Dynamic decision making for situational awareness using drones: Requirements, identification and comparison of decision support methods

DOI:
10.1016/j.eswa.2024.124057

Document Version
Accepted author manuscript

Citation for published version (APA):
Duxbury, D. J., Paton, N. W., & Keane, J. A. (2024). Dynamic decision making for situational awareness using drones: Requirements, identification and comparison of decision support methods. *Expert Systems with Applications*, Article 124057. Advance online publication. https://doi.org/10.1016/j.eswa.2024.124057

Published in:
Expert Systems with Applications

Dynamic decision making for situational awareness using drones: requirements, identification and comparison of decision support methods

Dominic J. Duxbury
University of Manchester
United Kingdom
dominic.duxbury@manchester.ac.uk

Norman W. Paton
University of Manchester
United Kingdom
norman.paton@manchester.ac.uk

John A. Keane
University of Manchester
United Kingdom
john.keane@manchester.ac.uk

Abstract

Decision makers increasingly operate in real-time information-rich environments, where limited time is available for interpreting data to inform decisions. These environments are driven by static or mobile sensing devices that can provide numerous dynamic data points. A prominent approach in this space is to utilise drones, which can be deployed to gather targeted information. However, deciding how best to deploy available drones is nontrivial, and stands to benefit from decision support aids that plan routes. Such a system must operate under time constraints created by the changing attributes of routes as the situation unfolds. This study describes a dynamic decision support system (DSS) for situational awareness with drones. The system applies Multi-Criteria Decision Making (MCDM) methods within a dynamic genetic algorithm to provide a continuously revised ranking of routes. Five desiderata for dynamic decision support are presented. It is shown how a dynamic DSS can be equipped with declarative specification of preferences (Desiderata 1), dynamic revision of recommendations (Desiderata 2), and high diversity of options (Desiderata 3). The study then compares four MCDM methods, namely the Weighted Product Model (WPM), the Analytic Hierarchy Process (AHP), the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS), and the Preference Ranking Organization METHod for Enrichment Evaluation (PROMETHEE), with regards to how consistently they trade-off between criteria (Desiderata 4) and the stability of results under small changes to criteria values (Desiderata 5). To evaluate the trade-offs between criteria we analyse the smoothness of change in criteria outcomes as criteria weightings increase for each algorithm. The outcomes are calculated by automating the selection of routes in a case study that applies drones to the task of harbour management. The stability of results for the different MCDM methods are compared. Perturbations were applied to sets of routes ranked by each algorithm then each algorithm was reapplied and the magnitude of the changes in ranking was assessed. Overall, TOPSIS was found to be the algorithm which made the most consistent trade-offs between criteria, only under-performing another algorithm with respect to a single criterion. AHP and WPM were the next most consistent algorithms and PROMETHEE was the least consistent algorithm. TOPSIS was also found to be the most stable method under small changes to criteria values. AHP was the second most stable, followed by PROMETHEE and WPM respectively. The results show that TOPSIS achieves the best result for both Desiderata 4 and 5 and consequently the study finds TOPSIS to be an appropriate MCDM method for dynamic decision support.

Keywords: Drones; Multi-Criteria Decision Making; Decision support systems; AHP; TOPSIS; PROMETHEE.

1. Introduction

Maritime facilities face challenging demands, including monitoring ocean traffic, port safety and emergency response. New technology is required to tackle these challenges, under the stresses of higher levels of traffic and an increased need for rigorous safety and prompt emergency response (Sardain et al. 2019; Pirotta et al. 2019). Drones are a technology that has been identified for this role, and managing these drones for various tasks is one aspect of the expanding role of harbour management. Situational awareness has been highlighted as crucial in domains where the effects of ever-increasing technological and situational complexity on the human decision maker are a concern (Saner et al. 2009). Drones provide a means for aerial situational awareness, within a harbour and beyond (Frederiksen and Knudsen 2018; Macrina et al. 2020). One task, which has the potential to improve situational awareness, is automatic identification of ships approaching a harbour. Drones can be employed to take photos of ships, for the identification of traffic and potential threats. Selecting an appropriate route for the drone is a complex decision that can be informed through a decision support system (DSS).

DSSs are enabled by decision analysis, which is the field concerned with the study of complex decisions. Multiple-criteria decision-making (MCDM) is a sub-discipline of decision analysis comprising techniques for evaluating solutions with multiple conflicting criteria. These criteria are often valued differently by different decision makers and so require an expression of preferences to identify appropriate trade-offs between objectives. These preferences are then applied to rank solutions from most to least optimal. To select an appropriate route, the decision maker must consider multiple conflicting objectives such as identifying as many ships as possible, identifying ships as early as possible and reducing fuel costs. Navigating this large space of potential routes and making consistent trade-offs between objectives is a difficult task. Therefore, the management of these drones can potentially be simplified through the use of MCDM methods, employed within a DSS. MCDM methods allow the user to specify preferences in a declarative manner, which supports the discovery of candidate solutions and consistent trade-offs between criteria. A challenge for such a system is that ocean traffic is constantly moving and quickly changes direction. This necessitates that the problem be solved using a dynamic DSS, which updates routes as the scenario unfolds. In this paper we outline our approach to building a dynamic DSS which provides situational awareness in a harbour zone using drones.

It is impossible to capture every nuance of a problem within a DSS. Therefore, in this kind of high-stakes decision making domain, experts are relied upon to make the decision, assisted by the DSS. Together they form a human-computer team, hopefully performing better than either the human or computer alone (Tomsett et al. 2020; Steiner et al. 2018). Consequently, it is important that the expert is presented with a diverse array of options, rather than multiple similar solutions which may fall prey to similar pitfalls that have been overlooked by the system. Options should be given to the decision maker ranked from the most to least likely suitable solution. This kind of ranking can be generated using an MCDM method, which takes into consideration user preferences. Different methods can produce different rankings when applied to an identical problem, even with identical user preferences. For an MCDM problem it is often impossible to say which resultant ranking is optimal. Therefore, selecting an appropriate MCDM method for a problem is difficult (Zanakis et al. 1998).

A solution to this is to pick some desired characteristics for a method (Keshavarz-Ghorabaee et al. 2018), for example, consistent trade-offs. To make effective use of preferences, it would be useful for changes in criteria weights to have predictable effects. An aspect underpinning the predictability of changes in criteria weights is the consistency of trade-offs between criteria. It is expected that as the weighting for a criterion increases, the trade-offs become more favoured towards that criterion. However, this relationship is not always predictable, as small changes in criteria weightings can lead to large changes in how an algorithm values certain trade-offs. We view consistency in trade-offs as a desired feature, as it gives rise to predictable effects when decision makers adjust their preferences.

Another desirable characteristic for dynamic decision support problems is stability of results. In this work we refer to the propensity to reorder under changes to criteria values as the stability of a ranking. For a ranking to be functional, the frequency of change must be less than the time it takes for a decision maker to act. For this to be fulfilled, the rankings must take into account changing criteria values, without reordering significantly when small changes are made. Therefore, it is desirable for the rankings to be stable under small changes to criteria values.

Drawing this together, we have the following 5 desiderata for dynamic multi-criteria DSSs:

1. declarative specification of preferences,
2. dynamic revision of recommendations,
3. high diversity of options,
4. consistent trade offs between criteria, and
5. high stability of results.

To demonstrate how these desiderata can be supported, we propose a dynamic DSS. This system applies an MCDM method as a fitness function within a continuously running genetic algorithm. The approach takes into consideration user preferences (Desiderata 1), to generate a continuously updated ranking (Desiderata 2), with a mechanism to control the diversity of options (Desiderata 3). In addition, MCDM methods are evaluated with respect to how consistently they trade-off between criteria (Desiderata 4) and the stability of their rankings (Desiderata 5). The methods evaluated are the Weighted Product Model (WPM) (Bridgman 1922), the Analytic Hierarchy Process (AHP) (Saaty 1987), the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) (Hwang and Yoon 1981) and the Preference Ranking Organization METHod for Enrichment Evaluation (PROMETHEE) (Brans and Vincke 1985).

Contributions

The paper contributes the following three items:

- Desiderata for dynamic DSSs, as outlined above. These items make explicit features and characteristics that can help DSSs to work effectively over real-time data.

- A DSS for situational awareness using drones that supports declarative specification of preferences, dynamic revision of recommendations and high diversity of options. The approach uses *a priori* knowledge of the decision-maker's preferences to reduce the multi-objective evolutionary algorithm (MOEA) approach that had been applied to multi-UAV mission planning support in the literature (outlined in Subsection 2.3) to a single step process that can be run over dynamic data to continuously refine results.

- An evaluation that brings a generalisable approach to assessing decision support methods (outlined in Subsection 2.2) into the dynamic space. We mimic the approach of *Belton and Gear* (Belton and Gear 1983), identifying rank reversal as a negative characteristic of MCDM methods through the identification and evaluation of desiderata for specifically dynamic problems. The evaluation uses our DSS to analyse stability and trade-offs of four MCDM methods to determine the method that best supports dynamic decision making by affording consistent trade-offs between criteria and high stability in the context of small changes in criteria values.

The rest of this paper is structured as follows: Section 2 presents a background to MCDM methods, including an evaluation of MCDM methods and their application to situational awareness with drones. Section 3 outlines the MCDM methods applied within our architecture. Section 4 details the harbour management task. The architecture of our DSS is given in Section 5, including our implementation for diversification of results. Section 6 follows with an evaluation of MCDM methods with respect to consistent trade-offs between criteria. Section 7 evaluates the stability of MCDM methods under small changes of criteria values. Finally, Section 8 discusses the findings, and concludes on the limitations of the study.

2. Literature Review

2.1. Multi-criteria decision making

Belton and Stewart (Belton and Stewart 2002) categorised MCDM methods into three broad categories:

- Value measurement models that assign numerical scores to solutions.

- Goal, aspiration and reference models where solutions are evaluated by how far they fall from an established objective.

- Outranking models (the French school) which compares solutions to identify the advantages/disadvantages of selecting one over another.

To find a suitable method for dynamic decision making, we evaluate WPM and AHP as value measurement methods, TOPSIS as a goal, aspiration and reference model, and PROMETHEE as an outranking method.

The Weighted Sum Method (WSM) is the earliest multi-dimensional decision making method (Triantaphyllou and Mann 1989). A value measurement method, this approach combines normalised scores for criteria in a linear model. WPM is a modification of WSM proposed by *P. Bridgman* in 1922 to overcome some of the weaknesses of the WSM approach (Bridgman 1922). WPM raises weights as powers of the criteria value (positive powers for benefits and negative powers for costs), eliminating any units of measure. The main benefit of this approach is that the different units do not require normalisation (Odu and Charles-Owaba 2013). As a result WPM is often referred to as dimensionless analysis (Triantaphyllou 2000).

In the 1970s, AHP was developed by *T. Saaty* as an alternative to these simplistic multi-dimensional models (Saaty 1987). AHP is a structured technique for organising and analysing complex decisions. AHP consists of an overall goal, a group of options or alternatives for reaching the goal, and a group of factors or criteria that relate the alternatives to the goal. The decision maker's preference between two alternatives are quantified on a scale of 1 to 9, with 1 representing no preference of x over y through to 9 representing a strong preference of x over y. These judgements are then used to synthesise an overall ranking of alternatives. This approach proved popular: by 2008 more publications applied AHP than any other MCDM method (Wallenius et al. 2008). AHP fits into the category of value measurement models, sometimes referred to as the American school of multi-criteria decision analysis (Lootsma 1990).

Alternatively, the French school was founded by *B. Roy*, who produced the series of ELECTRE (ELimination Et Choix Traduisant la REalité) methods (Roy 1990). This served as inspiration for the family of outranking methods, characterised by the limited degree to which a disadvantage on one criterion may be compensated by advantages in another. PROMETHEE is an outranking method developed by *J.P Brans* (Brans and Vincke 1985). PROMETHEE ranks a set of alternatives on the basis of several criteria by identifying pros and cons of the alternatives in a pair-wise fashion. Multiple versions of PROMETHEE have been introduced (Brans et al. 1986; Brans and Mareschal 1992, 1995). PROMETHEE I produces a partial ranking, whereas PROMETHEE II computes a complete ranking of alternatives.

Separately to the aforementioned schools of MCDM methods, we have goal, aspiration and reference models. The first goal, aspiration and reference model, TOPSIS, was developed in the 1980s by *Hwang and Yoon* (Hwang and Yoon 1981). TOPSIS defines a positive ideal solution which represent the best alternative. This imaginary solution is created by collecting the best possible values across all criteria. The same is done with the worst criteria values to create a negative ideal solution. The best alternative is then determined by minimising the euclidean distance from the positive ideal solution and maximising the euclidean distance from the negative ideal solution.

2.2. *Evaluating MCDM methods*

Evaluating MCDM methods is understood to be one of the most difficult problems in the field of decision analysis. *Zanakis et al.* (Zanakis et al. 1998) stated that it is impossible or difficult to answer questions such as: which method is more appropriate for what problem and what are the advantages or disadvantages of using one method over another. In this section we discuss literature attempting to answer these questions. Specifically, we outline the application of concern, the methods compared, and their evaluation metrics.

Zanakis et al. propose a simulation-based approach which compares the resultant ranking of an MCDM method to one generated through WSM, referred to in (Zanakis et al. 1998) as the Simple Additive Weighting method (SAW). In the absence of any other objective standard, WSM is chosen as a result of its simplicity and the popularity of the method at the time of publication. The paper evaluates rankings produced by WPM (referred to in the paper as Multiplicative Exponent Weighting), AHP, ELECTRE and TOPSIS. The methods are evaluated through a bundle of metrics, notably mean squared error of weights (MSEW), the same for ranks (MSER) and Spearman's correlation for ranks, comparing both the weights assigned to alternatives and the resultant rankings. The paper concludes that AHP behaves most similarly to WSM; with ELECTRE being the least similar; the rest of the methods fall between the two.

The same study also evaluated the frequency of rank reversal. Rank reversal, first observed by *Belton and Gear* (Belton and Gear 1983), refers to changes in the ranking of alternatives by addition or deletion of an alternative. This is an unintuitive and therefore undesirable characteristic of any MCDM method, as user understanding and faith in the results are imperative. The study introduces a new non-optimal alternative, then counts both how often the top ranked alternative remained the same, and the total number of ranks that are not altered as a percentage of the number of alternatives. This experiment finds that TOPSIS suffered the least from rank reversal, followed by AHP and ELECTRE respectively.

Selmi, Kormi and Ali (Selmi et al. 2016) compare the results of ELECTRE III, PROMETHEE I and II, TOPSIS, AHP and PEG in two case studies. The first study finds a similarity between PROMETHEE II and AHP, and observes the starkest difference between TOPSIS and ELECTREE III. The second study finds that ELECTREE III, PROMETHEE, and AHP agree on the first alternative being the best, while TOPSIS and PEG rank the third alternative as the highest. The study calculated the Gini Index to measure the rankings dispersion and uses the mean value of dispersion to perform comparisons between rankings. *Sarraf and McGuire* (Sarraf and McGuire 2020) highlight an issue with this approach as it does not consider that the top of the ranked list as being more important than the bottom. In the study AHP-PROMETHEE II switches one pair at the bottom of the ranked list, whilst TOPSIS-PROMETHEE II switches a pair at the top of the list, yielding the same mean value of the Gini index. They propose that a change in the ranks at the top of the list must have a higher negative impact on the evaluation.

