Fair Path Planning for Unmanned Aerial Vehicles (UAVs) in Emergency Response Missions

Germán Pardo González [a, b], Camilo Gómez [a], and Carlos Francisco Rodríguez [b]

[a] Department of Industrial Engineering, Universidad de los Andes, Bogotá, Colombia.

[b] Department of Mechanical Engineering, Universidad de los Andes, Bogotá, Colombia.

gr.pardo@uniandes.edu.co, gomez.ch@uniandes.edu.co, crodrigu@uniandes.edu.co

Abstract

The development of autonomous systems, such as Unmanned Aerial Vehicles (UAVs), has witnessed significant growth in recent years. Their deployment in emergency response missions has become increasingly crucial due to their potential benefits, such as faster response times. However, these benefits also raise ethical considerations that must be addressed when utilizing them in emergency scenarios.

This work aims to contribute to the fair automation of UAVs in disaster response missions. To achieve this, the Column Generation method is employed to solve the initial routing problem within a specific grid. The primary objectives of this approach are twofold: to maximise population coverage and maximise fairness in the distribution of coverage, with a focus on assisting the most vulnerable individuals. Since the data accounts for two conflicting objectives, only Pareto-optimal solutions are considered and compared based on the assigned weights for each objective. Moreover, the Q-Learning algorithm is utilised to dynamically evaluate and prioritise newly discovered survivors who were not initially expected. This algorithm enables the agent to adapt and make decisions based on the vulnerability of the survivors. In order to address uncertainties inherent in emergency situations, a random variable is incorporated, enabling the agent to learn and make decisions based on new information.

Keywords

Column Generation; Q-Learning; Reinforcement Learning; OR in emergency response; Drone Deployment.

1. Introduction

1.1. Ethical dilemmas in autonomous systems

With the rise of artificial intelligence (AI), autonomous systems now possess the ability to make optimal decisions based on their current state and pool of available actions. This means that robots, such as autonomous vehicles (AVs) or unmanned aerial vehicles (UAVs), can make decisions independently, without human guidance, potentially encountering ethical dilemmas without clear criteria for resolution. Considering the well-known Moral Machine experiment conducted in (Awad, et al., 2018), which explored various moral dilemmas faced by AVs when brakes are not working, like deciding whether to run over a group of elderly individuals or crash into a wall, resulting in harm to the passengers inside the car. The experiment collected over 40 million decisions across ten languages. Building upon these ethical dilemmas, numerous studies have attempted to design optimal trajectories that optimise specific metrics, depending on the philosophical theory employed.

One such article (Evans, de Moura, Chauvier, Chatila, & Dogan, 2020) proposes a strategy for AV decision-making based on the Ethical Valence Theory. They incorporate the conservation of linear momentum as part of the reward function's quantification, along with other parameters that vary based on moral profiles, such as risk-averse altruism or threshold egoism, and the probability of a MAIS3+ injury given the velocity difference between the vehicles. They model this scenario as a Markov Decision Process (MDP), precisely defining the state space, action set, transition probability and reward function. The ethical trajectory is also presented in (Geisslinger, Poszler, & Lienkamp, 2023). They employ a harm model based on linear momentum and logistic regression to quantify the severity of damage on a scale of 0 to 1, representing the probability of MAIS3+ injuries given the velocity difference between the involved agents. They also consider a proper distribution of risk, primarily based on four principles: Bayes, Equality, Maximin, and Responsibility, to define the cost function and predict potential trajectories of incoming agents. Other work that considers the social dilemma is (Ebina & Kinjo, 2021) where they use a social welfare function having as a base a parameter ρ that quantifies the inequality aversion.

To quantitatively address the ethical aspect in designing optimal trajectories, a common approach is to utilise logistic regression, which provides the probability of fatality based on the vehicle's velocity. In (Hussain, Feng, Grzebieta, Brijs, & Olivier, 2019) the authors analyse, filter and compare 55 studies regarding pedestrians being struck in motorised vehicle crashes. They concluded that setting speed limits of 30-40 km/h for pedestrian areas is commonly used by countries with the lowest fatality rates in roads. Additionally, one such study that provides the parameters of the logistic regression using real data in China and compares the results between pedestrians and cyclists is (Nie, Li, & Yang, 2015) where they conclude that fatality risks at 50 km/h are more than twice as high at 40 km/h and that velocity has a high impact on mortality in urban traffic, this clearly shows how a change of velocity can increase dramatically the damage generated.

Having in mind that there is a large body of literature implementing different strategies to design ethical trajectories in AVs varying the philosophic theory, there is also literature that shows the difficulties faced in this approach. In (Etzioni & Etzioni, 2017) they demonstrate that cars have no moral agency and claim that part of the challenge posed by robots can be tackled by the similar ethical choices made by humans. The top-down approach to render moral choices consists of programming moral principles into the car's trajectories design (as in the studies above). This can be done in many ways like choosing a general moral philosophy. The problem is that adhering to a particular moral philosophy knowing that any one of them will eventually lead to actions considered as morally unacceptable (Etzioni & Etzioni, 2017). Also, the selection of a particular philosophic theory can be biased by the programmer's own beliefs. Finally, the Veil of Ignorance (VoI) was proposed by the philosopher John Rawls as an experiment to identify principles for ruling a society. In (Weidinger, y otros, 2023) they apply the VoI in AI and conclude that participants who think under the VoI usually make decisions in terms of fairness.

1.2. Column Generation

Due to the potentially large number of routes in a complete graph, it is necessary to employ large-scale optimisation techniques to reduce computation time. The Column Generation technique is particularly well-suited when all routes are not initially enumerated. This method generates feasible and desirable routes iteratively, based on the reduced costs of the master problem (MP), in order to improve the objective function until the stopping criteria is met. Figure 1 illustrates the solution scheme.

