Braess Paradox in the Optimal Multiperiod Resource-Constrained Restoration Scheduling Problem

Abstract:

This study examines the Braess paradox in the context of the multiple-period restoration scheduling problem. A bilevel programming model is devised, where the upper-level problem is to determine the optimal sequence of recovery activities considering the limited resource constraint, while the lower-level problem is the traffic assignment model that captures passengers' responses to changes in the transportation network capacity. Then, a novel genetic algorithm (GA) is developed to solve the proposed restoration scheduling problem. Our case study first shows that the optimal restoration schedule does not concur with the results of the link importance measurement, and the former can achieve a 4% total travel time reduction compared with the latter. Then, various numerical experiments are conducted to illustrate the occurrence and properties of the Braess paradox, which is that the network performance in some restoration periods can be better than that before the disruption or after a damaged link is recovered. Moreover, it is revealed that with sufficient resources for multiple links to be repaired simultaneously, it is unnecessary to do so in the optimal rehabilitation schedule due to the existence of the Braess paradox. Finally, in terms of algorithmic performance, our proposed GA outperforms the particle swarm optimisation algorithm and can reduce the computation time by up to 14%.

Keywords: Braess paradox; Traffic assignment; Bilevel network design; Network resilience; Restoration scheduling

1 Introduction

Transportation networks are the backbone of modern cities. Nevertheless, they are vulnerable and can suffer significant damage from disruptive events, such as earthquakes, hurricanes, floods, and tsunamis, resulting in enormous economic losses. For instance, the 1994 Northridge earthquake caused more than $40 billion in monetary loss and more than 140 road closures [1]. In October 2012, Hurricane Sandy struck New York City, causing approximately $7.5 billion in damage to the transportation system [2, 3]. In particular, the 8.0-magnitude earthquake that struck Sichuan, China, in 2008 caused significant damage to 21 highways [4].

Although we cannot control losses during a disaster, we can carefully plan restoration and rehabilitation activities. Such activities also require significant time and cost [5] and profoundly impact the infrastructure's postdisaster performance [6]. Therefore, planning transportation infrastructure restoration given limited time and resources is highly important, and developing an efficient restoration schedule deserves additional research attention.

In principle, the restoration scheduling problem is closely related to the problem of analysing infrastructure resilience. The term “resilience” was first introduced in the ecological domain [7] and can be summarised as the ability to resist, absorb, adapt to, and recover from a disruption. Most studies on resilience either focus on quantifying resilience or evaluating network performance. In recent years, resilience has been extensively studied in the transportation field. According to Gu et al. [8], quantitative indicators of resilience can be classified into topology-based, system-based and economic-based indicators.

Typical topology-based indicators include network diameter [9], average degree [9], betweenness centrality [10], connectivity-based [9, 11, 12], average shortest path [13, 14] and efficiency [10, 15]. System-based indicators include travel time [16-20] and travel demand [17, 21]. Economic-based indicators include total cost [22-24] and the economic loss ratio [25]. In addition, with further analysis of the effect of restoration on resilience, other resilience indicators, including total recovery time [1, 6, 26], recovery trajectory [1, 6, 27], system crash frequency [19] and recovery ratios [28-30], have been identified.

Among the various previous indicators, travel time is one of the most widely adopted. In general, travel time is affected mainly by infrastructure capacity. The scheduling of restoration tasks profoundly impacts travel time by affecting changes in infrastructure capacity during the whole scheduling period. Moreover, passengers adjust their route choice according to the rehabilitated infrastructure capacity within each subperiod, such as a week or a month. A few studies have addressed the optimal restoration scheduling problem in the transportation domain, considering user choice behaviour. There are two prevailing approaches, the simulation approach [18, 31] and the analytical approach [32-35], where the latter primarily builds upon the assumption of user equilibrium. Comparably, the simulation approach enables the capture of more practical features with a longer computation time, while the analytical approach is better suited to analysing the properties of the problem.

In addition to the performance indicators considered, resource constraints are a critical factor affecting restoration scheduling. There are different types of limited resources, such as monetary budgets [22, 35] and materials and human resources [27, 28]. Resource constraints can affect restoration activities from various perspectives, among which is the number of restoration activities carried out within one period, which is primarily captured in mathematical models. The number of such activities could be either one [18, 20] or many [27, 32].

Intuitively, the sooner and the more infrastructures can be rehabilitated, the better the system performance is. However, this may not be the case in a transportation network, considering the existence of the Braess paradox. The paradox states that network performance may not improve after a new link is added to a transport network [36]. It has been extensively discussed for different cases in the transportation field, such as under elastic demand [37-39], stochastic assignment [40-42] and user equilibrium [43]. This study focuses on the Braess paradox during damaged infrastructure restoration periods, and resource constraints are considered an important factor affecting the restoration of transportation networks. Although the restoration procedure resembles the setting of the paradox, i.e., providing a new link to a network, and the occurrence of this paradox could be expected, it has never been examined, particularly in the setting of multiple periods with limited capacity constraints. It will be interesting to explore whether this paradox occurs in such a setting and how it affects the recovery trajectory of the system's performance. Compared with the traditional Braess paradox, the most important differences in this study are as follows: 1) In the multiperiod resource-constrained restoration scheduling problem, we split the whole restoration phase into periods and study the Braess paradox phenomenon in each period rather than analysing it once, as in the traditional Braess paradox. 2) Traditionally, the index that measures whether there is a Braess paradox relies on adding a new link to a transportation network, where this added link leads to an increase in total travel time. However, we apply a unified network performance measure to determine the paradoxical phenomenon that captures travel time, traffic flows, and the importance of damaged links.

To address this research gap, this study first formulates a bilevel programming model for the optimal multiperiod restoration scheduling problem with resource constraints. The upper-level problem is to determine the optimal restoration schedule that minimises the total travel time, while the lower-level problem is to model passengers' route choice behaviour and capture passengers' responses to changes in the transportation network topology and capacity. Afterwards, various experiments are conducted to illustrate the occurrence of the paradox during the restoration process.

The remainder of this study is organised as follows: Section 2 presents the model formulation for the restoration scheduling problem. Section 3 introduces a genetic algorithm approach to solve the bilevel optimisation model. Numerical examples are presented in Section 4 to examine the occurrence of the Braess paradox during the restoration process and how it affects the trajectory of network performance. Finally, Section 5 concludes the study and discusses future research directions.

2 Problem description and formulation

2.1 Problem description

We consider a general transport network denoted by $G=(N,A)$, where N and A represent the sets of nodes and links in the network, respectively. After a disruptive event, the set of links that need restoration is given by R. The planned restoration horizon is denoted by H, where the restoration time for each link is measured as a multiple of a common minimum period τ, such as one week or one month. Moreover, restoring each link requires both time and variable resources, such as human and construction materials. The total resources that can be utilised within one period are limited. Depending on the required resources for repairing a link, multiple links may be reconstructed within the same period. Developing a reasonable plan is the key to restoring the disrupted network as quickly as possible, and it is important to note that the network is considered to be eventually restored to its predamaged state. Based on this problem description, the investigated multiperiod resource-constrained restoration scheduling problem aims to determine the order of repair of the disrupted links considering the limited resource constraints.

The following assumptions are made to facilitate the model development. A1) For simplicity, different resources can be converted into one monetary value. Hence, the resource constraint can be formulated as a budget constraint associated with each period. A2) The resources required for each disrupted link are given and known. This is reasonable because the disruption associated with a link can be measured after a disaster, and the time and resources required to fix it can be estimated accordingly. A3) The restoration of each disrupted link is initiated at the beginning of each period. Moreover, once the reconstruction starts, it continues until the link is fully repaired. A4) The damaged part of the link will be functional only if it is fully repaired and its capacity returns to its designed level. A5) The total travel demand between each OD pair is constant over the planning horizon. A6) Travellers adjust their route choices based on the capacity of the transportation infrastructure from period to period, while their route choice is assumed to be calculated by the traffic assignment model.

2.2 Bilevel formulation

Based on the program description, a bilevel programming model is devised, where the upper-level problem is to determine the order of the repaired links with the resource constraint, and the lower-level problem is the traffic assignment problem. The upper- and lower-level formulations are presented in the following two subsections, and the notation used in developing the model is given in Table 1.