Sarraf and McGuire (Sarraf and McGuire 2020) compare the results of AHP, Fuzzy AHP, TOPSIS, Fuzzy TOPSIS and PROMETHEE through two real-world transport based case studies. They evaluate the similarity of each method compared to AHP. AHP is chosen as the baseline as it is the most widely used method in the literature. They attempt to rectify their issue with equal weighting being given to changes regardless of position in the *Selmi, Kormi and Ali* paper

(Selmi et al. 2016) by employing average overlap (AO) and discounted cumulative gain (DCG) as evaluation metrics. AO is highlighted by *Webber, Moffat and Zobel* (Webber et al. 2010) as an approach to assign greater weighting to differences at the top of the list. It is based on simple set overlap where the overlap of the two rankings is compared at an incrementally increased depth. DCG was proposed by *Jarvelin and Kekalainen* (Jarvelin and Kekalainen 2000) and is used in information retrieval to evaluate the relevance of results returned by search engines. This metric is an alteration of cumulative gain (CG) that discounts scores of alternatives using a logarithmic discount function as their rank increases. The paper finds that the PROMETHEE ranking fits well with the AHP ranking and these two methods produce the best results. They found that fuzzy AHP produced acceptable results and that TOPSIS and fuzzy TOPSIS produced poor ranking results. The writers recommend AHP as a simple and robust method for the transportation field.

The limitation of *Sarraf and McGuire's* (Sarraf and McGuire 2020) approach is that both DCG and AO only compare the rankings generated by each method to those generated through AHP. Unlike the study by *Zanakis et al.* (Zanakis et al. 1998), which measured the prevalence of rank reversal, no other characteristics of each method were analysed. In our research we propose five desiderata, of which two - (*(4) consistent trade-offs between criteria and (5) high stability of results*) - can be used as a basis to evaluate the suitability of MCDM methods for dynamic decision making problems.

The above studies assess decision support methodologies without focusing on a specific application. However given some context, it can be easier to assess different decision support methodologies. For example, in clinical DSS some contextual information can be applied to make more effective evaluations of different facets of DSSs. A study (Kawamoto et al. 2005), in clinical decision support has investigated validity of decision support features by assessing the improvement in clinical practice. This assessment was enabled through the measurement of patient outcomes or process measures. The study summarised 82 relevant comparisons of which 71 compared a clinical DSS with a control group (control-system comparisons) and 11 directly compared a system with the same system plus extra features (system-system comparisons). Another study (Roshanov et al. 2013), conducted a similar assessment with 162 randomised control trials. The above studies collate data from a multitude of papers, assessing features for purpose-built clinical DSS. This type of study is enabled in clinical decision support by the presence of clear success criteria (patient outcomes), to compare against the presence of decision support features. These studies do not assess decision support methodologies but instead look at context features of clinical DSSs such as "Use of a computer to generate the decision support" or "Automatic provision of decision support as part of clinician workflow".

A recent study (Vasey et al. 2022), applied a Delphi process with 127 experts in the first round and 138 experts in the second round to create an evaluation framework for clinical DSS. These experts were selected from 20 pre-defined stakeholder categories such as administrators/hospital management, allied health professionals, clinicians, engineers/computer scientists etc. The outcome guidance advised the use of patient outcomes as the primary method for the assessment of clinical DSS, with the reporting of disagreements between the system and the computer as a secondary measure of system performance.

These studies evaluate a diverse array of different features of DSSs but not the MCDM method underlying the ranking process for results. Still, the choice of MCDM is just another feature of a DSS that can be evaluated in the same way. As shown in these studies, when evaluating DSSs for a specific context, the outcomes of a scenario become a powerful tool for assessment. In our case, this motivates the use of scenario outcomes, through our simulations, for the assessment of consistent trade-offs between criteria (*Desiderata 4*).

2.3. Situational awareness with drones

The unmanned aerial vehicle task assignment problem (UAVTAP) consists of finding an optimal assignment of UAVs to a set of tasks (Coutinho et al. 2018). In this section we briefly discuss previous research in the area and their approach to solving UAVTAP as an MCDM problem.

Ries and Ishizaka (Ries and Ishizaka 2012) present a multi-criteria support system for a dynamic UAVTAP. Their case study employs UAVs for surveillance to investigate ships in a maritime environment. The approach applies AHP to calculate weightings for PROMETHEE, ranking ships from least to most suspicious as a means of facilitating an efficient priority for surveillance. The system then assigns a UAV to investigate the most suspicious ship. This approach differs from our own as it calculates only the next ship to be visited, whereas in our own system we attempt to qualify an entire tour, limited by the fuel of the drone. This approach allows us to plan further ahead to find more fuel and time efficient routes. As a result of planning further ahead, our case study features an exponentially larger set of possible solutions. To explore this large set of possible solutions, we integrate a genetic algorithm as a meta-heuristic search function.

Ramerirez *et al.* first formulated UAVTAP as a Constraint Satisfaction Problem (CSP), with the mission being modelled and solved using constraint satisfaction techniques (Ramirez-Atencia et al. 2014a). In a later paper (Ramirez-Atencia et al. 2014b), UAVTAP is formulated as a multi-objective optimisation problem. Their objective consists of minimising the number of drones employed, the total flight time, the total fuel, the total distance, total cost and the time taken. The original approach was combined with a MOEA (Ramirez-Atencia et al. 2017a); this algorithm provides an estimate of the Pareto Optimal Front (POF), i.e. the set of all non-dominated solutions of the problem. A solution s_1 is dominated by s_2 if s_1 is not better in any objective and s_2 is better in at least one. A solution is non-dominated if it is not dominated by any solution in the solution set. Their conclusion is that, as the complexity of the mission increases the number of solutions in the POF becomes huge, and therefore the time needed to calculate the complete POF becomes intractable.

To improve on this approach, Ramirez *et al.* (Ramirez-Atencia et al. 2017b) introduce a Knee-Point MOEA intending to reduce the POF to a set of the most likely best solutions. This reduces the size of the POF from hundreds to tens of solutions. This however still leaves a difficult task for decision makers who must select the most appropriate mission plan. In a later paper, the authors rank the outputted POF using a selection of MCDA algorithms according to user preferences (Ramirez-Atencia et al. 2020). The work assumes that the decision maker cannot provide *a priori* information regarding preferences on criteria. A limitation of this approach is that it can become costly in a dynamic setting, as the POF must be recalculated each time the mission scenario is updated. In a later paper, the authors attempt to improve on this approach by introducing more diversity into the decision space by replacing the crowding distance metrics from NSGA-II with a combination of Manhattan and crowding distance metrics (Javadi et al. 2021). Coelho *et al.* (Coelho et al. 2017) also proposed a multi-objective UAVTAP. Taking inspiration from a multi-criteria view of real systems, the approach considers seven different objective functions which it seeks to minimise using a Mixed-Integer Linear Programming model solved by a matheuristic algorithm. This produces an estimate of the POF but the paper does not attempt to rank the non-dominated solutions.

In our work, we apply the MCDM methods as a fitness function of a genetic algorithm, rather than applying an MOEA. We take this approach as it is infeasible to calculate a POF in a dynamic setting. Our approach is enabled by the fact that the decision maker's preferences are available prior to route calculation. Using a genetic algorithm with MCDM methods as a fitness function allows us to maintain a solution set ranked according to preferences, as the situation and therefore solutions, evolve over time.

3. Multi-Criteria Decision Making Methods

In this section we detail how we applied each of the MCDM algorithms, including any adaptations we made to fit the case study.

3.1. Weighted Product Model

WPM scores alternatives by a simple multiplicative model. The algorithm involves the following steps:

WPM Step 1. All criteria values for alternatives are normalised by applying Equation 1, where minX and maxX are the smallest and largest criteria values respectively.

$$Norm(x) = \frac{x - minX}{maxX - minX} \tag{1}$$

WPM Step 2. Weights are used to score each alternative using Equation 2. The criteria value of alternative a is represented as a_j for criterion j. The weighting for criterion j is given as w_j. For criteria we seek to minimise, we replace w_j with $-w_j$.

$$WPM(a) = \prod_{j=1}^{n} (a_j)^{w_j} \tag{2}$$

WPM Step 3. Rank alternatives according to the value of $WPM(a)$.

WPM Complexity. WPM can be computed in time $O(nm)$ where n represents the number of solutions and m represents the number of criteria.

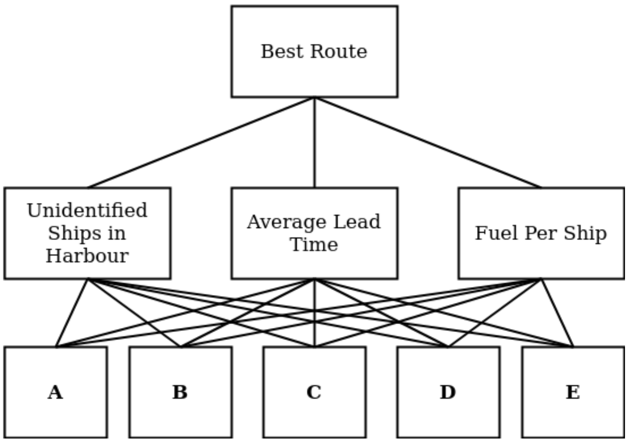

Figure 1: AHP hierarchy structure for the harbour management task

3.2. Analytic Hierarchy Process

AHP is implemented in the following steps:

AHP Step 1. The problem is modelled as a hierarchy. The goal of the problem is at the highest level, with the criteria below it. These criteria can be divided further into sub-criteria, then at the lowest level we have the alternatives. Figure 1 shows the hierarchy for the harbour management task.

AHP Step 2. Criteria values are normalised according to the range of values across all alternatives using the formula given in Equation 1, where minX and maxX are the smallest and largest criteria values respectively.

AHP Step 3. Priorities are established across the hierarchy by constructing a pairwise comparison matrix for each level. In our experiments, we evaluate a fixed set of weights and therefore it is only alternatives that require comparison. To produce a ranking, criteria values must be scored. The normalised values are compared pairwise to generate a comparison matrix. For three alternatives a_1, a_2 and a_3 and a criterion X with normalised criteria values x_1, x_2, x_3, we would generate a comparison matrix C.

$$\mathbf{C} = \begin{array}{c} \\ a_1 \\ a_2 \\ a_3 \end{array} \begin{array}{c} \begin{array}{ccc} a_1 & a_2 & a_3 \end{array} \\ \begin{bmatrix} 1 & f(x_1, x_2) & f(x_1, x_3) \\ f(x_2, x_1) & 1 & f(x_2, x_3) \\ f(x_3, x_1) & f(x_3, x_2) & 1 \end{bmatrix} \end{array}$$

Equation 3 is applied to compare criteria values. This formula maps two normalised values (x, y) to the fundamental scale proposed by Saaty (Saaty 1987).

$$f(x,y) = e^{x-y} \tag{3}$$

AHP Step 4. Comparison matrices generated through AHP have a concern with departure from consistency between judgements. When a matrix is inconsistent, the resultant vector of relative weights is viewed as untrustworthy. Therefore, the next step in AHP is to check the consistency of the matrix. In general practice, this is done by calculating the consistency ratio (CR) and discarding matrices with a CR greater than 0.1 (Siraj et al. 2015). A matrix $C = (c_{ij})$ is termed *consistent* (Basile and D'Apuzzo 2006) if:

$$c_{ij} \cdot c_{jk} = c_{ik} \ \forall \ i, j, k = 1, 2, ..., n$$

It is unnecessary to check consistency in our application of AHP as the comparison formula $f(x)$ produces comparison matrices that satisfy this equation for all values of i, j, k as shown below.

$$f(i,j) \cdot f(j,k) = f(i,k)$$
$$e^{i-j} \cdot e^{j-k} = e^{i-k}$$
$$e^{i-k} = e^{i-k}$$

AHP Step 5. The principal eigenvector P_j of the comparison matrix for each criterion j is calculated; this vector represents the priorities for each alternative. This priority vector P_j is then multiplied by the weighting w_j of each criterion j and summed to produce a global score vector G. For a problem with m alternatives and n criteria, the formula is given by Equation 4.

$$G = (g_i)_m = \sum_{j=1}^{n} P_j \cdot w_j \tag{4}$$

AHP Step 6. Rank alternatives according to their global score. For alternative i, the global score is entry g_i in G.

AHP Complexity. AHP can be computed in time $O(n^2 m)$ where n represents the number of solutions and m represents the number of criteria (Dewi et al. 2018).

3.3. TOPSIS

TOPSIS is described by the following steps:

TOPSIS Step 1. Create an evaluation matrix $(x_{ij})_{m \times n}$ consisting of m alternatives and n criteria, with the criteria values for each alternative i and criterion j given as x_{ij}.

TOPSIS Step 2. Calculate the normalised evaluation matrix $R = (r_{ij})_{m\times n}$ by applying the formula given in Equation 5.

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{k=1}^{m} x_{kj}^2}},\ \ i = 1, 2, \ldots, m,\ \ j = 1, 2, \ldots, n \tag{5}$$

TOPSIS Step 3. Calculate the weighted normalised decision matrix $\mathrm{T} = (t_{ij})_{m\times n}$ by applying the formula from Equation 6.

$$t_{ij} = r_{ij} \cdot w_{ij},\ \ i = 1, 2, \ldots, m,\ \ j = 1, 2, \ldots, n \tag{6}$$

TOPSIS Step 4. Compute the positive (A^+) and negative (A^-) ideal solutions. These serve as imaginary perfect and worst points in the solutions space, from which we can calculate the distance from real solutions as a form of evaluation.

$$A^+ = \{x_1^+, x_2^+, \ldots, x_n^+\}$$
$$\text{where } x_j^+ = \{max(x_{ij}) \text{ if } j \in B; min(x_{ij}) \text{ if } j \in C)\}$$

$$A^- = \{x_1^-, x_2^-, \ldots, x_n^-\}$$
$$\text{where } x_j^- = \{min(x_{ij}) \text{ if } j \in B; max(x_{ij}) \text{ if } j \in C)\}$$

where B is associated with benefit criteria (values we seek to maximise) and C with cost criteria (values we seek to minimise).

TOPSIS Step 5. Calculate the L^2-distance from positive ideal (d_i^+) and negative ideal (d_i^-) solutions for each alternative.

$$d_i^+ = \sqrt{\sum_{j=1}^{n} (x_{ij} - x_j^+)^2},\ \ i = 1, 2, \ldots, m$$
$$d_i^- = \sqrt{\sum_{j=1}^{n} (x_{ij} - x_j^-)^2},\ \ i = 1, 2, \ldots, m$$

TOPSIS Step 6. Calculate the similarity to the worst condition for each alternative (s_i^-).

$$s_i^- = \frac{d_i^-}{d_i^- + d_i^+},\ \ i = 1, 2, \ldots, m$$

TOPSIS Step 7. Rank the alternatives according to the similarity to the worst condition (s_i^-).

TOPSIS Complexity. TOPSIS can be computed in time $O(n^2m)$ where n represents the number of solutions and m represents the number of criteria (Hamdani 2016).

