Valid inequalities for the dynamic asset protection problem

Quentin Peña, Mehdi Serairi, Aziz Moukrim

► To cite this version:

Quentin Peña, Mehdi Serairi, Aziz Moukrim. Valid inequalities for the dynamic asset protection problem. 18th International Workshop on Project Management and Scheduling (PMS 2022), European Working Group on Project Management and Scheduling, Apr 2022, Ghent, Belgium. hal-04047936

HAL Id: hal-04047936
https://hal.utc.fr/hal-04047936

Submitted on 27 Mar 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Valid inequalities for the dynamic asset protection problem

Quentin Peña, Mehdi Serairi and Aziz Moukrim

Université de technologie de Compiègne, CNRS, Heudiasyc (Heuristics and Diagnosis of Complex Systems), CS 60 319 - 60 203 Compiègne Cedex
quentin.pena@hds.utc.fr

Keywords: team orienteering problem, bi-objective, synchronization, dynamic, wildfire

1 Introduction

In the recent years, we have seen a surge in the number and strength of wildfires around the globe. These events, whether natural or caused by human activities, can damage the wildlife, as well as people and infrastructures. When a wildfire breaks, Incident Management Teams (IMTs) need to dispatch their resources to contain the fire, evacuate the people, and protect community assets (hospitals, bridges, schools, etc.).

In this paper, we will focus more particularly on vehicle routing for asset protection. IMTs need to assign an heterogeneous fleet of vehicles to the different community assets to carry out preventive actions. Such actions include wetting the facade of a building or removing fuel material, for example. These actions effectively mitigate the damages if they are accomplished within a specified time window, not too soon or too late. Their complexity often requires the cooperation and synchronization of multiple teams and vehicles.

Unfortunately, the behaviour of wildfires and its consequences are hard to predict. Changes in wind conditions, road closures due to fallen trees, vehicles breakdowns, can make the vehicles' routes obsolete. IMTs need to react to any disruption by updating the routes of the vehicles. The new routes must protect as many assets as possible, but there is an incentive to limit the deviation from the initial routes, as some actions may require specific preparation and to limit communication issues.

The problem of routing vehicles for asset protection during a wildfire was presented as the Asset Protection Problem during escaped wildfire (APP) in Van der Merwe *et. al.* (2015). Van der Merwe *et. al.* (2017) defines the dynamic APP that aims at rerouting vehicles after a disruption occurs. As the dynamic APP is a bi-objective problem, the solution is a trade-off surface called Pareto front. The Pareto front is described by a set of solutions such that there is no feasible solution that strictly improves one objective without degrading the second. We say that these solutions are non-dominated. Multiple heuristic approaches have been studied for the mono-objective version of the APP ((Roozbeh *et al.* 2018), (Nuraiman *et. al.* 2020), (Yahiaoui *et al.* 2021)).

Our work focuses on improving the resolution of the optimal Pareto front for the dynamic APP by finding good valid inequalities that rely on the bi-objective or dynamic nature of our problem.

2 Problem presentation

The first mathematical formulation of the APP was proposed by Van der Merwe *et al.* (2015). The authors modeled the APP as a Synchronized Team Orienteering Problem with Time Windows (STOPTW). An instance is defined as a graph $G = (V, A)$. The vertices V represent the depots and the assets we seek to protect. Each asset has a resource requirement defined as a vector of integers. Resources are non-consumable. Each asset also has a time window in which the protection action must start. To carry out the protection actions, a set of heterogeneous vehicles is available, each having a capability vector. The aim is to assign the assets to the routes of vehicles, such that the total protected value is maximized. An asset is protected if the cumulative capability vectors of the vehicles assigned to the asset covers the resource requirements of the asset, and if the protection action starts within the time window of the asset. The protection action at an asset can start only when every vehicle assigned to the asset has arrived (synchronization).

