Computers & Industrial Engineering 164 (2022) 107866

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

# Sequential recovery of disrupted facilities

Hongwei Xuan, Chao Zhang, Yugang Yu, Lindong Liu *

*International Institute of Finance, School of Management, University of Science and Technology of China, Hefei, Anhui 230026, PR China*

ARTICLE INFO

*Keywords:*
Sequential disruption recovery
Uncapacitated and capacitated facility location
Integer linear program
Lagrangian relaxation
Greedy method

ABSTRACT

For a bipartite graph with disrupted facilities, we confront the key concern that disrupted facilities can only be recovered one after one due to resource shortage. The objective of this paper is to design a sequential recovery schedule such that the summation of waiting time and transportation cost of all customers to regain service is minimized. For such a sequential disruption recovery problem, we consider two situations: sequential disruption recovery for uncapacitated facilities (SDRUF) and sequential disruption recovery for capacitated facilities (SDRCF). We model the corresponding optimization problems as integer linear programs, and propose two heuristic algorithms for each situation, based on both Lagrangian relaxation and greedy method, to generate feasible solutions. In the end, computational experiments are carried out to implement the proposed heuristic algorithms. The computational results show that the two heuristic algorithms are efficient and effective for solving the SDRUF and the SDRCF problems, and act as good alternatives for each other based on problem size.

## 1. Introduction

With the rapid growth of economy and technology, the threat of large-scale failures, in terms of components and users, also increases. For instance, in 2018, a 7.4-magnitude earthquake occurred in Sulawesi Province in central Indonesia and triggered a tsunami, the wireless communication facilities in the five worst-hit areas were severely damaged, including Palu, Tobali, Pamona, Pendolo and Sumber sari. However, due to the shortage of local rescue materials and recovery professionals, communication in only one of the five worst-hit areas can be recovered at a time. In this case, how to organize the recovery of communication facilities is vital. Disaster like the earthquake happened in Indonesia is not rare, according to the International Federation of Red Cross and Red Crescent Societies(IFRC), more than 2800 disasters took place during 2010 and 2019. Meanwhile, the estimation from the IFRC shows that there were a loss of 1.92 trillion US dollars, more than 1.7 billion affected people during this time (see IFRC, 2020). These statistics indicate the importance of developing response strategies to reduce the impact of disasters for humankind.

In the literatures, two main strategies (i.e., proactive planning and aftermath rescheduling) have been proposed to ease off the negative impacts of unexpected disruptions. For proactive planning strategies, Brown, Carlyle, Salmerón, and Wood (2006) gathered open-source data on a real-world infrastructure system, and used them to plan an optimal defense. Jia, Ordonez, and Dessouky (2007) first considered Large-scale emergencies, which was added to three traditional models to optimize the locations of facilities for medical supplies. Cui, Ouyang, and Shen (2010) developed a mixed-integer program formulation and a continuum approximation model that considered unexpected failures to study the reliable uncapacitated fixed-charge location problem. Two heuristic algorithms for solving these problems were designed: Liberatore, Scaparra, and Daskin (2011) developed a max-covering type formulation for the stochastic R-interdiction median problem with fortification, which optimally allocates defensive resources among facilities to minimize the worst-case impact of an intentional disruption. Besides, Losada, Scaparra, and OHanley (2012) presented a mixed-integer linear program for protecting an uncapacitated median-type facility network against worst-case losses. Li, Zeng, and Savachkin (2013) presented the integer nonlinear program models of a reliable P-median and a reliable uncapacitated fixed-charge location problem. Bao, Zhang, Ouyang, and Miao (2019) proposed a tri-level model explicitly integrating the decision making on recovery strategies of disrupted facilities with the decision making on protecting facilities from intentional attacks.

For aftermath rescheduling strategies, Yu and Qi (2004) developed a disruption model for the airline scheduling problem and designed a system based on heuristic algorithms to solve the model. Walker, Snowdon, and Ryan (2005) developed an optimization model to resolve disruptions to an operating schedule in the rail industry. Tomlin (2006) studied two strategies to manage the risk of disruptions, including mitigation and contingency tactics. Furthermore, Xiao, Qi, and Yu

* Corresponding author.
*E-mail addresses:* hwxuan@mail.ustc.edu.cn (H. Xuan), zc123456@mail.ustc.edu.cn (C. Zhang), ygyu@ustc.edu.cn (Y. Yu), ldliu@ustc.edu.cn (L. Liu).

https://doi.org/10.1016/j.cie.2021.107866
Received 2 January 2021; Received in revised form 29 June 2021; Accepted 3 December 2021
Available online 9 December 2021

(2007) studied the coordination of supply-chain management when disruptions occurred and proposed a method to improve the robustness of an initial plan. Cauvin, Ferrarini, and Tranvouez (2009) proposed an approach based on cooperative distributed problem-solving to minimize the impact of disrupting events on the whole system. Hishamuddin, Sarker, and Essam (2013) explored a disruption recovery model for a two-stage production and inventory system with the possibility of transportation disruption. Paul, Sarker, and Essam (2015) developed a disruption recovery model for an imperfect single-stage production-inventory system, and rescheduling the production plan, after the occurrence of a disruption. Ivanov (2019) studied production-ordering behaviour in a supply chain with disruption risks in recovery and post-disruption periods and the influence of severe disruptions on production and distribution network design.

Regarding to facility location, we review some existing literatures as follows. The classic facility location problem originated with the work of Burstall, Leaver, and Sussams (1962), based on which the problem can be divided into the uncapacitated facility location (UFL) problem and the capacitated facility location (CFL) problem. Following Burstall et al. (1962), Erlenkotter (1978) developed and tested a method for the uncapacitated facility location problem based on a linear program dual farmation, and Van Roy (1986) carried out an implementation of the cross decomposition method to solve the capacitated facility location problem. There are also many other research branches based on facility location problem, e.g., the dynamic facility location problem, where locations are permitted change within a planning horizon of $r$ periods (see Wesolowsky, 1973); the sequential facility location problem, where there are already a number of facilities at various locations on a network, providing identical service, the key decision for the central authority to make is to find suitable locations for the next facility and to determine which demand points will be served by which facility (see Oded & Celik, 1982).

This paper is within scope of aftermath rescheduling and sequential facility location. However, our study differs from existing works in a combined way of the following three points. First, we focus on the disruption of facility sites rather than network arcs. Second, our facility recovery is sequential, not unordered. Thus, our recovery decisions change sequentially over time. Third, the evaluation standard for service levels of all customers is the summation of the total weighted waiting time and the total transportation cost of the customers to regain service. To be specific, we focus on a sequential disruption recovery problem, where there are a set of facilities from which a set of customers could get services. When disasters happen, some or all of the facilities are damaged. Then, due to the shortage of materials and manpower, only one disrupted facility can be recovered at a time. Since the recovery of the failed facilities takes time and resources, following different recovery plans results in different service levels for the whole customers. In this case, we are aiming to find out a recovery schedule such that the summation of weighted waiting time and transportation cost to regain services for all customers is minimized.

The main contributions of this study can be summarized as follows. First, we successfully formulate the sequential recovery problems of disrupted facilities as integer linear programs (ILPs). Second, we propose two types of heuristic algorithms for the sequential disruption recovery of either uncapacitated or capacitated case. Third, we demonstrate our algorithms by conducting some numerical experiments, and show that both of them are efficient and effective. More importantly, when the scales of problem are different, both heuristic algorithms are good alternatives for each other, better results can be obtained.

The remainder of this paper is organized as follows. In Section 2, we introduce the sequential disruption recovery for uncapacitated facilities (SDRUF), and propose two heuristic algorithms. In Section 3, we focus on the sequential disruption recovery for capacitated facilities (SDRCF). Section 4 presents the computational results from implementing the proposed algorithms on the SDRUF and the SDRCF problems. Section 5 summarizes our findings and provides directions for future research.

## 2. Sequential disruption recovery for uncapacitated facilities

In this section, we first study the sequential disruption recovery problem for uncapacitated facilities, where the disrupted facilities are uncapacitated after recovery and the transportation cost on arcs are identically equal to 1. In this case, under a given sequence, the facilities are recovered one after one. Once a facility is recovered, all connected and unserved customers will be served immediately. Our objective is to minimize the total weighted waiting time of the customers. We now illustrate the SDRUF problem via a small example.

**Example 1.** As shown in Fig. 1, there are three facilities, $\{A,B,C\}$, and four customers, $\{1,2,3,4\}$. The recovery time of each facility is $t_A = 1$, $t_B = 2$, and $t_C = 3$, respectively. The weight of each customer is $w_1 = 4$, $w_2 = 3, w_3 = 1$, and $w_4 = 2$, respectively. All of the unit transportation cost are equal to 1. The total weighted waiting time of customers associated with different recovery sequences is $\widehat{t}_{A\to B\to C} = 22, \widehat{t}_{A\to C\to B} = 21$, $\widehat{t}_{B\to A\to C} = 32, \widehat{t}_{B\to C\to A} = 42, \widehat{t}_{C\to A\to B} = 37, \widehat{t}_{C\to B\to A} = 44$, respectively. It is then direct that the optimal recovery sequence is $A\to C\to B$. When $A$ is recovered, customers 1 and 2 are served. When $C$ is recovered, customer 4 is served. When $B$ is recovered, customer 3 is served.

### 2.1. Model formulation

To formally model the SDRUF problem, we next list notations as in Table 1, based on which an integer linear program of the SDRUF problem can be modeled as follows.

***Objective***. The transportation cost for each arc equals 1. Thus, our objective is equivalent to find an optimal recovery sequence to minimize the total weighted waiting time for all the customers:

$$\min \sum_{j=1}^{n} w_j d_j \tag{1}$$

***Recovery-sequence constraints***. A recovery sequence is an arbitrary order of numbers $\{1, 2, \ldots, m\}$. Each facility is recovered for once and only once. Hence, the constraints on the recovery sequence are

$$\sum_{k=1}^{m} x_{ki} = 1, \forall i \in M, \tag{2}$$

$$\sum_{i=1}^{m} x_{ki} = 1, \forall k \in M. \tag{3}$$

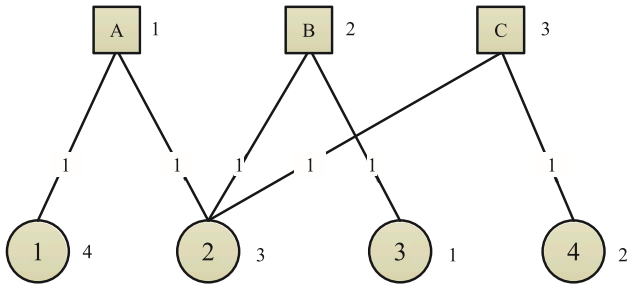


**Fig. 1.** Example of a SDRUF problem.

**Table 1**
Notation used in the SDRUF problem.

| | |
|---|---|
| $G$ | The bipartite graph, $G = \{M, N, E\}$ |
| $M$ | The set of facilities, $M = \{1,2,\ldots,m\}$, where $m$ is the total number of (disrupted) facilities |
| $N$ | The set of customers, $N = \{1,2,\ldots,n\}$, where $n$ is the total number of customers |
| $E$ | The set of arcs, $E = \{1,2,\ldots,z\}$, where $z$ is the total number of arcs |
| $w_j$ | The weight of customer $j, \forall j \in N$ |
| $d_j$ | The waiting time of customer $j$ before being served, $\forall j \in N$ |
| $c_k$ | The completion time of the $k$th recovered facility, $\forall k \in M$ |
| $c_{ij}$ | The unit transportation cost from facility $i$ to customer $j, \forall i \in M, \forall j \in N$ |
| $t_i$ | The recovery time of facility $i, \forall i \in M$ |
| $\widehat{t}$ | The total time of recovering all the disrupted facilities, $\widehat{t} = \sum_{i=1}^{m} t_i$ |
| $g_{ij}$ | The connection stamp. Let $g_{ij} = 1$, if facility $i$ and customer $j$ are connected; $g_{ij} = 0$, otherwise, $\forall i \in M, \forall j \in N$ |
| $y_{kj}$ | The service stamp. Let $y_{kj} = 1$, if the $k$th recovery is completed, customer $j$ is immediately served; $y_{kj} = 0$, otherwise, $\forall k \in M, \forall j \in N$ |
| $x_{ki}$ | Decision variable. Let $x_{ki} = 1$, if the $k$th recovered facility is $i$; $x_{ki} = 0$, otherwise, $\forall k \in M, \forall i \in M$ |