3.4. PROMETHEE

In our work, PROMETHEE is defined by the following procedure:

PROMETHEE Step 1. Pairwise comparisons $d_j(x, y)$ are made between each criteria value x_{ij} for alternative i and criterion j using Equation 7.

$$d_j(x_{ij}, x_{kj}) = x_{ij} - x_{kj} \tag{7}$$

PROMETHEE Step 2. Unicriterion preference degree is calculated by applying a preference function $P(x)$ to the difference as shown in Equation 8.

$$\pi_k(x_{ij}, x_{kj}) = P[d_k(x_{ij}, x_{kj})] \tag{8}$$

This function can be different for each criterion. Six types of preference function are proposed; usual criterion, quasi criterion, criterion with linear preference, level criterion, V-shape with indifference criterion, and Gaussian criterion (Brans et al. 1986). In this work, the criterion with linear preference is applied, as shown in Equation 9.

$$P(x) = \begin{cases} 0 & x \leq 0 \\ x & 0 < d \leq 1 \\ 1 & d > 1 \end{cases} \tag{9}$$

PROMETHEE Step 3. A multi-criteria (global) preference degree $\pi(x, y)$ is computed to globally compare every pair of alternatives as shown in Equation 10.

$$\pi(x_{ij}, x_{kj}) = \sum_{k=1}^{q} \pi_k(x_{ij}, x_{kj}) \cdot w_k \tag{10}$$

PROMETHEE Step 4. Calculate the positive ($\phi^+(a)$) and negative ($\phi^-(a)$) preference flows for each alternative.

$$\phi^+(a) = \frac{1}{n-1} \sum_{x \in A} \pi(a, x)$$

$$\phi^-(a) = \frac{1}{n-1} \sum_{x \in A} \pi(x, a)$$

PROMETHEE Step 5. Calculate the net preference flow $\phi(a)$.

$$\phi(a) = \phi^{+}(a) - \phi^{-}(a)$$

PROMETHEE Step Six. Rank alternatives according to the net preference flow.

PROMETHEE Complexity. PROMETHEE can be computed in time $O(mnlog(n))$ where n represents the number of solutions and m represents the number of criteria (Calders and Van Assche 2018).

4. Situational Awareness Scenario

To illustrate how our desiderata for dynamic multi-criteria decision support can be implemented, we consider an application of drones to a harbour management task. This application is then used as part of our evaluation of various MCDM methods with respect to our desiderata. In this section, we outline the harbour management task and its requirements in relation to our desiderata. We then describe the criteria used for the selection of routes and the scenarios used for our evaluation.

4.1. Harbour management task

In this task, a decision maker takes the role of a harbour master managing a harbour. The harbour master controls a single drone to identify ships close to the harbour zone. The job of the user is to select a route for the drone, with a view to identifying ships before they reach the harbour zone. The length of these routes is limited by the fuel of the drone. Once the drone has run out of fuel, it must return to the refueling point, located within the harbour zone.

We assume that the most suitable route may depend on different criteria: *unidentified ships in harbour*, *lead time* and *fuel per ship*.

- **Unidentified Ships In Harbour** - The number of unidentified ships which will arrive in the harbour over the course of the route.
- **Average Lead Time** - The average amount of time between a ship being identified and arriving in the harbour.
- **Fuel Per Ship** - The amount of fuel used by the drone per ship identified.

As with all MCDM problems, these criteria have different values for different decision makers, so we require a specification of preferences (Desiderata 1). In our case, criteria are predicted values based upon the current trajectory of ships in the area surrounding the harbour. Ships approaching a harbour can quickly change direction, causing the criteria values of a route to change. This has an impact on the ranking of options, necessitating dynamic revision of recommendations (Desiderata 2). It is also worth noting that a decision maker may have knowledge outside the scope of the system. An example for harbour management could be multiple ships that do not require identification. If every route visits these ships then the decision maker is left to manually plot a route, rendering the system useless. This suggests that a diverse set of routes would be desired (Desiderata 3).

4.2. Criteria

In this section we describe the motivation and calculations for each of the criteria used for route selection in our harbour management task.

4.2.1. Unidentified Ships in Harbour

The primary goal of the situational awareness task is to identify ships before they reach the harbour. As a result, the first criterion is the number of unidentified ships that will arrive in the harbour over the course of the route. The system calculates the length of time needed to complete a route, then uses the current trajectory and speed of all boats to calculate which of the unidentified ships will reach the harbour before the drone returns.

4.2.2. Average Lead Time

For situational awareness, information gained sooner is more valuable. If a ship is identified seconds before it passes the threshold into the harbour, there is no time to respond to the gathered information. As such, it is important to maximise the time between a ship's identification and its arrival in the harbour. We call this metric *lead time*. The system takes the speed and trajectory of each ship within a route to predict the lead time for each. The average of these values is calculated and used as the criterion *average lead time*.

4.2.3. Fuel Per Ship

An important part of managing any operation is minimising cost. For this task, that means reducing the amount of drone fuel we consume. To maximise efficiency, another objective is therefore to minimise the fuel cost per ship visited. For our simulation the drone burns a fixed amount of fuel per distance travelled. To compute a value for this criterion, we first determine the total fuel required for a route, by calculating the total distance of the route divided by the fuel efficiency of the drone. We calculate the distance of the route by predicting an intercept point for each ship then summing the distance between each journey plus the distance to return to harbour. The total fuel is then divided by the number of ships identified, giving us *fuel per ship*.

4.3. Scenarios

To analyse and evaluate our desiderata we have ten scenarios which simulate traffic in harbours around Great Britain and Ireland. The set of scenarios comprises simulations of the following locations:

1. Edinburgh
2. Liverpool
3. Dublin

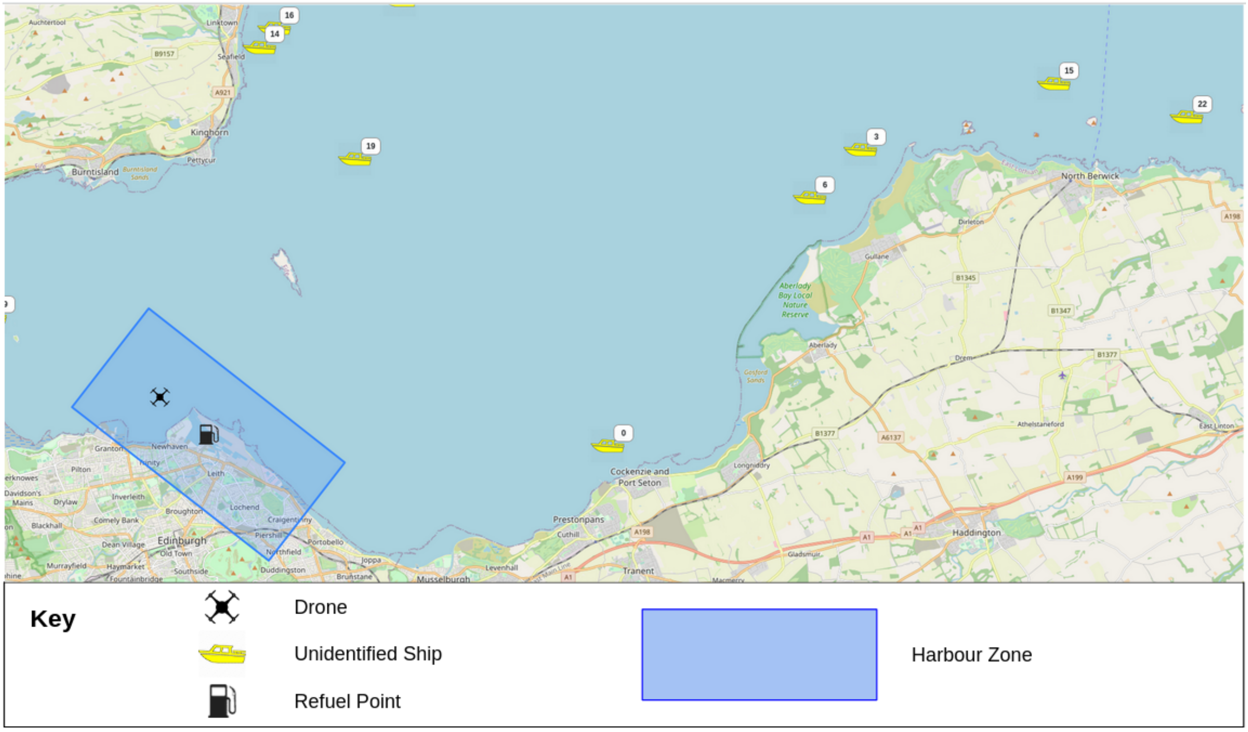

Figure 2: A visualisation of the Edinburgh scenario (colour print)

4. Belfast
5. Portsmouth
6. Plymouth
7. Oban
8. Douglas
9. Dover
10. Hull

A scenario is defined by a set of 600 points in time and the position of all ships at each point. To maintain comparability between scenarios, each simulation includes 30 ships, of which 20 arrive at the harbour. The gap between each time step is equivalent to 30 seconds, therefore the entire simulation plots the routes of ships over a duration of 5 hours. We chose a five hour window of time for each scenario as this represents a sensible maximum flight time for a harbour management drone (based on domain expertise). This allows the drone time to identify roughly 20/30 of the ships. Therefore, if the correct route is picked, the drone should be able to achieve a perfect score. Unfortunately, due to the ships changing directions and speed, this route is impossible to predict every time, and often ships will arrive in the harbour before the drone reaches them or ignored ships will change path towards the harbour. Figure 2 shows a screenshot of the Edinburgh scenario visualised through a web interface.

4.4. *Criteria Correlations*

For a multi-criteria decision making problem to be interesting, it is important to have criteria that do not correlate strongly or have a negative correlation. For example, when purchasing a car, buyers may choose to minimise cost whilst maximising top speed. These criteria are correlated but conflicting, producing a need for trade-offs between objectives and therefore a need for MCDM methods. If the criteria are too strongly correlated then often all objectives can be maximised at the same time. As a consequence, strongly correlated criteria would be unfit for evaluating the capabilities of a MCDM system.

4.4.1. *Experimental Set Up*

To analyse the suitability of our criteria, we designed an experiment to calculate the Pearson Correlation between pairs of criteria. We generated 20 routes across each of the 10 scenarios outlined in Subsection 4.3 using the criteria weightings given in Table 1 for a total of 1400 generated routes. Each of these routes has a criteria value for *Unidentified Ships in Harbour*, *Average Lead Time* and *Fuel Per Ship*. We then calculated the Pearson Correlation Coefficient between each pair of criteria values, with the results given in Table 2.

Table 1: Criteria weightings - USH: Unidentified Ships in Harbour; ALT: Average Lead Time; FPS: Fuel Per Ship

Weighting	USH	ALT	FPS
A	1	0	0
B	0	1	0
C	0	0	1
D	0.5	0.5	0
E	0.5	0	0.5
F	0	0.5	0.5
G	0.33	0.33	0.33

4.4.2. *Results*

Before analysing the results given in Table 2, it is worth noting that *Unidentified Ships in Harbour* and *Fuel Per Ship* are both criteria we seek to minimise whilst maximising the *Average Lead Time* as stated in Subsection 4.2. With this in mind, the following can be observed:

- *Average Lead Time* and *Fuel Per Ship* have a high degree of correlation, but are conflicting.
- *Unidentified Ships in Harbour* and *Fuel Per Ship* have a moderate degree of correlation.
- *Average Lead Time* and *Unidentified Ships in Harbour* have a low degree of correlation.

The highest correlation occurs between *Average Lead Time* and *Fuel Per Ship*. These two criteria are highly correlated because *Fuel Per Ship* is low when a route identifies groups of ships which are close to the harbour zone and therefore fuel efficient. For these kinds of routes, the *Average Lead Time* is also low as only a short amount of time passes between the identification of ships and their arrival in the harbour. Fortunately, these criteria are conflicting as one is a cost (*Fuel Per Ship*) and the other is a benefit (*Average Lead Time*), therefore the correlation is acceptable.

The second most correlated pair is *Unidentified Ships in Harbour* and *Fuel Per Ship*. These two criteria are correlated as identifying many ships which are close to the harbour often includes ships which are most likely to pass into the harbour zone. The pair are not highly correlated though as the most fuel efficient route often includes ships which are moving away or are otherwise unlikely to enter the harbour. *Average Lead Time* and *Unidentified Ships in Harbour* also have a weak correlation, explained as a transitive effect of the other correlations. Overall, the set of criteria are suitable as it is impossible to simultaneously minimise *Unidentified Ships in Harbour* and *Fuel Per Ship* whilst maximising *Average Lead Time*.

Table 2: Criteria Pearson correlation coefficient - USH: Unidentified Ships in Harbour; ALT: Average Lead Time; FPS: Fuel Per Ship

Criteria	USH	ALT	FPS
USH	1		
ALT	0.254679	1	
FPS	0.434399	0.644990	1

Criteria Objectives - Maximise: Unidentified Ships in Harbour, Fuel Per Ship; Minimise: Average Lead Time

5. Multi-Criteria Dynamic Genetic Algorithm

In our work, we apply MCDM methods as a fitness function, rather than applying a MOEA. This approach is enabled by the fact that the decision makers preferences are available prior to route calculation. The approach allows us to maintain a solution set ranked according to preferences *(Desiderata 1)*, via a continuously running evolutionary algorithm, as the situation and therefore solutions, evolve over time *(Desiderata 2)*. In this section we describe the genetic algorithm applied for route selection, how it fits into our streaming genetic algorithm architecture, and our approach to encourage diversification of results *(Desiderata 3)*.

5.1. Genetic Algorithm for Route Selection

In this section we describe the genetic algorithm for the harbour management task. We include evaluation of the optimal number of generations to compute for future experiments.

5.1.1. Chromosome encoding of Routes

The routes for the drone in the harbour management task can be represented by Chromosomes that consist of genes. Each gene of the Chromosome represents a ship to be identified. The routes can therefore be modelled as follows:

$$\text{Let S be the set of ships to be visited,}$$
$$\text{For } s_i \in S,$$
$$r = (s_1, s_2, .. s_n)$$

For the harbour management task, this chromosome represents the route

$$s_1 \rightarrow s_2 .. \rightarrow s_n$$

To ensure the validity of the route the following conditions are needed:

1. Each ship may only be visited once.
2. The total distance of the route must be less than the range of the drone.
3. A route must visit at least one ship.

5.1.2. Fitness Function

The fitness function is used to evaluate the quality of the route. As this is a MCDM problem, a MCDM algorithm is used calculate the fitness. This MCDM method could be one of AHP, WPM, TOPSIS or PROMETHEE, outlined in Section 3. The method is given a set of criteria weights, which are used to calculate a score for each solution. A diversity discount is then applied to each solution (described in Section 5.4) and the scores are normalised with the following formula:

$$Norm(x) = \frac{x - minX}{maxX - minX}$$

5.1.3. Mutate Function

The mutation operator exists to maintain the diversity of the population. Generally, for travelling salesmen problems, mutate is defined as a swap operation, swapping the order of two cities in a route. This is because in the travelling salesmen problem all cities must be visited in a valid route. In the harbour management task the drone can only visit a portion of the ships, limited by fuel. As a result we define mutate as a combination of two functions; *add a ship* (1) and *remove a ship* (2). When a route is chosen to be mutated, one of the two functions is applied with equal chance of each. The two functions are defined below:

$$\text{Let S be the set of ships to be visited,}$$
$$\text{Let } r = (s_1, .. s_n),\ s_x \in S, s_x \notin r,$$

$$A(r) = (s_1, s_x .. s_n) \tag{11}$$

Such that the position of s_x is selected at random.

$$R(r) = (s_1, .. s_n) \tag{12}$$

5.1.4. Crossover Function

The crossover operator combines two selected routes together. For our the harbour management task the crossover operator is a single-point crossover function $c(x, y)$ defined by the following steps.

