Title Page

Title:

Algorithms for restoring disaster-struck seaport operations considering interdependencies between infrastructure availability and repair team assignments

Authors:

Febri Zukhruf[a*], Chandra Balijepalli[b], Russ Bona Frazila[a], Taufiq Suryo Nugroho[a], Irma Susan Kurnia[c]

Affiliations:

[a] Faculty of Civil and Environmental Engineering, Institut Teknologi Bandung, Indonesia
[b] Institute for Transport Studies, University of Leeds, United Kingdom
[c] Sukarasa Community Health Center, Bandung City Government

Address:

[a] Faculty of Civil and Environmental Engineering, Institut Teknologi Bandung, Jl. Ganesha No 10, Bandung, 40132, Indonesia
[b] Institute for Transport Studies, University of Leeds, Leeds LS2 9JT England Indonesia
[c] Sukarasa Community Health Center, Jl. Gegerkalong Hilir No. 157, Bandung, 40153, Indonesia

Email Address:

Febri Zukhruf (Corresponding Author): febri.zukhruf@itb.ac.id
Chandra Balijepalli : N.C.Balijepalli@leeds.ac.uk
Russ Bona Frazila: frazila@yahoo.com
Taufiq Suryo Nugroho: taufiq.suryo@gmail.com
Irma Susan Kurnia : doc_suzan@yahoo.com

Acknowledgments:

This work was supported in part by Institut Teknologi Bandung through ITB research grant (PN-2020).

Declaration of interest :

Declarations of interest: none

This paper can be cited as below:

Zukhruf, F., **Balijepalli, C.**, Frazila, R.B.,Nugroho, T. and Kurnia, I. (2022) Algorithms for restoring disaster-struck seaport operations considering interdependencies between infrastructure availability and repair team assignments, *Computers and Industrial Engineering,* Volume 175 https://doi.org/10.1016/j.cie.2022.108894

Algorithms for restoring disaster-struck seaport operations considering interdependencies between infrastructure availability and repair team assignments

Abstract

This paper presents new algorithms for restoring seaport operations after a disaster and develops a model considering interdependencies to select an efficient course of action. The model prioritises the infrastructure to be repaired, identifies the equipment required and the number of repair teams to be deployed. This paper develops a new dynamic programming model to assign multicrew repair teams and shows that the solution is exact. This paper then develops a new variant of the Hungarian Algorithm by embedding an exploitation-exploration strategy to obtain an approximate solution for large-sized assignment problems. Furthermore, this paper solves the restoration problem in totality by accounting for interdependencies between marine/land-side infrastructure/equipment and repair team assignments. This paper also develops a new variant of Genetic Algorithm based on a deletion-mutation technique and explores reducing the computation time involved in solving optimisation problems. This paper applies the principles laid out to restore Pantoloan seaport in Indonesia which was struck by a tsunami. The approximate solution obtained by the extended Hungarian Algorithm for small problems is quicker and matches with the exact solution obtained by the new dynamic programming. In case of large-sized problems, the extended Hungarian Algorithm has been found to arrive at a solution which allows reopening the seaport 48% sooner than the other algorithms. The new variant of Genetic Algorithm outperforms the Genetic Algorithm with Local Search, needing *only 40% of the computation time* and the solution found to be particularly stable too.

Keywords: Operations research in disaster relief; Dynamic programming; Seaport restoration; Genetic algorithm; Hungarian algorithm.

1. Introduction

Aid supplies become primary concern when a disaster strikes as they are essential for avoiding loss of life (Ahmadi *et al*., 2015). However, transportation network poses a major challenge in delivering supplies due to the disruption caused by a disaster (Maya Duque *et al*., 2016). Therefore, the *network restoration problem* deals with assignment of equipment and/or teams for repairing the disrupted infrastructure, the aim of which is to seek an optimum schedule to restore the degraded parts of infrastructure. However, a classic feature of those studies means almost all of them are focused on road networks (see Çelik, 2016). Rarely ever seaports feature in disaster recovery studies despite their key role in reaching supplies in bulk to the affected areas.

Research body on disaster recovery acknowledges that they destroy not only the road networks but also seaport terminals (AHA Centre, 2018). For ensuring a smooth transfer of goods between sea and land transportation modes, a seaport is generally equipped with several pieces of physical infrastructure, for instance, breakwaters, quay, dockyard, stacking yard, and internal/external road network. Furthermore, seaport also includes loading/transport equipment such as cranes, fork-lift trucks, which are essential to move the goods between locations. Interaction among them defines the key nature of seaport operations, in which, disruption to any individual entity will result in an intertwined effect, adversely affecting the entire seaport operations. For instance, to carry the goods from the vessel to stacking yard, transport vehicles are essential, which practically rely on using the internal road network (Please see **Figure 1**). Thus, any disruption to road network will increase the transport delays potentially influencing the operational processes of the entire seaport. A similar situation arises even in the case of road network remaining in an excellent condition but without any unloading equipment being available to empty the vessel. Therefore, in restoring a seaport after a disruption, *interdependency* between the equipment/ infrastructure plays a critical role which needs to be properly accounted for.

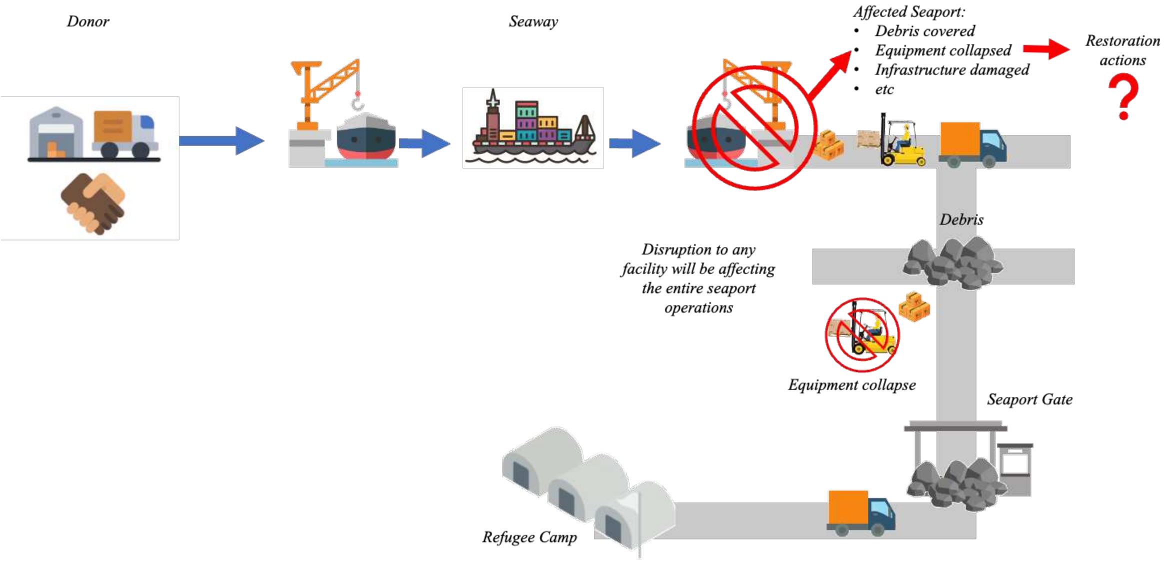

Figure 1: Illustration of seaport restoration problem

There are three different levels of decisions needed to restore a network. The first level is to select the priority infrastructure that must be repaired, and the next level is to assign/schedule a repair team (henceforth, RT) to the selected infrastructure. The final level is to decide the shortest operational route to reach the selected infrastructure location. In the literature, this decision-making process is described as the integrated network design/ scheduling problem (INDS), which combines network design with a restoration model (Nurre *et al.*, 2012). We note that the INDS problems in the literature are limited as they consider single crew (Maya Duque *et al.*, 2016) and neither they attempt to account for the interdependencies involved. This paper fills the gap in literature and develops a new model of restoration, considering the interdependency of operations between marine infrastructure, landside infrastructure and assigning/scheduling of repair teams. This paper develops a new dynamic programming model (DP) for handling restoration problems by considering multicrew assignment. The simultaneous work of multiple crews creates an opportunity to represent the restoration activities realistically, which is not possible with the single crew DP reported in the literature. The multicrew DP, however, can handle only small-sized problems due to the extensive computation involved. To handle large-sized problems, we extend the Hungarian Algorithm by adopting an exploitation-exploration based strategy with a continuous updating of the availability of repair teams and the status of the infrastructure.

Consideration of interdependencies adds complexity to decision-making process because of a very large number of feasible combinations to be evaluated in an efficient manner. Such complexities can be viewed as the core of an optimisation problem, for which we need to find a feasible solution. Metaheuristic approaches are generally found suitable for tackling complex optimisation problems, including Genetic Algorithm (GA) (Guo *et al*., 2021) though they are computationally intensive. This paper explores reducing the computation time involved in solving complex optimisation problems and proposes new variants of GA by modifying the mutation steps and adding an adaptive mutation rate.

Thus, the main aim of this paper is to develop an interdependency-focused model to restore the operations of a disaster-struck seaport in an efficient manner considering the interactions between infrastructure and repair teams. The specific contributions made by this paper are threefold – (i) a new dynamic programming algorithm to assign multicrew repair team which can efficiently deal with small-sized restoration problems, (ii) an extended Hungarian Algorithm to solve large-sized restoration problems; and (iii) a new variant of GA to improve the computational efficiency of optimisation. This paper, thus, adds to the growing body of research on disaster recovery of transport systems and helps the authorities in planning for restoration in real life facilitating the movement of aid supplies. Researchers also will benefit from the extended Hungarian Algorithm and new variants of GA to improve the efficiency of optimisation methods.

The remainder of the paper is structured as follows. Section two reviews the past studies conducted, followed by a mathematical modelling framework specified in the third. The case study in section four illustrates the application of principles to recover Pantoloan seaport in Indonesia describing the results of the new dynamic programming, the extended Hungarian Algorithm and the new variant of GA-based algorithms. Finally, section five concludes the paper.

2. Literature review

2.1 Network Restoration

A considerable number of researchers have proposed methods for tackling transport network disruption from a pre-disaster perspective (i.e., preparation and mitigation), or a post-disaster perspective (i.e., restoration and recovery). The pre-disaster perspective relies on identification of critical network elements for planning/strengthening (e.g. Bell *et al.*, 2017), whilst the post-disaster effort focuses on restoring the state to a normality as soon as possible (Lu *et al.*, 2016; Sanci and Daskin, 2019). Despite the pre-disaster planning involved, the impact of the disaster itself is unavoidable, which motivates an increasing number of post-disaster studies. The studies traditionally comprise of several problems, including the *last mile distribution problem* (Ferrer *et al.*, 2018), the *facility location problem* (Chen and Yu, 2016; Oliveira *et al*., 2019), and the *network restoration problem* (Maya Duque *et al*., 2016; Morshedlou *et al*., 2018).

In the past, network restoration used to be planned manually based on the knowledge of experienced decision-makers (Yan and Shih, 2009). However, there are notable developments since in handling the restoration systematically which are motivated by treating it as a scheduling problem (Chen and Tzeng, 1999; Feng and Wang, 2003). The objective function relates to network and assignment performance, for example, maximising the total length of accessible roads (Feng and Wang, 2003), minimising the total travel-time of all travellers, minimising the total reconstruction time (Yan *et al.*, 2014) and/or the idle time for RTs (Chen and Tzeng, 1999). As the restoration process cannot be separated from relief distribution, the model evolved further by integrating the relief distribution into the objective function, for instance, by aiming to minimise the completion time of operation (Yan and Shih, 2009), to minimise delivery time, and/or to maximise the demand satisfied (Liberatore *et al*., 2014).

Notable efforts were made by adding a network design perspective to the restoration models. This perspective emphasises the selection of infrastructure to be restored in a network (Nurre *et al*., 2012). Thus, the model evolved further by integrating three levels of decisions, namely, the restoration location selection decision, the task assignment of various activities to repair teams, and the scheduling of repair teams (Maya Duque *et al*., 2016). The class of problem named as the integrated network design and scheduling or simply the INDS, is formulated as a mathematical optimisation problem. The objective function is constructed to measure the recovery time (Averbakh, 2012), the cumulative flow in the network over a planning horizon (Nurre *et al*., 2012), the sum of flow costs, unsatisfied demand costs, assignment costs (Cavdaroglu *et al*., 2013), and the cumulative weighted distance between the demand and the closest open facilities (Iloglu and Albert, 2018).

Furthermore, the restoration processes have been formulated as vehicle routing problems too, where the main goal is to find the set of disrupted roads to be primarily restored. Consequently, the model builds synchronised routes for a single vehicle or multiple vehicles (Morshedlou *et al.*, 2018) to restore the disrupted roads in the shortest time possible. The routing incorporation has also been considered by Maya Duque *et al*. (2016) addressing the scheduling and routing of a single repair team while optimising the demand accessibility for distributing relief supplies.

Despite the extensive growth in restoration studies, earlier works heavily focused on road networks (Çelik, 2016; Morshedlou, *et al*., 2018), whereas, the disasters are also known to severely damage seaport operations (AHA Centre, 2018). Seaport operations face several uncertainties, thus, typical approaches evaluate the risks by identifying the causal factors, which may be categorised as operational, security, technical, organisational, and natural risk factors (John *et al*., 2014). The natural risk factors include catastrophic events of weather and seismic activity, which gather an increasing attention (Zhang and Lam, 2015), even though those studies are mainly developed in the context of pre-disaster mitigation. Furthermore, in pre-disaster mitigation studies, the economic-related, and resilience-related parameters have commonly appeared which are used for describing the impact of seaport disruption (Zhang and Lam, 2016; Chen *et al.*, 2017).

Based on the literature above, we can point out two major gaps that remain needing attention: i) most restoration studies concentrated on road network restoration while the impacted seaports play a pivotal role in disaster management, thus leaving the scope for carrying out further work; and ii) little evidence is available in the context of post-disaster recovery phase, specifically involving port component restoration for servicing relief distributions.

2.2 Solution techniques for solving network restoration

Several solution techniques have been proposed to tackle the network restoration (See **Table 1**). Maya Duque *et al*. (2016) developed an exact-solution based method (i.e., DP) to solve a small-sized problem. However, their DP can only fit with a single crew of restoration team, which is rarely applicable in practice. Therefore, this paper proposes a new DP that can deal with multiple crews of restoration teams, which is essential in real life practice when restoring a network.

Table 1: Network consideration and solution techniques of the reviewed literature

Authors	Network consideration	Solution techniques
Feng & Wang (2003)	Road network, multicrew	Mathematical programming using a software package
Yan, & Shih (2009)	Road network, multicrew	A heuristic algorithm based on weighting method
Yan *et al*. (2014)	Road network, multicrew	Ant colony system –based hybrid global search algorithm
Kasaei & Salman (2016)	Road network, single vehicle	Mixed integer programming formulation, variable neighborhood search algorithm, variable neighborhood descent algorithm
Maya Duque *et al*. (2016)	Road Network, single crew	Dynamic programming, iterated greedy-randomized constructive procedure
Akbari & Salman (2017)	Road network, multi-vehicle	Mixed integer programming and local search
Morshedlou *et al*. (2018)	Infrastructure network, multicrew	Initial solution pre-processing and feasibility algorithm
Almoghathawi *et al.* (2019)	Infrastructure network	Mixed integer programming
Moreno *et al.* (2020)	Road network, multicrew	Mixed integer programming
Ghannad *et al*.(2021)	Road network	A multiagent reinforcement learning model, and monte carlo simulation

Because of known limitation due to the problem size that can be handled by DP, heuristic algorithms (e.g., greedy algorithm) have been widely used in disaster recovery studies (Maya

Duque *et al*., 2016). Despite its efficiency, greedy heuristic approach, however, could trap into local optimum obtaining a feasible solution which may be far from the global optimal. On the other side, scheduling problems can practically be viewed as an extension to task assignment, if regularly updated. Gao *et al.* (2018) utilised Kuhn–Munkres algorithm, which is also known as the Hungarian Algorithm, for optimising packet scheduling problem in an integrated cellular network. They represent the packet scheduling problem using a bipartite graph, where the Hungarian Algorithm attempts to find a complete match on the graph by gradually considering more and more links. A similar approach could possibly be applied for scheduling the repair team assignment in a restoration model, which remains unexplored to date. This paper modifies the Hungarian Algorithm to handle large-sized seaport restoration problems and avoids local optima with the help of *an exploitation-exploration strategy* added.

