A Two-stage Resource Allocation Model for Lifeline Systems Quick Response with Vulnerability Analysis

C. Zhang[1,2], X. Liu*[1,2], YP. Jiang[2], B. Fan[3], X. Song[4]

1. Department of Industry Engineering, Shanghai Jiaotong University, P.R.China
2. Department of Industrial and Systems Engineering, National University of Singapore, Singapore
3. School of International and Public Affairs, Shanghai Jiaotong University, P.R.China
4. School of Mathematics, University of Portsmouth, UK

Corresponding author: Professor Xiao. LIU

Department of Industrial Engineering management, Shanghai Jiaotong University, 800 Dongchuan Road, Min-Hang District, Shanghai 200240, P.R.China

Email: x_liu@sjtu.edu.cn

Phone: +65-6601-4089

Fax: +65-6601-4089

Abstract: The internal complexity of lifeline systems and their interdependencies amplify the vulnerability of external disruptions. We consider lifeline infrastructures as a network system with supply, transshipment, demand nodes and arcs constructed between node-pair for conveying service flows. The complex interactive network system can be modeled as multi-layered graphs, whereby the power network depends on the gas network linked through the gasified power plants. Similarly, the water network depends on both quality and quantity of power supply. A successful emergency rescue can make lifeline infrastructures more resilient against natural disasters and unexpected accidents. This study focuses on a resource allocation and schedule problem to restore the most critical components quickly in the multiple interdependent lifeline infrastructures under disruptions. The key objectives of quick response model include reducing the overall losses caused by the accidents, and restoring system functions as quickly as possible. The Resource Allocation Model (RAM) for rescue was formulated as a two-stage mixed-integer programming, in which the first stage problem aims to minimize the total losses, while the second stage problem is to optimize resource allocation for rescue service within the rescue time horizon using the proposed heuristic algorithm in polynomial complexity. In the meantime, those tasks/components to be repaired are selected by the proposed vulnerability analysis method to guarantee the optimal whole network efficiency, and then put them into the Resource Allocation Model. The simulation results demonstrate that the proposed approaches are both efficient and effective to solve the real-life post-disaster resource allocation problem.

Keywords: Decision support systems, Quick response, Resource allocation, Two-stage programming, Vulnerability analysis

1. Introduction

In the past century, the infrastructures of urban cities have faced immense strains as a result of dramatic growth in population. Correspondingly, the increasing complexity and interdependencies of lifeline infrastructures pose new challenges for security and operations management because of their large-scale, nonlinear, and time-dependent properties. Such lifeline systems are often considered as a network system consisting of supply, demand, and transshipment components (nodes and arcs) including electric power, gas, water supply, food, telecommunications, and transportation, to provide platforms for service delivery. The complexity nature of the network makes the lifeline systems vulnerable to failures, which may cause widespread negative consequences. It has being becoming the most susceptible part for the economic, social, and environment development in all cities (De Sherbinin, 2007; Aven, 2011; Murray, 2013).

The occurrence of several cascading failures in the past typically causes huge property loss and significant restoration cost (Chai, 2011; Collier, 2008). For example, in July and August of 1996, the Western US grid experienced outages affecting 11 of the US States and 2 Canadian Provinces. More recently in December 1998 blackout in San Mateo cascaded to affect 2 million people in the San Francisco Bay Area. Therefore, the cities should take all feasible measures to strengthen their response capabilities to ensure essential services. From the viewpoint of sustainability, a city cannot achieve the goal of sustainability if the operations of its lifeline network are vulnerable (Turner, 2003; Turner II, 2010).

In the ensuing sections, we shall elaborate on the existing researches, which focus on the survivability of systems under nature disasters or man-made accidents (Murray, 2007; San, 2007; Kamissoko, 2014). The first stream of the research mainly focuses on malicious attacks and network interdiction problems based on the complex network topology methods (Azaiez, 2007; Hausken, 2011; Rocco, 2011). The second stream studies the network flow problems under disruptions (Garg and Smith, 2008; Sorokin, 2013;), which is formulated as IO model that could effectively evaluate the performance of the whole network at each time period. The third stream focuses on network vulnerability analysis including network design and operations against blackout based on the network topology, which is largely used to identify the critical components in the network (Fiedrich, 2000; Alguacil, 2010; Zio, 2012).

The approaches used to solve the post-disaster resource allocation problem include applied statistical and probabilistic models combined with multi-objective programming, two-stage model and dynamic model (Yan, 2009; Shan, 2012; Samuel, 2012; Yatesa, 2012; Srdjevic, 2013). Specifically, Barbarosoğlu and Arda (2004) proposed a two-stage stochastic programming model to plan the transportation of vital first-aid commodities to disaster-affected areas during emergency response; Lee (2007) formulated a mixed integer model to design optimal responding strategies for emergencies with the objective of minimizing cost; Scaparra and Church (2008) identified the most cost-effective way of allocating protective resources among the facilities of an existing but vulnerable system using bi-level programming in such a way that the impact of the most disruptive attack on the unprotected facilities is minimized; Cavdaroglu et al. (2013) formulated a service restoration and job scheduling in interdependent systems; and Wex et al. (2014) proposed and compared several heuristics for allocating available rescue units to incidents with the objective of minimizing the sum of completion times weighted by severity.
Furthermore, since the resource allocation problem could be generalized to the unrelated parallel machine scheduling problems, many heuristic algorithms could also be used to solve the resource allocation problem (Su, 2009; Lin, 2011; Yeh, 2013).

However, in the existing resource allocation studies, there are two problems that require further discussion. The first is that the objective of most models only focuses on minimization of the overall costs (Brown, 2005; Shen, 2013; Zhang, 2011), while studies focus on minimization of the completion time is by far limited (Faraj, 2006; Wex, 2014). The total losses could not be solely measured in terms of costs because the consequences as a result of accidents are hard to be assessed, in other word, it doesn’t make sense to trade off the costs and the restoration time. Therefore, during the rescue time horizon, the minimization of the restoration time should take priority for stakeholders in their decision-making process. The second problem is that in the accidents, the interconnectivities of the lifeline network may trigger cascading failures, which can result in the amplifications of the overall losses, therefore, the whole network efficiency shall be considered as the most important metric during the resource allocation assessment procedure. In this study, we consider the emergency allocation problem with limited resources and restoration time for the lifeline systems with the consideration of the whole network efficiency. To solve the resource allocation and scheduling problem, we utilize the network system vulnerability analysis method to sort those critical components to be repaired, and then put them into the two-stage mixed integer model formulated.

This paper is organized as follows. Section 2 presents the lifeline emergency resource allocation model, which is formulated as a two-stage programming. Section 3 presents the proposed algorithms to solve the two-stage programming. Section 4 demonstrates the computational results and some discussions. Conclusions are detailed in the final Section.

2. **Two-stage Resource Allocation Model**

2.1 Problem Description

In the study, we focus on lifeline systems with three sub-systems，which include gas, power, and water systems, whereby the power network depends on the gas network through the gasified power plants, and the water network depends on both quality and quantity of power supply. Thus, there exist functional connections among the different layers, which means the supply layers are too important to trigger the demand layers failure if any disruption happens. Meanwhile, the network is composed of supply, transshipment, and demand components in each layer as shown in Fig.1.