1. Two parents are selected from the population as parents; denoted as x and y.

$$\text{For } s_i, t_i \in S,$$
$$x = (s_1, s_2, .. s_n), y = (t_1, t_2, .. t_m)$$

2. A sub-path z is selected from parent x at a random point j of length $k < n$.

$$z = (s_j, s_{j+1} .. s_{j+k})$$

3. All ships from sub-path z are removed from parent y except the first ship in the path. This produces an intermediate path y_1

$$\text{For } s_j = t_2, s_{j+1} = t_4, s_{j+k} = t_6,$$
$$y_1 = (t_1, t_3, t_5, t_6 .. t_m)$$

4a. The path is merged into parent y at the point of the first ship in the path.

$$\text{For } s_j = t_2,$$
$$c(x, y) = (t_1, s_j, s_{j+1}, .. s_{j+k}, .. t_m)$$

4b. If the first ship in the path doesn't exist in y then add the path to the end of y.

$$\text{For } s_j \notin y,$$

$$c(x, y) = (t_1, .. t_m, s_j, s_{j+1}, .. s_{j+k},)$$

5.1.5. Solution Selection

The selection operator is used to select individuals from the last generation. To keep a high quality set of solutions, individuals are selected based on their fitness value. This fitness value is calculated by applying one of the MCDM methods described in Section 3. The route with the highest fitness is first selected for the next generation. To improve the diversity of the population, we then apply a fitness penalty to all other routes based upon their overlap with the chosen route. Another route is then chosen and the process is repeated with all other routes being discounted according to the overlap with the set of selected routes. This process continues until enough routes have been selected for the population.

The discount function includes a diversity weighting W. This weighting can be set to different levels to control to what extent overlapping routes are discounted to promote diversity. A value of 0 for W means that no discount is applied. The discount function $d(x)$ is described in Equation 13.

Let $f(r)$ be the fitness of route r,

R refers to the set of selected routes,

$o(r, R)$ be the overlap between r and R,

$$d(r) = e^{W\, o(r,R)} f(r) \tag{13}$$

This discount formula calculates the overlap between a route r and the selected routes R using the formula described in Equation 14.

$|R|$ refers to the cardinality of R,

$|r|$ refers to the numbers of ships in r,

$N(R, s)$ is the number of routes in R visiting ship s,

$$o(r, R) = 1 - \frac{1}{|r|} \sum_{s}^{r} \frac{N(R, s)}{|R|} \tag{14}$$

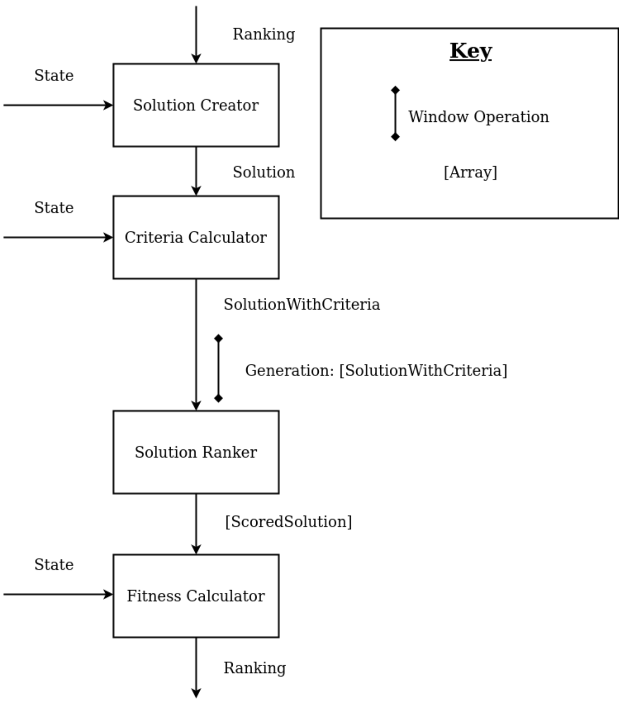

Figure 3: The flow of data through the continuous genetic algorithm

The discount function $d(r)$ is introduced in an attempt to improve diversity (*Desiderata 3*). We evaluate the effect changing the diversity weighting W has on diversity and the impact on quality of solutions in Subsection 5.4.

5.2. Streaming Genetic Algorithm

Data streams exist as an abstraction to support analysis of dynamic data as it is produced (Muthukrishnan 2005). DSSs exist to support users in navigating a space of options (Greco et al. 2016). Our streaming MCDM genetic algorithm was designed to bring together these complementary paradigms to support decision making with dynamic data. Current practice in stream data processing makes extensive use of Stream Processing Engines (SPEs) which provide a framework for acting upon elements in a stream. Apache Flink is an SPE developed by the Apache Software Foundation that we chose to implement our streaming genetic algorithm (Carbone et al. 2015).

Figure 3 shows the flow of data through the algorithm. The *Solution Creator* takes the current state of the scenario and the previous ranking, applying mutate(x) and crossover(x,y) operations to create new solutions. The details of these mutate/crossover functions for the harbour management task are given in Subsection 5.1. For the first generation, it creates a random set of solutions. The *Criteria Calculator* then calculates the criteria values for each route. In streaming, it is common to perform set-based aggregations over subsets of events that fall within a period of time.

This set of elements is referred to as a window. We take a window of recently validated routes in the *Solution Ranker*, and aggregate them into a ranking.

To produce a ranking, criteria values must also be scored. This is done by applying one of the MCDM methods described in Section 3. Initial calculations for each algorithm are applied within the solution ranker, by producing scores for each criterion. These intermediary values and criteria weightings are then passed to the *Fitness Calculator*. Within the *Fitness Calculator*, solutions are assigned a fitness value calculated using the MCDM method. When selecting routes for the next generation of the algorithm, we first chose the route with the highest fitness. We then applied a fitness penalty to routes that overlap with the set of routes chosen for the next generation. This was intended to promote a diverse set of routes for recommendation. The process is described in detail in Subsection 5.4. The application of the MCDM method for this procedure dominates the rest of the steps, giving rise to an overall complexity shown in Equation 15.

r represents the number of routes,

c refers to the number of criterion,

$O(rc)$ represents the complexity of the MCDM method employed as the fitness function

g refers to the number of generations,

$$gO(rc) \tag{15}$$

5.3. *Number of Generations Experiment*

In this section we measure the convergence of the population in our genetic algorithm to determine an appropriate number of generations for further experiments.

5.3.1. *Evaluation Methodology*

To determine the number of generations to run for future experiments, we measured the convergence of our population to a stable set of solutions. For this experiment we applied our genetic algorithm with WPM as the fitness function, to the first time-step of the 10 scenarios, with each of the 7 weightings outlined in Table 1. The population size for the genetic algorithm is chosen to be 30 solutions. We iterate the algorithm given in Subsection 5.2 and at each generation we output the current ranking of solutions in the population.

To quantify the similarity between rankings of subsequent generations, we utilise the evaluation metric AO discussed by *Sarraf and McGuire* (Sarraf and McGuire 2020) and *Webber et al.* (Webber et al. 2010). AO is a similarity measurement algorithm that assigns more weights to the top of the list. It is based on simple set overlap, where the user compares the overlap of the two rankings at incrementally increasing depths. The AO formula is given in Equation 16. The AO overlap between generations is given with the results shown in Figure 4.

P_t is the population at generation t

P_{t+1} is the population at generation $t+1$

k is the evaluation depth,

d is the depth.

$$AO(P_t, P_{t+1}, k) = \frac{1}{k} \sum_{d=1}^{k} \frac{|P_t \cap P_{t+1}|}{d} \tag{16}$$

5.3.2. Results

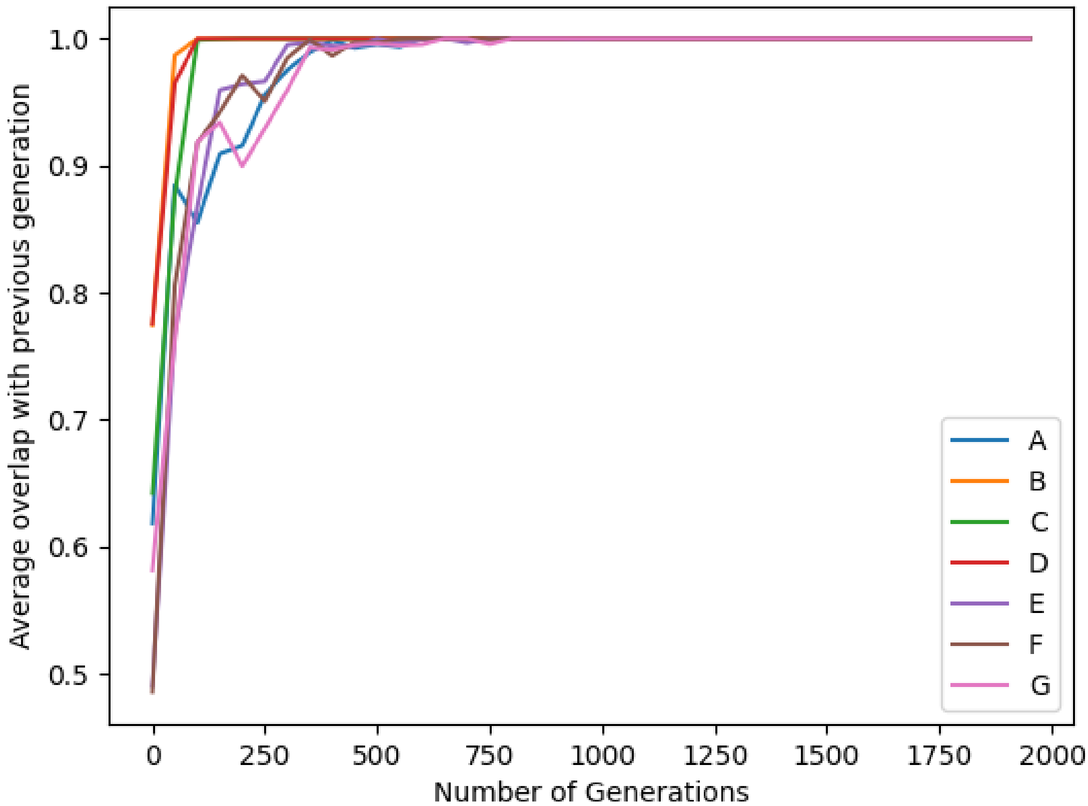

Figure 4: Average Overlap Between Generations (colour print)

The results show that different criteria weightings (as depicted in Table 1) converge at different times. The difference between these weightings can be explained by observing that weightings C, E, F and G include a non zero weighting for *Unidentified Ships in Harbour*. This criterion favours longer routes, which naturally creates a larger search space of potentially optimal routes than *Average Lead Time* which is optimal for short routes. *Unidentified Ships in Harbour* also takes longer to optimise than *Fuel per Ship*, because the heuristic used to initialise the population is most similar to *Fuel per Ship*.

As shown in Figure 4, weightings A, B and D converge the fastest, achieving an AO between populations of 1 after 103 generations. Weightings C, E, F and G converge more slowly, achieving an AO of 1 between populations after 789 generations. Therefore, 800 generations is chosen as the appropriate parameter for later experiments.

5.4. *Diversification of Options*

In this section, we analyse our approach to enabling a high diversity of options (*Desiderata 3*). We evaluate the effect of different diversity weightings on the discount function outlined in Subsection 5.1.5.

5.4.1. *Evaluation Methodology*

To evaluate the effect of the diversity discount function (shown in Equation 13), we generated rankings using different values for the diversity weighting W. A ranking was generated for each of the 10 scenarios, using each weighting from Table 1. The ranking was generated by running the genetic algorithm for 800 generations, using WPM as the fitness function.

To calculate the effect of W, diversity was measured for each of the resultant rankings. To calculate diversity, we removed each route from the ranking then calculated the overlap (given in Equation 14) between the route and the routes remaining in the ranking. Diversity d was then calculated as the average of these overlap values as shown in Equation 17.

R is the ranking

$|R|$ is the number of routes in the ranking

r_i is a route in the ranking,

$R \setminus r_i$ is the ranking without r_i.

$$\mathrm{d}(R) = \frac{1}{|R|} \sum_{i=1}^{|R|} o(r_i, R \setminus r_i) \tag{17}$$

It is also important to note that changing the fitness function has an effect on the quality of the solutions in a ranking. To measure this effect, we recorded the criteria values for each criterion. The average of these criteria values across the rankings are compared as the diversity weighting W changes. When analysing criteria values, we plot only

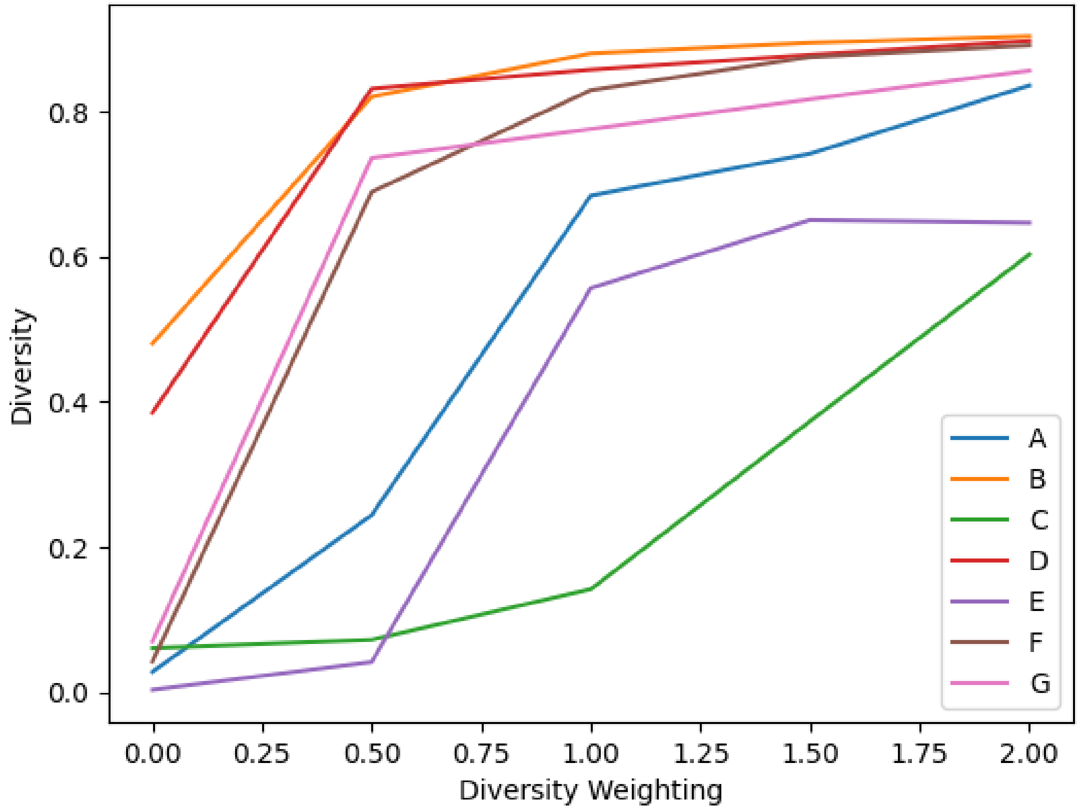

Figure 5: Effect of the diversity weighting on diversity of rankings (colour print)

weightings that include a non-zero weight for the respective criterion. We analyse only these weightings because maximising a specific criterion is only an objective of the algorithm when the weighting is non-zero.