***Service-mapping constraints***. Given any recovery sequence, once a facility is recovered, the unserved customers connected to this facility will be immediately served. According to the bipartite graph $G$ and the recovery sequence matrix $X$, the service-mapping matrix $Y$ can be determined by the following constraints.

$$\sum_{k=1}^{m} y_{kj} = 1, \forall j \in N, \tag{4}$$

$$y_{kj} \geqslant \sum_{i=1}^{m} \left( x_{ki} - \sum_{l=1}^{k-1} x_{li} \right) g_{ij}, \forall k \in M, \forall j \in N. \tag{5}$$

Essentially, constraints (4) indicate that each customer can be served by one and only one facility. Constraints (5) ensure that each customer is served by the first recovered facility that he is connected to.

**Lemma 1**. *For a given bipartite graph G, and a recovery sequence matrix X, the service-mapping matrix Y is determined by constraints (4) and (5).*

The service-mapping constraints are the key part of our model. The correctness of constraints (5) is not straightforward. We show the proof of Lemma 1 in Appendix A.1.

***Waiting-time constraints***. Given the matrix $X$ and $Y$, one can obtain the constraints of the completion time of each facility and the waiting time of each customer, respectively.

$$c_k = \sum_{l=1}^{k} \sum_{i=1}^{m} x_{li} t_i, \forall k \in M, \tag{6}$$

$$d_j \geqslant c_k - c_m (1 - y_{kj}), \forall k \in M, \forall j \in N. \tag{7}$$

For constraints (6), it is clear that term $\sum_{i=1}^{m} x_{ki} t_i$ denotes the recovery time of the $k$th recovered facility, and $c_m$ denotes the total completion time of recovering all of the disrupted facilities, which is equal to $\sum_{l=1}^{m} \sum_{i=1}^{m} x_{li} t_i$. According to constraints (7), if $y_{kj} = 1$, then $d_j \geqslant c_k$, otherwise, $d_j \geqslant c_k - c_m$. By noting that the objective function is $\min \sum_{j=1}^{n} w_j d_j$, so we can claim that $d_j = c_k$ when $y_{kj} = 1$.

***Binary constraints***. The decision variables, $x_{ki}$ and $y_{kj}$, are binary in the SDRUF problem.

$$x_{ki}, y_{kj} \in \{0,1\}, \forall k \in M, \forall i \in M, \forall j \in N. \tag{8}$$

To summerize, we can derive the mathematical formulation of the SDRUF problem as follows:

$$\begin{aligned}
\mathcal{LP}_1 \ :=\ & \min \sum_{j=1}^{n} w_j d_j \\
& s.t. \ \sum_{k=1}^{m} x_{ki} = 1, \forall i \in M, \\
& \sum_{i=1}^{m} x_{ki} = 1, \forall k \in M, \\
& \sum_{k=1}^{m} y_{kj} = 1, \forall j \in N, \\
& y_{kj} \geqslant \sum_{i=1}^{m} \left( x_{ki} - \sum_{l=1}^{k-1} x_{li} \right) g_{ij}, \forall k \in M, \forall j \in N, \\
& d_j \geqslant \sum_{l=1}^{k} \sum_{i=1}^{m} x_{li} t_i - c_m (1 - y_{kj}), \forall k \in M, \forall j \in N, \\
& x_{ki}, y_{kj} \in \{0,1\}, \forall k \in M, \forall i \in M, \forall j \in N.
\end{aligned}$$

It is clear that $\mathcal{LP}_1$ is an ILP with integer decision variables. We now propose two heuristic algorithms to solve the SDRUF problem in Sections 2.2 and 2.3.

### 2.2. Lagrangian heuristic algorithm I

The well-known Lagrangian relaxation is a very effective method in solving integer optimization problems, which is firstly applied by Held and Karp (1970) to solve the travelling salesman problem. In this subsection, we describe the framework of applying the Lagrangian relaxation procedure to generate heuristic algorithm, the Lagrangian heuristic algorithm I, to computing feasible solutions for the SDRUF problem.

***Lagrangian dual formulation***. Based on the $\mathcal{LP}_1$, we can relax constraints (5) and (7) to the objective function by using Lagrangian multipliers $\lambda$ and $\mu$. The resulting Lagrangian function, $\mathcal{L}$, is given by

$$\begin{aligned}
\mathcal{L} \ :=\ & \min \left[ \sum_{j=1}^{n} \left( w_j - \sum_{k=1}^{m} \mu_{kj} \right) d_j + \sum_{k=1}^{m} \sum_{j=1}^{n} (c_m \mu_{kj} - \lambda_{kj}) y_{kj} - \sum_{k=1}^{m} \sum_{j=1}^{n} c_m \mu_{kj} \right. \\
& \left. + \sum_{k=1}^{m} \sum_{i=1}^{m} \left( \sum_{j=1}^{n} \lambda_{kj} g_{ij} + \sum_{j=1}^{n} \sum_{h=k}^{m} \mu_{hj} t_i - \sum_{j=1}^{n} \sum_{h=k+1}^{m} g_{ij} \lambda_{hj} \right) x_{ki} \right] \\
& s.t. \quad (2), (3), (4) \text{ and } (8).
\end{aligned} \tag{9}$$

The correctness of $\mathcal{L}$ is implied by the following Lemma 2, and the proof is shown in Appendix A.2.

**Lemma 2**. *The following two terms are equivalent:*

$$\sum_{k=1}^{m} \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{l=1}^{k} t_i \mu_{kj} x_{li} = \sum_{k=1}^{m} \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{h=k}^{m} t_i \mu_{hj} x_{ki}. \tag{10}$$

For non-negative Lagrangian multipliers $\lambda$ and $\mu$, $\mathcal{L}(\lambda, \mu)$ provides a lower bound (LB) on the objective function value of $\mathcal{LP}_1$. To derive the sharpest lower bound, we need to optimize the Lagrangian dual problem:

$$\begin{aligned}
\mathcal{L}^* \ :=\ & \max_{\lambda \geqslant 0, \mu \geqslant 0} \mathcal{L}(\lambda, \mu) \\
& s.t. \ (2), (3), (4) \text{ and } (8).
\end{aligned} \tag{11}$$

***Problem decomposition***. Before solving the Lagrangian dual problem, $\mathcal{L}^*$, we first compute the value of $\mathcal{L}(\lambda, \mu)$ under any given $\lambda$ and $\mu$. To this end, we decompose $\mathcal{L}(\lambda, \mu)$ into four subproblems, i.e., $\mathcal{L}(\lambda, \mu) = \mathcal{L}_1(\lambda, \mu) + \mathcal{L}_2(\lambda, \mu) + \mathcal{L}_3(\lambda, \mu) + \mathcal{L}_4(\lambda, \mu)$. Next, we illustrate these four problems in detail.

***Subproblem one***: waiting-time constrained problem.

$$\mathcal{L}_1 \ :=\ \min \sum_{j=1}^{n} \left( w_j - \sum_{k=1}^{m} \mu_{kj} \right) d_j$$

To sharpen the Lagrangian lower bound, we additionally impose

some linear constraints to $\mathcal{L}_1$. Note that since these constraints are redundant to $\mathcal{L}(\lambda,\mu)$, the feasibility of the resulting solutions still holds. These constraints are given in Lemmas 3 and 4. Lemma 3 shows that the lower bound of $d_j$ is the minimal construction time among the facilities connected to $j$, and the proof of Lemma 4 is shown in Appendix A.3.

**Lemma 3.** *A lower bound of the waiting time for each customer, $j \in N$, is given by*

$$d_j \geqslant \min\left\{t_k g_{kj} + H\left(1-g_{kj}\right) : H \to +\infty, \forall k \in M\right\}, \forall j \in N. \tag{12}$$

**Lemma 4.** *An upper bound of the waiting time for each customer, $j \in N$, is given by*

$$d_j \leqslant \widehat{t} - \sum_{k=1}^{m} t_k g_{kj} + \max\left\{t_k g_{kj} : \forall k \in M\right\}, \forall j \in N. \tag{13}$$

According to the lower and upper bounds of $\{d_j : \forall j \in N\}$, the waiting-time constrained problem can be updated as the following LP, and we have Algorithm 2.1 to solve $\mathcal{L}_1$.

$$\begin{aligned} \mathcal{L}_1 \ := \ & \min \sum_{j=1}^{n}\left(w_j - \sum_{k=1}^{m}\mu_{kj}\right)d_j \\ & s.t. \ (12) \text{ and } (13). \end{aligned}$$

**Algorithm 2.1.** The specific steps of solving $\mathcal{L}_1$ are shown as follows:

**Step 1:** Given $t_k, w_j, \mu_{kj}, g_{kj}, \forall k \in M, \forall j \in N$.

**Step 2:** Let $j = 1$, if $w_j - \sum_{k=1}^{m}\mu_{kj} \leqslant 0$, let $d_j = \widehat{t} - \sum_{k=1}^{m} t_k g_{kj} + \max\left\{t_k g_{kj} : \forall k \in M\right\}$, otherwise, let $d_j = \min\left\{t_k g_{kj} + H(1-g_{kj}) : H \to +\infty, \forall k \in M\right\}$.

**Step 3:** Let $j = j+1$, repeat Step 2 until $j = n$.

***Subproblem two***: service-mapping stamps problem.

$$\begin{aligned} \mathcal{L}_2 \ := \ & \min \sum_{k=1}^{m}\sum_{j=1}^{n}\left(c_m\mu_{kj} - \lambda_{kj}\right)y_{kj} \\ & s.t. \ \sum_{k=1}^{m} y_{kj} = 1, \forall j \in N, \\ & y_{kj} \in \{0,1\}, \forall k \in M, j \in N. \end{aligned}$$

According to the expression of $\mathcal{L}_2$, there is a unique "1" in each column of matrix $Y$. In this case, we have the following Algorithm 2.2 to optimally solve the service-mapping stamps problem.