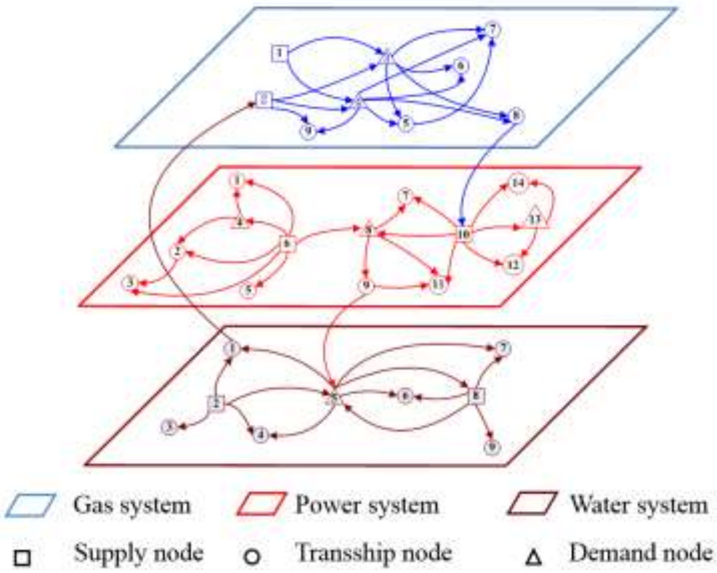


Fig. 1. Interdependent multi-layered lifeline systems network

Such three-layered network could be denoted as a directed graph $G(A,V)$ with nodes $v_n \in V\,(n = 1,2,\ldots,N)$, and directed arcs $a_i \in A\,(i = 1,2,\ldots,I)$ which connect service flow within each layer and between the node-pair $(v_m, v_n) \in P$, where P represents the set of node-pair. Moreover, a_i^r $(r = 1,2,\ldots,R')$ represents the rth destroyed arcs with $a_i^r \in$ □; accordingly, $v_n^r\,(r = R'+1, R'+2,\ldots,R)$ represents the rth destroyed nodes, with $v_n^r \in$ □ . In other word, the destroyed components include all destroyed arcs and nodes, which belong to destroyed components set □. For rescue tasks, we have rescue team k $(k \in K)$, each rescue team has different capabilities to repair the destroyed components as each rescue team can be a group consisting of technicians with different skills.

The rescue procedure is that the top-layer decision makers give orders to rescue teams, and then rescue teams have to meet the requirements of the task. In this study, we stand on the rescue teams' point of view, the goal is to take time priority against restoration costs because of the time sensitive character in emergency case to optimize the efficiency of the whole lifeline system. To achieve the goal, we first select critical destroyed components $a_i^r \in$ □ to be repaired to ensure the maximization of lifeline system network efficiency within rescue time horizon T, then, we assign the determined tasks from top-layer decision makers to each rescue team k ($k \in K$) within the time horizon T.

2.2 Notations and Variables

In order to facilitate our explanation, the following notations and variables will be used throughout this paper.

Parameters:

u_i	capacity of arc a_i
u'_n	capacity of transshipment node $v_n \in V_{n=}$
f_n	supply or demand at node v_n
c_{op}^i	cost associated with flow along arc a_i
c_{re}^r	repair cost of arc a_i^r
ρ_n	punishment cost for unmet demand of node v_n per unit
η	restoration cost of per time unit
τ_{rk}	time required by the k th rescue team to process disrupted arc a_i^r
T_{rk}	completion time for the k th rescue team to process arc a_i^r

Variables:

x_{it}	flow on arc a_i at period t
s_{nt}	slack associated with node v_n at period t
l_r	$l_r \in \{0,1\}$, $l_r = 1$ if arc a_i^r is selected to repair; otherwise, $l_r = 0$
z_{rk}	$z_{rk} \in \{0,1\}$, $z_{rk} = 1$ if arc a_i^r is allocated to the k th team; otherwise, $z_{rk} = 0$
$b_{rr'k}$	$b_{rr'k} \in \{0,1\}$, $b_{rr'k} = 1$ if a_i^r is a preceding component of $a_i^{r'}$ in the processing list

of the k th team; otherwise, $b_{rr'k} = 0$

y_{mt}^{n} $y_{mt}^{n} \in \{0,1\}$, $y_{mt}^{n} = 1$ if connection of node-pair (v_m, v_n) is effective at period t; otherwise, $y_{mt}^{n} = 0$

β_{it} $\beta_{it} \in \{0,1\}$, $\beta_{it} = 1$ if arc a_i is destroyed at period t; otherwise, $\beta_{it} = 0$

2.3 Problem Assumptions

The followings are the underlying assumptions in the model for the above mentioned problem.

- The initial condition of each component in the infrastructure system is known by the sensors/monitors in the systems once the event happened;
- Unmet demand of supply nodes will cause both transshipment and demand nodes total failure for any node-pair among different layers caused by their interdependencies among the system in terms of their supply and demand links (For example, if the power supply cannot be satisfied for the corresponding infrastructure in the water system, then the water infrastructure will be out of work);
- We assume that all the teams are capable to repair the destroyed components with different completing time to recover their original operation level except the dysfunction of the components;
- Without loss of generality, any un-repaired arc is assumed to lose its function during the time horizon.

2.4 Mathematical Model

We begin by giving definitions for the problem as below.

Definition 1. Define all arcs a_i flow into nodes v_n as the *inflow arcs*; which belong to set δ_n^+; accordingly, define all arcs a_i flow out of nodes v_n as *outflow arcs*, which belong to set δ_n^-.

Definition 2. Define the nodes v_n in any layer with solely *outflow arcs* as *supply nodes*, which belong to set V_-; accordingly, define the nodes v_n in any layer with solely *inflow arcs* as demand nodes, which belong to set V_+; while define those nodes v_n in any layer with both *inflow* and *outflow arcs* as transshipment nodes, which belong to set $V_=$; furthermore, we have $V_- \cap V_+ = V_+ \cap V_= = V_- \cap V_= = \varnothing$.

Definition 3. For the initial status of any destroyed arc $a_i^r \in \square$, we have $\beta_{it} = 0$ when $t = 0$; while for the initial status of any destroyed node $v_n^r \in \square$, we have $\beta_{it} = 0$ for all arcs $a_i \in \delta_n^+ \cup \delta_n^-$ when $t = 0$.

Definition 4. Let t_r be the restoration period of the rth destroyed component (a_i^r or v_n^r), when $t > t_r$, for arc a_i^r, we have $\beta_{it} = 1$; while for node v_n^r, we have $\beta_{it} = 1$ for all $a_i \in \delta_n^+ \cup \delta_n^- \setminus \square$, which are connected with node v_n^r.

Based on the above assumptions and definitions, the resource allocation model (RAM) can be formulated as two-stage programming model as follows.

