A decomposition approach for the stochastic asset protection problem

Nuraiman, Dian; Ozlen, Melih; Hearne, John

Nuraiman, D., Ozlen, M., & Hearne, J. (2022). A decomposition approach for the stochastic asset protection problem. Computers and Operations Research, 138(February 2022), 1–16.
https://doi.org/10.1016/j.cor.2021.105591
Document Version: Accepted Manuscript

Published Version: https://doi.org/10.1016/j.cor.2021.105591

A decomposition approach for the stochastic asset protection problem

Dian Nuraiman[a,b,*], Melih Ozlen[a], John Hearne[a]

[a]*School of Science, RMIT University, Melbourne, Australia*
[b]*Department of Mathematics, UIN Sunan Gunung Djati, Bandung, Indonesia*

Abstract

The deterministic Asset Protection Problem (APP) involves deploying firefighting resources of various capabilities to service as many community assets as possible within time windows determined by an advancing wildfire. A common situation arising during these situations is a wind change. Forecasts of changes in wind velocity (i.e. direction and speed) are reasonably accurate but there is some uncertainty around the time of a wind change. The timing has implications for which areas, and hence which assets, will be impacted by the wildfire. This presents a difficult problem for an Incident Management Team (IMT) operating under severe time pressure as the wildfire sweeps across the landscape. In this study, we extend the spatial decomposition-based math-heuristic originally developed for the deterministic APP to handle large real life sized stochastic APPs in operational time. This is achieved by regarding the deterministic and stochastic components as a coupled system. A two-stage stochastic programming (TSSP) model is thus only used for a smaller sub-problem around the uncertainty. We found this approach outperformed other methods when tested on a new set of benchmarks. More importantly, the accuracy and solution times make the method suitable for operational purposes.

Keywords: Wildfires, Asset protection problem, Team orienteering problem, Two-stage stochastic programming, Math-heuristic, Spatial Decomposition

1. Introduction

Wildfires can be catastrophic natural events leading to individual, community, economic, and environmental impacts. Australia is one of many countries vulnerable to wildfires. In the recent 2019-2020 bushfire season, for example, fires burnt an estimated 286000 km^2, destroyed nearly 6000 buildings, and killed at least 34 people. Under certain conditions a large wildfire can become so large that there is no reasonable prospect

*Corresponding author
Email addresses: dian.nuraiman@rmit.edu.au (Dian Nuraiman), melih.ozlen@rmit.edu.au (Melih Ozlen), john.hearne@rmit.edu.au (John Hearne)

of bringing it under control. Under such circumstances it is prudent to deploy firefighting resources to service as many critical assets as possible. This might involve hosing down a structure and clearing debris in its vicinity to reduce the risk of destruction by the wildfire. Such activities must be done within a time window associated with each asset. This time window is determined by the progress of the wildfire as it spreads across the landscape. Servicing an asset too early might mean, for example, that more debris is blown into the vicinity of the asset before the wildfire passes through. At the other end of the time window it is obvious that teams must be well clear of the fire before it arrives. The tasks required at some assets require a synchronous visit by more than one team. The Incident Management Team (IMT) must consider the values and requirements of each critical asset. Each resource (e.g. trucks, tankers, pumpers) is deployed along a particular route to arrive at various assets within their time windows. Moreover, this must be done to maximise the total value of assets serviced. This problem is known as the Asset Protection Problem (APP).

IMT's may face a wind change during escaped wildfires. It is a real phenomenon such as in the 2009 'Black Saturday' bushfires where the wind direction suddenly changed around 90 degrees from north-westerly to south-westerly. This meant that the flank of this long, narrow fire became the new and very broad fire-front. This had a significant impact on the path of the fire and the area burnt (Cruz et al., 2012). Such wind changes are included in the weather forecast in Australia by the Bureau of Meteorology. Wind velocity (i.e. speed and direction) and the expected time of change are provided (Bureau of Meteorology, 2020). The exact time of the wind change, however, is subject to some uncertainty. The timing of the wind change may affect time windows but more importantly it will affect which areas and assets will be in the path of the advancing fire.

The studies in the literature dealing with stochastic events for the APP are limited. First is the work by van der Merwe et al. (2017). Their re-routing approach utilised a Mixed Integer Programming model to amend routes and schedules in response to disruptions during the deployment to the original optimal plan. Second was the work done by Roozbeh et al. (2018a). They developed a two-stage stochastic program (TSSP) for the stochastic APP with uncertainty around the timing of a wind change. They proposed a dynamic rerouting approach with a large neighbourhood search (LNS) heuristic. This enabled approximate solutions to be obtained for large problems where CPLEX failed. No information on the quality of these solutions, however, could be provided.

Despite the developed dynamic rerouting approach, finding an efficient solution approach for realistically large-scale stochastic APPs in operational time is still challenging. Grass & Fischer (2016) highlighted that solving large-scale problems in the humanitarian context is still an issue for those who use commercial solvers. Heuristics provide an alternative to commercial solvers. In practice, however, they are not easy to implement especially for practitioners, such as humanitarian aid organizations, as the heuristics require a special kind of coding effort. Furthermore, a heuristic is often specific to a particular model and not generally applicable to other models. Hence finding computationally tractable solution techniques for the stochastic APP is still a big challenge. Although humanitarian practitioners may find it difficult to apply heuristics, we propose a heuristic approach that is easy to understand and use for existing solver users. We use the spatial decomposition based math-heuristic (SDM) approach as the best current solution technique for the deterministic version of the APP and extend it to solve large-scale stochastic APPs. We do this by decomposing the problem into several smaller sub-problems based on spatial considerations, and separating the deterministic and stochastic models. The stochastic model is only used for an area impacted by uncertainty, while the rest of the sub-problems are deterministic. The sub-problems of the decomposed problem are then solved using a commercial solver. We carry out experimental results of the proposed solution approach compared to other solution techniques that are applicable using commercial solvers. Note that the commercial solvers can easily be replaced with open-source solvers that are freely available if required.

The contributions of this paper against similar work in the literature include two aspects. First, we refine the existing model developed by Roozbeh et al. (2018a) which will be explained in Section 3. Second, we propose a new efficient solution method based on the SDM approach by Nuraiman et al. (2020). We extend the SDM method to solve the stochastic APP by a coupling strategy of the deterministic and stochastic models. We evaluate the performance of each solution approach with extensive experiments using new benchmark instances involving randomised, clustered, and randomised-clustered instances.

The rest of this paper is organised as follows. Section 2 reveals the position of this work in the related literature. In Section 3, the problem definition and the two-stage stochastic programming model are presented. Section 4 is focused on solution techniques comprising a dynamic rerouting approach, a two-phase decomposition approach, a deterministic SDM approach and the proposed SDM approach. Computational experiments are given in Section 5, followed by the Conclusion.

2. Literature review

The APP is an emergency resource routing and scheduling (ERRS) problem with the goal of finding the routes of vehicles and the visit times of assets. ERRS refers to a dispatching problem of critical resources to the community without delay with fast response time and restricted time windows. The resources include vehicles (e.g., ambulances (Talarico et al., 2015), evacuation vehicles (Zhao et al., 2020), fire trucks (Usanov et al., 2020)), manpower (e.g., nurses, firefighters), relief commodities (Bodaghi et al., 2018) and medical services (Aringhieri et al., 2017). While most studies focus on reducing the loss of human life or are human oriented, the APP concentrates on servicing community assets or properties. Such assets may take weeks or even months to be restored if they are not protected during wildfires.

Furthermore, the APP has similar characteristics to the team orienteering problem with time windows (TOPTW). The team must cooperate to maximise the total collected score or profit by servicing a given number of nodes within their pre-defined time windows. While the team in the classical TOPTW is homogeneous (Gunawan et al., 2016), the APP considers a fleet of heterogeneous vehicles where each vehicle type has different capability. This is a special characteristic of the APP that each asset may require a mix of services from different types of vehicles at the same time. A number of variants for TOPTW with heterogeneous vehicles exists in the literature. Saeedvand et al. (2020) studied rescue operations by allocating a team of varied humanoid robots. Each robot can accomplish any task but the energy consumption for each robot in each task may be different. Hanafi et al. (2020) studied the TOPTW with heterogeneous vehicles in a furniture company. Each customer is allowed to be visited more than once by a team of different qualified technicians based on a predetermined order to complete a set of assembly tasks. In the APP, each asset can only be visited once and hence the team of different vehicles required must synchronise their visits to complete the services. This synchronisation requirement with the TOPTW makes the APP a rare problem.

In the context of routing problems, the APP is similar to the vehicle routing problem (VRP). The main difference is that in the VRP vehicles must service all customers, while in the APP the available vehicles may not cover all assets due to time constraints and limited resources. Furthermore, the main objective of the VRP is the minimisation of total traveling cost, while the APP aims to maximise the total collected rewards. A variant of the VRP that is relevant to the APP is the VRP with profits. In logistics operations, for instance, frequent customers can be set as mandatory visits with fixed profits while visits to non-frequent customers with known profits are optional as studied in (Stavropoulou et al., 2019). This setting leads to a combination of a vehicle routing problem and an orienteering problem with the goal of maximising the total net profit collected while minimising the total traveling distance. This problem is closely related to a variant of TOPTW with mandatory visits (Lin & Yu, 2017). In the APP, there is no asset that must be visited, but the more valuable the asset, the higher the priority it should receive to be serviced.

In the VRP, synchronisation constraints have been widely used in various real-life applications (Drexl, 2012). For instance, in the home health care staff scheduling and routing problem, an elderly patient may require service from two different skilled caregivers at the same time (Bredström & Rönnqvist, 2008; Afifi et al., 2016; Liu et al., 2021). A similar problem addressed by Lianes et al. (2021) is a vessel routing problem. A fish farm may require more than one vessel with distinct capabilities to perform service simultaneously such as net installation. In terms of synchronisation requirements, the APP is similar to the mentioned studies in the VRP.

The deterministic model of the APP was first formulated by van der Merwe et al. (2015) as a mixed integer programming problem. They used a commercial solver to solve the model for small-size instances. The solver was not able to solve large problems within operational time. Subsequently an adaptive large neighborhood search (ALNS) was developed by Roozbeh et al. (2018b) to handle large-scale instances consisting of 100 and 200 nodes within times suitable for operational purposes. A recent endeavor to find more efficient solution methods led to a spatial decomposition based math-heuristic (SDM) by Nuraiman et al. (2020). This approach yielded better quality solutions compared with the ALNS against the same machine specifications and benchmark instances. The SDM exploits the nature of the APP, where time windows are geographically correlated, by decomposing a large APP into tractable sub-problems in line with the solver's ability to solve such smaller problems.

The ubiquity of uncertainty has led to the use of stochastic programming in many real-world applications. In the context of disaster management, the two-stage stochastic programming (TSSP) framework is often used to measure the expected natural hazard especially to the community. In fact, the magnitude, location, and timing of natural disasters are often unpredictable. While most studies in humanitarian operations assume complete information is known beforehand with certainty (Farahani et al., 2020), involving uncertain parameters can lead to a more realistic and meaningful approach.

In humanitarian logistics, travel times and demands are often uncertain in post-disaster relief operations (Li & Chung, 2019; Garrido et al., 2015). In addition, supply and network availability can be subject to uncertainty in the aftermath of an earthquake (Tofighi et al., 2016). Logistics providers may also face uncertain levels of patient severity as recently studied by Jamali et al. (2021). They consider three different injury severity degrees consisting of minor, moderate and serious conditions, where the number of victims in each severity level is uncertain. Medical supplies should be delivered to beneficiaries in a timely manner to minimise the deprivation cost. Thus decision makers should develop a distribution plan of essential supplies involving the flexibility and adaptability of vehicle routes in response to any sudden changes considered.

In the context of bushfire emergency management, a few routing problems under uncertainties have been addressed in the literature. Shahparvari et al. (2019) studied a bushfire evacuation routing problem under a short-notice condition with uncertainty due to road network and shelter disruption. Bodaghi et al. (2020) addressed emergency relief operations with both expandable and non-expandable resources under 1000 plausible scenarios in the severity level and needs at each bushfire affected demand point. Stochastic APP, the focus of this study, considers uncertain timing of a wind change during wildfires that requires consideration of multiple scenarios. This leads to uncertainty in two aspects; assets impacted and their time windows. While this is not the first study of the stochastic APP, we develop a new solution approach that is capable of handling large-scale stochastic APPs.

