to 5 changes in locations when a large portion of the road infrastructure is damaged (i.e.,  $|A'|$  is large relative to  $|A|$ ).

Fig. 3 provides a visual representation of the optimal restoration plan for the instance with  $K = 5$  and  $T = 10$  including the arcs used to assign demand to an emergency responder for time periods 0, 5 and 10.

**Table 3**

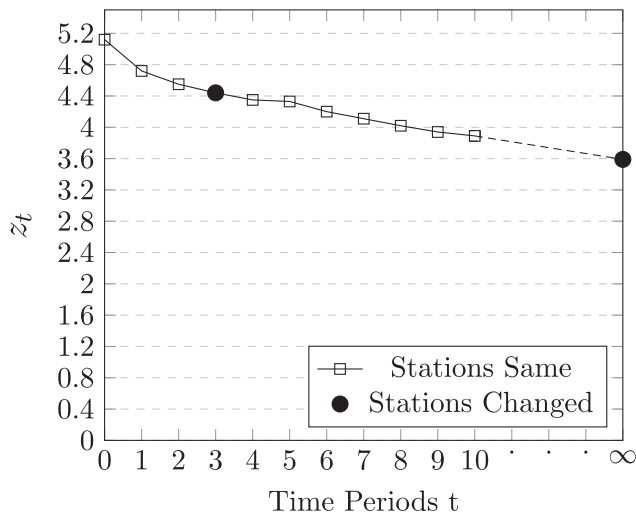
Open facilities for IRLP for the Hanover County data set in each time period for  $K = 5$  and  $T = 10$ .  $T = 0$  shows open facilities using only initial arcs in the network,  $T = \infty$  shows open facilities after repairing all arcs.

Facilities	Time periods $t$											$T = \infty$	
	$T = 0$	1	2	3	4	5	6	7	8	9	10		
1	×	×											×
2													×
3													×
4	×	×	×	×	×	×	×	×	×	×	×	×	×
5													
6	×	×	×	×	×	×	×	×	×	×	×	×	×
7	×	×	×	×	×	×	×	×	×	×	×	×	×
8	×	×	×	×	×	×	×	×	×	×	×	×	×
9								×	×	×	×		
10	×	×	×	×	×	×	×	×	×	×	×	×	×
11													×
12													
13													
14	×	×	×	×	×	×	×	×	×	×	×	×	
15			×	×	×	×	×	×	×	×	×	×	×
16	×	×	×	×	×	×	×	×	×	×	×	×	

**Table 4**

Open facilities for the c-IRLP for the Hanover County data set in each time period for  $K = 5$  and  $T = 10$ .  $T = 0$  shows open facilities using only initial arcs in the network,  $T = \infty$  shows open facilities after repairing all components.

Facilities	Time periods $t$											$T = \infty$	
	$T = 0$	1	2	3	4	5	6	7	8	9	10		
1	×	×	×	×	×	×	×	×	×	×	×	×	×
2													
3				×	×	×	×	×	×	×	×	×	×
4	×	×	×	×	×	×	×	×	×	×	×	×	×
5													
6	×	×	×	×	×	×	×	×	×	×	×	×	×
7	×	×	×	×	×	×	×	×	×	×	×	×	×
8	×	×	×	×	×	×	×	×	×	×	×	×	
9													
10	×	×	×	×	×	×	×	×	×	×	×	×	×
11													×
12													
13													
14	×	×	×										
15													×
16	×	×	×	×	×	×	×	×	×	×	×	×	