Table 1 List of notation

Notation	Definition
Sets	
N	Set of nodes
A	Set of links
R	Set of disrupted links to be restored
W	Set of OD pairs
H	Set of restoration periods, $H=(1,2,...,T_{\max})$
K_τ^w	Set of paths connecting OD pair w in period τ, $\tau \in H$, $w \in W$
Parameters	
g_τ^w	Travel demand of OD pair w in period τ, $\tau \in H$, $w \in W$
$T_{\max}$	Upper bound on the restoration period
$\bar{C}$	Maximum units of available resources in each period
T_a	Recovery periods required for restoring link a, $a \in R$
C_a	Resources required for restoring link a, $a \in R$
M	A sufficiently large positive number
Variables	
$v_{a,\tau}$	Flow travelling via link a in period τ, $a \in A, \tau \in H$
$t_{a,\tau}$	Travel time on link a in period τ, $a \in A$, $\tau \in H$
λ_τ^w	Equilibrium travel time between OD pair w in period τ, $\tau \in H$, $w \in W$
$f_{k,\tau}^w$	Flow travelling on path k connecting OD pair w in period τ, $k \in K_\tau^w$, $\tau \in H$, $w \in W$
s_a	Starting period of restoring link a
e_a	Ending period of restoring link a
$y_{a,\tau}$	Binary variable indicating whether a link is under restoration: $y_{a,\tau}=1$ if link a is under restoration during period τ; otherwise, $y_{a,\tau}=0$
$\delta_{a,k,\tau}^w$	Binary variable indicating whether a path traverses link a: $\delta_{a,k,\tau}^w=1$ if path k traverses link a in period τ; otherwise, $\delta_{a,k,\tau}^w=0$

2.2.1 Upper-level formulation

The upper-level problem focuses on minimising the total travel time. In this problem, although the restoration time is not explicitly taken as the objective function, it is expected that a rehabilitation plan that recovers the network in a short time leads to a lower total travel time, as a fully functional network generally has a lower total travel time than a network with disrupted links. Therefore, to a certain extent, if we use the objective of minimising the total travel time, we can capture a shorter restoration time requirement. Mathematically, the upper-level problem is formulated as follows:

$$\min_{\mathbf{s},\mathbf{v},\mathbf{y}} \sum_{\tau \in H} \sum_{a \in A} v_{a,\tau} t_{a,\tau} \tag{1}$$

$$e_a = s_a + T_a, \forall a \in R \tag{2}$$

$$\tau - s_a \le M\left(1 - y_{a,\tau}\right), \forall a \in R, \tau \in H \tag{3}$$

$$e_a - \tau \ge M\left(y_{a,\tau} - 1\right), \forall a \in R, \tau \in H \tag{4}$$

$$\sum_{a \in R} C_a y_{a,\tau} \le \bar{C}, \forall \tau \in H \tag{5}$$

$$s_a, e_a \in H, \forall a \in R \tag{6}$$

$$y_{a,\tau} \in \{0,1\}, \forall a \in R, \tau \in H \tag{7}$$

The objective function of the upper-level model, Eq. (1), is to minimise the total travel time through the whole restoration horizon. $\mathbf{s} = [s_a]$, $\mathbf{v} = [v_{a,\tau}]$, and $\mathbf{y} = [y_{a,\tau}]$ are the vectors associated with the decision variables, namely, the starting time of rehabilitating a disrupted link, the flow travelling via each link, and a binary variable indicating whether a link is under repair during period τ. Eq. (2) captures the relationship between the starting and ending times for restoring link *a*. Eqs. (3) and (4) together require that during the restoration period τ, $y_{a,\tau}$ is set to 1 if link *a* is under restoration; otherwise, it is equal to 0. Eq. (5) represents limited resource constraints, stating that the total resources used do not exceed the available resources for each period. Finally, Eqs. (6) and (7) are the definitional constraints for the decision variables.

2.2.2 Lower-level formulation

Within each period, the links that are not disrupted or repaired form the operational transport network for travellers. Based on assumption (A6), the passenger flows distributed within each period τ follow user equilibrium. In line with the classic Beckmann transformation [44, 45], the following minimisation problem is solved to obtain the passenger flow distribution for each period τ.

$$\min_{\mathbf{v}} \sum_{a \in A} \int_0^{v_{a,\tau}} t_{a,\tau}(v)\, dv \tag{8}$$

$$\sum_{k \in K_\tau^w} f_{k,\tau}^w = g_\tau^w, \forall w \in W \tag{9}$$

$$v_{a,\tau} = \sum_{w \in W} \sum_{k \in K_\tau^w} f_k^w \delta_{a,k,\tau}^w, \forall a \in A \tag{10}$$

$$f_{k,\tau}^w \ge 0, \forall w \in W, k \in K_\tau^w \tag{11}$$

The objective function in Eq. (8) is the sum of the integrals of the link performance functions. It does not have any intuitive economic or behavioural interpretation and should be viewed strictly as a mathematical construct that is utilised to solve equilibrium problems [45]. Eq. (9) is the flow conservation constraint for each OD pair. Eq. (10) is the definitional constraint between the link flow and path flow. Eq. (11) is the nonnegative constraint for the path flow. The lower-level formulation is essentially a user equilibrium traffic assignment problem. The equivalence between the user equilibrium condition and the mathematical programming problem can be established by checking the necessary Karush–Kuhn–Tucker (KKT) conditions, as shown in [45]. To guarantee the uniqueness of the solution to the problem, the feasible region,

which is defined by Eqs. (9) and (11), must be convex, and the objective function Eq. (8) must be strictly convex in the vicinity of the optimal link flow (and convex elsewhere). Specifically, the convexity of the feasible region is guaranteed by the linear equality constraints. It is worth noting that the convexity of the objective function in Eq. (8) is assured as long as the link performance function is strictly convex. Therefore, theoretically, it can be verified that at optimality, the KKT conditions of the above minimisation problem satisfy the user equilibrium condition, meaning that all the paths used have equal travel times.

3 Solution procedure

The developed model can be classified as a bilevel network design model, which is generally acknowledged as an NP-hard problem [46] and is challenging to obtain exact solutions for. The brute force method can enumerate all possible restoration schedules to ensure a global optimal solution, but this approach is not applicable to large-scale networks. In general, the prevailing methodology is to develop metaheuristic or heuristic methods, such as genetic algorithms [47], artificial bee colony algorithms [48], ant colony algorithms [49], Tabu search [50], and simulated annealing [51], to solve this problem. Shayanfar and Schonfeld [47] showed that the GA performs better in terms of finding the near-optimal solution for the sequence and scheduling problem. Therefore, the GA is employed to optimise the restoration scheduling problem. Afterwards, given a certain schedule, the Frank–Wolfe algorithm [45] is employed to solve the lower-level traffic assignment problem.

3.1 Chromosome representation and generation

In this study, each chromosome is encoded by two substrings of numbers, namely, g_{s1} and g_{s2}. Let $g_{s1}=\left(g_{s1}^{1},g_{s1}^{2},...,g_{s1}^{|R|-1},g_{s1}^{|R|}\right)$ be a substring consisting of $|R|$ gens, where $|R|$ is the number of disrupted links. The gen value g_{s1}^{i} of g_{s1} represents the index of the disrupted link, and the value of *i* is an integer between 1 and $|R|$. It is worth noting that each number between 1 and $|R|$ appears only once in g_{s1} to indicate the restoration sequence and the continuity of each restoration activity. Moreover, let $g_{s2}=\left(g_{s2}^{1},g_{s2}^{2},...,g_{s2}^{|R|-1},g_{s2}^{|R|}\right)$ be the second substring of a chromosome. The gen value g_{s2}^{i} of g_{s2} denotes the starting period of each disrupted link, and this value is an integer between 1 and $T_{\max}$. The smaller g_{s2}^{i} is, the earlier the end of the restoration activity. Fig. 1 shows an example of a chromosome representation in which the repair order and the starting period are generated randomly, whereas the disrupted links and the corresponding restoration periods are determined. The procedure of the GA is given below.

Step 1. Initialisation.

Step 1.1. Set the required parameters for the GA, including the maximum number of iterations $G_{\max}$, population size N, crossover rate p_c and mutation rate p_m.

Step 1.2. Set the number of iterations $G=0$ and generate N initial solutions by including all the disrupted links. Then, determine whether $\sum_{a\in R} C_a > \overline{C}$ or $\sum_{a\in R} C_a = 0$ in period τ. If so, the solution is not feasible, and another solution needs to be generated until a feasible one is found.

Step 1.3. Calculate the travel time in each restoration period according to the lower-level traffic assignment model, and then obtain the fitness value (i.e., the reciprocal of the total travel time) of each feasible solution.

Step 2. Iteration process.

Step 2.1. Sort the population from best to worst according to the fitness values, and then select the best solution and place it directly into the next-generation population.

Step 2.2. Select the parents by roulette wheel selection and generate the offspring using crossover and mutation operators.

Step 2.3. Repair the newly generated solution, if necessary, and generate the new population for the next iteration.

Step 2.4. Check the termination criterion. If $G=G_{\max}$, terminate the iteration process and output the optimal solution; otherwise, set $G=G+1$ and return to Step 2.1.

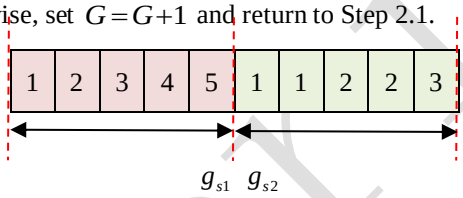


Fig. 1 Example of a chromosome

3.2 Operators

3.2.1 Selection

As mentioned above, the fitness value of a chromosome is equal to the reciprocal of the total travel time. A chromosome with the highest fitness is inserted directly into the new population as the elite individual to replace the worst chromosome. Moreover, the fitness determines the probability that the chromosomes are selected as parent chromosomes by the roulette wheel. If the fitness of the j^{th} chromosome satisfies the following condition, the j^{th} chromosome is selected as the parent chromosome.

$$\frac{\sum_{q=1}^{j-1} F_q}{\sum_{j=1}^{N} F_j} < \xi \le \frac{\sum_{q=1}^{j} F_q}{\sum_{j=1}^{N} F_j} \tag{12}$$

where F_j denotes the fitness value of the j^{th} chromosome and ξ is a uniformly distributed random number between 0 and 1.