Seaport operation involves interaction between physical infrastructure and facilities that are interdependent on each other. Disruption to one facility can potentially cascade to others, triggering serious consequences on the entire operation. In recent times, there has been a growing number of works conceptualising the infrastructure interdependencies in general (e.g., Cavdaroglu *et al.*, 2013; Sharkey *et al.*, 2015), however, there are none involving a seaport restoration, which to date remains unexplored.

The interdependencies involved will generate a significant complexity in decision making, which can be efficiently solved by invoking a metaheuristic approach. There are several types of metaheuristic algorithms, including Genetic Algorithm (Bodaghi *et al*., 2020; Guo *et al*., 2021), Tabu Search (Wei *et al*., 2014), Simulated Annealing (Zhou *et al*., 2019) and Swarm-based Algorithm (Lagaros and Karlaftis, 2011) among others. GA mimics the natural process which is represented by a pipeline process of operators. Although the component GA has been extensively improved, the recombination and mutation operators are practically assembled as primary operators. Recombination mimics the mating process to generate an offspring, while the mutation represents a random change in chromosome. The availability of new molecular technology has made more information available relating to the natural genome process (viz., recombination and mutation), which is significantly different from that previously available. As GA is motivated by natural process, the latest information available could be useful in improving its performance.

We found several gaps remaining in the case of solution techniques, which relate to i) DP that can handle multiple crews of restoration teams, ii) an algorithm to handle large-sized restoration problems, iii) adoption of recent findings from the mutation processes to improve the GA efficiency, and iv) study of interdependencies involving seaport restorations.

3. Modelling framework

The seaport restoration problem has been set up as an Integrated Network Design and Scheduling (INDS) optimisation model which aims at maximising the efficiency of restoration considering the interdependencies between marine/ land-based infrastructure and facilities/teams as described in this section.

Figure 2 outlines the modelling framework of the problem involved. The model aims to restore the seaport operations struck by a disaster. The restoration decision is formulated as a binary optimisation problem which is solved by using a new variant of GA. The GA helps to decide the *number of RTs* for restoring the internal road network, the number of *equipment to be purchased, rented, and repaired* for unloading the relief goods. The number of RTs is used as an input for handling the internal road restoration problem, which involves an integer optimization for assigning the teams to undertake repairs and the routing required to reach the disrupted road locations. A new DP is proposed and it is shown that it can generate an exact solution. A new variant of Hungarian Algorithm is also presented as the DP is not efficient in terms of computation time involved for large-sized problems.

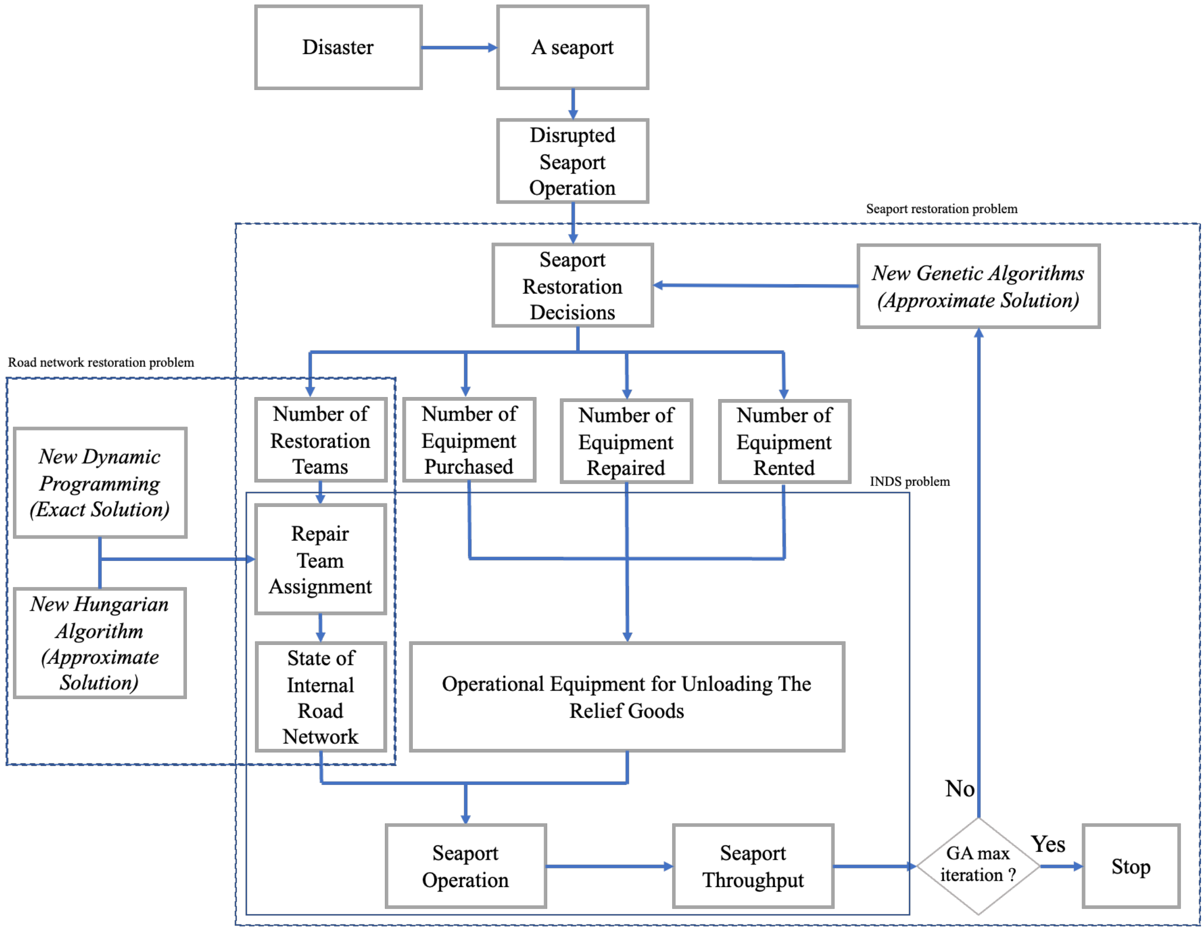

Figure 2: Modelling framework

We, first, start with stating the modelling assumptions involved.

3.1 Modelling assumptions

The proposed INDS for restoring seaport operations, makes the following assumptions:

- Seaport consists of physical infrastructure (e.g., breakwater, quay/dock/jetty, dockyard, stacking yard, internal road network) and equipment facilities (e.g., quay cranes, trucks, forklifts, teams of gangs), each of which has a known capacity.

- Each infrastructure and facility is subjected to disruptions due to natural disasters, in which it is either completely disrupted or undisrupted following a major incident. However, cascading disruptions are not taken into account.
- The infrastructure disruption is modelled from the perspective of a node accounting for the disruption to infrastructure, for example, debris blocking a road/terminal, road cut off, structural damage to the dock, and the collapse of crane etc.
- Each disrupted infrastructure and facility can be restored in a specific period of time at a known cost. Any element of infrastructure remains non-operational until it is fully restored.
- The interdependence among various infrastructure elements and facilities is considered. That is, for a dependent node or facility to become operational, it may require other node(s) from the network also to be operational.
- The number of available repair teams for restoring disrupted infrastructure is known, and each has a defined productivity.
- Each repair team can work on restoring a single disrupted infrastructure element at a time.
- A repair team is not allowed to move from a disrupted location to another unless they complete the restoration job assigned.
- The restoration includes the activities of repairing, renting/outsourcing, and purchasing, in which the number of units available and the capacity is known a-priori.

Table 2: Notation for the integrated optimization model for restoring seaport operations

Set and Indices	
T	set of time periods indexed by *t*

$\boldsymbol{R}$	set of restoration teams indexed by *r,* where $\lvert\boldsymbol{R}\rvert$ denotes the maximum number of teams available
$\boldsymbol{B}$	set of unloading equipment indexed by *b,* where $\lvert\boldsymbol{B}\rvert$ denotes the number of equipment types
$\boldsymbol{N}$	set of operational nodes, where $\lvert\boldsymbol{N}\rvert$ denotes the number of operational nodes
$\boldsymbol{N'}$	set of disrupted nodes, where $\lvert\boldsymbol{N'}\rvert$ denotes the number of disrupted nodes
$\boldsymbol{A}$	set of nodes in network $\boldsymbol{G} = (\boldsymbol{A}, \boldsymbol{L})$
$\boldsymbol{L}$	set of links in network $\boldsymbol{G} = (\boldsymbol{A}, \boldsymbol{L})$
$\boldsymbol{M}$	set of nodes with interdependency
Parameters	
q_t	maximum throughput of seaport in time period *t*
σ	unit value of goods, which is unloaded at seaport
c_{1r}	unit cost of team-*r* for restoring infrastructure
c_{2b}	unit cost for renting unloading equipment-*b*
c_{3b}	unit cost for purchasing new unloading equipment-*b*
c_{4b}	unit cost for repairing unloading equipment-*b*
n_1	number of repair teams assigned for restoring infrastructure
n_{2b}, n_{3b}, n_{4b}	numbers of equipment-*b* possible to be rented, purchased, and repaired, respectively
γ_{1r}, γ_{2b}	total working time for repair team-*r*, and equipment-*b*, respectively
u_{ijt}	freight flow capacity through link- (*i*,*j*) at time period *t*
x_{ijt}	freight flow through link-(*i*,*j*) at time *t*

$prod_{it}$	productivity of node-i at time period t
p_{2b}, p_{3b}, p_{4b}	productivity of equipment-b, which is rented, purchased, and repaired, respectively
E_1	length of GA's allele represents the decision of RTs number
$E_{2b}^{ship}, E_{2b}^{dock}, E_{2b}^{yard}$	length of GA's allele represents the number of equipment-b possible to be rented at ship, dock and yard respectively
$E_{3b}^{ship}, E_{3b}^{dock}, E_{3b}^{yard}$	length of GA's allele represents the number of equipment-b possible to be purchased at ship, dock and yard respectively
$E_{4b}^{ship}, E_{4b}^{dock}, E_{4b}^{yard}$	length of GA's allele represents the number of equipment-b possible to be repaired at ship, dock and yard respectively
$\Gamma_{\chi b}$	unloading capacity of equipment-b, which is rented, purchased, and repaired (i.e., $\chi = 2,3,4$, respectively)
$\Upsilon_{\chi b}(i, j, \boldsymbol{N}')$	Moving time of equipment-b from node-i to node-j by avoiding disrupted node $\boldsymbol{N}'$
ϱ	working hours per day
s_i	restoration duration of node i
τ	maximum time horizon of analysis

Decision Variables

β_1	binary decision variable related to the number of repair teams to be assigned
β_2	binary decision variable related to the number of equipment to be rented
β_3	binary decision variable related to the number of equipment to be purchased

β_4	binary decision variable related to the number of equipment to be repaired
$\theta_{2bit}, \theta_{3bit}, \theta_{4bit}$	binary variables indicating whether or not equipment-*b,* which is rented, purchased, and repaired at node-i is available at time *t*.
ω_{it}	binary variable indicating whether or not node- *i* is operational at time *t*
ς_{it}	binary variable indicating whether or not node-i is to be restored by repair team-*r* at time *t*
z_i	binary variable indicating whether or not node-i is to be restored

3.2 Optimisation model

The objective function in this paper (see Equation (1) and **Table 2** for the notation) represents the efficiency and is defined as the ratio of total throughput value to the total cost of implementing the required set of restoration actions. The value of throughput is calculated by multiplying the total volume of throughput over the horizon period with the value of goods unloaded at seaport. The seaport throughput is a function of infrastructure availability and equipment productivity, which is recovered gradually over a time period. The total throughput is calculated by summing the maximum throughput over all time periods. We used the throughput in our model since it represents the operational activity of seaport in handling the flow of goods, which is considered useful in the seaport disaster impact studies (e.g., Zhang and Lam, 2015, 2016). The throughput can also be viewed as the benefit produced by applying a set of restoration actions which is notionally different to the tangible revenue although both terms may have the same units. Each combination of restoration actions incur expenditure for restoring the infrastructure and/or for providing the equipment (e.g., cranes, forklifts, the crew), possibly by repairing, renting, and purchasing. The estimated benefit is then divided by the cost incurred for investigating the efficiency of the alternative actions.

$$\text{Max}\left(\sum_{t\in T}\frac{q_t\sigma}{\Delta}\right) \quad (1)$$

$$\Delta = \sum_{r=1}^{n_1(\beta_1)} c_{1r}\,\gamma_{1r} + \sum_{b=1}^{|\boldsymbol{B}|} c_{2b}\,\gamma_{2b} n_{2b}(\beta_2) + \sum_{b=1}^{|\boldsymbol{B}|} c_{3b}\,n_{3b}(\beta_3) + \sum_{b=1}^{|\boldsymbol{B}|} c_{4b}\,n_{4b}(\beta_4) \qquad (2)$$

$$\forall\, r \in \boldsymbol{R}, t \in \boldsymbol{T}, b \in \boldsymbol{B}$$

Equation (2) computes the total cost of implementing the seaport restoration. The number of RTs to be assigned and the number of equipment to be rented, purchased, and repaired are naturally formed as positive integers. However, to gain the benefit from binary GA, we decode a binary variable to calculate the positive integer using Equations (3) - (6). Please note that Equations (4) and (5) allow more than one equipment to be rented and purchased each at the ship, dock, and yard. However, the number of equipment to be repaired is bounded by the number of existing equipment.

$$n_1 = \sum_{e \in E_1} 2^{(e-1)} \beta_{1e} \qquad (3)$$

$$n_{2b} = \sum_{e \in E_{2b}^{ship}} 2^{(e-1)} \beta_{2be} + \sum_{e \in E_{2b}^{dock}} 2^{(e-1)} \beta_{2be} + \sum_{e \in E_{2b}^{yard}} 2^{(e-1)} \beta_{2be} \qquad (4)$$

$$n_{3b} = \sum_{e \in E_{3b}^{ship}} 2^{(e-1)} \beta_{3be} + \sum_{e \in E_{3b}^{dock}} 2^{(e-1)} \beta_{3be} + \sum_{e \in E_{3b}^{yard}} 2^{(e-1)} \beta_{3be} \qquad (5)$$

$$n_{4b} = \sum_{e \in E_{4b}^{ship}} \beta_{4e} + \sum_{e \in E_{4b}^{dock}} \beta_{4e} + \sum_{e \in E_{4b}^{yard}} \beta_{4e} \qquad (6)$$

The seaport operations consist of several stages, which have a physical interdependence among themselves. In order to define the interdependencies, a node-link representation of seaport operations is adopted in which the activities are assumed to be embedded in nodes (**Figure 3**). We represent the set of available time periods by $\boldsymbol{T}$, the set of nodes by $\boldsymbol{A}$, the operational node by $\boldsymbol{N}$, the disrupted node is denoted as $\boldsymbol{N'}$, and set of links by $\boldsymbol{L}$. We will denote the set of nodes with interdependency by $\boldsymbol{M}$. For instance, the interdependency between nodes *i* and *j* is denoted by $M\,(i,j)$, meaning node *i* is dependent on node *j*; specifically, node *i* can be regarded as the parent of child node *j*.