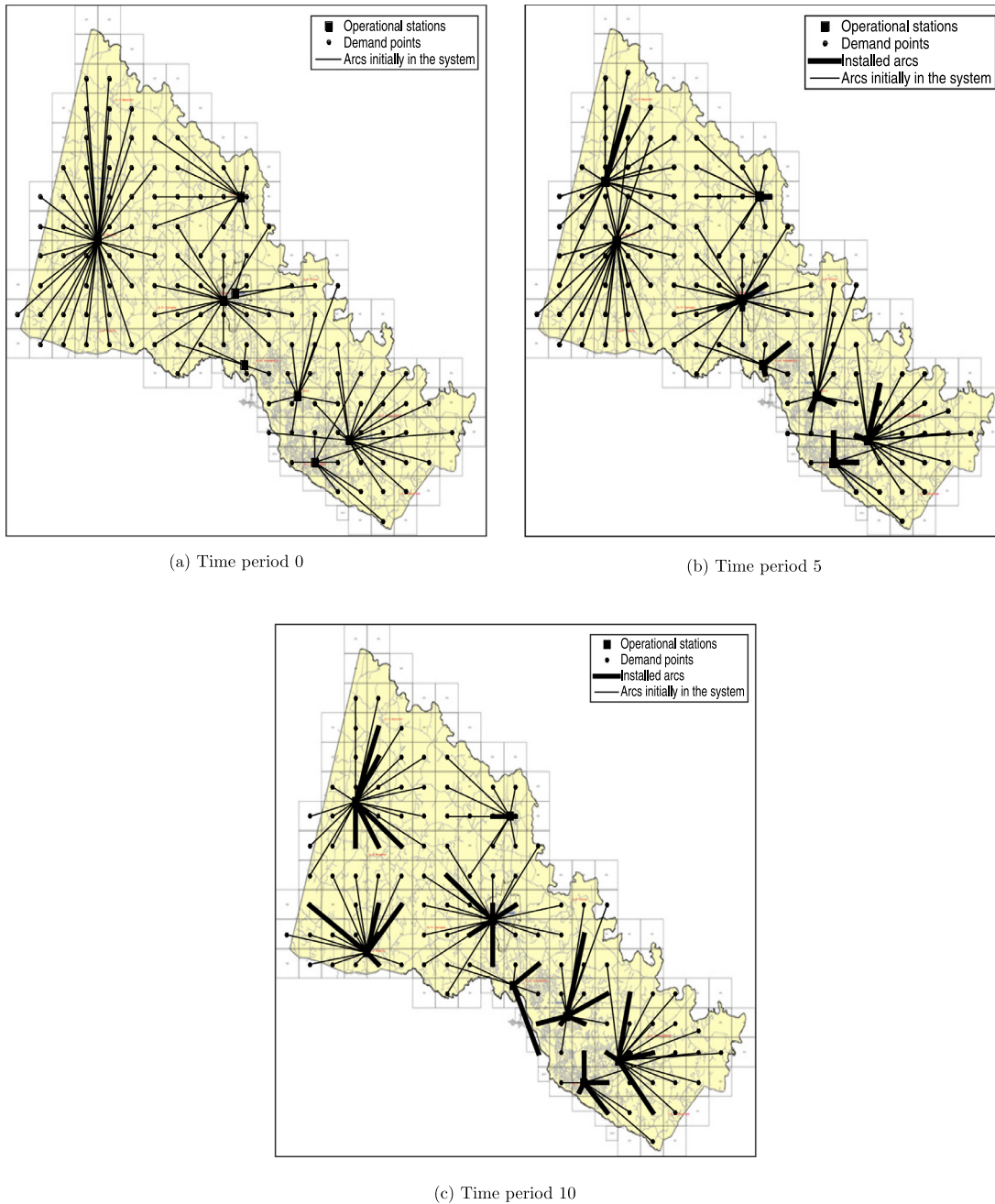


**Fig. 2.** Objective function value of the c-IRLP accrued in each time period for  $K = 5$  and  $T = 10$  for the Hanover County data set, where  $t = 0$  represents the objective value without any repair and  $t = \infty$  represents the objective value after all components are installed.

Thin lines show arcs from  $A$  and thick lines show installed arcs from  $A'$  used in the optimal solution for given time periods. Fig. 3(a) shows the optimal solution with initial arcs in the network selected from  $A$ . Fig. 3(b) and (c) show the solution after five and ten time periods, respectively, and the thick black lines in these sub-figures illustrate the arcs that have been installed. The number of installed arcs changes depending on the number of recovery crews, the time periods, and processing times of the installed arcs. Our computational experiments suggest that it is best to install arcs that are connected to the most critical facility locations prior to and after emergency responders are located there. As a result, the recovery crews are essentially co-located with the emergency responders. The solutions can help decision-makers decide where to initially locate emergency responders after a disaster strikes, as well as how to move these responders during the recovery phase. The solutions therefore shed light on how to prioritize the installation of arcs in a recovery plan to aid in the delivery of time-critical services.

