

# A two-stage stochastic programming model for collaborative asset protection routing problem enhanced with machine learning: a learning-based matheuristic algorithm

Nikzad, E & Bashiri, M

Published PDF deposited in Coventry University's Repository

Original citation:

Nikzad, E & Bashiri, M 2022, 'A two-stage stochastic programming model for collaborative asset protection routing problem enhanced with machine learning: a learning-based matheuristic algorithm', *International Journal of Production Research*, vol. (In-Press).

<https://dx.doi.org/10.1080/00207543.2022.2113928>

DOI 10.1080/00207543.2022.2113928

ISSN 0020-7543

ESSN 1366-588X

Publisher: Taylor and Francis Group

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited..



## A two-stage stochastic programming model for collaborative asset protection routing problem enhanced with machine learning: a learning-based matheuristic algorithm

Erfaneh Nikzad & Mahdi Bashiri

To cite this article: Erfaneh Nikzad & Mahdi Bashiri (2022): A two-stage stochastic programming model for collaborative asset protection routing problem enhanced with machine learning: a learning-based matheuristic algorithm, International Journal of Production Research, DOI: [10.1080/00207543.2022.2113928](https://doi.org/10.1080/00207543.2022.2113928)

To link to this article: <https://doi.org/10.1080/00207543.2022.2113928>



© 2022 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group



Published online: 07 Sep 2022.



Submit your article to this journal [↗](#)



Article views: 424



View related articles [↗](#)



View Crossmark data [↗](#)

# A two-stage stochastic programming model for collaborative asset protection routing problem enhanced with machine learning: a learning-based matheuristic algorithm

Erfaneh Nikzad<sup>a</sup> and Mahdi Bashiri<sup>b</sup>

<sup>a</sup>Industrial Engineering Department, Faculty of Engineering, Shahed University, Tehran, Iran; <sup>b</sup>Centre for Business in Society, Coventry University, Coventry, UK

## ABSTRACT

In this paper, a two-stage stochastic mathematical model is developed for an asset protection routing problem under a wildfire. The main aim of this study is to reduce the negative impact of a wildfire. Some parameters, such as travel and service times, obtaining profit by protecting an asset, and upper bounds of time windows, are considered as stochastic parameters. Generating proper scenarios for uncertain parameters has a large impact on the accuracy of the obtained solutions. Therefore, artificial neural networks are employed to extract possible scenarios according to previous actual wildfire events. The problem cannot be solved by exact solvers for large instances, so two matheuristic algorithms are proposed in this study to solve the problem in a reasonable time. In the first algorithm, a set of feasible routes is generated based on a heuristic approach, then a route-based mathematical model is used to obtain the final solution. Also, another matheuristic algorithm based on adaptive large neighbourhood search (ALNS) is proposed. In this algorithm, routing decisions are determined using the ALNS algorithm while other decisions are achieved by solving an intermediate mathematical model. The numerical analysis confirms the efficiency of both proposed algorithms; however, the first algorithm performs more efficiently.

## ARTICLE HISTORY

Received 31 August 2020  
Accepted 28 July 2022

## KEYWORDS

Asset protection; location routing; two-stage stochastic programming; neural network; matheuristic algorithm

## 1. Introduction

Wildfire disasters pose a major threat to humans and wildlife. Also, wildfire disasters have a significant economic impact (Nuraiman, Ozlen, and Hearne 2020). In Australia, 173 were killed and 414 people were injured during the Black Saturday wildfire in 2009. Also, it forced the evacuation of about 7500 people, and the cost of the damage was almost \$4.5 billion (Shahparvari, Abbasi, and Chhetri 2017). In the 2019/2020 fire season in Australia, almost 2000 homes were destroyed and at least 28 people were killed (Green 2020). Wildfire risk management not only mitigates the risk of wildfire occurrence but also reduces the costs of the damage caused by wildfires.

Wildfire risk management consists of four phases: prevention and mitigation; preparedness; response; and recovery. Prevention and mitigation include activities concerned with preventing wildfires and reducing their negative impact. The activities related to fuel treatment are carried out in the first phase in order to prevent wildfires and reduce their effects. In the second phase, activities focus on adequate preparation for controlling

wildfires. The response phase includes actions such as asset protection and the evacuation process required to reduce the costs of the damage caused by wildfires. The last phase includes actions to repair or rebuild infrastructure during and after wildfire incidents. The activities performed in the preparedness and response phases are more critical because they can affect the efficiency of the actions in other phases. In this study, the strategic and operational decisions for managing a wildfire incident are considered simultaneously. Decisions related to the selection and opening of protection centres, as well as assigning vehicles to them in the preparedness phase, are determined by considering their effects on asset protection in the response phase.

Assets can be part of production systems. A Phillips semiconductor plant faced a fire event in 2000 that led to disruption in production, with a loss of \$400 million to Ericsson (Chopra and Sodhi 2004). Many firms in global supply chains are facing to supply disruptions, and fire can be one of them. Some researchers, such as Wu, Blackhurst, and Chidambaram (2006) and Ho et al. (2015), have indicated that fire is the main risk factor

**CONTACT** Mahdi Bashiri  mahdi.bashiri@coventry.ac.uk  Centre for Business in Society, Coventry University, Coventry, UK

in production-related activities. Therefore, the protection of production-related assets in fires caused by natural disasters could be a very important application of the current study, which needs to be considered in the production research literature.

The values of some factors, such as wind direction, wind speed, monthly rainfall, and monthly maximum temperature, are unknown when decisions related to the first and second phases of wildfire risk management are made. These factors affect the intensity of wildfires and their spread patterns. As a result, some important parameters in wildfire risk management, such as travel and service times, utility obtained by protecting assets, and upper bounds of time windows, might be stochastic in nature. The values of these parameters are unknown, but they can be defined by different scenarios corresponding to each combination of the mentioned factors. Considering uncertainty related to these parameters helps managers to make decisions more accurately. Also, considering the uncertainty of these parameters is critical because it has a significant effect on saving human lives. So, in this study, a two-stage stochastic programming model is proposed for the asset protection problem by considering travel and service times, utility obtained by protecting assets, and upper bounds of time windows as stochastic parameters. The establishment of protection centres and assigning vehicles to them are the first-stage decision variables, which are independent of scenarios. Collaboration between main and backup protection centres and routing decisions are determined in the second stage after revealing the uncertain parameters.

For generating the scenarios, the wildfire spread pattern should be determined. Wildfire spread patterns depend on the severity of wildfire and wind direction. In this study, two neural networks, a generalised regression neural network (GRNN) and a multilayer perceptron artificial neural network are utilised for predicting the intensity of the wildfire based on data obtained from previous wildfires. The efficiency of these networks for estimating the severity of wildfires is examined.

The main contributions of this study can be described as follows: We propose a two-stage stochastic programming model for the asset protection routing problem. Collaboration between protection centres is considered in this paper. In the proposed mathematical model, some decisions related to the second and third phases of wildfire risk management are determined simultaneously. The travel and service times, utility obtained by protecting assets, and upper bounds of time windows are considered as uncertain parameters and are defined by scenarios generated based on real data. A generalised regression neural network and a multilayer perceptron artificial neural network are utilised to predict the severity of the

wildfire used for obtaining the wildfire spread pattern and generating the scenarios. A two-phase learning-based matheuristic algorithm is proposed for solving the mathematical model. In the first phase, a set of feasible routes is determined by a heuristic algorithm. Then a route-based model is solved to obtain the solutions. Also, a matheuristic algorithm based on adaptive large neighbourhood search is developed in this study. All notations, sets, parameters, and decision variables in this study are defined in Table 1 (Nomenclature).

This paper is organised as follows; studies related to optimisation problems under wildfire, asset protection problem, and machine learning are briefly reviewed in Section 2. The problem is defined and formulated in Section 3. The scenario generation method is introduced in Section 4, the solution algorithms are explained in Section 5. Section 6 is devoted to numerical examples. Finally, the conclusions and some managerial insight are provided in the last section.

## 2. Literature review

Many recent studies have focused to reduce wildfires' negative impacts. Wang et al. (2019) proposed a video fire detection that has good accuracy, especially in the indoor environment. Tian, Ren, and Zhou (2016) developed a multi-objective hybrid differential evolution particle swarm optimisation algorithm for scheduling rescue vehicles during forest fires. Wu et al. (2017) proposed a bi-objective model for scheduling fire engines during forest fires and developed an exact method based on dynamic programming as well as a greedy heuristic to solve it. Rashidi et al. (2017) proposed a robust maximal covering location-based model to determine ignition points of the wildfire that cause maximum damage to the landscape. Asset protection and evacuation are the activities done in the response phase to save human lives and to reduce the number of damaged infrastructures. Shahparvari, Abbasi, and Chhetri (2017) proposed a vehicle routing problem for evacuation during wildfires. Also, they developed a heuristic algorithm to solve it. Zhou and Erdogan (2019) proposed a multi-objective two-stage stochastic programming model for allocating firefighters and evacuation humans and then a goal programming approach is applied to solve the problem. Shahparvari and Abbasi (2017) developed a mathematical model to determine the scheduling and routing decisions for evacuating humans under wildfire. They proposed a greedy method to solve the problem.

There are also some previous studies related to the asset protection problem. A limited number of studies have been published on the asset protection problem related to wildfire risk management. However, as the asset

**Table 1.** Nomenclature table.

Type of notations	Notations	Definitions
Sets in Section 3	$\mathcal{I}$	Set of assets, $\mathcal{I} = \{1, 2, \dots, I\}$
	$\mathcal{D}$	Set of protection centres, $\mathcal{D} = \{1, 2, \dots, D\}$
	$\mathcal{D}_M$	Set of main protection centres, $\mathcal{D}_M = \{1, 2, \dots, d_M\}$
	$\mathcal{D}_B$	Set of backup protection centres, $\mathcal{D}_B = \{d_M + 1, d_M + 2, \dots, D\}$
	$\mathcal{M}$	Set of assets and main protection centres, $\mathcal{M} = \{1, 2, \dots, I, I + 1, \dots, I + d_M\}$
	$\mathcal{N}$	Set of assets, main and backup protection centres, $\mathcal{N} = \{1, 2, \dots, I, I + 1, \dots, I + D\}$
	$\mathcal{K}$	Set of vehicles, $\mathcal{K} = \{1, 2, \dots, K\}$
	$\mathcal{A}$	Set of required protections for the assets, $\mathcal{A} = \{1, 2, \dots, A\}$
	$\Xi$	Set of scenarios, $\Xi = \{1, 2, \dots, S\}$
	Parameter in Section 3	$EC_d$
$LB_i$		Lower bound of time window for the $i$ th ( $i \in \mathcal{I}$ ) asset
$UB_i(\xi)$		Upper bound of time window for the $i$ th ( $i \in \mathcal{I}$ ) asset under scenario $\xi$ ( $\xi \in \Xi$ )
$V_i(\xi)$		Obtained utility by protecting the $i$ th ( $i \in \mathcal{I}$ ) asset under scenario $\xi$ ( $\xi \in \Xi$ )
$C_{ij}$		Travel cost between nodes $i$ ( $i \in \mathcal{N}$ ) and $j$ ( $j \in \mathcal{N}$ )
$TT_{ij}(\xi)$		Travel time between nodes $i$ ( $i \in \mathcal{N}$ ) and $j$ ( $j \in \mathcal{N}$ ) under scenario $\xi$ ( $\xi \in \Xi$ )
$ST_i(\xi)$		Servicing time of the $i$ th ( $i \in \mathcal{M}$ ) node under scenario $\xi$ ( $\xi \in \Xi$ ). ( $ST_i(\xi) = 0, \forall i \in \{I + 1, I + 2, \dots, I + d_M\}$ )
$P(\xi)$		Probability of scenario $\xi$ ( $\xi \in \Xi$ )
$T$		An arbitrary large number, $T = \max_{i \in \mathcal{I}, \xi \in \Xi} (UB_i(\xi))$
$\pi_{ak}$		A binary parameter and is equal to 1 if the $a$ th ( $a \in \mathcal{A}$ ) protection activity can be done by the equipment in the $k$ th ( $k \in \mathcal{K}$ ) vehicle, 0 otherwise
$\gamma_{ia}$		A binary parameter and is equal to 1 if the $i$ th ( $i \in \mathcal{I}$ ) asset requires the $a$ th ( $a \in \mathcal{A}$ ) protection activity, 0 otherwise
$\theta_k$		Required equipment, protection components and space for vehicle $k$ ( $k \in \mathcal{K}$ ) in a protection centre
$Cap_d$		Capacity of protection centre $d$ ( $d \in \mathcal{D}$ )
Decision variables in Section 3		$z_d$
	$y_{ikd}(\xi)$	1 if vehicle $k$ ( $k \in \mathcal{K}$ ) starts its route from protection centre $d$ ( $d \in \mathcal{D}_M$ ) and protects node $i$ ( $i \in \mathcal{I}$ ), 0 otherwise
	$h_{kd'd}(\xi)$	1 if vehicle $k$ ( $k \in \mathcal{K}$ ) is transferred from backup protection centre $d'$ ( $d' \in \mathcal{D}_B$ ) to main protection centre $d$ ( $d \in \mathcal{D}_M$ ) under scenario $\xi$ ( $\xi \in \Xi$ ), 0 otherwise
	$q_{kd}$	1 if vehicle $k$ ( $k \in \mathcal{K}$ ) is assigned to protection centre $d$ ( $d \in \mathcal{D}$ ), 0 otherwise
	$x_{ijkd}(\xi)$	1 if vehicle $k$ ( $k \in \mathcal{K}$ ) starts from protection centre $d$ ( $d \in \mathcal{D}_M$ ) travels along $(i, j)$ ( $i, j \in \mathcal{M}$ ) under scenario $\xi$ ( $\xi \in \Xi$ ), 0 otherwise
	$t_{ik}(\xi)$	The time that servicing starts at the $i$ th ( $i \in \mathcal{M}$ ) asset by $k$ th ( $k \in \mathcal{K}$ ) vehicle under scenario $\xi$ ( $\xi \in \Xi$ )
	Parameters in Section 4	$IST_i$
$C_T(\xi)$		Normalised value corresponding to month's maximum temperature under scenario $\xi$ ( $\xi \in \Xi$ )
$C_{RF}(\xi)$		Normalised value corresponding to month's rainfall under scenario $\xi$ ( $\xi \in \Xi$ )
$C_{WS}(\xi)$		Normalised value corresponding to wind speed under scenario $\xi$ ( $\xi \in \Xi$ )
$\alpha_1, \alpha_2$ and $\alpha_3$		The coefficients of the month's maximum temperature, month's rainfall, and wind speed in calculation of service time, respectively
$NC(\xi)$		The number of cells in the region affected by the wildfire under scenario $\xi$ ( $\xi \in \Xi$ )
$Ar(\xi)$		The area of region affected by wildfire under scenario $\xi$ ( $\xi \in \Xi$ )
$AS$		Area of each cell
$IV_i$		The initial obtained utility value by protecting the $i$ th asset
$C_{FS}(\xi)$		The fire time coefficient under scenario $\xi$ ( $\xi \in \Xi$ )
$\Gamma_i(\xi)$		A binary parameter that is equal to 1 when the cell that the $i$ th asset is located on is affected by wildfire
$ITBC_i$		The initial time bound for the $i$ th asset
$\nu_1, \nu_2$ , and $\nu_3$		The coefficients of the month's maximum temperature, month's rainfall and wind speed in determining the upper bounds of time windows
$Itt_{ij}$		The initial travel time between nodes $i$ and $j$
$C_i(\xi)$	Travel time coefficient depends on the severity of wildfire and is determined based on the intensity of wildfire obtained from neural networks	
Sets in Section 5.1	$\mathcal{R}(\xi)$	Set of feasible routes obtained in the first phase of the proposed two-phase matheuristic algorithm under scenario $\xi$ ( $\xi \in \Xi$ ), $\mathcal{R} = \{1, 2, \dots, R(\xi)\}$
	$ln_{ns}$	Set of the assets in the visiting sequence of the $n$ th subset ( $VS_{ns}$ ) that time window constraints violated for them
	$l_{ns}$	Set of the assets in the $n$ th subset
	$route_{da}(\xi)$	The set of obtained routes that start their tour from the $d$ th ( $d \in \mathcal{D}_M$ ) main protection centre for doing protection activity type $a$ ( $a \in \mathcal{A}$ ) under scenario $\xi$ ( $\xi \in \Xi$ )
Parameters in Section 5.1	$\beta_{ir}(\xi)$	Binary parameter that is equal to 1 if the $i$ th node exists in the $r$ th route under scenario $\xi$ ( $\xi \in \Xi$ )
	$L_{id}(\xi)$	Critical level of the $i$ th ( $i \in \mathcal{I}$ ) asset for the main protection centre $d$ ( $d \in \mathcal{D}_M$ ) under scenario $\xi$ ( $\xi \in \Xi$ )
	$\epsilon$	The minimum allowable critical level for the assets to be selected as critical assets in the iterations of the two-phase matheuristic algorithm except iteration 1
	$\epsilon_2$	The minimum allowable critical level for the assets to be selected as critical assets in the first iteration of the two-phase matheuristic algorithm
	$NM_i(\xi)$	Neighbouring measure for the $i$ th ( $i \in \mathcal{I}$ ) asset in the modified nearest neighbour algorithm under scenario $\xi$ ( $\xi \in \Xi$ )

(continued)

**Table 1.** Continued.

Type of notations	Notations	Definitions
	$CV_{ipns}$	Cumulative value of violation in time window constraints when asset $i$ is located in position $p$ in visiting sequence related to the $n$ st subset
	$CC_r(\xi)$	Travelling cost corresponding to the $r$ th ( $r \in \mathcal{R}(\xi)$ ) route under scenario $\xi$ ( $\xi \in \Xi$ )
	$Float_r(\xi)$	The minimum value of violation between the arrival time and the upper bounds of time windows for the assets in the $r$ th ( $r \in \mathcal{R}(\xi)$ ) route under scenario $\xi$ ( $\xi \in \Xi$ )
	$NR_{id}(\xi)$	The number of routes in the set $\mathcal{R}(\xi)$ consisting of the $i$ th asset
	$\lambda_{id}(\xi)$	Random number generated from an interval $\{\min_{j \in \mathcal{N}} \{C_{ij}\}, 2 \times C_{i,(i+d)}\}$ for asset $i$ ( $i \in \mathcal{I}$ ) under scenario ( $\xi$ )
	$tr_i$	The time that servicing starts at node $i$ ( $i \in \mathcal{I}$ ) if it is added in route $VS_{ps}$ after the last node
	$\rho$	A weight to determine the importance of time window constraints in calculation of neighbouring measure
Decision variables in Section 5.1	$u_{rk}(\xi)$	1 if route $r$ ( $r \in \mathcal{R}(\xi)$ ) is assigned to the $k$ th ( $k \in \mathcal{K}$ ) vehicle under scenario $\xi$ ( $\xi \in \Xi$ ), 0 otherwise
Sets in Section 5.2	$\mathcal{R}_{ad}$	Set of routes are considered to serve the assets that require protection activities type $a$ ( $a \in \mathcal{A}$ ) in main protection centre $d$ , $\mathcal{R}_{ad} = \{1, 2, \dots, \sum_{k \in \mathcal{K}} \pi_{ak}\}$
	$NV(\xi)$	Set of assets that have not been visited in the current solution under scenario $\xi$ ( $\xi \in \Xi$ )
	$RL$	Restricted list of assets that their corresponding $MV$ are greater than zero
	$AL_{dr'a}(\xi)$	Set of positions in route $r'$ that requires protection activity type $a$ ( $a \in \mathcal{A}$ ) in main protection centre $d$ ( $d \in \mathcal{D}_M$ ) under scenario $\xi$ , $AL_{dr'a}(\xi) = \{1, 2, \dots,  NN_{dr'a}(\xi)  + 1\}$
Parameters in Section 5.2	$VT_{dar'}(\xi)$	Obtained profit if a vehicle is assigned to the $r'$ th route that requires protection activity type $a$ ( $a \in \mathcal{A}$ ) in main protection centre $d$ ( $d \in \mathcal{D}_M$ ) under scenario $\xi$ ( $\xi \in \Xi$ )
	$CT_{dar'}(\xi)$	Total travel cost if a vehicle is assigned to the $r'$ th route with requiring protection activity type $a$ ( $a \in \mathcal{A}$ ) in main protection centre $d$ ( $d \in \mathcal{D}_M$ ) under scenario $\xi$ ( $\xi \in \Xi$ )
	$zz_d$	A binary parameter is equal to 1 if at least one asset is assigned to the $d$ th ( $d \in \mathcal{D}_M$ ) main protection centre in the current solution
	$Float1_{dar'}(\xi)$	The minimum value of violation between the arrival time and upper bounds of time windows for the assets in the $r'$ th route corresponding to the $d$ th ( $d \in \mathcal{D}_M$ ) main protection centre for protection activity type $a$ under scenario $\xi$ ( $\xi \in \Xi$ )
	$L_d^o(\xi)$	Number of nodes removed by the selected removal strategy from the route corresponding to the $d$ th ( $d \in \mathcal{D}_M$ ) main depot under scenario $\xi$ ( $\xi \in \Xi$ )
	$I_d^o(\xi)$	Number of nodes inserted by the selected insertion strategy to the route corresponding to the $d$ th ( $d \in \mathcal{D}_M$ ) main depot under scenario $\xi$ ( $\xi \in \Xi$ )
	$SS$	Current solution
	$SS_{neighbour}$	Neighbour solution
	$\epsilon_1$	Random number
	$Cim_{dr'aj}(\xi)$	Total travel cost if selected asset ( $\hat{i}$ ) is inserted in the $j$ th position in the route $r'$ that requires protection activity type $a$ ( $a \in \mathcal{A}$ ) in main protection centre $d$ ( $d \in \mathcal{D}_M$ ) under scenario $\xi$ ( $\xi \in \Xi$ )
	$Vim_{dr'aj}(\xi)$	Total obtained profit by protecting the assets in route $r'$ that require protection activity type $a$ ( $a \in \mathcal{A}$ ) in main protection centre $d$ ( $d \in \mathcal{D}_M$ ) under scenario $\xi$ ( $\xi \in \Xi$ )
	$y_{kd}^{best}$	The values for variables $q_{kd}$ in the best solution obtained by the improved ALNS algorithm
	$z_d^{best}$	The values for variables $z_d$ in the best solution obtained by the improved ALNS algorithm
	$NN_{dr'a}(\xi)$	The number of assets in the route $r'$ that requires protection activity type $a$ ( $a \in \mathcal{A}$ ) in main protection centre $d$ ( $d \in \mathcal{D}_M$ ) under scenario $\xi$ ( $\xi \in \Xi$ )
Decision variables in Section 5.2	$w_{dkr'}(\xi)$	1 if vehicle $k$ ( $k \in \mathcal{K}$ ) is assigned to the route $r'$ corresponding to the $d$ th ( $d \in \mathcal{D}_M$ ) main protection centre when the protection activity provided by that vehicle is compatible with the protection activity required by assets in route $r'$ , 0 otherwise
	$ww_{dr'aj}(\xi)$	1 if asset $\hat{i}$ is assigned to the $j$ th position in the route $r'$ that requires protection activity type $a$ ( $a \in \mathcal{A}$ ) in main protection centre $d$ ( $d \in \mathcal{D}_M$ ) under scenario $\xi$ ( $\xi \in \Xi$ ), 0 otherwise.

protection problem can be formulated as a team orienteering problem with time windows (Nuraiman, Ozlen, and Hearne 2020), mentioned studies are reviewed in this section as well including the team orienteering problem with time windows and also the problem with uncertain parameters.