Stage I:

$$\text{Min} \quad C = \sum_{t\in T}\Big(\sum_{i:a_i\in A} c_{op}^i x_{it} + \sum_{m:v_m\in V_-\setminus(v_m,v_n)\in P} \rho_m s_{mt} + \sum_{m,n:(v_m,v_n)\in P} \rho_m f_m (1-y_{mt}^n)\Big) + \sum_{r:a_i^r\in\square} c_{re}^r l_r + \eta\Gamma \tag{1}$$

$$\text{s.t.} \quad \sum_{i:a_i\in\delta_n^-} x_{it} \le f_n \qquad \forall t,\ \forall n: v_n \in V_+ \tag{2}$$

$$s_{nt} + \sum_{i:a_i\in\delta_n^+} x_{it} = -f_n \qquad \forall t,\ \forall n: v_n \in V_- \tag{3}$$

$$\sum_{i:a_i\in\delta_n^+} x_{it} - \sum_{j:a_j\in\delta_n^-} x_{jt} = 0 \qquad \forall t,\ \forall n: v_n \in V_= \tag{4}$$

$$\sum_{i:a_i\in\delta_n^+} x_{it} \le u_n' \qquad \forall t,\ \forall n: v_n \in V_= \tag{5}$$

$$s_{mt} \le (1 - y_{mt}^{n})(-f_m) \qquad \forall t,\ \forall m,n:(v_m,v_n)\in P,\ v_m \in V_{-} \tag{6}$$

$$\sum_{i:a_i \in \delta_n^{-}} x_{it} \le f_n y_{mt}^{n} \qquad \forall t,\ \forall m,n:(v_m,v_n)\in P,\ v_m \in V_{+} \tag{7}$$

$$\sum_{i:a_i \in \delta_n^{+}} x_{it} \le -f_n y_{mt}^{n} \qquad \forall t,\ \forall m,n:(v_m,v_n)\in P,\ v_m \in V_{-} \tag{8}$$

$$\sum_{i:a_i \in \delta_n^{+}} x_{it} \le u'_n y_{mt}^{n} \qquad \forall t,\ \forall m,n:(v_m,v_n)\in P,\ v_m \in V_{=} \tag{9}$$

$$x_{it} \le u_i \beta_{it} \qquad \forall t,\ \forall i: a_i^r \in \square \tag{10}$$

$$b_{rr'k} + b_{r'rk} \le z_{rk} \qquad \forall r,r': a_i^r, a_i^{r'} \in \square,\ r \ne r',\ \forall k \tag{11}$$

$$b_{rr'k} + b_{r'rk} \le z_{r'k} \qquad \forall r,r': a_i^r, a_i^{r'} \in \square,\ r \ne r',\ \forall k \tag{12}$$

$$b_{rr'k} + b_{r'rk} \ge z_{rk} + z_{r'k} - 1 \qquad \forall r,r': a_i^r, a_i^{r'} \in \square,\ r \ne r',\ \forall k \tag{13}$$

$$T_{r'k} \ge T_{rk} + p_{r'k} - M(1 - b_{rr'k}) \qquad \forall r,r': a_i^r, a_i^{r'} \in \square,\ r \ne r',\ \forall k \tag{14}$$

$$T_{rk} \ge \sum_{k \in K} p_{rk} z_{rk} \qquad \forall r: a_i^r \in \square,\ \forall k \tag{15}$$

$$x_{it}, s_{nt}, T_{rk} \ge 0 \qquad \forall t,\ \forall n: v_n \in V,\ \forall i: a_i \in A \tag{16}$$

$$y_{mt}^{n}, z_{rk}, b_{rr'k}, \beta_{it} \in \{0,1\} \qquad \forall t,\ \forall i: a_i \in A,\ \forall r: a_i^r \in \square,\ \forall k \tag{17}$$

Stage II:

$$\text{Min} \quad \Gamma = \max_{k \in K} \sum_{r: a_i^r \in \square} z_{rk} \tau_{rk} \tag{18}$$

$$\text{s.t.} \quad \sum_{k \in K} z_{rk} = l_r \qquad \forall r: a_i^r \in \square \tag{19}$$

$$\sum_{r: a_i^r \in \square} (z_{rk} \tau_{rk}) \le T \qquad \forall k \tag{20}$$

$$z_{rk} \in \{0,1\} \qquad \forall k,\ \forall r: a_i^r \in \square \tag{21}$$

In **Stage I** model, expression (1) represents the minimization of the total costs of the network system during time horizon *T*, aggregated from the operation costs, punishment costs for the unmet demands, startup cost, and repairing costs. Constraints (2) to (10) are about the network flow problems, while Constraints (11) to (15) describe the restrictions on the scheduling problems. Constraint (2) limits the total outflow of each supply node to be lower than its supply capacity. Constraint (3) ensures that the slacking flow amount of each demand node, and Constraint (4) enforces the flow balance for transshipment nodes. Constraint (5) sets the restriction that the total outflow of each transship node should not exceed its own transship capacity. Constraint (6) reflects Assumption 2, for node-pair (v_m, v_n), the function of v_n will become invalid if the demand of v_m is not satisfied. Once the demand of v_m has been satisfied, the function of v_n will restore immediately. Constraints (7)-(9) ensure that node v_n is invalid if the binary variable y_{mt}^{n} is zero. In addition, each component has capacity constraint, which is represented in Constraint (10). A precedence relation exists between two components if and only if both of them are assigned to the same rescue team by Constraints (11)-(13). Constraint (14) defines the completion time of each repaired arc considering their precedence relation and processing time. Constraint (15) ensures that the completion time of the first job of each team is no less than its processing time. Constraint (16) ensures non-negativity restrictions and intermediate variables, whilst Constraint (17) ensures the binary variables.

The **Stage II** model can be formulated as a non-identical parallel machine scheduling problem, where the goal is to minimize the makespan as presented in formula (18). Constraint (19) ensures that each selected component can be assigned to sole rescue team. Constraint (20) ensures that the makespan lies within the time horizon, and Constraint (21) ensures the binary variable.

3. **Algorithms Development**

To solve the two-stage model, firstly, we have to select which component/group of components to be repaired, then put these critical components into the two-stage model as input, then a heuristic algorithm is developed to assign the disrupted components a_i^r to rescue team k in **Stage II,** which has been proven NP-hard; knowing the rescue team assignment plan, the rescue sequences of each team can be sorted to minimize the total cost in **Stage I** programming, which is a Mixed Integer Linear Programming problem and can be solved by CPLEX. As a result, the algorithm consists of the following three major steps and can be summarized as follows.

Step 1: Select the critical group of components to be repaired in rescue time horizon to make the whole network be the most efficient, the details of vulnerability analysis process are shown in Section 3.1;

Step 2: Decide the rescue teams assignment scheme based on the tasks/components to be repaired (result from **Step 1**) by solving the **Stage II** model using the proposed heuristic algorithm to ensure restoration time minimization restricted by whole network efficiency maximization, the details of **Stage II** algorithm are shown in Section 3.2;

Step 3: Compute the total rescue costs by solving **Stage I** model using CPLEX to ensure total rescue costs minimization restricted by restoration time minimization, the details are shown in Section 3.3.