3.2.2 Crossover

Crossover is used to generate offspring by exchanging the genetic information of the parent chromosomes. Notably, simple crossover strategies that may cause duplication and gaps do not apply to our problem because each disrupted link can be restored only once. Therefore, partially mapped crossover is employed for the substring g_{s1} in this study. Fig. 2 illustrates an example of partially mapped crossover.

3.2.3 Mutation

To prevent the algorithm from being trapped in a local optimum and improve its efficiency, we adopt a mutation operation on the chromosomes. Consistent with the processing objective of the crossover operation, random right-shift mutation is employed for the first substring. This operator, which is shown in Fig. 3, randomly selects a gen value and inserts it to the right.

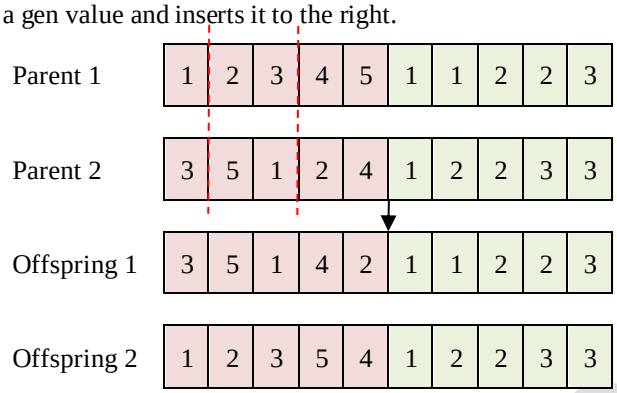


Fig. 2 Crossover operation

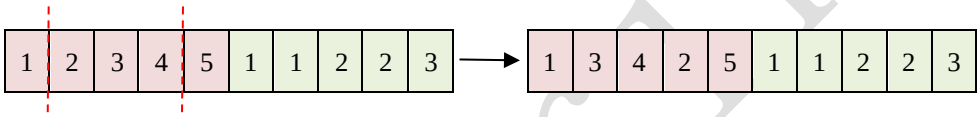


Fig. 3 Mutation operation

3.2.4 Repair

The procedures for performing crossover and mutation cannot guarantee the feasibility of the solution, which may result in $\sum_{a\in R} C_a > \overline{C}$ or $\sum_{a\in R} C_a = 0$ in period τ. To address this, a repair operator is constructed and applied to repair the newly generated chromosomes. After the crossover operation, the steps for repair are described as follows.

Step 1. Obtain the resource consumption.

Step 1.1. Set $j = 1$.

Step 1.2. Calculate the consumed resources for the j^{th} chromosome, $\sum_{a\in R} C_a$, that is, the sum of the resources required for restoring the disrupted links in period τ.

Step 2. Compare $\sum_{a\in R} C_a$ and $\overline{C}$.

Step 2.1. If $0 < \sum_{a\in R} C_a \le \overline{C}$ in period τ, let $j = j+1$ and return to Step 1.2; otherwise, proceed to Step 2.2.

Step 2.2. If $\sum_{a\in R} C_a > \overline{C}$ in period τ, select the corresponding gene of the chromosome and increase the starting period by one until $\sum_{a\in R} C_a = \overline{C}$, then set $j = j+1$ and return to Step 1; otherwise, proceed to Step 2.3.

Step 2.3. If $\sum_{a\in R} C_a = 0$ in period τ, select the corresponding gene of the chromosome and reduce the starting period by one until $\sum_{a\in R} C_a > 0$. Set $j = j+1$ and return to Step 1.2.

Step 3. Repeat Steps 1 and 2 until $j = N$.

In this study, the repair procedure after applying the mutation operator is consistent with the above procedure; thus, we do not go into much detail here.

4 Numerical examples

The following experiments are conducted to illustrate the properties of the proposed model and evaluate the performance of the proposed algorithm. We first use a six-node network to examine the effect of travel demand and tolerance in the event of complete damage. The second numerical example applies the Nguyen-Dupuis Network to show how the resource level and damage level affect the restoration schedule if a link is partially disrupted. All the experiments were coded using Microsoft Visual Studio Community 2022, and all the following numerical applications were performed on a laptop with 16 GB RAM and an Intel (R) Core (TM) i7-10510U CPU @ 1.80 GHz. The parameters for the GA in our study are as follows: maximum iteration number $G_{\max} = 2000$, population size $N = 20$, and crossover and mutation rates of 0.8 and 0.2, respectively.

4.1 Six-node network

The six-node network is constructed as shown in Fig. 4. The link performance functions are given by $t_1(v_1) = 8v_1$, $t_2(v_2) = 25 + 0.5v_2$, $t_3(v_3) = 2v_3$, $t_4(v_4) = 25 + 0.5v_4$, $t_5(v_5) = 50 + v_5$, $t_6(v_6) = 10v_6$, $t_7(v_7) = 10 + v_7$, $t_8(v_8) = 5v_8$, and $t_9(v_9) = 2 + v_9$. Moreover, the travel demand between OD pair 1-6 is set to 6.0. Five links, i.e., 4, 6, 7, 8, and 9, are disrupted.

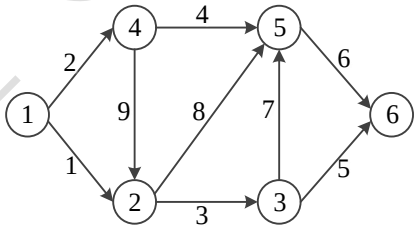


Fig. 4 Six-node network

4.1.1 Occurrence of the paradox

In the case of multiple disrupted links, the traditional empirical approach usually determines the restoration sequence according to the importance of each disrupted link. Therefore, based on the research of Qiang and Nagurney [52], determining the increment in the total travel time when a link is removed from the network is regarded as an empirical strategy. Table 2 presents the rankings of these links.

Table 2 Ranking of links under normal-level demand

Ranking	Link no.	Total travel time after removing the link
1	6	589.9
2	9	558.3
3	8	546.3
4	4*	526.1
5	7*	521.9

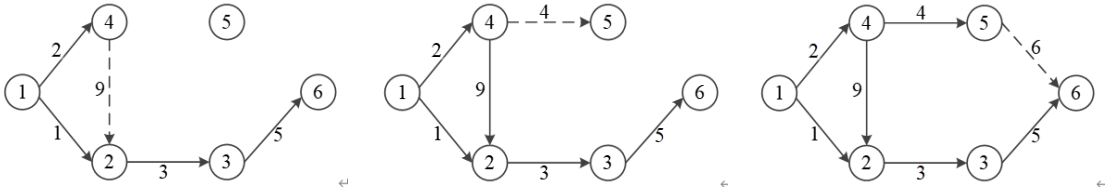
*Remark: Links 4 and 7 are considered to be Braess paradox links, as the total time without any disruption is 526.1.

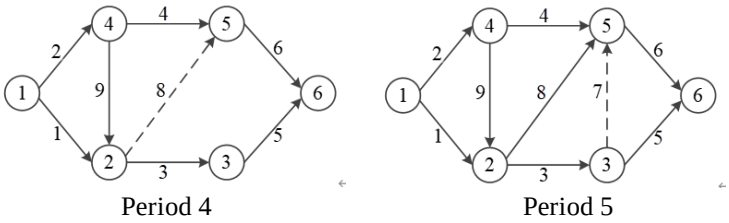
In this experiment, only one link can be restored in each period, and one period is needed to rebuild one disrupted link. Furthermore, the five disrupted links are assumed to be completely closed. To obtain the optimal restoration schedule for the six-node network using the proposed optimisation model and to explore and investigate the Braess paradox in the restoration scheduling problem, the brute force method is adopted to enumerate all the possible restoration schedules. Table 3 lists two restoration schedules. One is the optimal solution of the model, and the other is the schedule determined according to the importance of these links. In other words, a link with a higher importance rank will be scheduled to recover earlier. Interestingly, the optimal restoration schedule does not accord with the importance rank of the links. The total travel time obtained from the optimal schedule decreases by more than 4% compared with that from the schedule following the rank shown in Table 2.