3. Problem definition and formulation

This section presents the two-stage stochastic programming (TSSP) model developed by Roozbeh et al. (2018a) to solve the stochastic APP. We refined their model to make it more accurate based on practical requirements. First, we removed all big M values from the constraints in the existing model and replaced them with the most appropriate values to provide tighter constraints. Second, to be more realistic, we dropped the service duration from Constraints (6) and (7) in their model. This aims to ensure that stage two decision variables can only have an impact after the actual staging time, i.e., the vehicles cannot start travel to their stage 2 destinations before the staging time.

The model presented is capable of considering multiple scenarios where the wind change can occur at several possible times with a single staging time denoted by ST. The staging time refers to the earliest possible time of the wind change. We use this based on the assumption that the planning is made several hours before the wind change based on the forecast at that time with some uncertainty around the precise time. As time progresses, the forecast is updated. When it is close to the staging time, the forecast can give us an update on when the wind will change with high accuracy. This update then informs the IMT's of which scenario should be chosen at the staging time.

The wind change will affect the direction of fire propagation. Hence, this and the time of the change will determine the region impacted. As a consequence of multiple scenarios, several assets may be impacted by only one scenario or more. If an asset is at risk by three scenarios, for instance, then the asset may have three different time windows. The TSSP determines the optimal set of stage 1 assets to service with a view to maximising the total expected asset values serviced in stage 2. An asset is a stage 1 asset if it is serviced before the staging time, otherwise it is a stage 2 asset. If the time window of an asset is around the staging time, this asset can be serviced in either stage 1 or stage 2.

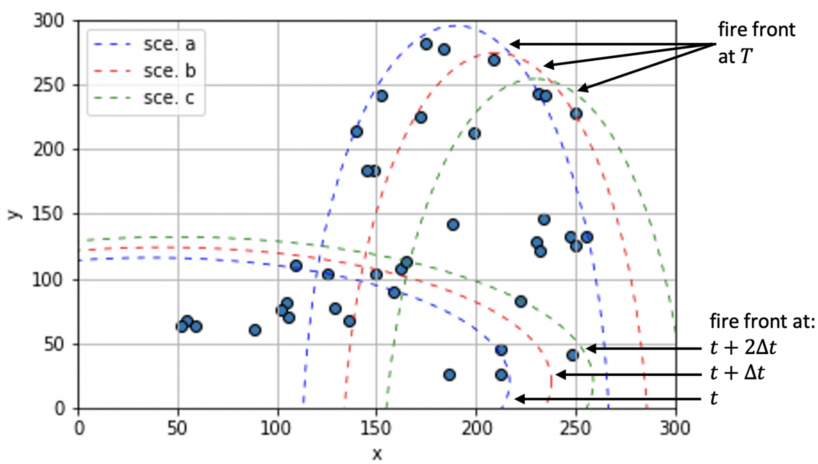


Figure 1: Problem illustration

Figure 1 illustrates an estimate of fire affected regions before and after a wind change based on a forecast of wind velocity and fire propagation speed. A number of assets marked with blue circles are at risk. The fires initially propagate from the lower left domain in an elliptical shape and then turn dramatically by 90 degrees in a vertical direction. For illustrative purposes, the figure shows that the wind direction can change in three possible times with three scenarios. First, scenario a refers to a change at time t with a probability of $P(a)$ where the fire front is indicated in blue lines. Second, scenario b reflects a possible delay of the change at time $t + \Delta t$ with a probability of $P(b)$ where the fire front is indicated in red lines. Third, the last scenario is a change at time $t + 2\Delta t$ with a probability of $P(c)$ where the fire front is indicated in green lines. Here Δt is the delay time between two scenarios and the probabilities are one in total. As the wind change in scenario b and c occurs at the later times, fires still spread in the initial direction and only change direction later. The fire spread is observed over a time horizon of T.

A set of assets $C = \{1, 2, ..., |C|\}$ is given with a starting depot 0 and a final depot $|C| + 1$, which may be at the same location as the start. A set of vehicle types $V = \{1, 2, \ldots, k, \ldots, |V|\}$ comprising Q_k vehicles for each vehicle of type k is available at the starting depot to service the given set of assets. Each asset i may require a mix of vehicle types denoted by $(r_{i1}, r_{i2}, \ldots, r_{ik}, \ldots, r_{i|V|})$ where r_{ik} refers to the number of vehicles of type k required at asset i. This means that at asset i, all required vehicles must synchronise their visits and start service at time s_i within a given time window $[o_i, c_i]$ for a duration of d_i. Furthermore, each asset i has an associated value ϑ_i to the community. This will affect its chance of being serviced as the objective function seeks to maximise the total value of assets serviced.

In order to reduce the model size, a set of accessible assets for vehicles of type k going out from, and coming into, each asset i are introduced and denoted by N_{i+}^k and N_{i-}^k, respectively. Vehicles of type k can go to asset j after servicing asset i if they meet time and vehicle type constraints. The first constraint requires the vehicles must be able to complete the service at i and travel to arrive at j by the closing time c_j for starting the service at j. The second constraint represents that both asset i and j require the same type k of vehicles.

The following is a list of the main notation used in the model.

Sets

C : set of asset nodes, where $C = \{1, 2, \ldots, |C|\}$
N : set of nodes, where $N = \{0, 1, 2, \ldots, |C| + 1\}$
S : set of scenarios in stage 2 indexed by g, where $S = \{1, 2, ..., |S|\}$
V : set of vehicle types indexed by k, where $V = \{1, 2, \ldots, |V|\}$
N_{i+}^k : set of accessible assets from asset i by vehicles of type k in stage 1
N_{i-}^k : set of accessible assets to asset i by vehicles of type k in stage 1
$N_{i+}^{k,g}$: set of accessible assets from asset i by vehicles of type k in stage $2g$
$N_{i-}^{k,g}$: set of accessible assets to asset i by vehicles of type k in stage $2g$

Parameters

ϑ_i : value of asset i
d_i : service duration at asset i
r_{ik} : number of vehicles of type k required at asset i
t_{ijk} : travel time from asset i to j by vehicles of type k
Q_k : number of vehicles of type k available at depot
o_i : opening service time at asset i in stage 1

c_i : closing service time at asset i in stage 1
o_i^g : opening service time at asset i in stage $2g$
c_i^g : closing service time at asset i in stage $2g$
P_g : probability of scenario g
ST : staging time

Decision variables

$$z_{ijk} = \begin{cases} 1, & \text{if vehicles of type } k \text{ travelling from asset } i \text{ to } j \text{ in stage 1} \\ 0, & \text{otherwise} \end{cases}$$

$$z_{ijk}^g = \begin{cases} 1, & \text{if vehicles of type } k \text{ travelling from asset } i \text{ to } j \text{ in stage } 2g \\ 0, & \text{otherwise} \end{cases}$$

$$y_i = \begin{cases} 1, & \text{if asset } i \text{ is serviced in stage 1} \\ 0, & \text{otherwise} \end{cases}$$

$$y_i^g = \begin{cases} 1, & \text{if asset } i \text{ is serviced in stage } 2g \\ 0, & \text{otherwise} \end{cases}$$

$$w_i = \begin{cases} 1, & \text{if asset } i \text{ is serviced just before the staging time} \\ 0, & \text{otherwise} \end{cases}$$

x_{ijk} : number of vehicles of type k travelling from asset i to j in stage 1
x_{ijk}^g : number of vehicles of type k travelling from asset i to j in stage $2g$
s_i : service starting time of asset i in stage 1
s_i^g : service starting time of asset i in stage $2g$

The objective function can be written as follows

$$\max \sum_{i \in C} \vartheta_i y_i + \sum_{g \in S} \sum_{i \in C} P_g \vartheta_i y_i^g \tag{1}$$

subject to:

$$y_i + y_i^g \leq 1, \quad \forall i \in C, g \in S \tag{2}$$

$$\sum_{j \in N_{0+}^k} x_{0jk} + \sum_{j \in N_{0+}^{k,g}} x_{0jk}^g \leq Q_k, \quad \forall k \in V, g \in S \tag{3}$$

$$\sum_{i \in N_{h-}^k} x_{ihk} + \sum_{i \in N_{h-}^{k,g}} x_{ihk}^g = \sum_{j \in N_{h+}^k} x_{hjk} + \sum_{j \in N_{h+}^{k,g}} x_{hjk}^g, \quad \forall h \in C, k \in V, g \in S \tag{4}$$

$$\sum_{i \in N^k_{h-}} x_{ihk} = r_{hk} y_h, \quad \forall h \in C, k \in V \tag{5}$$

$$\sum_{i \in N^{k,g}_{h-}} x^g_{ihk} = r_{hk} y^g_h, \quad \forall h \in C, k \in V, g \in S \tag{6}$$

$$t_{0jk} z_{0jk} \leq s_j, \quad \forall j \in N^k_{0+}, k \in V \tag{7}$$

$$t_{0jk} z^g_{0jk} \leq s^g_j, \quad \forall j \in N^{k,g}_{0+}, k \in V, g \in S \tag{8}$$

$$s_i - c_i + (c_i + d_i + t_{ijk}) z_{ijk} \leq s_j, \quad \forall i \in C, j \in N^k_{i+}, k \in V \tag{9}$$

$$s^g_i - c^g_i + (c^g_i + d_i + t_{ijk}) z^g_{ijk} \leq s^g_j, \quad \forall i \in C, j \in N^{k,g}_{i+}, k \in V, g \in S \tag{10}$$

$$\sum_{k \in V} \sum_{i \in N^k_{h-}} x_{ihk} - \sum_{k \in V} \sum_{j \in N^k_{h+}} x_{hjk} \leq \sum_{k \in V} r_{hk} w_h, \quad \forall h \in C \tag{11}$$

$$\sum_{k \in V} \sum_{i \in N^k_{h-}} x_{ihk} - \sum_{k \in V} \sum_{j \in N^k_{h+}} x_{hjk} \geq w_h, \quad \forall h \in C \tag{12}$$

$$s_i - c_i + c_i w_i \leq s^g_i, \quad \forall i \in C, g \in S \tag{13}$$

$$s_i \geq s^g_i - c^g_i + c^g_i w_i, \quad \forall i \in C, g \in S \tag{14}$$

$$s_j - c_j + c_j z_{ijk} < ST, \quad \forall i \in C, i \in N^k_{j-}, k \in V \tag{15}$$

$$s^g_j \geq ST z^g_{ijk}, \quad \forall i \in C, i \in N^k_{j-}, k \in V, g \in S \tag{16}$$

$$z_{ijk} + z^g_{ijk} \leq 1, \quad \forall i \in C, i \in N^k_{j-}, k \in V, g \in S \tag{17}$$

$$x_{ijk} \leq r_{jk} z_{ijk}, \quad \forall i \in N, j \in N^k_{i+}, k \in V \tag{18}$$

$$x^g_{ijk} \leq r_{jk} z^g_{ijk}, \quad \forall i \in N, j \in N^k_{i+}, k \in V, g \in S \tag{19}$$

$$o_i \leq s_i \leq c_i, \quad \forall i \in C \tag{20}$$

$$o^g_i \leq s^g_i \leq c^g_i, \quad \forall i \in C, g \in S \tag{21}$$

$$w_i, y_i, y^g_i \in \{0, 1\}, \quad \forall i \in C, \forall g \in S \tag{22}$$

$$z_{ijk}, z^g_{ijk} \in \{0, 1\}, \quad \forall i \in N, j \in N^k_{i+}, k \in V, g \in S \tag{23}$$

The objective (1) aims to maximise the total expected asset values serviced before and after a wind change. Each asset can be serviced only once in either stage 1 or stage 2 as written in constraint (2). Constraint (3) expresses that the number of vehicles of type k available at the initial depot is the upper limit of the total number of vehicles of type k going out from the depot. Constraint (4) guarantees the flow conservation of vehicles of type k at each asset. Constraints (5) and (6) enforce vehicles of type k required at each asset in both stages to synchronise their visits. The service starting times at the first assets serviced by vehicles of type k must be greater than or equal to the travel times from the initial depot. This is expressed in constraint (7) for stage 1 and constraint (8) for each scenario in stage 2. The service starting times of two sequentially serviced assets must fulfill constraint (9) for stage 1 and constraint (10) for each scenario in stage 2. A stage 1 asset is called a staging node if there is any vehicle going out from the node to stage 2 assets as expressed in constraints (11) and (12). Constraints (13) and (14) indicate that the service starting time of a staging node is copied from stage 1 to stage 2 variables. Constraints (15) and (16) ensure that stage 1 assets are serviced before the staging time and stage 2 assets are serviced after the staging time. Constraint (17) defines that if there is a flow of vehicles of type k from node i to j in stage 1, there must be no flow in stage 2 between the two nodes. On the contrary, if there is no flow from i to j in stage 1, there may be a flow in stage 2. Constraints (18) and (19) define that the number of vehicles of type k traveling from node i to j is limited by the number of vehicles of type k required at node j. Constraints (20) and (21) define the time windows of each asset in stage 1 and 2, respectively.