**Algorithm 2.2.** The specific steps of solving $\mathcal{L}_2$ are shown as follows:

**Step 1:** Given $c_m, \mu_{kj}, \lambda_{kj}, \forall k \in M, \forall j \in N$.

**Step 2:** Let $j = 1, y_{k'j} = 1$, where $k' = \arg_k \min\left\{c_m\mu_{kj} - \lambda_{kj} : \forall k \in M\right\}$, and $y_{kj} = 0, \forall k \in M'$, where $M' = \{1,2,\ldots k'-1, k'+1, \ldots, m\}$.

**Step 3:** Let $j = j+1$, repeat Step 2 until $j = n$.

***Subproblem three***: recovery-sequence problem.

$$\begin{aligned} \mathcal{L}_3 \ := \ & \min \sum_{k=1}^{m}\sum_{i=1}^{m}\left(\sum_{j=1}^{n}\lambda_{kj}g_{ij} + \sum_{j=1}^{n}\sum_{h=k}^{m}\mu_{hj}t_i - \sum_{j=1}^{n}\sum_{h=k+1}^{m} g_{ij}\lambda_{hj}\right)x_{ki} \\ & s.t. \ \sum_{k=1}^{m} x_{ki} = 1, \forall i \in M, \\ & \sum_{i=1}^{m} x_{ki} = 1, \forall k \in M, \\ & x_{ki} \in \{0,1\}, \forall k \in M, \forall i \in M. \end{aligned}$$

The recovery sequence problem is equivalent to an assignment problem, with $k \in M$ being the tasks, $i \in M$ being the agents, and term $\sum_{j=1}^{n}\lambda_{kj}g_{ij} + \sum_{j=1}^{n}\sum_{h=k}^{m}\mu_{hj}t_i - \sum_{j=1}^{n}\sum_{h=k+1}^{m} g_{ij}\lambda_{hj}$ being the cost to assign task $k$ to agent $i$. We know that an assignment problem can be optimally solved in polynomial time by the *Hungarian algorithm* (see Kuhn, 1955).

***Subproblem four***: constant-value problem.

$$\mathcal{L}_4 \ := \ \min \sum_{k=1}^{m}\sum_{j=1}^{n} -c_m\mu_{kj}$$

Given the Lagrangian multiplier, $\mu$, the value of this subproblem is constant.

***The subgradient method***. To find the optimal Lagrangian multipliers, $\lambda^*$ and $\mu^*$, to maximize the lower bound $\mathcal{L}(\lambda,\mu)$, we apply the subgradient method (see Fisher, 1981) to update $\{\lambda_{kj}, \mu_{kj}\}$ at each iteration, where

$$\lambda_{kj}^{v+1} = \left\{\lambda_{kj}^{v} + \alpha^{v}\left[\sum_{i=1}^{m}\left(x_{ki}^{v} - \sum_{l=1}^{k-1}x_{li}^{v}\right)g_{ij} - y_{kj}^{v}\right]\right\}^{+},$$

$$\mu_{kj}^{v+1} = \left\{\mu_{kj}^{v} + \alpha^{v}\left[\sum_{i=1}^{m}\sum_{l=1}^{k}x_{li}^{v}t_i - c_m\left(1-y_{kj}^{v}\right) - d_j^{v}\right]\right\}^{+},$$

$$\alpha^{v} = \frac{\rho^{v}[\mathrm{UB} - \mathcal{L}(\lambda^{v},\mu^{v})]}{\sum_{k=1}^{m}\sum_{j=1}^{n}\left\{\left[\sum_{i=1}^{m}\left(x_{ki}^{v} - \sum_{l=1}^{k-1}x_{li}^{v}\right)g_{ij} - y_{kj}^{v}\right]^2 + \left[\sum_{i=1}^{m}\sum_{l=1}^{k}x_{li}^{v}t_i - c_m\left(1-y_{kj}^{v}\right) - d_j^{v}\right]^2\right\}}.$$

In the above equations, $\{a\}^{+} = \max\{a,0\}$, $v$ is the iteration index, $\alpha^{v}$ is the step size at the $v$th iteration, UB is the best upper bound on the optimal value by the current iteration, and $\rho^{v}$ is a scalar initially set to a value greater than 1 and is reduced by a factor of 2 when the best Lagrangian function value found thus far fails to increase by a specified number of iterations, or a better Lagrangian function value is found. The procedure of Lagrangian heuristic algorithm I is shown in Algorithm 2.3.

**Algorithm 2.3.** The specific steps of Lagrangian heuristic algorithm I are shown as follows:

**Step 1:** For initialization, let $v = 0, \rho = 2, \mathrm{UB} = +\infty, \mathrm{LB} = 0, \lambda = \mathbf{0}$, $\mu = \mathbf{0}$.

**Step 2:** Let $v = v+1$, update $\lambda_{kj}^{v}$ and $\mu_{kj}^{v}$ by the subgradient method,

**Table 2**
Notation used in the backward heuristic algorithm I.

| | |
|---|---|
| $i^{(k)}$ | The $k$th recovered facility under a given recovery sequence, $\forall k \in M$ |
| $t^{(k)}$ | The recovery time of facility $i^{(k)}, \forall k \in M$ |
| $\widehat{w}^{(k)}$ | The total weights of the customers who are served before $i^{(k)}$ is recovered, $\forall k \in M$ |
| $\widehat{w}$ | The total weights of all the customers (i.e., $\widehat{w} = \sum_{k=1}^{m} w_k$) |

and obtain the $\mathcal{L}(\lambda^v, \mu^v)$.

**Step 3:** Obtain $UB_1$ based on the matrix $X$ derived from the sub-problem three.

**Step 4:** Compute the dot products of the $k$th row of the matrix $Y$ and each row of the matrix $g$, and the $i$th row of $g$ corresponding to the largest dot product identify the $k$th recovered facility $i, i \in M, k \in M$.

**Step 5:** Repeat Step 4 from $k = 1, k = 2$ …to $k = m$, and obtain the other feasible recovery sequence, based on which we can obtain $UB_2$.

**Step 6:** Select the smaller value of $UB_1$ and $UB_2$ as UB.

**Step 7:** Repeat Step 2, 3, 4, 5, 6 until $v$ reaches an iteration limit or the value of $(UB - \mathcal{L}(\lambda, \mu))/UB$ is less than a target gap $\xi$.

### 2.3. Backward heuristic algorithm I

In this subsection, we propose an algorithm, the backward heuristic algorithm I, based on the greedy method for the SDRUF problem. The greedy method is a strategy that makes the best optimal choice at each small stage with the goal of eventually reaching a global optimal solution (see, e.g., Edmonds, 1971). The notations in Table 2 are needed for describing this heuristic algorithm.

Note that during the recovery of facility $i^{(k)}$, customers who are not connected to facilities $\left\{i^{(1)}, i^{(2)}, \ldots, i^{(k-1)}\right\}$ are unserved and waiting. Based on the definition of $\widehat{w}^{(k)}$, the total weights of such unserved customers are given by term $\widehat{w} - \widehat{w}^{(k)}$. Accordingly, the total weighted waiting time of the customers during the recovery of facility $i^{(k)}$ is $t^{(k)} \times \left[\widehat{w} - \widehat{w}^{(k)}\right]$. In this case, the objective of the SDRUF problem can be written as,

$$\mathcal{BP}_1 := \min \sum_{k=1}^{m} t^{(k)}\left[\widehat{w} - \widehat{w}^{(k)}\right] = \max\left\{\sum_{k=1}^{m} t^{(k)}\widehat{w}^{(k)}\right\} + \sum_{k=1}^{m} t^{(k)}\widehat{w}$$

where $\sum_{k=1}^{m} t^{(k)}\widehat{w}$ is constant and equal to $\widehat{w}\widehat{t}, t^{(k)}\widehat{w}^{(k)}$ is called the backward recovery value of facility $i^{(k)}$.

Differing from $\mathcal{LP}_1, \mathcal{BP}_1$ has a straightforward relationship with the recovery sequence, which indicates its potential of directly generating a feasible recovery sequence for the SDRUF problem.

We now conclude the backward heuristic algorithm I based on greedy method as Algorithm 2.4 and show its computational complexity in Lemma 5. The main idea is recovering a facility, in preference, with smaller recovery time and larger total weights of its connected customers.

**Algorithm 2.4.** The specific steps of the backward heuristic algorithm I are shown as follows:

**Step 1:** For initialization, let $k = m + 1$.

**Step 2:** Let $k = k - 1$, for each remaining facility, compute its backward recovery value, and choose facility $i^{(k)}$ with the largest backward recovery value to recover.

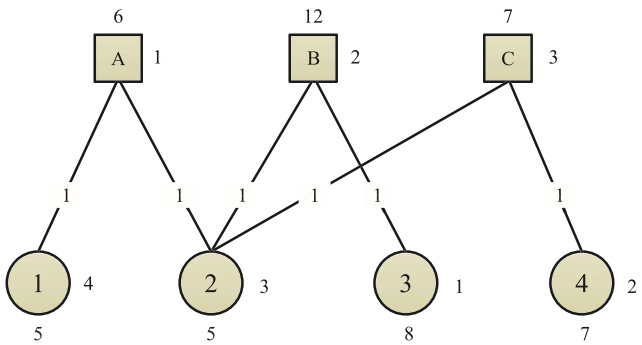


**Fig. 2.** Example of a SDRCF problem.

**Step 3:** Remove facility $i^{(k)}$ and the customers who are immediately served.

**Step 4:** Repeat Steps 2 and 3 until all facilities are chosen and recovered. The feasible recovery sequence matrix, $X$, is given by the inverted sequence of choosing facilities in Step 2.

**Lemma 5.** *Algorithm 2.4 can find a feasible recovery sequence in $O(m^2n)$, where $m$ is the total number of disrupted facilities, $n$ is the total number of customers. We show the proof of Lemma 5 in Appendix A.4.*

## 3. Sequential disruption recovery for capacitated facilities

In this section, we study the sequential disruption recovery problem for capacitated facilities, where the disrupted facilities are capacitated after recovery. We assume that the supply and demand are balanced. Our objective is to minimize the summation of the total weighted waiting time and the total transportation cost of customers. We now illustrate the SDRCF problem via a small example as follows.

**Example 2.** As shown in Fig. 2, there are three facilities, $\{A, B, C\}$, and four customers, {1,2,3,4}. The recovery time, weight and unit transportation cost are same as Example 1. The capacity of each facility is $k_A = 6, k_B = 12$, and $k_C = 7$, and the demand of each customer is $q_1 = 5$, $q_2 = 5$, $q_3 = 8$, and $q_4 = 7$, respectively. The summation of the total weighted waiting time and the total transportation cost of customers associated with different recovery sequences is $\widehat{t}_{A\to B\to C} = 53$, $\widehat{t}_{A\to C\to B} = 61$, $\widehat{t}_{B\to A\to C} = 60$, $\widehat{t}_{B\to C\to A} = 79$, $\widehat{t}_{C\to A\to B} = 71$, $\widehat{t}_{C\to B\to A} = 78$, respectively. It is then easy to find that the optimal recovery sequence is $A\to B\to C$. When $A$ is recovered, customer 1 is fully served and customer 2 is partially served. When $B$ is recovered, customers 2 and 3 are fully served. When $C$ is recovered, customer 4 is fully served.