**3.2. Computational results for component based IRLP (c-IRLP)**

Computations for the c-IRLP were performed on a computer with a 1.4 GHz Intel Core 5 Duo Processor with 4GB of RAM. We used Gurobi 6.5.2. to solve the integer programming model that was coded in



**Fig. 3.** The installation of arcs for IRLP for the Hanover County data set for optimal restoration effort  $K = 5$  and over the time horizon  $T = 10$ . Figure (a) shows the solution using initial arcs. The thick arcs represent installed arcs in  $A'$  that have been installed up until time periods 5 and 10 in Figures (b) and (c), respectively. The thin arcs represent arcs in  $A$  used in the solution.

Python. Each experiment was run with a 3600-s time limit.

We use the Hanover County data set as described in Section 3.1 for the c-IRLP with  $|C'| = 971$  installable components. The sets  $AC(c)$  represent the arcs in  $A'$  that are shortened when component  $c \in C'$  is installed, with  $|AC(c)| \leq 5$  and  $\cup_{c \in C'} AC(c) = A'$ . Processing times for installable components,  $p_c$ , are integers randomly generated from a discrete  $U[1, 5]$  distribution. If the installation of a component affects several arcs, the processing time of the component is assigned as a randomly generated integer from a discrete  $U[2, 5]$ . As in IRLP, we allow  $P = 8$  available facilities to be open in each time period, and we vary the number of time periods  $T$  and the number of network recovery crews  $K$  in the computational experiments.

Table 5 summarizes the objective values and running times for the Hanover County data set. The  $|A|$ ,  $|C'|$  and  $|J|$  columns report the

number of arcs initially in the network, number of installable components, and the number of facility locations, while the columns labeled  $K$  and  $T$  report the number of recovery crews and the length of the time horizon. The “Lagrangian Relaxation (LR) Lower Bound” column reports the lower bound values obtained from the Lagrangian relaxation and the running time in seconds in parentheses. The LR lower bounds are better than the lower bounds obtained from the linear programming relaxations for all experiments in Table 4. We report the upper bound we obtained from Lagrangian and linear relaxation heuristic introduced in Section 2.2.2 in the column “Lagrangian and Linear Relaxation Upper Bound” which yields better upper bounds than the upper bounds using the Lagrangian heuristic. The “Optimal Solution w/ Starting Partial Feasible Solution” column reports the optimal solution and the computation time when we set starting values for variables  $x$  and  $y$  for the

**Table 5**

The optimal solutions and bounds for the c-IRLP for the Hanover County data. Computational time, in seconds, is shown in the parentheses. "Lagrangian and Linear Relaxation Upper Bound" reports the upper bound obtained from the Lagrangian and linear heuristic and the computation time using Gurobi. "Optimal Solution w/ Starting Partial Feasible Solution" reports the optimal objective function value and computation time when we set starting values. "Optimal Solution w/ Standard Implementation" reports the optimal objective function value and the computation time when using the standard implementation with Gurobi.