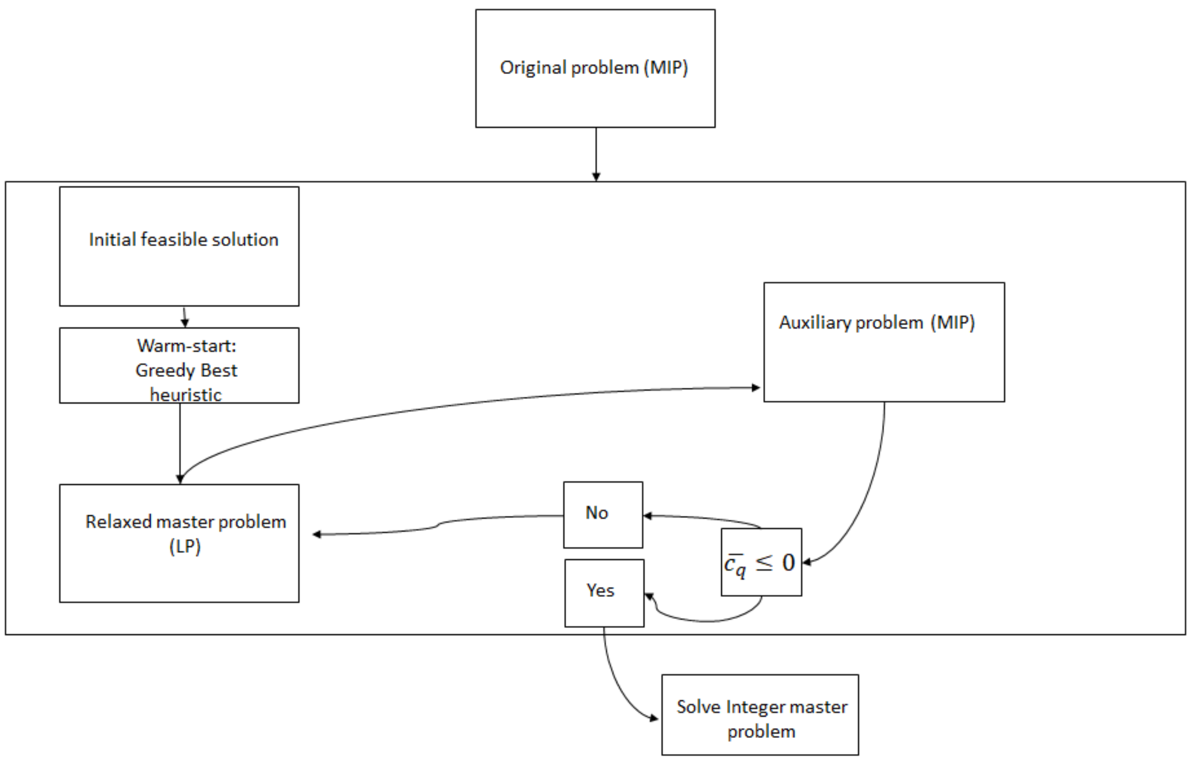

Fig 1. Column Generation solution scheme

To implement the Column Generation technique, the first step is to formulate a MP and an auxiliary problem (AP). The MP is formulated using the relaxed version of the original problem, where binary constraints are now continuous. This relaxed version allows for the calculation of the reduced costs.

To start, an initial feasible solution for the relaxed MP is required. Once the relaxed MP is solved, the dual variables provide the coefficients for the objective function of the AP. The AP is then solved, generating a new and attractive variable (a new route) for the MP. The addition of this variable is expected to improve the objective function. The stopping criteria for this technique is based on the reduced costs of the MP. In the case of maximising the objective function, the reduced costs should be less than or equal to zero to terminate the process. It is important to note that the Column Generation technique aims to iteratively generate and include new variables (routes) in the MP to improve the solution until the desired stopping criteria is met.

The Column Generation method is used in various contexts like the famous Vehicle Routing Problem (VRP) and all of its variants. Since the exact method is very time consuming, some algorithms have been developed to effectively reduce the computation time. In (Feillet, 2010) a branch-and-price approach tutorial using Column Generation exposes the main principles and theories in the context of VRPTW which is a VRP variant with time windows. Another approach in (Lozano, Duque, & Medaglia, 2016) propose a backtracking algorithm inspired by Branch & Bound bounds to effectively prune infeasible or dominated paths within a graph to solve the AP formulated as the Elementary Shortest Path Problem with Resource Constraints (ESPPRC) in a significant reduced computation time.

1.3. Stochastic optimisation

Sequential decision problems appear in many contexts and therefore many communities contribute to the solution of problems under uncertainty such as Reinforcement Learning (RL), Markov Decision Processes (MDPs), Optimal Control, among others. Some of the most important applications span in science, engineering, economics, finance and many others. In an attempt to unify every community that is subject to sequential decision analytics, (Powell, 2019) proposes a universal framework composed by five fundamental elements that every sequential decision problem should account in the formulation: state variables, decision variables, exogenous information, the transition function and the objective function. Summarised as follows:

State variables: the state S_t of the system saves all the information that is necessary to model the system. This is split into three types: the physical state R_t, which is usually controlled; other information I_t, which is known and not controlled not included in R_t and the belief state B_t, that contains information about parameters.

Decision variables: decisions that can be made and can be binary, discrete, continuous, among others and they are usually mapped by a policy $x_t = X^{\pi}(S_t)$. Our goal is to find these policies that maximise our objective function.

Exogenous information: contains the uncertainty of the system. This information is revealed after the actions or decisions are taken and it is denoted by W_t.

Transition function: the transition functions defines the state of the system after an action is taken and the exogenous information is revealed, it is denoted by $S_{t+1} = S^M(S_t, x_t, W_{t+1})$.

Objective function: represents the metric that we are optimising denoted by $C(S_t, x_t, W_{t+1})$.

As mentioned above, there are many communities that deal with sequential decision problems. One such community is MDPs that consist of the same five fundamental elements mentioned above. It is usually solved by backward induction using the optimality Bellman's equation:

$$V_t(S_t) = \max_{a_t \in A_t} \{C_t(S_t, a_t) + \gamma V_{t+1}(S_{t+1}(S_t, a_t))\}$$

But as mentioned before we need to deal with the fact that the reward function and the future state are uncertain and we want to select the most attractive action for each state (i.e., a function that maps a state into an action: a policy $X^{\pi}(S_t)$). But this set of optimal equations are usually huge and are subject to the curse of dimensionality. (Powell, 2022).

One of the most commonly used algorithms to solve MDPs is Value Iteration. It resembles backwards induction and is used for infinite horizon problems. Convergence for this algorithm is guaranteed under a specified criterion. One of the main issues with this algorithm is the curse of dimensionality, meaning that the computation time to solve the problem can become excessively long with a large state space. The algorithm computes the following equation recursively:

$$v^n(s) = \max_{a \in A} \left\{ C(s, a) + \gamma \sum_{s' \in S} P(s'|s, a) v^{n-1}(s') \right\}$$

And the convergence of the algorithm occurs when:

$$|v^n - v^{n-1}| < \delta$$

Where δ is a given parameter.