5.4.2. Results

Figure 5 shows the effect of diversity weighting w on diversity. For each criteria weighting, increasing w increases the diversity of the rankings. It can also be observed that the criteria weighting has a significant effect on the diversity. Criteria weightings C and E both have lower diversity, whereas weightings B and D both have higher diversity. This is because C and E are highly weighted towards *distance per ship*, and this criterion favours longer routes which therefore overlap more frequently. Whereas B and D have a high weighting for *average lead time* which favours shorter routes.

Figures 6, 7 and 8, show the effect of W on criteria values for *Unidentified Ships in Harbour*, *Average Time to Arrival* and *Fuel per Ship*, respectively. *Unidentified Ships in Harbour* is a (*cost*) criterion we seek to minimise, *Fuel per Ship* is another cost criterion. Whereas, we seek to maximise values for the *Average Time to Arrival* because it is

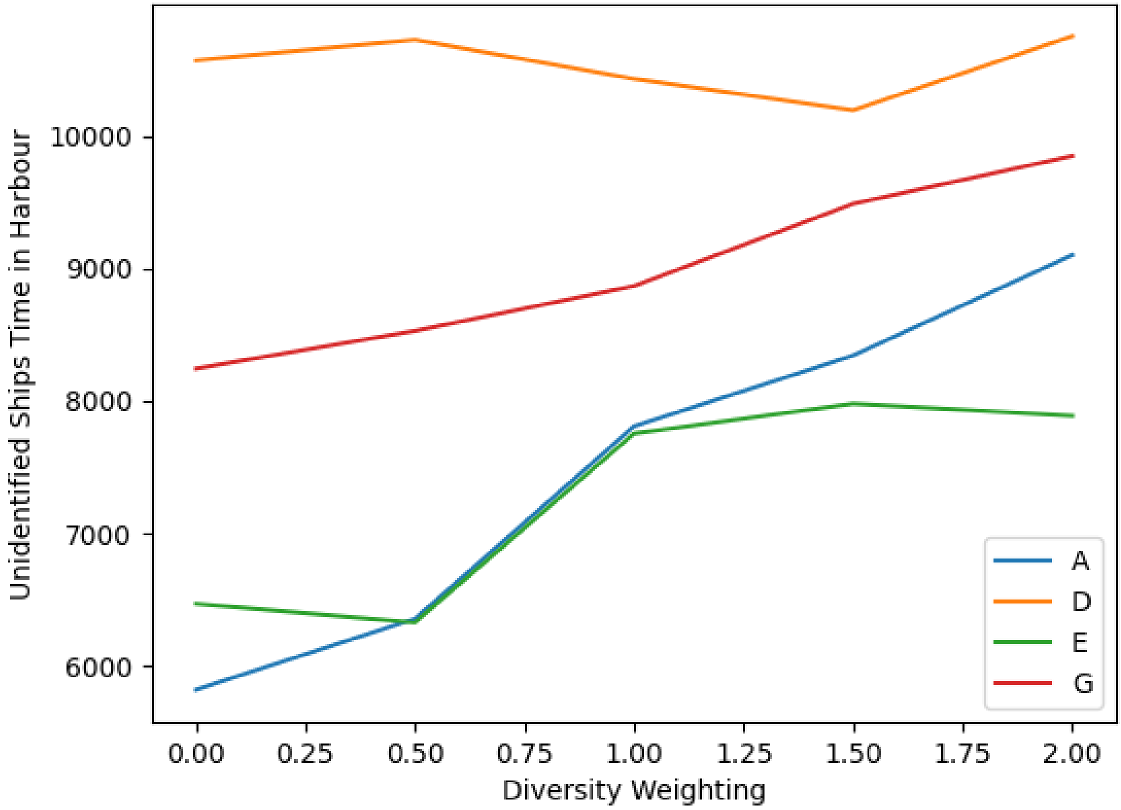

Figure 6: Effect of the diversity weighting on Unidentified Ships Time in Harbour (colour print)

a *benefit*. Weightings A, B and C are weighted towards only one criterion. Each of these weightings shows a decrease in quality as the diversity weighting increases. The effect is more complex for weightings D, E, F and G as a result of the relationships between criteria. Figure 7 shows that weighting G features the highest *Average Lead Time* values at a diversity weighting of 2.0. This is caused by weighting G having equal weighting across criteria. *Average Lead Time* is correlated with the other criteria and therefore more optimal values are attained as the solutions decline in overall quality.

5.4.3. Conclusion

Increasing diversity weighting W gives rise to the desired effect on diversity, at the cost of a decreased quality of solutions. This is because the optimal routes with 0 diversity weighting W often visit the same ships. It is worth noting that the recommended (first route) in each ranking is unchanged by altering W.

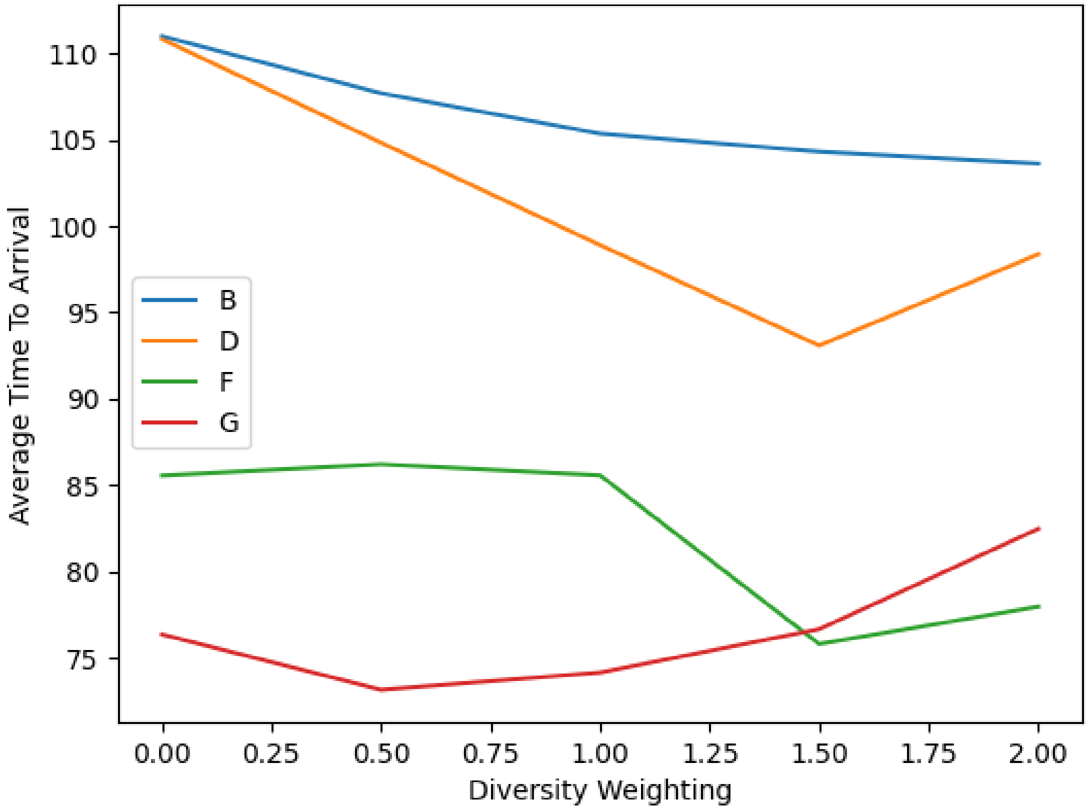

Figure 7: Effect of the diversity weighting on Average Time of Arrival (colour print)

6. Method Trade-off Evaluation

In this section we generate rankings using each of WPM, AHP, TOPSIS and PROMETHEE, with various weightings for criteria. The top 3 routes for each ranking were then simulated to calculate the outcomes. The smoothness of change in criteria outcomes as criteria weightings increases is used to evaluate the consistency of trade offs between criteria (*Desiderata 4*).

6.1. Motivation

Changing criteria weightings is an important part of a decision maker's process of exploring a solution space. For MCDM methods it is desirable for changing criteria weightings to create predictable effects on the outcomes of decisions. In a dynamic setting, the outcomes of decisions are uncertain. For example, in our situational awareness case study, the decision maker selects a route based upon the *predicted outcomes*. These predicted values for the criteria, *Unidentified Ships in Harbour*, *Average Lead Time* and *Fuel per Ship*, are not sure to be consistent with the actual values for the route. Therefore, in this experiment we simulate routes across the scenarios to calculate values

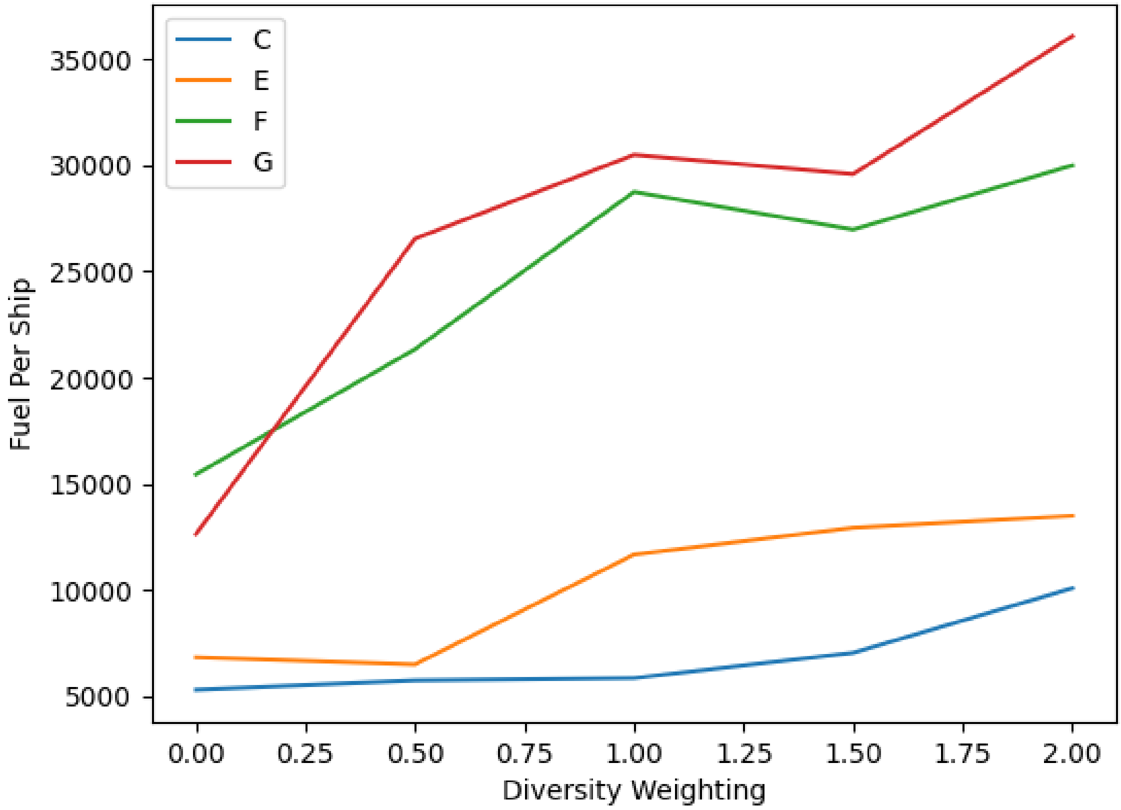

Figure 8: Effect of the diversity weighting on Fuel per Ship (colour print)

for the outcome of a route. The outcome is measured through three criteria that map to a corresponding predicted criterion.

- **Score** - The number of ships that arrive at the harbour identified minus the number of ships that arrive unidentified.
- **Average Lead Time** - The actual average lead time for identified ships.
- **Fuel Per Ship** - The actual amount of fuel used per ship identified.

Score is predicted through the number of unidentified ships in harbour, average lead time through predicted average lead time and fuel per ship through predicted fuel per ship. To measure the consistency of trade-offs in a dynamic setting, we compare the weighting for our criteria to the corresponding actual values of the top 3 routes in a ranking.

6.2. Evaluation Methodology

To generate rankings, the genetic algorithm was run over each of the 10 scenarios, for 800 generations using the weightings shown in Table 3. The weightings capture a smooth transition of the weight of C_1 between a weighting of 0 and 1. This process was repeated to generate 11 rankings for each criterion (C_1) and scenario pair. We then simulated the drone according to the first 3 routes, across all 600 time-steps of the corresponding scenario, allowing us to calculate each of the outcomes: *score*, *average lead time* and *fuel per ship*.

To combine these three outcomes into a single metric for each ranking, we used DCG. We chose DCG because when providing recommendations, the further an option is down the ranking, the less likely it is to be picked. DCG quantifies this by giving a higher weighting to higher ranked solutions. The result is three evaluation metrics for a ranking R: the DCG of score (score-DCG(R)), the DCG of average lead time (ALT-DCG(R)) and the DCG of fuel per ship (FPS-DCG(R)). The formulas are given in Equations 18, 19 and 20 respectively.

Let R be a ranking

Let r_i be the ith route in a ranking

$\mathrm{sc}(r_i)$ is the simulated score of r_i

$$\text{score-DCG}(R) = \frac{\mathrm{sc}(r_1)}{\log_3 2} + \frac{\mathrm{sc}(r_2)}{\log_3 3} + \frac{\mathrm{sc}(r_3)}{\log_3 4} \tag{18}$$

$\mathrm{lt}(r_i)$ is the simulated average lead time of r_i

$$\text{ALT-DCG}(R) = \frac{\mathrm{lt}(r_1)}{\log_3 2} + \frac{\mathrm{lt}(r_2)}{\log_3 3} + \frac{\mathrm{lt}(r_3)}{\log_3 4} \tag{19}$$

$\mathrm{fps}(r_i)$ is the simulated fuel per ship of r_i

$$\text{FPS-DCG}(R) = \frac{\mathrm{fps}(r_1)}{\log_3 2} + \frac{\mathrm{fps}(r_2)}{\log_3 3} + \frac{\mathrm{fps}(r_3)}{\log_3 4} \tag{20}$$

For each of the rankings generated through weightings 1-11 over a criterion C_1, the DCG of the corresponding outcome is calculated. The average DCG across all scenarios of each algorithm for *Unidentified Ships in Harbour*, *Average Lead Time* and *Fuel per Ship* are shown in Figures 9, 10 and 11 respectively.