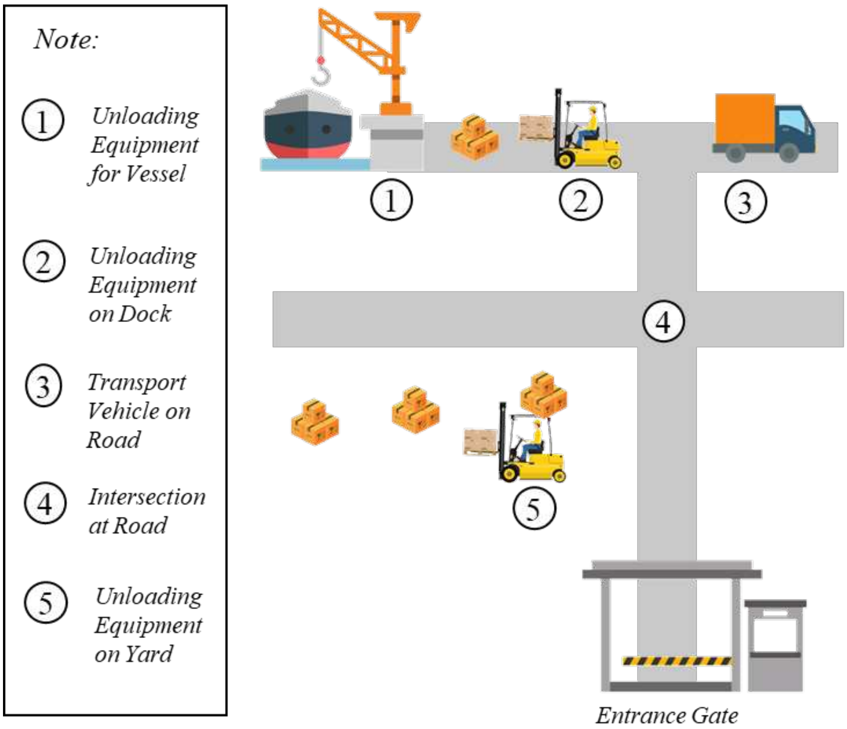

Figure 3: Network illustration

In the case of a berth not disrupted, it is possible for a vessel to moor at the dock. After berthing, the goods are placed in the loading dock by using equipment of the vessel. The unloading equipment at the dock is then employed for placing it over a transport vehicle. The transport vehicle carries the goods to stacking yard, which is located in the inner/outer seaport area. The road infrastructure plays an important role in the goods distribution smoothly. This intertwined operation thus defines the term, *maximum throughput*, which is equal to the minimum of capacities across all stages of an operational process (see Equation (7)). Furthermore, as can be inferred from Equation (8), the flow capacity at a link is strongly influenced by the productivity of an adjacent node. The network flow conservation is ensured by defining the constraints as in (9) and (10).

$$q_t = \min\{\dots, u_{ijt}, \dots\}, \forall t \in \boldsymbol{T}, (i,j) \in \boldsymbol{L}, i \in \boldsymbol{M}, j \in \boldsymbol{M} \tag{7}$$

$$u_{ijt} = \min\{prod_{it}, prod_{jt}\}, \forall t \in \boldsymbol{T}, (i,j) \in \boldsymbol{L}, i \in \boldsymbol{M}, j \in \boldsymbol{M} \tag{8}$$

$$\sum_{(i,j)\in \boldsymbol{L}} x_{ijt} - \sum_{(i,j)\in \boldsymbol{L}} x_{jit} = 0, \forall t \in \boldsymbol{T}, i \in \boldsymbol{M}, j \in \boldsymbol{M} \tag{9}$$

$$x_{ijt} - u_{ijt} \leq 0, \forall t \in \boldsymbol{T}, (i,j) \in \boldsymbol{L}, i \in \boldsymbol{M}, j \in \boldsymbol{M} \tag{10}$$

Equation (11) shows the productivity of node i in time period t in terms of the productivity of equipment, the number of equipment involved, and the availability of nodes. The availability of node-i at t-time (i.e., ω_{it}) will be updated based on the restoration processes, which is provided by the new DP or the strategy-based Hungarian Algorithm (see Section 3.3). Equation (12) represents the physical interdependence between nodes, in which it ensures that the child node-*j* is operated only if the parent node-*i* is operationalised. Equations (13)−(14) ensure the plausibility of flow on link *ij* at time *t* for the network with restoration decisions.

$$prod_{it} = \sum_{b=1}^{|\boldsymbol{B}|} (n_{2b} p_{2b} \theta_{2bit} + n_{3b} p_{3b} \theta_{3bit} + n_{4b} p_{4b} \theta_{4bit}) \omega_{it}, \forall t \in \boldsymbol{T}, i \in \boldsymbol{M} \tag{11}$$

$$\omega_{jt} - \omega_{it} \leq 0, \forall t \in \boldsymbol{T}, i \in \boldsymbol{M}, j \in \boldsymbol{M} \tag{12}$$

$$x_{ijt} - u_{ijt}\omega_{it} \leq 0, \forall t \in \boldsymbol{T}, i \in \boldsymbol{M}, j \in \boldsymbol{M} \tag{13}$$

$$x_{ijt} - u_{ijt}\omega_{jt} \leq 0, \forall t \in \boldsymbol{T}, i \in \boldsymbol{M}, j \in \boldsymbol{M} \tag{14}$$

$$\theta_{2bit}, \theta_{3bit}, \theta_{4bit}, \omega_{it}, \omega_{jt} \in \{0,1\} \tag{15}$$

As the restoration progresses resulting in nodes being made available, it changes the seaport throughput. The restoration not only provides the node for placing the unloading equipment but also opens up the link for horizontally transporting the goods. Let us assume that the productivity of equipment b (*i.e.,* truck or forklift) is dependent on the road network for their movement, then Equation (16) indicates the relationship between infrastructure restoration and its productivity.

$$p_{\chi b} = \Gamma_{\chi b} \left(\frac{\varrho}{\Upsilon_{\chi b}(i,j,\boldsymbol{N}')} \right) \forall \chi = \{2,3,4\}, (i,j) \in \boldsymbol{L} \tag{16}$$

Here $\Upsilon_{\chi b}(i, j, \boldsymbol{N'})$ is a function that returns the minimum travel time from i to j over all the paths that do not pass-through disrupted nodes. The function returns infinity if no such paths exist. Shortest path travel time in this paper is derived efficiently by invoking Dijkstra's algorithm (Dijkstra, 1959).

Road restoration constraints are defined by Equations (17) – (21). Constraint (17) guarantees that if node-*i* is to be restored, it is assigned to team-*r* at time period-*t*. Equation (18) ensures that team-*r* can only work on a single disrupted node during the restoration time-*t*. Equation (19) makes sure that if node-*i* is operational, it is attended by team-*r* until completion. Besides, Equation (20) ensures that the disrupted node-*i* is operationalised if only the required time for restoring is passed. Equation (21) imposes a similar condition from the perspective of the repair team-*r*, namely, team-*r* cannot complete the restoration process prior to the required time.

$$z_i = \sum_{r\in \boldsymbol{R}} \sum_{t\in \boldsymbol{T}} \varsigma_{rit}\, , \forall i \in \boldsymbol{N'}, t \in \boldsymbol{T}, z_i \in \{0,1\} \tag{17}$$

$$\sum_{i\in \boldsymbol{N'}} \sum_{l=t}^{\min\{\tau, l+s_i-1\}} \varsigma_{ril} \leq 1\, , \forall r \in \boldsymbol{R}, t \in \boldsymbol{T}, \boldsymbol{T} = [1, \ldots, \tau] \tag{18}$$

$$\omega_{it} \leq \sum_{r\in \boldsymbol{R}} \sum_{l-1}^{t} \varsigma_{ril}\ \forall t \in \boldsymbol{T}, i \in \boldsymbol{N'} \tag{19}$$

$$\sum_{t=1}^{s_i-1} \omega_{it} = 0, \forall t \in \boldsymbol{T}, i \in \boldsymbol{N'} \tag{20}$$

$$\sum_{r\in \boldsymbol{R}} \sum_{t=1}^{s_i-1} \varsigma_{rit} = 0, \forall t \in \boldsymbol{T}, i \in \boldsymbol{N'} \tag{21}$$

In the ensuing, we describe the essential algorithms for assigning repair teams to nodes in restoring the internal road network.

3.3 Algorithms for solving the internal road restoration problem

3.3.1 New Dynamic Programming to Assign Multicrew Restoration Teams

The internal road restoration problem involves assigning teams to disrupted nodes and also identifying the route to reach the location as some parts of the network have degraded. For tackling the road restoration problem, Maya Duque et al. (2016) proposed a method using DP that provides an exact solution. It is noted that their method works with a single crew of RT, which needs extending to deal with multiple crews for a real-life application. Therefore, we formulate a new DP-based method that can deal with multiple crews of RTs. The restoration process can be viewed as a sequence of RTs visiting disrupted nodes, which can be represented as a permutation set. Each subset of a permutation indicates the candidate solution, which must be evaluated for optimality. However, not all subsets can provide a feasible solution because the generation of permutation set may not consider the connectivity of a disrupted node with the location of RT. This phenomenon could mean that an RT might be blocked from completing a sequence as instructed by the permutation subset. For instance, assume that there are three disrupted nodes (say, A, B and C), and A is the only node that can be visited by RT in the initial condition, where C cannot be reached before restoring B. Therefore, any sequence without A as the first in the order will not be feasible. In addition, the subset that provides an order for visiting C before B will also give an infeasible solution. To remove such infeasible sets, we have embedded a new local *rule to reduce* the network. In the initial step, the rule will enable removing the subset(s) that include a non-connected node first in the order, for instance B-C-A or C-B-A in the above case. The rule also adjusts the iteration in case of RT movement being blocked, which can happen in the middle of the sequence, for instance A-C-B.

Table 3: Notation for the New Dynamic Programming

Set and Indices	
$\boldsymbol{N^{R}_{r,g}}$	Set of nodes that locates RT-*r* along the stage-*g*
$\boldsymbol{N^{P}}$	Set of all permutations of disrupted nodes

$\boldsymbol{N^a}$	the subsets of $\boldsymbol{N^P}$ which do not connect RT depot with disrupted nodes in the initial state, where Ψ denotes the number of subset members' (i.e., $\Psi = \lvert \boldsymbol{N^a} \rvert$)
$\boldsymbol{N}_{gate}$	Set of gate's nodes
$\boldsymbol{N}_{berth}$	Set of berth's nodes
$\boldsymbol{Z_R}$	Optimal solutions sets of RT assignment for permutation sets $\boldsymbol{N^P}$
Parameters	
g	Current state of system in dynamic programming
g'	Updated state of system in dynamic programming
$\varpi_{r,g}$	Road restoration cost incurred at state-*g* by assigning RT-*r*
t_r^{ava}	The time of RT-*r* to be available for restoring the disrupted node
Υ_r	The travel time to the disrupted node from the current node of RT-*r*
$f(g)$	The minimum cost of system from the initial state to the current state-*g*
$N_{r,1}^R$	Node of RT-*r* depot location
α_g	Disrupted node to be repaired at state-*g*
$d_{r,g'}$	The arbitrary cost for state due to restoration efforts of RT-*r*
$open$	The time of gate and berth to be connected

For dealing with multiple crews of RTs, we have a developed a DP that seeks to optimally assign RTs within each permutation subset. To restore the disrupted network, we configure the problem into a multistage decision process, where we aim to minimise the sum of restoring cost over all stages of the decision process. The stages are derived from the permutation subset, where in each stage the DP decides the particular RT crew to be assigned to the disrupted nodes given by the subset. The repairing order of nodes is then explored by generating a permutation set of disrupted nodes. RT assignment decision will change the state of system ($g \rightarrow g'$), which includes the disrupted nodes that are not yet restored, the current location of RT, and the time of RT to finish the job at the current node. Each decision will be incurring a restoring cost $\varpi_{r,g}$. The cost involves the time of RT available to restore the disrupted node (t_r^{ava}), the travel time to the disrupted node from the current node of RT (Υ_r) and the duration for repairing the node (s_i) (Please see **Table 3** for the notation used).

$$\varpi_{r,g} = t_r^{ava} + \Upsilon_r \ (i, j, \boldsymbol{N}') + s_i \ , \forall i \in \boldsymbol{N}', j \in \boldsymbol{N}_{r,g}^{\boldsymbol{R}}, r \in \boldsymbol{R} \tag{19}$$

The next state depends on the current state and the decision taken. Let us denote $f(g)$ as the minimum cost of system from the initial state to the current state, the objective function of network restoration can be written as follows:

$$f(g') = \min_{r \in \boldsymbol{R}} \big(f(g) + \varpi_{r,g}\big) \tag{20}$$

By recursively solving Equation (20), the optimal solution for each permutation subset can be obtained. However, in general, the road restoration in seaport aims to minimise the connection time of entrance gate and berth (i.e., $open$). This connection can be regarded as an operational performance indicator of a seaport. Therefore, the selection of optimal solution from the set of all permutations is conducted by sorting the connection time in a descending order. The algorithm for the new DP is shown as in **Algorithm 1.** The proposed DP requires running time of $O(|\boldsymbol{R}|^{N'})$, which increases with the number of RTs and disrupted nodes.

Algorithm 1: The New Dynamic Programming

Input: The set of operational nodes ($\boldsymbol{N}$), the set of disrupted nodes ($\boldsymbol{N}'$), the number of RTs (n_1), the location of gate ($\boldsymbol{N}_{gate}$), berth ($\boldsymbol{N}_{berth}$), and RT at depot ($N_{r,1}^{R}$)

Output: RT schedule for restoring the disrupted network

1: Generate vector $\boldsymbol{N^P}$ containing all permutations of disrupted nodes.

2: Reduce vector $\boldsymbol{N^P}$ to $\boldsymbol{N^a}$ by removing the subsets which do not connect RT depot with disrupted nodes in the initial stage. $\boldsymbol{N^a} = [\boldsymbol{N_1^a}, \dots, \boldsymbol{N_k^a}, \dots \boldsymbol{N_\Psi^a}]$ denotes the reduced permutation set and $\Psi = |\boldsymbol{N^a}|$.

3: Set k=1

4: **while** $k \leq \Psi$ **do**

5: Initialise number of RTs, which is available at depot, and select the permutation subset -k denoted by $\boldsymbol{N_k^a} = \left[\alpha_1 , \dots, \alpha_g , \dots, \alpha_{|\boldsymbol{N_k^a}|}\right], \forall \alpha_{\boldsymbol{g}} \in \boldsymbol{N'}$

6: Set $g = 1$

7: **while** $g \leq |\boldsymbol{N_k^a}|$ **do**

8: **for** r = 1 **to** n_1 **do**

9: Estimate the cost of restoring the node-α_g by RT-r

$$\varpi_{r,g} = t_r^{ava} + \Upsilon_r \left(N_{r,g}^R, \alpha_g , \boldsymbol{N'}\right) + s_{\alpha_g}, \forall \alpha_g \in \boldsymbol{N_k^a}, r \in \boldsymbol{R}$$

10: Calculate the arbitrary value of function by considering the cost incurred by RT-r $d_{r,g'} = f(g) + \varpi_{r,g}$

11: **end for**

12: **if** $\min\left[d_{1,g'}, \dots, d_{r,g'}, \dots, d_{|\boldsymbol{R}|,g'}\right] = \inf$ **then** $f(g) = \inf$ go to line 24:

13: **else**

14: Calculate objective value $f(g) = \min\left[d_{1,g'}, \dots, d_{r,g'}, \dots, d_{|R|,g'}\right]$

15: Assign the RT with smallest objective value to node α_g until the end of restoration (i.e., $\varpi_{r,g}$)

16: Add node-α_g to the operational node $\boldsymbol{N}$ at $\varpi_{r,g}$ and set $t = \varpi_{r,g}, i = \alpha_g$ and $\omega_{it} = 1$

Update the location of RT $N^{R}_{r,g+1} = \alpha_g$, $f(g) = f(g')$, and $g = g + 1$
end if
Estimate the shortest travel time $\Upsilon(\boldsymbol{N}_{gate}, \boldsymbol{N}_{berth}, \boldsymbol{N}')$ from the gate to berth avoiding the disrupted nodes $\boldsymbol{N}'$
if $\Upsilon(\boldsymbol{N}_{gate}, \boldsymbol{N}_{berth}, \boldsymbol{N}') \neq \text{inf}$ **then** $open(k, \boldsymbol{N}_{gate}, \boldsymbol{N}_{berth}) = \varpi_{r,g}$
else $open(k, \boldsymbol{N}_{gate}, \boldsymbol{N}_{berth}) = \tau, \boldsymbol{T} = [1, \dots, \tau]$
end if
end while
Save optimal solution of RT assignment for permutation set $\boldsymbol{N}^{\boldsymbol{a}}_{k}$ as $\boldsymbol{Z}_{\boldsymbol{R}k}$
k=k+1
end while
Create a list by sorting $open(k, \boldsymbol{N}_{gate}, \boldsymbol{N}_{berth})$ of the solution from the smallest value to largest value
Select the top solution of the list as the optimal solution, where $\boldsymbol{Z}^{*}_{\boldsymbol{R}k}$ denotes as optimal RT assignment

Preposition 1. *Let $\boldsymbol{N}^{\boldsymbol{P}}_{k}$ be an order of disrupted nodes to be repaired and let $\boldsymbol{Z}_{\boldsymbol{R}k}$ denote the optimal solution of RT assignment, which is provided by a new DP, then $\boldsymbol{Z}_{\boldsymbol{R}k}$ is an exact solution for $\boldsymbol{N}^{\boldsymbol{P}}_{k}$.*