$ A $	$ C $	$ I $	$ J $	$P$	$K$	$T$	Linear Relaxation Lower Bound	Lagrangian Relaxation Lower Bound	Lagrangian & Linear Relaxation Upper Bound	Optimal Solution w/ Starting Partial Feasible Solution	Optimal Solution w/ Standard Implementation
2000	971	125	16	8	3	25	93.52 (43)	96.93 (82)	104.49 (196)	102.35 (219)	102.35 (176)
2000	971	125	16	8	4	25	89.47 (46)	93.73 (85)	101.79 (381)	99.22 (262)	99.22 (281)
2000	971	125	16	8	5	25	86.30 (56)	91.35 (113)	99.76 (651)	96.91 (638)	96.91 (1371)
2000	971	125	16	8	3	30	109.12 (58)	113.85 (107)	123.46 (282)	120.41 (446)	120.40 (360)
2000	971	125	16	8	4	30	104.22 (102)	110.01 (163)	120.27 (629)	116.78 (978)	116.78 (217)
2000	971	125	16	8	5	30	100.40 (226)	107.31 (303)	117.74 (1283)	114.24 (3600) (Gap 0.11%)	114.24 (3600) (Gap 0.03%)
2000	971	125	16	8	3	35	124.22 (85)	130.41 (146)	141.85 (651)	138.17 (1093)	138.17 (1295)
2000	971	125	16	8	4	35	118.01 (187)	126.04 (278)	138.21 (1797)	134.09 (3600) (Gap 0.01%)	134.15 (3600) (Gap 0.05%)
2000	971	125	16	8	5	35	114.08 (205)	123.19 (292)	135.04 (2074)	131.08 (3600) (Gap 0.13%)	131.27 (3600) (Gap 0.26%)

indexes in the set  $S_1$  and  $S_2$  that we describe in the Section 2.2.2. Note that the computation time includes the time to compute sets  $S_1$  and  $S_2$ . The last column entitled "Optimal Solution w/ Standard Implementation" reports the optimal objective function value and the computation time in parentheses when solving the model using standard implementation. A time limit of 3600 s was imposed for all instances. Six out of nine experiments are solved optimally while three out of nine experiments yielded a solution with an optimality gap of less than 0.26% within the time limit. We observe that using the Lagrangian and linear relaxation heuristic, we obtain an upper bound within 3% of the optimal solution value within 2074 s for each experiment. In eight out of nine experiments, computational times of the upper bound are shorter than the computational times of the optimal solution using standard implementation. In addition, when we set starting values for the variables  $x$  and  $y$  for indexes in  $S_1$  and  $S_2$ , respectively, five out of nine experiments result in faster computation times or an improved gap.

Fig. 2 shows the values of  $z_t$  over the time horizon for the c-IRLP instance with  $K = 5$  and  $T = 10$  using the Hanover County data set. We observe a decreasing pattern in the objective function value  $z_t$  over the time horizon as the recovery crews install components from set  $C'$  and then emergency responders use arcs which benefit from the installed components. Table 5 shows the locations of the open facilities over the time horizon for same instance. The locations of open facilities change once during this experiment, which are also indicated in Fig. 2. In time period 3, Facility 14 closes and Facility 3 opens.

Fig. 4 provides a visual representation of the optimal restoration plan for the instance with  $K = 5$  and  $T = 10$  including the arcs used to assign demand to an emergency responder for time periods 0, 5 and 10. Thin lines show arcs from  $A$  and thick lines show arcs from  $A'$  whose distances shorten by using installed components from  $C'$  and used in the optimal solution up until that time period. At the end of the time horizon 10, 39 components were installed and 42 arc distances were shortened using those installed components. The number of installed components changes depending on the number of recovery crews, the time periods, and processing times of the installed components. As in IRLP experiments, the c-IRLP computational experiments suggest that it is best to install components that are beneficial to the most critical facility locations prior to and after emergency responders are located there with a more realistic and practical setting.