To measure the smoothness of trade-offs, the Pearson correlation between the weight of a criterion and the corresponding outcome is calculated. This metric reflects the linear correlation of variables, and thus provides an indication of the consistency of trade-offs as the weighting changes. The correlation is calculated for each scenario, from which we calculate a mean and standard error of the mean. Tables 4, 5 and 6 show the mean and standard error of the mean for *unidentified ships in harbour* weight vs *score*, predicted *average lead time* weight vs *average lead time* and *fuel per ship* weight vs *fuel per ship*.

Weighting	C_1 Weight	C_2 Weight	C_3 Weight
1	0	0.5	0.5
2	0.1	0.45	0.45
3	0.2	0.40	0.40
4	0.3	0.35	0.35
5	0.4	0.30	0.30
6	0.5	0.25	0.25
7	0.6	0.20	0.20
8	0.7	0.15	0.15
9	0.8	0.10	0.10
10	0.9	0.05	0.05
11	1.0	0	0

Table 3: Weighting for criteria to evaluate changing C_1 weight

6.3. Results

Figure 9 shows the weighting for *unidentified ships in harbour* against the score DCG. It can be observed that as the weight for *unidentified ships in harbour* increases, the score DCG also increases. The exception to this can be seen at the highest weights where score DCG decreases. This is caused by the fact that the other criteria also have an effect on score. Increasing the weighting past 0.6 gives diminishing returns for the criterion at the cost to other criteria. *Unidentified ships in harbour* is not a perfect predictor of score because ships can change direction unexpectedly, therefore having a small weighting towards *fuel per ship* improves performance.

Figure 10 shows the weighting for *average lead time* against the average lead time DCG. It can be observed that as the weighting for *average lead time* increases, the average lead time DCG also increases. Once the weight reaches 0.7 the average lead time DCG peaks with no further gains in outcome associated with a higher weighting. This is caused by the algorithm selecting the routes with the highest average lead time once the weight is equal to or greater than 0.7.

Figure 11 shows the weighting for *fuel per ship* against the fuel per ship DCG. It can be observed that as the weight for *fuel per ship* increases, the average fuel per ship DCG decreases. This is because the *fuel per ship* is a cost criterion. The fuel per ship DCG decreases quickly, then reaches an optimum value at a weighting of 0.5 with

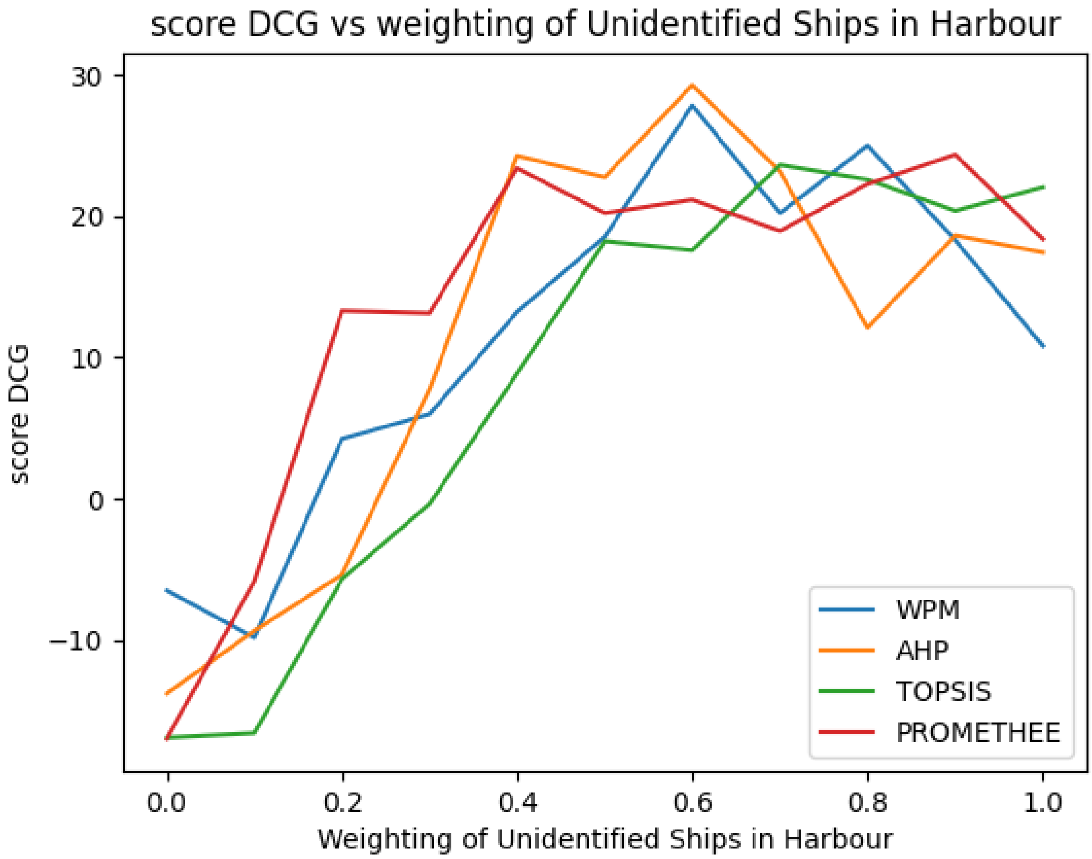

Figure 9: The change in score DCG as the weighting for Unidentified Ships in Harbour increases (colour print)

no further gains in outcome associated with a higher weighting. The exception to this is with the PROMETHEE algorithm, which shows an increase in fuel per ship DCG as the criterion weight increases. The likely cause of this is the preference function being unsuitable for the range of values of *fuel per ship*. In this case, a linear preference function was applied to all criteria, as described in Subsection 3.4, but it may be that a non-linear preference function better models the behaviour of our criteria. The values for fuel per ship fall closer to an exponential scale, therefore the trade-offs made are inappropriate. This creates a linear correlation for the predicted values. It is worth noting that there is a difference between predicted values and the actual outcomes measured. This noise is particularly great for *fuel per ship* due to the wide range of possible values. As a result, the correlation is reversed.

Table 4 shows the mean Pearson correlation and standard error of the mean for *unidentified ships in harbour* weight vs *score*. TOPSIS has the highest correlation, followed by PROMETHEE, AHP and WPM respectively. TOPSIS has substantially more consistent trade-offs than the other algorithms, with a correlation of 0.833. Table 5 shows the mean Pearson correlation and standard error of the mean for *average lead time* weight vs *average lead time*. WPM has the highest correlation, followed by PROMETHEE then TOPSIS and AHP. All the correlations are

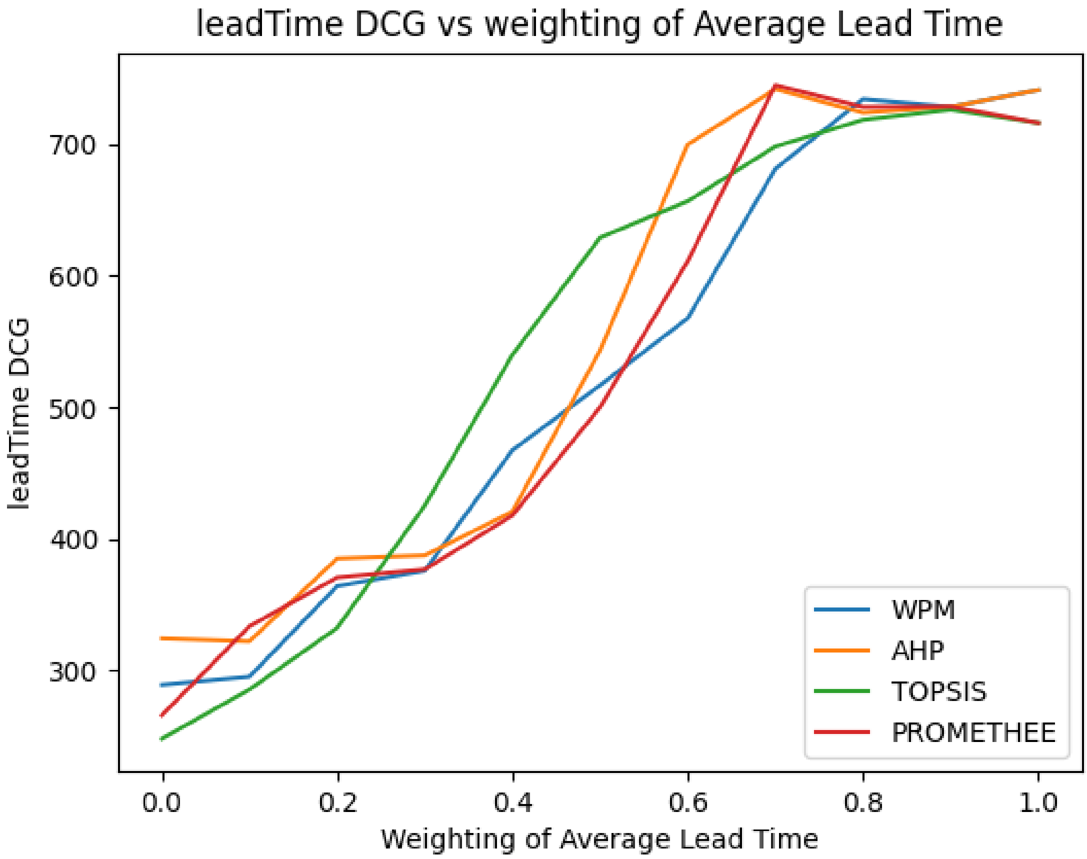

Figure 10: The change in Average Lead Time DCG as the weighting for Average Lead Time increases (colour print)

strong, with little difference between the algorithms. Table 6 shows the mean Pearson correlation and standard error of the mean for *fuel per ship* weight vs *fuel per ship*. *Fuel per ship* is a cost criterion, therefore a high negative correlation corresponds to consistent trade-offs between criteria. The correlations are negative with the exception of PROMETHEE. Generally the correlation is weak, showing inconsistent behaviour of changing weightings. The most consistent trade-offs are made by the TOPSIS algorithm, which had a correlation of -0.64. TOPSIS was followed by WPM, AHP and PROMETHEE respectively.

6.4. Conclusions

Overall, TOPSIS was found be the algorithm which made the most consistent trade-offs between criteria (Desiderata 4), only under-performing another algorithm with respect to *average lead time*. AHP and WPM were the next most consistent algorithms with no significant difference in correlation between weights and outcome. Finally, PROMETHEE was the least consistent algorithm, this was caused by the implementation using a fixed preference function $P(x)$ for all criteria.

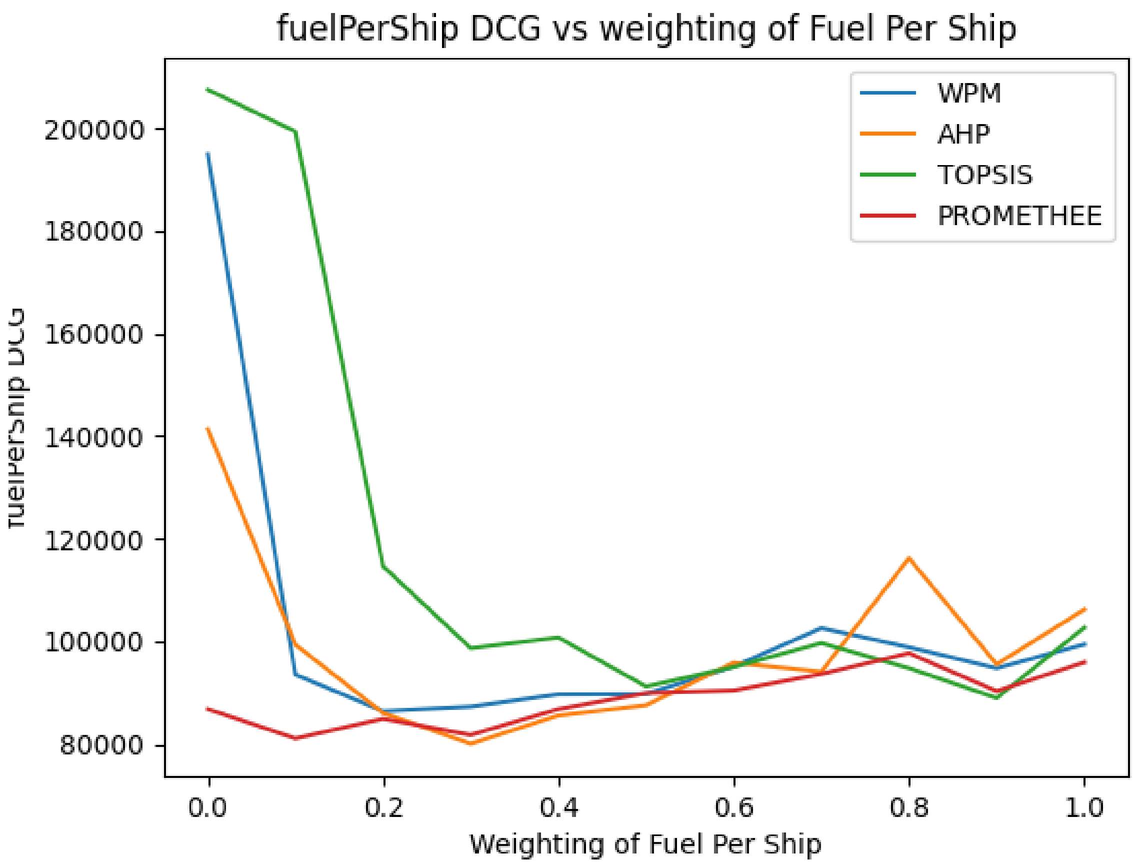

Figure 11: The change in Fuel per Ship DCG as the weighting for Fuel per Ship increases (colour print)

7. Method Criteria Values Sensitivity Evaluation

In this section we generate rankings using each of WPM, AHP, TOPSIS and PROMETHEE as a fitness function for our dynamic genetic algorithm. The stability of the resultant rankings is evaluated (*Desiderata 5*) under small changes to criteria values.

7.1. Evaluation Methodology

To evaluate the stability of rankings we used each MCDM method for 800 generations to generate a ranking. This process was repeated for each of the scenarios to generate a total of 10 rankings per method. Each algorithm used weighting G which assigns equal weighting to each criterion. For this experiment the diversity weight W was set to 0 to remove any effect of the diversity discount function on the outcome.

Once we generated a ranking we applied small changes to criteria values. To do this we modelled a random variable (X) using a gamma distribution (Γ) for each criteria value of each route in the ranking (Lukacs 1955). The

	USH weighting vs score DCG
WPM	0.541 +- 0.115
AHP	0.603 +- 0.049
TOPSIS	0.833 +- 0.0355
PROM	0.629 +- 0.0895

Table 4: Correlation between USH weighting and score DCG for each algorithm; USH refers to Unidentified Ships in Harbour

	ALT weighting vs ALT DCG
WPM	0.942 +- 0.00541
AHP	0.885 +- 0.0415
TOPSIS	0.885 +- 0.0245
PROM	0.893 +- 0.0208

Table 5: Correlation between ALT weighting and ALT DCG for each algorithm; ALT refers to Average Lead Time

gamma distribution is a two-parameter family of continuous probability distributions. The gamma distribution is parameterised using shape k and scale θ as shown in Equation 21. For a gamma distribution with shape k and scale θ, the mean (μ) and variance (σ^2) are given in Equations 22 and 23 respectively.

$$X \sim \Gamma(k, \theta) = Gamma(k, \theta) \tag{21}$$

$$\mu = k\theta \tag{22}$$

$$\sigma^2 = k\theta^2 \tag{23}$$

The shape parameter affects the shape of the distribution rather than shifting it or stretching it. The scale parameter spreads the distribution across a larger range. We created a gamma distribution with the mean (μ) as the original value for the criteria (c_x) and the variance (σ^2) being 0.1% of the value for the criteria. To achieve this we parameterised the gamma distribution with θ and k as shown in Equations 24 and 25 respectively. Resultant gamma distributions are shown in Figure 12 for criteria values (c_x): $c_1 = 5$, $c_2 = 15$, and $c_3 = 30$.

$$\theta = \frac{1}{1000} \tag{24}$$

$$k = \frac{c_x}{\theta} \tag{25}$$

Each X was then sampled 200 times to generate 200 new sets of solutions with small changes to all criteria values. The sampled rankings were then re-ranked according to the MCDM method used to generate the original ranking. We then measured the average change in rank for each route.