Table 3 Single-link restoration scheduling

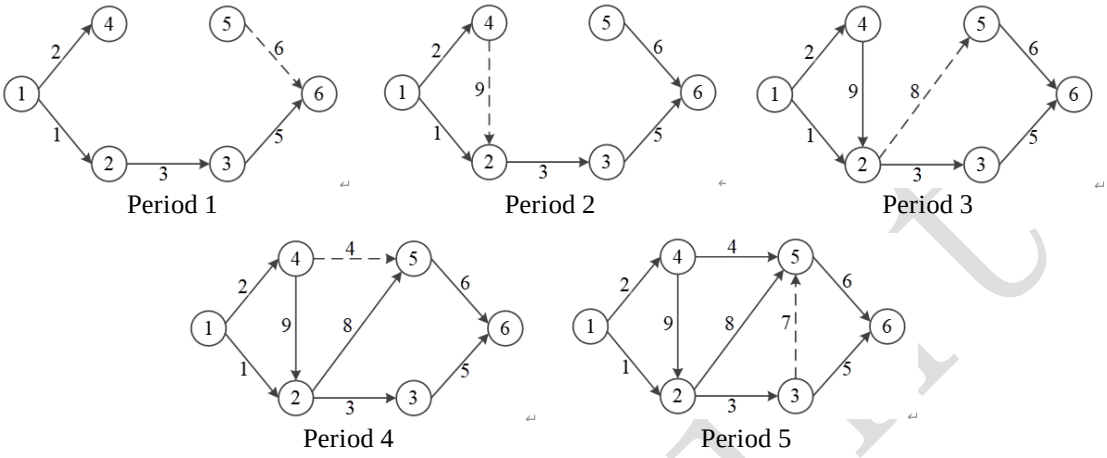
		Restoration period					Total travel time
		1	2	3	4	5	
Optimal schedule	Link under reconstruction	9	4	6	8	7	2895.7
	Travel time in each period	696.0	589.9	589.9	498.0	521.9	
Schedule based on Table 2	Link under reconstruction	6	9	8	4	7	3025.7
	Travel time in each period	696.0	696.0	589.9	521.9	521.9	

The difference in the total travel time between the two solutions lies in Periods 2 and 4. To explain their differences, Fig. 5 is plotted to compare the recovered networks of the two solutions in each period. In the figure, a solid line represents a fully functional link, while a dotted line represents the link under reconstruction in the period, which cannot be used. By recovering link 9 in the first period, a new path traversing links 2-9-3-5 is formed under the optimal plan compared with the plan based on the importance rank. The total travel time is thus reduced due to the reduced congestion of the path that traverses links 1-3-5. As shown in Fig. 5 (a), the newly recovered link 6 could divert flow from link 9 in period 4, which would mitigate the congestion cost and result in a lower total time. In Fig. 5 (b), a new path traversing links 2-9-8-6 is generated with the restoration of link 8; however, the newly recovered link plays a small role and cannot divert flow from link 9.





(a) Networks from the optimal schedule



(b) Networks from the schedule obtained according to the importance rank

Fig. 5 Comparison of the networks of the two solutions during the restoration phase

In summary, the results justify the motivation for developing an optimal restoration model instead of relying on the link importance indicators to determine the recovery schedule. Theoretically, the importance associated with one link is determined by removing the link independently from the network. Moreover, its importance is measured with respect to the network without disruption. When a disruptive event occurs, the network for determining link importance varies. Therefore, relying on the importance of the link element determined from a normal network to form a restoration plan may not be optimal.

Furthermore, we can plot the changes in the network performance curve (Fig. 6). In this study, we adopt the unified network performance measure (UNPM) proposed by Qiang and Nagurney [52] to measure network performance. Mathematically, it is expressed as

$$\varepsilon\left(\mathbf{v}_\tau\right)=\frac{1}{|W|}\sum_{w\in W}\frac{g_\tau^w}{\lambda_\tau^w\left(\mathbf{v}_\tau\right)},\forall\tau\in H \tag{13}$$

where $\mathbf{v}_\tau$ is the vector notation for the traffic flow and $\lambda_\tau^w\left(\mathbf{v}_\tau\right)$ denotes the equilibrated travel time between OD pair *w* in period τ. Both are obtained by solving the lower-level problem. g_τ^w is the travel demand between OD pair *w*, and $|W|$ is the number of OD pairs. According to the above equation, at a given demand level, a lower travel time leads to a higher $\varepsilon\left(\mathbf{v}_\tau\right)$. Then, for ease of comparison of different settings, we further normalise the network performance measurement as follows:

$$\varphi\left(\tau\right)=\frac{\varepsilon\left(\mathbf{v}_\tau\right)}{\varepsilon\left(\mathbf{v}_{\tau_0}\right)}\times100\%,\forall\tau\in H \tag{14}$$

where τ_0 denotes the normal time period before or after the disruptive event and $\varepsilon\left(\mathbf{v}_{\tau_0}\right)$ represents the UNPM computed in period τ_0. When there is no disruption and the network performs normally, $\varphi(\tau)=100\%$. Based on Eq. (13), the lower the value of $\varphi(\tau)$ is, the worse the network performance is, and vice versa. In particular, $\varphi(\tau)>100\%$ indicates that the network performance is better than the normal performance without disruption. This characteristic is utilised to visualise and identify the Braess paradox in the figures.

To illustrate the process clearly, in Fig. 6, we plot two additional normal periods, where the network performance is 100%, before and after the restoration periods. As shown in Fig. 6, the network performance for the optimal schedule is generally better than that based on ranking. In particular, following the former schedule, the network performance in time period 6 is 105.6%, which is greater than the original value. This means that before completely repairing all the transport infrastructure, the network performance is better than that in a normal period. Then, when all the links are repaired after period 8, the network performance decreases compared to that in period 6. This finding highlights the paradox that the existence of newly recovered links may degrade network performance.

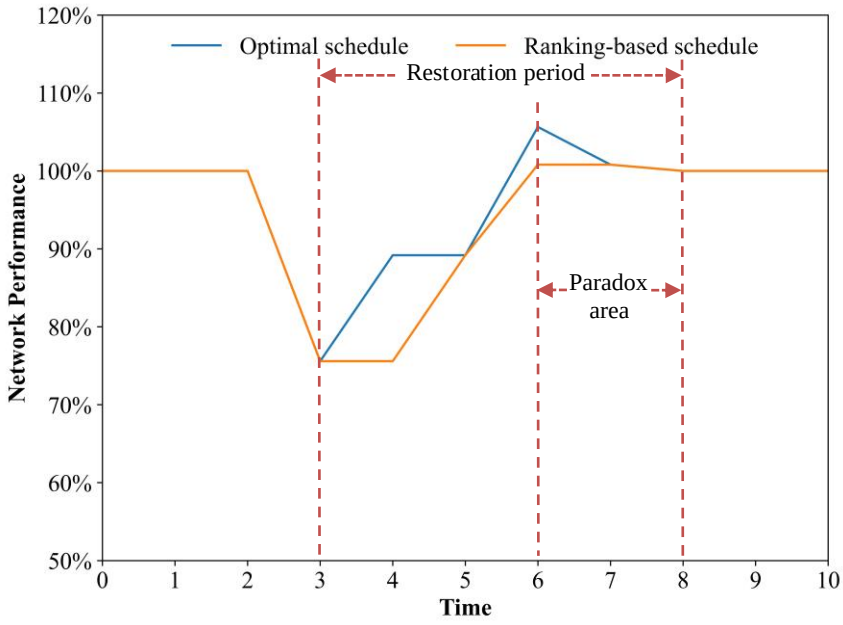


Fig. 6 Changes in network performance

For comparison, one of the most commonly used topology-based indicators, network efficiency $E(G)$, is also applied to quantify system resilience. In network graph theory, network efficiency is calculated as the average of the inverse of the shortest path distances between all pairs of nodes [3, 15] and is mathematically expressed as

$$E(G)=\frac{1}{|N|(|N|-1)}\sum_{i\neq j\in N}\frac{1}{d_{ij}} \tag{15}$$

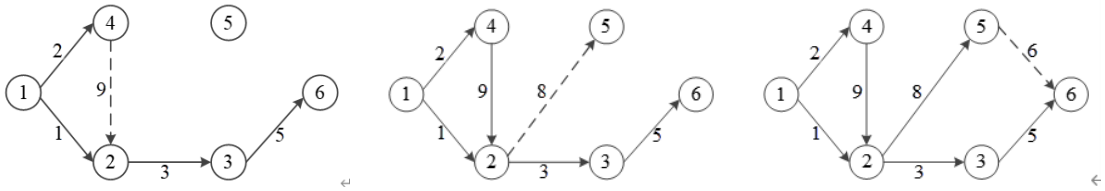
where $|N|$ is the total number of nodes and d_{ij} denotes the shortest path distance between node *i* and node *j*. The upper-level formulation is subsequently transformed into the problem of maximising the total network efficiency within the minimum restoration period, which is mathematically characterised as follows:

$$\max_{\mathbf{s,v,y}} \sum_{\tau \in H} \frac{1}{|N|(|N|-1)} \sum_{i \neq j \in N} \frac{1}{d_{ij\tau}} \tag{16}$$

where $d_{ij\tau}$ represents the d_{ij} in period τ. The results obtained from $E(G)$ and the UNPM are presented in Table 4. The main difference in the total network efficiency between the two solutions lies in periods 3-5. To better explain the solutions, in Fig. 7, we plot the network recovered from the schedule obtained from $E(G)$ in each period; the results obtained from the UNPM are shown in Fig. 5 (a). The solid and dotted lines in the figure below have the same meaning as described above. Note that by recovering link 8 in the second period, the shortest paths between node pairs 1-5, 2-5 and 4-5 traverse the newly recovered link simultaneously. The network efficiency is thus increased because the shortest path distances between these node pairs are no longer considered infinite. In contrast, as shown in Fig. 5 (a), the newly recovered link 4 is on the shortest path between two node pairs in period 3; however, the length of the link itself is great. As a result, the effect of early restoration of link 4 is not significant. A comparison of Fig. 7 with Fig. 5 (a) reveals that the first recovered link is 9 and the third is link 6, indicating that the link restored in period 2 leads to a change in network efficiency in period 4. Finally, two shortest paths traversing the newly recovered link 7 are formed in period 5 under the schedule based on $E(G)$ compared with the schedule based on the UNPM, which reduces the shortest path distances between the node pairs and results in greater network efficiency. In summary, when network efficiency is used as a resilience indicator, the optimal restoration sequence is always 9-8-6-7-4.