3.1 The Critical Groups Selection Based on Vulnerability Analysis

Step 1 is to identify the critical group of components to be repaired, because not all destroyed components can be repaired within given time horizon. Considering the interdependencies and cascading failures in the network, components are evaluated by group rather than individually. For example, although arc *a* may be more important than *b*, *c* individually, the combination of arcs *b* and *c* may have most efficient network performance compared with the arc *a* only. As a result, weighted network efficiency $E(\square^*)$ is employed as the indicator for the selection, which can assess the vulnerability improvement after certain group of arcs $\square^*$ are repaired. $E(\square^*)$ can be calculated by Eq. (22).

$$E(\square^*) = \frac{1}{N(N-1)} \sum_{m,n:v_m \neq v_n \in V} w_{mn} \cdot \frac{1}{d_{mn}} \tag{22}$$

Where $E(\square^*)$ denotes the network efficiency when all destroyed components belong to $\square^*$ are repaired, and N represents the total number of nodes in the network. The parameter d_{mn} denotes the shortest path length between any node v_m and v_n, which is calculated by Dijkstra algorithm. w_{mn} denotes the weight of each path in the network, which is normalized as $w_{mn} = x_{mn} / x_{\max}$, where x_{mn} is the minimal arc flow along the shortest path from v_m to v_n before the accidents and $x_{\max}$ is the maximal arc flow before the accidents.

At meanwhile, the other metric to evaluate the whole infrastructure system performance under study is to restore as many components as possible during time horizon. Since the critical group selection process from destroyed sets $\square$ is proved as NP-hard problem, we employ NSGA-II for the multi-objective problem to maximize network efficiency and minimize group size in order to obtain the list of critical groups in different size. The steps of NSGA-II can be shown as follows.

- **Initial solution generation**

For destroyed set $\square$, each chromosome is represented by R random 0-1 variables. If the components are selected, then the corresponding bits of chromosome are 1, otherwise, they are 0. From that we can generate an initial solution with the population size of S.

- **Generation of offspring population**

Following Step 1, we generate an offspring of size S from parents' generation using crossover, and mutation operators. The crossover procedure selects randomly from its parents' genes to generate the offspring as represented in Fig. 2. As for the mutation procedure, a gene is randomly selected and changed based on the mutation rate as shown in Fig. 3.

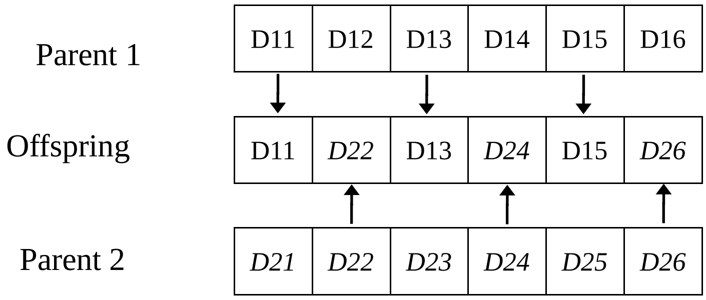


Fig. 2. Uniform crossover of chromosome

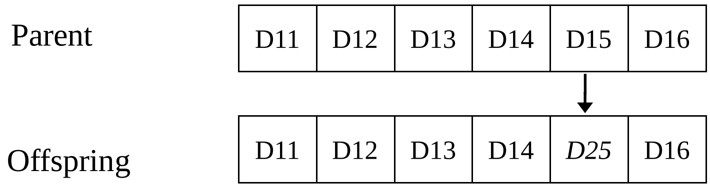


Fig. 3. The mutation mechanism

- **Determination of a new generation**

We first combine parent and offspring population into a set followed by calculating the network efficiency and group size for each individual. The new generation is then selected using the non-dominated ranking approach as shown follows.

Sort the chromosomes using the non-dominated ranking approach and identify fronts F_i*, then calculate the crowding distance of each individual in* F_i.

Set $G = \varnothing, i = 1$*.While* $|G| + |F_i| < S$*, do*

$G = \{G \cup F_i\}$

$i = i + 1$

Sort solutions in F_i *in descending sequence in accordance with crowding distance, then add the first* $M - |G|$ *individuals to* G.

G *is the next generation with size* S.

- **Iteration and stopping criteria**

As the population size and the number of generations depend on specific problem, the iteration is terminated if the stopping criteria are reached.

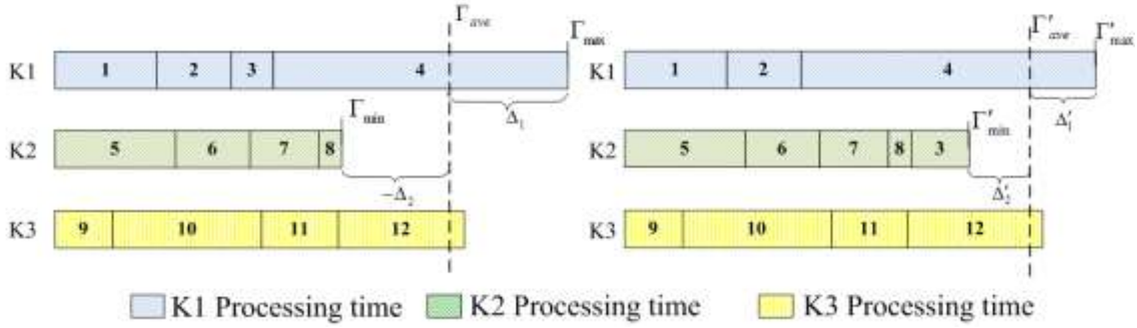
3.2 Algorithm Development of Stage II Model

The Stage II programming is treated as a non-identical parallel machine scheduling problem with components to be repaired by K emergency rescue teams, the difference is that Constraint (20) restricts the makespan to be within rescue time horizon *T*. In what follows, we will discuss about the number of components that could be repaired within the time horizon.

Definition 4: Let l_{up} be the biggest group size of destroyed components to be repaired, which means the number of destroyed components to be repaired by the most efficient rescue team during the time horizon, where the minimal processing time for each destroyed component is $\underset{k\in K}{p'_{rk}} = \min_{k\in K} p_{rk}$, $\forall r : \mathrm{a}_i^r \in \Box$.

Definition 5: Let Γ_{ave} be the optimal makespan if the repairing of each component is assumed to be detachable, so $\Gamma_{\max} = \min\{\Gamma_{ave} + \Delta_k, T\}$, and $\Gamma_{\min} = \Gamma_{ave} - \Delta_k$, $k \in K$.

The relationship among Γ_{ave}, $\Gamma_{\max}$, and $\Gamma_{\min}$ are shown in the following Fig. 4, where an initial solution with $\Gamma_{\max} = \Gamma_{ave} + \Delta_1$, $\Gamma_{\min} = \Gamma_{ave} - \Delta_2$ is represented in Fig. 4a , and $\Gamma_{\max}$ is shortened by reducing Δ_1 to Δ'_1, and Δ_2 to Δ'_2 in Fig. 4b, thus we could get an improved solution $\Gamma'_{\max}$.