Proof. Let us assume that we have an order of disrupted nodes $\boldsymbol{N}^{\boldsymbol{P}}_{k}$, the order reflecting the sequence for restoring the nodes. For instance, $\boldsymbol{N}^{\boldsymbol{P}}_{k} = (1,2,3)$ means that RT needs to firstly repair node 1, then node 2, and finally node 3. The repairing decision generates a cost $\varpi_{r,g}$ and will change the state of the system ($g \rightarrow g'$). In the case of multicrew RT, the solution technique i.e., the new DP should decide the appropriate RT to be assigned to each sequence by minimizing the system cost of restoration from the initial state to the final state. This objective is formulated as Equation (20), wherein $f(g)$ is the minimum cost of system from the initial state to the current state. Equation (20) follows the Bellman's principle of optimality (Bellman, 1954), by solving the equation recursively, the optimal solution of the order (i.e., $\boldsymbol{N_k^P}$) can be guaranteed. Therefore, $\boldsymbol{Z_{Rk}}$ denotes the optimal solution of RT assignment, which is provided by a new DP, then $\boldsymbol{Z_{Rk}}$ is an exact solution for $\boldsymbol{N_k^P}$

Preposition 2. *If* $\boldsymbol{N^P}$ *denotes all permutation sets of restoration order i.e.,* $\boldsymbol{N^p} = \left[\boldsymbol{N_1^p}, \dots, \boldsymbol{N_k^p}, \dots \boldsymbol{N_{|N^p|}^p}\right]$ *and* $\boldsymbol{Z_R} = \left[\boldsymbol{Z_{R1}}, \dots, \boldsymbol{Z_{Rk}}, \dots, \boldsymbol{Z_{R|Z_R|}}\right]$ *denotes the set of optimal solutions for each* $\boldsymbol{N_k^p}$ *, then* $\boldsymbol{Z_{Rk}^*}$ *provides the minimum value of the Bellman equation which is an optimal solution of all the permutation sets* $\boldsymbol{N^P}$ *.*

Proof. Preposition 1 proves that the $\boldsymbol{Z_{Rk}}$ guarantees the optimal solution of repair order $\boldsymbol{N_k^p}$. By evaluating all possible repair orders in $\boldsymbol{N^p}$, we can create a set of optimal solutions derived for each $\boldsymbol{N_k^p}$, namely $\boldsymbol{Z_R}$. As the restoration model aims to minimise the system cost (i.e., Equation (20)), the set $\boldsymbol{Z_{Rk}}(i.e., \boldsymbol{Z_{Rk}} \in \boldsymbol{Z_R})$ that gives a minimum value of the system cost then guarantees the optimal solution for all permutation sets. A similar situation happens in the case of the model aim to minimise the connection time of the entrance gate and berth. Please note that for reducing the permutation set, we refine the $\boldsymbol{N^p}$ by removing the set which does not connect RT depot with disrupted nodes in the initial state, namely $\boldsymbol{N^a}$.

3.3.2 Exploitation-exploration based Dynamic Hungarian Algorithm

Although the new DP provides an exact solution, it cannot deal with large-sized problems due to excessive computation time involved. Thus, there is a need to develop a new algorithm to efficiently solve large-sized problems as the available alternatives such as greedy algorithm may get trapped into local optima. In this research, we propose to extend the Hungarian

Algorithm for tackling large-sized problems which promises an efficient solution to the problem. The Hungarian method assigns a repair team to a disrupted node by evaluating several feasible combinations (Kuhn, 1955; Munkres, 1957). The original version of the Hungarian Algorithm (HA) is a single-pass method for assigning a repair team to a disrupted node, where the restoration cost is configured by the assignment matrix with ($|\boldsymbol{H_t}| \times |\boldsymbol{N_t^H}|$) dimension (see **Table 4** for the notation description). $\boldsymbol{N_t^H} \in \boldsymbol{N'}$ denotes the set of candidate nodes to be restored, and $\boldsymbol{H_t} \in \boldsymbol{R}$ represents the set of available teams at time- t. The candidate node is located in the top row of the matrix and the available teams in the left most column of the matrix (see **Table 5**). Please note that the perspective for calculating the restoration cost incurred by RT is different to that of the DP. HA counts the minimum cost for restoring the node-i at time-t whereas DP tracks the cost from an initial state to the current state of system. See **Appendix A** for the steps involved in running the original HA. Zukhruf and Frazila (2020) utilised the original HA for tackling the road restoration by dynamically updating the availability of repair teams and regularly revising the condition of nodes through time steps up to the horizon period (see **Appendix B** for the algorithm involved). We refer to their algorithm as the Dynamic HA or simply DHA. It is noted that the quality of the solution to DHA, however, largely depends on the initial conditions and thus can lead to poor solutions. In this research, we extend the DHA by adopting an exploitation-exploration based strategy as described further.

Table 4: Notation for Dynamic Hungarian Algorithm

Set and Indices	
$\boldsymbol{H}_t$	Set of available teams at time- t, where $|\boldsymbol{H}_t|$ describe the number of available team at time-t
$\boldsymbol{N}_t^{\boldsymbol{H}}$	Set of candidate nodes to be restored, where $|\boldsymbol{N}_t^{\boldsymbol{H}}|$ describe the number of candidate nodes to be restored at time-t

$\boldsymbol{N^{l}}$	Set of nodes resulted from the reduced algorithm
$\boldsymbol{N^{l1}}$	Subset of $\boldsymbol{N^{l}}$ that only containing disrupted nodes (i.e., $\boldsymbol{N^{l1}} = \{\boldsymbol{N^{l1}} \mid \boldsymbol{N^{l}} \in \boldsymbol{N'}\}$)
$\boldsymbol{N^{l2}}$	Subset of $\boldsymbol{N^{l}}$ that only containing operational nodes (i.e., $\boldsymbol{N^{l2}} = \{\boldsymbol{N^{l2}} \mid \boldsymbol{N^{l}} \in \boldsymbol{N}\}$)
$\boldsymbol{N^{c}}$	Set of nodes, which is sorted based on the fraction value (i.e.,δ_i) in descending order
$\boldsymbol{fit}$	Set of fitness values from Dynamic Hungarian Algorithm (i.e., $open$), indexed by fit_i
$\boldsymbol{cont}$	Set of operational nodes from Dynamic Hungarian Algorithm indexed by $cont_i$
$\boldsymbol{dis}$	Set of disrupted nodes from Dynamic Hungarian Algorithm indexed by dis_i
Parameters	
$\varpi_{r,i}$	Cost incurred by RT-*r* for restoring the disrupted node-*i*
$SP_{(r,k)}$	Shortest path from node-*r* to node-*k*
δ_i	Fraction value of node-*i* representing the value of betweenness centrality index
fit^*	Fitness value of optimal solution
$cont^*$	Operational nodes of optimal solution
dis^*	Disrupted nodes of optimal solution

Table 5: Assignment matrix

	1	…	*i*	…	$\lvert \boldsymbol{N_t^H} \rvert$
1	$\varpi_{1,1}$	…	$\varpi_{1,i}$	…	$\varpi_{1,\lvert \boldsymbol{N_t^H} \rvert}$

⋮	⋮		⋮		⋮
r	$\varpi_{r,1}$	$\cdots$	$\varpi_{r,i}$	$\cdots$	$\varpi_{r,\lvert N_t^H \rvert}$
⋮	⋮		⋮		⋮
$\lvert \boldsymbol{H_t} \rvert$	$\varpi_{\lvert H_t \rvert,1}$	$\cdots$	$\varpi_{\lvert H_t \rvert,i}$	$\cdots$	$\varpi_{\lvert H_t \rvert,\lvert N_t^H \rvert}$

The extended DHA includes an *exploitation-exploration strategy* within the DHA and is referred as *ee-DHA* henceforth. The strategy considers optimal solutions, firstly, by reducing the network (exploitation by narrowing the search space) and then gradually adding the disrupted node(s) to be optimised by the DHA (exploration by widening the search space). The strategy initially reshapes the disrupted network to a *reduced network*. The reduced network contains disrupted nodes that are included in the essential shortest path between gate and berth, which are called SP-nodes (see **Figure 4** (b)). To ensure that an RT can visit the SP-nodes, we also add the nodes that are included on the shortest paths between other nodes e.g. RT depot and SP-nodes (i.e., Depot-Nodes see **Figure 4** (c-d)). The algorithm for reducing the network is given in **Algorithm 2**.

Algorithm 2: **Reduced Network Algorithm**

Input: The set of operational nodes ($\boldsymbol{N}$), the set of disrupted nodes ($\boldsymbol{N'}$), the number of RTs (n_1), the location of gate ($\boldsymbol{N}_{gate}$), berth ($\boldsymbol{N}_{berth}$), and RT at depot ($N_{r,1}^{R}$)

Output: Reduced disrupted network consists of $\boldsymbol{N^l}$ nodes

1 : **for** i = 1 **to** $|\boldsymbol{N'}|$ **do**

2 : Estimate the time for repairing node-i (i.e., s_i).

Algorithm 2: **Reduced Network Algorithm**

---

end for
for *i* = 1 **to** $(|\boldsymbol{N'}| + |\boldsymbol{N}|)$ **do**
    for *j* = 1 **to** $(|\boldsymbol{N'}| + |\boldsymbol{N}|)$ **do**
        Estimate the travel time **if** $\Upsilon(i,j)$ for each pair of nodes (i,j)
        if $i \in \boldsymbol{N'}$ or $j \in \boldsymbol{N'}$ **then** add s_i (or s_j) to $\Upsilon(i,j)$
        end if
    end for
end for
for *r* = 1 **to** n_1 **do**
    for *k* = 1 **to** $(|\boldsymbol{N}_{gate}|+|\boldsymbol{N}_{berth}| + |\boldsymbol{N}_{yard}|)$ **do**
        Generate the shortest path $SP_{(r,k)}$ from to $N^R_{r,1}$, $\boldsymbol{N}_{gate}$, $\boldsymbol{N}_{berth}$ and $\boldsymbol{N}_{yard}$ based on the $\Upsilon(i,j)$
    end for
end for
Create a list of nodes $\boldsymbol{N^l}$ which is included in the $SP_{(r,k)}$
Refine $\boldsymbol{N^l}$ by containing only the member of $\boldsymbol{N'}$ (i.e., $\boldsymbol{N^l} \in \boldsymbol{N'}$) to create $\boldsymbol{N^{l1}}$
for *r* = 1 **to** n_1 **do**
    for *k* = 1 **to** $|\boldsymbol{N^{l1}}|$ **do**
        Generate the shortest path $SP_{(r,k)}$ from $\boldsymbol{N^R_{r,1}}$ to $\boldsymbol{N^{l1}}$ based on the $\Upsilon(i,j)$
    end for
end for
Update the $\boldsymbol{N^l}$ by adding the new nodes generated from $SP_{(r,k)}$

Algorithm 2: **Reduced Network Algorithm**

Refine $\boldsymbol{N^l}$ by containing only the member of $\boldsymbol{N'}$ to create $\boldsymbol{N^{l1}}$ and the rest (i.e., operational node) is stored to $\boldsymbol{N^{l2}}$

This reduction scheme forces the DHA to assign an RT only to important nodes connecting the route between gate and the berth. However, the reduction of network may drive the algorithm to trap into a local optimal solution. Hence, we gradually add the rest of disrupted nodes to the reduced network for avoiding the local optimal. From the perspective of optimisation, the reduction scheme creates a narrower search space, which is commonly called as the *exploitation strategy*. This strategy works well for improving the solution but can adversely lead to a local optimal solution. Furthermore, the addition of disrupted nodes drives a wider search space, which can be viewed as the *exploration strategy*. The broader search space poses difficulties to the algorithm in providing optimal solution, thus, a gradual addition is adopted in the ee-DHA, which balances the *exploitation-exploration* strategy.

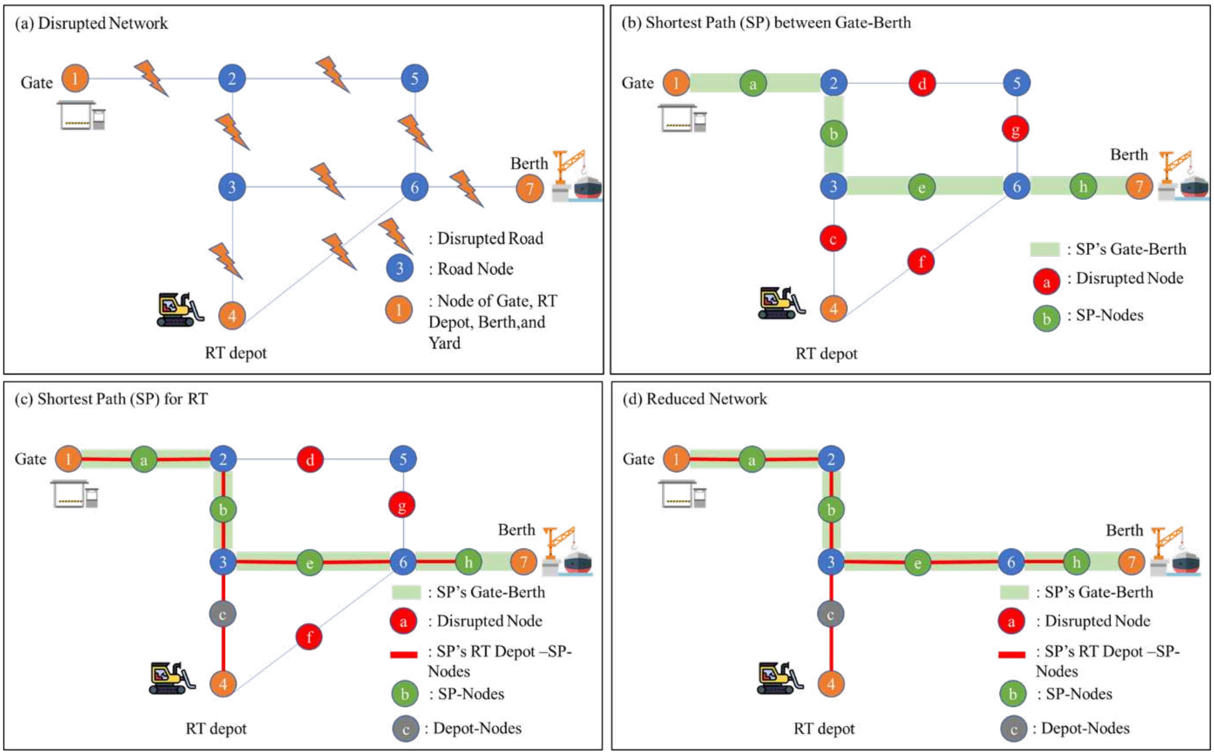

Figure 4: Illustration of reduced network

To facilitate gradual addition of the remaining disrupted nodes which are not part of SP-Nodes or Depot-Nodes, nodes are then sorted based on a connectivity index. More specifically, this paper invokes the *betweenness centrality index* (Freeman, 1977). The index value of a node tends to be higher if it is on shortest paths of many node-pairs in a network. In terms of seaport operations, we are mainly concerned with pairs of essential nodes, viz., entrance/exit gate - berth access, stacking yard - berth access and entrance/exit gate - stacking yard. The betweenness centrality of nodes is calculated by assuming the road network was intact i.e. not struck by a disaster. Therefore, the value can partly guide the RT to restore the vital nodes, which can connect the pairs of essential nodes within the shortest time. For calculating the betweenness centrality index value of each node, we firstly employ Dijkstra's algorithm (Dijkstra, 1959) for generating the shortest paths between essential node pairs. Then the fraction value of a disrupted node is calculated by dividing the number of shortest paths passing through the node with the number of node pairs evaluated. For instance, if there were four pairs of nodes evaluated, and the shortest paths between two of the pairs pass through the disrupted node-i, the fraction value (i.e.,δ_i) would be (2÷4) equal to, δ_i=1/2. The fraction value calculation is repeated for every disrupted node. In the case of a disrupted node that did not appear on any of the shortest paths, the fraction value of the node is set to a small number (e.g., δ_i= 0.001) to avoid the division by zero in the computation. The algorithm for implementing the ee-DHA can be found in **Algorithm 3**.