Finally, the policies are retrieved by:

$$\pi(s) = \operatorname*{argmax}_{a \in A} \left\{ C(s, a) + \gamma \sum_{s' \in S} P(s'|s, a) v^{n-1}(s') \right\}$$

(Powell, 2022) (Sutton & Barto, 2015)

On the other hand, an important approach to solving sequential decision problems arises in the Reinforcement Learning community, and a relevant algorithm in this regard is Q-Learning. This is an off-policy temporal difference algorithm developed to solve MDPs. In Q-Learning, the goal is to estimate an optimal action-value function, represented by the Q-table, which provides the expected returns for each state-action pair. The Q-Learning algorithm operates based on an exploration and exploitation scheme. Initially, the agent takes random actions and iteratively updates the Q-table. This process continues until the Q-table has been updated sufficiently to converge and allow for exploitation of the system. The following equation represents the algorithm:

$$Q^{n+1}(s, a) \leftarrow (1 - \alpha) Q^n(s, a) + \alpha \left[r(s, a) + \gamma \max_{a \in A} \{Q^n(s', a')\} \right]$$

The above equation is the update rule that estimates the value of taking action a being in state s. The parameter α represents the learning rate and γ is the discount factor for future rewards' importance, as in value iteration. (Sutton & Barto, 2015)

1.4. UAVs in emergency response missions

The need of UAVs in emergency response situations like earthquakes is increasing dramatically due to their potential benefits in these situations. One of the most critical goals in such situations is the effective and efficient exploration of the area to find survivors. In (Anastasiou, Kolios, Panayiotou, & Papadaki, 2020) they develop four heuristics where the objective is to maximise the node coverage withing a grid using the deployment of drones. They show promising results comparing the heuristics regarding the objective function, computation time and robustness. Finally, they simulate the results in Robot Operating System (ROS) to show the applicability of the approach. One important drawback is that they don't distinguish between the different nodes i.e., they don't take into account the special characteristic that a node can have such as the estimated number of people in a location or any metric related to quantify the importance to visit a specific location.

Regarding this last point, it is important to introduce the ethical dilemmas that can arise from this context. One study that deals with this topic conceptually is (Battistuzzi, Recchiuto, & Sgorbissa, 2021) where they argue that operations in disaster scenarios are full of ethical challenges. For example, it is of special importance where to concentrate the rescue efforts in such situations to decide where or who to search for first, who and why should be given priority, which is related to resource allocation and its specific metric could be determined to define these ethical dilemmas. One of their most important references is (Brandao, Jirotka, Webb, & Luff, 2020) where they plant a hypothetical case of a UAV deployment mission after a disaster to search and rescue survivals. The charging depot of the UAV is placed in the city centre of Oxford, UK where it is mostly filled with students that would potentially have more probability to survive than an elderly or a child. Therefore, they build and propose a pareto front that accounts for the distributive fairness (in terms of ethnicity, gender and age) against the total population coverage in the exploration. If the UAV explores the most area in the centre, it will be successful in terms of finding the greater number of people, but if the UAV explores in the surroundings of the city, it will be successful in terms of finding the most vulnerable people and achieve the distributive fairness proposed.

For a general review of the different techniques used in path planning for UAVs in (Aggarwal & Kumar, 2020) they discuss various techniques employed in the last years regarding the shortest paths in such scenarios but also collision-free environments to the UAVs. Their study contrast three types of techniques: representative techniques, cooperative techniques and non-cooperative techniques. The techniques discussed include machine learning models, multi-objective optimisation models, artificial intelligence techniques, among others. Concerning multi-objective models for UAVs path planning, (Saha, Vasegaard Elkjaer, Nielsen, Hapka, & Budzisz, 2021) propose a mixed-integer programming model under a multi-objective optimisation framework to design trajectories. The first objective in their model aims to maximise the cumulative probability of target detection. Whereas the second objective's goal is to minimise the cumulative path length to provide higher resource utilisation. To solve the multi-objective problem, they opt to transform the objective functions into one by incorporating a specific weight α and $1-\alpha$ to each objective function. Finally, they compare different results in a heatmap varying the weights of the objective function. One last relevant article proposes a behaviour-based cooperative search and rescue mission where simulation is used to gather situational awareness data during the first few hours after a natural disaster. They conclude that their simulation is successful in locating over 90% of survivor within an hour (Arnold, Yamaguchi, & Tanaka, 2018).

We have made several contributions to the field, including:

I. Contrasting and experimenting the trade-offs that can arise in an emergency response mission, specifically in an earthquake
II. Introducing a novel mixed integer programming model to ensure a fair route planning in earthquake scenarios for UAVs while accounting for the population coverage
III. Adapting large-scale optimisation techniques like Column Generation to new contexts
IV. Training an agent that can prioritise and decide dynamically between survivors given new information

The primary goal of this paper is to evaluate the trade-offs that can result in such situations and discuss the results from these ethical dilemmas. We aim to show that the proposed model can work to contrast between different prioritisation of the weights. We also seek to contribute to the growing body of literature on optimisation techniques in the deployment of UAVs regarding fairness and resource allocation in the natural disaster's context.

The structure of this document is as follows: The objectives of the model we are solving are presented in Section 2. The context, instance generation and mathematical models are in Section 3. Furthermore, the computational results and experiments are presented in Section 4. Finally, the conclusions and future work are showed in Section 5.

2. Research objectives

- Maximise population coverage in the initial routing while ensuring fairness within a reasonable computation time
- Address ethical dilemmas based on the number of people, battery time limitations and medical resource constraints
- Compare and contrast the results by varying the trade-offs in these situations in order to discuss and analyse the consequences of the chosen weights

3. Problem formulation

The context of this problem revolves around natural disasters, particularly earthquakes. Our objective is to address the routing problem of UAVs in such scenarios, with two primary objectives in mind: maximising population coverage and ensuring fairness in the distribution of coverage. Given the urgency and criticality of these situations, finding a prompt and effective solution is of utmost importance.

To compute the routes, we consider the estimated demographic population within specific locations, which are represented as nodes in the graph. Moreover, each node in the graph is assigned a fairness value that corresponds to its vulnerability. The graph is structured as a grid, providing a framework for the problem's representation and analysis.

The sets and the grid are defined:

N: Locations to visit. $\{1, 2, \ldots, n\}$

S: Feasible routes. $\{1,\ 2,\ 3,\ \ldots, P\}$

Where n is the number of nodes and P is the number of all possible routes.

Graph definition:

$$A = \{(i, j) | i \in N \cup \{0\},\ j \in N \cup \{n + 1\},\ i \neq j\}$$

$$G = (N \cup \{0,\ n + 1\}, A)$$

In our model, we assume that every node in the grid is connected to each other. To simplify the problem, we designate the depot as node 0 and node $n + 1$, allowing us to formulate the problem as a shortest/longest path problem. The grid structure can be observed in Figure 2.

As previously mentioned, our objective is to solve the multi-objective problem by obtaining Pareto-optimal solutions. This involves assigning specific weights to each objective. The goal is to construct the Pareto front for each instance, taking into account that its usefulness may vary depending on the data. In some cases, the Pareto front may not provide valuable insights if there are no conflicting objectives present (such as having the most vulnerable nodes coincide with the most populated nodes). Therefore, the generation of instances was strategically designed to position the most vulnerable nodes farther from the depot, while locating a majority of the population closer to the charging depot.