Table 4 Restoration results based on $E(G)$ and the UNPM

		Restoration period					Total network efficiency
		1	**2**	**3**	**4**	**5**	
Schedule based on $E(G)$	Link under reconstruction	9	8	6	7	4	0.2380
	$E(G)$ in each period	0.0033	0.0373	0.0540	0.0687	0.0747	
Schedule based on UNPM	Link under reconstruction	9	4	6	8	7	0.1886
	$E(G)$ in each period	0.0033	0.0373	0.0393	0.0400	0.0687	



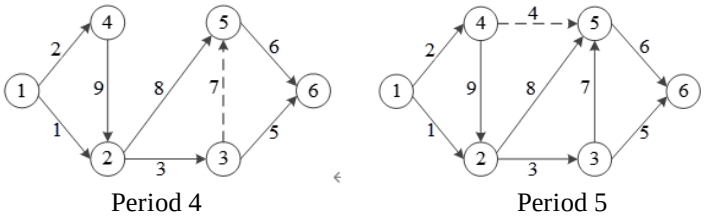


Fig. 7 Networks from schedules obtained according to network efficiency

4.1.2 Effect of the demand level on the paradox

To illustrate the effect of travel demand on the restoration schedule, the proposed model is tested under different demand scenarios. We introduce a demand scale parameter, ρ, for varying the demand level, which is set to 0.5, 1.0, 1.5 and 2.0. The resulting restoration schedules are presented in Table 5. The restoration schedule obtained under ρ = 1.0 is the same as that obtained under ρ = 1.5, while the schedules obtained under ρ = 0.5 and ρ = 2.0 are both distinctive. On the one hand, the results demonstrate that the demand level indeed affects the schedule. On the other hand, they show that the effect can be significant or minor. For example, when ρ = 0.5, links 9 and 4 are the last two links to be prepared. In contrast, the two links are prioritised when ρ is 1.0 or greater.

Table 5 Restoration scheduling under different demand scenarios

Demand level	Restoration period					Total travel time
	1	2	3	4	5	
ρ = 0.5	6	8	7	9	4	1100.2
ρ = 1.0	9	4	6	8	7	2895.7
ρ = 1.5	9	4	6	8	7	5045.2
ρ = 2.0	9	4	6	7	8	7670.5

* ρ is the demand scale parameter; e.g., ρ = 1.0 is the normal demand, and ρ = 2.0 is twice the normal demand.

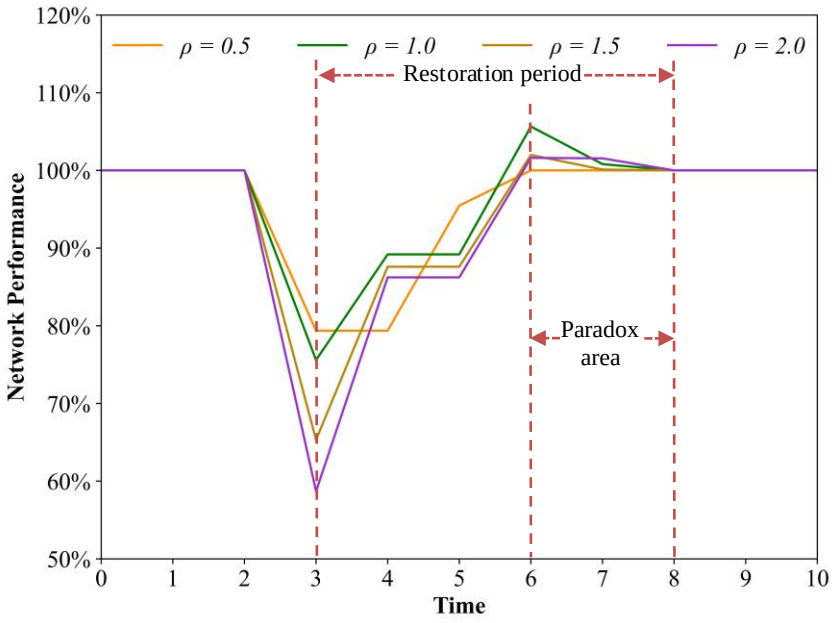


Fig. 8 Changes in network performance under different demand levels

To further analyse the effect of traffic demand on the occurrence of the paradox during the restoration of the disrupted network, we also trace the changes in network performance in each period, as illustrated in Fig. 8, which shows the effect of demand on the occurrence of the paradox. First, by examining the network performance values in time period 3, it is concluded that after a disaster occurs, the higher the demand level is, the more the network performance decreases. Then, between time periods 3 and 6, the recovery trajectory varies for different demand levels, implying that a monotonic relationship may not exist between the demand level and the recovery trajectory. Interestingly, by comparing the changes in the network performance measurements between periods 6 and 7, it can be seen that when a paradox occurs, higher demand does not necessarily mean a greater drop in network performance. To clarify this phenomenon, the detailed travel times are plotted in Fig. 9. In the figure, a dotted line represents the travel time without any disruption. When $\rho = 1.0$, the travel time gap between the two periods is the largest.

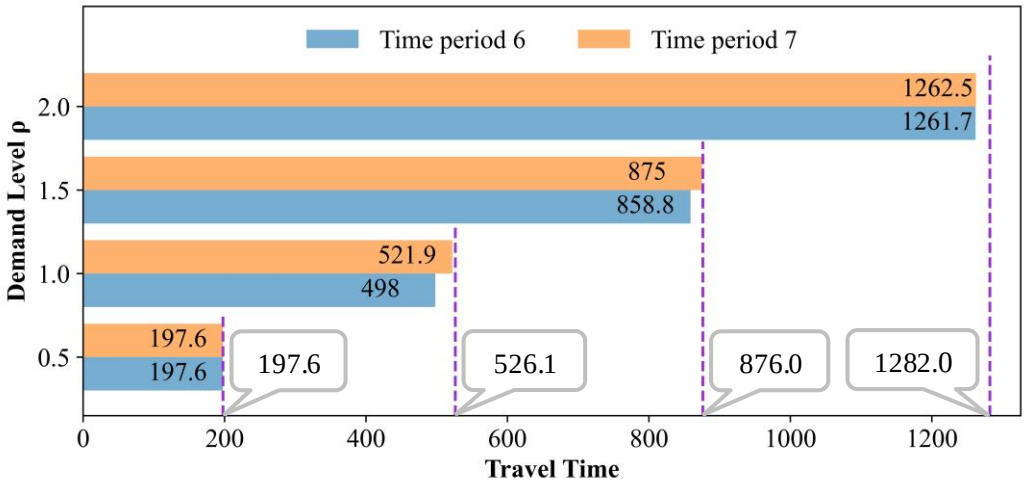


Fig. 9 Travel time in time periods 6 and 7

4.1.3 Effect of tolerance on the paradox

Our optimal restoration schedule depends on the flow values determined by the lower-level model, which are affected by the tolerance of the termination condition of the lower-level problem. Therefore, this subsection presents the numerical application to determine the effect of the tolerance. In this test, it is assumed that each link requires two periods to be rebuilt, and the maximum number of links to be reconstructed is 2 in each period.

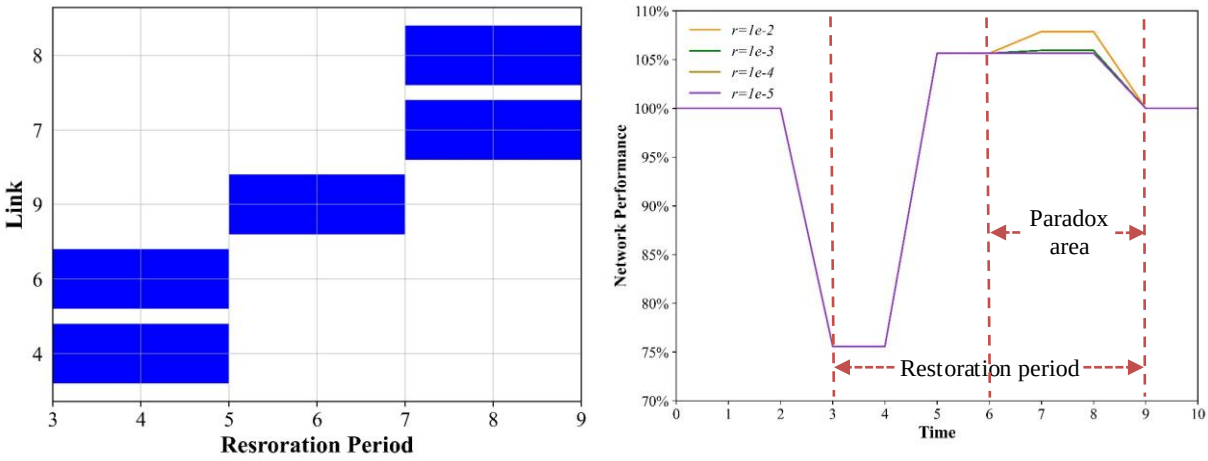
The numerical results with respect to the optimal values and restoration schedules, considering the tolerance factor of lower-level problems, are summarised in Table 6. Although the tolerance and the total system travel time vary widely, the restoration schedules are exactly the same. This result, while specific to small-scale networks, also holds for real networks as long as the tolerance meets a certain threshold. In addition, the results reveal that the objective values gradually converge with decreasing tolerance.