### 3.1. Model formulation

The additional notations in Table 3 are used in the SDRCF problem formulation.

***Objective***. The objective of the SDRCF problem is minimizing the summation of the total weighted waiting time and the total transportation cost of the customers to get services:

$$\min\left\{\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}q_{ij} + \sum_{j=1}^{n} w_j d_j\right\} \tag{14}$$

***Recovery sequence constraints***. Similar to the SDRUF case, each facility in the SDRCF problem will be recovered for once and only once. The recovery sequence is constrained by:

$$\sum_{k=1}^{m} x_{ki} = 1, \forall i \in M, \tag{15}$$

$$\sum_{i=1}^{m} x_{ki} = 1, \forall k \in M. \tag{16}$$

***Supply and demand constraints***. Each facility uses up its capacity after recovery, and each customer gets full services for the recovered facilities. Hence, the resulting constraints are given by:

**Table 3**
Notation used in the SDRCF problem.

| | |
|---|---|
| $k_i$ | The capacity of facility $i, \forall i \in M$ |
| $q_{ij}$ | The quantity of goods shipped from facility $i$ to customer $j, \forall i \in M, \forall j \in N$ |
| $q_j$ | The demand of customer $j, \forall j \in N$ |
| $f_{ij}$ | Delivery decision, if some goods are shipped from facility $i$ to customer $j$, then $f_{ij} = 1$; and $f_{ij} = 0$, otherwise, $\forall i \in M, \forall j \in N$ |

$$\sum_{i=1}^{m} q_{ij} = q_j, \forall j \in N, \tag{17}$$

$$\sum_{j=1}^{n} q_{ij} = k_i, \forall i \in M. \tag{18}$$

***Quantities of transportation constraints***. Given a bipartite graph, the quantity of transportation matrix $Q$ is restricted by the connection stamp matrix and the capacity constraints.

$$q_{ij} \leqslant g_{ij} q_j, \forall i \in M, \forall j \in N. \tag{19}$$

***Waiting-time constraints***. Given the recovery sequence matrix $X$ and the transportation decision matrix $F$, we can obtain some constraints on the completion time and the waiting time, respectively.

$$c_k = \sum_{l=1}^{k} \sum_{s=1}^{m} x_{ls} t_s, \forall k \in M, \tag{20}$$

$$d_j \geqslant \sum_{l=1}^{k} \sum_{s=1}^{m} x_{ls} t_s - 2c_m + c_m \left( x_{ki} + f_{ij} \right), \forall i \in M, \forall k \in M, \forall j \in N. \tag{21}$$

In constraints (20), it is clear that term $\sum_{s=1}^{m} x_{ks} t_s$ denotes the recovery time of the $k$th recovered facility, thus, $c_k$ is corresponding completion time. According to constraints (21), if $x_{ki} = 1, f_{ij} = 1$, then $d_j \geqslant c_k$. Otherwise, if $x_{ki} + f_{ij} \leqslant 1$, then $d_j \geqslant c_k - c_m$, where $c_k - c_m \leqslant 0, \forall i \in M, \forall j \in N$. Since the objective is minimizing the waiting time, we have that $d_j = c_k$, if $x_{ki} = 1$ and $f_{ij} = 1$.

***Transportation decision constraints***. Given a bipartite graph, the transportation decision matrix $F$ is restricted by the quantities of transportation matrix. Hence, the transportation decision constraints are

$$q_{ij} \leqslant W f_{ij}, \forall i \in M, \forall j \in N. \tag{22}$$

To sum up, we can derive the mathematical formulation of the SDRCF problem as follows:

$$\begin{aligned}
\mathcal{LP}_2 \ := \ min & \left\{ \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} q_{ij} + \sum_{j=1}^{n} w_j d_j \right\} \\
s.t. \ & \sum_{k=1}^{m} x_{ki} = 1, \forall i \in M, \\
& \sum_{i=1}^{m} x_{ki} = 1, \forall k \in M, \\
& \sum_{i=1}^{m} q_{ij} = q_j, \forall j \in N, \\
& \sum_{j=1}^{n} q_{ij} = k_i, \forall i \in M, \\
& q_{ij} \leqslant g_{ij} q_j, \forall i \in M, j \in N, \\
& q_{ij} \leqslant W f_{ij}, \forall i \in M, j \in N, \\
& d_j \geqslant \sum_{l=1}^{k} \sum_{s=1}^{m} x_{ls} t_s - 2c_m + c_m \left( x_{ki} + f_{ij} \right), \forall i \in M, j \in N, k \in M, \\
& x_{ki}, f_{ij} \in \{0,1\}, q_{ij} \in \mathbb{N}, \forall i \in M, \forall j \in N, \forall k \in M.
\end{aligned}$$

We now propose two heuristic algorithms to solve the SDRCF problem in Sections 3.2 and 3.3.

### 3.2. Lagrangian heuristic algorithm II

Similar to Section 2.2, we describe the framework of the Lagrangian relaxation procedure and generate the associating algorithm in this part.

***Lagrangian dual formulation***. Based on the $\mathcal{LP}_2$, we can relax constraints (19), (21) and (22) to the objective function by using Lagrangian multipliers $\eta, \theta, \beta$. The resulting Lagrangian function $\mathcal{L}'$ is given by

$$\begin{aligned}
\mathcal{L}' \ := \ \min & \left\{ \sum_{j=1}^{n} \left( w_j - \sum_{i=1}^{m} \sum_{k=1}^{m} \theta_{ijk} \right) d_j + \sum_{i=1}^{m} \sum_{j=1}^{n} \left( \sum_{k=1}^{m} c_m \theta_{ijk} - W \beta_{ij} \right) f_{ij} + \sum_{i=1}^{m} \sum_{j=1}^{n} \left( c_{ij} + \eta_{ij} + \beta_{ij} \right) q_{ij} \right. \\
& + \sum_{k=1}^{m} \sum_{i=1}^{m} \left( \sum_{j=1}^{n} \sum_{h=k}^{m} \sum_{p=i}^{m} \theta_{pjh} t_i + \sum_{j=1}^{n} \sum_{h=k}^{m} \sum_{p=1}^{i} \theta_{pjh} t_i - \theta_{ijh} t_i + \sum_{j=1}^{n} \theta_{ijk} c_m \right) x_{ki} \\
& \left. + \left( - \sum_{i=1}^{m} \sum_{j=1}^{n} \eta_{ij} g_{ij} q_j - 2c_m \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{m} \theta_{ijk} \right) \right\} \\
& s.t. \ (15), (16), (17), (18) \text{ and } (23).
\end{aligned} \tag{24}$$

where $W$ is a large positive value. From (22) we can see that if $q_{ij} = 1$, then $f_{ij} = 1$. Otherwise, if $q_{ij} = 0$, then $f_{ij} \leqslant 1$. By additionally noting that $d_j = c_k$, if $x_{ki} = 1$ and $f_{ij} = 1$, according to (21). We can conclude that $q_{ij} = 1$ if and only if $f_{ij} = 1$.

***Binary and integer constraints***. The decision variables, $x_{ki}$ and $f_{ij}$, are binary, and $q_{ij}$ is non-negative integer in the SDRCF problem.

$$x_{ki}, f_{ij} \in \{0,1\}, q_{ij} \in \mathbb{N}, \forall i \in M, k \in M, \forall j \in N. \tag{23}$$

The correctness of the equation in $\mathcal{L}'$ is implied by the following Lemma 6, and the proof of Lemma 6 is shown in Appendix A.5.

**Lemma 6.** *The ILP $\mathcal{L}'$ holds since the following two terms are equivalent:*

$A_{kij} = B_{kij}$,

where $A_{kij} = \sum_{k=1}^{m} \sum_{i=1}^{m} \left( \sum_{j=1}^{n} \sum_{l=1}^{k} \sum_{s=1}^{m} \theta_{ijk} x_{ls} t_s + c_m \theta_{ijk} x_{ki} \right)$,

$B_{kij} = \sum_{k=1}^{m} \sum_{i=1}^{m} \left( \sum_{j=1}^{n} \sum_{h=k}^{m} \sum_{p=i}^{m} \theta_{pjh} t_i + \sum_{j=1}^{n} \sum_{h=k}^{m} \sum_{p=1}^{i} \theta_{pjh} t_i - \theta_{ijh} t_i + \sum_{j=1}^{n} \theta_{ijk} c_m \right) x_{ki}$.

For non-negative Lagrangian multipliers, $\eta, \beta, \theta$, $\mathcal{L}'(\eta, \beta, \theta)$ provides a lower bound on the objective value of $\mathcal{LP}_2$. To derive the sharpest lower bound, we should optimize the Lagrangian dual problem.

$$\begin{aligned} \mathcal{L}'^* &:= \max_{\eta\geqslant 0,\beta\geqslant 0,\theta\geqslant 0} \mathcal{L}'\Big(\eta,\beta,\theta\Big) \\ &s.t.\ (15),(16),(17),(18) \text{ and } (23). \end{aligned} \tag{25}$$

***Problem decomposition***. Given any $\eta, \beta$, and $\theta$, we can decompose $\mathcal{L}'(\eta, \beta, \theta)$ into five subproblems. Below, we illustrate these five subproblems in detail.

***Subproblem one***: transportation problem.

$$\begin{aligned} \mathcal{L}_1{}' &:= \min \sum_{i=1}^{m}\sum_{j=1}^{n}\Big(c_{ij}+\eta_{ij}+\beta_{ij}\Big)q_{ij} \\ &s.t.\ \sum_{i=1}^{m} q_{ij} = q_j, \forall j \in N, \\ &\sum_{j=1}^{n} q_{ij} = k_i, \forall i \in M, \\ &q_{ij} \in \mathbb{N}, \forall i \in M, \forall j \in N. \end{aligned}$$

This is a transportation problem where $i \in M$ are the supply sites, $j \in J$ are the demand sites and $c_{ij}+\eta_{ij}+\beta_{ij}$ is the transportation cost of shipping goods from $i$ to $j$. This transportation problem can be solved in polynomial time by the *Vogel algorithm* and *Stepping-Stone method*.

***Subproblem two***: recovery-sequence problem.

$$\begin{aligned} \mathcal{L}_2{}' := \min \sum_{k=1}^{m}\sum_{i=1}^{m}\Bigg(\sum_{j=1}^{n}\sum_{h=k}^{m}\sum_{p=i}^{m}\theta_{pjh}t_i + \sum_{j=1}^{n}\sum_{h=k}^{m}\sum_{p=1}^{i}\theta_{pjh}t_i - \theta_{ijh}t_i + \sum_{j=1}^{n}\theta_{ijk}c_m\Bigg)x_{ki} \\ s.t. \sum_{k=1}^{m} x_{ki} = 1, \forall i \in M, \\ \sum_{i=1}^{m} x_{ki} = 1, \forall k \in M, \\ x_{ki} \in \{0,1\}, \forall k \in M, i \in M. \end{aligned}$$

The is an assignment problem where $k \in M$ is the set of tasks, $i \in M$ is the set of agents, and $\sum_{j=1}^{n}\sum_{h=k}^{m}\sum_{p=i}^{m}\theta_{pjh}t_i + \sum_{j=1}^{n}\sum_{h=k}^{m}\sum_{p=1}^{i}\theta_{pjh}t_i - \theta_{ijh}t_i + \sum_{j=1}^{n}\theta_{ijk}c_m$ is the cost of assigning task $k$ to agent $i$. This problem can be solved in polynomial time by the *Hungarian algorithm*.