Fig 2. Instance used

From figure 2 it is possible to notice that the "warmest" nodes are the most vulnerable, whereas the point closer to the triangle (depot) are the most populated.

From the instance generated in figure 2 the Pareto front in figure 3 was built:

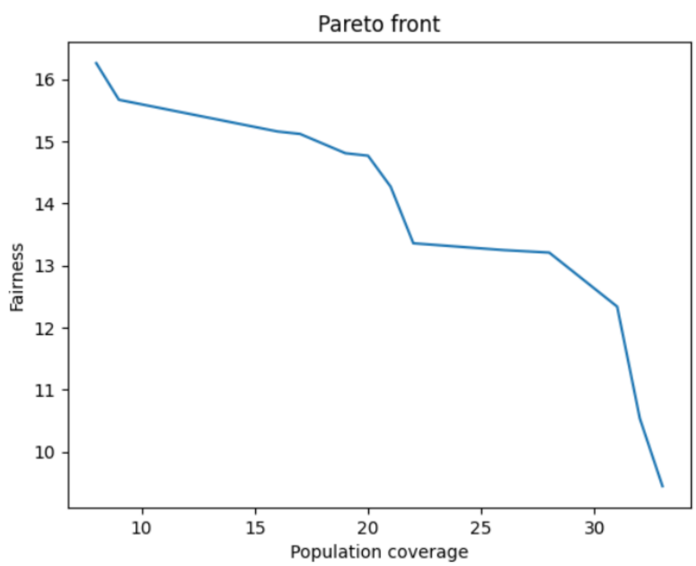

Fig 3. Pareto front

The Pareto front in figure 3 shows how both objectives are in conflict with each other. In our model we want to maximise the coverage of most vulnerable nodes (fairness) while maximising the population coverage as well. Hence, it is possible to analyse and contrast the trade-offs generated in such situations. The more the population is covered, the less fair the exploration is.

3.1. Mathematical programming formulation

In this section the models for the Column Generation technique are explained.

The parameters for the MP are:

$$b_{ijs}: \begin{cases} 1 \text{ if route s} \in \text{S traverses arc (i, j)} \in \text{A.} \\ 0 \qquad\qquad \text{on the contrary.} \end{cases}$$

k: number of UAVs available

And conveniently we define:

$$c_{ij} = \omega_1 l_{ij} + \omega_2 P_{ij}$$

$$a_{is} = \sum_{j|(i,j)\in A} b_{ijs}$$

$$c_s = \sum_{(i,j)\in A} c_{ij} b_{ijs}$$

Where

$$a_{is}: \begin{cases} 1 \text{ if route s} \in \text{S traverses node i} \in \text{N.} \\ 0 \qquad\qquad \text{on the contrary.} \end{cases}$$

c_{ij}: cost of entering node $j|(i,j) \in A$

c_s: price of the route $s \in S$.

The MP is formulated as follows:

$$max \sum_{s\in S} c_s \theta_s \tag{1.1}$$

$$s.t.$$

$$\sum_{s\in S} a_{is}\theta_s \leq 1, \ \forall i \in N \tag{1.2}$$

$$\sum_{s\in S} \theta_s \leq k \tag{1.3}$$

$$\theta_s \in \{0,1\}, \ \forall s \in S \tag{1.4}$$

The objective function (1.1) selects the routes with the higher costs as they are associated with population and fairness considerations. Constraints (1.2) guarantees that the UAVs don't visit nodes already visited by other UAVs.

Constraint (1.3) ensures that the model doesn't select more routes than the available UAVs. Ultimately, Equation (1.4) represent the nature of the variables.

And the relaxed MP is formulated as:

$$max \sum_{s \in S} c_s \theta_s \tag{1.1}$$

$$s.t.$$

$$\sum_{s \in S} a_{is} \theta_s \leq 1,\ \forall i \in N \qquad [\lambda_i \geq 0] \tag{1.2}$$

$$\sum_{s \in S} \theta_s \leq k \qquad [\sigma \geq 0] \tag{1.3}$$

$$\theta_s \geq 0,\ \forall s \in S \tag{1.5}$$

Where the binary constraints are relaxed in Equation (1.5) and we retrieve dual variables ($\lambda_i \in N$ and σ) from each constraint to build de AP.

The AP can be formulated as a single UAV, because we are assuming that every drone has the same specifications, therefore the model is homogenous and can be separated by drones.

The parameters for the AP are:

T: battery time for each UAV

P_{ij}: probability of vulnerability of arc $j|(i,j) \in A$

λ_i: dual variables from MP

ω_i: weights for each objective

And the costs of each node are defined by:

$$c'_{ij} = \begin{cases} c_{ij} - \lambda_i, & i \in N \\ c_{ij}, & i \notin N \end{cases} \quad \forall (i,j) \in A$$

Where:

c_{ij}: cost of entering node $j|(i,j) \in A$

c'_{ij}: altered costs for the AP

And the formulation is as follows:

$$\max \sum_{(i,j)\in A} c'_{ij} b_{ij} \tag{2.1}$$

$$s.t.$$

$$\sum_{(0,j)\in A} b_{0j} = 1 \tag{2.2}$$

$$\sum_{(i,n+1)\in A} b_{i,n+1} = 1 \tag{2.3}$$

$$\sum_{j|(i,j)\in A} b_{ij} - \sum_{j|(i,j)\in A} b_{ji} = 0,\ \forall i \in N \tag{2.4}$$

$$\sum_{(i,j)\in A} t_{ij} b_{ij} \leq T \tag{2.5}$$

$$u_i - u_j \leq M\left(z_j - b_{ij}\right) - 1,\ \forall i \in N, J \in N | i \neq j \tag{2.6}$$

$$b_{ij} \in \{0,1\},\ \forall\ (i,j) \in A \tag{2.7}$$

$$z_i \in \{0,1\},\ \forall\ i \in A \tag{2.8}$$

$$u_i \geq 0,\ \forall\ i \in N \tag{2.9}$$

The objective function (2.1) aims to maximise the cost of the route. Constraint (2.2) guarantees that the UAV starts from the depot, and constraint (2.3) ensures that the drones finishes its route at the depot. Constraints (2.4) ensures flow through the grid. Constraint (2.5) prevents the route from being longer than the battery time of the UAV. Constraints (2.6) are the subtour constraints. Ultimately, Equations $(2.7) - (2.9)$ represent the nature of the variables.

3.2. Pareto front procedure

Since this is a multi-objective optimisation problem, decision-makers may struggle to determine the appropriate weights for each component of the objective function. It would be easier to access the degree of improvement in one component relative to the other. In this particular problem, the aim is to maximise population coverage and select the most vulnerable locations. However, these two objectives result conflictive for the instances generated in this study. That's why the Pareto front was built using the augmented ϵ constraint method outlined in (Mavrotas, 2009). The first step to build the Pareto front involved calculating the payoff table, i.e., each objective's maximum and minimum value. To achieve this, the model was computed with one weight set to 0 in order to identify the maximum value for the other component.