Machine learning helps decision-makers to diagnose the complex relationship between data and make decisions more accurately. Machine learning can be classified into different classes such as supervised, unsupervised, semi-supervised, reinforcement, active, and deep learning (e.g. Ieracitano et al. 2020; Shi, Kang, et al. 2020; Cao et al. 2018; Principi et al. 2019). The studies related to the machine learning method can be

categorised into three classes including optimisation (e.g. Wu et al. 2020; Priore et al. 2019; Liu et al. 2020; Liu, Pang, and Qi 2020; Pandey, Wang, and Boyles 2020; Kim, Jeong, and Shin 2020; Shi, Fan, et al. 2020), learning of an operator behaviour in a heuristic (e.g. Burke, Kendall, and Soubeiga 2003; Bai et al. 2012; Sabar and Kendall 2015; Qu et al. 2020; Choong, Wong, and Lim 2018; Zhao and Zhang 2020; Wang et al. 2020; Shao, Pi, and Shao 2018; Tinós 2020), and estimation of a parameter (e.g. Barteczko-Hibbert, Gillott, and Kendall 2009; Li and Bai 2016; Cui, Rajagopalan, and Ward 2020; Chen, Guo, and Zhao 2020; Yao and Cao 2020).

In this section, previous studies are investigated in the following three sub-sections: Studies related to the Asset

Protection Problem, Studies related to the use of machine learning for optimisation, and finally, studies related to using machine learning for prediction in wildfire-related researches.

### 2.1. The asset protection problem

van der Merwe et al. (2015) proposed a mathematical model for the asset protection problem by considering different types of vehicles and defining travel time as a function of vehicles' speed. They used the CPLEX solver to optimise the proposed model. Roozbeh, Ozlen, and Hearne (2018) developed an adaptive large neighbourhood search (ALNS) algorithm for asset protection problem during a wildfire. Nuraiman, Ozlen, and Hearne (2020) proposed a two-phase decomposition-based matheuristic algorithm for asset protection problem in a wildfire. In the first phase, the assets are classified into different stages and a vehicle routing problem is solved for each stage. Then the obtained routes in different stages are connected in the second phase. In the mentioned studies related to asset protection in a wildfire, it is assumed that the locations of active protection centres are known and only optimal routes for asset protection are determined. Also, all parameters such as travel and service times are considered as deterministic parameters. Bashiri et al. (2021) proposed a two-stage stochastic programming mathematical model for the asset protection problem. Their proposed model determines opening protection centres and routing decision variables, simultaneously. Also, in their mathematical model, one type of protection centre was considered. They used the Frank–Wolfe progressive hedging algorithm to solve the problem. In our study, a two-stage stochastic programming model is proposed for the asset production routing problem by considering two types of centres including main and backup protection centres. It is assumed that vehicles can be transferred from a backup protection centre to the main protection centre to deal with uncertainty. Also, we developed two matheuristic algorithms to solve the proposed model.

As mentioned before, the asset protection problem can be formulated as a team orienteering problem with time windows. Therefore, the team orienteering problem with time windows and orienteering problem with uncertain parameters are reviewed. The orienteering problem is an NP-hard problem (Tsiligirides 1984) and the team orienteering problem with time windows can be defined as an extension of that problem. Gunawan, Lau, and Vansteenwegen (2016) reviewed the studies related to variants of orienteering problem and the solution methods developed to solve them. Different approaches have been proposed for solving the team orienteering problem

with time windows. Lin and Vincent (2017) proposed a multi-start simulated annealing algorithm for the team orienteering problem with time windows by considering mandatory and optional customers. Yu, Perwira Redi, et al. (2019) proposed a selective discrete particle swarm optimisation for solving the team orienteering problem with time windows and partial scores. Hu, Fathi, and Pardalos (2018) developed a multi-objective evolutionary algorithm based on the combination of decomposition and constraint programming approaches for a multi-objective team orienteering problem with time windows. Yu, Jewpanya, et al. (2019) proposed a hybrid artificial bee colony algorithm for the team orienteering problem with time windows when the obtained score for each node depends on the time of visit. Verbeeck, Vansteenwegen, and Aghezzaf (2016) developed a metaheuristic algorithm for solving the time-dependent orienteering problem by considering travel time as a stochastic parameter. Amarouche et al. (2020) proposed the neighbourhood search algorithm for the team orienteering problem with time windows. They improved the efficiency of the neighbourhood search algorithm by considering the memory mechanism and splitting algorithm. Saeedvand, Aghdasi, and Baltas (2020) proposed a multi-objective team orienteering problem with time windows for robots in rescue operations. They developed a multi-objective evolutionary algorithm for solving the proposed model and also used Q-Learning to consider the dynamic nature of disasters and probable changes.

Based on the authors' knowledge, the uncertainty was considered in limited studies related to the orienteering problem. While considering uncertainty will cause more accurate decisions. The studies related to asset protection problem and orienteering problem are reviewed in Table 2 and compared with this study. In this study, travel and service times, utility value, and upper bounds of time windows are considered as stochastic parameters and a two-stage stochastic model is proposed.

As shown in Table 2, no previous study related to the asset protection and team orienteering problem with time windows considers collaboration between depots. Also, the heuristic and metaheuristic algorithms are applied in most studies related to the asset protection problem and the team orienteering problem with time windows. Recently, matheuristic algorithms have been used widely in studies related to vehicle routing problems e.g. Yu, Fang, et al. (2019). In some types of these algorithms, a heuristic or metaheuristic algorithm is developed to obtain feasible routes. Then a route-based version of the proposed model is solved to achieve a feasible solution. Bertazzi et al. (2019) proposed a matheuristic algorithm for a multi-depot Inventory routing problem. They developed a heuristic algorithm based on the

**Table 2.** Review the studies related to asset protection and orienteering problem.

Reference	Problem type	Uncertain			Assumptions					Solution method	LH
		SP	RO	TUP	Deterministic	Depot	TV	TW	DC		
van der Merwe et al. (2015)	APP				*	M	He	*		E	
Roosbeh, Ozlen, and Hearne (2018)	APP				*	S	He	*		Meta-H	
Nuraiman, Ozlen, and Hearne (2020)	APP				*	M	He	*		Math-H	
Amarouche et al. (2020)	TOP				*	S	Ho	*		H	
Verbeeck, Vansteenwegen, and Aghezzaf (2016)	OP	*		TT		S	Ho	*		Meta-H	
Lin and Vincent (2012)	TOP				*	S	Ho	*		Meta-H	
Hu and Lim (2014)	TOP				*	S	Ho	*		Math-H	
Keshtkaran et al. (2016)	TOP				*	S	Ho			E	
Hanafi, Mansini, and Zanotti (2020)	TOP				*	S	He			E	
Labadie et al. (2012)	TOP				*	S	Ho	*		Meta-H	
Zhang, Ohlmann, and Thomas (2014)	OP	*		WT		S	Ho	*		Meta-H	
Bian and Liu (2018)	OP	*		ST,TT		S	Ho			SAP	
Dolinskaya, Shi, and Smilowitz (2018)	OP	*		TT		S	Ho			H	
Evers et al. (2014)	OP	*		W		S	Ho			Math-H	
Yu, Perwira Redi, et al. (2019)	TOP				*	S	Ho	*		Meta-H	
Karabulut and Tasgetiren (2020)	TOP				*	S	Ho	*		H	
Yu, Jewpanya, et al. (2019)	TOP				*	S	Ho	*		Meta-H	
Amarouche et al. (2020)	TOP				*	S	Ho	*		H	
Saeedvand, Aghdasi, and Baltes (2020)	TOP				*	M	He	*		H	*
Hapsari, Surjandari, and Komarudin (2019)	TOP				*	S	Ho			H	
Chou, Gambardella, and Montemanni (2021)	OP	*		OU, C		S	Ho			H	
Bashiri et al. (2021)	APP	*		TT, ST, UB, OU		M	He	*		FW-PH	
This study		*		TT, ST, UB, OU		M	He	*	*	Math-H	*

Note: Abbreviations: APP, asset protection problem; OP, orienteering problem; TOP, team orienteering problem; M, multi; S, single; He, heterogeneous; Ho, homogeneous; E, exact; Meta-H, metaheuristic; H, heuristic; Mat-H, matheuristic; SAP, simulated aided approach; SP, stochastic programming; RO, robust optimisation; TW, time window; TV, types of vehicles; DC, depot collaboration; TUP, type of uncertain parameter; LH, learning-based heuristic; TT, travel time; WT, waiting time at customers; ST, service time; W, weight; UB, upper bound of the time window; OU, obtained utility; C, customer, FW-PH; Frank-Wolfe progressive hedging algorithm.

clustering of nodes for generating feasible routes. Then a route-based formulation of the multi-depot Inventory routing problem is optimised to obtain a feasible solution. Hu and Lim (2014) proposed a matheuristic algorithm for the orienteering problem with time windows. The simulated annealing and local search algorithms were used to construct a set of feasible routes in the first phase. In the second phase, a route-based formulation of the team orienteering problem with time windows is solved to achieve a high-quality solution. Mancini (2017) developed a two-phase matheuristic algorithm for a vehicle routing problem with time-dependent travel time. In the first phase, feasible routes are constructed by a multi-start random constructive heuristic. A set partitioning formulation of the proposed model is solved in the second phase to obtain a near-optimal solution. Grenouilleau et al. (2019) developed a matheuristic algorithm for home healthcare routing and scheduling problem. In the proposed algorithm, a set partitioning-based formulation is solved based on the feasible routes obtained by a large neighbourhood search heuristic. In this paper, a two-phase matheuristic algorithm is proposed for the asset protection problem. In the first phase, a set of feasible routes are generated based on a developed heuristic algorithm. Then a set partitioning formulation of the asset protection problem is solved to obtain a near-optimal solution. Another matheuristic algorithm based on the ALNS algorithm is developed in this study, as well.

## 2.2. Using machine learning to solve combinatorial problems

Using machine learning for solving combinatorial problems has attracted much interest in recent years. Bengio, Lodi, and Prouvost (2020) reviewed the studies that used machine learning algorithm for solving combinatorial optimisation. Talbi (2016) proposed a combination of a metaheuristic algorithm and machine learning technique to solve optimisation problems. Abbasi et al. (2020) proposed an approach to solve large-sized instances of stochastic optimisation problems by using a machine learning method. They used classification and regression tree, k-nearest neighbours, random forest, and multilayer perceptron artificial neural network models to approximate the relationship between the parameters of the problem and the optimal values of decision variables. Václavík et al. (2018) used an online machine learning technique to approximate a tight bound for the pricing problem in the branch and price algorithm and, as a result, the computational time of the algorithm was reduced. Bayliss et al. (2020) proposed a learning-based heuristic algorithm for solving team orienteering problem. They assumed that travel time between two nodes depends on a drone's flight path between previous nodes. Therefore, a machine learning-based heuristic approach is developed in this paper to predict travel times. Different types of vehicle routing problems are combinatorial problems and a machine learning algorithm is



applied for solving them. A combination of reinforcement learning and metaheuristic or heuristic algorithm is used widely for solving different types of vehicle routing problems. In some studies reinforcement learning is applied to manage neighbourhood search in local search algorithm proposed for solving VRPs. Peng et al. (2020) proposed a learning-based memetic algorithm by applying reinforcement learning for pickup and delivery vehicle routing problem. Chen, Qu, et al. (2020) developed a variable neighbourhood search algorithm with reinforcement learning for open vehicle routing problem with time windows. Qu et al. (2020) developed a reinforcement learning-based grey wolf optimiser algorithm for determining the routing decisions of unmanned aerial vehicles.

### 2.3. Using machine learning for prediction under wildfires

As mentioned before, machine learning can also be applied in the studies to better estimation of parameters in the models and as a result to reduce the uncertainty of decisions and make more accurate decisions. Chen, Bai, et al. (2020) proposed a multi-objective model for scheduling a single bus corridor. They used machine learning to obtain a pattern based on the historical data to mitigate uncertainty. Silva et al. (2020) used machine learning and fuzzy logic to obtain wildfire risk and danger indices according to available data that are used in the dynamic wildfire warning system. Zema et al. (2020) used an artificial neural network to predict the hydrological treatments as well as erosion in burned and non-burned soils after forest wildfires. Watson et al. (2019) examined 10 machine learning models including elastic net regression, generalised additive models (GAM), gradient boosting, k-nearest neighbour regression, lasso regression, linear models, multivariate adaptive regression splines (MARS), neural network, random forest, and support vector machines with a radial basis kernel (SVM) to predict ozone exposure during wildfires. Zhai et al. (2020) used a neural network to predict wildfire spread. Sayad, Mousannif, and Al Moatasime (2019) used artificial neural networks and support vector machine to predict wildfires. In this paper, machine learning is utilised to estimate the fire spread pattern in the wildfire based on historical data. Artificial neural networks are used to determine fire spread pattern approximately. Artificial neural networks are the methods used in supervised learning and applied invariant fields of study. Fire spread pattern can affect other parameters such as utility value and upper bound of time windows. Therefore, an accurate estimation can mitigate the uncertainty of the obtained decisions.

According to the above-mentioned literature review, it is observed that there are no previous studies to consider the cooperation between protection centres for the asset protection problem with efficient solution method development. The main contributions of the current study can be summarised as follows: Developing a two-stage stochastic mathematical model for the asset protection problem with collaboration between protection centres, using the ANN to generate more precise possible scenarios, developing two matheuristic algorithms with a learning mechanism in the first algorithm.

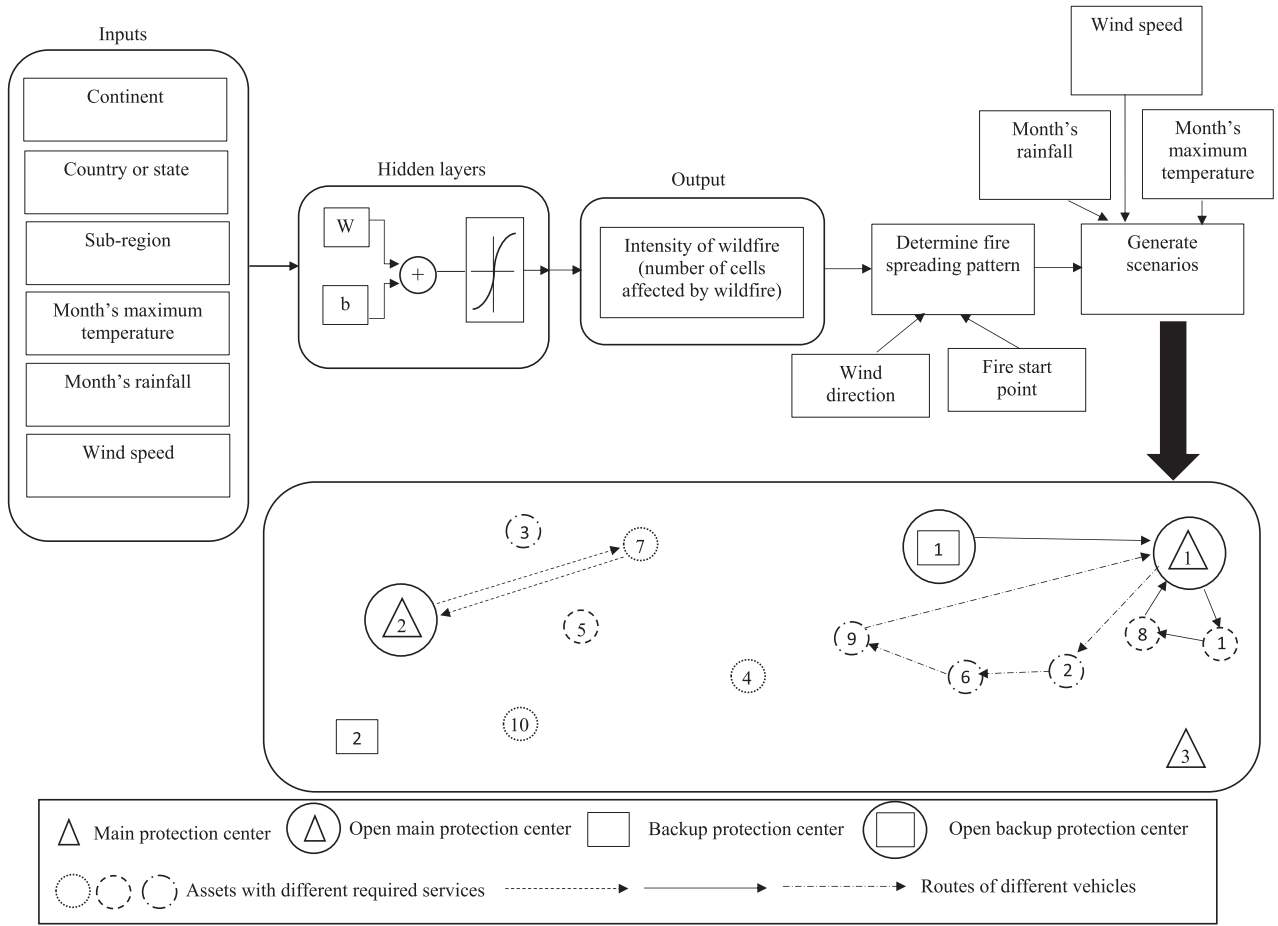
## 3. Problem description and formulation

### 3.1. Problem description

In this paper, a stochastic asset protection routing problem is proposed for determining the locations of fire protection centres and protection of assets in a wildfire by considering uncertainty. Two types of fire protection centres, main and backup protection centres, are considered, and different types of vehicles are assigned to these centres. The vehicles depart from the main protection centres, and backup protection centres are used to transfer more vehicles to the main centres to deal with uncertainty. The locations of the fire protection centres are determined according to the utility obtained by protecting assets and routing costs. Also, the assets are selected for protection based on a trade-off between their obtained utility and corresponding routing cost/time. Therefore, in some cases, assets that are far from the main protection centres might be selected to be protected because of their higher obtained utility.

The wildfire spread pattern may affect some of the model parameters, such as utility obtained by protecting assets and upper bounds of time windows. Therefore, estimating wildfire intensity is necessary to determine the values of uncertain parameters such as the utility obtained by protecting assets and upper bounds of time windows under different scenarios. An artificial neural network is applied to estimate the intensity of the wildfire based on historical data. Also, some additional factors, such as temperature, wind direction, wind speed, etc. may affect the model parameters, so considering these factors makes the decisions more accurate. However, the deterministic values of these factors are unknown in the planning stage, but their probability distribution can be estimated according to historical data. The problem is illustrated in Figure 1. The assumptions of the proposed model are as follows:

- Two types of fire protection centres are considered, main and backup centres. Vehicles are transferred



**Figure 1.** A schematic view of machine learning application on stochastic asset protection problem with protection centres collaboration.

from backup protection centres to main centres to deal with uncertainty.

- A limited number of vehicles can be assigned to each protection centre according to its capacity.
- Different types of vehicles, such as tankers, pumpers, and aerial vehicles, are considered.
- Different types of protection activities, such as cleaning debris and hosing down structures, are considered.
- The travel and service times, upper bounds of time windows, and utility obtained by protecting assets are considered as uncertain parameters and defined by the scenarios.

### 3.2. Problem formulation

In this section, a two-stage stochastic programming model for an asset protection routing problem in a wildfire is developed, with the possibility of transferring vehicles from backup protection centres to main centres. A stochastic asset protection routing problem is defined on a direct network,  $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ , where  $\mathcal{N}$  is the set of nodes

partitioned into a set of assets  $\mathcal{I}$  and a set of protection centres  $\mathcal{D}$  and  $\mathcal{E}$  is the arc set  $\mathcal{E} = \{(i, j) | i, j \in \mathcal{N}\}$ . The sets, parameters, and decision variables of the proposed model are shown in Table 1.

Here, a two-stage stochastic programming model is proposed for an asset protection problem by considering uncertain parameters. Determining the locations of the protection centres and assignment of vehicles to the centres are strategic decision variables that are considered as the first-stage variables. Also, assignment and routing decisions are the second-stage variables. The proposed two-stage stochastic programming model, called SAPP-PC, is presented below.