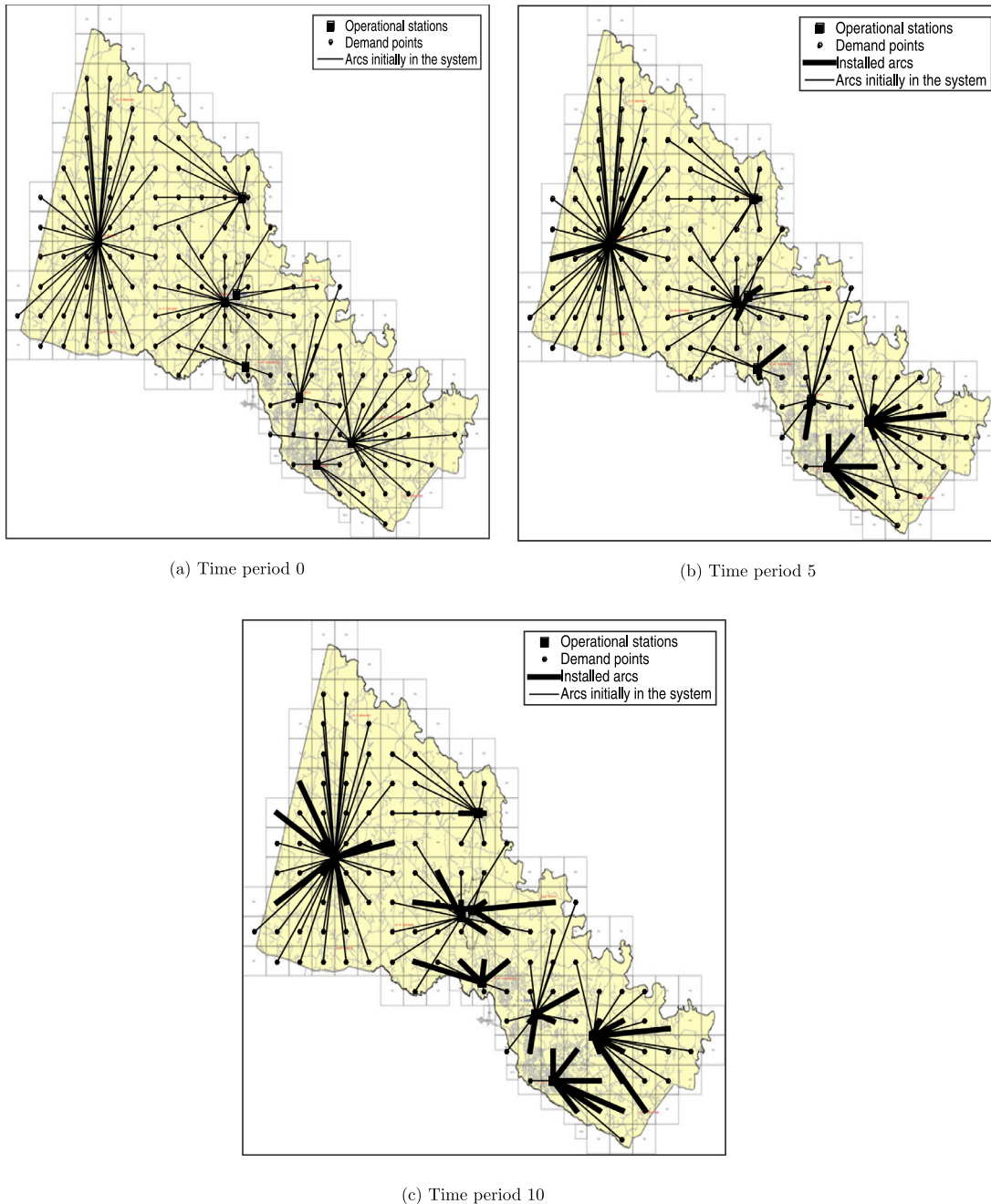
#### 4. Conclusions

In this paper, we introduce an integrated restoration and location problem (IRLP) and model the interdependencies between critical

infrastructure systems and service providers. We study a P-median model variation with the goal of minimizing the cumulative weighted distance between emergency responders and demand nodes over a time horizon. We also introduce the c-IRLP, an extension to this model to approximately model components of disrupted arcs between demand and facility locations. The models can be used to model the delivery of a variety of time-sensitive critical services after a disaster event occurs. The models provide insight into how the activities of the two types of service providers should be coordinated and which network components should be restored sooner during recovery.