Algorithm 3: **Exploitation-exploration based Dynamic Hungarian Algorithm (ee-DHA)**

Input: The set of operational nodes ($\boldsymbol{N}$), the set of disrupted nodes ($\boldsymbol{N'}$), the number of RTs (n_1), the location of gate ($\boldsymbol{N}_{gate}$), berth ($\boldsymbol{N}_{berth}$), RT at depot ($\boldsymbol{N}^{\boldsymbol{R}}_{r,1}$), and set of nodes as resulted from reduced algorithm ($\boldsymbol{N}^{\boldsymbol{l}}$)

Output: RTs schedule for restoring the disrupted network

1: Calculate the betweenness centrality index for all disrupted nodes and sort from the largest value to the smallest and obtain the sorted nodes $\boldsymbol{N^c}$

2: Invoke **Algorithm 2** for reducing the disrupted network

3: Set $\boldsymbol{N^{l1}}$ and $\boldsymbol{N^{l2}}$ as the initial subset of that contains only the disrupted nodes and the operational nodes, respectively

4: **for** *i*=1 **to** ($|\boldsymbol{N^c}| - |\boldsymbol{N^{l1}}| + 1$)

5: Utilise $\boldsymbol{N^{l1}}$ and $\boldsymbol{N^{l2}}$ as input for **Dynamic Hungarian Algorithm** (see **Appendix B**) by replacing $\boldsymbol{N'}$ (i.e., $\boldsymbol{N'} = \boldsymbol{N^{l1}}$) and $\boldsymbol{N}$ (i.e., $\boldsymbol{N} = \boldsymbol{N^{l2}}$), respectively

6: Record $open(i, \boldsymbol{N_{gate}}, \boldsymbol{N_{berth}})$, $\boldsymbol{N}$, and $\boldsymbol{N'}$ from **DHA** (see **Appendix B**) as fit_i , cont_i , and dis_i

7: **for** *j*=1 **to** $|\boldsymbol{N^c}|$

8: **if** $\alpha_j \notin \boldsymbol{N^{l1}}$ $\forall \alpha_j \in \boldsymbol{N^c}$ **then** add α_j to $\boldsymbol{N^{l1}}$

9: **end if**

10: **end for**

11: **end for**

12: Create a list of solutions as $\boldsymbol{fit} = \left[fit_1, \dots, fit_i, \dots, fit_{|\boldsymbol{N^c}|-|\boldsymbol{N^{l1}}|}\right]$ and then sort fit from the smallest value to largest value.

13: Create $\boldsymbol{cont}$ and $\boldsymbol{dis}$ as a set of $cont_i$ and dis_i by following order of sorted $\boldsymbol{fit}$

14: Select the top of list of $\boldsymbol{fit}$ as the fitness value of optimal solution (i.e., fit^*) and the related $cont_i$, dis_i as $cont^*$, dis^*, respectively

15: From the optimal solution, check whether all disrupted nodes are restored

16: **if** $|dis^*| > 0$ **then**

17: invoke **DHA** by replacing $\boldsymbol{N'}$ with dis^* *a*nd $\boldsymbol{N}$ with $cont^*$

18: update fit^*, dis^*, and $cont^*$

19:**end if**

3.4 Algorithms for solving the outer optimisation problem

3.4.1 Simple Genetic Algorithm (SGA)

As a popular solution technique for solving engineering optimisation problems, GA has grown extensively. Simple GA (SGA) involves three basic operations, viz., recombination, mutation, and selection. SGA is formed by a group of individuals that has specific chromosome information (i.e., set of alleles), which constructs the fitness value. The offspring is produced from the recombination (i.e., also known as "crossover") by mating two individuals in the population. Each mating (i.e., parents) produces two off-springs, in which a single point crossover or a uniform crossover method is applied. The mutation operator is employed for enriching the chromosome information of an individual because it is possible to generate the information that is not inherited from its parents. In addition, a selection process is included ensuring the number of individuals is kept constant in each generation. The roulette wheel-based selection procedure is incorporated into the SGA.

3.4.2 GA with insertion mutation (GAINMUT)

Mutation and recombination are natural processes in genetics, which are identified as the source of variability. Both processes have a different mechanism (Barton, 2010), which could change (or not change) the phenotype of living things. Following a similar idea, GA also involves both processes for generating a candidate solution (i.e., individual) by varying the chromosome information to avoid a local-optimal solution. Taking advantage from the current genome evidence, Zukhruf et al. (2019) proposed GA with insertion mutation (GAINMUT). They modified the mutation processes by applying the "insertion" based approach. In SGA, the mutation is implemented by changing the bit value from a stated condition, for instance, if a bit equal to 1, it is then mutated to 0, and otherwise (see **Figure 5** (a)). In GAINMUT, the mutation process is applied by inserting the binary random value to the allele, which significantly alters the chromosome information. The last allele is then deleted to keep the chromosome length (see **Figure 5** (b)).

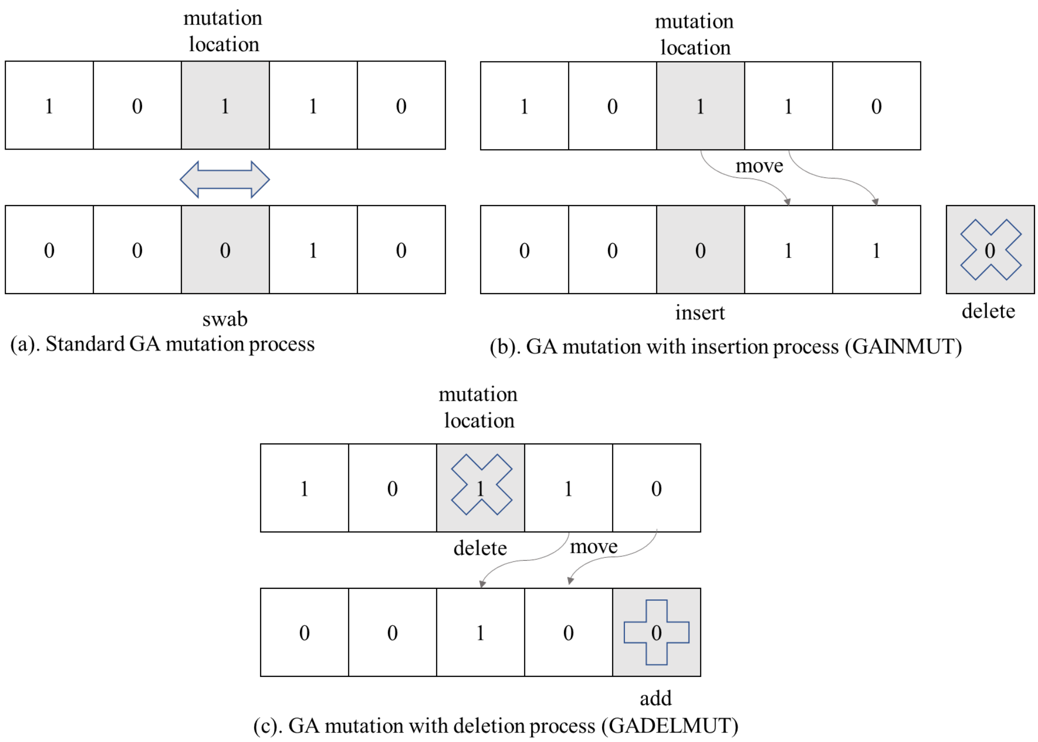

Figure 5: Illustration of the mutation process

3.4.3 GA with deletion mutation (GADELMUT)

We present GA with another variant of mutation processes called GA with deletion mutation (GADELMUT). In this mutation process, a randomly selected allele is deleted from the chromosome. A binary random value is then added to the last allele to keep the length of the chromosome consistent (see **Figure 5** (c)). In addition, we also invoke an adaptive approach for updating the mutation rate, which has been successfully implemented by Israeli and Gilad (2018). The adaptive approach aims to prevent the homogeneity of chromosomes in the population. It combines the time-dependent mutation rate with the rule that can reset the mutation rate if the chromosome variance of the population decreases below a predetermined threshold (see **Figure 6**). To prevent the loss of good individuals in selection processes, we also keep a number of best individuals in each generation using the elitist mechanism (Yamada *et al*., 2009).

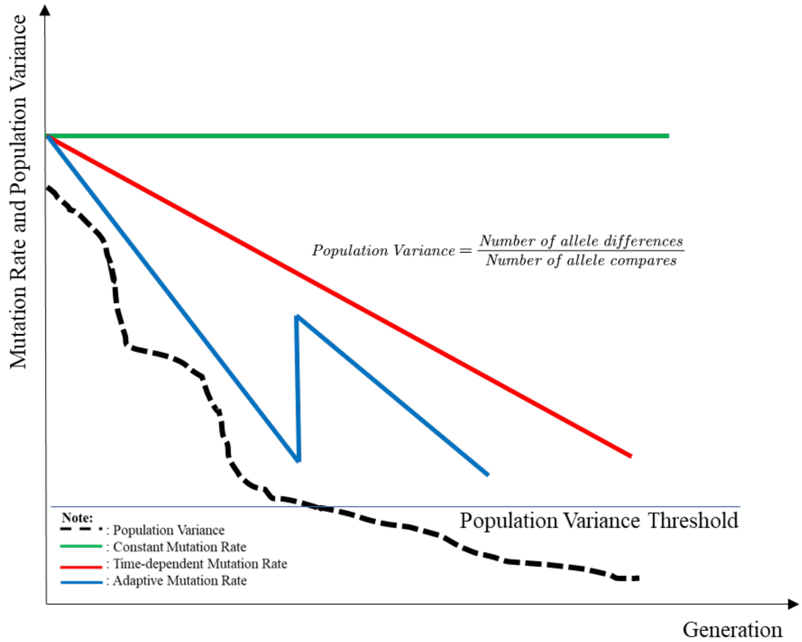

Figure 6: Possible ways for updating mutation rate

Table 6: Notation for Genetic Algorithm

Set and indices	
pop	Set of candidate solutions of genetic algorithm
elite	Set of elite individuals of genetic algorithm
Parameters	
V	Number of individuals
gen	Number of generations
chromo	Length of chromosomes

κ	Number of elites
ϕ	Crossover rate
ϵ	Mutation rate
binrand	A random binary value
$child_1$	Offspring of *V* resulted from cross over procedure
ρ	Population variance
$locmut_i$	Mutation location of $child_1$-*i*
randin	Random number between 0 and 1
$child_2$	Offspring resulted from mutation procedure
threshold	Threshold for the population variance

The GADELMUT procedure is then set out as **Algorithm** 4 (see **Table 6** for the notation) as follows:

Algorithm 4: **GADELMUT Algorithm**

Input: Number of individuals (i.e., *V*), number of generations (i.e., *gen*), length of chromosomes (i.e., *chromo*), number of elites (i.e, κ), crossover rate (i.e., ϕ), and mutation rate (i.e., ϵ).

Output: Optimal solution to seaport restoration problem

1: **for** *i*=1 **to** *V*

2: **for** *j*=1 **to** *chromo*

3: *pop* (*i,j*)= *binrand,* where *binrand* is a random binary value

4: **end for**

5: **end for**

6: **for** *l*=1 **to** *gen*

for i=1 **to** $V/2$
Randomly select two individuals
Apply uniform crossover procedure to produce offspring of V (i.e., $child_1$).
end for
Determine the population variance, $\rho = \frac{number\ of\ allele\ differences}{number\ of\ allele\ compares}$
if ρ < *threshold* **then** $\epsilon = 0.9$
else if then $\epsilon = 1/l$
end if
for i=1 **to** V
Randomly select the mutation location of $child_1$-i (i.e., $locmut_i$)
Generate a random number between 0 and 1 (i.e., *randin*)
if *randin* < ϵ **then**
Delete the allele of $child_1$ offspring to the mutation location to yield $child_2$
Insert the *binrand* in the last allele to keep the chromosome length
end if
end for
Evaluate the fitness value of each parent (i.e., V) and $child_1$& $child_2$
Select κ best individuals to make ***elite*** (i.e., elitist selection)
end for

4. Restoration of Pantoloan seaport operations

4.1 Description of the case

Pantoloan seaport, in Sulawesi, Indonesia was struck by a tsunami in 2018 causing a major disruption to its operations. The seaport located in Sulawesi (Latitude / Longitude: -0.7133916°

/ 119.8552°) is essential for reaching aid supplies in bulk to the seismic hinterland. Before the disruption, the seaport was equipped with two units of Quay Cranes (QC), two units of Rubber Tyred Gantries (RTG), four units of Reach Stackers (RS), three units of Forklifts (FL), five units of trucks, and 47 teams of gang. The seaport was operating as a single continuous berth with two yards for stacking, which were accessed by a road network of 20 nodes and 32 links (see **Figure 7**).

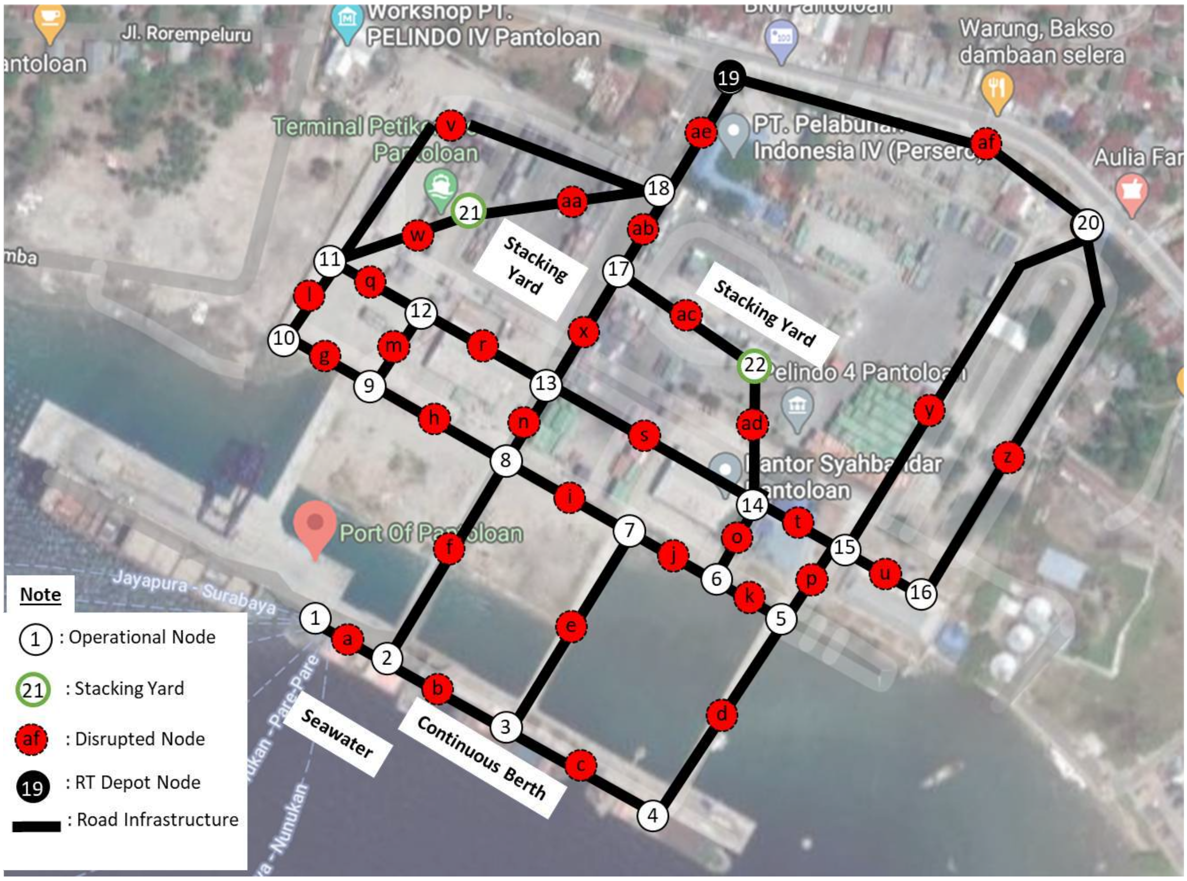

Figure 7: Node-link diagram of Pantoloan seaport

Based on the data derived from satellite imagery obtained after the tsunami, it was noted that all areas of Pantoloan seaport were covered by debris. The debris location is represented by disrupted nodes (i.e., node-*a* to node-*af*) that lie not only on the road network of the port but also on the roads to stacking yard. The yard is key to maintaining the throughput of a seaport, as the goods will need to be stacked in the outer area if the yard is disrupted. The non-availability of yard thus reduces the vehicle carrying capacities, and the overall throughput. Moreover, each disrupted node has a different volume of debris that needs to be cleared by the repair teams, thus, the restoration duration varies based on their capacity. For estimating the infrastructure restoration cost, the total time of RT is multiplied with its unit charge, accounting for mobilising and restoring involved.