4. Solution techniques

This section presents the dynamic rerouting (DR) approach and three new proposed solution techniques, i.e. two-phase decomposition (TPD), deterministic SDM (D-SDM), and deterministic-stochastic SDM (DS-SDM). All solution techniques decompose the given set of assets C into $|S|+1$ smaller subsets of assets at the beginning of the process where $|S|$ refers to the number of scenarios. C is decomposed into C_1 and C_{2g} for each $g \in S$. C_1 refers to a set of stage 1 assets that are potentially serviced before the staging time ST, while C_{2g} are stage 2 assets that are at risk after the staging time in scenario $g \in S$. An asset i is included in C_1 if the closing time window $c_i < ST$ and is included in C_{2g} if the closing time window $c_i^g \geq ST$. If ST occurs during the time window of an asset i, i.e. $o_i < ST < c_i$, the asset is included in both C_1 and C_{2g}. This means that the asset can be serviced either in stage 1 or stage 2 and should be considered in both stages. All such overlapping assets are then denoted as the assets in a buffer zone.

The DR, TPD and D-SDM utilise the solution of deterministic models while the DS-SDM utilise the solution of both deterministic and TSSP models. However, the deterministic model has no constraints related to overlapping assets in Constraint (2) and the imposed staging time in Constraints (15) and (16). Hence for any solution technique utilising a fully deterministic model must be consistent with the TSSP. This can be achieved in two ways. First the time window of each asset in the buffer zone can be split into two smaller time windows, i.e. before and after the staging time. The time window before the staging time is only utilised in stage 1 decisions, and stage 2 time windows are only utilised in stage 2. Second if an overlapping asset has been serviced in stage 1, this asset must be removed from stage 2 consideration.

4.1. Dynamic rerouting approach

The dynamic rerouting approach used in this paper was developed by Roozbeh et al. (2018a) for the stochastic APP. It is a solution technique where the event that is most likely to occur is chosen as the initial plan and a response to a disruption is applied only after the disruption. Given $|S|$ scenarios, this approach decomposes the problem into $|S|$ sub-problems. The first sub-problem involves stage 1 assets C_1 and the most likely scenario in stage 2, denoted as C_{2G}. Other sub-problems involve only the assets impacted by each scenario $g \in S$ in stage 2, denoted as C_{2g}, where $g \neq G$. Each sub-problem is solved using the deterministic model of the APP.

Algorithm 1 presents the DR approach. After decomposing the assets into smaller subsets of assets in step 2, the first sub-problem, as the most likely scenario of a wind change, is first solved as written in step 4. The last assets serviced before the staging time, referred to as staging locations, need to be extracted from the solution of the first sub-problem, as written in step 5. They are used as the initial locations of vehicles for the rest of sub-problems in stage 2 denoted as a set of depots D_2. The rest of the algorithm is dedicated to solve the rest of the scenarios independently as written in step 6-12.

Algorithm 1: Dynamic rerouting approach algorithm

Input: all sets and parameters listed in Section 2

Output: $x_{ijk}, z_{ijk}, y_i, s_i$

1 $D_1 \leftarrow \{0\}$

2 Decompose C into C_1 and C_{2g} for each $g \in S$

3 Find a scenario G in stage 2 which has the highest probability

4 Solve the deterministic model for $C_1 \cup C_{2G}$ with depot D_1 and objective $\max \sum_{i \in C_1} \vartheta_i y_i + \sum_{i \in C_{2G}} P_G \vartheta_i y_i$

5 Extract staging locations from the solution and assign them as depots D_2

6 $g \leftarrow 1$

7 **while** $g \leq |S|$ **do**

8   **if** $g \neq G$ **then**

9   Solve the deterministic model for C_{2g} with depots D_2 and objective $\max \sum_{i \in C_{2g}} P_g \vartheta_i y_i$

10         $g \leftarrow g + 1$

11   **end**

12 **end**

4.2. Two-phase decomposition approach

A TSSP model basically considers uncertainty taking place within a planning horizon and introduces a two-stage problem as illustrated in Figure 2. This can lead to a basic decomposition approach called a two-phase decomposition (TPD) approach by splitting the planning horizon into two phases on a stage basis, i.e., pre- and post- an uncertain event. The first phase is the stage before the realisation of the random event, while the second phase is for after the realisation. The second phase consists of a number of scenarios where each scenario has a given probability. Both phases are solved using the deterministic model.

The full algorithm of the TPD approach is listed in Algorithm 2. To deal with the uncertainty of which scenario will occur, the objective function of the first phase

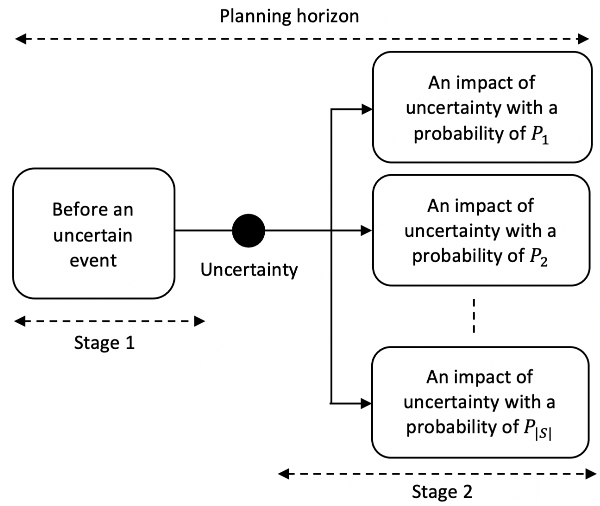


Figure 2: TSSP framework

Algorithm 2: Two-phase decomposition approach algorithm

Input: all sets and parameters listed in Section 2

Output: $x_{ijk}, z_{ijk}, y_i, s_i$

1 $D_1 \leftarrow \{0\}$

2 Decompose C into C_1 and C_{2g} for each $g \in S$

3 Calculate a direction angle $\bar{\theta} = \sum_{g \in S} \frac{P_g \sum_{i \in C_{2g}} w_i \theta_i}{\sum_{i \in C_{2g}} w_i}$

4 Solve the deterministic model for C_1 with depot D_1 and objective
$\max \left(\sum_{i \in C_1} \left(\vartheta_i - \alpha |\theta_i - \bar{\theta}| \right) y_i - \sum_{i \in C_1} s_i / \sum_{i \in C_1} c_i \right)$

5 Extract the last serviced nodes from the solution and assign them as depots D_2

6 $g \leftarrow 1$

7 **while** $g \leq |S|$ **do**

8   Solve the deterministic model for C_{2g} with depots D_2 and objective
$\max \sum_{i \in C_{2g}} P_g \vartheta_i y_i$

9     $g \leftarrow g + 1$

10 **end**

is directed towards a direction angle. This draws on the decomposition procedure by Nuraiman et al. (2020) who used a direction angle from the subsequent stage to influence the solution in the current stage. Before solving the first phase in step 4, a target direction $\bar{\theta}$ is first calculated in step 3 as a probability-weighted direction angle from each scenario in the second phase. It is written as

$$\bar{\theta} = \sum_{g \in S} \frac{P_g \sum_{i \in C_{2g}} w_i \theta_i}{\sum_{i \in C_{2g}} w_i} \tag{24}$$

where θ_i is an angle of node i and w_i is the weight which can be calculated by

$$w_i = \vartheta_i / \sum_{k \in V} \frac{r_{ik}}{Q_k} \tag{25}$$

The weighting constant includes the asset value as well as the ratio between the requirements of the node and the availability of vehicles. This target direction is then used for a secondary objective to minimise the angle of deviation from each asset serviced.

In addition to aligning the solution between two phases using a target direction, the protective services at each asset in the first phase should start as early as possible within a given time window to give more flexibility for vehicles in the next phase. Minimising start servicing times is another important secondary objective for a decomposition approach (Nuraiman et al., 2020). Hence the complete objective of the first phase is written as

$$\max \left(\sum_{i \in C_1} \left(\vartheta_i - \alpha |\theta_i - \bar{\theta}| \right) y_i - \sum_{i \in C_1} s_i / \sum_{i \in C_1} c_i \right) \tag{26}$$

where $\alpha \in (0, 1]$ is a constant to reduce the magnitude of angle deviation against asset value ϑ.

The solution of the first phase becomes an input for the second phase as written in step 5. The assets serviced last in the first phase are the starting locations for vehicles in the next phase denoted as a set of depots D_2. Steps 6-10 of the algorithm solve the sub-problem of each scenario in the second phase. As the second phase is the last one, the two secondary objectives of minimising the start times and angle deviation are removed from the objective function. Hence the objective in the second phase is only maximising the asset values weighted by its scenario probability as written in step 8.

4.3. Spatial decomposition based math-heuristic approach

The spatial decomposition based math-heuristic (SDM) approach was first proposed by Nuraiman et al. (2020) for the deterministic APP. It takes advantage of the nature of problems where the time windows are pre-defined and spatially correlated as in the APP. The time window of an asset is generated based on the expected time the fire will reach the location. We introduce two different implementations of the SDM approach for the stochastic APP as explained below.

4.3.1. Deterministic SDM approach

The deterministic SDM (D-SDM) approach uses the deterministic model to solve the stochastic APP. This approach is basically similar to the TPD approach where stage 1 is first solved then the solution is used as an input for stage 2. However, the TPD approach may have difficulty when the sub-problem in both stages is still large for solvers. Hence the D-SDM approach decomposes each stage into equally tractable sub-problems as illustrated in Figure 3. Given $|S|$ scenarios, assets impacted in stage 1 and each scenario $g \in S$ in stage 2 are split into λ_1 and λ_{2g} parts, respectively. The number

Algorithm 3: D-SDM approach algorithm

Input: all sets and parameters listed in Section 2,α, λ_1, λ_{2g}
Output: $x_{ijk}, z_{ijk}, y_i, s_i$
$D_1 \leftarrow \{0\}$
Decompose C into C_1 and C_{2g} for each $g \in S$
Sort C_1 and each C_{2g} based on their opening time windows in an ascending order
Decompose C_1 into λ_1 parts and C_{2g} into λ_{2g} parts for each $g \in S$ where set of assets in stage st part h denoted as $C_{st,h}$
`Deterministic` $(1, D_1, C_{1,h}, \lambda_1, 0)$
$g \leftarrow 1$
while $g \leq |S|$ **do**
    `Deterministic` $(2, D_{\lambda_1+1}, C_{2g,h}, \lambda_{2g}, g)$
    $g \leftarrow g + 1$
end

Function `Deterministic(`$st, D, C_{st,h}, \lambda, g$`):`
    $h \leftarrow 1$
    $D_h \leftarrow D$
    while $h \leq \lambda$ **do**
        Calculate node angle θ_i for each node i in $C_{st,h}$
        if $st = 1$ & $h = \lambda$ **then**
            Calculate a direction angle $\bar{\theta}_h = \sum_{g \in S} \frac{P_g \sum_{i \in C_{2g,1}} w_i \theta_i}{\sum_{i \in C_{2g,1}} w_i}$
        else if $h < \lambda$ **then**
            Calculate a direction angle $\bar{\theta}_h = \frac{\sum_{i \in C_{st,h+1}} w_i \theta_i}{\sum_{i \in C_{st,h+1}} w_i}$
        end
        if $st = 1$ **then**
            Solve the deterministic model for $C_{st,h}$ with depots D_h and objective $\max\left(\sum_{i \in C_{st,h}} \big(\vartheta_i - \alpha|\theta_i - \bar{\theta}_h|\big) y_i - \sum_{i \in C_{st,h}} s_i / \sum_{i \in C_{st,h}} c_i\right)$
        else if $st = 2$ & $h < \lambda$ **then**
            Solve the deterministic model for $C_{st,h}$ with depots D_h and objective $\max\left(\sum_{i \in C_{st,h}} \big(P_g \vartheta_i - \alpha|\theta_i - \bar{\theta}_h|\big) y_i - \sum_{i \in C_{st,h}} s_i / \sum_{i \in C_{st,h}} c_i\right)$
        else if $st = 2$ & $h = \lambda$ **then**
            Solve the deterministic model for $C_{st,h}$ with depots D_h and objective $\max \sum_{i \in C_{st,h}} P_g \vartheta_i y_i$
        end
        Extract the last serviced nodes and assign them as depots D_{h+1}
        Involve vehicles that are not in service and their locations into D_{h+1}
        $h \leftarrow h + 1$
    end
End Function

of partitions either in stage 1 or each scenario in stage 2 can be different depending on the number of nodes impacted.