4a. Initial solution 4b. Iteration Solution

Fig. 4. The explanation of the heuristic algorithm

Lemma 1: The optimal group size of the critical group satisfies $l^* < l_{up}$.

For each possible group size, we have to choose the critical group to ensure the makespan within time horizon T. Therefore, for a given set of components to be repaired, we propose a heuristic algorithm to search the optimal critical the group size in the available rescue teams from l_{up}.

Lemma 2: Let Γ_{low} be the lower bound of optimal solution, if repairing each component is assumed to be detachable, there exists $\Gamma_{low} = \Gamma_{ave}$.

Proof: For a contradiction, we assume that $\Gamma_{ave} > \Gamma_{low}$ in some cases, which means $\Gamma_{low} = \Gamma_{ave} - \delta$ ($\delta > 0$). If all jobs could be completed within Γ_{low}, then Γ_{low} is the optimal makespan when the repairing of each component is assumed to be detachable, which contracts with Definition 5. For another contradiction, assume $\Gamma_{ave} < \Gamma_{low}$, which means $\Gamma_{low} = \Gamma_{ave} + \delta$ ($\delta > 0$). As δ could be reduced to zero by assigning any parts of jobs to other rescue teams, which contracts with our assumption. Therefore, neither $\Gamma_{ave} > \Gamma_{low}$ nor $\Gamma_{ave} < \Gamma_{low}$ is right, thus we can deduce that $\Gamma_{low} = \Gamma_{ave}$ exists.

Lemma 3: In an optimal solution, there will be $\Gamma_{opt} = \min \Gamma_{\max}$.

Proof: Assume that the optimal solution is $\Gamma_{\max}$, then from Definition 5, we know that $\Gamma_{\max} = \min\{\Gamma_{ave} + \Delta_k, T\}$; meanwhile there also exists another feasible solution $\Gamma'_{\max} = \min\{\Gamma_{ave} + \Delta'_k, T\}$, with $\Delta'_k < \Delta_k$. According to Lemma 2, we have $\Gamma_{ave} \le \Gamma_{opt}$ and hence, $\Delta'_k > \Delta_k > 0$. This implies $\Gamma'_{\max} \ge \Gamma_{\max}$, which contracts with the assumption.

- **How to obtain lower bound** Γ_{low}

To calculate Γ_{low}, we consider a scenario that the repair of each component is detachable. The Stage II model can be transformed into a Linear Programming (LP) in which $z_{rk} \in \{0,1\}$ is replaced by $z_{rk} \geq 0$ ($\forall k, r$). According to Lemma 1, the optimal solution of LP can be seen as the lower bound.

- **How to obtain the optimal solution**

According to Lemma 2, we proposed an algorithm to minimize Δ_k by ordinal assignment and swap between the longest team and the shortest team as shown in Fig. 2b. The detailed steps are described as below.

Step 1: Initial solution
We first find the optimal solution using LP, then the initial solution is obtained by assigning each component to team $k^* = \arg\max_{k \in K} z_{rk}$.
Step 2: Ordinal re-assignment process
Set $B = \varnothing$.
Select the kth *team for which* $\Gamma_k = \Gamma_{\max}$, *and sort components in the* kth *team according to* $|\Gamma_k - \Gamma_{low}|$ *by non-decreasing sequence.*
For team $k' = \arg\min_{k \in K \setminus B} \Gamma_k$
If $B \neq K \setminus \{k\}$, *then*
for any component i *in the* k *th team*
if $\Gamma_{k'} + \tau_{ik'} < \Gamma_{\max}$, *assign the* i *th component to the* $k'th$ *team and return to* ***Step 2****;*
else set $B = B \bigcup \{k'\}$;
Else, go to ***Step 3****.*
Step 3: Ordinal interchange process
Set $B = \varnothing$.
Select the kth *team for which* $\Gamma_k = \Gamma_{\max}$.
For team $k' = \arg\min_{k \in K \setminus B} \Gamma_k$
If $B \neq K \setminus \{k\}$, *then*
for any component i *in the* kth *team*

for any component j *in the* $k'th$ *team*

if $\min\{Gain1 = \tau_{ik} - \tau_{jk}\ ,\ Gain2 = \Gamma_{\max} - (\Gamma_{k'} - \tau_{jk'} + \tau_{ik'})\} > 0$, *then*

exchange (i, j) *and return to* ***Step 2;***

else if no arc pair could be exchanged, set $B = B \cup \{k'\}$ *;*

Else, go to ***Step 4****.*

Step 4: Ending

Set $\Gamma_{\max} = \max_{k \in K}\{\Gamma_k\}$*, end the algorithm.*

- **Computational complexity analysis**

From the above procedure, one can find that the Stage II is an assignment problem with k rescue teams and r jobs, the computational complexity is $O(kr) + O(kr^2) = O(kr^2)$.

Table 1 shows the proposed heuristic algorithm efficiency compared with branch and bound algorithm by CPLEX in running six times. One can see from the result that the time consumption of branch and bound method increases with the size of the problem, while the proposed heuristic algorithm is capable to solve large-size problem efficiently and with the gap less than 3.1%.

Table 1. Comparison with heuristic algorithm and branch and bound algorithm

No. of Teams	No. of Components	No. of Constraints	No. of Variables	CPLEX		Heuristic Algorithm		GAP (%)
				Results	Time (s)	Results	Time (s)	
4	10	14	54	79	0.39	79	0.12	0
4	20	16	64	22	430	22	0.12	0
5	10	15	65	62	0.41	62	0.11	0
5	20	25	125	95	387	98	0.16	3.1
9	50	59	509	42	365	42	0.13	0
10	50	60	560	56	393	56	0.13	0

3.3 Stage I Solution Procedure

Knowing the critical group obtained by the heuristic algorithm in Stage II, the rescue teams are assigned to those components in the critical group. In this section, the sequences for those components to be repaired are sorted in each team using ILOG CPLEX. The objective of Stage I is to minimize the total costs by iterating the repairing sequence in the sub-group.

4. **Case Study**

4.1 Problem Statement

Shanghai is a coastal city in China. It lies on the southeastern frontier of Yangtze Delta, therefore, it contains many rivers, canals, and lakes, and is known for its rich water resources. In recent years, the disaster prone city has experienced natural disasters such as typhoons and floods quite often, which has resulted in considerable damage in lifeline systems. In particular, the scenarios under consideration include lifeline systems of gas, power, and water systems in 4 districts of Fengxian, Jinshan, Nanhui, and Minhang in Shanghai, as shown in Fig. 5.