***Subproblem three***: regular minimization problem.

$$\begin{aligned} \mathcal{L}_3{}' &:= \min \sum_{i=1}^{m}\sum_{j=1}^{n}\Bigg(\sum_{k=1}^{m} c_m\theta_{ijk} - W\beta_{ij}\Bigg)f_{ij} \\ &s.t.\ f_{ij} \in \{0,1\}, \forall i \in M, j \in N. \end{aligned}$$

This is a regular minimization problem that can be optimally solved as follows. Let $f_{ij} = 0$, if $\sum_{k=1}^{m} c_m\theta_{ijk} - W\beta_{ij} \geqslant 0$. Otherwise, let $f_{ij} = 1, \forall i \in M, \forall j \in N$.

***Subproblem four***: constant-value problem.

$$\mathcal{L}_4{}' := \min -\sum_{i=1}^{m}\sum_{j=1}^{n}\eta_{ij}g_{ij}q_j - 2c_m\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{m}\theta_{ijk}.$$

Given the Lagrangian multipliers, $\eta$ and $\theta$, the value of this subproblem is a constant.

***Subproblem five***: waiting-time constrained problem.

**Table 4**
Notation used in the supply demand problem.

| | |
|---|---|
| $U$ | The set of facilities, $U = \{1,2,\ldots,u\}$, where $u$ is the number of disrupted facilities that serve customer $j, \forall j \in N$ |
| $k_p$ | The capacity of facility $p$ that serves customer $j, \forall p \in U, \forall j \in N$ |
| $t_p$ | The recovery time of disrupted facility $p, \forall p \in U$ |
| $z_p$ | Decision variable. Let $z_p = 1$, if facility $p$ is recovered; $z_p = 0$, otherwise, $\forall p \in U$ |

$$\mathcal{L}_5{}' := \min \sum_{j=1}^{n}\Bigg(w_j - \sum_{k=1}^{m}\sum_{i=1}^{m}\theta_{ijk}\Bigg)d_j.$$

To sharpen the Lagrangian lower bound, we now impose some constraints on the waiting time of each customer to $\mathcal{L}_5{}'$. Note that these constraints are redundant to $\mathcal{L}'$. Thus the feasibility of the resulting solutions still holds. The constraints are given in Lemmas 7 and 8. The correctness of Lemma 8 is not straightforward. We show the proof of Lemma 8 in Appendix A.6.

**Lemma 7**. *An upper bound of the waiting time for each customer, $j \in N$, is given by*

$$d_j \leqslant \sum_{i=1}^{m} t_i, \forall j \in N, \tag{26}$$

**Lemma 8**. *A lower bound of the waiting time for each customer, $j \in N$, is given by*

$$d_j \geqslant d_j', \forall j \in N, \tag{27}$$

The $d_j{}'$ is given by a supply demand problem and the additional notations given in Table 4 are used in formulating the supply demand problem.

$$d_j{}' := \min \sum_{p=1}^{u} t_p z_p \tag{28}$$

$$s.t. \sum_{p=1}^{u} k_p z_p \geqslant q_j, \forall p \in U, \forall j \in N, \tag{29}$$

$$z_p \in \{0,1\}, \forall p \in U. \tag{30}$$

Eq. (28) is the objective function of the supply demand problem that minimizes the total recovery time of the selected facilities in $U$. Constraints (29) mean that the facilities in $U$ are selected to meet the demand of customer $\forall j \in N$. Constraints (30) are binary constraints of the supply demand problem. The optimal solution of this supply demand problem is the lower bound of the waiting time $d_j, j \in N$.

According to the lower and upper bounds of $\{d_j : \forall j \in N\}$, the waiting-time constrained problem can be updated as the following LP.

$$\begin{aligned} \mathcal{L}_5{}' &:= \min \sum_{j=1}^{n}\Bigg(w_j - \sum_{k=1}^{m}\sum_{i=1}^{m}\theta_{ijk}\Bigg)d_j \\ &s.t. (26) \text{ and } (27). \end{aligned}$$

This is also a regular minimization problem that can be optimally solved as follows. For each $j \in N$, if $w_j - \sum_{k=1}^{m}\sum_{i=1}^{m}\theta_{ijk} > 0$, let $d_j = d_j{}'$. Otherwise, let $d_j = \sum_{i=1}^{m} t_i$.

***The subgradient method***. To find the optimal Lagrangian multipliers, $\eta^*, \beta^*$, and $\theta^*$, to maximize $\mathcal{L}'(\eta, \beta, \theta)$, we apply the subgradient method to update $\{\eta_{ij}, \beta_{ij}, \theta_{ikj}\}$ at each iteration:

$$\eta_{ij}^{v'+1} = \Big\{\eta_{ij}^{v'} + \alpha^{v'}\Big[q_{ij}^{v'} - q_j g_{ij}\Big]\Big\}^{+}, \beta_{ij}^{v'+1} = \Big\{\beta_{ij}^{v'} + \alpha^{v'}\Big[q_{ij}^{v'} - Wf_{ij}^{v'}\Big]\Big\}^{+},$$

$$\theta_{ikj}^{v'+1} = \Bigg\{\theta_{ikj}^{v'} + \alpha^{v'}\Bigg[\sum_{l=1}^{k}\sum_{s=1}^{m} x_{ls}^{v'}t_s - 2c_m + c_m\Big(x_{ki}^{v'} + f_{ij}^{v'}\Big) - d_j^{v'}\Bigg]\Bigg\}^{+},$$

where

$$\alpha^{v'} = \frac{\rho' v'[\mathrm{UB}' - \mathcal{L}'(\eta^{v'}, \beta^{v'}, \theta^{v'})]}{V1 + V2},$$

$$V1 = \sum_{i=1}^{m}\sum_{j=1}^{n}\left\{\left[q_{ij}^{\nu'} - q_j g_{ij}\right]^2 + \left[q_{ij}^{\nu'} - W f_{ij}^{\nu'}\right]^2\right\}, V2$$

$$= \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{m}\left[\sum_{l=1}^{k}\sum_{s=1}^{m} x_{ls}^{\nu'} t_s^{\nu'} - 2c_m + c_m\left(x_{ki}^{\nu'} + f_{ij}^{\nu'}\right) - d_j^{\nu'}\right]^2.$$

In the above equations, the parameters are similar to the SDRUF problem, The procedure of Lagrangian heuristic algorithm II is shown in Algorithm 3.1.

**Algorithm 3.1.** The specific steps of Lagrangian heuristic algorithm II are shown as follows:

**Step 1:** For initialization, let $\nu' = 0, \rho' = 2, \mathrm{UB}' = +\infty, \mathrm{LB}' = 0, \eta = \mathbf{0}$, $\beta = \mathbf{0}, \theta = \mathbf{0}$.

**Step 2:** Let $\nu' = \nu' + 1$, update $\eta_{ij}^{\nu'}, \beta_{ij}^{\nu'}, \theta_{ikj}^{\nu'}$ by the subgradient method, and obtain the $\mathcal{L}'(\eta^{\nu'}, \beta^{\nu'}, \theta^{\nu'})$.

**Step 3:** Obtain the $\mathrm{UB}_1'$ based on the matrix $Q$ derived from the subproblem one.

**Step 4:** Obtain the $\mathrm{UB}_2'$ based on the matrix $X$ derived from the subproblem two.

**Step 5:** Obtain the $\mathrm{UB}'$, where $\mathrm{UB}' = \mathrm{UB}_1' + \mathrm{UB}_2'$.

**Step 6:** Repeat Step 2, 3, 4, 5 until $\nu'$ reaches an iteration limit or the value of $(\mathrm{UB}' - \mathcal{L}'(\eta, \beta, \theta))/\mathrm{UB}'$ is less than a target gap $\xi'$.

### 3.3. Backward heuristic algorithm II

Similar to subSection 2.3, in this subsection, we develop a new algorithm for the SDRCF problem based on greedy method, i.e., the backward heuristic algorithm II. We now conclude the resulting heuristic algorithm in Algorithm 3.2.

**Algorithm 3.2.** The backward heuristic algorithm II includes the following steps:

**Step 1:** For a given bipartite graph, $G$, simply regard the original problem as a transportation problem without considering sequential facility recovery and find the optimal quantity of transportation matrix, $Q$, by *Vogel algorithm* and *Stepping-Stone method*. $Q$ is thus feasible for the original SDRCF problem, $\mathcal{LP}_2$.

**Step 2:** According Step 1, remove the arcs from the original graph, $G$, that did not ship goods from facility $i$ to customer $j, \forall i \in M, j \in N$. Thus, $\forall i \in M, j \in N$. When $q_{ij} = 0$, set $g_{ij} = 0$. In this case, obtain a new graph, $G'$.

**Step 3:** Based on the new graph, $G'$, simply regard the original SDRCF problem, $\mathcal{LP}_2$, as a SDRUF problem without considering the transportation problem, which can be solved by Algorithm 2.4. Combine Steps 1 and 3 to get a feasible solution to the original SDRCF problem, $\mathcal{LP}_2$.

**Lemma 9.** *Algorithm 3.2 can find a feasible recovery sequence for the SDRCF problem in $O(m^3n^3)$, where $m$ is the total number of disrupted facilities, $n$ is the total number of customers. We show the proof of Lemma 9 in Appendix A.7.*

## 4. Computational results and discussion

We conduct all computational experiments on a Windows 10 PC with an Intel Core i7-8700 and 8 GB RAM, 3.2 GHz CPU. All algorithms are implemented by Matlab Release 2019a.

### 4.1. Computational results of the SDRUF problem

We now show the effectiveness and efficiency of the Lagrangian heuristic algorithm I and the backward heuristic algorithm I in the uncapacitated cases. In the experiments, we randomly construct 12 types of disrupted bipartite graphs with different scales, and same density of arcs (i.e., 50%). The normal recovery time of different facilities $\{t_i : \forall i \in M\}$ and the weights of customers $\{w_j : \forall j \in N\}$ are integers uniformly distributed in interval $[1, 10]$. The computational result is the average of 50 simulated instances for each problem size, and the detailed results are presented in Table 5 and Fig. 3.

In Table 5, columns "$\mathrm{T}_{\mathrm{LR}_1}$", "$\mathrm{T}_{\mathrm{B}_1}$" and "$\mathrm{T}_{\mathrm{Gurobi}}$" separately represent the average computational time of carrying out the Lagrangian heuristic algorithm I, the backward heuristic algorithm I and obtaining optimal solutions with Gurobi 9.0.2. Columns "Heuristic", "Heuristic One" and "Heuristic Two" respectively represent the values of $(\mathrm{UB}_1^*\text{-}\Omega_1)/\Omega_1$, $(\mathrm{LR}_1\text{-}\Omega_1)/\Omega_1$ and $(\mathrm{B}_1\text{-}\Omega_1)/\Omega_1$, where $\mathrm{LR}_1$ reports the upper bound derived from the Lagrangian heuristic algorithm I, $\mathrm{B}_1$ shows the upper bound derived from the backward heuristic algorithm I, $\Omega_1$ represents the optimal value of $\mathcal{LP}_1$, and $\mathrm{UB}_1^* = \min\{\mathrm{LR}_1, \mathrm{B}_1\}$. We set the average runtime limit is 2 h (i.e., 7200 s).