We develop integer programming formulations of our models. We introduce Lagrangian relaxation dual problems to obtain a lower bound to the models. A subgradient algorithm is used to solve the Lagrangian relaxation dual problems. Solving the models using Lagrangian relaxation results in a better lower bound than the linear relaxation lower bound in all computational experiments we report. We further develop a Lagrangian heuristic to identify a feasible solution to IRLP model, which provides an upper bound that is used to decrease the size of the branch and bound tree. Using the upper bound obtained from the Lagrangian heuristic, we show an improvement in the computational time of IRLP. For comparison, we also show the time to solve original model without the upper bounds. Numerical experiments with real-world data sets and with data from Beasley's OR Library are used to examine the quality of the Lagrangian relaxation lower bound and the improvement in IRLP. For the c-IRLP, we develop a Lagrangian and linear relaxation heuristic by using a feasible solution to Lagrangian relaxation of the c-IRLP and enhancing the feasible solution using linear relaxation of the c-IRLP. The resulting upper bound from the Lagrangian and linear relaxation heuristic improved the Lagrangian relaxation heuristic upper bound in all cases tested. We also use this enhanced feasible solution to set starting values for some variables in the standard implementation. This approach helps the solver obtain an incumbent solution faster than without setting starting values, and hence, leads to improved computational times for some experiments. Numerical experiments with real-world data are used to examine the quality of the Lagrangian relaxation lower bound, as well as the Lagrangian and linear relaxation heuristic upper bound.

After an extreme event, damage to infrastructure systems can delay the delivery of essential services. These proposed models and analysis demonstrate the importance of modeling the interdependencies between infrastructure systems and emergency services. Solutions to the models provide a plan for restoring the most critical network components to the models and provide insight into how to deliver essential services during the recovery. We were able to solve most instances in less than one hour of CPU time for Hanover County and Beasley's data



**Fig. 4.** The installation of arcs for the c-IRLP for the Hanover County data set for optimal restoration effort  $K = 5$  and over the time horizon  $T = 10$ . Figure (a) shows the solution using initial arcs. The thick arcs in Figure represent installed components in  $C'$  that have been installed up until time periods 5 and 10 in Figures (b) and (c), respectively. The thin arcs represent arcs in  $A$  used in the solution.

sets, which suggests that the models can be used to aid decision makers with emergency response and recovery issues in real-time after a disaster.

There are several directions which the models can be extended. First, other objectives such as coverage are often used to model the quality of service in disaster settings. Second, we could take transportation restrictions under consideration, where both emergency responders and recovery crews can only move to “adjacent” facilities or components in sequential time periods. In the next section, we present a new model that consider coverage objective and also consider relocation restrictions for emergency responders.

### Acknowledgments

This work was funded by the National Science Foundation (awards 1444219 and 1541165). The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the National Science Foundation. The authors wish to thank the editor and the four anonymous referees for their helpful comments and suggestions, which has resulted in a significantly improved manuscript.

## Appendix A

In the original c-IRLP problem, we consider each component in the set  $C'$  to have an associated processing time. One particular case of interest is when each component in  $C'$  has identical processing times. Without loss of generality, we assume that all processing times are one. In this case, we can simplify the formulation, and we present the simplified formulation here.

In the recovery unit, we no longer need to assign component installations to individual network recovery crews. Instead, we select the  $K$  components that are installed by any of the network recovery crews in each time period. As a result, the design decision variables  $\tilde{x}_{cijt}$ ,  $c \in C'$ ,  $(i, j) \in A'$ ,  $t = 1, \dots, T$  and  $y_{jt}$ ,  $j \in J$ ,  $t = 1, \dots, T$  remain the same. The recovery decision variables are simplified to:

- $\xi_{ct} = 1$  if one of the recovery crews completes the repairing of component  $c \in C'$  and 0 otherwise, for  $t = 1, \dots, T$ .