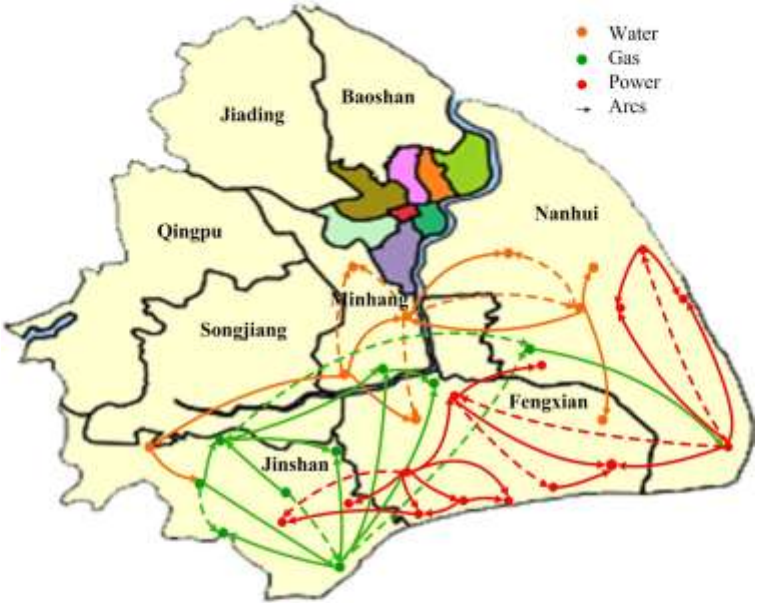


Fig. 5. Scenario based lifeline infrastructures in Shanghai

We consider the key infrastructures in the 4 districts, which consist of 9 gas, 14 power, and 9 water infrastructures and 51 links, which constitute a three-layered network as a whole lifeline system. Each sub-system (water, gas, power) can be seen as a network with three categories of nodes, which are supply, transshipment and demand nodes. For each time period, the demand nodes require certain amount of supply to maintain their normal operations.

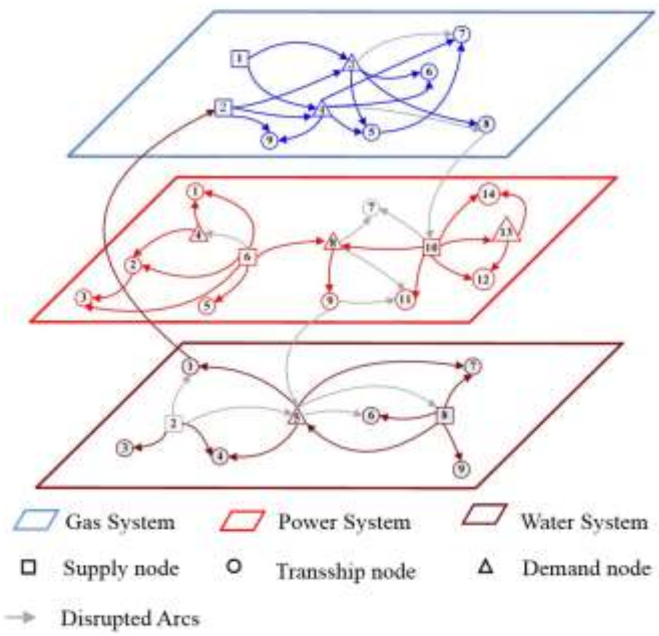


Fig. 6. Disrupted multi-layer lifeline network

Generally speaking, the capacity of the supply nodes is always equal to the demand nodes, and the demand data of nodes of lifeline systems show is in Table 2. Once events happened, there are 12 links disrupted in different functions in the lifeline system, which can be seen in Fig.6. Knowing the task of repairing 16 arcs, the goal of the post-disaster rescue is to minimize the cost under the guarantee of completion time minimized. In this case, the destroyed components $R = 12$, the emergency rescue teams $K = 3$, and the time horizon $T = 24$ hours. The details of those destroyed components are presented in Table 3 and the details of operation cost, and capacity of each arc are presented in Table 4.

Table 2. Demand data of nodes of lifeline systems

Node	Gas System		Power System		Water System	
	Type	Average Demand	Type	Average Demand	Type	Average Demand
1	supply	-150	demand	20	demand	50
2	supply	-100	demand	30	supply	-180
3	supply	0	demand	40	demand	40
4	transit	0	transit	0	demand	110
5	demand	25	demand	35	transit	0
6	demand	35	supply	-250	demand	20
7	demand	40	demand	80	demand	80
8	demand	100	transit	0	supply	-200
9	demand	10	demand	10	demand	50
10	-	-	supply	-150	-	-

11	-	-	demand	30	-	-
12	-	-	demand	50	-	-
13	-	-	transit	25	-	-
14	-	-	demand	75	-	-

Table 3. Data of destroyed arcs

Infrastructure	Source	Sink	Repairing Cost	Upper limit
Gas	1	3	47	70
Gas	2	9	50	10
Gas	3	8	32	45
Gas	4	8	28	70
Power	6	1	63	10
Power	8	9	51	30
Power	10	8	10	80
Power	10	13	4	120
Water	2	1	70	30
Water	5	1	15	40
Water	5	8	20	20
Water	8	6	8.6	40

Table 4. Capacity data of arcs of lifeline systems

Components	Energy System				Power System				Energy System			
	O	E	C	Cons	O	E	C	Cons	O	E	C	Cons
1	1	3	70	70	2	3	10	10	2	1	30	30
2	1	4	80	80	4	1	15	15	2	3	35	50
3	2	3	0	90	4	2	10	20	2	4	40	60
4	2	4	50	70	6	1	5	10	2	5	33	100
5	2	9	8	10	6	4	25	40	5	1	10	40
6	3	7	20	20	6	2	30	30	5	4	60	60
7	3	6	10	25	6	5	35	40	5	6	0	15
8	3	5	0	35	6	8	3	50	5	7	13	15
9	3	8	40	45	6	3	30	30	5	8	0	20
10	4	7	20	20	8	7	70	70	8	5	50	50
11	4	6	25	25	8	9	8	30	8	6	25	40
12	4	5	25	30	8	11	5	30	8	9	55	55
13	4	8	60	70	9	11	0	10	8	7	65	65
14	4	9	0	10	10	8	80	80	-	-	-	-
15	5	7	0	10	10	7	10	25	-	-	-	-
16	-	-	-	-	10	11	25	25	-	-	-	-
17	-	-	-	-	10	14	18	25	-	-	-	-
18	-	-	-	-	10	13	118	120	-	-	-	-
19	-	-	-	-	10	12	0	25	-	-	-	-
20	-	-	-	-	13	14	60	60	-	-	-	-
21	-	-	-	-	13	12	58	60	-	-	-	-

O：Source; E: Sink; C: Cost of each flow; Cons: Flow constraint of each flow.

4.2 Computational Results

According to the procedures of the proposed algorithm, we will demonstrate the 3 steps and the results as follows. Meanwhile, we have tested that the proposed algorithm can solve large-scale problem.

- **Select the critical group**

We first select the critical group in different size from the disrupted components set □ by the NSGA-II described in Section 3.1. The algorithm is coded by Matlab 2013 version with an Intel processor operating at 3.40 GHz and 16 GB RAM, where population size $S = 80$, cross-over rate $\alpha = 0.8$, mutation rate $\beta = 0.2$, and number of generations $G = 20$ respectively. The computational results and improvement rates of the corresponding network efficiency are shown in Table 5, which demonstrates that the network efficiency increases with the group size, therefore, the more components are repaired, the better network efficiency is achievable.