Van der Merwe *et al.* (2017) extended the mathematical formulation of the APP to the dynamic APP. On top of the assets and vehicles, the dynamic APP is based on initial routes for the vehicles before the disruption occurred. The objective is to maximize the total protected value while minimizing the deviation from the initial routes. The deviation can be represented as the number of asset/vehicle reassignments, *i.e.* for each vehicle the number of assets added to or removed from its pre-disruption route.

An important property of the dynamic APP is that an asset that does not participate to the total protected value can be visited outside of its time window. Assets that are not protected in the solution do not have to be removed from the routes of any vehicle, hence entailing no deviation. In other words, an asset that does not improve the first objective (total protected value) does not degrade the second one (deviation). It is thus always possible to build a solution with null deviation from the pre-disruption routes, even if a route became infeasible due to the disruption.

3 Valid Inequalities

We studied the problem to deduce valid inequalities based on properties specific to its bi-objective and dynamic nature. We will first present sets of valid inequalities that bound the deviation implied by the protection of an asset, based on their resource requirements. We will then present incompatibility between assets based on their time window, and deduce a set of valid inequalities. We will finally present a set of valid inequalities that combine incompatibility between assets and resource requirements.

3.1 Deviation-based inequalities

In a non-dominated solution, if an asset has been added to the route of at least one vehicle, the asset is protected in the solution. Otherwise, we could construct a feasible solution with the same protected value but strictly lower deviation (hence dominating our original solution) by simply not adding this asset to the route of the vehicle. Given upper and lower bounds $ub_v^+(i)$ and $lb_v^+(i)$ on the number of vehicles required to protect asset i, we have that:

$$ub_v^+(i)Y_i \geq \sum_{p \in \mathcal{P}} Z_{ip}^+ \geq lb_v^+(i)Y_i \tag{1}$$

where Y_i is the decision variable representing the protection status of asset i, $\mathcal{P}$ the set of available vehicles, and Z_{ip}^+ the decision variable representing the addition of asset i to the route of vehicle p.

We compute values for these bounds based on the resource requirement of the asset and the capability vector of the available vehicles.

The lower bound $lb_v^+(i)$ is the minimum number of vehicles required to cover the resource requirement of asset i. We can compute this bound by solving a Mixed Integer Program (MIP) for each asset, that can be written as a multi-dimensional knapsack problem. The size of the MIP is small enough to efficiently compute the value of the lower bound in spite of the problem being NP-hard.

The upper bound $lb_v^+(i)$ is the minimum number of vehicles required to cover the resource requirement of asset i without adding a redundant vehicle. A vehicle is redundant if the resource requirement of the asset is still covered if the vehicle is not considered. Thus, adding a redundant vehicle to the protection of an asset does not improve the total protected value but increases the deviation. We can compute this bound by solving a MIP for each asset. We believe this problem to be NP-hard, but the size of the MIP is small enough for the computation to be efficient.

Similar bounds are computed for the number of removals from routes for asset i.

3.2 Incompatibility/vehicle clique inequalities

Two assets i and j are *incompatible/vehicle* for vehicle p if vehicle p cannot visit asset i and asset j within their respective time windows. In other words, if vehicle p visits one of the asset at the opening of its time window, it always reaches the second asset after the closing of its time window.

Let $G_p^{inc/v}$ be the graph of incompatibilities between assets for vehicle p. Each node of this graph is an asset of our problem. There is an edge between two nodes i and j if assets i and j are incompatible/vehicle. A clique is a subset of nodes in an undirected graph that are pairwise adjacent. Vehicle p can visit at most one of the asset of the clique within their time window, which is necessary for the asset to be protected. Thus, we define valid inequalities to ensure that at most one asset is protected among the assets of the clique visited by vehicle p.

3.3 Incompatibility/solution clique inequalities

We call two assets i and j *incompatible/solution* if assets i and j are incompatible/vehicle for every available vehicle. Consider $\mathcal{C}$ a clique of incompatible/solution assets. Then, each vehicle can be used for the protection of at most one asset of $\mathcal{C}$.