The integer programming model is formulated as follows.

$$Z = \min \sum_{t=1}^T \sum_{c \in C'} \sum_{(i,j) \in AC(c)} w_{it} d_{cij} \tilde{x}_{cijt} + \sum_{t=1}^T \sum_{(i,j) \in A} w_{it} d_{0ijt} \tilde{x}_{0ijt} \quad (\text{A.1})$$

$$\text{s. t. } \sum_{c \in C'} \sum_{j: (i,j) \in AC(c)} \tilde{x}_{cijt} + \sum_{j: (i,j) \in A} \tilde{x}_{0ijt} = 1 \quad \text{for } i \in I, t = 1, \dots, T \quad (\text{A.2})$$

$$\sum_{j \in J} y_{jt} \leq P \quad \text{for } t = 1, \dots, T \quad (\text{A.3})$$

$$\tilde{x}_{cijt} - y_{jt} \leq 0 \quad \text{for } c \in C', (i, j) \in AC(c), t = 1, \dots, T \quad (\text{A.4})$$

$$\tilde{x}_{0ijt} - y_{jt} \leq 0 \quad \text{for } (i, j) \in A, t = 1, \dots, T \quad (\text{A.5})$$

$$\tilde{x}_{cijt} \in \{0, 1\} \quad \text{for } c \in C', (i, j) \in AC(c), t = 1, \dots, T \quad (\text{A.6})$$

$$\tilde{x}_{0ijt} \in \{0, 1\} \quad \text{for } (i, j) \in A, t = 1, \dots, T \quad (\text{A.7})$$

$$y_{jt} \in \{0, 1\} \quad \text{for } j \in J, t = 1, \dots, T \quad (\text{A.8})$$

$$x_{cijt} \leq \sum_{s=1}^t \xi_{cs} \quad \text{for } c \in C', (i, j) \in AC(c), t = 1, \dots, T \quad (\text{A.9})$$

$$\sum_{c \in C'} \xi_{ct} \leq K \quad \text{for } t = 1, \dots, T \quad (\text{A.10})$$

$$\xi_{ct} \in \{0, 1\} \quad \text{for } c \in C', t = 1, \dots, T \quad (\text{A.11})$$

The objective and constraint sets (A.1)–(A.8) are the same as in the c-IRLP. Constraint set (A.9) ensures that a component can be used to assign demand only if its installation was completed at any time prior to time  $t$ . Constraint set (A.10) ensures that the number of component installed cannot exceed the total number of recovery crews. Constraint set (A.11) requires the recovery decision variables to be binary.

Computational experiments suggest that this formulation drastically improves computational time when solving the models using off-the-shelf MILP solvers such as Gurobi as compared to the original component based IRLP formulation, particularly in cases when there are a large number of recovery crews.

## Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ejor.10.1016/j.ejor.10.1016/j.ejor.2018.08.001.

## References

- [1] Hurricane Sandy Rebuilding Task Force. Hurricane Sandy rebuilding strategy. Washington DC: US Department of Housing and Urban Development; 2013.
- [2] Kaufman S, Qing C, Levenson N, Hanson M. Transportation during and after Hurricane Sandy. Tech. Rep. Rudin Center for Transportation, NYU Wagner Graduate School of Public Service; 2012.
- [3] Jennings CR. Fires during the 2012 Hurricane Sandy in Queens, New York: a first report. Tech. Rep. John Jay College of Criminal Justice The City University of New York, USA; 2013.
- [4] FEMA. Hurricane Sandy recovery efforts one year later Tech. Rep. Federal Emergency Management Agency; 2013. <https://www.fema.gov/media-library-data/1382967173777-7411aa1b6d729a8a97e84dbba62083d8/FEMASandyOneYearFactSheet508.pdf>.
- [5] Lee EE, Mitchell JE, Wallace WA, et al. Restoration of services in interdependent infrastructure systems: a network flows approach. Syst Man Cybern Part C: Appl Rev IEEE Trans 2007;37(6):1303–17.
- [6] Nurre SG, Cavdaroglu B, Mitchell JE, Sharkey TC, Wallace WA. Restoring infrastructure systems: an integrated network design and scheduling (INDS) problem. Eur J Oper Res 2012;223(3):794–806.
- [7] Cavdaroglu B, Hammel E, Mitchell JE, Sharkey TC, Wallace WA. Integrating restoration and scheduling decisions for disrupted interdependent infrastructure systems. Ann Oper Res 2013;203(1):279–94.
- [8] Sharkey TC, Cavdaroglu B, Nguyen H, Holman J, Mitchell JE, Wallace WA. Interdependent network restoration: on the value of information-sharing. Eur J Oper Res 2015;244(1):309–21.
- [9] Lee EE, Mitchell JE, Wallace WA. Network flow approaches for analyzing and managing disruptions to interdependent infrastructure systems. 2009.
- [10] Nurre SG, Sharkey TC. Integrated network design and scheduling problems with parallel identical machines: complexity results and dispatching rules. Networks 2014;63(4):306–26.
- [11] Gutfraind A, Bradonjić M, Novikoff T. Optimal recovery of damaged infrastructure network. Tech. Rep. University of Texas at Austin, Austin; 2012.
- [12] Baxter M, Elgindy T, Ernst AT, Kalinowski T, Savelsbergh MW. Incremental network design with shortest paths. Eur J Oper Res 2014;238(3):675–84.
- [13] Engel K, Kalinowski T, Savelsbergh MW. Incremental Network Design with Minimum Spanning Trees. Tech. Rep.. 2013arXiv:1306.1926. arXiv preprint.