Table 5. The critical group with different size and its efficiency improvement rate

Group Size	Set of Components	Efficiency Improvement (%)
0	/	0
1	7	9.6
2	6, 7	17.4
3	6, 7, 11	21.4
4	2, 3, 6, 7	28.0
5	2, 3, 6, 7, 11	34.0
6	2, 3, 6, 7, 10, 11	36.0
7	2, 3, 6, 7, 10, 11, 12	37.5
8	**1, 2, 3, 6, 7, 10, 11, 12**	**38.7**

According to Lemma 1, we first get that the optimal group size $l^* \leq 8$, then we try if the makespan of the critical group of size 8 is within the time horizon $T = 24$. If the makespan exceeds the time horizon T, we will go to the critical group of size 7 till we find the feasible group size. The computational result demonstrates that the optimal group size $l^* = 8$, with the makespan being 24 hours, the corresponding network efficiency improvement being 38.7%, and the critical group of arc is a set of a_i^r, where the number of arcs

$r = \{1,\ 2,\ 3,\ 6,\ 7,\ 10,\ 11,\ 12\}$.

- **Allocate rescue teams**

We employ the proposed heuristic algorithm in **Stage II** model for the allocation problem with 8 critical groups and 3 rescue teams, which can be solved by Matlab with the computational time of 0.62s. Based on the processing times matrix shown in Table 6, the optimal allocation plan assigned to each team is: $K1$: {2, 7, 11, 12}, $K2$: {3, 6}, $K3$: {1, 10}.

Table 6. The data of processing time for each rescue team (unit: hours)

Team \ Arc	1	2	3	4	5	6	7	8	9	10	11	12
K1	15	9	6	5	12	6	3	9	6	3	6	6
K2	23	17	12	17	18	12	7	15	9	7	15	18
K3	15	9	8	12	14	9	6	10	11	6	13	12

- **Optimize the total costs**

Obtained the assignment plan of each team, we then minimize the total costs by adjustment of the operating sequence using **Stage I** model by the CPLEX solver. The optimal cost equal to $1,400,351 in total, and the computational time is 2.02s. The final processing sequence is as shown in Fig. 7.

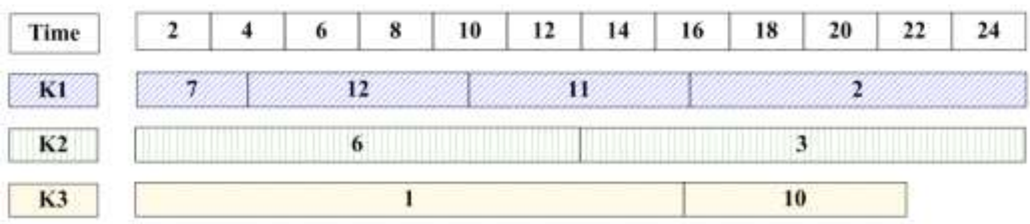


Fig. 7. The optimal scheduling of the rescue teams

- **Test for the algorithm efficiency**

We also test our network efficiency by several scales of network by parallel computing. The scale of the problem can be determined by the number of nodes, the number of arcs, the number of destroyed components of the whole lifeline network and the number of rescue teams. In parallel computing, as the computational complexity of arc selection is $O(P^2)$, where P means the size of population size in NSGA-II, and the computational complexity of stage II is $O(kr)+O(kr^2)=O(kr^2)$, therefore the maximal computational time of the three steps is constrained by the Branch and Bound in Stage I. The computational results under different problem scales can be seen in Table 7 and Fig. 8.

Table 7. The computational time under different scales of problem (unit: second)

No. of Scale	Scale ($N \times A \times R \times K$)	No. of Constraints	No. of Variables	Computational Time(s)
1	100*160*20*5	1549060	9121	7.69
2	100*160*20*10	1551580	11521	7.97
3	200*320*30*5	6936330	16801	12.08
4	200*320*30*10	6940050	20404	17.08
5	300*480*40*5	23085200	26881	37.7
6	300*480*40*10	23090120	31681	38.08
7	400*640*50*5	38446870	32161	61.65
8	400*640*50*10	38452990	38161	119.29
9	500*800*60*5	69178140	39841	186.28
10	500*800*60*10	69185460	47041	237.5
11	600*960*70*5	112965410	47521	497.05
12	600*960*70*10	112973930	55921	542.12

The biggest network scale the proposed method can solve is 400 nodes with 640 arcs as the computational time begins to increase exponentially from the scale 400*640*50*5, which can be seen from Fig. 8.

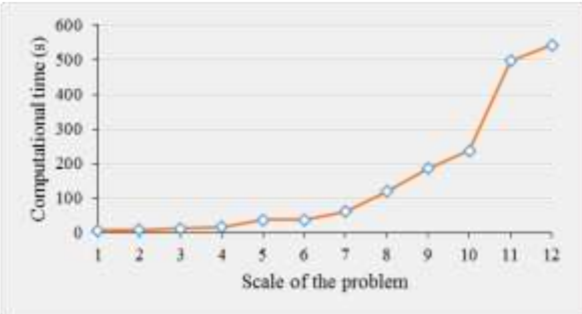


Fig.8. The trend of computational time under different scales of the problem

5. Conclusions and Future Work

The study has addressed a resource allocation and schedule problem to optimally restore the most critical components in the interdependent lifeline systems under disruptions. A general two-stage programming model has been developed, where Stage I programming aims to minimize total losses during the time horizon, while stage II programming targets to restore system functions as quickly as possible. To solve the two-stage programming, the critical group selection approaches are designed to maximize the whole network efficiency firstly. Then, a heuristic algorithm has been presented to determine the rescue teams' assignment scheme based on the tasks/components selection process by vulnerability analysis, and the optimal repairing sequence is determined in order to minimize the total cost. The problem under study has several unique features over previous research: (i) the model provides metrics of the whole network and fully integrates the vulnerability analysis into the restoration strategy; (ii) the two-stage model takes time priority strategy between total losses and the restoration time under time sensitive scenarios; (iii) a solution procedure is developed to solve the rescue problem and capable to be applied into real cases. Emergency response stakeholders could therefore optimize the resource allocation and scientifically organize the rescue procedure, which will greatly improve the capability to respond to emergencies that can disrupt the lifeline services. Furthermore, the network vulnerability analysis and resource allocation model could be extended and applied to the lifeline system protection strategies.

According to the computational results, the network scale the proposed method can solve is 400 nodes with 640 arcs, which is enough for the 4 districts in the Case Study. To solve larger scale problems, the whole network can be separated into sub-networks with smaller scale as the infrastructure connections among districts are usually weak, then it can be solved by the proposed method. Further, the proposed approaches under study are only considered determined physical status of infrastructures standing on the view of rescue teams. It cannot work if we consider from the perspective of top decision makers. In the potential future work, we may address the dynamic response strategies from the view of top decision makers considering the very complex scenarios.