Table 1. Payoff table

Optimisation	OF_1	OF_2
$max\ OF_1$	35	8,37
$max\ OF_2$	5	16,26

Where $OF_1 = \sum_{(i,j)\in A} l_{ij} b_{ij}$ and $OF_2 = \sum_{(i,j)\in A} P_{ij} b_{ij}$

Now, with the payoff table computed, the optimisation problem to build the Pareto front consisted of making one of the objectives as a constraint, Equations $(2.1) - (2.11)$ formulate the iterative process to compute the Pareto front:

$$\max OF_1 - (\beta / r) SV \tag{2.10}$$

$$OF_2 + SV = \varepsilon \tag{2.11}$$

Where β is a small number between 10^{-3} and 10^{-6}, SV is a slack variable, r is the range of OF_2 and ϵ is calculated each iteration by:

$$\epsilon = \underline{OF_2} + \delta$$

Being δ a step, and $\underline{OF_2}$ is the minimum value for OF_2. This is a process of reoptimising every time the parameter ϵ is updated on each iteration. This way, the Pareto front in figure 3 was obtained.

3.3. Sequential decision formulation

For the second approach, we modelled the system so that it can dynamically respond to new information in the environment. Therefore, we adopted the universal frameworks from (Powell, 2019) explained above to model the trajectory and let the UAV decide under uncertain situations. The five essential elements of the framework as defined above are explained next:

State space:

$$S_t = \{(x, y), r, done\}$$

The state space is defined as the position of the drone in the (x, y) plane withing the grid, r is the available resource and the binary variable $done$ takes the value of 1 if the drone has already helped survivors and it is out of resource r. This last component is of special importance because is responsible for the UAV to learn to return to the depot.

Actions:

$$X_t = \{(x, y), r\}$$

The available actions of the UAV correspond to moving in the (x, y) plane withing the grid, or to assign resource r when in a survivors' node.

Exogenous information:

$$W = \{Real\ number\ of\ survivors\ in\ the\ node\}$$

Therefore, we define the random variable:

Y: number of survivors in the node

$$Y \sim Unif[\bar{n} - q, \bar{n} + q]$$

Where it is modelled as a discrete Uniform distribution and $\bar{n}$ is the initial demographic estimation of people in the node and q is a tuneable parameter that gives the range of people in the node. This information is revealed after the drone reaches the node.

Reward function:

$$R_t(S_t, W_t, S_{t+1})$$

Since we have thousands of state spaces there are a few that have specific rewards defined as:

Move penalty: -5, every time the UAV moves.

Station reward: 200, when the UAV reaches the station having visited survivors before (i.e., $S_t = \{(0,0), r, 1\}$ independently of the resource not consumed.

Battery penalty: -300, if the drone goes off of battery level without reaching the depot.

Visiting a survivor and having resource available: n

Visiting a survivor already visited: $-n$

From the rewards defined, it is possible to notice how we are addressing the ethical dilemmas as a first approximation. The UAV will tend to visit the nodes with the greatest number of people, since the reward is greater.

Finally, the objective function is:

$$\max R_t$$

The challenge in this problem results in addressing the uncertainty, the huge number of states and the optimal policy design.

4. Designing fair UAVs navigation systems

All the results were run in processor Intel(R) Core (TM) i5-8250U CPU @ 1.60GHz 1.80 GHz, RAM 8,00 GB (7,86 GB usable) in type 64-bit operating system, x64-based processor in Gurobi optimiser under version 10,0 and academic license[1].

Column Generation approach

Following the solution scheme in figure 1, the Greedy Best heuristic (Anastasiou, Kolios, Panayiotou, & Papadaki, 2020) was selected to provide initial routes to the MP of the Column Generation. The pseudo code can be seen in figure 4:

Algorithm 2 Greedy Best (GB) Heuristic

Step 0:
$S_k = \{s\}$; $S = \cup_{k=1}^{K} S_k$; $P_k = \emptyset$, $R = \{1, \ldots, K\}$.
Step 1:
for $k \in R$ **do**
  Find minimum cost arc (i, j) with $i \in S_k$, $j \in N \setminus S$.
  if $|P_k \cup \{(i, j)\}| + c_{js} \leq B(k)$ **then**
    $P_k \leftarrow P_k \cup \{(i, j)\}$, $S_k \leftarrow S_k \cup \{j\}$, Update S.
  else
    Remove k from R.
  end if
end for
Step 2:
If R not empty repeat Step 1, else go to Step 3.
Step 3:
The tours are $P_k \cup \{(l(S_k), s)\}$ for each k.

Fig 4. Greedy Best (GB) heuristic (Anastasiou, Kolios, Panayiotou, & Papadaki, 2020)

With the initial routes provided as a warm start, we can assure that our model starts with a set of feasible routes that aims to maximise the node coverage within the grid. The main drawback of this heuristic is that it does not takes into account for the cost of each node, but strictly on the time to traverse each arc.

From this heuristic the following route is generated for one UAV:

[1] https://www.gurobi.com/academia/academic-program-and-licenses/

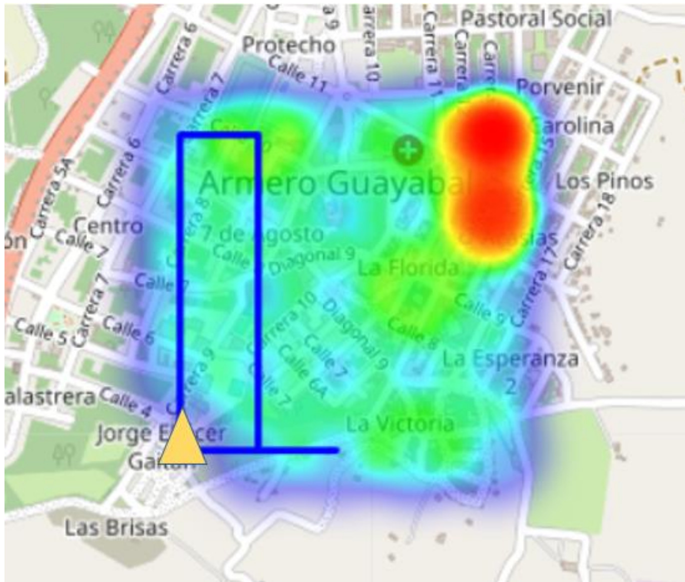

Fig 5. Initial route provided by GB heuristic

The objective function (OF) strictly depends on the chosen weights, that's why for every change in the weights, the initial OF will be calculated to prove that the Column Generation method actually improves throughout the iterations. The heuristic objective function (HOF) will be shown in each case along with the relaxed objective function (ROF) and the integer objective function (IOF).