	FPS weighting vs FPS DCG
WPM	-0.268 +- 0.109
AHP	-0.166 +- 0.176
TOPSIS	-0.64 +- 0.0997
PROM	0.436 +- 0.159

Table 6: Correlation between FPS weighting and FPS DCG for each algorithm; FPS refers to Fuel Per Ship

The change in rank for each route for the Edinburgh scenario is shown in Figures 13, 14, 15 and 16 for WPM, AHP, TOPSIS and PROMETHEE respectively. The changes in rank were then averaged to calculate an average change across each ranking for each algorithm and scenario, as shown in Table 7. Finally, these scores were averaged across all scenarios to calculate the average change for each algorithm, given in Table 8.

7.2. Results

Figures 13-16 show the ranks of routes across 200 samples for each algorithm over the Edinburgh scenario. The original rank of the route is shown across the bottom with the box plot showing the mean, upper quartile, lower quartile and range of ranks for each route, circles are drawn represent outlying values.

Figure 13 shows that the WPM ranking is very unstable, with an average rank change of 2.91. It can be observed that the ranks of routes varies through almost the entire range of potential values, with the trend of increasing rank from left to right only visible through the mean and interquartile range. Figure 14 shows the most stable ranking, generated by AHP, resulting in an average rank change of 0.604. The top four routes are always the same, with positions varying more for lower ranked routes. Figure 15 shows a ranking generated by TOPSIS with an average rank change of 1.43. The first two routes are always the same, with positions varying over a large range for the rest of the ranking. Figure 16 shows a relatively stable ranking generated by PROMETHEE, with an average rank change of 0.853. For this ranking, the lowest ranked routes are the most stable, with the top results varying across a wider range of ranks.

Table 7 shows the average rank change for each algorithm across each scenario. It can be observed that the scenario has a significant effect on the stability of the rankings. For example, the Liverpool scenario on average generated much more stable rankings, resulting in an average rank change of 0.189. This was caused by the fact the scenario tended towards shorter routes which caused larger differences between criteria values in the resultant rankings. On the other hand, scenarios which tended towards longer routes, such as Oban which had an average rank change of 3.24, generated clusters of very similar routes. This created rankings with much smaller differences between criteria values that were therefore much less stable under small changes.

Table 8 shows the average change for each algorithm across all scenarios. The standard error resulting from the randomness of sampling is also given. We found that applying TOPSIS resulted in the most stable rankings, followed

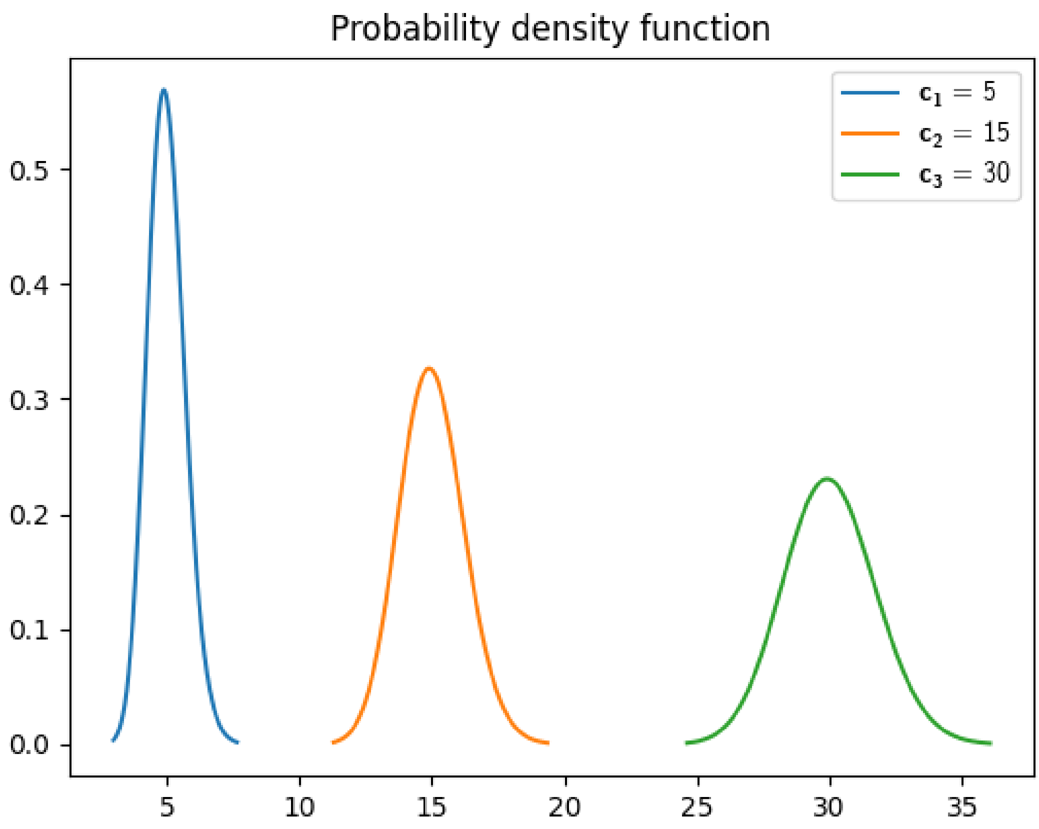

Figure 12: Gamma distribution probability density function with criteria values (c_x): $c_1 = 5$, $c_2 = 15$, and $c_3 = 30$. (colour print)

by AHP, PROMETHEE and WPM respectively. The standard error for each algorithm was very small, meaning that all the differences were highly significant, under the epistemic uncertainty generated by random variable X from Equation 21. This was calculated via t-test between all pairs of distributions (each generated by a different MCDM method).

7.3. Conclusions

TOPSIS was found to be the most stable method under small changes to criteria values (Desiderata 5). AHP was the second most stable with a slightly higher change in rank than TOPSIS. AHP was followed by PROMETHEE and WPM respectively, which were found to be substantially less stable.

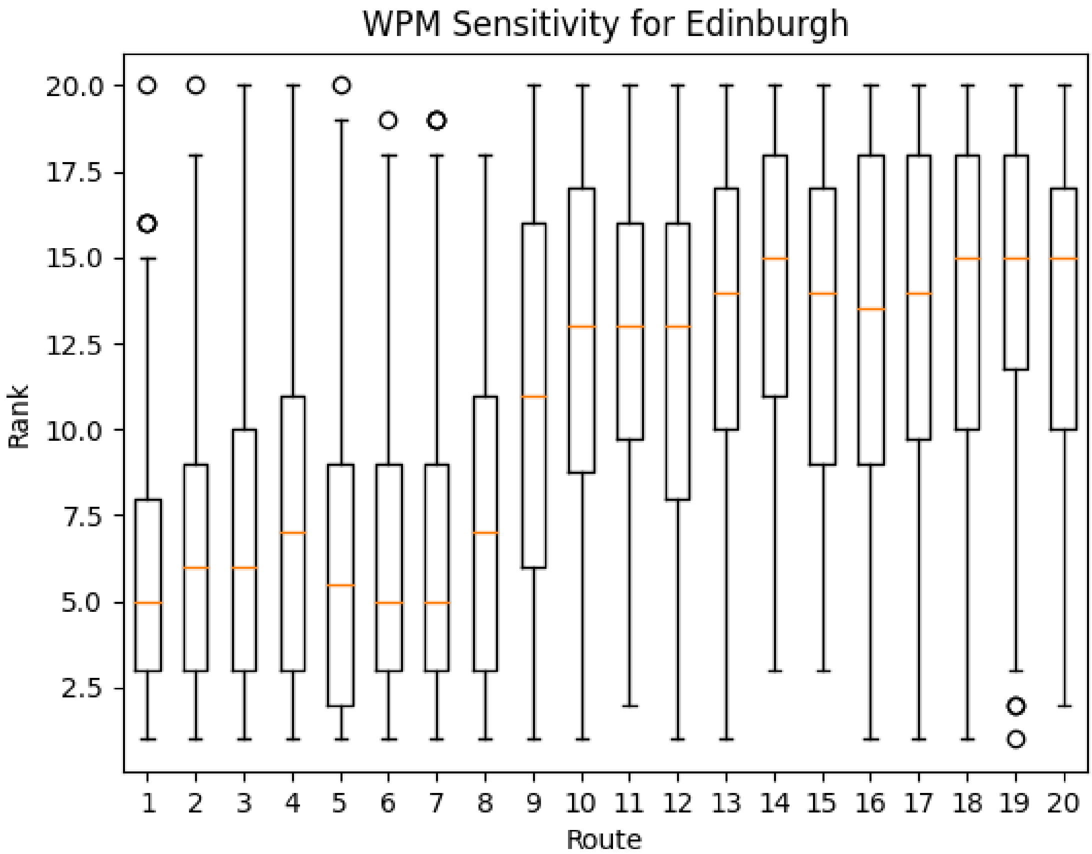

Figure 13: WPM ranking sensitivity box plot, showing an average rank change of 2.91 across 200 samples.

8. Discussion and Conclusion

8.1. Discussion

There are few comparative studies which compare different MCDM methods. Such studies fall into two categories: those that compare the results of MCDM methods and those that compare the attributes. The first category includes *Sarraf et al.* (Sarraf and McGuire 2020) and *Triantaphyllou et al.* (Triantaphyllou and Mann 1989), that compare the results of different methods to a benchmark. Generally, these benchmarks are results generated using the most popular MCDM method at the time. The second category includes *Zanakis et al.* (Zanakis et al. 1998), that evaluates the frequency of rank reversal in MCDM methods. Rank reversal is an undesired characteristic for MCDM methods, therefore understanding its prevalence is invaluable when selecting an appropriate algorithm. Our study falls into the second category, comparing desirable attributes of methods in the context of dynamic decision support. The two attributes highlighted as imperative for dynamic problems are the consistency of trade-offs between criteria and the stability of results under small changes to criteria.

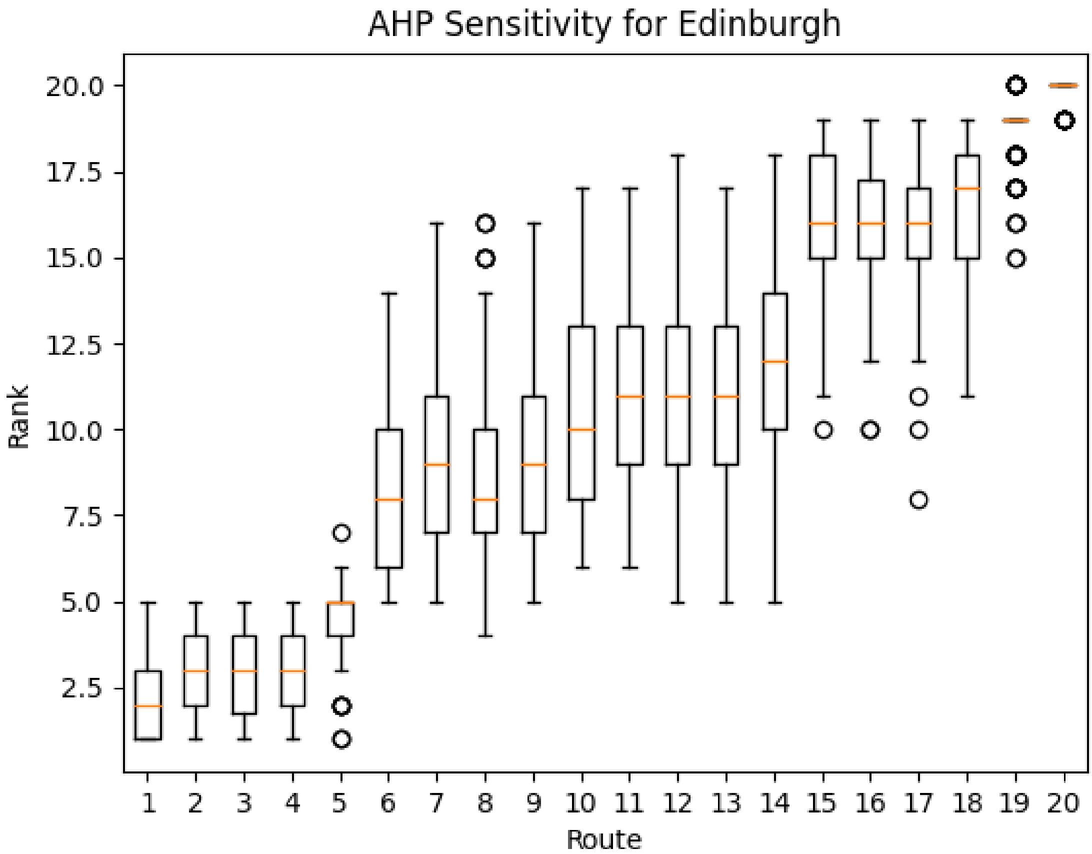

Figure 14: AHP ranking sensitivity box plot, showing an average rank change of 0.604 across 200 samples.

The effect of different strategies for the integration of weighting is not well studied but is an important characteristic contributing to the suitability of an MCDM method. For example, TOPSIS was found to be the MCDM method that made the most consistent trade-offs between criteria. The trade-offs between criteria in TOPSIS are controlled by the scaling of dimensions, each representing a criterion, in the solution space. This strategy has been shown to create consistently linear trade-offs, that can easily be understood by a decision maker.

TOPSIS was also found to be the most stable method under small changes to criteria values. This result is consistent with the study by *Zanakis et al.* (Zanakis et al. 1998), which found TOPSIS to be the most stable algorithm (suffering the least from rank reversals) when the members of a ranking are changed. Albeit using a different definition for stability, these results show the robustness of the goal, aspiration and references based models. This is because this class of models relates the solutions to an objective, leading to less sensitivity in the rankings when solutions are added/removed or criteria values are altered.