Furthermore, all equipment was severely damaged by the tsunami, although we assumed that one QC and three forklifts can possibly be repaired at a specific cost (see **Table 7**). It is also assumed that one team of the gang is available for manual unloading each at the ship, dock, and yard, respectively. The gang also can be enlarged by outsourcing from third parties. A similar approach can possibly be followed in terms of crane and forklift renting, with the advantage of quicker availability of equipment and the variety of choices available. Let us assume that three teams of gangs, three units of trucks, and one RT are ready in the first hour after the disaster. Furthermore, a QC, three FLs at docks, and at the yard can possibly be repaired. One heavy crane, three light cranes, and six FLs are available for hire. Nine outsourcing gangs are also available to be added for handling the loading processes at the vessel, dock, and stacking yard. The horizon is set to 72 hours because it is regarded as the golden period in the response phase of disaster management (Sanci and Daskin, 2019). The unit value of relief supplies is estimated at 1825 USD/tonne (See **Appendix C** for details). In addition, one RT has been assigned to restore the infrastructure network, where the port authority can allow a maximum of 4 RTs to work simultaneously. Constraint on the number of RTs available is then reflected in a binary-decision structure introduced earlier.

Table 7: Possible restoration actions with unit costs[1]

No	Action	Available at the Hr	Productivity		Unit Cost	
			Value	Unit	Value	Unit
1	Repairing QC	60th	330	tonnes/hr	13,800[2]	USD/unit
2	Renting Heavy Crane	24th	495	tonnes/hr	14,400	USD/day/unit
3	Renting Light Crane	12th	100	tonnes/hr	7,000	USD/day/unit
4	Repairing Forklift	24th	18[3]	tonnes/hr	5,950	USD/unit
5	Renting Forklift	12th	27	tonnes/hr	900	USD/day/unit
6	Outsourcing Gangs	1st	5	tonnes/hr	300	USD/day/gang
7	Renting truck	12th	18	tonnes/trip	50	USD/hr/unit
8	Repair team	1st	75	cubic m/hr	100	USD/hr

1) Derived from the website of rental provider (e.g., bigge.com, bigrentz.com)

2) Derived Burden, 2012 from Werner and Cooke (2009)

3)The productivity of the repaired equipment is assumed to be two-thirds the productivity of similar rented equipment.

Since aid supplies are very time-sensitive, the time of equipment availability becomes an important factor. The lateness in equipment availability directly obstructs the unloading and distribution processes, which can potentially affect the number of casualties. Therefore, this paper considers the time of availability of equipment for deciding the optimal action to restore the seaport operations.

4.2 Interdependent nature of seaport restoration activities

Before attempting to solve the full optimisation problem, firstly, we illustrate the interdependent nature of decision making in seaport restoration process. Let's consider a simple example in which the port authority decides to employ two RTs, rent a heavy crane, and rent FLs to be utilised for restoration process at the dock and stacking yard. The FLs and the crane will be available in 12 hours, and 24 hours after the disruption, respectively. The ee-DHA predicts that the restoration of road access to the berth and to the stacking yard will have finished in 11 hours after the disruption. In this illustration, assume that the berth was not disrupted, and a ship request for berthing at 12 hours after the disruption.

Based on the example case described as above, **Figure 8** shows the interdependency of decisions in the restoration process. For example, the internal road is available at the 11^{th} hour, and the ship berths at the 16^{th} hour yet the loading/unloading of the ship cannot be started immediately since the crane is not available. The loading/unloading process can only be started from the 24^{th} hour when the rented crane arrives. A similar situation prevails even in the case of loading/unloading process at the dock and at the stacking yard. Although, the road access and the equipment have been restored by the 11^{th} hour, the loading/unloading at the dock cannot start before the loading/unloading of the ship is started. Subsequently, the loading/unloading at the stacking yard cannot be initiated before FLs at the dock begin to load/unload. Note that there is buffer time (slackness) for restoring the road access, renting the FLs for dock and yard. The slack time should be minimised by evaluating critical decisions in which the goal is to speed up the loading/unloading at the berth. For instance, if the port authority revises their decision by changing the type of crane rented which can arrive earlier, the slackness might be reduced, allowing the seaport to operate sooner. However, such decisions must be taken carefully as each decision is interdependent on several other factors and involves evaluating a huge number of possible combinations.

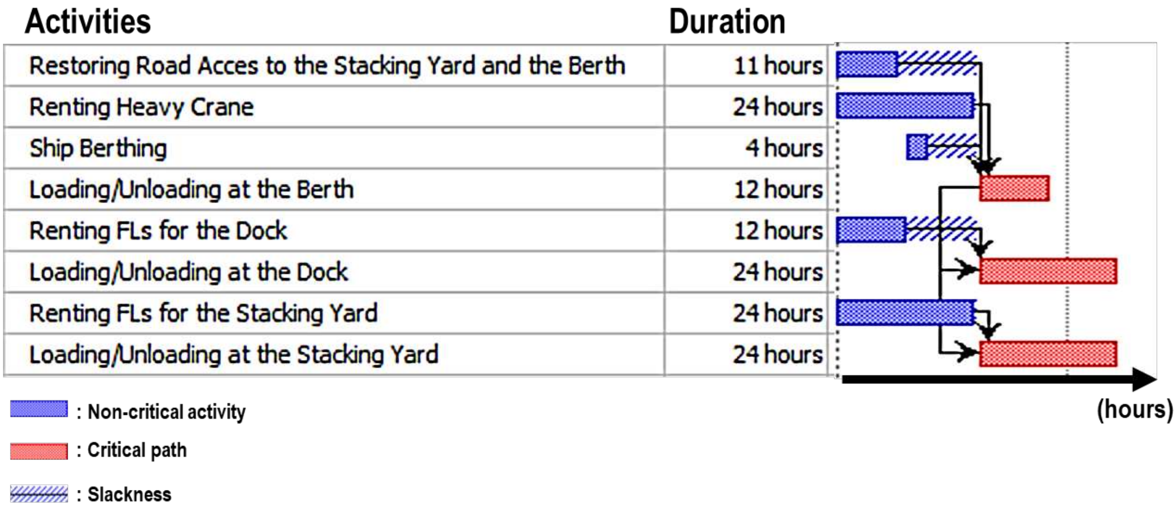

Figure 8: Illustration of interdependency of restoration activities

4.3 Optimal restoration plan for Pantoloan seaport

This section evaluates the optimal restoration time of seaport operations at Pantoloan.

We now maximise the fitness value i.e. ratio of total throughput to the cost of restoring seaport, by varying the number of RTs, unloading equipment and gangs involved in handling the goods. Based on the results from GADELMUT, the optimal outcome will involve a set of decisions comprising repairing the existing QC, renting a heavy crane, repairing up to three FLs at docks, and renting up to three FLs at docks and yards (Action Set A). To develop a better understanding of the optimal result, we compare it with sub-optimal restoration actions –Action Sets B-E, which can be derived from a complete enumeration of alternatives. These sets have components as shown in **Table 8**.

Table 8: Details of restoration action sets

Action Set	Components of action set
A	1 RT added, 1 heavy crane rented, 3 FLs each rented at the dock and yard, three gangs each at the dock, yard, respectively

B	1 RT added, 1 heavy crane rented, 3 FLs each rented at the dock and yard, respectively.
C	3 RTs added,1 heavy crane rented, 3 FLs each rented at the dock and yard, three gangs each at the dock, yard, respectively.
D	3 RTs added.
E	3 RTs, 1 QC repaired, 3 FLs each rented at the dock and yard, respectively

Figure 9 compares the performance of Action Set A with Action Sets B-E. Action Set A is the best performing of all with the highest fitness value. Action Set B has a lower cost without deploying the gangs as in Action Set A, though it results in lower productivity. Action Set C is almost identical to Action Set A, except that it deploys a higher number of RTs. Since both sets can restore the gate to berth connection after the 11^{th} hour, the productivity resulted from both actions is also similar. However, Action Set C incurs extra cost for mobilising the RTs, thus, it has a slightly lower fitness value than Action Set A. Action Set D involves increasing the number of RTs without seeking to rent or repair the unloading equipment. The productivity of this set is significantly lower. This outcome indicates that the decision to support the road restoration in isolation is useless without considering the seaport operations as a whole. Finally, Action Set E illustrates the dis-benefit of repairing QC in the golden hours for humanitarian distribution. This repairing action requires a longer time than the renting option, and hence, the value of the total throughput is also significantly lower.

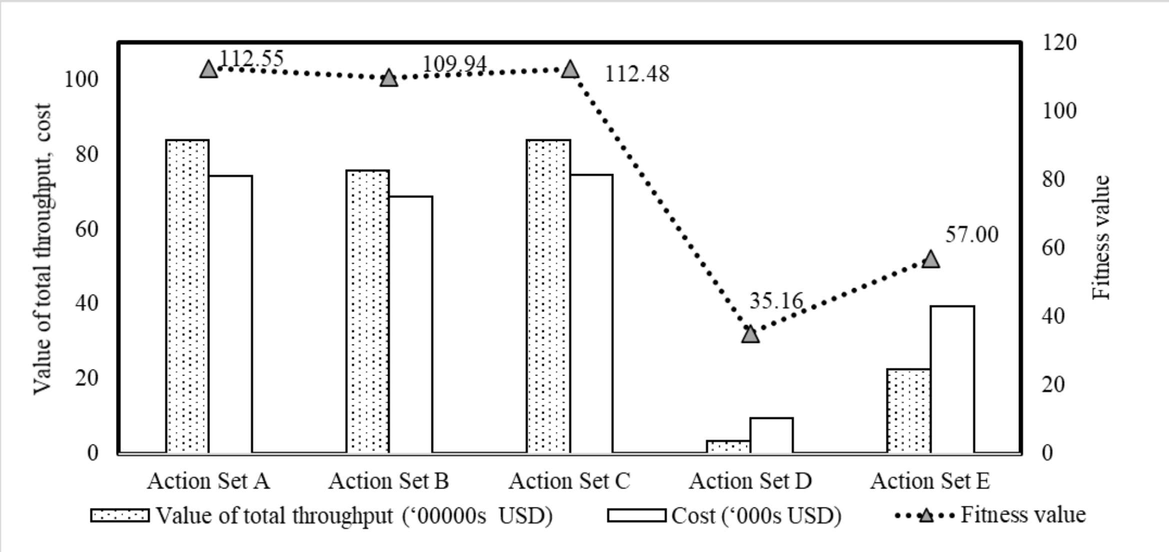

Figure 9: Value of total throughput, cost, and overall efficiency of seaport restoration

Figure 10 shows the maximum throughput by Action Set A over the time horizon, noting that the seaport operation can be restored within 12 hours after a disaster. Although one gang has been ready for unloading in the 1st hour, the seaport remains closed due to the blockage of internal roads. As the restoration of roads progresses, the seaport opens up for operations at the 12th hour. In this period, 25% of the disrupted road network (8 nodes out of 32) will have been restored, constituting the nodes on the critical path from berth to gate and stacking yard. However, rest of the road network (75%) is restored gradually which takes 53 hours in all. Note that the maximum throughput of the seaport will be reached in the 24th hour as the heavy crane rented arrives at that time. Before this time, the unloading processes at the ship are mainly dependent on manual handling by gangs.

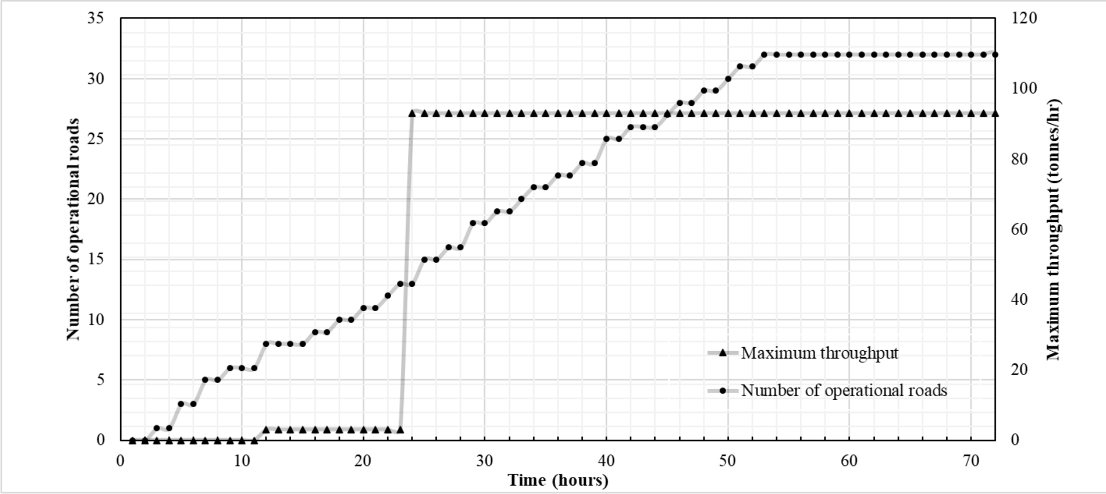

Figure 10: Maximum throughput and the number operational roads over the restoration time horizon

4.3 Comparing the performance of algorithms for solving internal road restoration problem

In this section we evaluate the performance of new DP and the ee-DHA to handle the internal road restoration problem. The road network was adopted from inside the Pantoloan seaport. We varied the number of disrupted nodes - 6 or 9, 22 and 32, with a combination of different number of RTs made available, to create instances namely S1−S6 (small problem), M1− M3 (medium problem), and L1− L3 (large problem) (see **Table 9**). The performance is assessed based on the computation time and the time required for opening the gate-berth connection. We also compare the performance of the proposed algorithms with DHA and greedy algorithm. Greedy algorithm is chosen as it is commonly applied in the literature for tackling the road restoration problem (e.g., Lu *et al*., 2016, Maya Duque *et al*., 2016). The greedy algorithm is specified as in the **Appendix D**.

Table 9: Performance comparison of algorithms for restoring road network

Proble m size	Networ k instanc e	Number of disrupte d nodes	Numbe r of RTs	Hour of operationalising the path of gate-berth				Computation Time (Sec)			
				New DP	ee-DHA	DHA	Greed y	New DP	ee-DHA	DH A	Greed y
Small	S1	6	2	3	3	8	3	24.99	2.37	0.81	0.93
	S2	6	3	3	3	6	3	28.22	1.66	0.80	0.53
	S3	6	4	3	3	5	3	26.44	1.64	0.71	0.58
	S4	9	2	5	5	13	6	3427.54	4.37	1.25	0.55
	S5	9	3	5	5	9	6	3366.53	4.53	0.67	1.67
	S6	9	4	5	5	9	6	6562.71	4.01	0.71	0.87
Medium	M1	22	2		11	29	12		10.34	0.61	0.88
	M2	22	3		11	18	10		7.60	0.70	1.26
	M3	22	4	NA	9	14	10	NA	7.92	0.94	0.85
Large	L1	32	2		11	24	21		7.79	0.71	0.57
	L2	32	3		11	19	18		11.05	0.63	0.82
	L3	32	4		11	18	13		9.66	0.70	0.71

The ee-DHA obtained an identical solution as generated by the new DP in the case of S1 − S6, though the CPU time required is only a fraction of that by new DP (e.g. 0.06% for S6). In the case of M1− M3, and L1− L3 comprising a large number of disrupted nodes, the solution by new DP is not available due to unacceptably high computation time involved. The ee-DHA, DHA, and greedy algorithm could handle this type of problem with ease producing solutions in a very short time. Comparing the solutions provided by the algorithms, we note that the ee-DHA outperforms the DHA and the greedy algorithm. In almost all instances, the ee-DHA could provide a better solution for operationalising the critical gate-berth link, though it requires a slightly higher computation time. The higher the number of disrupted nodes, the better the solution produced by ee-DHA. For instance, in case of L1 where all 32 nodes are disrupted, the solution by ee-DHA allows operationalising the seaport in 11 hours compared to 24 by DHA and 21 by greedy algorithm, thus saving at least 48% of the restoration time.

4.4 Performance of optimisation with GA variants

For handling the binary optimisation problem, we use a new variant of GA, which is compared to GA with Local Search (GALS) and SGA among other types of metaheuristics. The maximum possible number of solutions is set to 4500, similar to Yamada *et al.*, (2009) and Yamada and Zukhruf (2015) to examine the performance of the proposed solution technique. The best parameter values used in each of the GAs are determined by conducting several numerical tests, which are summarised in **Table 10**. The numerical tests check the most suitable value for each parameter in which we test a value from 10 to 100 for the number of individuals, 0.1 to 1.0 for the crossover rate, 0.01 to 0.1 for the mutation rate, 0.1 to 0.9 for the population variance threshold (for GADELMUT), and 1 to 10 for the number of elites (for GALS and GADELMUT). Each combination of these is evaluated over a total of twenty runs to determine the most suitable set of values to arrive at the best, average, and worst solutions.