REFERENCES

Aven, T., 2011. On some recent definitions and analysis frameworks for risk, vulnerability, and resilience. Risk Analysis, 31(4), 515-522.

Alguacil, N., Arroyo, J.M., & Carrión, M., 2010. Transmission network expansion planning under deliberate outages. In Handbook of Power Systems, 365-389.

Azaiez, M. N., & Bier, V. M., 2007. Optimal resource allocation for security in reliability systems. European Journal of Operational Research 181(2), 773-786.

Barbarosoğlu, G., 2004. A two-stage stochastic programming framework for transportation planning in disaster response. Journal of the Operational Research Society, 55(1), 43-53.

Brown, Gerald G., Carlyle, W. Matthew., Salmerón, Javier., Wood, Kevin., 2005. Analyzing the Vulnerability of Critical Infrastructure to Attack, and Planning Defenses. Tutorials in Operations Research, INFORMS, 102-123.

Cavdaroglu, B., Hammel, E., Mitchell, J.E., Sharkey, T.C., & Wallace, W.A., 2013. Integrating restoration and scheduling decisions for disrupted interdependent infrastructure systems. Annals of Operations Research 203(1), 279-294.

Chai, C.-L., Liu, X., Zhang, W.J., Baber, Z., 2011. Application of social network theory to prioritizing Oil & Gas industries protection in a networked critical infrastructure system. Journal of Loss Prevention in the Process Industries 24(5), 688–694.

Collier, S., & Lakoff, A., 2008. The vulnerability of vital systems: How 'critical infrastructure' became a security problem. The Politics of Securing the Homeland: Critical Infrastructure, Risk and Securitisation, 40-62.

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 2,182-197.

De Sherbinin, A., Schiller, A., & Pulsipher, A., 2007. The vulnerability of global cities to climate hazards. Environment and Urbanization, 19(1), 39-64.

Faraj, S., Xiao, Y., 2006. Coordination in fast-response organizations. Management Science 52, 1155–1169.

Fiedrich, F., Gehbauer, F., & Rickers, U., 2000. Optimized resource allocation for emergency response after earthquake disasters. Safety Science 35, 41-57.

Garg, Manish., Smith, J. Cole., 2008. Models and algorithms for the design of survivable multicommodity flow networks with general failure scenarios. Omega 36(6), 1057–1071.

Hausken, K., 2011. Protecting complex infrastructures against multiple strategic attackers. International Journal of Systems Science 42(1), 11-29.

Kamissoko D, Zaraté P, Pérès F., 2014. Decision process in large-scale crisis management. Environment Systems and Decisions 34(2): 277-287.

Lee, E.E., Mitchell, J.E., & Wallace, W.A., 2007. Restoration of services in interdependent infrastructure systems: A network flows approach.Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 37(6), 1303-1317.

Lin, Y.K., Pfund, M. E., & Fowler, J. W., 2011. Heuristics for minimizing regular performance measures in unrelated parallel machine scheduling problems. Computers & Operations Research, 38(6), 901-916.

Murray, A.T., Matisziw, T.C., & Grubesic, T.H., 2007. Critical network infrastructure analysis: interdiction and system flow. Journal of Geographical Systems 9(2), 103-117.

Murray, A.T., 2013. An overview of network vulnerability modeling approaches. GeoJournal 78(2), 209-221.

Rocco, C.M., Ramirez-Marquez, J.E., Salazar, D.E., Yajure, C., 2011. Assessing the Vulnerability of a Power System Through a Multiple Objective Contingency Screening Approach. IEEE Transactions on Reliability 60(2), 394-403.

Samuel, Andrew., Guikema, Seth D., 2012. Resource Allocation for Homeland Defense: Dealing with the Team Effect. Decision Analysis 9, 238-252.

Shan, Siqing., Wang, Li., Li, Ling., 2012. The Assessment of Information Technology Maturity in Emergency Response Organizations. Information Technology and Management, Online first.

San Martin, P. A., 2007. Tri-level optimization models to defend critical infrastructure. Naval Postgraduate School, Monterey, CA.

Scaparra, Maria P., Church, Richard L., 2008. A bilevel mixed-integer program for critical infrastructure protection planning. Computers & Operations Research 35, 1905–1923

Shen, S., 2013. Optimizing designs and operations of a single network or multiple interdependent infrastructures under stochastic arc disruption. Computers & Operations Research 40(11), 2677-2688.

Sorokin, A., Boginski, V., Nahapetyan, A., & Pardalos, P. M., 2013. Computational risk management techniques for fixed charge network flow problems with uncertain arc failures. Journal of Combinatorial Optimization 25(1), 99-122.

Su, L. H., & Lien, C. Y., 2009. Scheduling parallel machines with resource-dependent processing times. International Journal of Production Economics 117(2), 256-266.

Turner, B. L., Kasperson, R. E., Matson, P. A., McCarthy, J. J., Corell, R. W., Christensen, L., Eckley, N., Kasperson, J. X., Luers, A., Martello, M. L., Polsky, C., Pulsipher, A., & Schiller, A., 2003. A framework for vulnerability analysis in sustainability science. Proceedings of the National Academy of Sciences 100, 8074-8079.

Turner II, B. L., 2010. Vulnerability and resilience: Coalescing or paralleling approaches for sustainability science. Global Environmental Change 20, 570-576.

Wex F., Schryen G., Feuerriegel S., Neumann D., 2014. Emergency response in natural disaster management: Allocation and scheduling of rescue units. European Journal of Operational Research 235, 697–708

Yatesa, Justin., Battab, Rajan., Karwanb, Mark., Casasc. Irene., 2012. Establishing public policy to protect critical infrastructure: Finding a balance between exposure and cost in Los Angeles County. Transport Pollicy 24, 109-117.

Yan, S., Shih, Y.L., 2009. Optimal emergency roadway repair and subsequent relief distribution . Computers and Operations Research 36 (6), 2049–2065.

Yeh, W. C., Lai, P. J., Lee, W. C., & Chuang, M. C., 2013. Parallel-machine scheduling to minimize makespan with fuzzy processing times and learning effects. Information Sciences.

Rocco, C.M., Ramirez-Marquez, J.E., Salazar, D. E., & Yajure, C., 2011. Assessing the vulnerability of a power system through a multiple objective contingency screening approach. Reliability. IEEE Transactions on, 60(2), 394-403.

Srdjevic, B., Srdjevic, Z., Blagojevic, B., & Suvocarev, K. 2013. A two-phase algorithm for consensus building in AHP-group decision making. Applied Mathematical Modelling, 37(10), 6670-6682.

Zhang, Pengcheng., Peeta, Srinivas., 2011. A generalized modeling framework to analyze interdependencies among infrastructure systems. Transportation Research Part B: Methodological 45(3), 553–579.

Zio, E., Golea, L. R., & Rocco S, C. M., 2012. Identifying groups of critical edges in a realistic electrical network by multi-objective genetic algorithms. Reliability Engineering & System Safety 99, 172-177