After the warm start is employed, the Column Generation method starts running:

Generate an initial set of columns Ω_1

Do

  Solve $MP(\Omega_1)$

  $\Gamma \leftarrow$ column(s) provided by the subproblem

  $\Omega_1 \leftarrow \Omega_1 \cup \Gamma$

While $\Gamma \neq \emptyset$

Fig 6. Column Generation method (Feillet, 2010)

In figure 6 the iterative Column Generation method is exposed.

<u>Case 1.</u>

In the first case we wanted to test when the model only accounted for the population coverage, the results can be seen in figure 7 and the parameters employed:

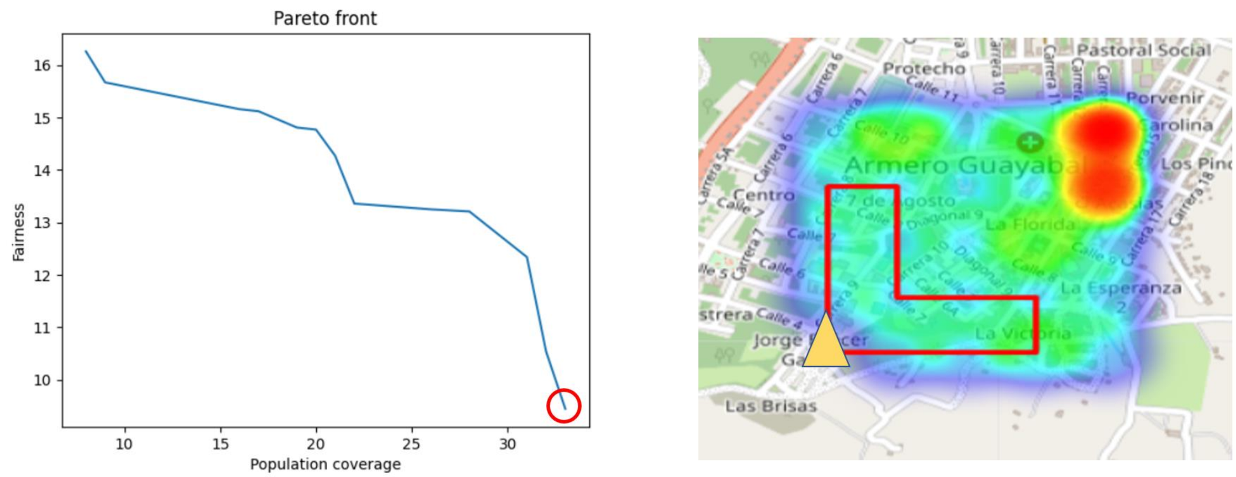

Fig 7. Results maximising on population coverage

$$c_{ij} = \omega_1 l_{ij} + \omega_2 P_{ij}$$

Where:

$$\omega_1 = 1$$

$$\omega_2 = 0$$

As specified above, the instance was conveniently generated so that the population is mostly closer to the depot and the most vulnerable is further. Using this approach it is relevant to notice that the route doesn't even care about the "warm" point in the grid. It only accounts for the population in the given locations.

Regarding the OFs:

Table 2. comparison of objective functions in the CG method.

HOF	30,0
ROF	35,0
IOF	35,0

<u>Case 2.</u>

In the second case we wanted to test when the model accounted for a mid-point between population coverage and fairness, the results can be seen in figure 8 and the parameters employed:

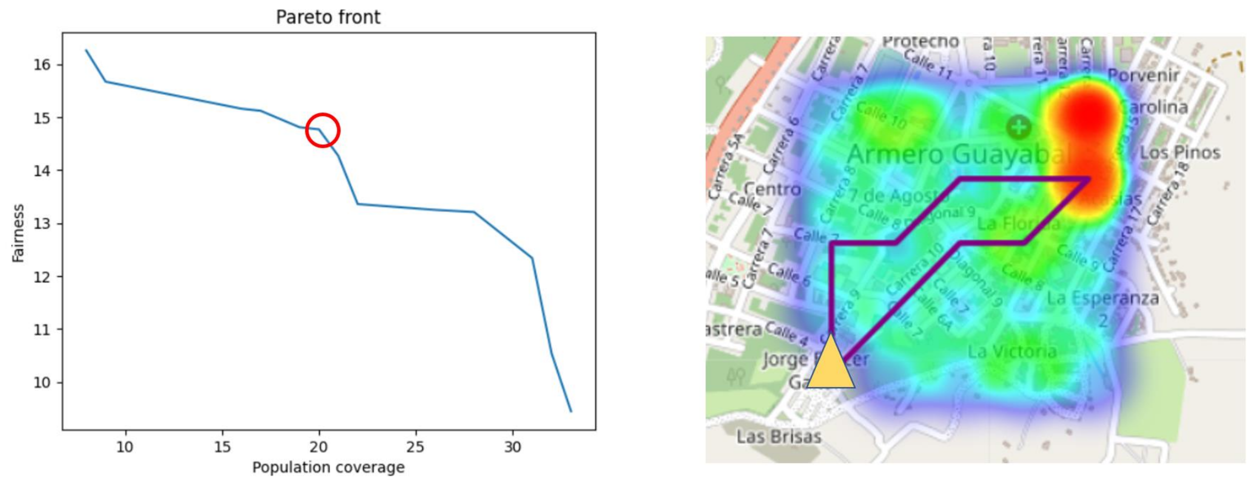

Fig 8. Results maximising on mid-point

$$c_{ij} = \omega_1 l_{ij} + \omega_2 P_{ij}$$

Where:

$$\omega_1 = 1$$

$$\omega_2 = 6$$

In this case, the model is not completely maximising on both objectives. Therefore, the route traverse one of the "warmest" nodes but also ensures that a certain amount of population is covered.

Regarding the OFs:

Table 3. comparison of objective functions in the CG method.

HOF	68,76
ROF	108,62
IOF	108,62

Case 3.