Table 10: Parameter values of GAs

Parameters	SGA	GALS	GAINMUT	GADELMUT
Number of generations	20	30	50	100
Number of individuals	225	50	90	45
Crossover rate	0.7	0.5	0.5	0.9
Population variance threshold	-	-	-	0.7
Mutation rate	0.07	0.06	0.03	0.04
Number of elites	-	10	-	10
Number of neighbourhoods	-	3	-	-

The performance of algorithms is tested for the case of large problem (**Table 9**) by optimising the number of RTs, unloading equipment and gangs as presented in the previous section. This problem requires 16 binary decisions (i.e., length of chromosomes), generating 65,535 possible combinations. The performance is also compared to the solution by a complete enumeration, considering all possible solutions to assess whether the global optimal solution is arrived. The comparison will then ensure the quality of solution provided by the GAs. It was found that the fitness value of the solution by GADELMUT is similar to the fitness value resulted from the global optimal solution despite there being several local optima present (see **Figure 11**).

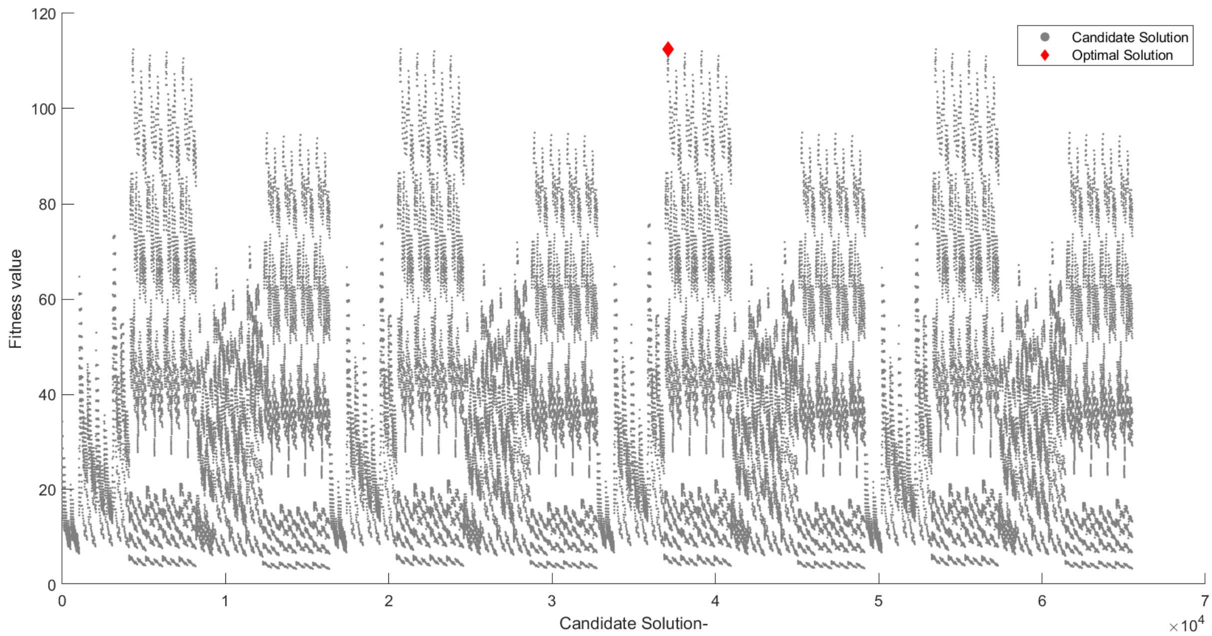

Figure 11: Fitness values with full enumeration of candidate solutions

Table 11 compares the values of objective function obtained by the new variant GADELMUT, with SGA, GALS, and GAINMUT. The computation time was recorded on a PC with Intel(R) Core (TM) i7-9700 CPU @ 3.00GHz and 4.0 GB RAM. To reduce the computation time, we use '*memorization*' scheme for storing the result of road restoration problem based on the number of RTs. The incorporation of *memorization* scheme ensures that a similar RT input numbers will only be computed once by the ee-DHA. This scheme significantly decreases the computation time for solving the optimization problem, as the assignment of RTs by ee-DHA depends on the number of RTs for a given number of disrupted nodes and thus there is no merit in running the routine again.

Table 11 shows that GADELMUT performs better than SGA, both in terms of average and worst values being close to the best value. In addition, GADELMUT provides comparable performance in terms of stability and faster searching ability. GADELMUT could achieve a similar best value as can be obtained by GALS but required only 40% of the CPU time in doing so. Furthermore, GADELMUT demonstrates a better stability in searching for the best value, which is showed by a higher average value than GALS. Therefore, it can be concluded that GADELMUT outperforms GALS in case of 16 binary decisions.

Table 11: Performance comparison of GAs

	Complete Enumeration	SGA	GALS	GAINMUT	GADELMUT
Best Fitness Value	112.55	112. 55	112. 55	112. 55	112. 55
Average Fitness Value		109.86	112.08	109.06	112.50
Worst Fitness Value		102.50	111.69	99.60	111.54
Average CPU Time (sec)	5637.15	109.58	410.68	99.51	166.94

4.5 Performance of GADELMUT with very large-sized problem

We have investigated the performance of GADELMUT further by enlarging the possible number of combinations of decisions to restore the seaport operations to compare with the large problem described earlier. We expand the set of decisions to include the number of light cranes and trucks to hire and the FLs to repair at the yard (henceforth referred as 'very large problem'). This effort generates a higher complexity than before as the problem size now increases to 2,097,151 combinations (i.e., 2^21-1). GADELMUT required 1344 seconds of CPU time in this case to reach the best fitness value. The outcome, referred as Action Set 'VL', involves adding one RT, renting a light crane, adding three teams of the gang each at the dock and yard, and renting two units of FL each at dock and yard, respectively.

Table 12 compares Action Set VL with optimal solution obtained earlier viz., Action Set A. Note that the Action Sets A and VL restore the seaport operation in 12 hours. Yet, Action Set A depends only on one gang available from the 12th to 24th hour, because the heavy crane rented can only be made available from the 24th hour. The Action Set VL indicates the effectiveness of renting a different type of equipment, thus producing higher throughput. The light crane rented in this solution has lower productivity than the heavy crane (see **Table 6**). However, as the light crane arrives sooner at the seaport, Action Set VL will generate higher throughput than Action Set A.

Furthermore, Action Sets A and VL highlight the significance of interdependencies involved in the restoration. The higher productivity of trucks (i.e., 125.87 tonnes/hour) and cranes (i.e., 100-495 tonnes/hour) cannot be met by the productivity of FLs, being equal to 81 tonnes/hour (+3 tonnes/hr by manual RT = 84 tonnes/hr). Hence, the maximum throughput is not as high as the productivity of cranes and trucks. This fact also explains why the optimal solution does not include the addition of trucks.

Table 12: Comparing the solutions for smaller and larger problems

Description	Large problem Action Set A	Very Large problem

		Action Set VL
Number of possible combinations	65,535	2,097,151
Average CPU time by GADELMUT (sec)	166.94	1,344
Starting time of seaport operation (hours)	12th	12th
Maximum throughput over time horizon (tonnes/hour)	93	84
Total throughput (tonnes)	4,593	5,124
Value of total throughput ('000s of USD)	8,382.23	9,351.30
Cost (USD)	74,475	27,925
Fitness value	112.55	334.87

5. Concluding remarks

This paper presented an integrated model to efficiently restore disaster-struck seaport operations considering the interdependencies between infrastructure, equipment and gangs/repair teams. The model selects an appropriate set of restoration actions from a number of possible alternatives. The problem has been set up as an integrated network design and scheduling - INDS, which includes three different decisions, namely, restoration selection decision, task assignment to repair teams and scheduling of repair teams. The uniqueness of INDS problem in this paper is that the interdependencies experienced at a seaport operation are modelled explicitly. In contrast with the INDS problems in the literature, the restoration selection decision not only prioritises the disrupted infrastructure to be repaired but also determines the number of equipment to be added/repaired/rented along with the number of repair teams to be deployed.

To restore the internal road network at a seaport, we propose a new Dynamic Programming that can provide an exact solution. The new DP can optimally solve small-sized problems, however, it requires an unreasonably high computation time for large-sized problem. We then develop an *exploitation-exploration strategy* based Dynamic Hungarian Algorithm – ee-DHA, to efficiently assign/schedule the repair teams for large problems. Resulting decisions are tested within the framework of binary optimisation, by employing a new variant of GA. The variant updates the mutation processes, where it applies the deletion mechanism (i.e., GADELMUT) with an adaptive scheme for updating the mutation rate.

The main conclusions from this paper are summarised as below:

- Restoration of seaport operations involves interdependencies between marine-side and land-side infrastructure/equipment, and any recovery effort must consider the impact of one element on the other in an integrated manner for an efficient restoration. Pantoloan seaport operations can be restored within 12 hours after a disaster albeit with varying levels of throughput depending on the availability of teams and equipment. The optimal solution involves renting of a crane and FLs whereas the repairing option will not be as efficient due to a longer lead time involved potentially threatening the total throughput which is critical in delivering humanitarian aid supplies. Renting a heavy crane will be efficient only if it is matched by the productivity of FLs. If not, renting a light crane, if available, will be a better decision as it produces higher total throughput at a lower cost.
- The proposed DP can efficiently handle the small-sized restoration problems, but in case of large-sized problems, it involves excessive computation time, a fact also noted earlier. Exploitation and exploration-based Dynamic Hungarian Algorithm significantly reduced the computation effort compared to the new DP when dealing with the small-sized problems and is able to match the exact solution. Moreover, ee-

DHA generates better solution by making the seaport operational at least 48% sooner, making the overall restoration operation highly efficient.

- The GA with deletion mutation GADELMUT has been found to perform better than the GALS. It produces comparable solutions to that of GALS in terms of stability and in doing so it needs *only 40% of time* required by the GALS. This is a highly significant outcome and can be adopted into the GA as a standard which saves vast amounts of computation time.

The research presented in this paper can be enhanced further by setting up optimisation problems by other alternative approaches such as Particle Swarm Optimization, Simulated Annealing to compare the solution quality and the computation effort required in solving real-life problems. Also, the new DP and the ee-DHA allow only one crew to operate at each disrupted node which can be relaxed by allowing two or more teams to work simultaneously at any given location potentially quickening the restoration process further. Another worth-studying direction might be related to the development of exact-solution based method for tackling the seaport restoration with a lower computation time.

References

Abdelhafez, M.A., Ellingwood, B., Mahmoud, H., 2021. Vulnerability of seaports to hurricanes and sea level rise in a changing climate: A case study for mobile, AL. Coast. Eng. 167, 103884.

AHA CENTRE, 2018. Situation Update M 7.4 Earthquake & Tsunami, Sulawesi, Indonesia.

Ahmadi, M., Seifi, A., Tootooni, B., 2015. A humanitarian logistics model for disaster relief operation considering network failure and standard relief time: A case study on San Francisco district. Transp. Res. Part E Logist. Transp. Rev. 75, 145–163.

Akbari, V., Salman, F.S., 2017. Multi-vehicle synchronized arc routing problem to restore post-disaster network connectivity. Eur. J. Oper. Res. 257, 625–640.

Almoghathawi, Y., Barker, K., & Albert, L. A. 2019. Resilience-driven restoration model for interdependent infrastructure networks. Reliability Engineering and System Safety, 185, 12–23.

Averbakh, I., 2012. Emergency path restoration problems. Discret. Optim. 9, 58–64.

Averbakh, I., Pereira, J., 2012. The flowtime network construction problem. IIE Trans. (Institute Ind. Eng. 44, 681–694.

Barton, N.H., 2010. Mutation and the evolution of recombination. Philos. Trans. R. Soc. B Biol. Sci. 365, 1281–1294.

Bell, M.G.H., Kurauchi, F., Perera, S., Wong, W., 2017. Investigating transport network vulnerability by capacity weighted spectral analysis. Transp. Res. Part B Methodol. 99, 251–266.

Bellman, R., 1954. The Theory of Dynamic Programming. Bull. Am. Math. Soc.

Bodaghi, B., Shahparvari, S., Fadaki, M., Lau, K.H., Ekambaram, P., Chhetri, P., 2020. Multi-resource scheduling and routing for emergency recovery operations. Int. J. Disaster Risk Reduct. 50, 101780.

Brown, T., 2002. Mutation, Repair and Recombination - Genomes - NCBI Bookshelf. In: Genomes 2nd Edition.

Cavdaroglu, B., Hammel, E., Mitchell, J.E., Sharkey, T.C., Wallace, W.A., 2013. Integrating restoration and scheduling decisions for disrupted interdependent infrastructure systems. Ann. Oper. Res. 203, 279–294.

Çelik, M., 2016. Network restoration and recovery in humanitarian operations: Framework, literature review, and research directions. Surv. Oper. Res. Manag. Sci. 21, 47–61.

Chen, A.Y., Yu, T.Y., 2016. Network based temporary facility location for the Emergency

Medical Services considering the disaster induced demand and the transportation infrastructure in disaster response. Transp. Res. Part B Methodol.91, 408–423

Chen, H., Cullinane, K., Liu, N., 2017. Developing a model for measuring the resilience of a port-hinterland container transportation network. Transp. Res. Part E Logist. Transp. Rev. 97, 282–301.

Chen, Y.-W., Tzeng, G.-H., 1999. A fuzzy multi-objective model for reconstructing the post-quake road-network by genetic algorithm. Int. J. Fuzzy Syst. 1, 85–95.

Dijkstra, E.W., 1959. A note on two problems in connexion with graphs. Numer. Math. 1, 269–271.

Feng, C.-M., Wang, T.-C., 2003. Highway Emergency Rehabilitation Scheduling in Post-Earthquake 72 Hours. J. East. Asia Soc. Transp. Stud. 5, 3276–3285.

Ferrer, J.M., Martín-Campo, F.J., Ortuño, M.T., Pedraza-Martínez, A.J., Tirado, G., Vitoriano, B., 2018. Multi-criteria optimization for last mile distribution of disaster relief aid: Test cases and applications. Eur. J. Oper. Res. 269, 501–515.

Gao, Z., Chen, D., Sun, P., Cai, S., 2018. KM-based efficient algorithms for optimal packet scheduling problem in celluar/infostation integrated networks. Ad Hoc Networks 77, 84–94.

Gharehgozli, A.H., Mileski, J., Adams, A., von Zharen, W., 2017. Evaluating a "wicked problem": A conceptual framework on seaport resiliency in the event of weather disruptions. Technol. Forecast. Soc. Change.121, 65–75.

Ghannad, P., Lee, Y.-C., & Choi, J. O. (2021). Prioritizing Postdisaster Recovery of Transportation Infrastructure Systems Using Multiagent Reinforcement Learning. Journal Of Management in Engineering, 37, 1–13.

Guo, J., Du, Q., He, Z., 2021. A method to improve the resilience of multimodal transport network: Location selection strategy of emergency rescue facilities. Comput. Ind. Eng.

161, 107678.

Iloglu, S., Albert, L.A., 2018. An integrated network design and scheduling problem for network recovery and emergency response. Oper. Res. Perspect. 5, 218–231.

Israeli, E., Gilad, E., 2018. Novel genetic algorithm for loading pattern optimization based on core physics heuristics. Ann. Nucl. Energy 118, 35–48.

John, A., Paraskevadakis, D., Bury, A., Yang, Z., Riahi, R., Wang, J., 2014. An integrated fuzzy risk assessment for seaport operations. Saf. Sci. 68, 180–194.

Kasaei, M., & Salman, F. S. (2016). Arc routing problems to restore connectivity of a road network. Transportation Research Part E: Logistics and Transportation Review, 95, 177–206.

Kuhn, H.W., 1955. The Hungarian Method for the Assignment Problem. Nav. Res. Logist. Q. banner 2, 83–97.

Lagaros, N.D., Karlaftis, M.G., 2011. A critical assessment of metaheuristics for scheduling emergency infrastructure inspections. Swarm Evol. Comput. 1, 147–163.

Lee, E.E., Mitchell, J.E., Wallace, W.A., 2007. Restoration of services in interdependent infrastructure systems: A network flows approach. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. 37, 1303–1317.