Table 6 Restoration scheduling under different tolerance conditions

Tolerance	Restoration period						Total travel time
	1	2	3	4	5	6	
1.0e - 2	4,6	4,6	9	9	7,8	7,8	3363.4
1.0e - 3	4,6	4,6	9	9	7,8	7,8	3381.3
1.0e - 4	4,6	4,6	9	9	7,8	7,8	3383.8

1.0e - 5	4,6	4,6	9	9	7,8	7,8	3383.8

In addition to presenting the optimal restoration schedule under different tolerance conditions, we plot the changes in network performance in each period, as shown in Fig. 10. Fig. 10 illustrates that, for the optimal schedule, only one link is selected to be recovered during time periods 5 and 6. This indicates that although there are enough resources for repairing multiple links simultaneously in one period, all of them may not necessarily be used. One reason is the possibility of inducing the Braess paradox when a certain link is covered, which results in a longer total travel time. This can be verified by the changes in the total travel time. The network performance increases to 105.6% following the full recovery of links 4 and 6. Afterwards, as the tolerance value decreases, it does not change until the restoration activity is complete, implying that link 9 cannot divert flow from link 1 in periods 7 and 8. Finally, by recovering links 7 and 8 in the last two periods, the network performance drops to 100% because the newly recovered links 7 and 8 make a negative contribution. The above observation verifies that if a link is completely disrupted, the paradox can still occur under multiresource constraints.



(a) Restoration sequence (b) Recovery trajectories

Fig. 10 Characteristics of the optimal restoration schedule

4.2 Nguyen-Dupuis Network

In this example, the proposed method is applied to solve the problem of restoring the Nguyen–Dupuis Network (see Fig. 11) to test the algorithm's performance. Figure 11 shows four OD pairs, namely, OD pairs 1-12, 1-13, 2-12 and 1-13, the travel demands of which are set to 100, 300, 400 and 200, respectively. The detailed attributes of each link in the normal state are listed in Table 7. Then, the travel time on link a in period τ is calculated by the following Bureau of Public Roads (BPR) function:

$$t_{a,\tau}=t_{a,\tau}^{0}\left[1+\alpha\left(\frac{v_{a,\tau}}{Q_{a,\tau}}\right)^{\beta}\right],\forall a\in A,\tau\in H \tag{17}$$

where $Q_{a,\tau}$ denotes the traffic capacity of link a in period τ and $t_{a,\tau}^{0}$ represents the free-flow travel time on link a in period τ. α and β are constant parameters equal to 0.15 and 4, respectively. Table 8 shows

that the length of each potential disrupted link l_a is an important parameter affecting the recovery periods required to restore the disrupted link.

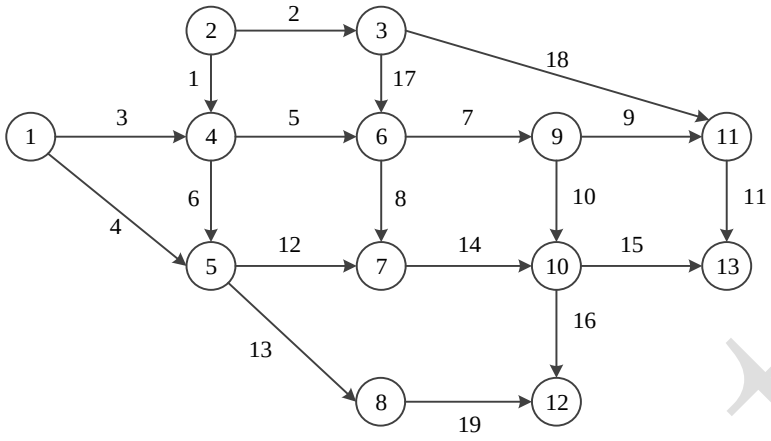


Fig. 11 Nguyen-Dupuis Network

Table 7 Link attributes of the Nguyen-Dupuis Network

Link no.	Free-flow travel time	Capacity	Link no.	Free-flow travel time	Capacity
1	7	300	11	9	500
2	9	200	12	10	550
3	9	200	13	9	200
4	12	200	14	6	400
5	3	350	15	9	300
6	9	400	16	8	300
7	5	500	17	7	200
8	13	250	18	14	200
9	5	250	19	11	200
10	9	300			

Table 8 Potentially disrupted links and their lengths

Link no.	Length	Link no.	Length
2	9	12	10
4	12	13	9
9	5	17	7
11	9	19	11

The above experiments are based on a situation in which the disrupted links are completely closed; however, in many cases, links may still be running but with a decreased capacity and free-flow speed. Thus, by focusing on partial closure, the following subsections develop the optimal recovery sequence and explore the Braess paradox during the restoration process.

After a given disruptive event, the eight disrupted links are assumed to be partially disrupted at the same damage level m_a. Therefore, the capacity and free-flow speed of link *a* are reduced by m_a. Additionally, assume that the maximum number of links to be reconstructed is 2 in each period and that the recovery time required for restoring link *a* is $\lceil \eta \times m_a \times l_a \rceil$, where $\lceil \cdot \rceil$ is the ceiling operator and η is the manipulated variable. To facilitate the analysis, the manipulated variable is set as $\eta = 0.5$.

4.2.1 Algorithmic performance

According to our previous tests, the proposed bilevel model can determine the optimal restoration plan and verify the occurrence of paradoxes if multiple links are completely disrupted. Hence, in this section, we examine the effectiveness of the proposed model and algorithm when links are partially disrupted. The damage level m_a is set to 0.3.

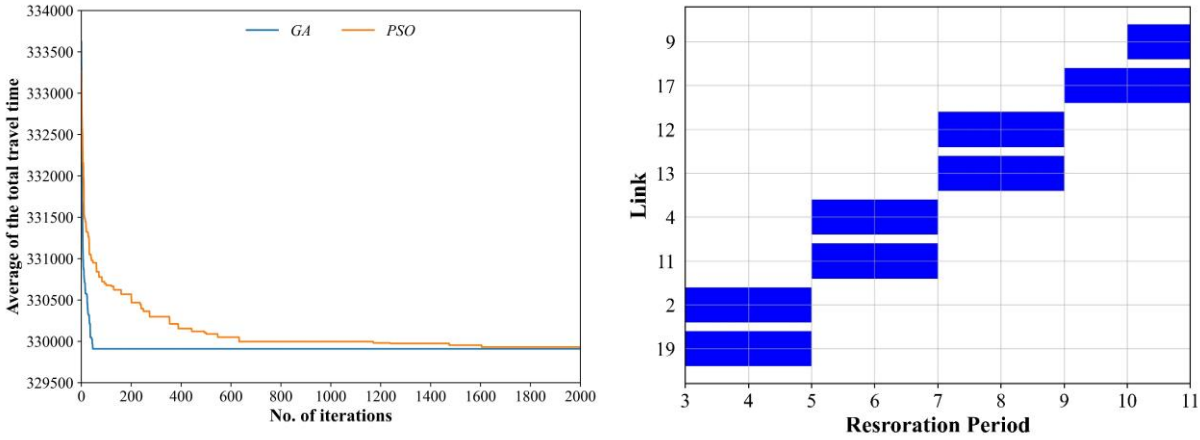
To better illustrate the performance of the GA, the brute force method and particle swarm optimisation algorithm (PSO) are used for comparison. The former is applied to enumerate all the possible restoration schedules, while the latter is performed in 20 runs with the same random seeds as our proposed GA. For a fair comparison, the two metaheuristic algorithms use the same solution representation, population size and termination criterion. In PSO, the two acceleration constants, the initial inertia weight and the largest velocity are set to 2, 0.9 and 30, respectively, and the initial positions of the particles are randomly generated. Additionally, for better comparison, we appropriately adjust the tolerance of the UE on the original basis.

The numerical results with respect to the total travel time, CPU time and maximum number of best solutions obtained out of 20 runs for the different methods are summarised in Table 9. The average total travel times are almost the same; however, the GA obtains a better total travel time than does the PSO. Regarding the CPU times, the average CPU time for the 20 runs of the GA is 781.7, which is approximately 14% less than that of the PSO algorithm. According to the results in Table 9, the GA succeeded in obtaining the best solutions in each of the 20 runs, with a success rate 20% higher than that of PSO.

Table 9 Comparison between the GA and the compared methods

	Brute force	PSO	GA
Average of total travel time	329909.1	329930.8	329909.1
Average CPU time (s)	43319.1	892.8	781.7
Best optimal total travel time	329909.1	329909.1	329909.1
Number of best solutions obtained	20	16	20

The restoration sequences and the convergence plots are presented in Fig. 12. The GA begins to converge at the 46th generation, which is much earlier than the convergence of PSO. This indicates that the global optimal solution can be obtained by using metaheuristic algorithms. Nevertheless, the GA outperforms the PSO algorithm in finding high-quality solutions within a small number of iterations. Moreover, we can see that only link 17 is selected to be recovered during time periods 9 and 10. The results verify that the available resources may not necessarily be exhausted in each period despite the partial damage to multiple links.