To better compare the computational results of the Lagrangian heuristic algorithm I and the backward heuristic algorithm I, we show in Fig. 3 the detailed number of outperforming instances of either heuristic, where $\mathrm{N}_{\mathrm{LR}_1 > \mathrm{B}_1}$ represents the number of instances $\mathrm{LR}_1$ outperforming $\mathrm{B}_1$ and $\mathrm{N}_{\mathrm{LR}_1 \leqslant \mathrm{B}_1}$ represents the number of instances $\mathrm{B}_1$ outperforming $\mathrm{LR}_1$. From Table 5 and Fig. 3, we have the following observations.

First, both the heuristic algorithms are effective in solving the SDRUF problem. The values under columns "Heuristic One", "Heuristic Two" show that the average gaps between the feasible solutions produced by both the heuristic algorithms and the optimal solutions are less than 10% under all situations.

Second, both the heuristic algorithms are efficient in solving the SDRUF problem. We can see from columns "$\mathrm{T}_{\mathrm{LR}_1}$" and "$\mathrm{T}_{\mathrm{B}_1}$" that all

**Table 5**
The computational results of the SDRUF problem.

| $(m,n)$ | Heuristic (%) | | | Heuristic One (%) | | | Heuristic Two (%) | | | CPU Time (s) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Avg. | Max. | Min. | Avg. | Max. | Min. | Avg. | Max. | Min. | $\mathrm{T}_{\mathrm{LR}_1}$ | $\mathrm{T}_{\mathrm{B}_1}$ | $\mathrm{T}_{\mathrm{Gurobi}}$ |
| (5,10) | 0.99 | 13.55 | 0.00 | 1.40 | 13.55 | 0.00 | 6.86 | 60.33 | 0.00 | 0.02 | 0.00 | 0.01 |
| (5,20) | 1.54 | 10.22 | 0.00 | 2.11 | 10.22 | 0.00 | 5.00 | 23.40 | 0.00 | 0.03 | 0.00 | 0.03 |
| (5,200) | 0.14 | 2.96 | 0.00 | 0.55 | 5.74 | 0.00 | 0.55 | 4.18 | 0.00 | 0.20 | 0.00 | 0.94 |
| (10,20) | 3.35 | 20.97 | 0.00 | 5.34 | 20.97 | 0.00 | 9.02 | 45.66 | 0.00 | 0.07 | 0.00 | 0.60 |
| (10,40) | 4.56 | 19.60 | 0.00 | 5.32 | 28.26 | 0.00 | 8.06 | 27.62 | 0.00 | 0.12 | 0.00 | 3.11 |
| (10,200) | 1.88 | 7.27 | 0.00 | 6.82 | 45.13 | 0.00 | 2.94 | 21.58 | 0.00 | 0.45 | 0.00 | 28.37 |
| (15,30) | 5.88 | 20.93 | 0.26 | 8.02 | 39.11 | 0.26 | 9.60 | 25.00 | 0.62 | 0.19 | 0.00 | 11.36 |
| (15,60) | 4.40 | 15.03 | 0.00 | 5.77 | 33.64 | 0.01 | 7.11 | 23.28 | 0.00 | 0.33 | 0.00 | 78.76 |
| (20,200) | 3.85 | 10.71 | 0.16 | 4.94 | 14.63 | 0.92 | 5.27 | 13.68 | 0.16 | 1.57 | 0.00 | – |
| (30,200) | – | - | – | – | – | – | – | – | – | 3.87 | 0.01 | – |
| (40,200) | – | – | – | – | – | – | – | – | – | 7.94 | 0.01 | – |
| (50,200) | – | – | – | – | – | – | – | – | – | 14.76 | 0.02 | – |

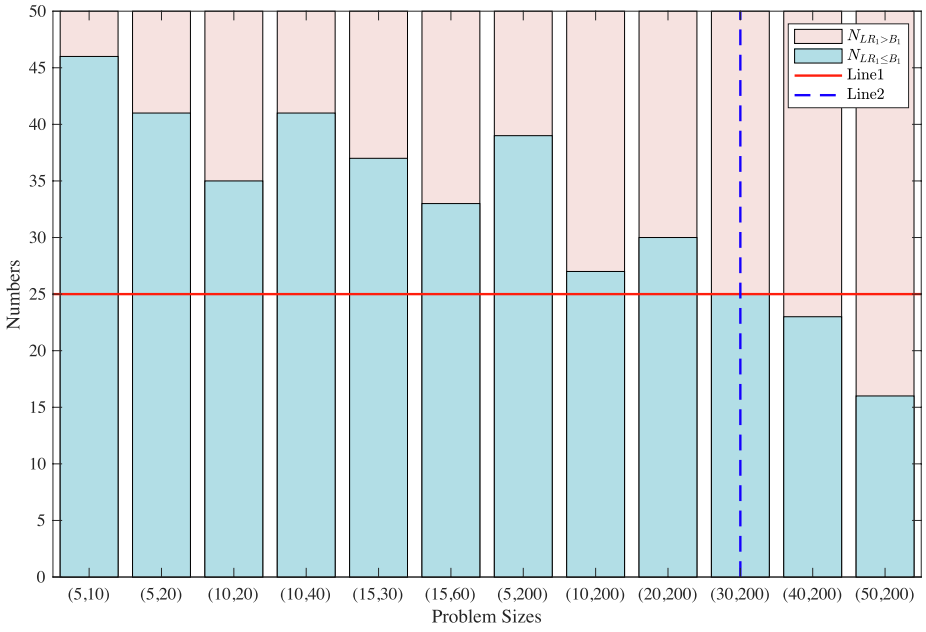


**Fig. 3.** Complementarity roles of the two heuristic algorithms.

**Table 6**
Computational results of the SDRCF problem.

| $(m,n)$ | $LR_2/LB_2$(%) | | | $B_2/LB_2$(%) | | | CPU Time (s) | | |
|---|---|---|---|---|---|---|---|---|---|
| | Avg. | Max. | Min. | Avg. | Max. | Min. | $T_{LR_2}$ | $T_{B_2}$ | $T_{Gurobi}$ |
| (5,10) | 100.15 | 100.29 | 100.06 | 100.11 | 100.23 | 100.05 | 0.05 | 0.04 | 0.04 |
| (5,20) | 100.16 | 100.30 | 100.07 | 100.11 | 100.24 | 100.05 | 0.05 | 0.04 | 0.11 |
| (5,200) | 100.14 | 100.23 | 100.05 | 100.11 | 100.19 | 100.05 | 0.80 | 0.09 | 2.48 |
| (10,20) | 100.42 | 100.59 | 100.26 | 100.33 | 100.56 | 100.17 | 0.07 | 0.06 | 27.58 |
| (10,40) | 100.44 | 100.77 | 100.24 | 100.33 | 100.50 | 100.17 | 0.12 | 0.05 | 95.80 |
| (10,200) | 100.44 | 100.59 | 100.30 | 100.33 | 100.47 | 100.18 | 1.88 | 0.17 | 2401.91 |
| (15,30) | 100.69 | 101.12 | 100.37 | 100.53 | 100.81 | 100.32 | 0.11 | 0.04 | – |
| (15,60) | 100.72 | 101.02 | 100.50 | 100.53 | 100.85 | 100.29 | 0.31 | 0.06 | – |
| (20,200) | 101.01 | 101.27 | 100.61 | 100.73 | 100.98 | 100.39 | 4.17 | 0.40 | – |
| (30,200) | 101.64 | 102.10 | 101.22 | 101.17 | 101.66 | 100.83 | 7.15 | 0.69 | – |
| (40,200) | 102.24 | 102.86 | 101.79 | 101.57 | 102.04 | 101.20 | 10.78 | 1.07 | – |
| (50,200) | 102.88 | 103.55 | 102.39 | 102.08 | 102.81 | 101.70 | 14.97 | 1.55 | – |

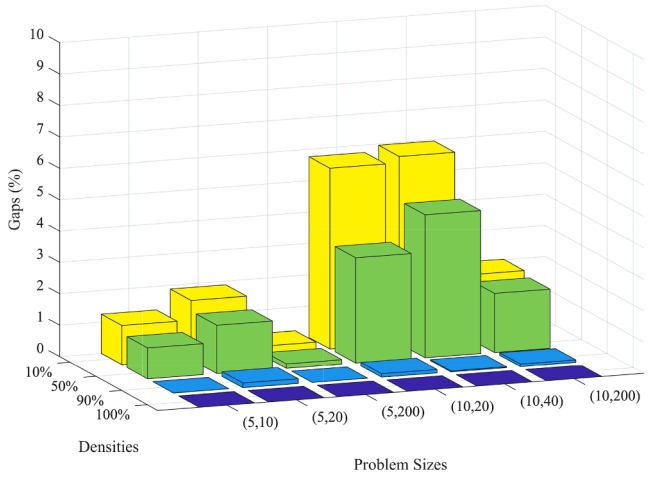


**Fig. 4.** The impacts of density level on computational effectiveness in the SDRUF problem.

problems can be solved within 15 s via both the heuristic algorithms, which are much shorter than the runtime of Gurobi 9.0.2 shown in column "$T_{Gurobi}$". It is remarkable that $T_{Gurobi}$ increases significantly when problem size goes larger causing more variables introduced in $\mathcal{LP}_1$.

Third, both the heuristic algorithms can act as good alternatives for each other in the uncapacitated cases. In particular, as shown in Fig. 3, all $N_{LR_1 \leq B_1}$ are larger than 25 on the left side of Line 2, namely, the Lagrangian heuristic algorithm I is more effective than the backward heuristic algorithm I under small-medium scales, while all $N_{LR_1 > B_1}$ are smaller than 25 on the right side of Line 2, namely, the backward heuristic algorithm I can do better under large scales.

### 4.2. Computational results of the SDRCF problem

We now show the effectiveness and efficiency of the two heuristic algorithms in the capacitated cases. The normal transportation cost of different arcs $\{c_{ij} : \forall (i,j) \in E\}$ and the demand of customers $\{q_j : \forall j \in N\}$ are integers uniformly distributed in interval $[10, 40]$. The other parameters are the same as those in the previous Section 4.1. The

computational result is the average of 50 simulated instances for each problem size, and the detailed results are shown in Table 6.

In Table 6, we replace $\Omega_2$ with $LB_2$ to measure performance of the both algorithms because the optimal solutions are difficult to obtain with Gurobi 9.0.2 under medium-large scales, where $LB_2$ represents the lower bound derived from the Lagrangian heuristic algorithm II. We set the average runtime limit is 2 h (i.e., 7200 s). From the above computational results, we have the following observations.

First, both the heuristic algorithms are effective in solving the SDRCF problem. From columns "$LR_2/LB_2$" and "$B_2/LB_2$", we can easily see that all of the percentage ratios are less than 105%, which means that the average gaps between the feasible solutions derived from both the heuristic algorithms and the optimal solutions are less than 5%.