Liberatore, F., Ortuño, M.T., Tirado, G., Vitoriano, B., Scaparra, M.P., 2014. A hierarchical compromise model for the joint optimization of recovery operations and distribution of emergency goods in Humanitarian Logistics. Comput. Oper. Res. 42, 3–13.

Lu, C.C., Ying, K.C., Chen, H.J., 2016. Real-time relief distribution in the aftermath of disasters - A rolling horizon approach. Transp. Res. Part E Logist. Transp. Rev. 93, 1–20.

Maya Duque, P.A., Dolinskaya, I.S., Sörensen, K., 2016. Network repair crew scheduling and routing for emergency relief distribution problem. Eur. J. Oper. Res. 248, 272–285.

Moreno, A., Alem, D., Gendreau, M., & Munari, P. (2020). The heterogeneous multicrew scheduling and routing problem in road restoration. Transportation Research Part B: Methodological, 141, 24–58.

Morshedlou, N., González, A.D., Barker, K., 2018. Work crew routing problem for infrastructure network restoration. Transp. Res. Part B Methodol. 118, 66–89.

Munkres, J., 1957. Algorithms for the assignment and transportation problems. J. Soc. Ind. Appl. Math. 5, 32–38.

Nagae, T., Fujihara, T., Asakura, Y., 2012. Anti-seismic reinforcement strategy for an urban road network. Transp. Res. Part A Policy Pract. 46, 813–827.

Nurre, S.G., Cavdaroglu, B., Mitchell, J.E., Sharkey, T.C., Wallace, W.A., 2012. Restoring infrastructure systems: An integrated network design and scheduling (INDS) problem. Eur. J. Oper. Res. 223, 794–806.

O'Rourke, T., 2007. Infrastructure Interdependencies and Resilience. Bridg. Natl. Acad. Eng. 37 37, 22–29.

Oliveira, L. De, Albergaria, R., Bandeira, D.M., 2019. Proposal to planning facility location using UAV and geographic information systems in a post-disaster scenario. Int. J. Disaster Risk Reduct. 36, 101080.

Rabta, B., Wankmüller, C., Reiner, G., 2018. A drone fleet model for last-mile distribution in disaster relief operations. Int. J. Disaster Risk Reduct. 28, 107–112.

Sanci, E., Daskin, M.S., 2019. Integrating location and network restoration decisions in relief networks under uncertainty. Eur. J. Oper. Res. 279 (1), 335–350.

Shafieezadeh, A., Ivey Burden, L., 2014. Scenario-based resilience assessment framework for critical infrastructure systems: Case study for seismic resilience of seaports. Reliab. Eng. Syst. Saf. 132, 207–219.

Sharkey, T.C., Cavdaroglu, B., Nguyen, H., Holman, J., Mitchell, J.E., Wallace, W.A., 2015. Interdependent network restoration: On the value of information-sharing. Eur. J. Oper.

Res. 244, 309–321.

Sharkey, T.C., Nguyen, H., Wallace, W.A., Nurre, S.G., Chow, J.H., Mitchell, J.E., 2016. Identification and classification of restoration interdependencies in the wake of hurricane sandy. J. Infrastruct. Syst. 22.

Taylor, M.A.P., Susilawati, 2012. Remoteness and accessibility in the vulnerability analysis of regional road networks. Transp. Res. Part A Policy Pract. 46, 761–771.

Wei, P., Chen, L., Sun, D., 2014. Algebraic connectivity maximization of an air transportation network: The flight routes' addition/deletion problem. Transp. Res. Part E Logist. Transp. Rev. 61, 13–27.

Yamada, T., Russ, B.F., Castro, J., Taniguchi, E., 2009. Designing multimodal freight transport networks: A heuristic approach and applications. Transp. Sci. 43, 129–143.

Yamada, T., Zukhruf, F., 2015. Freight transport network design using particle swarm optimisation in supply chain-transport supernetwork equilibrium. Transp. Res. Part E Logist. Transp. Rev. 75, 164–187.

Yan, S., Chu, J.C., Shih, Y.L., 2014. Optimal scheduling for highway emergency repairs under large-scale supply-demand perturbations. IEEE Trans. Intell. Transp. Syst. 15, 2378–2393.

Yan, S., Shih, Y.L., 2009. Optimal scheduling of emergency roadway repair and subsequent relief distribution. Comput. Oper. Res. 36, 2049–2065.

Zhang, Y., Lam, J.S.L., 2015. Estimating the economic losses of port disruption due to extreme wind events. Ocean Coast. Manag. 116, 300–310.

Zhang, Y., Lam, J.S.L., 2016. Estimating economic losses of industry clusters due to port disruptions. Transp. Res. Part A Policy Pract. 91, 17–33.

Zhou, L., Lin, Y., Wang, X., Zhou, F., 2019. Model and algorithm for bilevel multisized terminal location-routing problem for the last mile delivery. Int. Trans. Oper. Res. 26, 131–156.

Zukhruf, F., Frazila, R.B., 2020. Optimization Model for Restoring Road Network Disrupted by Disaster.Civil Engineering Journal 27. (in Bahasa)

Zukhruf, F., Kurnia, I.S., Bona Frazila, R., Irawan Nugraha, G., Rizky A.A Syamsunarno, M., 2019. Exploring human genome feature for improving genetic algorithm performance. MATEC Web Conf. 270, 03001.

Appendix A: **The original single-pass Hungarian Algorithm**

Algorithm: **The original single-pass Hungarian Algorithm**

Input: The set of operational nodes ($\boldsymbol{N}$), the set of disrupted nodes ($\boldsymbol{N'}$), the number of RTs (n_1), the location of gate ($\boldsymbol{N}_{gate}$), berth ($\boldsymbol{N}_{berth}$), yard ($\boldsymbol{N}_{yard}$), and RT at depot ($\boldsymbol{N}^{\boldsymbol{R}}_{r,1}$)

Output: a single RT assignment for restoring the disrupted node

1: Set *t*=1, create an assignment matrix with $|\boldsymbol{H_t}| \times |\boldsymbol{N}^{\boldsymbol{H}}_t|$) dimension. $\boldsymbol{N}^{\boldsymbol{H}}_t \in \boldsymbol{N'}$ denotes the set of candidate nodes to be restored, and $\boldsymbol{H}_t \in \boldsymbol{R}$ represents the set of available teams at time- *t*. Locate the candidate node in the top row of the matrix and the available teams in the left most column of the matrix (see **Table 5**).

2: Calculate the assignment cost $\varpi_{r,i} = \Upsilon_r \left(N^R_{r,t}, \alpha_i , \boldsymbol{N'}\right) + s_{\alpha_i} , \forall \alpha_i \in \boldsymbol{N'}, r \in \boldsymbol{R}$

3: **for** *i*=1 to *H* **do**

4: Identify the smallest value of each row *U*

5: Subtract it from every element in its row to obtain *U'*

6: **end for**

7: **for** *j*=1 to *W'* **do**

8: Identify the smallest value of each column *U'*

9: Subtract it from every element in its column to obtain *U''*

10: **end for**

11: For each zero entry, denoted as Z in the matrix U'' .

12: If there is no starred zero in its row and none in its column, star Z, and check each zero in the matrix in turn.

13: Cover every column containing a starred zero,

14: **if** all the columns cover **then** go to line 33:

15: **end if**

16: Find an uncovered zero element in U'' and prime it.

17: **if** there is no starred zero Z in the row containing it **then**

18: Construct a series of alternating primed and starred zeros by:

19: Let Z_0 represent the uncovered primed zero found.

20: Let Z_1 denote the starred zero in the column of Z_0 (if any).

21: Let Z_2 denote the primed zero in the row of Z_1 and there is only one.

22: Continue until the series terminates at a primed zero that has no starred zero in its column.

23: Un-star each starred zero of the series, star each primed zero of the series, erase all primes, and uncover ever line in U.

25: Return to line **14:**

26: **else then**

27: Cover the row and uncover the column of Z until all zeros are covered

28: Record the minimum value of elements that are not covered

29: Add the minimum value found to every element of each covered row

30: Subtract it from every element of each uncovered column.

31: Return to line 17: without altering any stars, primes, or covered lines.

32: **end if**

33: The required assignments are indicated by the positions of the starred zeros in U'' .

34: If *U''* [*r, i*] is a starred zero, then the element associated with row *r* is assigned to the element associated with column *i*.

Appendix B: Dynamic Hungarian Algorithm for Road Restoration

Algorithm: **Dynamic Hungarian Algorithm**

Input: The set of operational nodes ($\boldsymbol{N}$), the set of disrupted nodes ($\boldsymbol{N'}$), the number of RTs (n_1), the location of gate ($\boldsymbol{N}_{gate}$), berth ($\boldsymbol{N}_{berth}$), yard ($\boldsymbol{N}_{yard}$), and RT at depot ($\boldsymbol{N}^{\boldsymbol{R}}_{r,1}$)

Output: RT schedule for restoring the disrupted network

1: Set t=1, create an assignment matrix with ($|\boldsymbol{H_t}| \times |\boldsymbol{N}^{\boldsymbol{H}}_t|$) dimension. $\boldsymbol{N}^{\boldsymbol{H}}_t \in \boldsymbol{N}$ denotes the set of candidate nodes to be restored, and $\boldsymbol{H}_t \in \boldsymbol{R}$ represents the set of available teams at time- t.

2: Locate the candidate node in the top row of the matrix and the available teams in the left most column of the matrix (see **Table** 4).

3: **while** $t \leq \tau$ **do**

4: **for** i=1 **to** $|\boldsymbol{N'}|$ **do**

5: **for** r = 1 **to** n_1 **do**

6: **if** $\varsigma_{rit} = 0$ **then**

7: Calculate the assignment cost,

$$\varpi_{r,i} = \left(\Upsilon_r \ \left(N^R_{r,t}, \alpha_i \ , \boldsymbol{N'}\right) + s_{\alpha_i}\right)\left(\frac{1}{\delta_i}\right), \forall \alpha_i \ \in \boldsymbol{N'}, r \in \boldsymbol{R}$$

8: Update the assignment matrix

9: **end if**

10: **end for**

11: **end for**

12: **if** $\sum_{r=1}^{n_1} \varsigma_{rit} = n_1$ **then** go to 20:

---

else

Invoke the HA for solving the assignment problem (**Appendix A**)

Assign team-r to the selected node-i until the end of restoration duration $\varsigma_{ri,t:t+s_i} = 1$

Add node-α_i to the operational node $\boldsymbol{N}$ and remove node-α_i from $\boldsymbol{N'}$ at $t + s_{\alpha_i}$ and set $\omega_{i,t+s_{\alpha_i}} = 1$

Update the location of RT $N^R_{r,t+s_{\alpha_i}} = \alpha_i$

end if

Estimate the shortest travel time from the gate to berth $\Upsilon(\boldsymbol{N}_{gate}, \boldsymbol{N}_{berth}, \boldsymbol{N'})$ avoiding the disrupted nodes $\boldsymbol{N}$

if $\Upsilon(\boldsymbol{N}_{gate}, \boldsymbol{N}_{berth}, \boldsymbol{N'}) \neq inf$ **then** $open(\boldsymbol{N}_{gate}, \boldsymbol{N}_{berth}) = t$

else $open(\boldsymbol{N}_{gate}, \boldsymbol{N}_{berth}) = \tau$

end if

$t=t+1$

end while

Appendix C: Estimating the price of relief supply package

No	Items	Price per item (CHF)	Weight per item (kg)	Quantity per Package	Price per package (CHF)	Weight per package (kg)	Reference	
							Specifications	links
1	**Family kit (5 persons)**							
a	Hygiene parcel	11.80	8.10	1.00	11.80	8.10	Hygienic parcel for 5 persons/1 month	https://itemscatalogue.redcross.int/wash--5/hygiene--9/hygiene-products--53/hygiene-parcel--HHYGPERS10.aspx
b	Kitchen set	21.30	5.00	1.00	21.30	5.00	Kitchen set family of 5 persons	https://itemscatalogue.redcross.int/relief--4/household--8/household-equipment-kits-and-sets--53/kitchen-set-type-a--KRELCOOSETA.aspx
c	Buckets of 20 litres	15.80	1.22	2.00	31.60	2.44	Bucket, plastic with lid, 20l	https://itemscatalogue.redcross.int/support--4/household--8/household-equipment-and-ustensils--51/buckets--HCONBUCK.aspx
d	Mosquito nets	9.80	1.00	2.00	19.60	2.00	Mosquito net, circular, double bed, d:1050cm, h:220cm	https://itemscatalogue.redcross.int/relief--3/household--8/bedding-and-clothes--7/mosquito-net--HSHEMNET01.aspx
e	Jerry cans, 20 litres	1.50	0.32	2.00	3.00	0.63	Jerrycan, collapsible, 20l, food grade ldpe., screw cap	https://itemscatalogue.redcross.int/relief--4/household--8/household-equipment-kits-and-sets--53/jerrycan-plastic-foldable--HCONJCAN02.aspx

No	Items	Price per item (CHF)	Weight per item (kg)	Quantity per Package	Price per package (CHF)	Weight per package (kg)	Reference	
							Specifications	links
f	Blankets	4.10	1.75	5.00	20.50	8.75	Blanket, woven, 50%wool, 1.5x2m, medium thermal resistance	https://itemscatalogue.redcross.int/relief--4/household--8/bedding-and-clothes--7/blanket-woollen--HSHEBLAN01.aspx
g	Sleeping mats	1.80	0.86	2.00	3.60	1.72	Mat, plastic 180 x 90cm	https://itemscatalogue.redcross.int/food-and-livelihood--3/household--8/bedding-and-clothes--8/mats-for-floor--HSHEMATT.aspx
h	Tarpaulins	10.40	4.20	2.00	20.80	8.40	Tarpaulins, woven plastic, 4 x 6 m, white/white, piece	https://itemscatalogue.redcross.int/water-and-habitat--6/shelter-and-construction-materials--21/family-tents-tarpaulins-accessories--35/tarpaulins-and-plastic-sheeting-roll--HSHETARP.aspx
2	**Food items (5 persons for a month)**							
a	Rice	0.45	1.00	45.00	20.25	45.00	Rice, white broken 5 %, 1kg	https://itemscatalogue.redcross.int/relief--4/food--5/cereals--18/rice--FCERRICE.aspx
b	Oil	1.26	0.96	15.00	18.90	14.40	Oil, palm, 1liter	https://itemscatalogue.redcross.int/relief--4/food--5/edible-oils--31/vegetable-oil--FOILGROU.aspx
c	Sugar	0.66	1.00	7.50	4.95	7.50	Sugar, white, 1kg	https://itemscatalogue.redcross.int/relief--4/food--5/other-food-products--90/sugar-white-crystal--FBAFSUGA.aspx
d	Salt	0.80	1.00	0.75	0.60	0.75	Salt, iodised edible, 1kg	https://itemscatalogue.redcross.int/relief--4/food--5/other-food-products--90/iodized-salt--FBAFSALT.aspx
Total price of package (CHF) and weight (kg)					**176.90**	**104.69**		

No	Items	Price per item (CHF)	Weight per item (kg)	Quantity per Package	Price per package (CHF)	Weight per package (kg)	Reference	
							Specifications	links
Price of total package (USD) #					**191.05**			
Price of total package per tonne (USD)					**1,825**			

1 CHF= 1.08 USD as on 17/4/2021 (https://www.xe.com)

Appendix D: The Greedy algorithm

Algorithm: **The Greedy algorithm**

Input: : The set of operational nodes ($\boldsymbol{N}$), the set of disrupted nodes ($\boldsymbol{N'}$), the number of RTs (n_1), the location of gate ($\boldsymbol{N}_{gate}$), berth ($\boldsymbol{N}_{berth}$), yard ($\boldsymbol{N}_{yard}$), and RT at depot ($\boldsymbol{N}^{\boldsymbol{R}}_{r,1}$)

Output: RT schedule for restoring the disrupted network

1: **for** *t*=1 **to** τ **do**

2: Calculate the assignment cost $\varpi_{r,i} = \left(\Upsilon_r \left(N^R_{r,t}, \alpha_i , \boldsymbol{N'}\right) + s_{\alpha_i}\right), \forall \alpha_i \in \boldsymbol{N'}, r \in \boldsymbol{R}$ for all teams and all disrupted nodes

3: Rank them $\varpi_{r,i}$ in the ascending order of costs

3: Gradually assign the team-*r* to disrupted node-*i* based on the rank

4: Update the number of available teams and candidate nodes

5: **end for**