- [14] Duque PAM, Dolinskaya IS, Sørensen K. Network repair crew scheduling and routing for emergency relief distribution problem. *Eur J Oper Res* 2016;248(1):272–85.
- [15] Averbakh I, Pereira J. The flowtime network construction problem. *IIE Trans* 2012;44(8):681–94.
- [16] Wang Q, Batta R, Bhadury J, Rump CM. Budget constrained location problem with opening and closing of facilities. *Comput Oper Res* 2003;30(13):2047–69.
- [17] Snyder LV, Daskin MS. Reliability models for facility location: the expected failure cost case. *Transp Sci* 2005;39(3):400–16.
- [18] Cui T, Ouyang Y, Shen Z-JM. Reliable facility location design under the risk of disruptions. *Oper Res* 2010;58(4-part-1):998–1011.
- [19] O'Hanley JR, Scaparra MP, García S. Probability chains: a general linearization technique for modeling reliability in facility location and related problems. *Eur J Oper Res* 2013;230(1):63–75.
- [20] Scaparra MP, Church RL. A bilevel mixed-integer program for critical infrastructure protection planning. *Comput Oper Res* 2008;35(6):1905–23.
- [21] Church RL, Scaparra MP. Protecting critical assets: the r-interdiction median problem with fortification. *Geogr Anal* 2007;39(2):129–46.
- [22] Church RL, Scaparra MP, Middleton RS. Identifying critical infrastructure: the median and covering facility interdiction problems. *Ann Assoc Am Geogr* 2004;94(3):491–502.
- [23] Losada C, Scaparra MP, O'Hanley JR. Optimizing system resilience: a facility protection model with recovery time. *Eur J Oper Res* 2012;217(3):519–30.
- [24] Çelik M, Ergun Ö, Keskinocak P. The post-disaster debris clearance problem under incomplete information. *Oper Res* 2015;63(1):65–85.
- [25] Pinedo M, Hadavi K. *Scheduling: theory, algorithms and systems development*. Operations research proceedings 1991. Springer; 1992. p. 35–42.
- [26] Gendreau M, Laporte G, Semet F. The maximal expected coverage relocation problem for emergency vehicles. *J Oper Res Soc* 2006;57(1):22–8.
- [27] Fisher ML. The Lagrangian relaxation method for solving integer programming problems. *Manage Sci* 2004;50(12 Suppl):1861–71.
- [28] Held M, Karp RM. The traveling-salesman problem and minimum spanning trees: part II. *Math Program* 1971;1(1):6–25.
- [29] Martin RK. *Large scale linear and integer optimization: a unified approach*. Springer Science & Business Media; 2012.
- [30] Danna E, Rothberg E, Le Pape C. Exploring relaxation induced neighborhoods to improve MIP solutions. *Math Program* 2005;102(1):71–90.
- [31] Beasley J.E. *The operational research library*. <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/pmedinfo.html>; 2015.
- [32] Polyak BT. *Introduction to optimization*. Translations series in mathematics and engineering. Optim Software 1987.