Algorithm 3 provides a complete procedure of the D-SDM approach. Before splitting

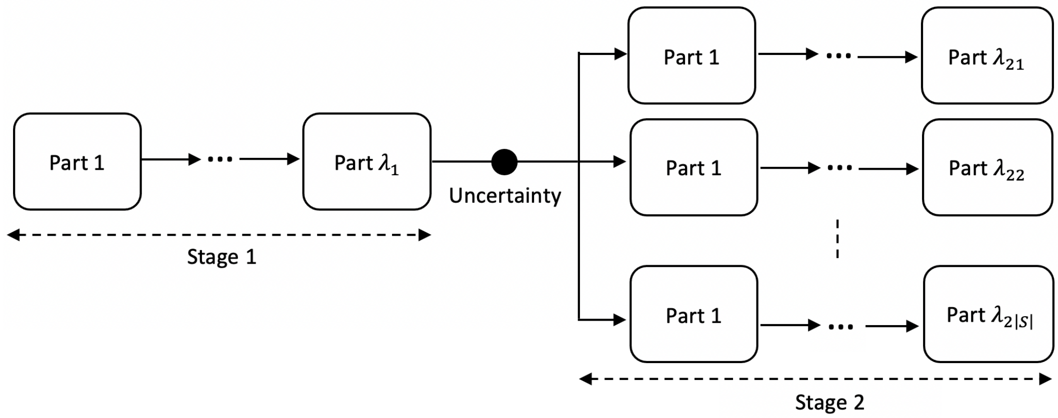


Figure 3: D-SDM approach for the stochastic APP

each stage into several parts in step 4, the assets impacted in each stage need to be sorted based on the opening time windows as written in step 3. The next step is solving each part in stage 1 by calling a function called *Deterministic* as written in step 5. This function is then called to solve each part of each scenario in stage 2 as written in steps 6-10. The last serviced assets from the solution of the last part in stage 1 denoted as D_{λ_1+1} are depots for the first part in each scenario in stage 2.

The function *Deterministic* first calculates node angle θ_i for each node i in the current part that will be used in the objective function. A direction angle is then calculated using step 18 or 20 depending on the stage and part considered. If the current part is the last part in stage 1 (condition in line 17), the direction angle needs to consider all first parts of each scenario in stage 2 (step 18). If the current part is not the last part of any stage (condition in line 19), the direction angle is considered only from the subsequent part (step 20). The current part is then solved with three possible objectives. The objective in step 23 is used if the current part is in stage 1, while the objective in step 25 is used if the current part is not the last part of any scenario in stage 2. The main difference is that if an asset is serviced in stage 2 the asset value is weighted by its probability. The last objective in step 27 is used only if the current part is the last part of any scenario in stage 2. This part is not necessarily solved with minimising the start times and aligning the solution to any direction. The last assets serviced in each part are the starting locations of vehicles for the next part as written in step 29. Any vehicles that are not in service in the current part are included for the next part (step 30).

4.3.2. Deterministic-Stochastic SDM approach

Solving the TSSP model for a large problem in a single run is costly in computing time. To overcome this, the deterministic-stochastic SDM (DS-SDM) approach is implemented by decomposing the problem into pieces. This method then couples both deterministic and stochastic models for the decomposed problem. A stochastic part is composed in between the two stages. It involves the overlapping nodes in the buffer

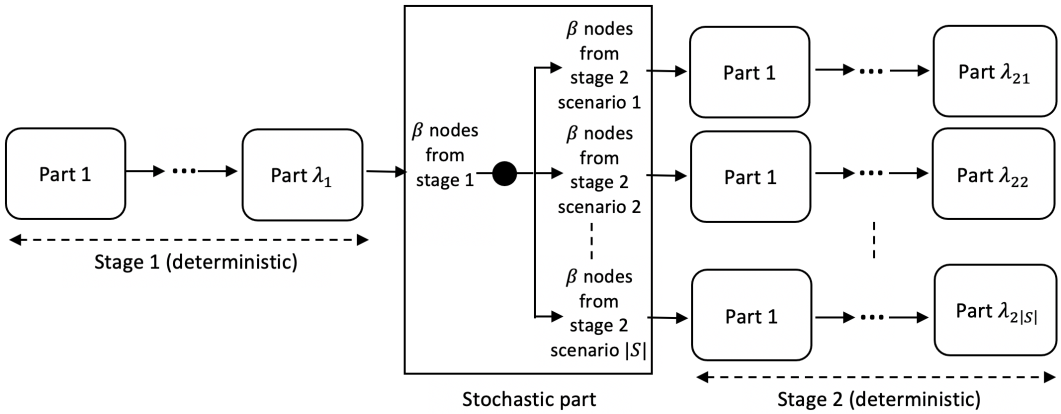


Figure 4: DS-SDM approach for the stochastic APP

zone and β additional nodes taken from each stage around the buffer zone. The rest of parts is solved using the deterministic model. The main idea of this approach is illustrated in Figure 4 and the algorithm is shown in Algorithm 4.

Algorithm 4: DS-SDM approach algorithm

Input: all sets and parameters listed in Section 2,$\alpha, \beta, \lambda_1, \lambda_{2g}$
Output: all decision variables listed in Section 2

1 $D_1 \leftarrow \{0\}$
2 Decompose C into C_1 and C_{2g} for each $g \in S$
3 Sort C_1 and each C_{2g} based on their opening time windows in an ascending order
4 Compose a stochastic part C_s with parameter β
5 Decompose $C_1 \setminus C_s$ into λ_1 parts and $C_{2g} \setminus C_s$ into λ_{2g} parts for each $g \in S$ where set of assets in stage st part h denoted as $C_{st,h}$
6 `Deterministic` $(1, D_1, C_{1,h}, \lambda_1, 0)$
7 Solve the TSSP model for the stochastic part C_s with depots D_{λ_1+1}
8 Extract the last serviced nodes from each scenario g and assign them as D_{2g} for each $g \in S$
9 Involve vehicles that are not in service in each scenario g and their locations into D_{2g} for each $g \in S$
10 $g \leftarrow 1$
11 **while** $g \leq |S|$ **do**
12   `Deterministic` $(2, D_{2g}, C_{2g,h}, \lambda_{2g}, g)$
13     $g \leftarrow g + 1$
14 **end**

Algorithm 4 is generally the same as the D-SDM algorithm. The main difference is the DS-SDM approach composes a stochastic part in step 4 and solves this part in the middle of the two stages as written in step 7. After the stochastic part is composed, the rest of assets in each stage is then split into λ_1 parts for stage 1 and λ_{2g} parts for stage 2 scenario g as written in step 5. The *Deterministic* function called in this algorithm is the same as written in Algorithm 3 with a change in step 18. In this step the D-SDM calculates a direction angle based on information from the first part of each scenario in stage 2. Meanwhile, the DS-SDM utilises information from the assets in the stochastic part C_s.

5. Computational experiments

5.1. Fire spread model

In the APP, the time window of an asset is generated based on the estimated time the fire front will reach that asset. If the wind is blowing during wildfires, the fires propagate in an elliptical spread. The position of a fire front at time t with a constant wind speed can be determined by

$$x = Ft\cos\gamma + Gt \tag{27}$$

$$y = Ht\sin\gamma \tag{28}$$

where F and H are fire speeds for the semi-major and semi-minor axis, respectively, and G represents the moving speed of the centre of the ellipse in the x-direction, while γ is an angular parameter. Eq. (28) can be written as

$$\sin\gamma = \frac{y}{Ht} \tag{29}$$

or

$$\cos\gamma = \frac{(Ht)^2 - y^2}{Ht} \tag{30}$$

Substituting Eq.(30) to Eq.(27) yields

$$x = F\frac{(Ht)^2 - y^2}{H} + Gt \tag{31}$$

Solving Eq.(31) for t yields

$$t = \frac{F\sqrt{F^2y^2 - G^2y^2 + H^2x^2} - GHx}{H(F^2 - G^2)} \tag{32}$$

Given F, G, and H, Eq.(32) calculates the estimated time of the fire front will reach a particular location (x, y). This arrival time is then used to generate the time window of the asset. A time window $[o_i, c_i]$ for an asset i can be defined as

$$c_i = t_i - d_i - 0.5 \tag{33}$$

and

$$o_i = c_i - \sigma \tag{34}$$

where d_i is a service duration required at asset i and σ is the time window length. A constant of 0.5 in Eq. (33) represents that the crews require about half an hour to move away from the fire if the asset is serviced at the latest start time.

Table 1: Data for an illustrative example

i	x_i	y_i	ϑ_i	d_i	o_i	c_i	o_i^a	c_i^a	o_i^b	c_i^b	o_i^c	c_i^c	r_i
0	150	150	0	0	-	-	-	-	-	-	-	-	(0,0)
1	162	107	30	1	4.19	5.19	-	-	-	-	-	-	(1,1)
2	199	213	20	1	-	-	6.78	7.78	7.26	8.26	8.23	9.23	(1,1)
3	125	103	10	1	3.47	4.47	-	-	-	-	-	-	(1,0)
4	230	128	20	1	5.76	6.66	-	-	5.66	6.66	5.76	6.76	(1,1)
5	55	68	10	1	0.92	1.92	-	-	-	-	-	-	(1,1)
6	186	26	20	1	2.77	3.77	-	-	-	-	-	-	(1,1)
7	231	243	30	1	-	-	8.47	9.47	8.11	9.11	8.26	9.26	(1,0)
8	247	132	20	1	-	-	-	-	-	-	6.08	7.08	(1,0)
9	89	61	30	1	1.19	2.19	-	-	-	-	-	-	(1,0)
10	234	146	20	1	-	-	-	-	6.18	7.18	6.16	7.16	(1,1)
11	250	228	30	1	-	-	-	-	-	-	8.19	9.19	(1,1)
12	159	89	10	1	3.44	4.44	-	-	-	-	-	-	(0,1)
13	105	81	40	1	2.22	3.22	-	-	-	-	-	-	(0,1)
14	149	183	20	1	-	-	7.33	8.33	-	-	-	-	(0,1)
15	209	269	40	1	-	-	8.16	9.16	8.39	9.39	-	-	(1,1)
16	212	26	10	1	3.39	4.39	-	-	-	-	-	-	(0,1)
17	152	241	30	1	-	-	8.28	9.28	-	-	-	-	(1,0)
18	222	83	30	1	4.49	5.49	-	-	-	-	-	-	(0,1)
19	136	68	40	1	2.32	3.32	-	-	-	-	-	-	(0,1)
20	150	104	20	1	3.88	4.88	-	-	-	-	-	-	(1,1)
21	212	45	40	1	3.53	4.53	-	-	-	-	-	-	(0,1)
22	232	121	30	1	5.61	6.61	-	-	-	-	5.61	6.61	(1,1)
23	255	132	10	1	-	-	-	-	-	-	6.34	7.34	(1,1)
24	165	113	40	1	4.48	5.48	-	-	-	-	-	-	(1,1)
25	248	41	30	1	4.33	5.33	-	-	-	-	-	-	(1,1)
26	109	110	40	1	3.60	4.60	-	-	-	-	-	-	(1,0)
27	106	71	30	1	1.85	2.85	-	-	-	-	-	-	(1,0)
28	102	76	20	1	1.97	2.97	-	-	-	-	-	-	(1,0)
29	172	225	20	1	-	-	7.20	8.20	8.38	9.38	-	-	(1,0)
30	52	63	10	1	0.65	1.65	-	-	-	-	-	-	(1,1)
31	175	281	20	1	-	-	8.34	9.34	-	-	-	-	(1,1)
32	250	126	20	1	-	-	-	-	-	-	6.04	7.04	(0,1)
33	188	142	10	1	-	-	5.18	6.18	6.02	7.02	-	-	(1,1)
34	59	64	40	1	0.80	1.80	-	-	-	-	-	-	(1,1)
35	235	242	30	1	-	-	-	-	8.20	9.20	8.25	9.25	(0,1)
36	140	214	30	1	-	-	8.46	9.46	-	-	-	-	(1,0)
37	145	184	40	1	-	-	7.59	8.59	-	-	-	-	(0,1)
38	184	277	40	1	-	-	8.13	9.13	-	-	-	-	(1,1)
39	129	77	40	1	2.48	3.48	-	-	-	-	-	-	(1,1)