In the third case we test when the model only accounts for fairness, the results can be seen in figure 9 and the parameters employed:

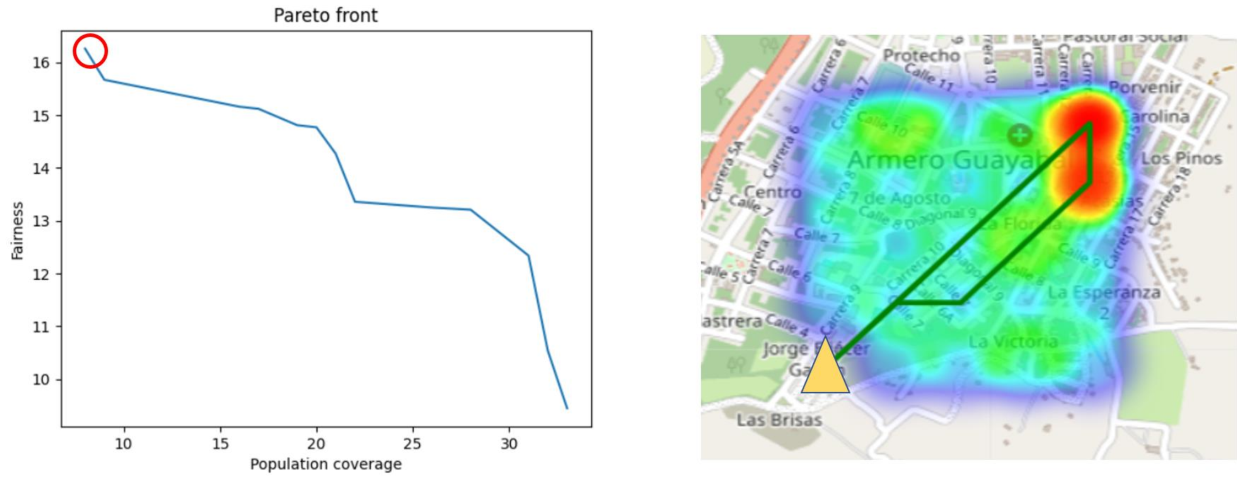

Fig 9. Results maximising on fairness

$$c_{ij} = \omega_1 l_{ij} + \omega_2 P_{ij}$$

Where:

$$\omega_1 = 0$$

$$\omega_2 = 1$$

In the final case, the model didn't account for the population. Therefore, the route generated focused on the "warmest" nodes in the graph.

Regarding the OFs:

Table 4. comparison of objective functions in the CG method.

HOF	6,46
ROF	16,26
IOF	16,26

From tables 2-4 it is noticeable how the objective function improves using the Column Generation method and the heuristic solution is far from the optimal.

Computation time comparison with the new Elementary Shortest Path Problem with Resource Constraints (ESPPRC)

As can be noticed, the AP formulation corresponds to an Elementary Longest Path Problem with Resource Constraints since the OF is maximising the node coverage, it has cycles and there is a resource constraint related to the battery time of the UAV. By experimenting with the model, we notice a benefit in the computation time if the original OF:

$$\max \sum_{(i,j)\in A} c'_{ij} b_{ij}$$

Is converted to:

$$\min \sum_{(i,j)\in A} -c'_{ij} b_{ij}$$

We generated five different instances and tested the computation time and OF for each of them. The results can be seen next:

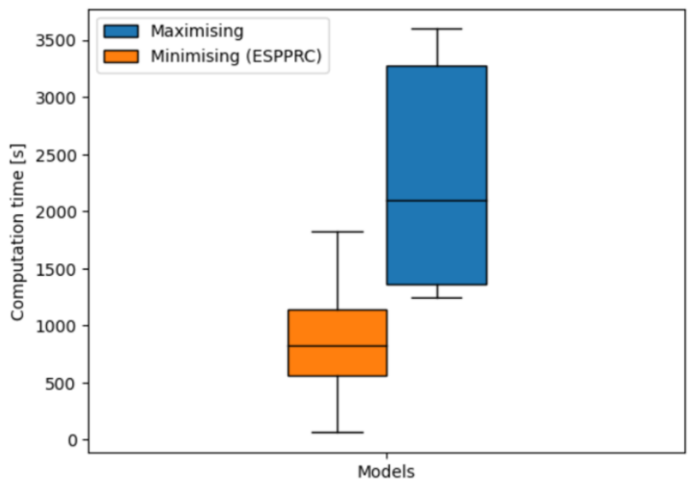

Fig 10. Computation time comparison

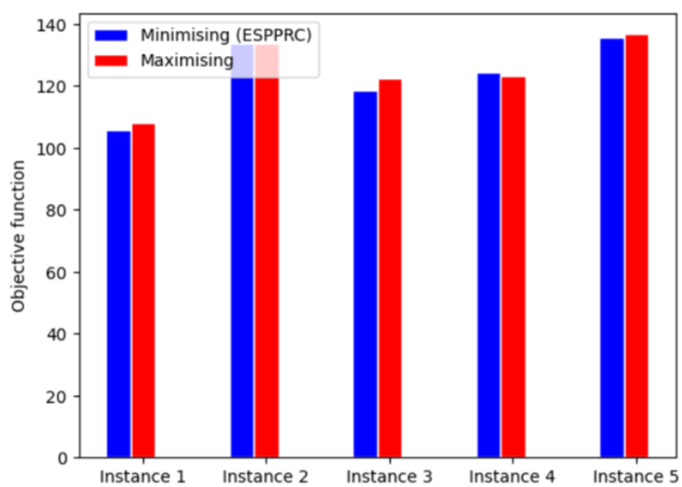

Fig 11. Objective function comparison

In figure 10 one can see that the computation time was significantly reduced when the ESPPRC formulation is used. Additionally, in figure 11 one can contrast between the OF obtained: resulting in an almost equal value for each formulation. Hence, we can conclude that it is more beneficial to employ the ESPPRC formulation to reduce the computation time.

Q-Learning approach

For the Q-Learning some hyper-parameters needed to be defined:

Episodes: 140000

Steps: 100

ϵ: $0{,}9$

Decay: $0{,}9998$

α: $0{,}1$

γ: $0{,}95$

After training the agent under the specified universal framework and hyper-parameters the following learning curve was developed:

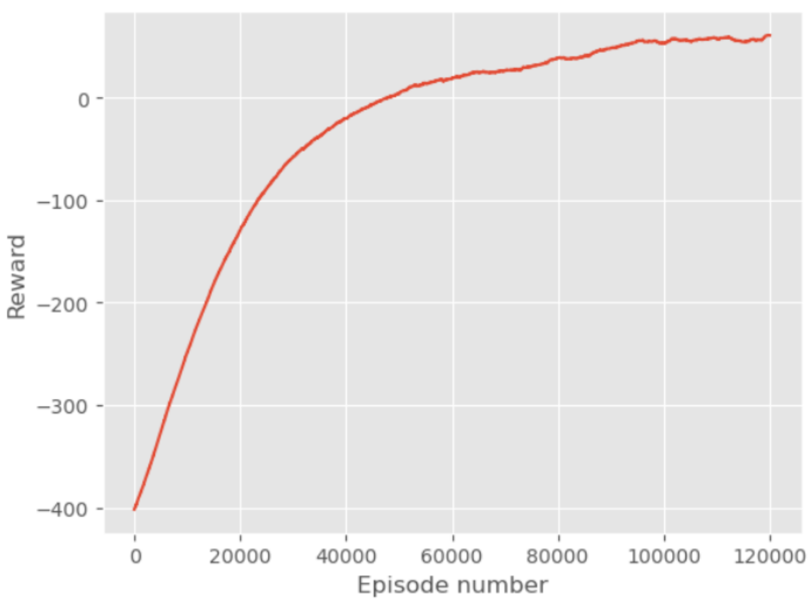

Fig 12. Learning curve for the agent using Q-Learning

From figure 12, one can see that the agent effectively learns to explore the area, search and rescue survivors, and finally return to the depot having battery available. This approach helps to quantify the dynamics of having new information and to decide dynamically if any changes have to made from the initial routing problem using Column Generation.