(SAPP – PC) max

$$\sum_{\xi \in \Xi} P(\xi) \left\{ \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}} \sum_{d \in \mathcal{D}_M} V_i(\xi) \times y_{ikd}(\xi) \right. \\ \left. - \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{M}} \sum_{d \in \mathcal{D}_M} \sum_{k \in \mathcal{K}} C_{ij} \times x_{ijkd}(\xi) \right\}$$

$$\left. \begin{aligned}
& - \sum_{d' \in \mathcal{D}_B} \sum_{d \in \mathcal{D}_M} \sum_{k \in \mathcal{K}} C_{(I+d')(I+d)} \times h_{kd'd}(\xi) \Big\} \\
& - \sum_{d \in \mathcal{D}} EC_d \times z_d \tag{1} \\
\text{s.t. } & \sum_{d \in \mathcal{D}} q_{kd} \leq 1, \quad \forall k \in \mathcal{K} \tag{2} \\
& q_{kd} \leq z_d, \quad \forall k \in \mathcal{K}, d \in \mathcal{D} \tag{3} \\
& \sum_{k \in \mathcal{K}} q_{kd} \geq z_d, \quad \forall d \in \mathcal{D} \tag{4} \\
& y_{ikd}(\xi) \leq q_{kd} + \sum_{d' \in \mathcal{D}_B} h_{kd'd}(\xi), \\
& \quad \forall i \in \mathcal{I}, k \in \mathcal{K}, d \in \mathcal{D}_M, \xi \in \Xi \tag{5} \\
& 2h_{kd'd}(\xi) \leq q_{kd'} + z_d, \\
& \quad \forall d' \in \mathcal{D}_B, k \in \mathcal{K}, d \in \mathcal{D}_M, \xi \in \Xi \tag{6} \\
& y_{ikd}(\xi) \leq y_{(I+d)kd}(\xi), \\
& \quad \forall i \in \mathcal{I}, k \in \mathcal{K}, d \in \mathcal{D}_M, \xi \in \Xi \tag{7} \\
& \sum_{k \in \mathcal{K}} \sum_{d \in \mathcal{D}_M} \left( \sum_{a \in \mathcal{A}} (\pi_{ak} \gamma_{ia}) y_{ikd}(\xi) \right) \leq 1, \\
& \quad \forall i \in \mathcal{I}, \xi \in \Xi \tag{8} \\
& \sum_{\substack{j \in \mathcal{M} \\ i \neq j}} x_{ijkd}(\xi) = y_{ikd}(\xi), \\
& \quad \forall i \in \mathcal{M}, k \in \mathcal{K}, d \in \mathcal{D}_M, \xi \in \Xi \tag{9} \\
& \sum_{\substack{j \in \mathcal{M} \\ i \neq j}} x_{ijkd}(\xi) = \sum_{\substack{j \in \mathcal{M} \\ i \neq j}} x_{jikd}(\xi), \\
& \quad \forall i \in \mathcal{M}, k \in \mathcal{K}, d \in \mathcal{D}_M, \xi \in \Xi \tag{10} \\
& \sum_{k \in \mathcal{K}} \theta_k \times q_{kd} \leq Cap_d, \quad \forall d \in \mathcal{D} \tag{11} \\
& t_{(I+d)k}(\xi) \geq TT_{(I+d')(I+d)}(\xi) \times h_{kd'd}(\xi), \\
& \quad \forall k \in \mathcal{K}, d' \in \mathcal{D}_B, d \in \mathcal{D}_M, \xi \in \Xi \tag{12} \\
& t_{jk}(\xi) \geq t_{ik}(\xi) + ST_i(\xi) + TT_{ij}(\xi) \\
& \quad - T(1 - x_{ijkd}(\xi)), \\
& \quad \forall i \in \mathcal{M}, j \in \mathcal{I}, k \in \mathcal{K}, d \in \mathcal{D}_M, \xi \in \Xi \tag{13} \\
& LB_i \leq t_{ik}(\xi) \leq UB_i(\xi), \quad \forall i \in \mathcal{I}, k \in \mathcal{K}, \xi \in \Xi \tag{14} \\
& z_d, q_{kd} \in \{0, 1\} \quad \forall k \in \mathcal{K}, d \in \mathcal{D} \tag{15} \\
& y_{ikd}(\xi), x_{ijkd}(\xi) \in \{0, 1\}, \\
& \quad \forall i, j \in \mathcal{M}, k \in \mathcal{K}, d \in \mathcal{D}_M, \xi \in \Xi \tag{16} \\
& h_{kd'd}(\xi) \in \{0, 1\},
\end{aligned} \right\}$$

$$\forall k \in \mathcal{K}, d' \in \mathcal{D}_B, d \in \mathcal{D}_M, \xi \in \Xi \tag{17}$$

$$t_{ik}(\xi) \geq 0, \quad \forall i \in \mathcal{I}, k \in \mathcal{K}, \xi \in \Xi \tag{18}$$

The objective function (1) maximises the total achieved profit through the obtained value of protecting assets minus establishment and transportation costs. The third term of the objective function contains the cost corresponding to vehicles transferring from backup to main protection centres. Constraints (2) guarantee that each vehicle can be assigned to at most one protection centre. Constraints (3) ensure that vehicles can be assigned to the established protection centre. Constraints (4) guarantee that at least one vehicle should be assigned to each protection centre. Constraints (5) state that the  $k$ th vehicle can start its route from the main protection centre  $d$  and visit nodes if this vehicle is assigned to the main protection centre  $d$  or transferred from a backup protection centre to this protection centre. Constraints (6) ensure that the  $k$ th vehicle can be transferred from the  $d'$ th backup protection centre to main protection centre  $d$  if this vehicle is assigned to the  $d'$ th backup protection centre and the  $d$ th main protection centre is established. Constraints (7) ensure that the  $k$ th vehicle can protect the  $i$ th asset if this vehicle leaves its protection centre. Constraints (8) ensure that each asset can be visited by at most one vehicle that has the required equipment to protect that asset. Constraints (9) state that if the  $i$ th node is assigned to the  $k$ th vehicle departs from the main protection centre  $d$ , that node should be visited in the route of vehicle  $k$ . Constraints (10) are flow conservation constraints. Constraints (11) state that each protection centre has a capacity, therefore, a limited number of vehicles can be assigned to it. Constraints (12) calculate the arrival time at the main protection centres for vehicles transferred from backup protection centres. Constraints (13) are sub-tour elimination constraints. Constraints (14) are time windows constraints. Constraints (15) to (18) define the domain of the decision variables.

#### 4. Scenario generation method

In this study, scenarios for stochastic parameters are generated based on real data in the south region of Hobart, Tasmania (Australia) based on the procedure proposed by Bashiri et al. (2021). In this study, the mentioned procedure has been improved by using an artificial neural network (ANN). In the above-mentioned procedure, a predefined value is considered to recognise the affected region during a wildfire. While in this study, we assume that the area of the region affected by wildfire depends on some factors such as month's rainfall, month's maximum temperature, etc. Therefore, an artificial neural

network is applied to estimate the area of the affected region during the wildfire based on historical data, more accurately.

Different values of some factors such as wind direction, wind speed, month's rainfall, and month's maximum temperature can affect the parameters of the proposed model. Different levels of factors considered in this paper for generating the scenarios are shown in Table 3. These factors have different dimensions so that mentioned levels should be normalised to remove dimension and balance magnitude differences when they are considered to generate scenarios. Therefore, the normalised values corresponding to each level of wind speed, month's rainfall, and month's maximum temperature factors are reported in Table 3. Different levels of wind direction, month's rainfall, and month's max temperature are determined based on the data obtained from bom.gov.au. Also, the levels of wind speed factors are determined based on historical data about wind speed in previous wildfires in Canada, the United States, and Australia. Parameters under each scenario are calculated based on the normalised values of mentioned factors. The notations used in this section are as mentioned in Table 1.

The service time of each asset under a scenario is obtained by Equation (19).

$$ST_i(\xi) = IST_i \times \alpha_1 \times C_T(\xi) \times \alpha_2 \times C_{RF}(\xi) \times \alpha_3 \times C_{WS}(\xi), \quad \forall i \in \mathcal{I}, \xi \in \Xi \quad (19)$$

The obtained utility by protecting the asset and upper bounds of time windows depends on the fire spread pattern and start point of wildfire as well as the mentioned factors. The fire spread pattern depends on the intensity of the wildfire and wind direction. The intensity of wildfire under each scenario is unknown but some factors that affect it such as the month's maximum temperature, month's rainfall, and wind speed are known. Therefore, the intensity of wildfire (area of the region affected by wildfire) and as a result, the region that has been affected by wildfire should be predicted. As mentioned before, artificial neural networks (ANN) are applied to predict the area of the region that can be affected by a wildfire. More details about the artificial neural network can be found in Section 4.1. Artificial neural networks forecast the intensity of wildfire by considering 6 features as inputs

**Table 3.** Different levels of factors are used for generating scenarios.

Factors	Levels	Normalised values
Wind direction	West (W), Northwest (NW), Southwest (SW)	
Wind speed (km/h)	5, 14, 24	1, 1.5, 2
Month's rainfall (mm)	9, 53, 115	2, 1.6, 1
Month's max temperature (°C)	16, 26, 36	1, 1.5, 2

including continent, country or state, and sub-region, month's maximum temperature, month's rainfall, and wind speed. Data obtained from the wildfires occurred in Canada, the United States, and Australia.

It is assumed that the region is partitioned into  $M$  sub-regions. These sub-regions are called 'cells' and their area are equal. The selected region in Hobart is divided into 36 cells in Figure 2. The number of cells in the region affected by the wildfire under scenario  $\xi$  ( $NC(\xi)$ ) is determined based on the following equation.

$$NC(\xi) = \frac{Ar(\xi)}{AS} \quad (20)$$

The fire spread pattern depends on the start point of wildfire (the cell that wildfire starts from it) and the wind direction as well as the number of cells in the region affected by wildfire. Therefore, the start point of wildfire must be determined under each scenario. The wildfire spread patterns when wind directions are west (W), northwest (NW), and southwest (SW) are shown in Figure 3. It is assumed that the wildfire starts from cell 13 and the number of affected cells by wildfire is equal to 11.

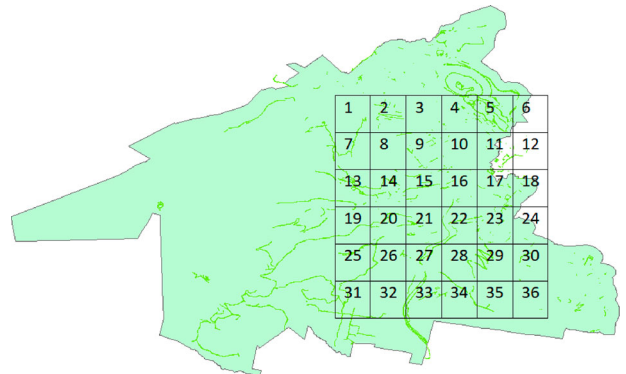
Based on obtained wildfire spread pattern, scenarios for utility value and upper bound of time window are calculated by Equations (21) and (22), respectively.

$$V_i(\xi) = IV_i \times C_{FS}(\xi) \times \Gamma_i(\xi), \quad \forall i \in \mathcal{I}, \xi \in \Xi \quad (21)$$

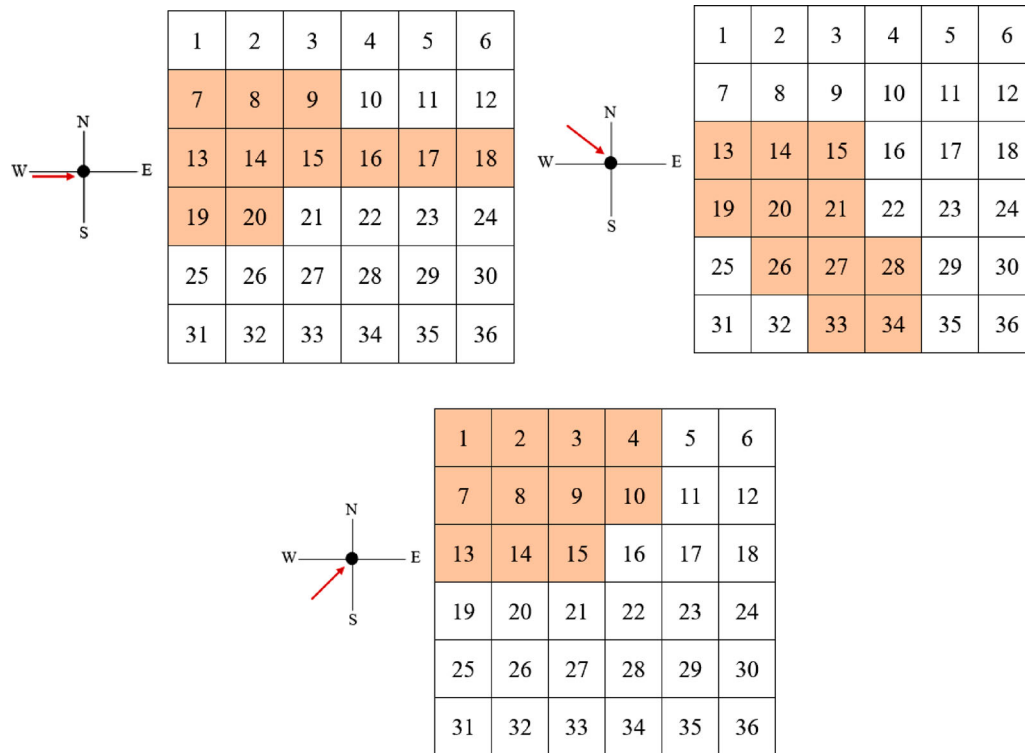
$$UB_i(\xi) = ITBC_i \times v_1 \times C_T(\xi) \times v_2 \times C_{RF}(\xi) \times v_3 \times C_{WS}(\xi) \times \Gamma_i(\xi), \quad \forall i \in \mathcal{I}, \xi \in \Xi \quad (22)$$

The fire time coefficient depends on escape fire time. This coefficient is equal to 3, 2, and 1 for working hours in non-school holidays, working hours in school holidays, and non-working hours, respectively. The travel time between the nodes under each scenario is determined by the following equation.

$$TT_{ij}(\xi) = Itt_{ij} \times C_l(\xi), \quad \forall i, j \in \mathcal{N}, \xi \in \Xi \quad (23)$$



**Figure 2.** Partitioning selected region to 36 sub-regions.



**Figure 3.** The wildfire spread patterns for different wind directions.

#### 4.1. Artificial neural network

An artificial neural network (ANN) is inspired by the neurons in the human brain. Artificial neural networks can be categorised into two main classes including multilayer feedforward networks and recurrent networks based on the network architectures (Zhang 2018). Two different neural networks including general regression neural network algorithm (GRNN) and multilayer perceptron artificial neural network (MLPNN) are used for predicting the wildfire intensity. The efficiency of these types of neural networks for predicting the intensity of wildfire is based on data obtained from the previous wildfire in Canada, the United States, and Australia.

The general regression neural network was proposed by Specht (1991). This algorithm can be applied for regression, prediction, and classification. The GRNN is a one-pass learning algorithm and as a result, backpropagation is not required. Also, the GRNN requires less data for convergence. This algorithm has been used widely in studies in the forecasting fields (e.g. Liang, Niu, and Hong 2019; Niu, Liang, and Hong 2017; Haidar et al. 2011).

The multilayer perceptron artificial neural network (MLPNN) is one of the well-known and useful types of feedforward neural networks. The MLPNN consists of three main layers including input, hidden, and output. One or more hidden layers can be considered in this

algorithm. The MLPNN used backpropagation for training and can estimate the non-linear relationship between inputs and outputs accurately (Bishop et al. 1995).

The parameters of the neural networks are reported in Table 4. The number of input and output layers are 6 and 1, respectively. The input features include continent, country or state and sub-region, month's maximum temperature, month's rainfall, and wind speed. Data obtained from the wildfires occurred in Canada, the United States, and Australia. Also, the month's maximum temperature, month's rainfall, and wind speed in the location and time that wildfire occurred are obtained from the weather spark website (<https://weatherspark.com>).

The MATLAB source code was used to build neural network models. For tuning the parameter of MLPNN including the number of hidden layers and the proportion of data used for training, this algorithm runs 50

**Table 4.** Parameters of multilayer perceptron artificial neural network.

Parameters	Levels
Number of input layers	6
Number of output layers	1
Number of hidden layers	10
Number of neurons in hidden layers	10
Number of samples	85
Proportion of data used for training	0.7
Activation function	Hyperbolic tangent sigmoid

**Table 5.** Mean and standard deviation of MSE under different combinations of number of hidden layers and proportion of data used for training.

Number of hidden layer	Proportion of data used for training	Mean of MSE	Standard deviation of MSE	Average training time
5	0.5	0.025	0.0043	5.05
	0.7	0.023	0.0040	4.93
	0.9	0.028	0.0102	5.06
10	0.5	0.025	0.0023	5.09
	<b>0.7</b>	<b>0.023</b>	<b>0.0028</b>	5.04
	0.9	0.024	0.0049	4.99
50	0.5	0.027	0.0094	10.54
	0.7	0.022	0.0067	10.66
	0.9	0.034	0.0229	10.76
100	0.5	0.091	0.0639	32.17
	0.7	0.030	0.0278	31.96
	0.9	0.021	0.0211	27.27

times under different combinations of these parameters. The average and standard deviation of mean square error (MSE) for each combination are reported in Table 5. Also, the average computational times of learning MLPNN are reported in this table. As reported in this table, the average and standard deviation of MSE when the number of hidden layers and proportion of data used for training are 10 and 0.7, respectively are less than other configurations. Therefore, 10 hidden layers are selected and 0.7 of the data are used for training the MLPNN.

## 5. Solution algorithms

### 5.1. Two-phase matheuristic algorithm

To solve the SAPP-PC efficiently, a two-phase matheuristic algorithm is proposed and the pseudo-code of the developed algorithm is described in Algorithm 1. Also, the main steps of the algorithm are illustrated in Figure 4. In the first phase, the feasible routes are constructed based on the proposed heuristic algorithm. Then, a route-based formulation of SAPP-PC called RBSAPP-PC is constructed according to the obtained routes from the previous phase to be solved and achieve a near-optimal solution for SAPP-PC. The sets, parameters, and decision variables in the matheuristic algorithm have been defined in Table 1.

#### 5.1.1. Constructing feasible routes

Generating a restricted number of feasible routes for SAPP-PC is the aim of this phase. The feasible routes are constructed according to an iterative algorithm. In each iteration, a set of assets that have a greater chance to be selected for protection (critical asset) is determined at first. For this purpose, a random number ( $\lambda_{id}$ ) is generated from an interval  $\{\min_{j \in \mathcal{N}} \{C_{ij}\}, 2 \times C_{i,(I+d)}\}$  for each

asset under each scenario. If the difference between  $\lambda_{id}$  and obtained utility by protecting the  $i$ th asset under scenario  $\xi$  is less than a predefined value ( $\epsilon_2$ ), it is selected as a critical asset. While in other iterations, the critical assets are determined based on a learning approach. In these iterations, a measure ( $L_{id}(\xi)$ ) is calculated for each combination of nodes under each scenario to determine the critical level. The critical level for each asset is determined based on the number of obtained feasible routes which include mentioned asset in previous iterations. On the other hand, the critical level of the assets is achieved based on the learning from the patterns of assets' selection in the previous iterations. The critical level is calculated by the following equation:

$$L_{id}(\xi) = \frac{NR_{id}(\xi)}{|\mathcal{R}(\xi)|} \quad \forall i \in \mathcal{I}, d \in \mathcal{D}_M, \xi \in \Xi \quad (24)$$

A diversification mechanism is applied to avoid trapping in a local optimum solution. In this mechanism, the critical levels are 1 for the assets that did not exist in the set of critical assets in the previous iteration. Therefore, these assets are selected as critical assets in the current iteration. This diversification mechanism is done in all iterations except the first iteration. Then the set of critical assets with the same required activities for protection is defined for each main protection centre under each scenario. For generating the feasible routes, all subsets corresponding to set  $CA_{da}(\xi)$  are generated, and then the travelling salesman problem with time windows (TSPTW) should be solved for each subset to achieve the visiting sequence. To solve the TSPTW efficiently a heuristic algorithm based on the nearest neighbour algorithm (NN) (Kizilates and Nuriyeva 2013) is proposed. The pseudo-code of the developed heuristic algorithm is presented in Algorithm 2. In the nearest neighbour algorithm, the nearest node (having a minimum value of travelling cost) is selected from the set of the unvisited nodes in each iteration. However, in the modified nearest neighbour algorithm (MNN) the nearest node not only is determined based on travelling cost but also the time window constraints are considered to select the nearest node. For this reason, a neighbouring measure ( $NM_i(\xi)$ ) is proposed and calculated for each asset under each scenario based on the following equation.

$$NM_i(\xi) = C_{i^-} + \rho \times (UB_i(\xi) - tr_i) \quad \forall i \in \mathcal{I}, \xi \in \Xi \quad (25)$$

where  $i^-$  is the last node added to the route corresponding to the  $n$ st subset ( $VS_{ns}$ ). Considering the term  $UB_i(\xi) - tr_i$  in the calculation of the neighbouring measure guarantees that the assets have more priority to select as a nearest node when the difference between