Let $ub(\mathcal{C})$ be the maximum number of assets in $\mathcal{C}$ that can be protected using every vehicle at most once, based on the resource requirements of the assets. We thus have that:

$$\sum_{i \in \mathcal{C}} Y_i \leq ub(\mathcal{C}) \tag{2}$$

We can compute $ub(\mathcal{C})$ by solving a MIP for each clique $\mathcal{C}$. The problem is NP-hard, but the size of the MIP is small enough for the computation to be efficient.

4 Results

We generated 10 test instances following the specifications of Van der Merwe *et al.* (2015). We implemented the model in Julia. We solved the model with CPLEX 12.8, on a computer with an Intel Core i7-8550U processor and 8GB of RAM.

For every instance, we generated the extreme point allowing for maximum deviation for three different vehicle breakdowns, with a time limit of 1800 seconds. Table 1 shows the average solve time to compute this extreme point for instances based on which valid inequalities were added to the model. The number of instances solved within the time limit is shown in parenthesis.

We greatly improved the model when introducing our valid inequalities. Incompatibility/vehicle inequalities yield the best result: we solved 6 more instances with 50 assets and 5 more with 60 assets within the time limit. On average, we reduced by a third the solve time for the 4 instances solved to optimality with the initial model.

Table 1. Generation of one extreme point: solve time in seconds, number of optimal solutions

	Instance size (number of assets)			
Valid inequalities	30	40	50	60
None	42 (30/30)	29 (20/30)	58 (09/30)	458 (04/30)
Deviation	22 (30/30)	27 (20/30)	198 (09/30)	268 (03/30)
Inc/veh	1.4 (30/30)	22 (25/30)	267 (15/30)	841 (09/30)
Inc/sol	12 (30/30)	28 (24/30)	37 (11/30)	474 (04/30)
All above	1.2 (30/30)	61 (26/30)	252 (18/30)	518 (11/30)

Deviation-based inequalities greatly improve the relaxation of the model. However, as we compute the extreme point with highest deviation, the introduction of these inequalities alone did not seem to have an impact on the solve time. But, when combined with incompatibility/vehicle inequalities, they solved two more instances than incompatibility/vehicle inequalities alone, and almost halved the solve time for the 9 instances already solved within the time limit.

5 Conclusion

We deduced from the properties and structure of our problem valid inequalities that improved the resolution of our model. We improved the viability of using this model to evaluate the quality of the choices made by IMTs in retrospect. Additional inequalities or changes to the model would be needed to solve large-size instances more consistently.

However using this model for providing in real time the exact Pareto front when a disruption occurs seems out of reach. We would need to consider a heuristic approach to obtain a good approximation of the Pareto front in reasonable time.

Acknowledgements

This study was carried out within the framework of GEOSAFE (Geospatial Based Environment For Optimization Systems Addressing Fire Emergencies).

References

Nuraiman, D., Ozlen, M. and Hearne, J., 2020. A spatial decomposition based math-heuristic approach to the asset protection problem. *Operations Research Perspectives*, 7, p. 100141.

Ozlen, M. and Azizoglu, M., 2009. Multi-objective integer programming: A general approach for generating all non-dominated solutions. *European Journal of Operational Research*, 199(1), pp. 25-35.

Roozbeh, I., Ozlen, M. and Hearne, J., 2018. An Adaptive Large Neighbourhood Search for asset protection during escaped wildfires. *Computers & Operations Research*, 97, pp. 125-134.

Van der Merwe, M., Minas, J., Ozlen, M. and Hearne, J., 2015. A mixed integer programming approach for asset protection during escaped wildfires. *Canadian Journal of Forest Research*, 45(4), pp. 444-451.

Van der Merwe, M., Ozlen, M., Hearne, J. and Minas, J., 2017. Dynamic rerouting of vehicles during cooperative wildfire response operations. *Annals of Operations Research*, 254(1-2), pp. 467-480.

Yahiaoui, A-E., Moukrim, A. and Serairi, M., 2021. GRASP-ILS and set cover hybrid heuristic for the synchronized team orienteering problem with time windows. *International Transactions in Operational Research* Preprint, hal-03110595v3f.