5. Conclusion

In this study, we consider the routing problem of UAVs in emergency situations such as an earthquake. Depending on the instance used, we can reveal important trade-offs between population coverage and the assistance provided to the most vulnerable individuals. The large-scale optimisation technique, Column Generation, was used along with a warm start to solve the problem. This approach demonstrated progressive improvement throughout the iterations. Additionally, the reformulation of the AP into an ESPPRC showed a significant reduction in the computation time to solve the model optimally. On the other hand, the Pareto front analysis showed that prioritising one objective can lead to a compromise in the other in some cases. Furthermore, the successful Q-Learning curve indicated the effectiveness of the algorithm in adapting and improving decision-making under uncertainty. Finally, it is important to consider the ethical aspects of robots. We discuss why a robot cannot take an ethical position and why it would be biased if the decision-maker has the opportunity to select a moral philosophy to embed in the robot. It is relevant to note that ethical concerns are underexamined in the literature, and these issues must be addressed with the rise of AI.

Future work includes using a real UAV to simulate and demonstrate the applicability of the approaches in this work. It would also be a great contribution to improve and properly define fairness quantification in order to be able to compare different resource allocation metrics. To make the model more realistic, it would be relevant to include collision-free environments, consider stochastic battery levels, and weather fluctuations. Finally, regarding the efficiency of the models, it is important to accelerate the AP's computation time and enhance the Q-Learning agent.

All the code can be found in GitHub.[2]

References

Aggarwal, S., & Kumar, N. (2020). Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges. *Computer Communications*, 270-299. doi:https://doi.org/10.1016/j.comcom.2019.10.014

Anastasiou, A., Kolios, P., Panayiotou, C., & Papadaki, K. (2020). Swarm Path Planning for the Deployment of Drones in Emergency Response Missions. *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, (pp. 456-465). Athens. doi:10.1109/ICUAS48674.2020.9213876

[2] https://github.com/germanrpardo1/Tesis-Pregrado

Arnold, R. D., Yamaguchi, H., & Tanaka, T. (2018). Search and rescue with autonomous flying robots through behavior-based cooperative intelligence. *Journal of International Humanitarian Action*, 3-18. doi:https://doi.org/10.1186/s41018-018-0045-4

Awad, E., Dsouza, S., Kim, R., Schulz, J., Henrich, J., Shariff, A., . . . Rahwan, I. (2018). The Moral Machine experiment. *Nature*, 59-64. doi:https://doi.org/10.1038/s41586-018-0637-6

Battistuzzi, L., Recchiuto, C. T., & Sgorbissa, A. (2021). Ethical concerns in rescue robotics: a scoping review. *Ethics and Information Technology*, 863-875. doi:https://doi.org/10.1007/s10676-021-09603-0

Brandao, M., Jirotka, M., Webb, H., & Luff, P. (2020). Fair navigation planning: A resource for characterizing and designing fairness in mobile robots. *Artificial Intelligence*. doi:https://doi.org/10.1016/j.artint.2020.103259

Ebina, T., & Kinjo, K. (2021). Approaching the social dilemma of autonomous vehicles with a general social welfare function. *Engineering Applications of Artificial Intelligence*. doi:https://doi.org/10.1016/j.engappai.2021.104390

Etzioni, A., & Etzioni, O. (2017). Incorporating Ethics into Artificial Intelligence. *The Journal of Ethics*, 403-418. doi:DOI 10.1007/s10892-017-9252-2

Evans, K., de Moura, N., Chauvier, S., Chatila, R., & Dogan, E. (2020). Ethical Decision Making in Autonomous Vehicles: The AV Ethics project. *Science and Engineering Ethics*, 3285-3312. doi:https://doi.org/10.1007/s11948-020-00272-8

Feillet, D. (2010). A tutorial on column generation and branch-and-price for vehicle routing problems. *A Quarterly Journal of Operations Research*, 407-424. doi:https://doi.org/10.1007/s10288-010-0130-z

Geisslinger, M., Poszler, F., & Lienkamp, M. (2023). An Ethical Trajectory Planning Algorithm for Autonomous Vehicles. *Nature Machie Intelligence*, 137-144. doi:https://doi.org/10.1038/s42256-022-00607-z

Hussain, Q., Feng, H., Grzebieta, R., Brijs, T., & Olivier, J. (2019). The relationship between impact speed and the probability of pedestrian fatality during a vehicle-pedestrian crash: A systematic review and meta-analysis. *Accident Analysis and Prevention*, 241-249. doi:https://doi.org/10.1016/j.aap.2019.05.033

Lozano, L., Duque, D., & Medaglia, A. L. (2016). An Exact Algorithm for the Elementary Shortest Path Problem with Resource Constraints. *Transportation Science*, 348-357. doi:http://dx.doi.org/10.1287/trsc.2014.0582

Mavrotas, G. (2009). Effective implementation of the e-constraint method in Multi-Objective Mathematical Programming problems. *Applied Mathematics and Computation*, 455-465.

Nie, J., Li, G., & Yang, J. (2015). A Study of Fatality Risk and Head Dynamic Response of Cyclist and Pedestrian Based on Passenger Car Accident Data Analysis and Simulations. *Traffic Injury Prevention*, 76-83. doi:10.1080/15389588.2014.881477

Powell, W. B. (2019). A unified framework for stochastic optimization. *European Journal of Operational Research*, 795-821. doi:https://doi.org/10.1016/j.ejor.2018.07.014

Powell, W. B. (2022). *Reinforcement Learning and Stochastic Optimization: A Unified Framework for Sequential Decisions.* John Wiley & Sons.

Saha, S., Vasegaard Elkjaer, A., Nielsen, I., Hapka, A., & Budzisz, H. (2021). UAVs Path Planning under a Bi-Objective Optimization Framework for Smart Cities. *Electronics*. doi:https://doi.org/10.3390/electronics10101193

Sutton, R., & Barto, A. (2015). *Reinforcement Learning: An Introduction.* The MIT Press.

Weidinger, L., Mckee, K. R., Everett, R., Huang, S., Zhu, T. O., Chadwick, M. J., . . . Gabriel, I. (2023). Using the Veil of Ignorance to align AI systems with principles of justice. doi:https://doi.org/10.1073/pnas.2213709120