**Algorithm 1:** The framework of the proposed two-phase matheuristic algorithm.

---

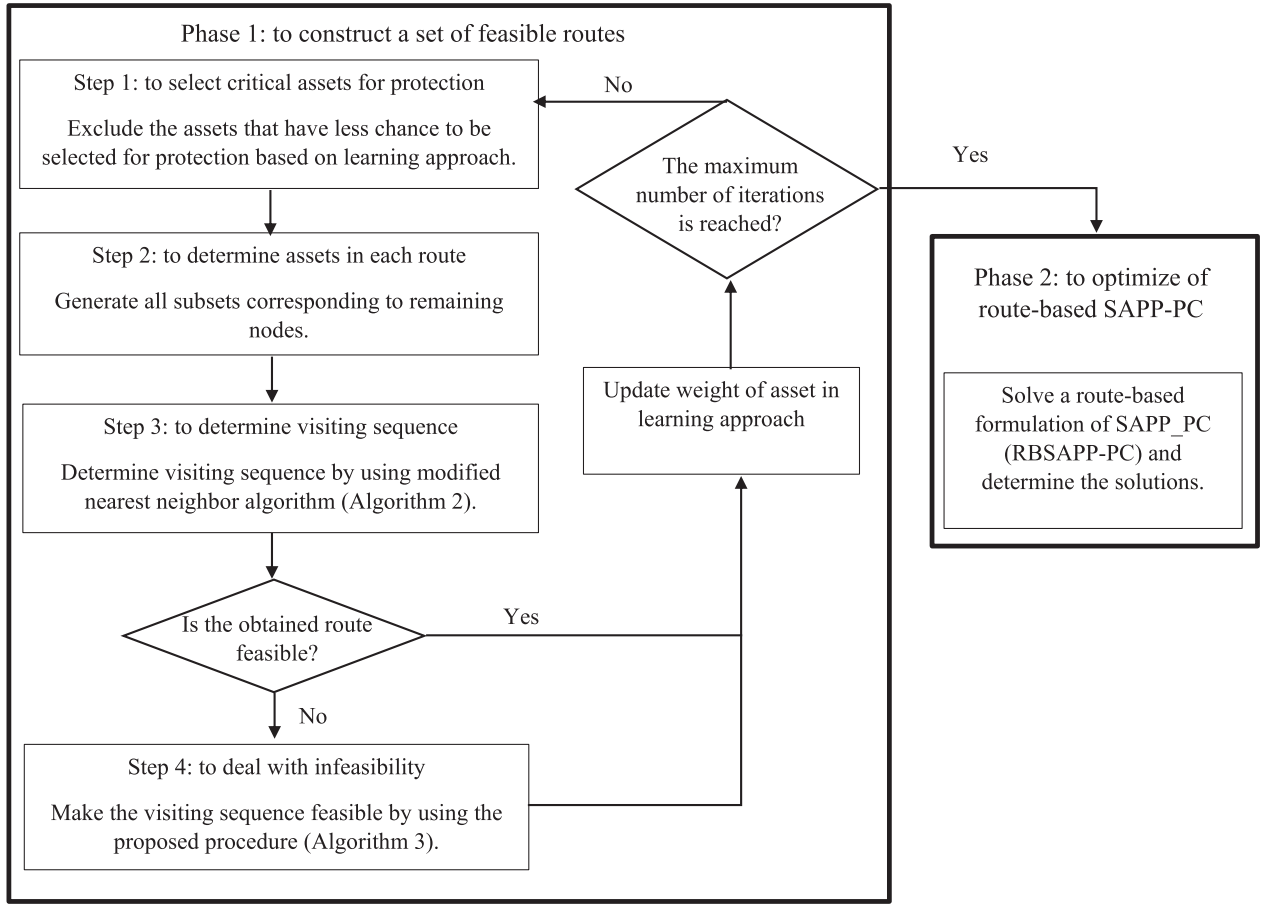
**Input:**  $C_{ij}$ ,  $ST_i(\xi)$ ,  $TT_{ij}(\xi)$ ,  $UB_i(\xi)$ ,  $Maxiteration$ ;  
 $CA_{da}(\xi) \leftarrow []$ ;  
 $iter \leftarrow 1$ ;  
**while**  $iter \leq Maxiteration$  **do**  
  **for**  $d \in \mathcal{D}_M$  **do**  
    **for**  $\xi \in \Xi$  **do**  
      **for**  $i \in \mathcal{I}$  **do**  
        **if**  $iter=1$  **then**  
          Generate the random number  $\lambda_{id}(\xi)$  from the interval  $\{\min_{j \in \mathcal{N}}\{C_{ij}\}, 2 \times C_{i,(I+d)}\}$ ;  
          **if**  $V_i(\xi) - \lambda_{id}(\xi) \geq \epsilon_2$  **then**  
            **for**  $a \in \mathcal{A}$  **do**  
              **if**  $\gamma_{ia} = 1$  **then**  
                 $CA_{da}(\xi) = CA_{da}(\xi) \cup \{i\}$ ;  
            **else**  
              Calculate critical level ( $L_{id}(\xi)$ ) by using Equation (24) ;  
              **if**  $L_{id}(\xi) \geq \epsilon$  **then**  
                **for**  $a \in \mathcal{A}$  **do**  
                  **if**  $\gamma_{ia} = 1$  **then**  
                     $CA_{da}(\xi) = CA_{da}(\xi) \cup \{i\}$ ;  
              **for**  $a \in \mathcal{A}$  **do**  
                Generate all subsets of the set  $CA_{da}(\xi)$ ;  
                **for**  $ns \in \{1, 2, \dots, 2^{|CA_{da}|} - 1\}$  **do**  
                  Obtain the visiting sequence of  $ns^{th}$  subset ( $VS_{ns}$ ) based on the MNN algorithm (Algorithm 2);  
                  **for**  $i \in ns^{th} subset$  **do**  
                    Calculate the arrival time of asset  $i$  in  $VS_{ns}$ ;  
                    **if** the arrival time of asset  $i$  in  $VS_{ns} \geq UB_i(\xi)$  **then**  
                       $In_{ns} \leftarrow In_{ns} \cup \{i\}$ ;  
                    **if**  $In_{ns}$  is empty **then**  
                      **if**  $VS_{ns}$  dose not exist in  $route_{da}(\xi)$  **then**  
                         $route_{da}(\xi) \leftarrow route_{da}(\xi) \cup VS_{ns}$ ;  
                    **else**  
                      Use the proposed algorithm to make the visiting sequence feasible (Algorithm 3);  
                   $\mathcal{R}(\xi) = \mathcal{R}(\xi) \cup route_{da}(\xi)$ ;  
                 $iter = iter + 1$ ;

A route-based formulation of SAPP-PC model (RBSAPP-PC) according to the obtained routes  $\mathcal{R}(\xi)$  is solved to obtain the solution;

---

their arrival times and upper bounds of time windows are small for them. In some cases, the obtained routes from the MNN algorithm are not feasible, therefore, an algorithm is developed to repair the infeasible routes. The pseudo-code of the proposed algorithm is illustrated in Algorithm 3.

In Algorithm 3, the cumulative value of violation in time window constraints is calculated for each asset in different positions of  $VS_{ns}$ . Let  $CV_{ipns}$  is the cumulative value of violation if the  $i$ th asset is located in the  $p$ th position of the route  $VS_{ns}$  when visiting a sequence of other assets in this route is constant. To calculate



**Figure 4.** Overview of the proposed matheuristic algorithm.

$CV_{ipns}$ , the asset  $i$  is removed from its current position in the route  $VS_{ns}$  and is located in the  $p$ th position. Let,  $redVS_{ns}$  is constructed by omitting the  $i$ th asset from the visiting sequence  $VS_{ns}$ . Then the  $i$ th asset is inserted in the  $p$ th position of  $redVS_{ns}$ , therefore, the positions of the nodes located in the positions  $\{1, 2, \dots, p-1\}$  are not changed. However, the nodes in the positions  $\{p, p+1, \dots, |VS_{ns}|-1\}$  are located in new positions as  $\{p+1, p+2, \dots, |VS_{ns}|\}$ , respectively. It can be concluded that the obtained route by locating the  $i$ th asset in the  $p$ th position of the route  $VS_{ns}$  is feasible if  $CV_{ipns}$  is equal to zero. Two situations can occur based on the obtained value of  $CV_{ipns}$  for the assets. In the first condition, there is at least one position for the assets the cumulative value of violation in time window constraints is equal to zero for that position. In this condition, the asset and position corresponding to the cumulative value of violation are selected and the selected asset is located in the new position (selected position). In the second condition, there are not any positions for the assets, therefore, this route can not be considered as a feasible route. Therefore, finding a feasible visiting sequence for all nodes in

the route  $VS_{ns}$  or removing all nodes from the set  $I_{ns}$  are the stopping condition of this algorithm.

### 5.1.2. Optimisation of route-based SAPP-PC

In this phase, a route-based model for the SAPP-PC is proposed according to the feasible routes obtained from the previous phase. A feasible solution for the SAPP-PC is achieved by solving the RBSAPP-PC model.

(RBSAPP – PC) max

$$\begin{aligned}
 & \sum_{\xi \in \Xi} P(\xi) \left\{ \sum_{r \in \mathcal{R}(\xi)} \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{I}} \right. \\
 & \quad \times V_i(\xi) \beta_{ir}(\xi) \times u_{rk}(\xi) \\
 & \quad - \sum_{r \in \mathcal{R}(\xi)} \sum_{k \in \mathcal{K}} CC_r(\xi) \times u_{rk}(\xi) \\
 & \quad \left. - \sum_{d \in \mathcal{D}_M} \sum_{d' \in \mathcal{D}_B} \sum_{k \in \mathcal{K}} C_{(I+d')(I+d)} \times h_{kd'd}(\xi) \right\} \\
 & - \sum_{d \in \mathcal{D}} EC_d \times z_d
 \end{aligned} \tag{26}$$



---

**Algorithm 2:** The framework of modified near-est neighbour (MNN) heuristic for TSPTW.

---

**Input:**  $CA_{da}(\xi)$ ,  $C_{ij}$ ,  $ST_i(\xi)$ ,  $TT_{ij}(\xi)$ ,  $UB_i(\xi)$ ,  $d$ ,  $\xi$ ,  $a$ ,  $ns$ ;  
 $VS_{ns} \leftarrow []$ ;  
 $i^* \leftarrow I + d$ ;  
 $VS_{ns} \leftarrow VS_{ns} \cup \{i^*\}$ ;  
 $Condition \leftarrow false$ ;  
 Define the set of unvisited nodes (UN) (the set of nodes in the  $ns$ th subset of set  $CA_{da}(\xi)$  except  $i^*$ );  
**while**  $condition$  is false **do**  
    $i^- \leftarrow i^*$ ;  
   **for**  $i \in UN$  **do**  
     **if**  $|VS_{ns}| = 1$  **then**  
        $tr_i = TT_{i^*i}(\xi)$ ;  
     **else**  
        $tr_i = tr_{i^*} + ST_{i^*}(\xi) + TT_{i^*i}(\xi)$ ;  
   Calculating the neighbouring measure for each asset in the set  $UN$  based on the Equation (25);  
   Selecting an asset with minimum value of neighbouring measure  
   ( $i^* = \arg \min_{i \in UN} \{NM_i(\xi)\}$ );  
    $VS_{ns} \leftarrow VS_{ns} \cup \{i^*\}$ ;  
    $UN \leftarrow UN \setminus \{i^*\}$ ;  
   **if**  $UN$  is empty **then**  
      $Condition \leftarrow true$ ;  
      $VS_{ns} \leftarrow VS_{ns} \cup \{I + d\}$   
**Result:**  $VS_{ns}$

---

s.t.: (2)–(4), (6), (11)

$$\beta_{(I+d)r}(\xi) u_{rk}(\xi) \leq z_d, \quad \forall r \in \mathcal{R}(\xi), k \in \mathcal{K}, d \in \mathcal{D}_M, \xi \in \Xi \quad (27)$$

$$\beta_{(I+d)r}(\xi) u_{rk}(\xi) \leq q_{kd} + \sum_{d' \in \mathcal{D}_B} h_{kd'd}(\xi), \quad \forall r \in \mathcal{R}(\xi), k \in \mathcal{K}, d \in \mathcal{D}_M, \xi \in \Xi \quad (28)$$

$$\begin{aligned} & Float_r(\xi) + T(1 - \beta_{(I+d)r}(\xi) \times u_{rk}(\xi)) \\ & \geq TT_{(I+d')(I+d)}(\xi) \times h_{kd'd}(\xi), \\ & \forall k \in \mathcal{K}, d' \in \mathcal{D}_B, d \in \mathcal{D}_M, \xi \in \Xi, r \in \mathcal{R}(\xi) \end{aligned} \quad (29)$$

$$\sum_{r \in \mathcal{R}(\xi)} \sum_{k \in \mathcal{K}} \left( \sum_{a \in A} (\pi_{ak} \gamma_{ia}) \beta_{ir}(\xi) u_{rk}(\xi) \right) \leq 1, \quad \forall i \in \mathcal{I}, \xi \in \Xi \quad (30)$$

$$z_d, q_{kd} \in \{0, 1\} \quad \forall k \in \mathcal{K}, d \in \mathcal{D} \quad (31)$$

---

**Algorithm 3:** The framework of developed algorithm to repair infeasible routes.

---

**Input:**  $route_{da}(\xi)$ ,  $VS_{ns}$ ,  $ST_i(\xi)$ ,  $TT_{ij}(\xi)$ ,  $UB_i(\xi)$ ,  $I_{ns}$ ;  
 $condition3 \leftarrow false$ ;  
**while**  $condition3$  is false **do**  
   Selected  $i$  randomly from  $I_{ns}$ ;  
   Omit asset  $i$  from route  $VS_{ns}$  without changing in visiting sequence of other nodes in the route and obtain route  $redVS_{ns}$ ;  
    $p \leftarrow 2$ ;  
   **while**  $p \leq |VS_{ns}| - 1$  **do**  
     Insert the  $i$ th asset in the  $p$ th position of route  $redVS_{ns}$  and update the positions of other nodes;  
     **for**  $j \in VS_{ns}$  **do**  
       Calculate the arrival time to asset  $j$  in the route  $VS_{ns}$ ;  
       **if** Arrival time to asset  $j \geq UB_j(\xi)$  **then**  
          $CV_{ipns} = CV_{ipns} + (\text{arrival time to asset } j - UB_j(\xi))$   
       **if**  $CV_{ipns} = 0$  **then**  
          $condition3 \leftarrow true$ ;  
          $route_{da}(\xi) = route_{da}(\xi) \cup VS_{ns}$ ;  
          $p \leftarrow |VS_{ns}|$ ;  
       **else**  
          $p = p + 1$ ;  
         **if**  $p = |VS_{ns}|$  **then**  
           Remove  $i$  from  $I_{ns}$ ;  
           **if**  $I_{ns}$  is empty **then**  
              $condition3 \leftarrow true$ ;  
**Result:**  $route_{da}(\xi)$

---

$$u_{rk}(\xi) \in \{0, 1\}, \quad \forall r \in \mathcal{R}(\xi), k \in \mathcal{K}, \xi \in \Xi \quad (32)$$

$$\begin{aligned} & h_{kd'd}(\xi) \in \{0, 1\}, \\ & \forall k \in \mathcal{K}, d' \in \mathcal{D}_B, d \in \mathcal{D}_M, \xi \in \Xi \end{aligned} \quad (33)$$

The objective function (26) maximises the profit through obtained values by protecting the assets minus opening and transportation costs. Constraints (27) ensure that the  $r$ th route can be assigned to the established main protection centres. Constraints (28) states that the  $r$ th route can be assigned to the  $k$ th vehicle under scenario  $\xi$  if this vehicle exists in the main protection centre corresponding to the  $r$ th route. Constraints (29) guarantee that the  $r$ th route can be assigned to the vehicles transported from backup to main protection centres if the travel time between these protection centres does not make the

route infeasible. In this constraint,  $Float_r(\xi)$  is the minimum value of violation between the arrival times and the upper bounds of time windows for the assets in the  $r$ th route under scenario  $\xi$  and is calculated by ( $Float_r(\xi) = \min_{i \in \mathcal{I} | \beta_{ir}(\xi)=1} \{UB_i(\xi) - \text{arrival time of asset } i \text{ in route } r\}$ ). Constraints (30) state that each asset should be visited by at most one vehicle that has suitable equipment to do the required protection by this asset. Constraints (31) to (33) define the domain of the decision variables.

## 5.2. The improved adaptive large neighbourhood search

Because of the large number of decision variables in the problem, the classic ALNS (Hemmelmayr, Cordeau, and Crainic 2012) algorithm does not have proper efficiency, so we improved it by enhancing the algorithm by adding mathematical models as additional steps in the algorithm. The pseudo-code of the improved version of the ALNS is illustrated in Algorithm 4. Also, the proposed removal and insertion strategies are explained in this section. Moreover, the improvement phase of the proposed algorithm is illustrated in Algorithm 5. Extra sets, parameters, and decision variables for this algorithm have been defined in Table 1.

As shown in Algorithm 4, first  $L_d^{ro}(\xi)$  nodes are removed from the routes related to the  $d$ th main protection centre under scenario  $\xi$  by using one of the removal strategies. Then  $L_d^{io}(\xi)$  assets are inserted to the routes related to the  $d$ th main protection centre under scenario  $\xi$  according to the selected insertion strategy. The roulette wheel procedure is used for selecting the removal and insertion strategies on the current solution (SS) according to their achieved scores in the previous steps. The feasibility of obtained routes is checked and infeasible routes are repaired. If there is not any feasible location for an asset in the route, it is removed from the route. Other decision variables are determined by solving an intermediate model called the Loc-Int model and then the objective function value of the new solution  $SS_{neighbour}$  is calculated. The neighbour solution is acceptable to be considered as a current solution if  $f(SS_{neighbour}) > f(SS)$  or when a random number ( $\epsilon_1$ ) satisfies  $\epsilon_1 > e^{-\frac{f(SS_{neighbour}) - f(SS)}{Temp}}$ .

### 5.2.1. Solution representation for the improved ALNS algorithm

The solution representation used in this study for the improved ALNS algorithm is defined based on the location routing problem solution and consists of assets ( $\{1, 2, \dots, I\}$ ), main protection centres ( $\{I + 1, I + 2, \dots, I + d_M\}$ ), and a dummy node ( $I + D + 1$ ). Each

row in the solution representation illustrates the routes of the corresponding scenario. For example, the first row shows the routes related to the first scenario. Assets after each main protection centre are assigned to that main protection centre. If an asset is not assigned to the main protection centre under different scenarios, that main protection centre is not established. Assets that have not been visited in the current solution under each scenario are added after the dummy node in the solution. An example with 20 assets, 3 main, and 2 backup protection centres is shown in Figure 5. In this example, the assets are assigned to the main protection centres 2 and 3 (corresponding nodes are 22 and 23, respectively), therefore main depot 1 (the corresponding node is 21) is not established. The assets 10, 12, 18, 9, and 11 are assigned to main protection centre 2, and asset 16 is allocated to main protection centre 3 under scenario 5. Node 26 is a dummy node and the set of assets  $NV(\xi)$  that have not been visited in the current solution for each scenario are mentioned after this node. In the mentioned example  $NV(5) = \{1, 2, 3, 4, 5, 6, 7, 8, 13, 14, 15, 17, 19, 20\}$ .

### 5.2.2. creating an initial solution

The initial solution is created randomly. The first main protection centre is located in the first cell of the solution for all scenarios then a random permutation of the set including assets and other main protection centres is generated for each scenario to determine the routing decision. The assets allocated to each main protection centre are ordered based on their required protection activity without changing the visiting sequence of the asset with the same protection activity. The reason is that assets with different protection activities can not be served by the same vehicle. Figure 6 illustrates the initial solution for a scenario. In this solution, the assets 10, 12, 18, 9, and 11 are assigned to depot 2 (node 22). Assets 10, 12, and 18 require protection activity type 1, and assets 9 and 11 need protection activity type 2. Therefore, ordering the assets based on their required activities will lead to a reasonable allocation of assets to vehicles as shown in this figure.

The modified solution can be infeasible because of time window constraints violation. Therefore, the obtained solution should be repaired. For repairing the infeasible routes the same procedure as mentioned in Algorithm 3 for the two-phase matheuristic algorithm is used. If a feasible sequence does not exist for an asset in a route. It can be visited by another vehicle, therefore, it can be inserted into another route corresponding to that main protection centre with the same type of service. For more explanation of repairing procedure, an example is explained. It is assumed that the obtained visiting sequence for the assets which require protection

**Algorithm 4:** The framework of the improved ALNS algorithm.

**Input:**  $C_{ij}$ ,  $ST_i(\xi)$ ,  $TT_{ij}(\xi)$ ,  $UB_i(\xi)$ ,  $iteration_{max}$ ,  $Temp$  (Initial temperature),  $T_0$  (Frozen temperature),  $\delta$  (Cooling ratio);

Generate an initial solution  $SS$  (as a current solution);

Check the feasibility of obtained routing solution and repair it;

Solve Loc-Int model and determine other decision variables;

Calculate the objective function value according to the obtained values for the decision variables  $f(SS)$ ;

$SS_{best} \leftarrow SS$ ,  $F_{best} \leftarrow f(SS)$ ;

Condition  $\leftarrow$  false;

**while** Condition is false **do**

$iteration \leftarrow 0$ ;

**while**  $iteration < iteration_{max}$  **do**

$SS_{neighbour} \leftarrow SS$ ;

**for**  $d \in \mathcal{D}_M$  **do**

**for**  $\xi \in \Xi$  **do**

                Generate  $L_d^{ro}(\xi)$  and  $L_d^{io}(\xi)$ ;

                Select removal and insertion strategies (RO and IO, respectively) using a roulette wheel scheme;

                Remove  $L_d^{ro}(\xi)$  nodes from routes corresponding to the  $d$ th main protection centre under scenario  $\xi$  in  $SS_{neighbour}$  using RO strategy;

                Insert  $L_d^{io}(\xi)$  nodes to routes corresponding to the  $d$ th main protection centre under scenario  $\xi$  in  $SS_{neighbour}$  using IO strategy;

        Solve Loc-Int model and determine other decision variables;

        Calculate the objective function value according to the obtained values for the decision variables

$f(SS_{neighbour})$ ;

**if**  $f(SS_{neighbour}) > f(SS)$  **then**

$iteration \leftarrow 0$ ;

$SS \leftarrow SS_{neighbour}$ ;

**else**

            Generate a random number  $\epsilon$  ;

**if**  $\epsilon > e^{-\frac{f(SS_{neighbour}) - f(SS)}{Temp}}$  **then**

$iteration \leftarrow 0$ ;

$SS \leftarrow SS_{neighbour}$ ;

**if**  $F_{best} < f(SS_{neighbour})$  **then**

$SS_{best} \leftarrow SS_{neighbour}$ ;

$F_{best} \leftarrow f(SS_{neighbour})$ ;

$iteration_{best} \leftarrow iteration_{best} + 1$

**else**

$iteration \leftarrow iteration + 1$ ;

    Update scores of the insertion and removal strategies according to the achieved objective function values during the algorithm;

$Temp = Temp \times \delta$ ;

**if**  $iteration_{best} > iteration_{best}^{max}$  or  $Temp < T_0$  **then**

        Condition  $\leftarrow$  true;

Improve the best solution by a fix and optimise procedure described in Algorithm 5;

activity type 1 in the modified solution of Figure 6 is infeasible. The Time window constraints are violated for

asset 18. Firstly, according to Algorithm 3, it is added to different positions of the current route to obtain the

21	22	10	23	15	26	1	2	3	4	5	6	7	8	9	11	12	13	14	16	17	18	19	20
21	22	10	12	17	23	15	26	1	2	3	4	5	6	7	8	9	11	13	14	16	18	19	20
21	22	10	12	17	11	13	23	15	16	26	1	2	3	4	5	6	7	8	9	14	18	19	20
21	22	10	17	18	5	11	23	4	15	16	26	1	2	3	6	7	8	9	12	13	14	19	20
21	22	10	12	18	9	11	23	16	26	1	2	3	4	5	6	7	8	13	14	15	17	19	20
21	22	1	6	10	12	18	11	23	16	26	2	3	4	5	7	8	9	13	14	15	17	19	20
21	22	1	23	26	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
21	22	1	11	23	26	2	3	4	5	6	7	8	9	10	12	13	14	15	16	17	18	19	20
21	22	1	8	11	16	23	26	2	3	4	5	6	7	9	10	12	13	14	15	17	18	19	20
21	22	23	26	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Figure 5. Solution representation for the improved ALNS algorithm.