Second, both the heuristic algorithms are efficient in solving the SDRCF problem. We can see from columns "$T_{LR_2}$" and "$T_{B_2}$" that all problems can be solved within 15 s, which are much shorter than the runtime of Gurobi 9.0.2 shown in column "$T_{Gurobi}$". As what we could expected, once again, $T_{Gurobi}$ increases significantly when problem size goes larger causing more variables introduced in $\mathcal{LP}_2$.

Third, the backward heuristic algorithm II is more dominant compared with the Lagrangian heuristic algorithm II in capacitated cases, however, the latter can provide sharp lower bound $LB_2$ to measure performance of the near-optimal feasible solutions derived from both the heuristic algorithms when the optimal solutions are difficult to obtain with Gurobi 9.0.2 under medium-large scales.

### 4.3. Impacts of density level on computational results

In the end, we show the impacts of different density levels of arcs on the computational gaps, i.e., $(UB_1^*-\Omega_1)/\Omega_1$ and $(UB_2^*-\Omega_2)/\Omega_2$. Taking the SDRUF problem as example, we conclude the results in Fig. 4, and the detailed results are shown in A.8.

In Fig. 4, We can see that the average gap is smaller when the network is denser. The reason may be that denser network has greater flexibility (i.e., each recoverd facility can serve more customers), and thus, the optimal recovery sequence might be multiple, leading to better performance of heuristic algorithms. The computational results of the SDRCF problem shows similar phenomenon to the SDRUF problem, although the average CPU time is longer. More detailed results are shown in A.9.

## 5. Conclusion

The focus of this paper is on the sequential disruption recovery for uncapacitated facilities and capacitated facilities problems. The resulting optimization problems are formulated as two integer linear programs. Motivated by the Lagrangian relaxation and greedy method, we propose the Lagrangian heuristic algorithms and the backward heuristic algorithms both of which, as shown in Tables 5 and 6, can generate feasible solutions for sequential disruption recovery problems effectively and efficiently. In addition, the two heuristic algorithms can act as good alternatives for each other. The results can guide the central authority making reasonable post-disaster recovery policies.

There are some limitations of this paper, indicating some interesting directions of future research. First, one can consider cases where partially recovered facilities can provide partial services to users, and thus, the corresponding optimization problems could be modeled as some new mixed integer linear programs. Second, sometimes multiple disrupted facilities can be recovered simultaneously, in this case the new challenge lies in how to assign the recovery resources effectively. Third, in practice, disruption recovery problems are also affected by uncertainties, e.g., uncertainty over demands of customers and recovery time of facilities, one can investigate relevant stochastic optimization problems or robust optimization problems. Suffice it to say that the results of sequential disruption recovery problem in this paper provide some preliminary study for the above research topics.

## CRediT authorship contribution statement

**Hongwei Xuan:** Writing - review & editing, Methodology, Software, Investigation. **Chao Zhang:** Writing - original draft. **Yugang Yu:** Supervision. **Lindong Liu:** Supervision, Conceptualization, Methodology, Validation. All authors contribute equally to this paper.

## Acknowledgements

The work was supported by the National Natural Science Foundation of China [Grant Nos. 72022018, 71701192, 71921001].

## Appendix A

### A.1. Proof of Lemma 1

**Proof.** Constraints (4) ensure that each unserved customer can be served by a unique connecting facility in the uncapacitated cases. Mathematically, there is a unique "1" in each column of matrix $Y$. Constraints (5) indentify that which element equals to "1" in each column of $Y$. For each customer, if customer $j$ is connected to the $k$th recovered facility $i^k$, we get $\sum_{i=1}^{m} x_{ki} g_{ij} = 1$, otherwise, $\sum_{i=1}^{m} x_{ki} g_{ij} = 0$. Accordingly, if $j$ is connected to facilities recovered before $i^k$, we get $\sum_{l=1}^{k-1}\sum_{i=1}^{m} x_{li} g_{ij} \geqslant 1$, otherwise, $\sum_{l=1}^{k-1}\sum_{i=1}^{m} x_{li} g_{ij} = 0$, in this case, $\sum_{i=1}^{m}\left(x_{ki} - \sum_{l=1}^{k-1} x_{li}\right) g_{ij} = 1$, which determines $y_{kj}$. Moreover, since there is a unique "1" in each column of $Y$, the service-mapping matrix $Y$ is determined. □

### A.2. Proof of Lemma 2

**Proof.** To prove Eq. (10), we need to make the following transformations:

$$\sum_{k=1}^{m}\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{l=1}^{k} t_i \mu_{kj} x_{li} = \sum_{i=1}^{m} t_i \left(\sum_{k=1}^{m}\sum_{j=1}^{n}\sum_{l=1}^{k} \mu_{kj} x_{li}\right), \tag{Eq.(A.1))}$$

$$\sum_{k=1}^{m}\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{h=k}^{m} t_i \mu_{hj} x_{ki} = \sum_{i=1}^{m} t_i \left(\sum_{k=1}^{m}\sum_{j=1}^{n}\sum_{h=k}^{m} \mu_{hj} x_{ki}\right). \tag{Eq. (A.2))}$$

In Eqs. (Eq. (A.1)) and (Eq. (A.2)), we use $A_i$ and $B_i$ to replace the formulas in brackets. Thus,

$A_i = \sum_{k=1}^{m}\sum_{j=1}^{n}\sum_{l=1}^{k}\mu_{kj}x_{li}, B_i = \sum_{k=1}^{m}\sum_{j=1}^{n}\sum_{h=k}^{m}\mu_{hj}x_{ki}.$

To prove Eq. (10), we only need to show that $A_i = B_i$. For $A_i$, we expand it into the following form:

(1) when $k = 1$, bring $k = 1$ into $\sum_{j=1}^{n}\sum_{l=1}^{k}\mu_{kj}x_{li}$, and we derive $\sum_{j=1}^{n}\sum_{l=1}^{1}\mu_{1j}x_{li} = x_{1i}\sum_{j=1}^{n}\mu_{1j}$,
(2) when $k = 2$, bring $k = 2$ into $\sum_{j=1}^{n}\sum_{l=1}^{k}\mu_{kj}x_{li}$, and we derive $\sum_{j=1}^{n}\sum_{l=1}^{2}\mu_{2j}x_{li} = (x_{1i} + x_{2i})\sum_{j=1}^{n}\mu_{2j}$,
(3) ……,
(4) when $k = m$, bring $k = m$ into $\sum_{j=1}^{n}\sum_{l=1}^{k}\mu_{kj}x_{li}$, and we derive $\sum_{j=1}^{n}\sum_{l=1}^{m}\mu_{mj}x_{li} = (x_{1i} + x_{2i} + \ldots + x_{mi})\sum_{j=1}^{n}\mu_{mj}$.

Add $x_{1i}\sum_{j=1}^{n}\mu_{1j}, (x_{1i} + x_{2i})\sum_{j=1}^{n}\mu_{2j}$, …, $(x_{1i} + x_{2i} + \ldots + x_{mi})\sum_{j=1}^{n}\mu_{mj}$ together and extract common factors to obtain

$$A_i = x_{1i}\left(\sum_{j=1}^{n}\mu_{1j} + \sum_{j=1}^{n}\mu_{2j} + \ldots + \sum_{j=1}^{n}\mu_{mj}\right) + x_{2i}\left(\sum_{j=1}^{n}\mu_{2j} + \ldots + \sum_{j=1}^{n}\mu_{mj}\right) + \ldots + x_{mi}\left(\sum_{j=1}^{n}\mu_{mj}\right). \quad \text{(Eq. (A.3))}$$

For Eq. (Eq. (A.3)), we make the following replacement:

$$x_{ki}\sum_{j=1}^{n}\mu_{kj} = x_{ki}\sum_{h=k}^{k}\sum_{j=1}^{n}\mu_{hj}, \forall k \in M.$$

Add $x_{1i}\sum_{h=1}^{m}\sum_{j=1}^{n}\mu_{hj}, x_{2i}\sum_{h=2}^{m}\sum_{j=1}^{n}\mu_{hj}$, …, $x_{mi}\sum_{h=m}^{m}\sum_{j=1}^{n}\mu_{hj}$ together, we have

$$x_{1i}\sum_{h=1}^{m}\sum_{j=1}^{n}\mu_{hj} + x_{2i}\sum_{h=2}^{m}\sum_{j=1}^{n}\mu_{hj} + x_{mi}\sum_{h=m}^{m}\sum_{j=1}^{n}\mu_{hj} = \sum_{k=1}^{m}\sum_{h=k}^{m}\sum_{j=1}^{n}\mu_{hj}x_{ki}.$$

Then, $A_i$ is transformed into the following form:

$$A_i = \sum_{k=1}^{m}\sum_{h=k}^{m}\sum_{j=1}^{n}\mu_{hj}x_{ki}.$$

Since $B_i = \sum_{k=1}^{m}\sum_{h=k}^{m}\sum_{j=1}^{n}\mu_{hj}x_{ki}$, we have $A_i = B_i$. Finally, we obtain the following equation:

$$\sum_{k=1}^{m}\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{l=1}^{k}t_i\mu_{kj}x_{li} = \sum_{k=1}^{m}\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{h=k}^{m}t_i\mu_{hj}x_{ki}.$$

□

### A.3. Proof of Lemma 4

**Proof.** For each customer, $j \in N$, the upper bound of waiting time $d_j$ is the summation of the total construction time of the facilities not connected to $j$ and the maximal construction time among the facilities connected to $j$. That is, $d_j$ is less than the summation of terms $\widehat{t} - \sum_{k=1}^{m}t_k g_{kj}$ and $\max\left\{t_k g_{kj} : \forall k \in M\right\}$. □

### A.4. Proof of Lemma 5

**Proof.** In the worst case, the number of calculations to find the best backward recovery value for the $m$th time is $m * n$, with the $(k-1)$th time being $(m-1) * n$, the $(k-2)$th time being $(m-2) * n$ and so on, the number of calculations to find the best backward recovery value for the 1th time is $n$, the sum of the number of calculations above is $(1 + m) * m * n/2$, so Algorithm 2.4 can find a feasible recovery sequence for the SDRUF problem in $O(m^2n)$.□

### A.5. Proof of Lemma 6

**Proof.** First, we derive $A_{kij} = \sum_{k=1}^{m}\sum_{i=1}^{m}(\sum_{j=1}^{n}\sum_{h=k}^{m}\sum_{s=1}^{m}\theta_{ijh}x_{ks}t_s + c_m\theta_{ijk}x_{ki})$ by Lemma 2.