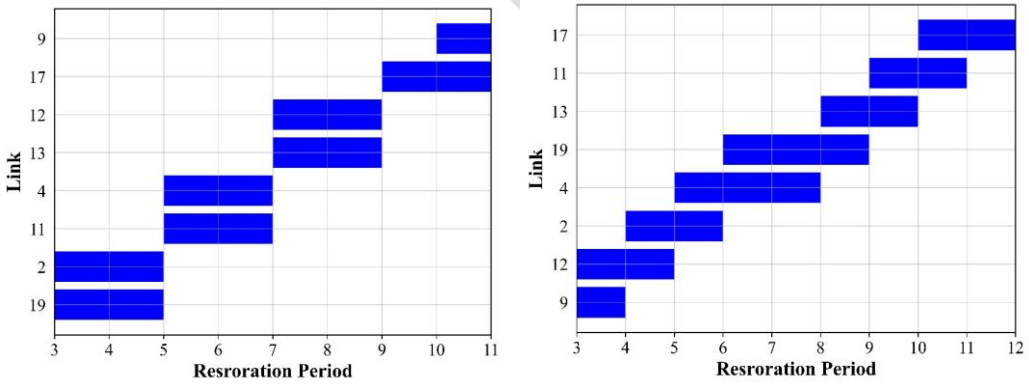


(a) Convergence of the algorithms (b) Restoration sequence

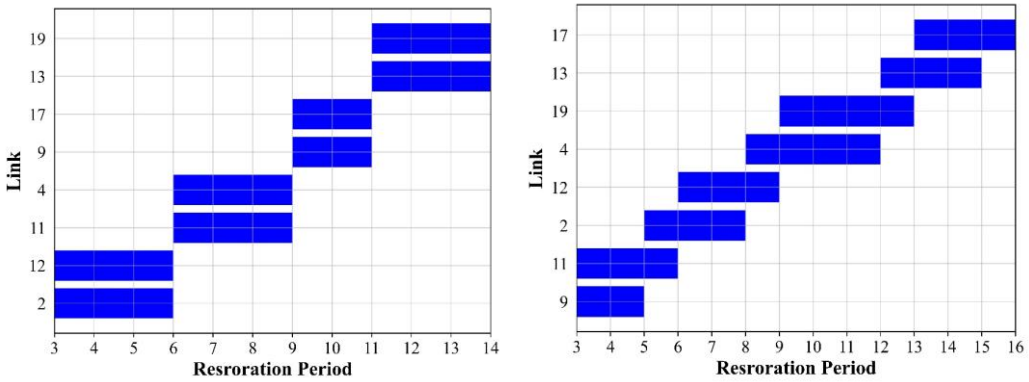
Fig. 12 Optimal results for $m_a = 0.3$

4.2.2 Effect of damage level on the paradox

The damage level m_a is determined by the disruptive event. Keeping the other parameters constant, Figs. 13 and 14 present the resulting optimal restoration schedules and the changes in network performance under different damage levels. In this experiment, the damage level m_a is increased from 0.3 to 0.8. As mentioned in Section 4.1.1, we also plot some additional normal periods before and after the corresponding restoration periods. It is obvious that the restoration period is longer for a larger m_a, as shown in Fig. 13. In addition, the figure shows that the restoration sequences vary with increasing m_a. As expected, not all the available resources in each period are consumed, as observed in Figs. 13 (a) and (f).



(a) $m_a = 0.3$ (b) $m_a = 0.4$



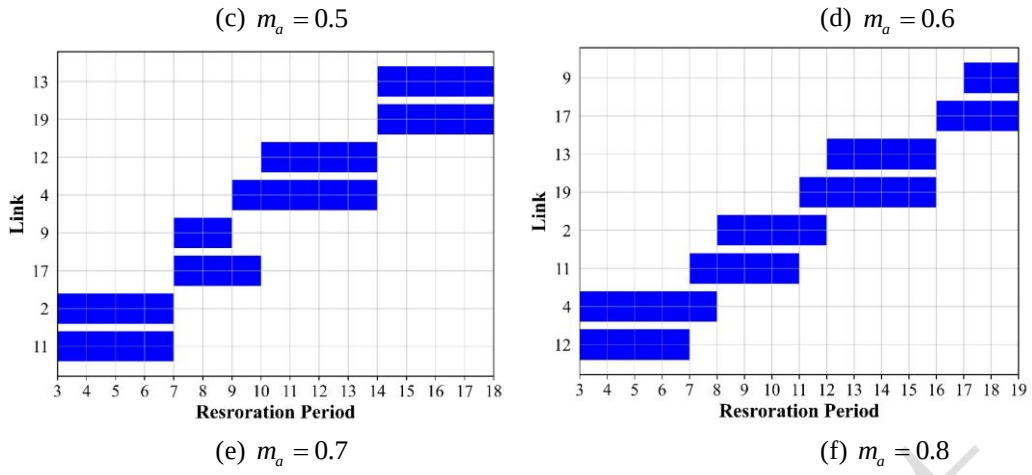
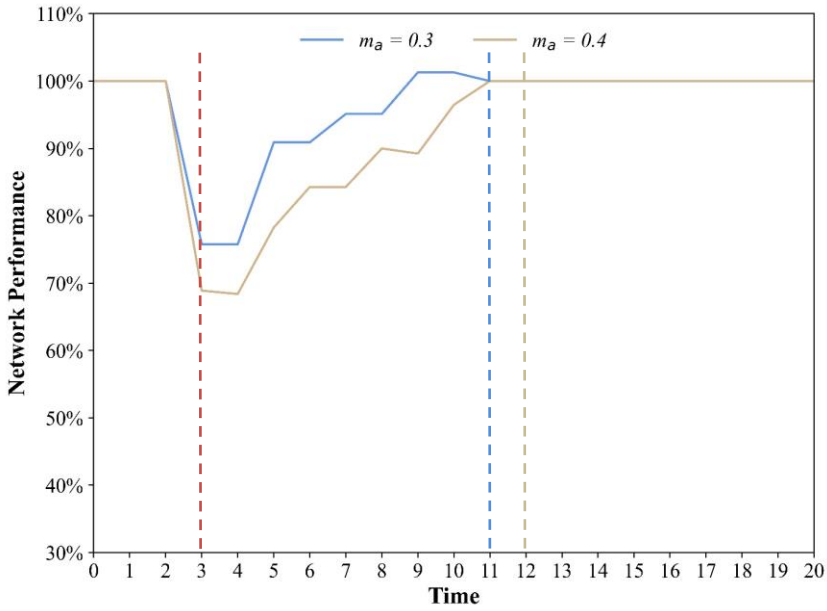


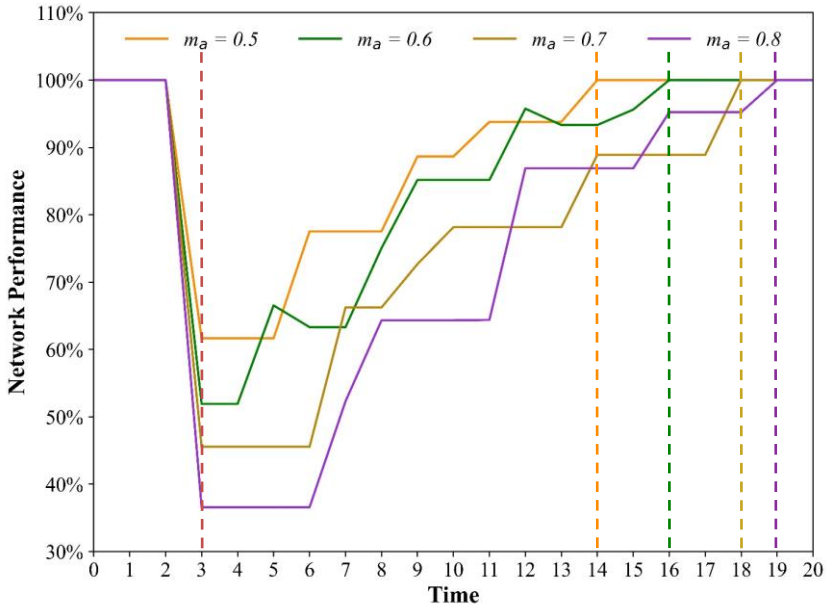
Fig. 13 Restoration schedules under different damage levels

In Fig. 14, a red dotted line represents the beginning of a restoration period, while the other coloured dotted lines represent the ends of the corresponding restoration periods. Initially, the higher the damage level is, the worse the network performance. Fig. 14 (a) shows that for the optimal restoration schedule under a partially damaged state, the paradox can still occur. The network performance increases to 101.3% following the full recovery of links 12 and 13 when $m_a = 0.3$, and after period 10, it drops to 100%. Similar results are found when the damage level is increased to 0.4. For $m_a = 0.4$, the network performance peaks at 100.1% in period 11, after which the Braess paradox also arises. At damage levels of 0.5, 0.7 and 0.8, network performance improves progressively along with restoration activities. A full picture of the network performance with respect to m_a and τ is shown in Fig. 14 (b). It is notable that the performance curve at $m_a = 0.6$ has a reverse mutation in both periods 5 and 12, which clearly does not occur at the other three damage levels. The main reason for this is that, at this damage level, the main routes in use pass through the newly recovered links 11 and 19, and the traffic flows on the last restored links 9 and 4 exceed the initial capacity.