5.2. *An illustrative example*

We demonstrate an illustrative example showing the performance of each solution technique presented above using a small-size instance. It consists of 39 assets that are at risk during wildfires before and after a change in wind direction as shown in

Table 1. There are three possible times for the wind change. Scenario a, b, and c consider respectively a wind change at 6 (hours from now) with a probability of 0.5, 6.5 (hours from now) with a probability of 0.3, and 7 (hours from now) with a probability of 0.2. As 6 is the earliest possible time for the wind change, it is referred to as the staging time (ST). We use a time limit of 0.15 s in the experiment to see how each algorithm works. The problem illustration for the example is the same as depicted in Figure 1 where the blue, red and green lines refer to the fire front for scenarios a, b and c, respectively.

There are three types of assets: (i) the assets that can only be serviced before ST in stage 1 indicated if their closing time windows are less than ST, (ii) the assets that can only be serviced after ST in stage 2 indicated if their opening time windows are greater than ST, (iii) the assets that can be serviced either in stage 1 or stage 2 indicated if their opening time windows are less than ST but their closing time windows are greater than ST. In this example, there are 22 assets that potentially can be serviced before ST in stage 1, 11 assets impacted by scenario a, 8 assets impacted by scenario b, and 10 assets impacted by scenario c. The example considers two different types of vehicles with only one vehicle of each type available at a depot. The vehicles will be assigned to service the assets with a total expected asset value of 827.

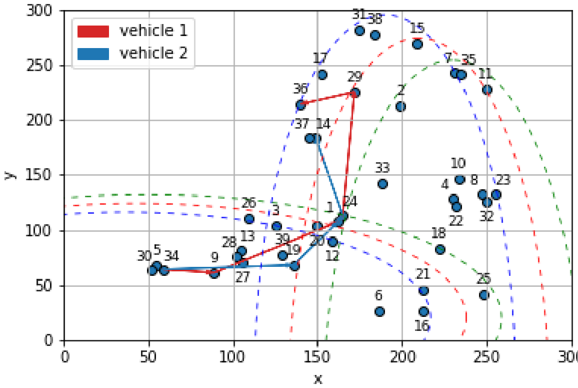
5.2.1. DR approach

By applying Algorithm 1, the solution represented by routes for each vehicle and each scenario can be seen in Figure 5, while the protective service timeline is provided in Table 2. Both vehicles start by going to service node 34 at 0.83, as this asset requires both vehicles, and end up at the same location, i.e. node 24, in stage 1 with the start time of 5.48. In stage 2 vehicle 1 has three options between servicing node 29 at 8.20 for scenario a, node 10 at 6.99 for scenario b, and node 8 at 7.04 for scenario c depending on the realisation of the wind change. Meanwhile, vehicle 2 can go to either node 14, 10, or 35 for scenario a, b or c, respectively.

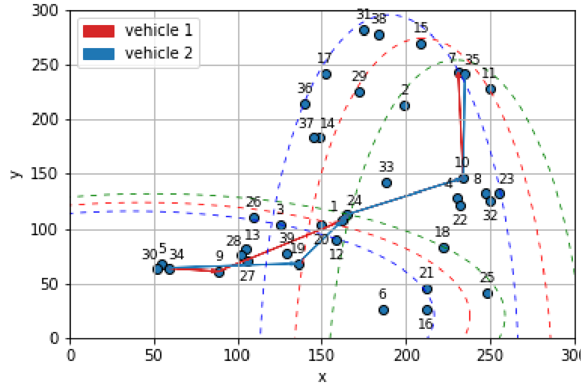
Table 2: Service timeline given by the DR approach

Vehicle 1								Vehicle 2							
Stage 1		Stage 2a		Stage 2b		Stage 2c		Stage 1		Stage 2a		Stage 2b		Stage 2c	
i	s_i	i	s_i	i	s_i	i	s_i	i	s_i	i	s_i	i	s_i	i	s_i
34	0.83	29	8.20	10	6.99	8	7.04	34	0.83	14	8.33	10	6.99	35	9.25
9	2.19	36	9.46	7	9.11	7	9.26	19	2.35			35	9.20		
1	4.19							1	4.19						
24	5.48							24	5.48						

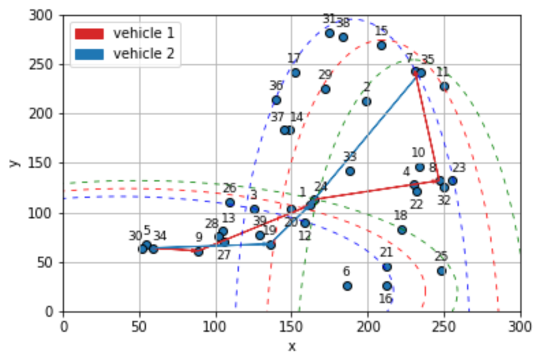
Nodes 34, 9, 1, 19 and 24 are scheduled to be serviced in stage 1 with an expected asset value of 180. In stage 2 expected asset values are 35 for scenario a, 24 for scenario b and 16 for scenario c. Hence the expected value of serviced assets for both stages is 255 which corresponds to 30.83% of the total expected values of assets.



(a) Routes for scenario *a*



(b) Routes for scenario *b*



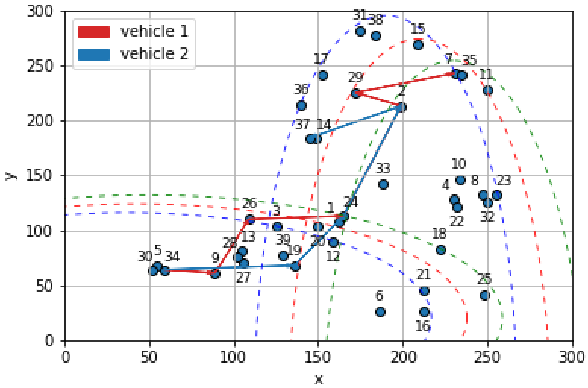
(c) Routes for scenario *c*

Figure 5: Routes of the DR approach

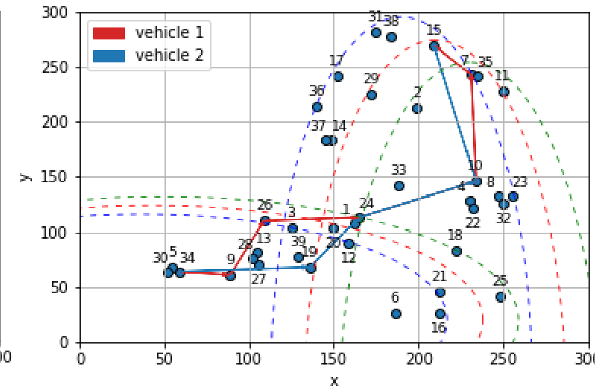
5.2.2. TPD approach

The solution of the first phase is directed towards a direction angle of 47.39°, obtained using Eq. (24). The angle reflects the most promising areas considering the probability of each scenario in the second phase. The TPD approach gives the solution as shown in Figure 6 with the service timeline presented in Table 3. Similar to the DR approach, both vehicles start to service node 34 and end up at node 24 in stage 1. However, the start servicing time at node 24 is earlier than the DR as the TPD minimises the start time. As a result, the feasibility of servicing assets in stage 2 is improved.

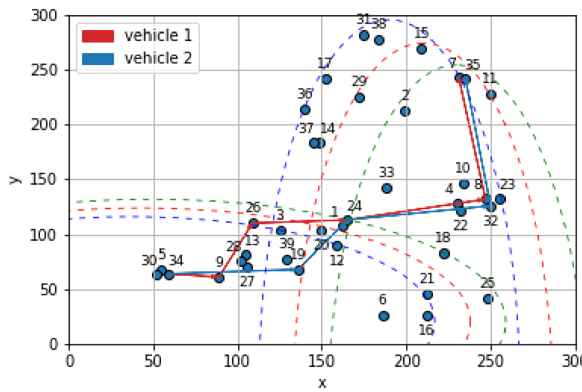
If the wind changes at ST, vehicle 1 will visit node 2 by 6.78 and finish at node 7, otherwise the vehicle will go to node 10 by 6.49 and finish at node 15 for scenario *b* or go to node 8 by 6.54 and finish at 7 for scenario *c*. Meanwhile, vehicle 2 will service nodes 2 and 37 for scenario *a*, nodes 10 and 15 for scenario *b*, or nodes 32 and 35 for scenario *c*. The expected value of serviced assets is 35.31% of the total expected asset values.



(a) Routes for scenario a



(b) Routes for scenario b



(c) Routes for scenario b

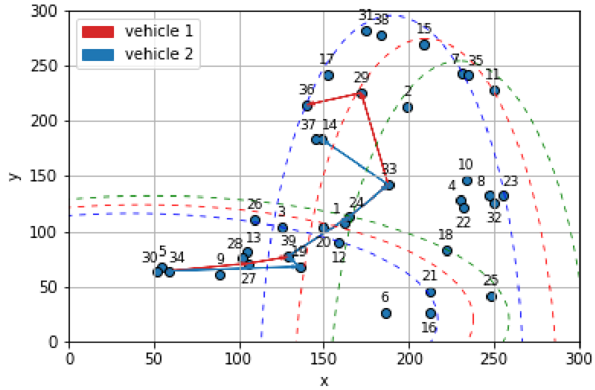
Figure 6: Routes of the TPD approach

Table 3: Service timeline given by the TPD approach

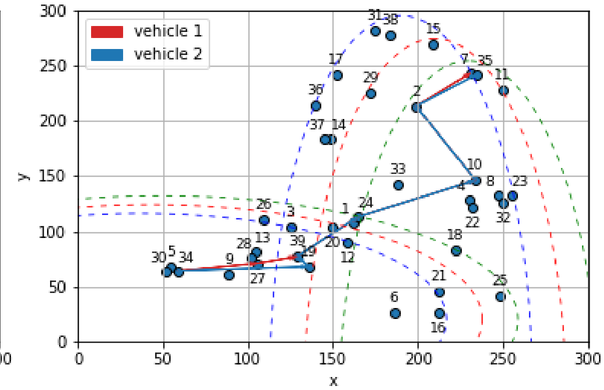
Vehicle 1								Vehicle 2							
Stage 1		Stage 2a		Stage 2b		Stage 2c		Stage 1		Stage 2a		Stage 2b		Stage 2c	
i	s_i	i	s_i	i	s_i	i	s_i	i	s_i	i	s_i	i	s_i	i	s_i
34	0.83	2	6.78	10	6.49	8	6.54	34	0.83	2	6.78	10	6.49	32	6.34
9	2.04	29	7.98	7	8.16	7	9.26	19	2.35	37	8.59	15	9.39	35	9.25
26	3.60	7	9.47	15	9.39			24	4.97						
24	4.97														