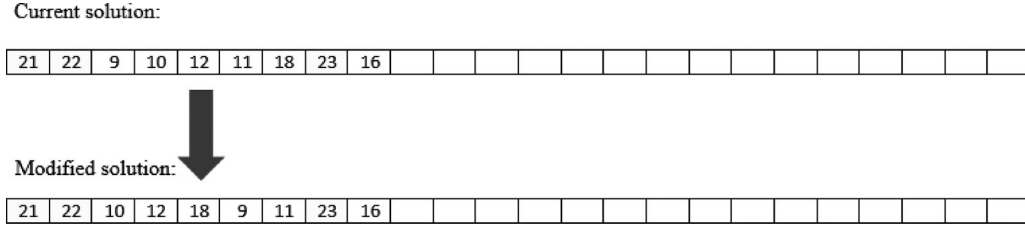


Figure 6. Ordering the routes for each depot according to the assets' required type of protection activities.

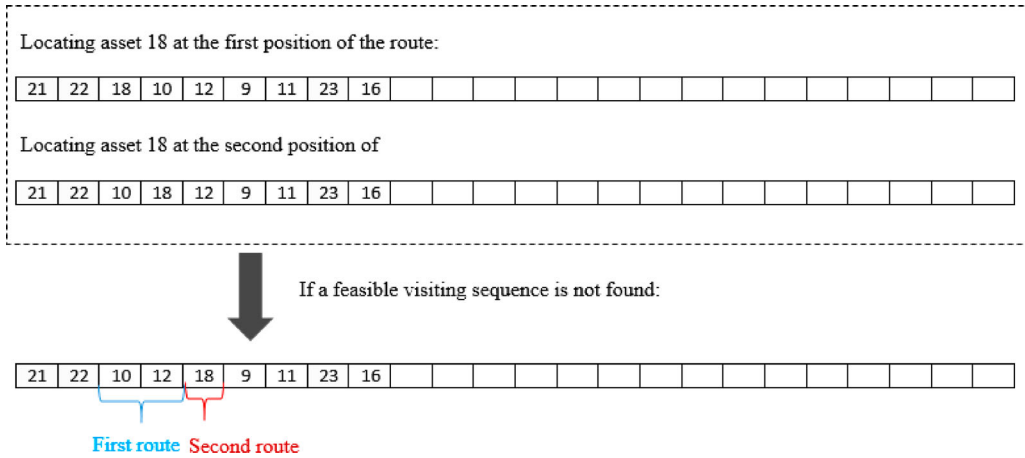


Figure 7. An example of making a solution to be feasible regarding the time window constraints.

feasible visiting sequence. If any feasible visiting sequence is not found, it is added to the second route. The steps are shown in Figure 7. In this example, three routes are considered for the assets assigned to the main protection centre 2. Two routes are related to the assets which require protection activity type 1 and one route includes the assets that need protection activity type 2. Also, a route for the assets that requires protection activity type 3 is assigned to main protection centre 3. Because the type of vehicles assigned to each main protection centre is not determined yet, the maximum number of routes can be considered to serve the assets with requiring protection activity type  $a$  in each main protection centre is determined by  $\sum_{k \in \mathcal{K}} \pi_{ak}$ . If the feasibility can not be guaranteed for an asset assigned to the main protection, then it should be added after the dummy node in the solution.

The decisions about the establishment of protection centres, assigning vehicles to protection centres,

collaborations between protection centres, and selecting the routes based on the vehicles assigned to each protection centre are determined by using the Loc-Int model.

The proposed Loc-Int model is given by:

$$\begin{aligned}
 (\text{Loc} - \text{Int}) \quad & \max \\
 & \sum_{\xi \in \Xi} P(\xi) \left\{ \sum_{d \in \mathcal{D}_M} \sum_{k \in \mathcal{K}} \sum_{a \in \mathcal{A}} \sum_{r' \in \mathcal{R}_{ad}} \right. \\
 & \quad \times (VT_{dar'}(\xi) - CT_{dar'}(\xi)) \times \pi_{ak} w_{dkr'}(\xi) \\
 & \quad \left. - \sum_{d' \in \mathcal{D}_B} \sum_{d \in \mathcal{D}_M} \sum_{k \in \mathcal{K}} C_{(I+d')(I+d)} \times h_{kd'd}(\xi) \right\} \\
 & - \sum_{d \in \mathcal{D}} EC_d \times z_d
 \end{aligned} \tag{34}$$

s.t.: (2)–(4), (6), (11)

$$z_d \leq zz_d, \quad \forall d \in \mathcal{D}_M \quad (35)$$

$$w_{dkr'}(\xi) \leq q_{kd} + \sum_{d' \in \mathcal{D}_B} h_{kd'd}(\xi) \quad (36)$$

$$\forall d \in \mathcal{D}_M, k \in \mathcal{K}, a \in \mathcal{A}, r' \in \mathcal{R}_{ad}, \xi \in \Xi$$

$$\sum_{r' \in \mathcal{R}_{ad}} \pi_{ak} \times w_{dkr'}(\xi) \leq 1 \quad (37)$$

$$\forall d \in \mathcal{D}_M, k \in \mathcal{K}, a \in \mathcal{A}, \xi \in \Xi$$

$$\sum_{k \in \mathcal{K}} \pi_{ak} \times w_{dkr'}(\xi) \leq 1 \quad (38)$$

$$\forall d \in \mathcal{D}_M, r' \in \mathcal{R}_{ad}, a \in \mathcal{A}, \xi \in \Xi$$

$$\sum_{k \in \mathcal{K}} (1 - \pi_{ak}) \times w_{dkr'}(\xi) = 0 \quad (39)$$

$$\forall d \in \mathcal{D}_M, r' \in \mathcal{R}_{ad}, a \in \mathcal{A}, \xi \in \Xi$$

$$\text{Float}1_{dar'}(\xi) + T(1 - w_{dkr'}(\xi) \times \pi_{ak}) \geq TT_{(I+d')(I+d)}(\xi) \times h_{kd'd}(\xi), \quad (40)$$

$$\forall k \in \mathcal{K}, d' \in \mathcal{D}_B, d \in \mathcal{D}_M, \xi \in \Xi, a \in \mathcal{A}, r' \in \mathcal{R}_{ad}$$

$$w_{dkr'}(\xi), z_d, q_{kd}, h_{kd'd}(\xi) \in \{0, 1\} \quad \forall d \in \mathcal{D}_M, d' \in \mathcal{D}_B, k \in \mathcal{K}, r' \in \mathcal{R}_{ad}, a \in \mathcal{A}, \xi \in \Xi \quad (41)$$

Constraints (35) guarantee that the main protection centre can be constructed if at least one asset is assigned to it in the current solution. Constraints (36) ensure that the  $k$ th vehicle can be allocated to the  $r'$ th route in the main protection centre  $d$  if that vehicle is assigned to this main protection centre or is transferred from a backup protection centre to the  $d$ th main protection centre. Constraints (37) guarantee that each vehicle can be assigned to at most one route when it has the required equipment to protect the assets in this route. Constraints (38) state that at most one vehicle with the proper equipment can be assigned to each route. Constraints (39) guarantee that the vehicles can not be assigned to the routes that have not the proper equipment to protect them. Constraints (40) ensure that the vehicles which are transferred from the backup protection centres can be assigned to a route if the time window constraints can be met. Constraints (41) determine the domain of variables.

The solution obtained by the Loc-Int model needs to be modified. The assets that are assigned to the main protection centre that is not established according to the solution of the Loc-Int model are removed from their current cells and are transferred after a dummy node

in the current solution. Also, if any vehicle is not allocated to a route, the assets in this route are removed from current cells and are transferred after the dummy node. For example, if no vehicle is assigned to the route that includes asset 18, it should be removed from its current cell and is considered as a not visited asset.

### 5.2.3. Removal strategies

Two removal strategies are used in the improved ALNS algorithm which were applied by Hammami, Rekik, and Coelho (2020).

- *Random removal*: in this strategy, a specific number of nodes ( $L_d^{ro}(\xi)$ ) are selected randomly for each main protection centre under each scenario and are removed from their corresponding routes.
- *Least profit removal*: in this strategy, a specific number of nodes ( $L_d^{ro}(\xi)$ ) with the least profit are removed from the routes corresponding to the  $d$ th main protection centre under scenario  $\xi$ .

### 5.2.4. Insertion strategies

Five different insertion strategies are designed in this study which is mentioned as follows. The nodes are selected by following strategies and then they are inserted in a feasible cell with the cheapest cost. In the first four strategies, nodes are selected among non-visited and removed nodes, however, in the last strategy nodes are selected among only non-visited nodes.

- *Random selection by considering types of assigned vehicles to main protection centres*: in this strategy, a predefined number of assets ( $L_d^{io}(\xi)$ ) are selected randomly from the non-visited assets (placed after dummy node) and removed assets from other main protection centres under scenario  $\xi$ . In this strategy, during the selection, it is ensured that the required protection activities by the selected assets should be compatible with the types of assigned vehicles to the  $d$ th main protection centre in the SS solution.
- *Selection of highest profit assets by considering types of assigned vehicles to main protection centres*: a predefined number of assets ( $L_d^{io}(\xi)$ ) with top higher profits are selected from the non-visited (placed after dummy node) and removed assets from other main protection centres under scenario  $\xi$ . In this strategy, during the selection, it is ensured that the required protection activities by the selected assets should be compatible with the types of assigned vehicles to the  $d$ th main protection centre in SS solution.

- *Random selection*: in this strategy, a specific number of assets ( $L_d^{io}(\xi)$ ) are selected randomly from the non-visited and removed assets from other main protection centres under scenario  $\xi$ .
- *Highest profit selection*:  $L_d^{io}(\xi)$  number of assets with top higher profits are selected from the non-visited and removed assets from other main protection assets under scenario  $\xi$ .
- *The highest mean value of profit selection*: in this strategy, the mean value of potential achievable profit by adding the  $i$ th asset to the current routes is calculated by  $MV_i = \frac{\sum_{\xi \in \Xi | i \in NV(\xi)} V_i(\xi)}{S} \forall i \in \mathcal{I}$ . Then  $L_d^{io}(\xi)$  number of assets with top higher  $MV$  values are selected from non-visited nodes under scenario  $\xi$ .

### 5.2.5. Improvement procedure

In this phase, the obtained solution from the ALNS algorithm is improved by using a fix and optimise procedure. The pseudo-code of this procedure is illustrated in Algorithm 5. In this procedure, an asset is selected from a restricted list of non-visited assets in each iteration and we try to add it to be visited. The restricted list of assets is selected according to the  $MV$  values. And assets with positive values of  $MV$  are added to the restricted list. In each iteration, an asset with the largest value of  $MV$  is selected from the restricted list. Then the costs related to inserting the selected asset to different positions of routes corresponding to each main protection centre are calculated. If insertion of an asset to a position of a route makes it infeasible, then its cost is assumed to be a large value. This guarantees to achieve a feasible solution using the improvement procedure. Then an intermediate model called the Imp-Int model is solved for the selected asset. The proposed the Imp-Int model is given by:

$$\begin{aligned}
 (\mathbf{Imp} - \mathbf{Int}(\hat{i})) \quad & \max \\
 & \sum_{\xi \in \Xi} P(\xi) \left\{ \sum_{d \in \mathcal{D}_M} \sum_{j \in AL_{dr'a}(\xi)} \sum_{a \in \mathcal{A}} \sum_{r' \in \mathcal{R}_{ad}} \right. \\
 & (V_{\hat{i}}(\xi) + Vim_{dr'aj}(\xi) - Cim_{dr'aj}(\xi)) \times ww_{dr'aj}(\xi) \\
 & \left. - \sum_{d' \in \mathcal{D}_B} \sum_{d \in \mathcal{D}_M} \sum_{k \in \mathcal{K}} C_{(I+d')(I+d)} \times h_{kd'd}(\xi) \right\} \\
 & - \sum_{d \in \mathcal{D}_B} EC_d \times z_d \quad (42)
 \end{aligned}$$

s.t.: (2)–(6), (11), (36)–(41)

$$z_d \leq z_d^{best}, \quad \forall d \in \mathcal{D}_M \quad (43)$$

$$q_{dk} \geq y_{dk}^{best}, \quad \forall d \in \mathcal{D}, k \in \mathcal{K} \quad (44)$$

$$\sum_{d \in \mathcal{D}_M} \sum_{r' \in \mathcal{R}_{ad}} \sum_{a \in \mathcal{A}} \sum_{j \in AL_{dr'a}(\xi)} ww_{dr'aj}(\xi) \leq 1 \quad \forall \xi \in \Xi \quad (45)$$

$$\begin{aligned}
 ww_{dr'aj}(\xi) & \leq w_{dkr'}(\xi) \\
 \forall d \in \mathcal{D}_M, r' \in \mathcal{R}_{ad}, \\
 a \in \mathcal{A}, j \in AL_{dr'a}(\xi), \xi \in \Xi \quad (46)
 \end{aligned}$$

$$\begin{aligned}
 ww_{dr'aj}(\xi) & \in \{0, 1\} \forall d \in \mathcal{D}_M, r' \in \mathcal{R}_{ad}, a \in \mathcal{A}, \\
 j \in AL_{dr'a}(\xi), \xi \in \Xi \quad (47)
 \end{aligned}$$

Equation (42) is the objective function that calculates the expected achieved profit by inserting asset  $\hat{i}$  in the best position. Constraints (43) assure that available active main protection centres be active in the solutions after improvement. constraints (44) state that at least the vehicle assigned to each depot in the best solution of the ALNS algorithm should be assigned to the depots after the improvement phase. Constraints (45) guarantee that the selected asset can be assigned only to a position of one route of a depot. Constraints (46) ensure that the selected asset can be assigned to a route if a vehicle is allocated to that route. Constraints (47) determine the domain of variables.

---

**Algorithm 5:** The framework of the improvement procedure.

---

**Input:**  $SS_{best}$ ;

Calculate  $MV$  for all assets;

Define the set  $RL$  that includes assets with positive  $MV$  values sorted in a descending order

**while**  $RL$  is not empty **do**

Select the first element of set  $RL$  named  $\hat{i}$

Calculate the insertion cost of  $\hat{i}$  in the  $j$ th position of each route;

Solve Imp-Int( $\hat{i}$ ) model and update the decisions ( $z_d^{best}$  and  $y_{dk}^{best}$ )

Remove asset  $\hat{i}$  from  $RL$

---

## 6. Computational results

In this section, the validity of the proposed model (SAPP-PC) and neural networks to generate the scenarios are examined through numerical examples. The efficiency of the proposed matheuristic algorithms is evaluated using the instances generated based on Barreto's benchmark (Schneider and Drexl 2017) and a real case that are available at [http://prodhonc.free.fr/Instances/instances\\_us.htm](http://prodhonc.free.fr/Instances/instances_us.htm). This benchmark has been developed for location routing problem but the considered problem in this paper includes both location and team orienteering problems, simultaneously. Also, all parameters of the

**Table 6.** Specifications of different sizes of problem.

Size of problem	$I$	$K$	$d_M$	$D$	$S$	Number of instances
Small	20, 25, 30	5	3	5	10, 20	30
Medium	35, 40, 45	5	3	5	10, 20	30
Large	50, 55, 60, 70, 80	5	3	5	20	25

**Table 7.** Additional parameters generation scheme based on the benchmark data set for the numerical analysis.

Parameters	Corresponding values or distribution
$C_{ij}$	$\sqrt{(xx_i - xx_j)^2 + (yy_i - yy_j)^2} \times 0.5$
$TT_{ij}(\xi)$	$C_{ij} \times \text{Uniform}[0.5, 1.5]$
$ST_i(\xi)$	$\text{Uniform}[10, 40]$
$V_i(\xi)$	$demand_i \times \text{uniform}[0, 3]$
$UB_i(\xi)$	$\text{Uniform}[0, 0.2] \times \sum_{\xi \in \Xi} \sum_{j \in \mathcal{N}} TT_{ij}(\xi)$
$\rho$	0.2
$\alpha_1, \alpha_2, \alpha_3$	1
$v_1, v_2, v_3$	1

benchmark are deterministic however, there are some stochastic parameters in this study. Therefore, we used the above-mentioned instances with some modifications adding required parameters such as travel times, service times, time windows, and profits to be used in the proposed model. Instances in this paper are generated based on coordChrist50, coordChrist75, and coordChrist100 instances of Barreto's benchmark. To access the data related to this study has been explained in the 'Data availability statement' section. Three sizes of problems are examined in this section and five instances (Ins 1, Ins 2, ..., Ins 5) are considered for each combination of their specifications. As a result, 85 instances are examined in this section. Their specifications are shown in Table 6. The values of some required parameters are not available in the above-mentioned benchmark data set, so the values generating scheme is introduced in Table 7. Based on the locations of nodes that are reported in Barreto's benchmark, the whole area is partitioned into 100 cells. Different scenarios are designed according to the different start points of fire and spread patterns. If asset  $i$  is not located in the cells that are affected by the fire under scenario  $\xi$ , the obtained value by protecting this asset and upper bounds of time windows of that under scenario  $\xi$  will be considered to be zero.

The computations were performed on a 3.5 GHz Workstation with 32 GB RAM and 6 cores operating with Windows 10 (64-bit) and using the Julia software. CPLEX 12.7.1 has been used as a solver.

### 6.1. Examining the effect of considering protection centre collaboration

The collaboration between protection centres reduces the worth of damage by protecting more assets, especially

**Table 8.** Effect of considering collaboration between protection centres in asset protection routing problem.

$I$	$EV_{WDC}$	$EV_{WODC}$	$RGU$	$I$	$EV_{WDC}$	$EV_{WODC}$	$RGU$
20	1135.31	927.81	1.22	30	1377.35	1062.33	1.30
	1139.66	873.99	1.30		1176.86	849.40	1.38
	989.90	774.14	1.28		1422.67	997.70	1.43
	1118.06	883.97	1.26		1472.38	1055.28	1.39
	1123.90	1072.24	1.05		1371.44	1033.11	1.33
25	1266.24	1019.88	1.24	35	1713.80	1033.11	1.66
	1103.24	842.58	1.31		1632.06	1151.96	1.42
	1278.47	997.60	1.28		1715.81	1266.73	1.35
	1267.05	915.83	1.38		1644.60	1134.02	1.45
	1088.88	815.10	1.33		1521.96	1098.99	1.38

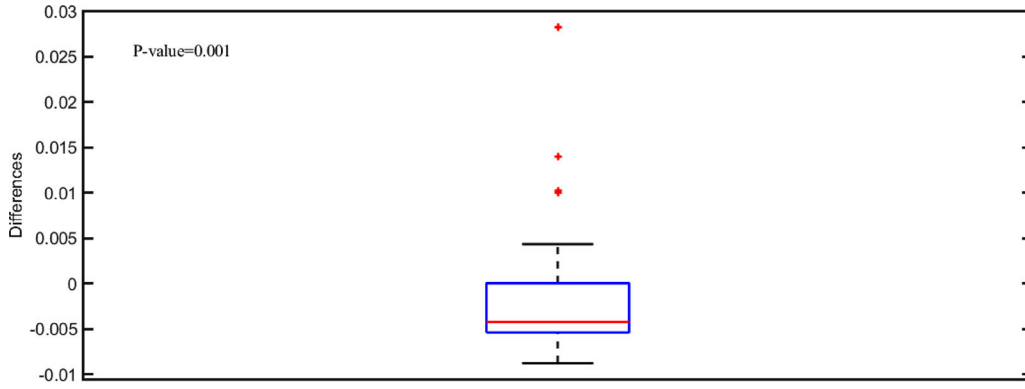
in the case that the available capacity in the main depots is not enough to assign different types of vehicles to them. The expected value of obtained utility with ( $EV_{WDC}$ ) and without ( $EV_{WODC}$ ) considering the collaboration between the main and backup protection centres under the different number of assets are reported in Table 8. In addition, the relative growth in obtained utility ( $RGU$ ) is investigated.  $RGU$  is calculated by  $RGU = EV_{WDC}/EV_{WODC} \times 100\%$ . It is assumed that the numbers of scenarios, main and backup protection centres are 10, 2, and 3, respectively. The results indicate that by considering the collaboration between protection centres, the obtained utility increases 1.34 times on average. It confirms that the collaboration between protection centres will be effective in a wildfire.