In general, the changes in network performance, similar to the formulation of the restoration schedule, are related to the damage level. Paradoxically, the recovery trajectory varies with time and is not always negatively correlated with damage levels. Fig. 14 (b) demonstrates this counterintuitive phenomenon in detail. As shown in the figure, for $m_a = 0.5$, the network performance in time period 5 increases to 61.7%, while for $m_a = 0.6$, it can reach 66.6%. Furthermore, we note that the performance measurements between periods 12 and 13 and between periods 16 and 17 obtained under $m_a = 0.7$ are much lower than those obtained under $m_a = 0.8$. This result reveals that a higher damage level does not necessarily mean a lower network performance. It can be concluded that the damage level influences the restoration schedule and the recovery trajectory. Nevertheless, a monotonic relationship may not exist between the damage level and the recovery trajectory. This counterintuitive phenomenon could be considered a paradox. The paradox in this section also implies that an increase in damage level may sometimes substantially degrade performance.



(a) Slight damage



(b) Moderate damage and severe damage

Fig. 14 Changes in network performance under different damage levels

4.2.3 Effect of resource level on the paradox

The effect of the resource level on the paradox is examined using the change in the maximum available resources from 1 to 4 in each period. In other words, this subsection addresses the problem of multiple-link restoration scheduling and examines the effect of resource constraints. In this experiment, we set the damage level to 0.3. The optimal schedules and changes in network performance are illustrated in Figs. 15 and 16, respectively. The dotted lines have the same meaning as in Section 4.2.2.

A comparison of the restoration schedules in Fig. 15 reveals that the restoration sequences under the different resource levels are completely consistent. This means that with further increases in resource levels, the restoration period is indeed reduced but does not change the restoration sequence. The incomplete

utilisation of resources is also observed, as shown in Figs. 15 (b) to (d), which is consistent with the view presented in the previous sections.

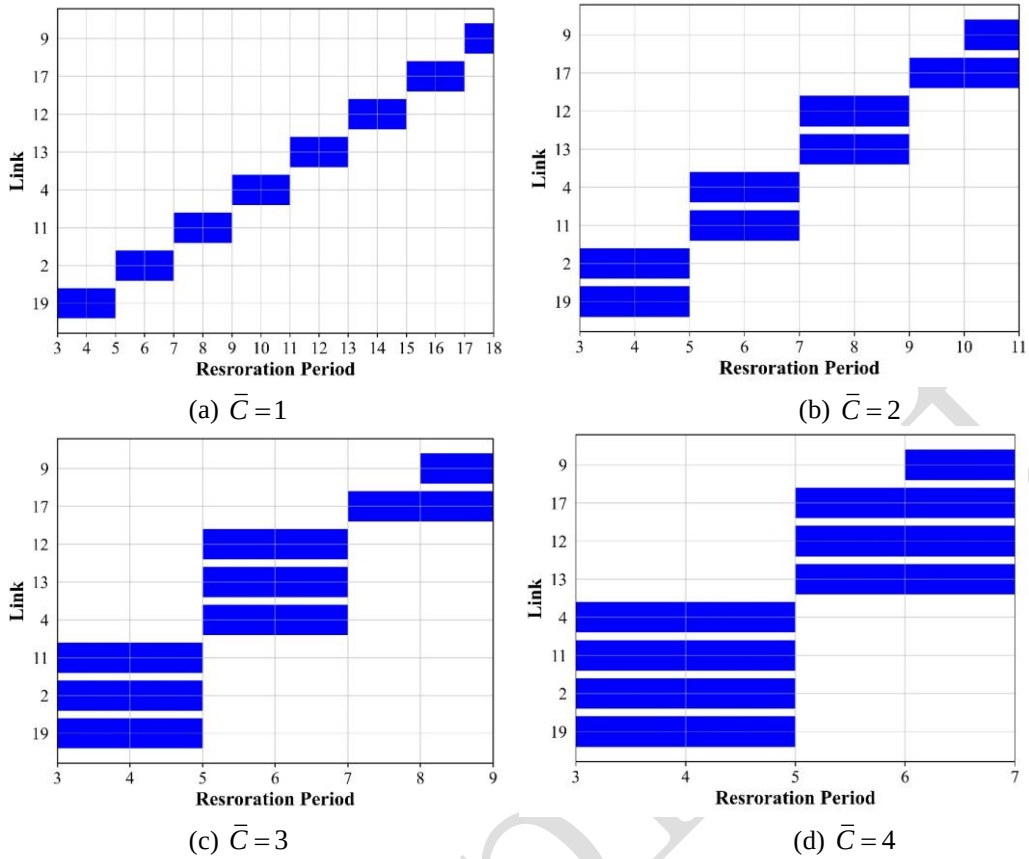


Fig. 15 Restoration schedules under different resource levels

Fig. 16 shows that as the time period progresses, the network performance generally shows an upwards trend, and for $\bar{C}=3$, there are two peaks in the curves where the performance is better than that without any disruption. However, during the restoration process, the network performance for $\bar{C}=4$ is less than 100%. This implies that the Braess paradox is more likely to occur when the resource level is lower. Furthermore, by comparing the changes in the network performance measurements between periods 6 and 10 and between periods 15 and 16, it can be seen that when a paradox occurs, a higher resource level does not necessarily indicate better network performance.

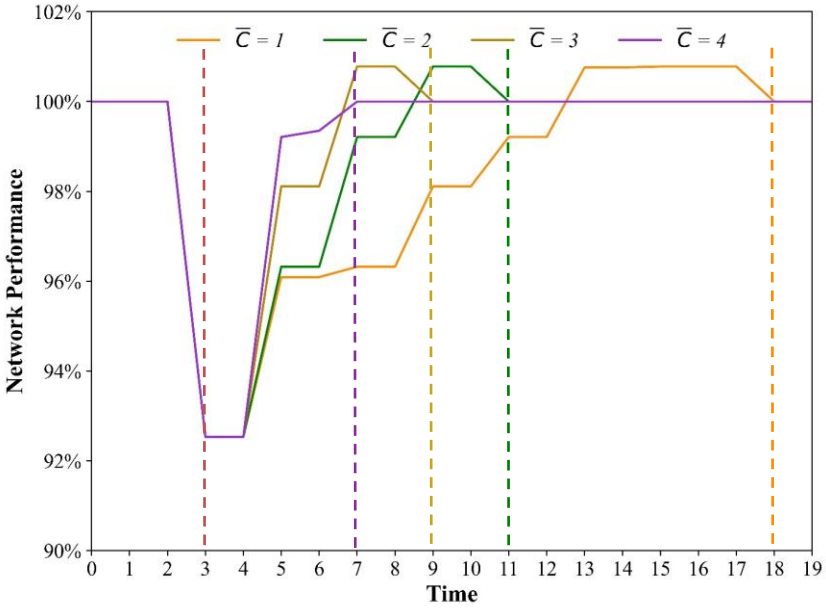


Fig. 16 Changes in network performance under different resource levels

5 Conclusions

This study analyses the classic Braess paradox in the multiperiod restoration scheduling problem with limited resources, which mainly affects the number of links that can be repaired simultaneously within one period. A bilevel model is formulated, where the upper-level problem is to determine the optimal restoration scheduling considering resource constraints, while the lower-level problem is to determine the traffic assignment model that captures passengers' response to changes in network topology and capacity as a result of restoration activities. To investigate the Braess paradox under the optimal solution, a modified GA method is implemented to search for the optimal restoration schedule for the upper-level problem. The Frank–Wolfe algorithm is adopted to solve the lower-level traffic assignment model. The main findings from the numerical studies include the following: 1) The optimal restoration schedule differs from the schedule based on the link importance measurement. This contradicts our intuition that a more critical link should be repaired first but justifies the necessity of developing a mathematical model to determine the optimal restoration schedule. 2) The Braess paradox can occur during the restoration period, meaning that without repairing all disrupted links, network performance is better than that before the disruption or after a new link is recovered. 3) When multiple links can be repaired within the same period, the Braess paradox can still occur, and not all resources need to be exhausted in the optimal rehabilitation schedule. 4) The optimal restoration schedule is related to the damage level, but this does not mean that a large value of m_a leads to much worse network performance. 5) We compare the optimal schedules under different resource levels, and the results indicate that the restoration sequences are identical. Remarkably, a higher resource level does not imply a greater improvement in network performance when the Braess paradox occurs.

This study yields various directions for future research. First, the study only examines the optimal restoration schedule for road networks, assuming that the time and the resources required to recover each disrupted component are given. It would be interesting to analyse network performance under uncertainty. Second, the lower-level model is the static deterministic traffic assignment model, which can be extended

and enhanced using a stochastic or dynamic traffic assignment model. Third, this study considers only a single resilience indicator to quantify network performance. Considering the diversity of network performance measurements, developing a multiobjective bilevel model is a necessary research direction. Fourth, although the proposed methodology is applied to restoration scheduling problems for transportation networks, it can be useful for other research areas involving scheduling problems. In future studies, it would also be worthwhile to consider the application of these methods in other contexts, such as power distribution network restoration projects and road network capacity improvement projects.