5.2.3. D-SDM approach

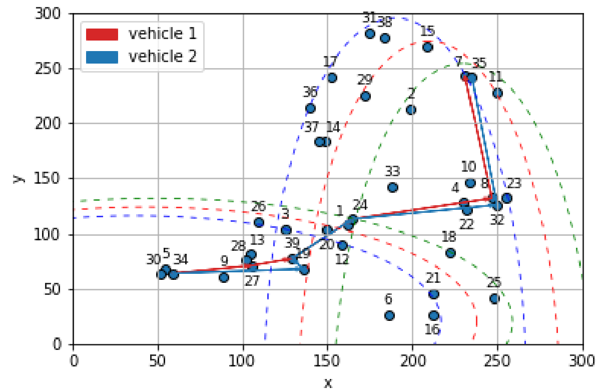
In this example, as the number of nodes in stage 1 is twice as large as the number of assets in stage 2, stage 1 is split into two parts. The first part in stage 1 has a target direction of 26.86° obtained from the assets in the second part. As the last part in stage 1, this second part is solved using a target direction considering the assets in the first part of each scenario in stage 2 weighted by its probability. The target is 47.39° where the direction angles of scenario a, b and c are 52.67°, 45.40° and 37.16°, respectively. Routes for the D-SDM approach are depicted in Figure 7 with the schedule of services



(a) Routes for scenario a



(b) Routes for scenario b



(c) Routes for scenario c

Figure 7: Routes of the D-SDM approach

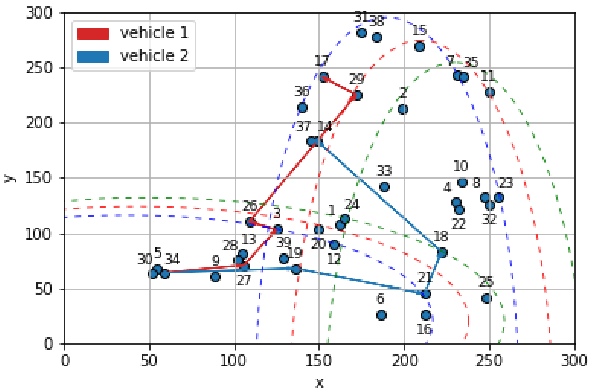
given in Table 4. For this example, the D-SDM gives 36.28% of total expected asset values serviced.

Table 4: Service timeline given by the D-SDM approach

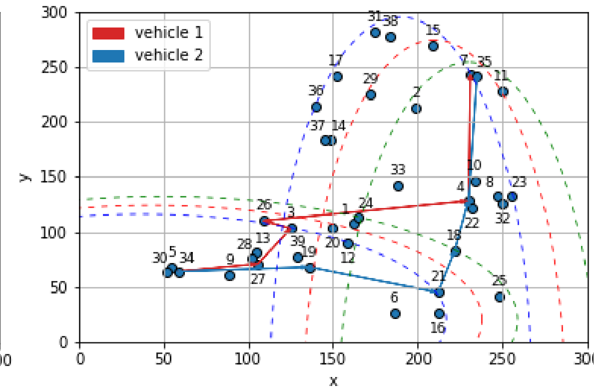
Vehicle 1								Vehicle 2							
Stage 1		Stage 2a		Stage 2b		Stage 2c		Stage 1		Stage 2a		Stage 2b		Stage 2c	
i	s_i	i	s_i	i	s_i	i	s_i	i	s_i	i	s_i	i	s_i	i	s_i
34	0.83	33	6.01	10	6.28	8	6.33	34	0.83	33	6.01	10	6.28	32	6.34
27	2.15	29	8.20	2	7.82	7	9.26	19	2.35	14	7.56	2	7.82	35	9.25
39	3.42	36	9.46	7	9.11			39	3.42	37	8.59	35	9.20		
24	4.77							24	4.77						

5.2.4. DS-SDM approach

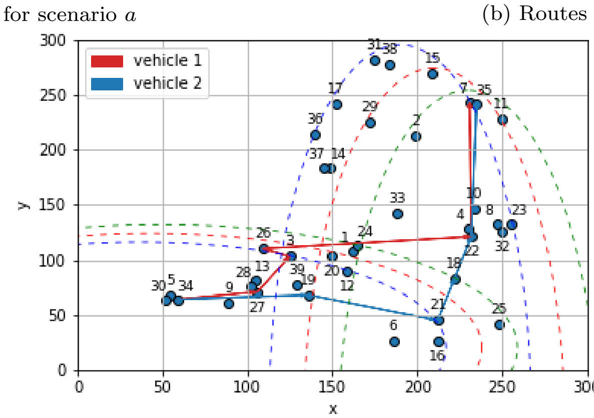
This approach is implemented by setting one deterministic part in each stage with a stochastic part in between. For this example, the stochastic part is composed by the assets in the buffer zone with $\beta = 2$. When solving the deterministic part in stage 1, the target direction is 28.48° obtained from the assets in the stochastic part. When solving the stochastic part, the objective for scenario a is directed towards a target direction of 53.28° obtained from the assets in the last part of scenario a. Meanwhile, for scenario b and c, the target direction is 48.49° and 43.30°, respectively, obtained from the assets in the last part of each scenario b and c. We can see that the target direction for scenario a is steeper than scenario b, as scenario a occurs earlier than b. Similarly, the target direction for scenario b is steeper than c. Routes for the DS-SDM approach is illustrated in Figure 8 with the service timetable provided in Table 5. The figure shows that the DS-SDM prefers to choose node 26 and 18 as the staging locations. The DS-SDM can service 39.54% of the total expected asset values.



(a) Routes for scenario a



(b) Routes for scenario b



(c) Routes for scenario c

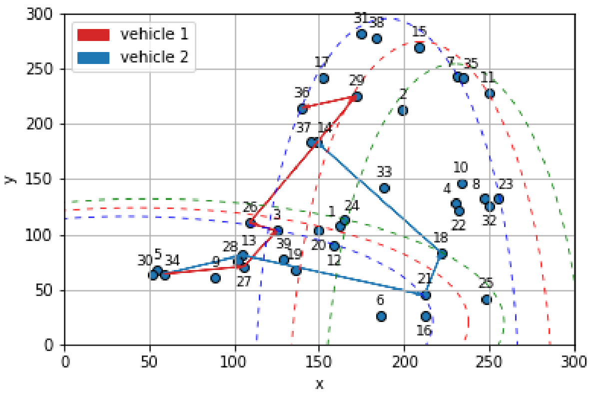
Figure 8: Routes of the DS-SDM approach

5.2.5. *Optimal solution*

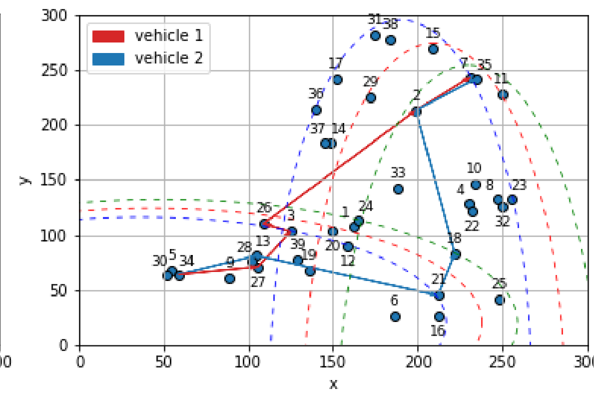
Routes for the optimal solution of the TSSP model obtained by CPLEX is shown in Figure 9 with the service timeline is given in Table 6. It represents 39.78% of the expected asset values serviced in total. The staging locations are 26 and 18 for vehicle 1 and 2, respectively.

Table 5: Service timeline given by the DS-SDM approach

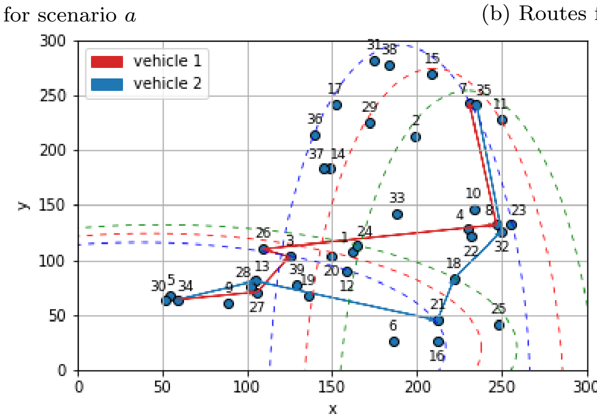
Vehicle 1								Vehicle 2							
Stage 1		Stage 2a		Stage 2b		Stage 2c		Stage 1		Stage 2a		Stage 2b		Stage 2c	
i	s_i	i	s_i	i	s_i	i	s_i	i	s_i	i	s_i	i	s_i	i	s_i
34	0.83	29	7.20	4	6.44	22	6.41	34	0.83	14	7.33	4	6.44	22	6.41
27	2.15	17	9.28	7	9.11	7	9.26	19	2.35	37	8.59	35	9.20	35	9.25
3	3.47							21	3.88						
26	4.59							18	5.14						



(a) Routes for scenario a



(b) Routes for scenario b



(c) Routes for scenario c

Figure 9: Routes of the optimal solution

The complete comparison between the optimal solution and each solution technique used in this paper is given in Table 7.

5.3. Benchmark instances

In the APP, the time window at each asset location is generated based on the expected time of fires arriving at the asset. It is obtained by applying a fire spread model to the assets that are at risk. The ideal benchmark instances have not only various classes of point distribution, but also each instance in the same class has different

Table 6: Service timeline given by the optimal solution

Vehicle 1								Vehicle 2							
Stage 1		Stage 2a		Stage 2b		Stage 2c		Stage 1		Stage 2a		Stage 2b		Stage 2c	
i	s_i	i	s_i	i	s_i	i	s_i	i	s_i	i	s_i	i	s_i	i	s_i
34	0.83	29	8.20	2	7.26	8	7.08	34	0.83	14	7.33	2	7.26	32	7.04
27	2.15	36	9.46	7	9.11	7	9.26	13	2.22	37	8.59	35	9.20	35	9.25
3	3.47							21	3.97						
26	4.59							18	5.38						

Table 7: Results for an illustrative example where P is the percentage of expected asset values protected

Solution technique	Expected value of assets protected					P (%)
	Stage 1	Stage 2a	Stage 2b	Stage 2c	Total	
DR	180	35	24	16	255	30.83
TPD	190	55	27	20	292	35.31
D-SDM	190	60	30	20	300	36.28
DS-SDM	230	55	24	18	327	39.54
Optimal	230	55	24	20	329	39.78

coordinates of assets. As there are no benchmark instances found in the literature matching with this criteria, we generate new sets of benchmark instances. The time windows of each asset are generated for three possible timings of a wind change.

Table 8: Operators for generating nine instances with different point coordinates from a master instance using geometric transformations where Ω indicates the length of both axes

Instance	New coordinates	Operator
1	$(x', y') = (x, y)$	Original point
2	$(x', y') = (-x + \Omega, y)$	Reflection to y-axis
3	$(x', y') = (x, -y + \Omega)$	Reflection to x-axis
4	$(x', y') = (-x + \Omega, -y + \Omega)$	Rotation 180° clockwise
5	$(x', y') = (y, x)$	Reflection to $y = x$
6	$(x', y') = (-y + \Omega, x)$	Rotation 270° clockwise
7	$(x', y') = (y, -x + \Omega)$	Rotation 90° clockwise
8	$(x', y') = (-y + \Omega, -x + \Omega)$	Reflection to $y = -x$
9	$(x', y') = (x + \Omega/2 \mod \Omega, y)$	Translation $\Omega/2$ along x-axis
10	$(x', y') = (x, y + \Omega/2 \mod \Omega)$	Translation $\Omega/2$ along y-axis

The instances are generated from the well-known Homberger and Gehring's instances for the VRP with time windows (Homberger & Gehring, 2005). We only took the coordinates of each point and generated the time windows using our fire spread model. Their instances consist of three different classes, i.e., randomised (R), clustered

(C), and randomised-clustered (RC) point distribution. We selected a problem size of 600 nodes where the points spread over a 300 x 300 Cartesian plane. To reflect a more realistic landscape, the domain is scaled to represent a landscape of 100 x 100 km.

Each instance with the same class in the referred benchmark instances has the same point distribution. Hence we only took an instance from each R, C and RC classes and referred to them as the three master instances. As we need 10 instances with different point coordinates in each class, we then generate other nine instances using some geometric transformations applied to each master instance in each class. This includes rotation, reflection and translation. Each point in the master instance is transformed using operators listed in Table 8 where each operator produces an instance. Hence we have a total of 30 instances.