### 6.2. Examining the effects of GRNN and MLPNN in predicting the wildfire intensity

A paired  $t$ -test is used to test the null hypothesis that MSE for MLPNN is equal to that obtained for GRNN. For this reason, MLPNN and GRNN were run 50 times and a paired  $t$ -test was investigated to consider a one-tailed alternative hypothesis. The box plot of differences between the obtained MSE values is illustrated in Figure 8. Based on the obtained  $p$ -value (.001), the null hypothesis should be rejected, therefore, it can be concluded that MLPNN has a more accurate estimation of the intensity of wildfire. According to this analysis, MLPNN was used in this study.

### 6.3. Efficiency evaluation of the proposed matheuristic algorithms

The efficiency of the proposed two-phase matheuristic algorithm for solving the SAPP-PC model is presented in this section. The review of previous studies on efficient solution algorithms for the orienteering problem (Hammami, Rekik, and Coelho 2020), asset protection problem (Roosbeh, Ozlen, and Hearne 2018), and



**Figure 8.** Box plot of differences between MSE values when  $H_0$ : mean value of  $(MSE_{MLPNN} - MSE_{GRNN}) = 0$ ;  $H_1$ : mean value of  $(MSE_{MLPNN} - MSE_{GRNN}) < 0$ .

**Table 9.** The tuned parameters for the improved ALNS algorithm.

Parameter	Value
$iteration_{max}$	2000
Initial temperature ( $Temp$ )	100
Cooling rate ( $\delta$ )	0.99
$iteration_{best}^{max}$	10
Frozen temperature ( $T_0$ )	0.001
Improving solution score	1
Global solution score	2
Worse solution score	0

other routing-based problems (Liu, Tao, and Xie 2019; Liu and Jiang 2019; Sun et al. 2020; Hellsten, Sacramento, and Pisinger 2020) confirms that the adaptive large neighbourhood search (ALNS) based algorithms have performed efficiently for these types of problems. Therefore, in this study, the efficiency of the proposed matheuristic algorithm is examined comparing to the improved version of the adaptive large neighbourhood search algorithm as well as the best objective function value obtained by the CPLEX solver in 10,000 s. The tuned parameters for the improved ALNS are mentioned in Table 9.

### 6.3.1. Examining the efficiency of the proposed matheuristic algorithm

The results obtained from the proposed two-phase matheuristic algorithm are compared with the results achieved by using the CPLEX solver for small-size instances (up to 35 assets and considering the different number of scenarios), as reported in Table 10. The best objective function value of the SAPP-PC model ( $Obj_C$ ) obtained by the CPLEX in 10,000 s is reported in this table. In some cases the optimal solutions can not be found in 10,000 s, therefore, in these cases, the best solutions provided by the commercial solver is reported as  $Obj_C$ . The obtained objective function values achieved by the two-phase matheuristic algorithm with learning procedure ( $Obj_{Math-L}$ ) and

without learning procedure ( $Obj_{Math-WL}$ ) are reported in this table as well. The improved ALNS algorithm is solved 10 times for each instance and the best obtained objective function value ( $Obj_{ImpALNS}^B$ ) and the average value of objective function values ( $Obj_{ImpALNS}^M$ ) achieved by this algorithm are reported in Table 10. Also, the average computational time for the improved ALNS ( $Time_{ImpALNS}$ ) algorithm is reported in this table.  $Gap_{c-Math}$  and  $Gap_{c-ImpALNS}$  represent the deviation of the value obtained by the CPLEX from both algorithms of the two-phase matheuristic with learning procedure and improved ALNS, respectively:  $Gap_{c-Math(c-ImpALNS)} = (Obj_C - Obj_{Math-L}(Obj_{ImpALNS}^B)) / Obj_C \times 100\%$ . In addition, the computational times of CPLEX and the matheuristic algorithm are reported in Table 10.

As shown in this table, the achieved objective values for the two-phase matheuristic algorithm with and without the learning mechanism are almost equal but the average computational time of the two-phase matheuristic with the learning mechanism is 2 times less than the average computational time of the two-phase matheuristic without the learning mechanism. Therefore, applying learning mechanism has a great impact to reduce the computational time of the proposed algorithm. The obtained results show that the two-phase matheuristic algorithm with learning mechanism has good performance and it can find near-optimal solutions (0.21% gap on average). As reported in this table, the average computational time of the CPLEX solver is 2 times greater than the average computational time of the two-phase matheuristic algorithm with learning mechanism for small-size instances. While the computational time of the two-phase matheuristic algorithm with learning mechanism for instances with less than 30 assets is greater than the computational time of the CPLEX solver, therefore, it is better to use the CPLEX solver for solving these instances instead of the two-phase matheuristic algorithm. Also, the matheuristic algorithm with the



**Table 10.** Computation results for small-size instances.

<i>I</i>	<i>S</i>	Instance	CPLEX solver (C)		Matheuristic algorithm with learning		Matheuristic algorithm without learning		Improved ALNS algorithm			<i>Gap<sub>C-Math</sub></i> (%)	<i>Gap<sub>C-ImpALNS</sub></i> (%)
			<i>Obj<sub>C</sub></i>	<i>Time<sub>C</sub></i> (s)	<i>Obj<sub>Math-L</sub></i>	<i>Time<sub>Math-L</sub></i> (s)	<i>Obj<sub>Math-WL</sub></i>	<i>Time<sub>Math-WL</sub></i> (s)	<i>Obj<sub>ImpALNS</sub><sup>B</sup></i>	<i>Obj<sub>ImpALNS</sub><sup>M</sup></i>	<i>Time<sub>ImpALNS</sub></i> (s)		
20	10	Ins 1	953.75	6.74	952.91	5.57	952.91	8.01	943.65	851.42	173.28	<b>0.09</b>	1.06
		Ins 2	1008.71	1.74	1007.45	5.10	1007.45	7.59	993.59	932.32	205.94	<b>0.12</b>	1.50
		Ins 3	833.39	1.51	833.05	5.59	833.05	7.49	815.6	755.29	167.146	<b>0.04</b>	2.13
		Ins 4	964.82	7.13	964.47	5.15	964.47	7.49	952.45	883.09	140.77	<b>0.04</b>	1.28
		Ins 5	961.79	1.71	961.79	5.35	961.79	7.57	899.74	843.04	140.96	<b>0.00</b>	6.45
	20	Ins 1	861.44	4.46	860.78	9.70	860.78	19.35	834.26	786.10	470.27	<b>0.08</b>	3.15
		Ins 2	698.30	4.51	694.46	11.43	696.46	13.86	625.87	557.66	532.86	<b>0.55</b>	10.37
		Ins 3	714.38	4.59	713.24	11.53	713.24	14.55	662.64	579.33	369.27	<b>0.16</b>	7.24
		Ins 4	727.39	3.24	727.16	11.31	727.16	14.08	694.13	637.67	434.60	<b>0.03</b>	4.57
		Ins 5	770.74	2.46	770.29	11.79	770.29	13.33	744.48	669.44	497.03	<b>0.06</b>	3.41
25	10	Ins 1	1093.67	3.09	1093.18	8.40	1093.18	14.96	1088.75	1053.807	181.14	<b>0.04</b>	0.45
		Ins 2	992.87	4.45	991.87	8.15	991.87	15.23	988.56	935.11	165.86	<b>0.10</b>	0.43
		Ins 3	1093.42	2.58	1091.47	8.74	1091.47	16.61	1063.17	1024.20	143.05	<b>0.18</b>	2.77
		Ins 4	1097.45	3.07	1095.12	8.47	1095.12	16.61	1095.53	1025.60	298.60	0.21	<b>0.17</b>
		Ins 5	919.91	2.03	917.15	8.77	917.15	14.84	831.13	801.97	199.54	<b>0.30</b>	9.65
	20	Ins 1	939.95	5.39	939.10	14.99	939.10	27.48	893.75	833.62	512.33	<b>0.09</b>	4.91
		Ins 2	1021.76	4.34	1019.34	14.44	1019.34	27.13	991.02	928.04	735.01	<b>0.24</b>	3.01
		Ins 3	787.26	12.38	785.54	14.62	785.54	27.20	767.01	719.09	632.57	<b>0.22</b>	2.57
		Ins 4	1016.02	6.57	1008.82	14.37	1008.82	27.72	951.80	855.66	503.95	<b>0.71</b>	6.30
		Ins 5	987.55	4.32	986.12	13.98	986.12	27.00	938.02	892.4	621.38	<b>0.14</b>	5.01
30	10	Ins 1	1180.26	28.31	1176.60	9.31	1176.60	32.17	1179.43	1104.65	269.78	0.31	<b>0.07</b>
		Ins 2	989.43	9.32	984.34	11.65	984.34	31.59	956.01	872.07	231.47	<b>0.52</b>	3.38
		Ins 3	1223.56	397.63	1217.51	13.26	1217.51	38.60	1159.23	1083.48	295.74	<b>0.49</b>	5.26
		Ins 4	1268.56	5.05	1265.75	12.56	1265.75	34.32	1267.92	1121.76	244.98	0.22	<b>0.05</b>
		Ins 5	1169.28	16.57	1165.12	12.45	1165.12	35.01	1144.78	1087.93	345.44	<b>0.36</b>	2.09
	20	Ins 1	989.51	126.21	988.11	22.41	988.62	78.90	946.78	849.18	554.72	<b>0.14</b>	4.32
		Ins 2	1234.36	22.01	1232.45	19.85	1233.41	73.47	1128.82	1056.70	695.21	<b>0.15</b>	8.55
		Ins 3	1164.84	26.32	1161.46	20.61	1161.46	74.77	1116.34	977.96	544.26	<b>0.29</b>	4.16
		Ins 4	1296.13	18.80	1293.50	19.96	1293.50	71.07	1233.04	1085.60	622.84	<b>0.21</b>	4.87
		Ins 5	1176.14	38.14	1174.77	23.65	1174.77	70.28	1118.02	1006.47	604.97	<b>0.12</b>	4.94
Average				25.82	1002.40	11.84	1002.45	28.94		393.79	0.21	3.80	

Note: Index Math-L, Math-WL, C, and ImpALNS denote the obtained results by our the proposed two-phase matheuristic with learning, the two-phase matheuristic without learning, the CPLEX, and the improved ALNS, respectively.

**Table 11.** Computation results for medium-size instances.

$I$	$S$	Instance	CPLEX solver (C)		Matheuristic algorithm with learning		Matheuristic algorithm without learning		Improved ALNS algorithm			$Gap_C - Math$ (%)	$Gap_C - ImpALNS$ (%)
			$Obj_C$	$Time_C$ (s)	$Obj_{Math-L}$	$Time_{Math-L}$ (s)	$Obj_{Math-WL}$	$Time_{Math-WL}$ (s)	$Obj_{ImpALNS}^B$	$Obj_{ImpALNS}^M$	$Time_{ImpALNS}$ (s)		
35	10	Ins 1	1494.66	9.49	1492.03	23.00	1492.03	97.43	1433.89	1350.75	361.25	<b>0.31</b>	4.07
		Ins 2	1416.75	28.90	1414.55	23.85	1414.55	92.67	1394.96	1292.80	325.67	<b>0.15</b>	1.54
		Ins 3	1513.78	15.50	1503.45	19.83	1503.45	84.72	1470.19	1340.16	357.71	<b>0.68</b>	2.88
		Ins 4	1425.38	26.46	1413.90	22.05	1413.90	80.56	1308.49	1191.95	299.57	<b>0.80</b>	8.20
		Ins 5	1343.78	33.39	1337.29	20.84	1337.29	71.50	1322.38	1222.97	360.33	<b>0.48</b>	1.59
	20	Ins 1	1446.62	10,000	1439.45	38.51	1439.60	211.26	1378.74	1318.16	786.19	<b>0.50</b>	4.69
		Ins 2	1414.80	10,000	1409.38	40.80	1409.38	179.24	1365.51	1249.69	752.47	<b>0.38</b>	3.48
		Ins 3	1441.68	10,000	1435.27	1435.27	1435.27	44.48	1379.50	1317.08	810.75	<b>0.44</b>	4.31
		Ins 4	1456.16	6227.47	1449.80	91.49	1449.80	166.20	1375.84	1268.82	773.25	<b>0.44</b>	5.51
		Ins 5	1684.78	3621.81	1680.91	43.22	1680.91	161.14	1562.12	1456.47	674.17	<b>0.23</b>	7.28
40	10	Ins 1	1591.49	3193.14	1587.04	45.55	1587.28	169.55	1517.58	1367.47	282.85	<b>0.28</b>	4.64
		Ins 2	1715.70	852.61	1708.90	31.06	1708.90	173.10	1668.19	1564.42	346.84	<b>0.40</b>	2.80
		Ins 3	1869.63	377.96	1860.93	28.56	1861.47	161.52	1790.19	1645.08	374.13	<b>0.47</b>	4.25
		Ins 4	1676.63	984.77	1664.63	34.88	1665.04	134.94	1593.82	1451.48	280.64	<b>0.72</b>	4.94
		Ins 5	1752.61	1982.36	1743.72	37.08	1743.72	194.69	1700.60	1538.84	341.78	<b>0.51</b>	2.97
	20	Ins 1	1735.13	10,000	1725.61	60.38	1725.61	336.97	1671.81	1559.28	933.83	<b>0.55</b>	3.65
		Ins 2	1537.48	10,000	1529.32	69.32	1529.33	289.84	1463.40	1348.73	819.51	<b>0.53</b>	4.82
		Ins 3	1372.11	10,000	1365.36	69.33	1365.36	337.61	1350.62	1274.77	828.16	<b>0.49</b>	1.57
		Ins 4	1646.63	10,000	1637.36	66.14	1637.50	336.91	1576.99	1416.88	685.19	<b>0.56</b>	4.23
		Ins 5	1522.46	10,000	1512.05	57.23	1512.05	337.07	1434.52	1398.77	903.51	<b>0.68</b>	5.78
45	10	Ins 1	2064.45	612.59	2054.67	94.48	2054.67	404.51	1977.31	1790.59	401.46	<b>0.47</b>	4.22
		Ins 2	1827.67	10,000	1813.17	64.49	1813.17	351.34	1722.47	1617.42	473.04	<b>0.79</b>	5.76
		Ins 3	2047.97	2183.16	2034.55	122.50	2034.55	403.96	1869.25	1687.61	363.22	<b>0.66</b>	8.73
		Ins 4	1982.96	10,000	1969.84	72.53	1970.55	413.341	1808.70	1693.60	331.47	<b>0.66</b>	8.79
		Ins 5	1742.16	10,000	1727.03	82.81	1727.03	312.32	1726.10	1593.93	741.22	<b>0.87</b>	0.92
	20	Ins 1	1782.69	10,000	1766.63	119.21	1766.63	599.99	1751.1	1612	829.64	<b>0.90</b>	3.79
		Ins 2	1858.43	10,000	1845.69	143.28	1845.69	595.21	1798.80	1690.59	913.46	<b>0.69</b>	3.21
		Ins 3	1821.66	10,000	1808.07	141.68	1808.07	701.25	1759.05	1674	719.39	<b>0.75</b>	3.44
		Ins 4	1824.03	10,000	1813.48	157.51	1813.87	601.74	1688.38	1615.54	438.01	<b>0.58</b>	7.43
		Ins 5	1712.05	10,000	1701.64	143.83	1701.64	574.26	1668.21	1560.20	1000.15	<b>0.61</b>	2.56
Average				1648.19	67.00	1648.28	292.44			583.63	0.55	4.40	

Note: Index Math-L, Math-WL, C, and ImpALNS denote the obtained results by the proposed two-phase matheuristic with learning, the two-phase matheuristic without learning, the CPLEX, and the improved ALNS, respectively.

learning mechanism has a better performance comparing to the improved ALNS algorithm. The average gap of the obtained solutions by the improved ALNS algorithm is 31 times greater than the average gap of achieved solutions from the two-phase matheuristic algorithm with the learning mechanism. The computational time of the two-phase matheuristic algorithm with the learning mechanism is significantly less than the computational time of the improved ALNS. The efficiency of the proposed two-phase matheuristic algorithm for medium-size instances (up to 45 assets under the different number of scenarios) is examined and the results are reported in Table 11.

Based on the results reported in Table 11, the CPLEX solver can find optimal solutions for 14 instances in less than 10,000 s, while the proposed two-phase matheuristic algorithm with the learning mechanism could find good quality solutions with 0.55% gap on average. The average computational time of the CPLEX solver for the instances that optimal solutions are found is 1997.60 s, while the average computational time of the proposed two-phase matheuristic algorithm with the learning procedure is 67 s. Also, the solutions achieved by the improved ALNS have 4.40% gap on average, therefore, the proposed two-phase matheuristic algorithm with the learning mechanism has a better performance comparing

to the improved ALNS algorithm. Therefore, it can be concluded that the proposed two-phase matheuristic algorithm with learning mechanism is efficient for solving medium-size instances. Also, the results show that the quality of the proposed two-phase matheuristic with and without learning mechanism is equal but the average computational time of the two-phase matheuristic algorithm without the learning mechanism is 4 times more compared to the two-phase matheuristic algorithm with the learning mechanism. It is concluded that using the learning mechanism will reduce computational time without changing the quality of the obtained solutions.

The computational time corresponding to generating the subsets is increased by increasing the number of scenarios. Therefore, in the two-phase matheuristic algorithm, the critical assets in each iteration are partitioned in a predefined number of clusters and the subsets are generated for each cluster in large-size instances. Then combinations of the feasible routes constructed based on the subsets of different clusters are examined to generate other routes. The efficiency of the two-phase matheuristic algorithm for large-size instances (up to 80 assets) are examined and results are reported in Table 12. As reported in this table, the CPLEX solver could not find the optimal solution for any of the instances during a restricted 10,000 s time. While the average computational

**Table 12.** Computation results for large-size instances by considering five vehicles and three main depots.

<i>I</i>	Instance	Commercial solver (C)		Matheuristic algorithm		Improved ALNS algorithm			<i>Gap<sub>C-Math</sub></i> (%)	<i>Gap<sub>C-ALNS</sub></i> (%)
		<i>Obj<sub>C</sub></i>	<i>Time<sub>C</sub></i> (s)	<i>Obj<sub>Math</sub></i>	<i>Time<sub>Math</sub></i> (s)	<i>Obj<sub>ALNS</sub><sup>B</sup></i>	<i>Obj<sub>ALNS</sub><sup>M</sup></i>	<i>Time<sub>ALNS</sub></i> (s)		
50	Ins 1	2026.22	10,000	2010.88	224.72	2014.91	1940.87	1297.42	0.76	<b>0.56</b>
	Ins 2	2229.33	10,000	2207.20	202.93	2145.43	2079.26	1089.94	0.99	3.76
	Ins 3	2132.32	10,000	2116.37	207.16	2059.88	1974.12	1066.61	0.75	3.40
	Ins 4	2110.90	10,000	2097.71	187.64	2029.01	1961.15	1040.16	0.62	3.88
	Ins 5	2209.13	10,000	2187.48	208.94	2137.88	2017.80	1057.89	0.98	3.22
55	Ins 1	3939.04	10,000	3899.59	519.28	3605.30	3450.94	1436.41	1.00	8.47
	Ins 2	4349.53	10,000	4313.26	666.08	4104.31	3942.29	1187.00	0.83	5.63
	Ins 3	4489.85	10,000	4484.54	650.96	4171.01	4106.54	1566.60	0.12	7.10
	Ins 4	4295.79	10,000	4249.92	711.96	4097.87	3944.83	1658.91	1.10	4.61
	Ins 5	4377.38	10,000	4317.70	497.99	4035.18	3685.36	1263.59	1.36	7.82
60	Ins 1	4631.76	10,000	4609.88	777.95	4394.10	4230.23	1497.81	0.47	5.13
	Ins 2	4333.72	10,000	4309.75	963.33	4040.76	3806.45	1611.44	0.55	6.76
	Ins 3	4854.59	10,000	4827.89	917.35	4577.96	4407.47	1480.39	0.55	5.70
	Ins 4	4862.07	10,000	4828.00	778.85	4636.70	4421.41	1161.03	0.70	4.63
	Ins 5	4788.64	10,000	4760.44	947.24	4482.47	4374.86	1555.80	0.59	6.39
70	Ins 1	9770.81	10,000	9746.91	2116.24	9249.84	9033.90	1124.17	0.24	5.33
	Ins 2	5303.27	10,000	5193.300	2006.82	5052.44	4886.96	1466.98	2.07	4.73
	Ins 3	5479.99	10,000	5440.11	2642.86	5258.95	5130.30	1631.55	0.73	4.03
	Ins 4	5230.72	10,000	5137.29	2470.01	5066.71	4814.50	1308.56	1.79	3.13
	Ins 5	5438.43	10,000	5387.11	2065.18	5187.50	5005.29	1477.25	0.94	4.61
80	Ins 1	4352.89	10,000	4290.42	2006.96	4167.99	4028.79	1548.86	1.43	4.25
	Ins 2	4481.92	10,000	4414.20	2957.60	4263.80	4140.55	1957.41	1.51	4.87
	Ins 3	4643.56	10,000	4555.69	3168.65	4518.9	4348.3	1624.20	1.89	2.70
	Ins 4	4415.06	10,000	4339.83	4051.81	4232.09	4120.33	1883.22	1.70	4.14
	Ins 5	4529.62	10,000	4442.24	5100.58	4325.95	4230.71	1439.66	1.92	4.50
Average				1481.96				1417.31	1.02	4.78

Note: Index Math-L, C, and ImpALNS denote the obtained results by our proposed two-phase matheuristic with learning, the CPLEX, and the improved ALNS, respectively.

time of the proposed two-phase matheuristic algorithm with the learning mechanism is 1481.96 s. Also, the average gap of 1.02% confirms a good quality obtained solutions by the two-phase matheuristic algorithm. The efficiency of the proposed algorithm is examined by comparing it with the improved ALNS. The average computational time of the improved ALNS (1417.31 s) is less than the average computational time of the proposed two-phase matheuristic algorithm with the learning mechanism but the quality of the obtained solutions by the improved ALNS is less than the solutions achieved by the two-phase matheuristic algorithm. On the other hand, the average gap of obtained solutions from the improved ALNS is almost 4 times greater than the average gap of the obtained solutions by the two-phase matheuristic algorithm. Therefore, the proposed two-phase matheuristic algorithm with the learning mechanism has much better performance in large-size instances comparing with the improved ALNS algorithm.