Second, let $C_{kij} = \sum_{k=1}^{m}\sum_{i=1}^{m}(\sum_{j=1}^{n}\sum_{h=k}^{m}\sum_{s=1}^{m}\theta_{ijh}x_{ks}t_s + c_m\theta_{ijk}x_{ki})$, then $C_{kij}$ can be decomposed into the following formula:

$$C_{kij} = \sum_{k=1}^{m}\sum_{i=1}^{m}\left(\sum_{j=1}^{n}\sum_{h=k}^{m}\left(\sum_{s=1}^{i}\theta_{ijh}x_{ks}t_s + \sum_{s=i}^{m}\theta_{ijh}x_{ks}t_s - \theta_{ijh}x_{ki}t_i\right) + \sum_{j=1}^{n}c_m\theta_{ijk}x_{ki}\right).$$

Third, we can derive following equations by Lemma 2,

$$\sum_{j=1}^{n}\sum_{h=k}^{m}\sum_{s=1}^{i}\theta_{ijh}x_{ks}t_s=\sum_{j=1}^{n}\sum_{h=k}^{m}\sum_{p=i}^{m}\theta_{pjh}x_{ki}t_i,\sum_{j=1}^{n}\sum_{h=k}^{m}\sum_{s=i}^{m}\theta_{ijh}x_{ks}t_s=\sum_{j=1}^{n}\sum_{h=k}^{m}\sum_{p=1}^{i}\theta_{pjh}x_{ki}t_i.$$

Then, $C_{kij}$ is transformed into the following form:

$$C_{kij}=\sum_{k=1}^{m}\sum_{i=1}^{m}\left(\sum_{j=1}^{n}\sum_{h=k}^{m}\sum_{p=i}^{m}\theta_{pjh}t_i+\sum_{j=1}^{n}\sum_{h=k}^{m}\sum_{p=1}^{i}\theta_{pjh}t_i-\theta_{ijh}t_i+\sum_{j=1}^{n}\theta_{ijk}c_m\right)x_{ki}.$$

Finally, we can derive $C_{kij}=B_{kij}\Rightarrow A_{kij}=B_{kij}$. □

### A.6. Proof of Lemma 8

**Proof.** For each customer, $j\in N$, the lower bound of waiting time $d_j'$ is the total shortest completion time of the facilities in $U$ selected to meet the demand of customer $j$. Obviously, this is a deformed "0 −1" knapsack problem that can be optimally solved in pseudo-polynomial time by a *dynamic programming algorithm* (see Andonov, Poirriez, & Rajopadhye, 2000).□

### A.7. Proof of Lemma 9

**Proof.** First, we transform the SDRCF problem into a transportation problem without considering recovery, the computational complexity of solving the transportation problem included in the SDRCF problem by the traditional vogel algorithm is $O(m^2n+mn^2)$. Second, we solve the transportation problem and obtain a transportation matrix $Q$, based which we can construct a new graph $g'$, and transform the SDRCF problem into a SDRUF problem without considering the transportation cost. Third, we solve the above SDRUF problem based on the new graph $g'$ by Algorithm 2.4. Proof Appendix A.4 shows that the computational complexity of Algorithm 2.4 for the SDRUF problem is $O(m^2n)$, so Algorithm 3.2 can find a feasible recovery sequence for the SDRCF problem in $O(m^4n^2+m^3n^3)$. Since $m$ is less than $n$, $O(m^4n^2+m^3n^3)$ can be unified into $O(m^3n^3)$.□

### A.8. The impacts of density level on computational efficiency in the SDRUF problem

| $(m,n)$ | $(UB_2^*-\Omega_2)/\Omega_2$ (%) | | | | $T_{Gurobi}$ (s) | | | |
|---|---|---|---|---|---|---|---|---|
| | $d=10\%$ | $d=50\%$ | $d=90\%$ | $d=100\%$ | $d=10\%$ | $d=50\%$ | $d=90\%$ | $d=100\%$ |
| (5,10) | 1.25 | 0.99 | 0.00 | 0.00 | 0.02 | 0.01 | 0.02 | 0.01 |
| (5,20) | 1.88 | 1.54 | 0.15 | 0.00 | 0.03 | 0.03 | 0.02 | 0.01 |
| (5,200) | 0.25 | 0.14 | 0.00 | 0.00 | 0.77 | 0.94 | 0.18 | 0.09 |
| (10,20) | 5.76 | 3.35 | 0.11 | 0.00 | 1.46 | 0.60 | 0.07 | 0.02 |
| (10,40) | 5.96 | 4.56 | 0.03 | 0.00 | 3.43 | 3.11 | 0.16 | 0.06 |
| (10,200) | 2.02 | 1.88 | 0.08 | 0.00 | 26.16 | 28.37 | 1.20 | 0.55 |
| (15,30) | – | 5.88 | 0.16 | 0.00 | – | 11.36 | 0.36 | 0.15 |
| (15,60) | – | 4.40 | 0.12 | 0.00 | – | 78.76 | 0.80 | 0.38 |
| (20,200) | – | – | 0.18 | 0.00 | – | – | 13.82 | 4.41 |
| (30,200) | – | – | – | – | – | – | – | – |

### A.9. The impacts of density level on computational efficiency in the SDRCF problem

| $(m,n)$ | $(UB_2^*-\Omega_2)/\Omega_2$, d = 50% | | | $(UB_2^*-\Omega_2)/\Omega_2$, d = 100% | | | CPU Time (s), $d=100\%$ | | |
|---|---|---|---|---|---|---|---|---|---|
| | Avg. | Max. | Min. | Avg. | Max. | Min. | $T_{LR_2}$ | $T_{B_2}$ | $T_{Gurobi}$ |
| (5,10) | 0.01 | 0.05 | 0.00 | 0.01 | 0.04 | 0.00 | 0.04 | 0.04 | 0.08 |
| (5,20) | 0.00 | 0.02 | 0.00 | 0.00 | 0.04 | 0.00 | 0.07 | 0.04 | 0.16 |
| (5,200) | 0.00 | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 3.11 | 0.08 | 2.67 |
| (10,20) | 0.04 | 0.09 | 0.01 | 0.04 | 0.12 | 0.00 | 0.11 | 0.04 | 55.42 |
| (10,40) | 0.03 | 0.09 | 0.01 | 0.03 | 0.08 | 0.00 | 0.31 | 0.04 | 283.29 |
| (10,200) | 0.01 | 0.02 | 0.00 | 0.00 | 0.02 | 0.00 | 7.30 | 0.18 | 4533.29 |
| (15,30) | – | – | – | – | – | – | – | – | – |

## References


Andonov, R., Poirriez, V., & Rajopadhye, S. (2000). Unbounded knapsack problem: Dynamic programming revisited. *European Journal of Operational Research, 123*, 394–407. https://doi.org/10.1016/S0377-2217(99)00265-9

Bao, S., Zhang, C., Ouyang, M., & Miao, L. (2019). An integrated tri-level model for enhancing the resilience of facilities against intentional attacks. *Annals of Operations Research, 283*, 87–117. https://doi.org/10.1007/s10479-017-2705-y

Brown, G., Carlyle, M., Salmerón, J., & Wood, K. (2006). Defending critical infrastructure. *Interfaces, 36*, 530–544. https://doi.org/10.1287/inte.1060.0252

Burstall, R. M., Leaver, R. A., & Sussams, J. E. (1962). Evaluation of transport costs for alternative factory sites-a case study. *Journal of the Operational Research Society, 13*, 345–354. https://doi.org/10.1057/jors.1962.51

Cauvin, A. C. A., Ferrarini, A. F. A., & Tranvouez, E. T. E. (2009). Disruption management in distributed enterprises: A multi-agent modelling and simulation of cooperative recovery behaviours. *International Journal of Production Economics, 122*, 429–439. https://doi.org/10.1016/j.ijpe.2009.06.014

Cui, T., Ouyang, Y., & Shen, Z. J. M. (2010). Reliable facility location design under the risk of disruptions. *Operations Research, 58*, 998–1011. https://doi.org/10.1287/opre.1090.0801

Edmonds, J. (1971). Matroids and the greedy algorithm. *Mathematical Programming, 1*, 127–136. https://doi.org/10.1007/BF01584082

Erlenkotter, D. (1978). A dual-based procedure for uncapacitated facility location. *Operations Research, 26*, 992–1009. https://doi.org/10.1287/opre.26.6.992

Fisher, M. L. (1981). The lagrangian relaxation method for solving integer programming problems. *Management Science, 27*, 1–18. https://doi.org/10.1287/mnsc.27.1.1

Held, M., & Karp, R. M. (1970). The traveling-salesman problem and minimum spanning trees. *Operations Research, 18*, 1138–1162. https://doi.org/10.1287/opre.18.6.1138

Hishamuddin, H., Sarker, R. A., & Essam, D. (2013). A recovery model for a two-echelon serial supply chain with consideration of transportation disruption. *Computers & Industrial Engineering, 64*, 552–561. https://doi.org/10.1016/j.cie.2012.11.012

IFRC (2020). World disasters report 2020: Come heat or high water. https://media.ifrc.org/ifrc/world-disaster-report-2020/. Accessed December 30, 2020.

Ivanov, D. (2019). Disruption tails and revival policies: A simulation analysis of supply chain design and production-ordering systems in the recovery and post-disruption periods. *Computers & Industrial Engineering, 127*, 558–570. https://doi.org/10.1016/j.cie.2018.10.043

Jia, H., Ordonez, F., & Dessouky, M. (2007). A modeling framework for facility location of medical services for large-scale emergencies. *IIE Transactions, 39*, 41–55. https://doi.org/10.1080/07408170500539113

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval Research Logistics Quarterly, 2*, 83–97. https://doi.org/10.1002/nav.3800020109

Li, Q., Zeng, B., & Savachkin, A. (2013). Reliable facility location design under disruptions. *Computers & Operations Research, 40*, 901–909. https://doi.org/10.1016/j.cor.2012.11.012

Liberatore, F., Scaparra, M. P., & Daskin, M. S. (2011). Analysis of facility protection strategies against an uncertain number of attacks: The stochastic r-interdiction median problem with fortification. *Computers & Operations Research, 38*, 357–366. https://doi.org/10.1016/j.cor.2010.06.002

Losada, C., Scaparra, M. P., & OHanley, J. R. (2012). Optimizing system resilience: a facility protection model with recovery time. *European Journal of Operational Research, 217*, 519–530. https://doi.org/10.1016/j.ejor.2011.09.044

Oded, B., & Celik, P. (1982). Sequential facility location with distance dependent demand. *Journal of Operations Management, 2*, 261–268. https://doi.org/10.1016/0272-6963(82)90014-6

Paul, S. K., Sarker, R., & Essam, D. (2015). Managing disruption in an imperfect production-inventory system. *Computers & Industrial Engineering, 84*, 101–112. https://doi.org/10.1016/j.cie.2014.09.013

Tomlin, B. (2006). On the value of mitigation and contingency strategies for managing supply chain disruption risks. *Management Science, 52*, 639–657. https://doi.org/10.1287/mnsc.1060.0515

Van Roy, T. J. (1986). A cross decomposition algorithm for capacitated facility location. *Operations Research, 34*, 145–163. https://doi.org/10.1287/opre.34.1.145

Walker, C. G., Snowdon, J. N., & Ryan, D. M. (2005). Simultaneous disruption recovery of a train timetable and crew roster in real time. *Computers & Operations Research, 32*, 2077–2094. https://doi.org/10.1016/j.cor.2004.02.001

Wesolowsky, G. O. (1973). Dynamic facility location. *Management Science, 19*, 1241–1248. https://doi.org/10.1287/mnsc.19.11.1241

Xiao, T., Qi, X., & Yu, G. (2007). Coordination of supply chain after demand disruptions when retailers compete. *International Journal of Production Economics, 109*, 162–179. https://doi.org/10.1016/j.ijpe.2006.11.013

Yu, G., & Qi, X. (2004). *Disruption management: framework, models and applications*. World Scientific.