We then applied our fire spread model to determine which assets are at risk of fire and the time windows of each asset. The fire spreads from the bottom left side of the domain with a speed of 10 km/h along the semi-major axis of an elliptical shape both before and after a wind change. The wind change can happen in three potential timings, i.e., 6.00, 6.30, and 7.00 hours from the time of planning within an 11-hour planning horizon with probabilities of 50%, 30% and 20%, respectively. This yielded 280 out of 600 nodes on average in total impacted by three scenarios and denoted as set 2 instances. To obtain a lower problem size, set 1 instances are obtained by removing the last 30% nodes from each set 2 instance obtaining 197 nodes on average. Hence we have three sets of instances for each problem size. They are denoted as R1, C1 and RC1 for set 1 instances, and R2, C2 and RC2 for set 2 instances.

Other attributes such as asset values and vehicle requirements associated with each asset are randomly generated. Assets value of an asset is a random number in the range of $[1, 5]$ multiplied by 10, while vehicle requirements are randomly taken from a set of $\{(2, 0, 0), (0, 2, 0), (0, 0, 2), (1, 1, 0), (1, 0, 1), (0, 1, 1), (1, 1, 1)\}$. For each problem size, we use two sets of vehicles available at a depot with a uniform vehicle speed of 50 km/h. They are $(6, 6, 6)$ and $(8, 8, 8)$ for the set 1 instances, and $(9, 9, 9)$ and $(12, 12, 12)$ vehicles for the set 2 instances. Our benchmark instances can be accessed through `https://github.com/dn6514/Stochastic-APP`.

5.4. Experimental setup

All experiments are performed on Australia's computational infrastructure called NCI using a single core and thread. We use a commercial solver CPLEX V12.10 for all experiments. As we are working on an emergency problem, a time limit TL representing operational time is needed especially for solving a large problem. We use a time limit of 10 minutes as a reasonable computation time to test the solver's ability in solving the TSSP in a single run. All methods presented in this paper basically utilise decomposition techniques. We determine a time limit for each sub-problem so that the total elapsed time for each solution technique is comparable with around 10 minutes.

The DR approach decomposes the problem into $|S|$ sub-problems where $|S|$ refers to the number of scenarios. As the first sub-problem is larger than other sub-problems, the time limit for the first sub-problem needs to be longer than others. The time limit for the first sub-problem can be determined as $2(TL/(|S|+1))$, while other sub-problems can be set as $TL/(|S|+1)$. In our experiments, we use a time limit of 300 seconds for the first sub-problem, while for the rest of the sub-problems we use a time limit of 150 seconds.

Given $|S|$ scenarios, the TPD approach decomposes the problem into $|S|+1$ sub-problems. The time limit for each sub-problem is then determined as $TL/(|S|+1)$. As we consider three scenarios, the time limit given for each sub-problem is 150 seconds.

The D-SDM and DS-SDM decompose the problem into relatively smaller sub-problems. Solvers may solve some sub-problems to their optimality under a given time limit to each sub-problem. Hence for the D-SDM and DS-SDM, the time limit given to each sub-problem is found by trial and error. The time limit is adjusted so that the total elapsed time is comparable with around 600 seconds. For the D-SDM, we use a time limit of 300 seconds per sub-problem, while for the DS-SDM we use a time limit of 150 seconds for each sub-problem. For some cases, the total elapsed time can be over 600 seconds especially for larger problems.

5.4.1. D-SDM approach

We carried out experiments to find the best parameters of λ representing the number of partitions in each stage. We took 10 instances from set 2 instances with three different classes using (9,9,9) vehicles. They are three instances from each R2 and C2 and four instances from RC2. The results is shown in Table 9. Based on a single factor ANOVA test, the differences are statistically significant with a significance level of 0.05. The table indicates that splitting the stage 1 and 2 assets into 3 and 2 partitions, respectively, has improved the solution quality. This setting represents the characteristic of the

Table 9: Comparison of the percentage of expected asset values serviced using different values of λ per stage for the D-SDM approach

Instance	$\lambda=(4,3)$	$\lambda=(3,2)$	$\lambda=(2,2)$	$\lambda=(2,1)$
r2-01	44.37	46.94	45.96	43.31
r2-02	44.45	49.33	49.33	46.02
r2-03	45.69	49.33	49.09	45.89
c2-01	40.35	42.41	42.41	38.92
c2-02	38.65	40.29	37.00	34.00
c2-03	43.60	46.33	48.97	46.69
rc2-01	44.30	45.66	45.00	42.71
rc2-02	40.40	42.60	40.75	36.63
rc2-03	45.45	50.19	48.66	45.18
rc2-04	43.81	46.17	46.79	42.08
Avg.	43.11	45.93	45.40	42.14

benchmark instances where the number of assets that are potentially impacted in stage

1 is larger than those impacted in each scenario in stage 2.

Furthermore, the above results indicate that decomposing the problem into the right sub-problem size is important. The more partitions we have, the worse solution quality we get. In contrast, CPLEX may find it harder to solve within operational time if the sub-problem size is still large. This is acceptable as the increasing number of partitions can reduce the proximity to the optimal solution. We also found that the complexity of a sub-problem is not only determined by the number of nodes, but also the length of time windows and the number of vehicles. The longer the time windows and the more vehicles considered, the more computational time is required. The time window length determines the flexibility of start times, while the number of vehicles specify the number of decision variables required. Therefore, there is no exact formula to determine the sub-problem size.

5.4.2. DS-SDM approach

Similar to the D-SDM approach, decomposing the problem into appropriate sub-problem size for the DS-SDM approach is a key. In addition to the size for deterministic parts, the size for the stochastic part needs to be determined. We carried out experiments using the same instances as in Table 9 for tuning the number of deterministic parts λ and β additional nodes for the stochastic part. The results provide in Table 10 showing a comparison between $\lambda = (2, 1)$ and $\lambda = (3, 2)$ with various β.

Table 10: Comparison of the percentage of expected asset values serviced using different values of λ and β for the DS-SDM approach

Instance	$\lambda = (2, 1)$			$\lambda = (3, 2)$		
	$\beta = 0$	$\beta = 10$	$\beta = 20$	$\beta = 0$	$\beta = 10$	$\beta = 20$
r2-01	51.42	49.45	49.16	50.49	49.06	46.96
r2-02	53.07	51.63	51.39	50.48	45.95	46.20
r2-03	52.67	52.04	53.36	52.94	49.67	46.89
c2-01	44.87	45.40	44.26	44.97	43.06	42.61
c2-02	42.08	37.59	41.01	41.32	39.63	39.42
c2-03	54.00	53.81	52.89	50.46	48.39	47.63
rc2-01	49.36	48.54	47.94	46.83	46.64	46.56
rc2-02	45.12	45.88	45.40	45.30	43.52	43.43
rc2-03	54.81	56.18	52.53	49.53	50.49	47.21
rc2-04	49.40	51.60	50.53	51.02	48.22	44.30
Avg.	49.68	49.21	48.85	48.33	46.46	45.12

The results show that splitting the deterministic parts with $\lambda = (2, 1)$ is better than $\lambda = (3, 2)$. This may be caused by the sub-problem size generated by $\lambda = (3, 2)$ is too small for solvers and hence it is further away from the best solution that can be achieved. We then test the significance of results produced with various β and $\lambda = (2, 1)$. Based on a single factor ANOVA test, the differences are statistically not significant with a significance level of 0.05. This shows that considering the overlapping assets in the buffer zone for the stochastic part with or without β additional assets from each stage is not practically significant. This means that we can take any value of β. Furthermore, the existence of a stochastic part has improved the solution produced by the D-SDM approach provided in Table 9. This means that solving the TSSP model around uncertainty plays a key role in handling the uncertainty. We then tried to analyse the significance of means obtained by each problem class, i.e. clustered, randomised, and randomised-clustered. A further single factor ANOVA test shows that the means obtained by all problem characteristics are not statistically significant. This means that the above analysis applies for all problem characteristics.

5.5. Numerical results

In this section we carry out experimental results comparing the performance of each solution technique presented in this paper on our benchmark instances. First we solve the TSSP model given in Section 3 using CPLEX to test the solver's ability in a single run with a given time limit of 10 minutes. The resulting lower bound (LB) is the best solution achieved by the solver and is presented as LB (10m) in the result tables. We then evaluate the performance of the proposed DS-SDM approach and other solution techniques with comparable computation time. The performance of each solution technique is indicated by the percentage of expected asset values serviced, denoted as P(%). While the optimal solution of each instance is unknown, we run CPLEX to solve the TSSP in a single run with a longer time limit of an hour to improve the upper bound (UB) of feasible solution. This UB (1h) is used to measure the solution quality obtained by each solution technique.

Table 11: Comparison of P(%) for set 1 instances with (6,6,6) vehicles

Instance	LB (10m)	DR	TPD	D-SDM		DS-SDM		UB (1h)
	P(%)	P(%)	P(%)	P (%)	T (s)	P (%)	T (s)	
R1	16.99	35.38	41.47	45.73	22.67	51.85	550.02	78.47
C1	9.90	33.43	36.79	41.76	67.17	47.72	510.94	75.05
RC1	13.16	33.97	38.59	43.42	43.70	49.66	529.06	75.49
Avg.	13.35	34.26	38.95	43.64	44.51	49.74	530.00	76.34

For the set 1 instances, the DS-SDM outperforms other solution techniques with significant improvements as shown in Tables 11 and 12. The best solution achieved by CPLEX under a time limit of 600 s, indicated by LB (10m), is around 13% of total expected asset values either for (6,6,6) vehicles or for (8,8,8) vehicles. Compared with LB (10m), the DR approach shows an increase of around 7% from 34.26% for (6,6,6) vehicles to 41.64% for (8,8,8) vehicles. Furthermore, as the number of vehicles increases from (6,6,6) to (8,8,8), the TPD approach shows a higher increase of more than 9% from 38.95% to 48.57%. Meanwhile, the D-SDM and DS-SDM give more increases in

Table 12: Comparison of P(%) for set 1 instances with (8,8,8) vehicles

Instance	LB (10m)	DR	TPD	D-SDM		DS-SDM		UB (1h)
	P(%)	P(%)	P(%)	P (%)	T (s)	P (%)	T (s)	
R1	18.51	44.11	50.08	57.89	76.54	65.26	579.54	90.48
C1	10.73	39.38	45.56	53.49	161.30	60.05	592.41	86.96
RC1	11.36	41.44	50.08	54.77	76.81	62.37	601.00	87.72
Avg.	13.54	41.64	48.57	55.39	104.88	62.56	590.98	88.39

solution quality by 11.75% and 12.82%, respectively, with additional two vehicles for each type. This indicates that the DS-SDM is better at handling an increase in the number of vehicles to improve the solution quality.

Furthermore, the DS-SDM produces higher quality solutions than other solution techniques. This is indicated that the P(%) DS-SDM is closer to the UB (1h) than other methods reflecting the proximity to the optimal solution. The DS-SDM obtains a gap of around 26% for both (6,6,6) and (8,8,8) vehicles. The D-SDM is in second place with a gap of around 33%. Third place is for the TPD with a gap of 37.39 for (6,6,6) vehicles and 39.82% for (8,8,8) vehicles. The DR obtains a gap of 42.08% for (6,6,6) vehicles and 46.75% for (8,8,8) vehicles. A huge gap is shown by the LB (10m) compared to the UB (1h) with a gap of 63% and 74.85% for (6,6,6) and (8,8,8) vehicles, respectively. Note that the increase in the number of vehicles leads to an increase in feasible solutions. Hence the UB (1h) increases from Table 11 to Table 12.