### 6.3.2. Examining the efficiency of the improved ALNS comparing to the classic ALNS algorithm

In this section efficiency of the improved ALNS algorithm is compared with the classic ALNS for small-size and

medium-size instances. In the classic ALNS, all decisions are considered in the solution representation and the mathematical models Loc-Int and Imp-Int are not utilised. Each algorithm runs 10 times and the best obtained objective values for the improved ALNS ( $Obj_{ImpALNS}^B$ ) and the classic ALNS ( $Obj_{ALNS}^B$ ) are reported in Tables 13 and 14. Also, the average objective values for the improved ALNS and the classic ALNS ( $Obj_{ImpALNS}^M$  and  $Obj_{ALNS}^M$ , respectively) and the average computational time of the algorithms are reported in these tables. The gap between the best objective values obtained by these algorithms is calculated by  $Gap_{ImpALNS-ALNS} = (Obj_{ImpALNS}^B - Obj_{ALNS}^B) / Obj_{ImpALNS}^B \times 100\%$ .

The obtained results in Table 13 show that the average computational time of the classic ALNS is almost 12 times less than the average computational time of the improved ALNS. The gap between the best objective values by these algorithms is 22.10% on average. Therefore, the improved ALNS has a better efficiency compared with the classic ALNS for small-size instances. Also, the obtained results in Table 14 show that the achieved solutions by improved ALNS algorithm have better quality compared with the classic ALNS for medium-size

**Table 13.** Computation results for small-size instances.

$I$	$S$	Instance	Improved ALNS algorithm			Classic ALNS algorithm			$Gap_{ImpALNS-ALNS} (\%)$	
			$Obj_{ImpALNS}^B$	$Obj_{ImpALNS}^M$	$Time_{ImpALNS} (s)$	$Obj_{ALNS}^B$	$Obj_{ALNS}^M$	$Time_{ALNS} (s)$		
20	10	Ins 1	943.65	851.42	173.28	820.16	759.103	28.373	13.09	
		Ins 2	993.59	932.32	205.94	873.21	833.54	28.42	12.12	
		Ins 3	815.6	755.29	167.146	640.32	623.09	29.09	21.49	
		Ins 4	952.45	883.09	140.77	867.28	823.31	29.136	8.94	
		Ins 5	899.74	843.04	140.96	853.19	811.57	29.336	5.17	
	20	Ins 1	834.26	786.10	470.27	587.18	529.60	34.15	29.62	
		Ins 2	625.87	557.66	532.86	488.76	435.65	34.17	21.91	
		Ins 3	662.64	579.33	369.27	443.90	403.38	34.22	37.86	
		Ins 4	694.13	637.67	434.60	505.61	449.70	34.47	27.16	
		Ins 5	744.48	669.44	497.03	507.02	463.44	34.59	31.90	
	25	10	Ins 1	1088.75	1053.807	181.14	889.07	813.40	31.21	18.3
			Ins 2	988.56	935.11	165.86	827.64	759.63	31.32	16.28
			Ins 3	1063.17	1024.20	143.05	962.89	917.91	31.08	9.43
			Ins 4	1095.53	1025.60	298.60	893.97	853.77	30.78	18.40
			Ins 5	831.13	801.97	199.54	814.56	780.51	30.84	1.99
20		Ins 1	893.75	833.62	512.33	654.62	598.31	34.15	26.76	
		Ins 2	991.02	928.04	735.01	712.11	682.07	33.48	28.14	
		Ins 3	767.01	719.09	632.57	551.70	473.84	33.76	28.07	
		Ins 4	951.80	855.66	503.95	713.77	645.25	34.78	25.01	
		Ins 5	938.02	892.4	621.38	717.83	678.69	34.10	23.47	
30		10	Ins1	1179.43	1104.65	269.78	1034.70	988.33	33.30	12.27
			Ins 2	956.01	872.07	231.47	832.35	784.16	33.32	12.93
			Ins 3	1159.23	1083.48	295.74	1020.40	984.80	32.67	11.98
			Ins 4	1267.92	1121.76	244.98	1046.95	973.04	33.22	17.43
			Ins 5	1144.78	1087.93	345.44	1007.29	945.13	33.06	12.01
	20	Ins 1	946.78	849.18	554.72	650.10	580.15	40.37	31.33	
		Ins 2	1128.82	1056.70	695.21	835.94	779.66	41.11	25.95	
		Ins 3	1116.34	977.96	544.26	732.99	691.24	41.01	34.34	
		Ins 4	1233.04	1085.60	622.84	800.47	762.78	39.76	35.08	
		Ins 5	1118.02	1006.47	604.97	709.73	686.91	41.03	36.52	
	Average				393.79			33.68	22.10	

Note: Index ImpALNS and ALNS denote the obtained results by the improved ALNS and the classic ALNS, respectively.

**Table 14.** Computation results for medium-size instances.

<i>I</i>	<i>S</i>	Instance	Improved ALNS algorithm			classic ALNS algorithm			<i>Gap<sub>ImpALNS-ALNS</sub></i> (%)
			<i>Obj<sub>ImpALNS</sub><sup>B</sup></i>	<i>Obj<sub>ImpALNS</sub><sup>M</sup></i>	<i>Time<sub>ImpALNS</sub></i> (s)	<i>Obj<sub>ALNS</sub><sup>B</sup></i>	<i>Obj<sub>ALNS</sub><sup>M</sup></i>	<i>Time<sub>ALNS</sub></i> (s)	
35	10	Ins 1	1433.89	1350.75	361.25	1225.15	1149.36	33.51	14.56
		Ins 2	1394.96	1292.80	325.67	1084.15	1041.02	33.22	22.28
		Ins 3	1470.19	1340.16	357.71	1182.58	1036.34	33.56	19.56
		Ins 4	1308.49	1191.95	299.57	1104.55	1061.71	34.43	15.59
		Ins 5	1322.38	1222.97	360.33	1051.70	1019.50	34.16	20.47
	20	Ins 1	1378.74	1318.16	786.19	1014.74	935.18	39.32	26.40
		Ins 2	1365.51	1249.69	752.47	945.14	879.75	40.18	30.84
		Ins 3	1379.50	1317.08	810.75	934.11	883.83	39.60	32.29
		Ins 4	1375.84	1268.82	773.25	1013.12	956.95	40.95	26.36
		Ins 5	1562.12	1456.47	674.17	1088.46	1014.54	41.01	30.32
40	10	Ins 1	1517.58	1367.47	282.85	1271.09	1190.51	36.58	16.24
		Ins 2	1668.19	1564.42	346.84	1393.05	1332.48	37.22	16.49
		Ins 3	1790.19	1645.08	374.13	1316.12	1271.17	37.06	26.48
		Ins 4	1593.82	1451.48	280.64	1358.74	1291.16	37.29	14.75
		Ins 5	1700.60	1538.84	341.78	1357.28	1303.67	36.76	20.19
	20	Ins 1	1671.81	1559.28	933.83	1205.85	1143.47	42.17	27.87
		Ins 2	1463.40	1348.73	819.51	1075.54	987.93	43.07	26.50
		Ins 3	1350.62	1274.77	828.16	932.79	851.53	42.93	30.94
		Ins 4	1576.99	1416.88	685.19	1107.70	1048.81	43.02	29.76
		Ins 5	1434.52	1398.77	903.51	1081.18	998.04	43.31	24.63
45	10	Ins 1	1977.31	1790.59	401.46	1719.22	1589.24	38.80	13.05
		Ins 2	1722.47	1617.42	473.04	1456.49	1419.00	37.27	15.44
		Ins 3	1869.25	1687.61	363.22	1639.95	1576.55	38.48	12.27
		Ins 4	1808.70	1693.60	331.47	1476.25	1422.40	38.44	18.38
		Ins 5	1726.10	1593.93	741.22	1307.33	1245.45	38.34	24.26
	20	Ins 1	1751.1	1612	829.64	1242.72	1213.56	46.46	29.03
		Ins 2	1798.80	1690.59	913.46	1323.48	1220.74	48.02	26.42
		Ins 3	1759.05	1674.00	719.39	1295.19	1228.75	48.27	26.37
		Ins 4	1688.38	1615.54	438.01	1302.66	1181.23	49.02	22.85
		Ins 5	1668.21	1560.20	1000.15	1220.98	1138.36	48.51	26.81
Average					583.63		40.03	22.91	

Note: Index ImpALNS and ALNS denote the obtained results by the improved ALNS and the classic ALNS, respectively.

instances. The improved ALNS found better solutions compared with the classic ALNS (22.91% on average).

#### 6.4. Examining the efficiency of the two-phase matheuristic by increasing numbers of protection centres and vehicles

In this section, the effect of an increase in the number of protection centres and vehicles on the efficiency of the two-phase matheuristic algorithm is examined by considering 50, 55, and 60 assets. Also, it is assumed that the number of protection centres and vehicles are 8 and 8, respectively. The objective function values and computational times of the matheuristic algorithm for different instances are examined by the best objective value obtained by the CPLEX solver in 10,000 s. In nine instances, the algorithm outperforms the commercial solver in less computational time. Also, the average gap between the best objective value obtained by the CPLEX solver and the objective value of the two-phase matheuristic algorithm is  $-0.19\%$ . It is concluded that the two-phase matheuristic algorithm can achieve good quality solutions in a reasonable time (Table 15).

**Table 15.** Examining the efficiency of the matheuristic algorithm by considering eight protection centres and eight vehicles.

<i>I</i>	Instance	Commercial solver (C)		Matheuristic algorithm		<i>Gap<sub>C-Math</sub></i> (%)
		<i>Obj<sub>C</sub></i>	<i>Time<sub>C</sub></i> (s)	<i>Obj<sub>Math</sub></i>	<i>Time<sub>Math</sub></i> (s)	
50	Ins 1	4264.08	10,000	<b>4274.77</b>	1335.12	$-0.25$
	Ins 2	3937.33	10,000	<b>3953.72</b>	1274.90	$-0.42$
	Ins 3	3920.02	10,000	<b>3920.54</b>	809.17	$-0.01$
	Ins 4	3989.09	10,000	<b>3990.84</b>	1311.04	$-0.04$
	Ins 5	3948.21	10,000	3928.04	1277.36	0.51
55	Ins 1	4654.26	10,000	4639.77	1444.77	0.31
	Ins 2	4389.09	10,000	4362.16	1459.79	0.61
	Ins 3	4390.13	10,000	<b>4452.00</b>	1507.91	$-1.41$
	Ins 4	4600.14	10,000	4582.70	1467.39	0.38
	Ins 5	4726.56	10,000	<b>4743.67</b>	1513.96	$-0.36$
60	Ins 1	4415.90	10,000	4396.39	1679.19	0.44
	Ins 2	4518.84	10,000	<b>4541.84</b>	1618.81	$-0.51$
	Ins 3	4871.15	10,000	4870.71	1760.21	0.01
	Ins 4	4295.30	10,000	<b>4336.76</b>	1764.11	$-0.96$
	Ins 5	4241.48	10,000	<b>4288.14</b>	1610.61	$-1.10$
Average					1347.70	$-0.19$

#### 6.5. Evaluating the effects of the diversification procedure in the two-phase matheuristic algorithm

In the two-phase matheuristic algorithm, a diversification approach is used in each iteration. In this section, the effect of applying a diversification mechanism on

the solution quality is examined for various instances by considering 50 assets, 5 vehicles, 5 protection centres, and 20 scenarios. The obtained objective values from the two-phase matheuristic algorithm with and without considering diversification approach ( $Obj_{div}$  and  $Obj_{wdiv}$ , respectively) under different values of  $\epsilon_2$  are reported in Table 16. Based on the reported results in this table,

**Table 16.** Examining the effect of considering diversification mechanism on the quality of the obtained solution by the two-phase matheuristic algorithm.

Ins	$\epsilon_2$	$Obj_{div}$	$Obj_{wdiv}$	Ins	$\epsilon_2$	$Obj_{div}$	$Obj_{wdiv}$
1	100	1959.39	1959.39	4	100	2053.24	1953.51
	150	1963.68	1625.34		150	2048.10	1748.67
	200	1970.84	1413.14		200	2052.32	1489.84
	250	2000.29	1166.43		250	2086.66	1342.74
	300	2010.88	1033.77		300	2097.36	1142.24
	350	1991.19	783.93		350	2076.99	848.80
2	100	2164.57	2070.98	5	100	2136.03	2034.89
	150	2135.67	1919.11		150	2123.76	1855.79
	200	2166.44	1686.29		200	2124.18	1671.25
	250	2179.06	1401.01		250	2185.08	1449.36
	300	2186.92	1171.26		300	2156.08	1261.69
	350	2190.72	1030.65		350	2170.27	1051.45
3	100	2090.61	1936.49				
	150	2062.73	1779.13				
	200	2070.67	1565.93				
	250	2112.45	1338.40				
	300	2112.45	1186.35				
	350	2114.51	936.05				

**Table 17.** Computation results for real case by considering eight vehicles and three main protection centres.

S	Commercial solver (C)		Matheuristic algorithm		$Gap_{C-Math}$ (%)
	$Obj_C$	$Time_C$ (s)	$Obj_{Math}$	$Time_{Math}$ (s)	
5	123,016.49	10,000	123,015.92	132.51	0.00
10	132,881.42	10,000	131,761.80	157.94	0.84
20	134,659.41	10,000	134,099.36	505.17	0.41
30	136,949.30	10,000	136,202.77	1066.23	0.54
40	137,565.62	10,000	137,937.45	1079.88	-0.27
50	128,170.28	10,000	128,209.82	1092.35	-0.03
Average				672.35	0.25

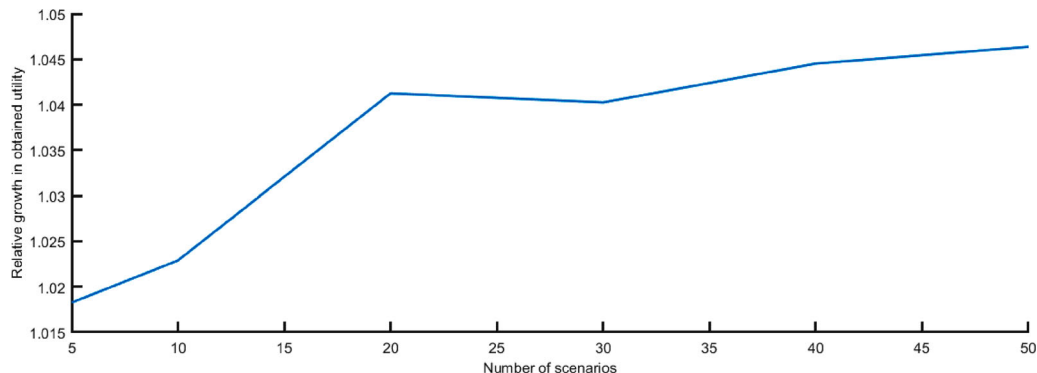
Note: Index Math-L and C denote the obtained results by our proposed matheuristic with learning and the CPLEX, respectively.

considering diversification mechanism has a significant impact on the quality of the obtained solutions by the two-phase matheuristic algorithm especially in the case of using a large value for  $\epsilon_2$ .

## 6.6. Real case application

Hobart, Tasmania (Australia) is one of the most fire-hazardous areas in the world. In this area, many activities such as fuel treatment are implemented to reduce the risk of damage by wildfire. Therefore, the efficiency of the proposed two-phase matheuristic algorithm and the effects of considering collaboration between protection centres are examined based on a real case application in Hobart. For this reason, 20 assets are selected in Hobart and their pair distances are calculated according to their latitudes and longitudes. 5 random locations are considered for protection centres and it is assumed that 8 vehicles are available. The scenarios are generated based on the method proposed in Section 4 by using real data about wind direction, month's rainfall, and month's maximum temperature in Hobart, Tasmania (Australia). The best objective function values achieved by the CPLEX in 10,000 s and the objective values and computational times of the two-phase matheuristic algorithm are reported in Table 17. In these cases, a time limit equal to 1000 s is considered for solving the route-based version of SAPP-PC in the second phase of the two-phase matheuristic algorithm. The results show that the proposed matheuristic algorithm has a satisfactory performance for real data as well. The CPLEX solver can not find the optimal objective function values in any cases while the proposed two-phase matheuristic algorithm has found good quality solutions (0.25% gap on average) in a reasonable time (672.35 s in average).

The effect of considering collaboration between protection centres in the obtained utility profit is illustrated in Figure 9. In this figure, the RGU (relative



**Figure 9.** Relative growth in the saved assets (obtained utility) under different number of scenarios.

growth in obtained utility that is calculated by  $RGU = EV_{WDC}/EV_{WODC}$  is shown for the real case under a different number of scenarios. As shown in Figure 9, considering collaboration between protection centres has a significant effect to increase the obtained utility by protecting assets.

## 7. Conclusion

Protecting assets is one possible strategy during a wildfire. Therefore, proper planning of asset protection before an actual wildfire may result in efficient actions during the wildfire. Since there are some sources of uncertainty related to the asset protection problem, it is considered as two-stage stochastic programming. The more uncertain parameters there are, the more possible scenarios there are. However, there is no full data for the scenarios, so an artificial neural network was employed to extract the required data for all the scenarios, considering previous wildfire data. This made our research very close to real-world situations.

The results of our numerical studies show that the proposed approach can reduce the costs of damage by considering collaboration between protection centres. In real case applications, the protection centres have limited capacity; therefore, they do not have enough capacity for different types of vehicles and equipment for asset protection problems. So, considering collaboration between protection centres in a disaster situation is critical to reducing the costs of damage. Also, results obtained based on the real case show that the utility obtained from protecting assets by considering collaboration between protection centres is increased for increases with a large number of scenarios.

Two solution approaches, a two-phase matheuristic and an improved ALNS, were developed to solve the problem efficiently. The efficiency of the proposed algorithms is examined for instances with different sizes. The results show that the two-phase matheuristic algorithm performs well for small-size instances and can find near-optimal solutions (0.21% gap on average). Also, the computational time of the two-phase matheuristic algorithm is 2 times less than the computation time of the CPLEX solver for small-size instances. The CPLEX solver can find optimal solutions in 10,000 s for 14 medium-size instances from 30 instances, and the average computational time for these cases is 1997.60 s, while the average gap and computational time for the two-phase matheuristic algorithm are 0.55% and 67.00 s, respectively. The CPLEX solver cannot find optimal solutions within 10,000 s in any large-size instances. But the two-phase matheuristic algorithm can find good quality solutions in 1481.96 s on average. Therefore, the

numerical analysis confirms that the proposed two-phase matheuristic solution method is efficient enough regarding quality and computational time.

The results of a numerical study confirm that the proposed solution algorithm is much more efficient when the algorithm is equipped with a learning stage during the algorithm iterations. The average computational times of the two-phase matheuristic algorithm with the learning procedure for small-size and medium-size instances are 2 and 4 times, less, respectively than the average computational times of the two-phase matheuristic algorithm without the learning procedure. Moreover, the performance of the proposed two-phase matheuristic algorithm is compared with an improved version of the ALNS algorithm, and the results confirm the effectiveness of the proposed algorithm.

Since this study can also be used for the protection of production centres during wildfires, it can be mentioned as a managerial insight for wildfire management teams. Therefore, they can make protection activities more collaborative between protection centres, which may not only enhance protection effectiveness but also decrease total asset protection costs. So, it can be suggested that production centres be allocated and protected by protection centres with collaboration potential. In this study, the fire spread pattern was not determined precisely. Future studies could predict fire spread on the cells by machine learning equipped with image processing before, during, and after a wildfire in the asset protection problem. Another direction for future research could be improving the first phase of the matheuristic algorithm to extract feasible routes more efficiently.

## Disclosure statement

No potential conflict of interest was reported by the author(s).

## Data availability statement

The data that support the findings of this study are openly available in GitHub at <https://github.com/bashirimahdi/SAPP-PC>.

## Notes on contributors



*Erfaneh Nikzad* received the B.Sc. degree in industrial engineering from Alzahra University, in 2011, the M.Sc. degree in industrial engineering from Shahed University, Tehran, in 2014, and the Ph.D. in industrial engineering from Shahed University, Tehran, Iran, in 2020. She currently works as a lecturer in West Tehran Branch, Islamic Azad University, Tehran, Iran. Her research

interests include vehicle routing problems, matheuristic algorithms, stochastic programming, and machine learning.



**Mahdi Bashiri** has over 17 years of academic experience with a particular interest in operations and supply chain management, operations research, transportation planning, Heuristic and Matheuristic algorithms. Also, he has participated in industrial and business projects in different levels. He has been involved in two UKRI funded projects. He has supervised more than 5 Ph.D. students and more than 60 M.Sc. students. He is an active reviewer for reputable academic journals. He is a recipient of the 2013 young national top scientist award from the Academy of Sciences of the Islamic Republic of Iran.