Table 13: Comparison of P(%) for set 2 instances with (9,9,9) vehicles

Instance	LB (10m)	DR	TPD	D-SDM		DS-SDM		UB (1h)
	P(%)	P(%)	P(%)	P (%)	T (s)	P (%)	T (s)	
R2	1.86	33.65	37.72	48.97	568.09	53.10	647.64	93.26
C2	2.01	30.95	31.75	44.38	661.26	49.30	676.75	89.49
RC2	0.40	30.01	36.72	46.56	733.11	51.57	661.60	91.16
Avg.	1.43	31.53	35.40	46.64	654.15	51.32	662.00	91.31

Table 14: Comparison of P(%) for set 2 instances with (12,12,12) vehicles

Instance	LB (10m)	DR	TPD	D-SDM		DS-SDM		UB (1h)
	P(%)	P(%)	P(%)	P (%)	T (s)	P (%)	T (s)	
R2	0.95	39.33	47.46	62.24	767.55	66.06	675.02	98.10
C2	0.86	34.00	40.55	56.95	878.15	61.58	709.48	95.79
RC2	0.39	34.97	42.35	59.13	841.91	63.91	702.15	97.21
Avg.	0.73	36.10	43.45	59.44	829.20	63.85	695.55	97.03

Tables 13 and 14 provide results on set 2 instances with two different number of vehicles. Compared to set 1 instances, the problem size increases from 197 nodes on average in set 1 instances to 280 nodes on average in set 2 instances. In addition to the problem size, the number of vehicles also increases to handle the increase in the number of nodes. While CPLEX with a single run suffers for this set 2 instances, the proposed DS-SDM approach yields reasonable quality solutions and still outperforms other methods especially the D-SDM approach. The gap between the P(%) DS-SDM and the UB (1h) is 40% for (9,9,9) vehicles and 33.18% for (12,12,12) vehicles. As the main competitor, the D-SDM resulted in gaps of 44.67% and 37.59% for (9,9,9) and (12,12,12) vehicles, respectively. These huge gaps might be influenced by the quality of upper bound obtained by CPLEX after an hour for these larger problems.

Looking at the results for each class of instances, the clustered distribution seems to be the hardest class of problems to solve, while the randomised distribution gets easier to solve. For instance, if we compare the performance of each solution technique for each class in Table 11, C1 received the lowest P(%), followed by RC1 and R1. The main factor influencing this difference is the density of points. The points in C1 are denser than RC1, while RC1 as a combination of randomised and clustered distribution is denser than R1. This density factor leads to an increase in feasible solutions. As the points are close to one another in the clustered class, each vehicle visiting a particular node will have more options to visit the subsequent node. Hence the denser the points are, the harder the problem to solve.

In addition to the percentage of expected asset values serviced, the solution provides the itinerary of each vehicle with the visit time at each location. This informs the Incident Management Team (IMT) in response to wildfires on which assets should be serviced and which scenario should be taken when the wind changes. The detailed results can be seen at https://github.com/dn6514/Stochastic-APP.

6. Conclusion

During uncontrollable wildfires, an IMT is responsible for planning the optimal deployment of resources to service various community assets, and thus reduce the risk of their destruction. In this paper we have extended a method used for the solution of the deterministic APP to handle a stochastic variant where a wind change is forecast with uncertainty in the timing. The timing affects which assets are subject to the wildfire hazard and hence the routing and scheduling of the resources under the control of the IMT. A standard formulation and solution procedure cannot solve a realistic size of this type of problem within the times and accuracy required for operational purposes.

In this work we have developed a method that is capable of solving this problem with multiple scenarios of the timing of a wind change. Benchmark tests show that the proposed method meets operational requirements. The method proposed is relatively easy to implement. Once the problem has been divided into appropriate components, a commercial solver can be used to solve each of these components. The main challenge lies in coupling the components appropriately. Moreover, larger scale problems simply require division into further components each of which are chosen to be well within the capabilities of the solver to solve within the times required. Thus, a change in scale is much easier to accommodate than most heuristic methods for this type of problem.

The method proposed was shown to handle uncertainty in the timing of a wind change. This is an important issue, and one of the most challenging, in south-eastern Australia and elsewhere. The method, however, is capable of dealing with other uncertainties that arise in the APP. For example, uncertainty associated with changes in either or both wind speed and wind direction. Once the new time windows for each asset have been determined for each scenario, the rest of the solution procedure remain unchanged. Uncertainty in wind speed may lead to uncertain magnitude of impacted areas. This will also affect the time windows of assets impacted. Furthermore, while the proposed DS-SDM approach has shown to be efficient to handle large-scale problems, finding a more efficient approach is an interesting future research direction. A stochastic version of ALNS, for instance, may be proposed to solve the stochastic APP studied in this paper. A feasible solution for the stochastic ALNS can be produced by constructing routes for each possible scenario considered, where the assets serviced after the staging time are independent of the scenario chosen.

Acknowledgement

DN gratefully acknowledges Ridlo W. Wibowo for valuable discussion on the fire spread model. DN is supported by the 5000 Doctors Program of the Ministry of Religious Affairs Indonesia.

References

Afifi, S., Dang, D.-C., & Moukrim, A. (2016). Heuristic solutions for the vehicle routing problem with time windows and synchronized visits. *Optim Lett*, *10*, 511–525. doi:10.1007/s11590-015-0878-3.

Aringhieri, R., Bruni, M., Khodaparasti, S., & van Essen, J. (2017). Emergency medical services and beyond: Addressing new challenges through a wide literature review. *Comput Operat Res*, *78*, 349–368. doi:10.1016/j.cor.2016.09.016.

Bodaghi, B., Palaneeswaran, E., & Abbasi, B. (2018). Bi-objective multi-resource scheduling problem for emergency relief operations. *Prod Plan Control*, *29*, 1191–1206. doi:10.1080/09537287.2018.1542026.

Bodaghi, B., Palaneeswaran, E., Shahparvari, S., & Mohammadi, M. (2020). Probabilistic allocation and scheduling of multiple resources for emergency operations; a victorian bushfire case study. *Comp Environ Urban Syst*, *81*, 101479. doi:10.1016/j.compenvurbsys.2020.101479.

Bredström, D., & Rönnqvist, M. (2008). Combined vehicle routing and scheduling with temporal precedence and synchronization constraints. *Eur J Operat Res*, *191*, 19–31. doi:10.1016/j.ejor.2007.07.033.

Bureau of Meteorology (2020). Water and the land: Forecast wind. http://www.bom.gov.au/watl/wind/forecast.shtml. Accessed: 2020-02-27.

Cruz, M., Sullivan, A., Gould, J., Sims, N., Bannister, A., Hollis, J., & Hurley, R. (2012). Anatomy of a catastrophic wildfire: The Black Saturday Kilmore East fire in Victoria, Australia. *For Ecol Manag*, *284*, 269-285. doi:10.1016/j.foreco.2012.02.035.

Drexl, M. (2012). Synchronization in vehicle routing - A survey of VRPs with multiple synchronization constraints. *Transp Sci*, *46*, 297–316. doi:10.1287/trsc.1110.0400.

Farahani, R. Z., Lotfi, M., Baghaian, A., Ruiz, R., & Rezapour, S. (2020). Mass casualty management in disaster scene: A systematic review of OR&MS research in humanitarian operations. *Eur J Operat Res*, *287*, 787–819. doi:10.1016/j.ejor.2020.03.005.

Garrido, R. A., Lamas, P., & Pino, F. J. (2015). A stochastic programming approach for floods emergency logistics. *Transp Res Part E: Logist Transp Rev*, *75*, 18–31. doi:10.1016/j.tre.2014.12.002.

Grass, E., & Fischer, K. (2016). Two-stage stochastic programming in disaster management: A literature survey. *Surv Operat Res Manag Sci*, *21*, 85 – 100. doi:10.1016/j.sorms.2016.11.002.

Gunawan, A., Lau, H. C., & Vansteenwegen, P. (2016). Orienteering problem: A survey of recent variants, solution approaches and applications. *Eur J Operat Res*, *255*, 315-332. doi:10.1016/j.ejor.2016.04.059.

Hanafi, S., Mansini, R., & Zanotti, R. (2020). The multi-visit team orienteering problem with precedence constraints. *Eur J Operat Res*, *282*, 515–529. doi:10.1016/j.ejor.2019.09.045.

Homberger, J., & Gehring, H. (2005). A two-phase hybrid metaheuristic for the vehicle routing problem with time windows. *Eur J Operat Res*, *162*, 220–238. doi:10.1016/j.ejor.2004.01.027.

Jamali, A., Ranjbar, A., Heydari, J., & Nayeri, S. (2021). A multi-objective stochastic programming model to configure a sustainable humanitarian logistics considering deprivation cost and patient severity. *Annals Operat Res*, (pp. 1–36).

Li, Y., & Chung, S. H. (2019). Disaster relief routing under uncertainty: A robust optimization approach. *IISE Transactions*, *51*, 869–886. doi:10.1080/24725854.2018.1450540.

Lianes, I. M., Noreng, M. T., Fagerholt, K., Slette, H. T., & Meisel, F. (2021). The aquaculture service vessel routing problem with time dependent travel times and synchronization constraints. *Comput Operat Res*, *134*, 105316. doi:10.1016/j.cor.2021.105316.

Lin, S.-W., & Yu, V. F. (2017). Solving the team orienteering problem with time windows and mandatory visits by multi-start simulated annealing. *Comput Ind Eng*, *114*, 195–205. doi:10.1016/j.cie.2017.10.020.

Liu, W., Dridi, M., Fei, H., & El Hassani, A. H. (2021). Hybrid metaheuristics for solving a home health care routing and scheduling problem with time windows, synchronized visits and lunch breaks. *Expert Syst Appl*, *183*, 115307. doi:10.1016/j.eswa.2021.115307.

van der Merwe, M., Minas, J. P., Ozlen, M., & Hearne, J. W. (2015). A mixed integer programming approach for asset protection during escaped wildfires. *Can J For Res*, *45*, 444-451. doi:10.1139/cjfr-2014-0239.

van der Merwe, M., Ozlen, M., Hearne, J. W., & Minas, J. P. (2017). Dynamic rerouting of vehicles during cooperative wildfire response operations. *Ann Operat Res*, *254*, 467-480. doi:10.1007/s10479-017-2473-8.

Nuraiman, D., Ozlen, M., & Hearne, J. (2020). A spatial decomposition based math-heuristic approach to the asset protection problem. *Operat Res Perspect*, *7*, 100141. doi:10.1016/j.orp.2020.100141.

Roozbeh, I., Hearne, J., Abbasi, B., & Ozlen, M. (2018a). A two-stage stochastic approach for the asset protection problem during escaped wildfires with uncertain timing of a wind change. arXiv:1811.03272 unpublished.

Roozbeh, I., Ozlen, M., & Hearne, J. (2018b). An adaptive large neighbourhood search for asset protection during escaped wildfires. *Comput Operat Res*, *97*, 125-134. doi:10.1016/j.cor.2018.05.002.

Saeedvand, S., Aghdasi, H. S., & Baltes, J. (2020). Novel hybrid algorithm for team orienteering problem with time windows for rescue applications. *Appl Soft Comput*, *96*, 106700. doi:10.1016/j.asoc.2020.106700.

Shahparvari, S., Abbasi, B., Chhetri, P., & Abareshi, A. (2019). Fleet routing and scheduling in bushfire emergency evacuation: A regional case study of the black saturday bushfires in australia. *Transp Res Part D: Transp Environ*, *67*, 703–722. doi:10.1016/j.trd.2016.11.015.

Stavropoulou, F., Repoussis, P., & Tarantilis, C. (2019). The vehicle routing problem with profits and consistency constraints. *Eur J Operat Res*, *274*, 340–356. doi:10.1016/j.ejor.2018.09.046.

Talarico, L., Meisel, F., & Sörensen, K. (2015). Ambulance routing for disaster response with patient groups. *Comput Operat Res*, *56*, 120–133. doi:10.1016/j.cor.2014.11.006.

Tofighi, S., Torabi, S., & Mansouri, S. (2016). Humanitarian logistics network design under mixed uncertainty. *Eur J Operat Res*, *250*, 239–250. doi:10.1016/j.ejor.2015.08.059.

Usanov, D., van de Ven, P., & van der Mei, R. (2020). Dispatching fire trucks under stochastic driving times. *Comput Operat Res*, *114*, 104829. doi:10.1016/j.cor.2019.104829.

Zhao, X., Ji, K., Xu, P., wen Qian, W., Ren, G., & nian Shan, X. (2020). A round-trip bus evacuation model with scheduling and routing planning. *Transp Res Part A: Policy Pract*, *137*, 285–300. doi:10.1016/j.tra.2020.05.008.