## References

- Abbasi, Babak, Toktam Babaei, Zahra Hosseinifard, Kate Smith-Miles, and Maryam Dehghani. 2020. "Predicting Solutions of Large-Scale Optimization Problems via Machine Learning: A Case Study in Blood Supply Chain Management." *Computers & Operations Research* 119: 104941.
- Amarouche, Youcef, Rym Nesrine Guibadj, Elhadja Chaalal, and Aziz Moukrim. 2020. "Effective Neighborhood Search with Optimal Splitting and Adaptive Memory for the Team Orienteering Problem with Time Windows." *Computers & Operations Research* 123: 105039.
- Bai, Ruibin, Jacek Blazewicz, Edmund K. Burke, Graham Kendall, and Barry McCollum. 2012. "A Simulated Annealing Hyper-heuristic Methodology for Flexible Decision Support." *4OR* 10 (1): 43–66.
- Barteczko-Hibbert, Christian, Mark Gillott, and Graham Kendall. 2009. "An Artificial Neural Network for Predicting Domestic Hot Water Characteristics." *International Journal of Low-Carbon Technologies* 4 (2): 112–119.
- Bashiri, Mahdi, Erfaneh Nikzad, Andrew Eberhard, John Hearne, and Fabricio Oliveira. 2021. "A Two Stage Stochastic Programming for Asset Protection Routing and a Solution Algorithm Based on the Progressive Hedging Algorithm." *Omega* 104: 102480.
- Bayliss, Christopher, Angel A. Juan, Christine S. M. Currie, and Javier Panadero. 2020. "A Learnheuristic Approach for the Team Orienteering Problem with Aerial Drone Motion Constraints." *Applied Soft Computing* 92: 106280.
- Bengio, Yoshua, Andrea Lodi, and Antoine Prouvost. 2020. "Machine Learning for Combinatorial Optimization: A Methodological Tour D'horizon." *European Journal of Operational Research* 290 (2): 405–421.
- Bertazzi, Luca, Leandro C. Coelho, Annarita De Maio, and Demetrio Laganà. 2019. "A Matheuristic Algorithm for the Multi-Depot Inventory Routing Problem." *Transportation Research Part E: Logistics and Transportation Review* 122: 524–544.
- Bian, Zheyong, and Xiang Liu. 2018. "A Real-Time Adjustment Strategy for the Operational Level Stochastic Orienteering Problem: A Simulation-Aided Optimization Approach." *Transportation Research Part E: Logistics and Transportation Review* 115: 246–266.
- Bishop, Christopher M., 1995. *Neural Networks for Pattern Recognition*. Oxford: Oxford University Press.
- Burke, Edmund K., Graham Kendall, and Eric Soubeiga. 2003. "A Tabu-Search Hyperheuristic for Timetabling and Rostering." *Journal of Heuristics* 9 (6): 451–470.
- Cao, ZhengCai, ChengRan Lin, MengChu Zhou, and Ran Huang. 2018. "Scheduling Semiconductor Testing Facility by Using Cuckoo Search Algorithm with Reinforcement Learning and Surrogate Modeling." *IEEE Transactions on Automation Science and Engineering* 16 (2): 825–837.
- Chen, Bing, Ruibin Bai, Jiawei Li, Yueni Liu, Ning Xue, and Jianfeng Ren. 2020. "A Multiobjective Single Bus Corridor Scheduling Using Machine Learning-Based Predictive Models." *International Journal of Production Research*: 1–16. doi:10.1080/00207543.2020.1766716.
- Chen, Shunqin, Zhengfeng Guo, and Xinlei Zhao. 2020. "Predicting Mortgage Early Delinquency with Machine Learning Methods." *European Journal of Operational Research*. 290 (1): 358–372.
- Chen, Binhui, Rong Qu, Ruibin Bai, and Wasakorn Laesanklang. 2020. "A Variable Neighborhood Search Algorithm with Reinforcement Learning for a Real-Life Periodic Vehicle Routing Problem with Time Windows and Open Routes." *RAIRO - Operations Research* 54 (5): 1467–1494.
- Choong, Shin Siang, Li-Pei Wong, and Chee Peng Lim. 2018. "Automatic Design of Hyper-heuristic Based on Reinforcement Learning." *Information Sciences* 436–437: 89–107.
- Chopra, S., and M. S. Sodhi. 2004. "Managing Risk to Avoid Supply-Chain Breakdown." Fall ed.
- Chou, Xiaochen, Luca Maria Gambardella, and Roberto Montemanni. 2021. "A Tabu Search Algorithm for the Probabilistic Orienteering Problem." *Computers & Operations Research* 126: 105107.
- Cui, Hailong, Sampath Rajagopalan, and Amy R. Ward. 2020. "Predicting Product Return Volume Using Machine Learning Methods." *European Journal of Operational Research* 281 (3): 612–627.
- Dolinskaya, Irina, Zhenyu Edwin Shi, and Karen Smilowitz. 2018. "Adaptive Orienteering Problem with Stochastic Travel Times." *Transportation Research Part E: Logistics and Transportation Review* 109: 1–19.
- Evers, Lanah, Kristiaan Glorie, Suzanne van der Ster, Ana Isabel Barros, and Herman Monsuur. 2014. "A Two-Stage Approach to the Orienteering Problem with Stochastic Weights." *Computers & Operations Research* 43: 248–260.
- Green, Matthew. 2020. "Australia's Massive Fires Could Become Routine, Climate Scientists Warn." <https://www.reuters.com/article/us-climate-change-australia-report/australias-massive-fires-could-become-routine-climate-scientists-warn-idUSKBN1ZD06W>.
- Grenouilleau, Florian, Antoine Legrain, Nadia Lahrichi, and Louis-Martin Rousseau. 2019. "A Set Partitioning Heuristic for the Home Health Care Routing and Scheduling Problem." *European Journal of Operational Research* 275 (1): 295–303.
- Gunawan, Aldy, Hoong Chuin Lau, and Pieter Vansteenwegen. 2016. "Orienteering Problem: A Survey of Recent Variants, Solution Approaches and Applications." *European Journal of Operational Research* 255 (2): 315–332.
- Haidar, Ahmed M. A., M. W. Mustafa, Faisal A. F. Ibrahim, and Ibrahim A. Ahmed. 2011. "Transient Stability Evaluation of Electrical Power System Using Generalized Regression Neural Networks." *Applied Soft Computing* 11 (4): 3558–3570.



- Hammami, Farouk, Monia Rekik, and Leandro C. Coelho. 2020. "A Hybrid Adaptive Large Neighborhood Search Heuristic for the Team Orienteering Problem." *Computers & Operations Research* 123: 105034.
- Hanafi, Saïd, Renata Mansini, and Roberto Zanotti. 2020. "The Multi-Visit Team Orienteering Problem with Precedence Constraints." *European Journal of Operational Research* 282 (2): 515–529.
- Hapsari, Indri, Isti Surjandari, and K. Komarudin. 2019. "Solving Multi-Objective Team Orienteering Problem with Time Windows Using Adjustment Iterated Local Search." *Journal of Industrial Engineering International* 15 (4): 679–693.
- Hellsten, Erik Orm, David Sacramento, and David Pisinger. 2020. "An Adaptive Large Neighbourhood Search Heuristic for Routing and Scheduling Feeder Vessels in Multi-Terminal Ports." *European Journal of Operational Research* 287 (2): 682–698.
- Hemmelmayr, Vera C., Jean-François Cordeau, and Teodor Gabriel Crainic. 2012. "An Adaptive Large Neighborhood Search Heuristic for Two-Echelon Vehicle Routing Problems Arising in City Logistics." *Computers & Operations Research* 39 (12): 3215–3228.
- Ho, William, Tian Zheng, Hakan Yildiz, and Srinivas Talluri. 2015. "Supply Chain Risk Management: A Literature Review." *International Journal of Production Research* 53 (16): 5031–5069.
- Hu, Wanzhe, Mahdi Fathi, and Panos M. Pardalos. 2018. "A Multi-objective Evolutionary Algorithm Based on Decomposition and Constraint Programming for the Multi-Objective Team Orienteering Problem with Time Windows." *Applied Soft Computing* 73: 383–393.
- Hu, Qian, and Andrew Lim. 2014. "An Iterative Three-Component Heuristic for the Team Orienteering Problem with Time Windows." *European Journal of Operational Research* 232 (2): 276–286.
- Ieracitano, Cosimo, Annunziata Paviglianiti, Maurizio Campolo, Amir Hussain, Eros Pasero, and Francesco Carlo Morabito. 2020. "A Novel Automatic Classification System Based on Hybrid Unsupervised and Supervised Machine Learning for Electrospun Nanofibers." *IEEE/CAA Journal of Automatica Sinica* 8 (1): 64–76.
- Karabulut, Korhan, and M. Fatih Tasgetiren. 2020. "An Evolution Strategy Approach to the Team Orienteering Problem with Time Windows." *Computers & Industrial Engineering* 139: 106109.
- Keshtkaran, Morteza, Koorush Ziarati, Andrea Bettinelli, and Daniele Vigo. 2016. "Enhanced Exact Solution Methods for the Team Orienteering Problem." *International Journal of Production Research* 54 (2): 591–601.
- Kim, Byeongseop, Yongkuk Jeong, and Jong Gye Shin. 2020. "Spatial Arrangement Using Deep Reinforcement Learning to Minimise Rearrangement in Ship Block Stockyards." *International Journal of Production Research* 58: 5062–5076.
- Kizilates, Gözde, and Fidan Nuriyeva. 2013. "On the Nearest Neighbor Algorithms for the Traveling Salesman Problem." In *Advances in Computational Science, Engineering and Information Technology*, edited by Dhinaharan Nagamalai, Ashok Kumar, and Annamalai Annamalai, 111–118. Konya, Turkey: Springer.
- Labadie, Nacima, Renata Mansini, Jan Melechovsk'y, and Roberto Wolfler Calvo. 2012. "The Team Orienteering Problem with Time Windows: An Lp-based Granular Variable Neighborhood Search." *European Journal of Operational Research* 220 (1): 15–27.
- Li, Xia, and Ruibin Bai. 2016. "Freight Vehicle Travel Time Prediction using Gradient Boosting Regression Tree." In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 1010–1015. Anaheim, CA: IEEE. doi:10.1109/ICMLA.2016.0182.
- Liang, Yi, Dongxiao Niu, and Wei-Chiang Hong. 2019. "Short Term Load Forecasting Based on Feature Extraction and Improved General Regression Neural Network Model." *Energy* 166: 653–663.
- Lin, Shih-Wei, and Vincent F. Yu. 2012. "A Simulated Annealing Heuristic for the Team Orienteering Problem with Time Windows." *European Journal of Operational Research* 217 (1): 94–107.
- Lin, Shih-Wei, and Vincent F. Yu. 2017. "Solving the Team Orienteering Problem with Time Windows and Mandatory Visits by Multi-Start Simulated Annealing." *Computers & Industrial Engineering* 114: 195–205.
- Liu, Ran, and Zhibin Jiang. 2019. "A Hybrid Large-Neighborhood Search Algorithm for the Cumulative Capacitated Vehicle Routing Problem with Time-Window Constraints." *Applied Soft Computing* 80: 18–30.
- Liu, Jue, Zhan Pang, and Linggang Qi. 2020. "Dynamic Pricing and Inventory Management with Demand Learning: A Bayesian Approach." *Computers & Operations Research* 124: 105078.
- Liu, Yezheng, Yang Qian, Yuanchun Jiang, and Jennifer Shang. 2020. "Using Favorite Data to Analyze Asymmetric Competition: Machine Learning Models." *European Journal of Operational Research* 287 (2): 600–615. doi:10.1016/j.ejor.2020.03.074.
- Liu, Ran, Yangyi Tao, and Xiaolei Xie. 2019. "An Adaptive Large Neighborhood Search Heuristic for the Vehicle Routing Problem with Time Windows and Synchronized Visits." *Computers & Operations Research* 101: 250–262.
- Mancini, Simona. 2017. "A Combined Multistart Random Constructive Heuristic and Set Partitioning Based Formulation for the Vehicle Routing Problem with Time Dependent Travel Times." *Computers & Operations Research* 88: 290–296.
- Niu, Dongxiao, Yi Liang, and Wei-Chiang Hong. 2017. "Wind Speed Forecasting Based on EMD and GRNN Optimized by FOA." *Energies* 10 (12): 2001.
- Nuraiman, Dian, Melih Ozlen, and John Hearne. 2020. "A Spatial Decomposition Based Math-Heuristic Approach to the Asset Protection Problem." *Operations Research Perspectives* 7: 100141.
- Pandey, Venkatesh, Evana Wang, and Stephen D. Boyles. 2020. "Deep Reinforcement Learning Algorithm for Dynamic Pricing of Express Lanes with Multiple Access Locations." *Transportation Research Part C: Emerging Technologies* 119: 102715.
- Peng, Bo, Yuan Zhang, Zhipeng Lü, T. C. E. Cheng, and Fred Glover. 2020. "A Learning-Based Memetic Algorithm for the Multiple Vehicle Pickup and Delivery Problem with LIFO Loading." *Computers & Industrial Engineering* 142: 106241.
- Principi, Emanuele, Damiano Rossetti, Stefano Squartini, and Francesco Piazza. 2019. "Unsupervised Electric Motor Fault Detection by Using Deep Autoencoders." *IEEE/CAA Journal of Automatica Sinica* 6 (2): 441–451.

- Priore, Paolo, Borja Ponte, Rafael Rosillo, and David de la Fuente. 2019. "Applying Machine Learning to the Dynamic Selection of Replenishment Policies in Fast-Changing Supply Chain Environments." *International Journal of Production Research* 57 (11): 3663–3677.
- Qu, Chengzhi, Wendong Gai, Maiying Zhong, and Jing Zhang. 2020. "A Novel Reinforcement Learning Based Grey Wolf Optimizer Algorithm for Unmanned Aerial Vehicles (UAVs) Path Planning." *Applied Soft Computing* 89: 106099.
- Rashidi, Eghbal, Hugh Medal, Jason Gordon, Robert Grala, and Morgan Varner. 2017. "A Maximal Covering Location-Based Model for Analyzing the Vulnerability of Landscapes to Wildfires: Assessing the Worst-Case Scenario." *European Journal of Operational Research* 258 (3): 1095–1105.
- Roozbeh, Iman, Melih Ozlen, and John W. Hearne. 2018. "An Adaptive Large Neighbourhood Search for Asset Protection During Escaped Wildfires." *Computers & Operations Research* 97: 125–134.
- Sabar, Nasser R., and Graham Kendall. 2015. "Population Based Monte Carlo Tree Search Hyper-Heuristic for Combinatorial Optimization Problems." *Information Sciences* 314: 225–239.
- Saeedvand, Saeed, Hadi S. Aghdasi, and Jacky Baltes. 2020. "Novel Hybrid Algorithm for Team Orienteering Problem with Time Windows for Rescue Applications." *Applied Soft Computing* 96: 106700.
- Sayad, Younes Oulad, Hajar Mousannif, and Hassan Al Moatasime. 2019. "Predictive Modeling of Wildfires: A New Dataset and Machine Learning Approach." *Fire Safety Journal* 104: 130–146.
- Schneider, Michael, and Michael Drexler. 2017. "A Survey of the Standard Location-Routing Problem." *Annals of Operations Research* 259 (1-2): 389–414.
- Shahparvari, Shahrooz, and Babak Abbasi. 2017. "Robust Stochastic Vehicle Routing and Scheduling for Bushfire Emergency Evacuation: An Australian Case Study." *Transportation Research Part A: Policy and Practice* 104: 32–49.
- Shahparvari, Shahrooz, Babak Abbasi, and Prem Chhetri. 2017. "Possibilistic Scheduling Routing for Short-Notice Bushfire Emergency Evacuation Under Uncertainties: An Australian Case Study." *Omega* 72: 96–117.
- Shao, Weishi, Dechang Pi, and Zhongshi Shao. 2018. "A Hybrid Discrete Teaching-Learning Based Meta-Heuristic for Solving No-Idle Flow Shop Scheduling Problem with Total Tardiness Criterion." *Computers & Operations Research* 94: 89–105.
- Shi, Daming, Wenhui Fan, Yingying Xiao, Tingyu Lin, and Chi Xing. 2020. "Intelligent Scheduling of Discrete Automated Production Line Via Deep Reinforcement Learning." *International Journal of Production Research* 58 (11): 3362–3380.
- Shi, Xudong, Qi Kang, MengChu Zhou, Abdullah Abusorrah, and Jing An. 2020. "Soft Sensing of Nonlinear and Multimode Processes Based on Semi-Supervised Weighted Gaussian Regression." *IEEE Sensors Journal* 20 (21): 12950–12960.
- Silva, I. D. B., Marcos Eduardo Valle, Laécio C. Barros, and João Frederico C. A. Meyer. 2020. "A Wildfire Warning System Applied to the State of Acre in the Brazilian Amazon." *Applied Soft Computing* 89: 106075.
- Specht, Donald F. 1991. "A General Regression Neural Network." *IEEE Transactions on Neural Networks* 2 (6): 568–576.
- Sun, Peng, Lucas P. Veelenturf, Mike Hewitt, and Tom Van Woensel. 2020. "Adaptive Large Neighborhood Search for the Time-dependent Profitable Pickup and Delivery Problem with Time Windows." *Transportation Research Part E: Logistics and Transportation Review* 138: 101942.
- Talbi, El-Ghazali. 2016. "Combining Metaheuristics with Mathematical Programming, Constraint Programming and Machine Learning." *Annals of Operations Research* 240 (1): 171–215.
- Tian, Guangdong, Yaping Ren, and MengChu Zhou. 2016. "Dual-Objective Scheduling of Rescue Vehicles to Distinguish Forest Fires via Differential Evolution and Particle Swarm Optimization Combined Algorithm." *IEEE Transactions on Intelligent Transportation Systems* 17 (11): 3009–3021.
- Tinós, Renato. 2020. "Artificial Neural Network Based Crossover for Evolutionary Algorithms." *Applied Soft Computing* 95: 106512.
- Tsiligirides, Theodore. 1984. "Heuristic Methods Applied to Orienteering." *Journal of the Operational Research Society* 35 (9): 797–809.
- Václavík, Roman, Antonín Novák, Přemysl Šůcha, and Zdeněk Hanzálek. 2018. "Accelerating the Branch-and-Price Algorithm Using Machine Learning." *European Journal of Operational Research* 271 (3): 1055–1069.
- van der Merwe, Martijn, James P. Minas, Melih Ozlen, and John W. Hearne. 2015. "A Mixed Integer Programming Approach for Asset Protection During Escaped Wildfires." *Canadian Journal of Forest Research* 45 (4): 444–451.
- Verbeeck, Cédric, Pieter Vansteenwegen, and E.-H. Aghezzaf. 2016. "Solving the Stochastic Time-Dependent Orienteering Problem with Time Windows." *European Journal of Operational Research* 255 (3): 699–718.
- Wang, Teng, Leping Bu, Zhikai Yang, Peng Yuan, and Jineng Ouyang. 2019. "A New Fire Detection Method Using a Multi-Expert System Based on Color Dispersion, Similarity and Centroid Motion in Indoor Environment." *IEEE/CAA Journal of Automatica Sinica* 7 (1): 263–275.
- Wang, Yiyuan, Shiwei Pan, Chenxi Li, and Minghao Yin. 2020. "A Local Search Algorithm with Reinforcement Learning Based Repair Procedure for Minimum Weight Independent Dominating Set." *Information Sciences* 512: 533–548.
- Watson, Gregory L., Donatello Telesca, Colleen E. Reid, Gabriele G. Pfister, and Michael Jerrett. 2019. "Machine Learning Models Accurately Predict Ozone Exposure During Wildfire Events." *Environmental Pollution* 254: 112792.
- Wu, Teresa, Jennifer Blackhurst, and Vellayappan Chidambaram. 2006. "A Model for Inbound Supply Risk Analysis." *Computers in Industry* 57 (4): 350–365.
- Wu, Peng, Feng Chu, Ada Che, and MengChu Zhou. 2017. "Bi-objective Scheduling of Fire Engines for Fighting Forest Fires: New Optimization Approaches." *IEEE Transactions on Intelligent Transportation Systems* 19 (4): 1140–1151.
- Wu, Cheng-Hung, Fang-Yi Zhou, Chi-Kang Tsai, Cheng-Juei Yu, and Stéphane Dauzère-Pérès. 2020. "A Deep Learning Approach for the Dynamic Dispatching of Unreliable Machines in Re-entrant Production Systems." *International Journal of Production Research* 58 (9): 2822–2840.
- Yao, Yuan, and Yi Cao. 2020. "A Neural Network Enhanced Hidden Markov Model for Tourism Demand Forecasting." *Applied Soft Computing* 94: 106465.
- Yu, Qinxiao, Kan Fang, Ning Zhu, and Shoufeng Ma. 2019. "A Matheuristic Approach to the Orienteering Problem with

- Service Time Dependent Profits.” *European Journal of Operational Research* 273 (2): 488–503.
- Yu, Vincent F., Parida Jewpanya, Shih-Wei Lin, and A. A. N. Perwira Redi. 2019. “Team Orienteering Problem with Time Windows and Time-Dependent Scores.” *Computers & Industrial Engineering* 127: 213–224.
- Yu, Vincent F., A. A. N. Perwira Redi, Parida Jewpanya, and Aldy Gunawan. 2019. “Selective Discrete Particle Swarm Optimization for the Team Orienteering Problem with Time Windows and Partial Scores.” *Computers & Industrial Engineering* 138: 106084.
- Zema, Demetrio Antonio, Manuel Esteban Lucas-Borja, Lidia Fotia, Domenico Rosaci, Giuseppe M. L. Sarnè, and Santo Marcello Zimbone. 2020. “Predicting the Hydrological Response of a Forest After Wildfire and Soil Treatments Using an Artificial Neural Network.” *Computers and Electronics in Agriculture* 170: 105280.
- Zhai, Chunjie, Siyu Zhang, Zhaolou Cao, and Xinmeng Wang. 2020. “Learning-based Prediction of Wildfire Spread with Real-time Rate of Spread Measurement.” *Combustion and Flame* 215: 333–341.
- Zhang, Zhihua. 2018. “Artificial Neural Network.” In *Multivariate Time Series Analysis in Climate and Environmental Research*, edited by Zhihua Zhang, 1–35. Cham: Springer.
- Zhang, Shu, Jeffrey W. Ohlmann, and Barrett W. Thomas. 2014. “A Priori Orienteering with Time Windows and Stochastic Wait Times at Customers.” *European Journal of Operational Research* 239 (1): 70–79.
- Zhao, Haitong, and Changsheng Zhang. 2020. “A Decomposition-Based Many-Objective Artificial Bee Colony Algorithm with Reinforcement Learning.” *Applied Soft Computing* 86: 105879.
- Zhou, Siqiong, and Ayca Erdogan. 2019. “A Spatial Optimization Model for Resource Allocation for Wildfire Suppression and Resident Evacuation.” *Computers & Industrial Engineering* 138: 106101.