On the other hand, both AHP and PROMETHEE rely substantially on comparisons between all solutions and therefore have increased risk of rank reversal both under small changes to criteria values or when solutions are

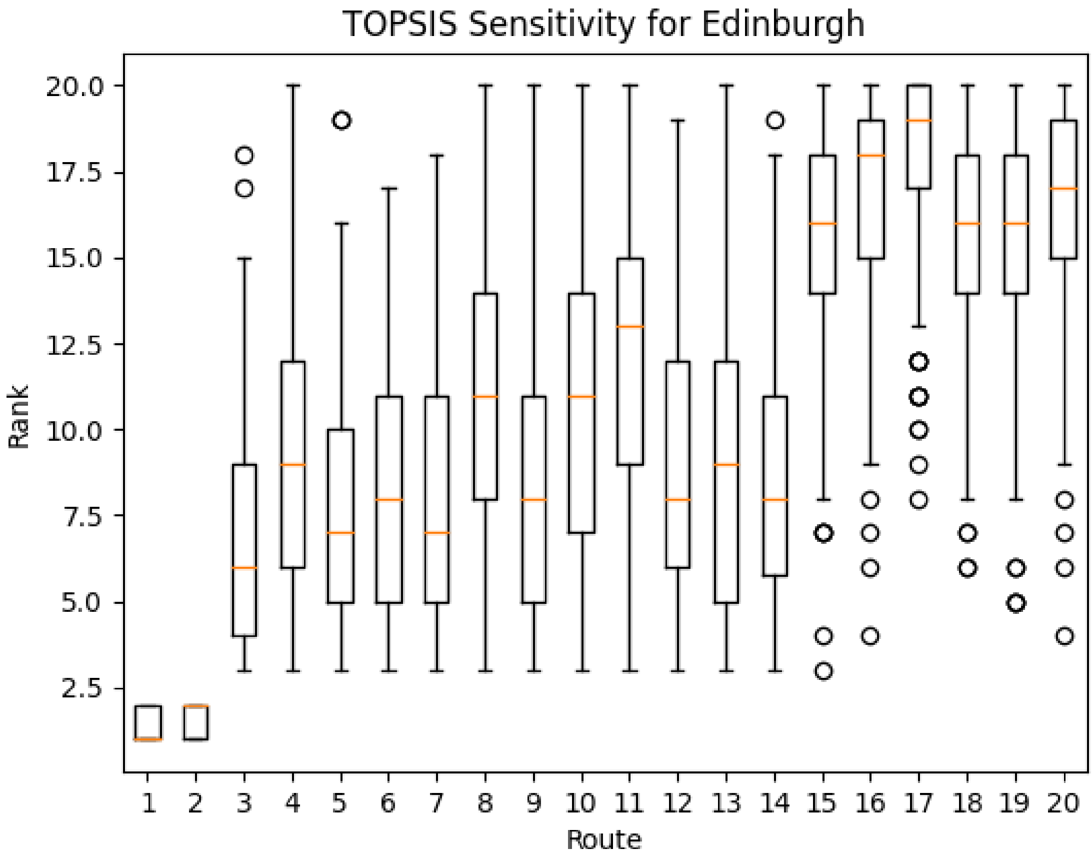

Figure 15: TOPSIS ranking sensitivity box plot, showing an average rank change of 1.43 across 200 samples.

added/removed. *Zanakis et al.* showed that WPM is relatively stable under addition/removal of solutions, because there is no relative scoring of solutions beyond normalisation. Unfortunately, this normalisation causes a large amount of instability when criteria values change. When the largest criteria value changes, the multiplicative approach of WPM can cause a substantial change in the relative values of criteria. This leads to unstable rankings, under small changes to criteria values.

8.2. Conclusion

In this paper, a dynamic multi-criteria DSS was presented for situational awareness using drones. It was shown how the DSS enabled declarative specification of preferences (Desiderata 1), dynamic revision of recommendations (Desiderata 2) and high diversity of options (Desiderata 3). The approach uses declarative specification of preferences to set weights for an MCDM method. The MCDM methods were applied within a streaming genetic algorithm running in real-time to enable dynamic revision of recommendations. Our methodology goes further than the literature by applying *a priori* knowledge of the decision-maker's preferences to reduce the multi-objective evolutionary algorithm

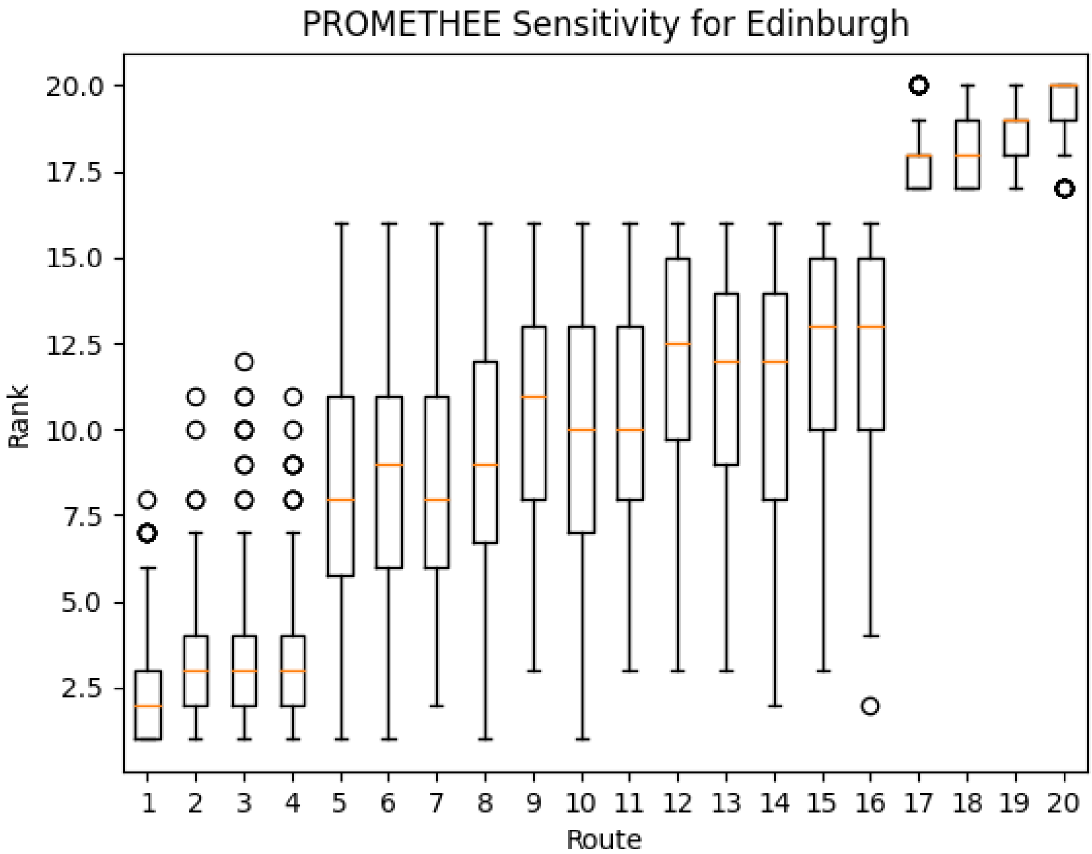

Figure 16: PROMETHEE ranking sensitivity box plot, showing an average rank change of 0.853 across 200 samples.

(MOEA) approach to a single step process that can be run over dynamic data to continuously refine results.

This approach was combined with a diversity discount function (as shown in Equation 13) that provides a lever to control the diversity of solutions. It was then shown that this function provided the desired effect, creating a diverse set of routes at the cost of reduced attainment of criteria. The mechanism leaves the recommended route (the route ranked first) unchanged.

This was followed by an evaluation of WPM, AHP, TOPSIS and PROMETHEE for suitability as components of a dynamic DSS using our situational awareness case study. The evaluation was designed as a generalisable approach to assessing decision support methods in the dynamic space. We mimic the approach of *Belton and Gear*, who identified rank reversal as a negative characteristic of MCDM methods through the identification and evaluation of desiderata for specifically dynamic problems. Our identified desiderata for comparing MCDM methods were the consistency of trade-offs (Desiderata 4) and the stability of results (Desiderata 5). These were both assessed in the context of our situational awareness scenario, using simulated route rankings to assess the stability of results and simulated route outcomes to assess the consistency of trade-offs. Based on the results, we found that TOPSIS best fulfils both

Algorithm	WPM	AHP	TOPSIS	PROM	AVG
Edinburgh	2.91	0.604	1.43	0.853	1.45
Liverpool	0.294	0.185	0.222	0.0547	0.189
Belfast	1.57	1.3	1.11	1.79	1.44
Dublin	0.633	0.743	0.47	0.408	0.564
Portsmouth	4.90	1.51	1.35	3.27	2.76
Plymouth	2.32	0.41	0.486	0.661	0.969
Oban	4.98	2.17	1.00	4.82	3.24
Douglas	4.98	0.489	0.853	0.586	1.73
Dover	4.89	1.54	1.31	3.39	2.78
Hull	0.153	0.244	0.132	0.11	0.160

Table 7: Average change of rank across rankings for each algorithm and scenario; AVG refers to the average for all algorithms across each scenario; PROM refers to PROMETHEE.

Algorithm	Average Change in Rank
WPM	4.40 +- 0.0172
AHP	1.5 +- 0.0204
TOPSIS	0.931 +- 0.0107
PROM	2.88 +- 0.0403

Table 8: Average rank change and standard error for each algorithm across all scenarios; PROM refers to PROMETHEE.

Desiderata 4 and 5. We therefore suggest TOPSIS as an appropriate MCDM method for dynamic decision making.

Acknowledgements. Dominic Duxbury is supported by an EPSRC iCASE award in association with BAE Systems and by BAE Systems personnel.

References

Basile, L. and D'Apuzzo, L. (2006). Transitive matrices, strict preference order and ordinal evaluation operators. *Soft Computing*, 10(10):933–940.

Belton, V. and Gear, T. (1983). On a short-coming of Saaty's method of analytic hierarchies. *Omega*, 11(3):228–230.

Belton, V. and Stewart, T. (2002). *Multiple criteria decision analysis: an integrated approach*. Springer Science & Business Media.

Brans, J. and Mareschal, B. (1992). Promethee V: Mcdm Problems With Segmentation Constraints. *INFOR: Information Systems and Operational Research*, 30(2):85–96.

Brans, J. and Mareschal, B. (1995). The promethee VI PROCEDURE: How to differentiate hard from soft multicriteria problems. *Journal of Decision Systems*, 4(3):213–223.

Brans, J. and Vincke, P. (1985). Note—A Preference Ranking Organisation Method. *Management Science*, 31(6):647–656.

Brans, J., Vincke, P., and Mareschal, B. (1986). How to select and how to rank projects: The Promethee method. *European Journal of Operational Research*, 24(2):228–238.

Bridgman, P. (1922). *Dimensional analysis*. Yale university press.

Calders, T. and Van Assche, D. (2018). Promethee is not quadratic: An o (qnlog (n)) algorithm. *Omega*, 76:63–69.

Carbone, P., Katsifodimos, A., Ewen, S., Markl, V., Haridi, S., and Tzoumas, K. (2015). Apache flink: Stream and batch processing in a single engine. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 36(4).

Coelho, B., Coelho, V., Coelho, I., Ochi, L., Haghnazar, R., Zuidema, D., Lima, M., and da Costa, A. (2017). A multi-objective green UAV routing problem. *Computers and Operations Research*, 88:306–315.

Coutinho, W., Battarra, M., and Fliege, J. (2018). The unmanned aerial vehicle routing and trajectory optimisation problem, a taxonomic review. *Computers and Industrial Engineering*, 120(June 2017):116–128.

Dewi, R., Hanggara, B., and Pinandito, A. (2018). A comparison between ahp and hybrid ahp for mobile based culinary recommendation system. *International Journal of Interactive Mobile Technologies (iJIM)*, 12:133.

Frederiksen, M. H. and Knudsen, M. P. (2018). Drones for offshore and maritime missions: Opportunities and barriers. *Innovation Fund Denmark*.

Greco, S., Ehrgott, M., and Figueira, J. (2016). *Multiple Criteria Decision Analysis*. Springer.

Hamdani, H. (2016). The complexity calculation for group decision making using topsis algorithm. In *AIP Conference Proceedings*, volume 1755, page 070007.

Hwang, C. and Yoon, K. (1981). *Multiple Attribute Decision Making*, volume 186. Springer Berlin Heidelberg, Berlin, Heidelberg.

Jarvelin, K. and Kekalainen, J. (2000). IR evaluation methods for retrieving highly relevant documents. *SIGIR Forum (ACM Special Interest Group on Information Retrieval)*, 51:41–48.

Javadi, M., Ramirez-Atencia, C., and Mostaghim, S. (2021). Combining manhattan and crowding distances in decision space for multimodal multi-objective optimization problems. *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*, 55:131–145.

Kawamoto, K., Houlihan, C., Balas, E., and Lobach, D. (2005). Improving clinical practice using clinical decision support systems: A systematic review of trials to identify features critical to success. *British Medical Journal*, 330(7494):765–768.

Keshavarz-Ghorabaee, M., Amiri, M., Zavadskas, E. K., Turskis, Z., and Antucheviciene, J. (2018). A comparative analysis of the rank reversal phenomenon in the EDAS and TOPSIS methods. *Economic Computation and Economic Cybernetics Studies and Research*, 52(3):121–134.

Lootsma, F. (1990). The french and the american school in multi-criteria decision analysis. *RAIRO-Operations Research*, 24(3):263–285.

Lukacs, E. (1955). A Characterization of the Gamma Distribution. *The Annals of Mathematical Statistics*, 26(2):319–324.

Macrina, G., Di Puglia Pugliese, L., Guerriero, F., and Laporte, G. (2020). Drone-aided routing: A literature review. *Transportation Research Part C: Emerging Technologies*, 120:102762.

Muthukrishnan, S. (2005). Data streams: Algorithms and applications. *Foundations and Trends in Theoretical Computer Science*, 1(2).

Odu, G. and Charles-Owaba, O. (2013). Review of multi-criteria optimization methods–theory and applications. *IOSR Journal of Engineering*, 3(10):01–14.

Pirotta, V., Grech, A., Jonsen, I., Laurance, W., and Harcourt, R. (2019). Consequences of global shipping traffic for marine giants.

Ramirez-Atencia, C., Bello-Orgaz, G., R-Moreno, M., and Camacho, D. (2014a). A simple CSP-based model for Unmanned Air Vehicle Mission Planning. In *INISTA 2014 - IEEE International Symposium on Innovations in Intelligent Systems and Applications, Proceedings*, pages 146–153. IEEE.

Ramirez-Atencia, C., Bello-Orgaz, G., R-Moreno, M., and Camacho, D. (2014b). Solving uav mission planning based on temporal constaint satisfaction problem using genetic algorithms. In *Proceedings of the 20th International Conference on Principles and Practice of Constraint Programming*, pages 65–71.

Ramirez-Atencia, C., Bello-Orgaz, G., R-Moreno, M., and Camacho, D. (2017a). Solving complex multi-UAV mission planning problems using multi-objective genetic algorithms. *Soft Computing*, 21(17):4883–4900.

Ramirez-Atencia, C., Mostaghim, S., and Camacho, D. (2017b). A knee point based evolutionary multi-objective optimization for mission planning problems. In *GECCO 2017 - Proceedings of the 2017 Genetic and Evolutionary Computation Conference*, volume 1755, pages 1216–1223.

Ramirez-Atencia, C., Rodriguez-Fernandez, V., and Camacho, D. (2020). A revision on multi-criteria decision making methods for multi-UAV mission planning support. *Expert Systems with Applications*, 160:113708.

Ries, J. and Ishizaka, A. (2012). A multi-criteria support system for dynamic aerial vehicle routing problems. In *CCCA12*, pages 1–4. IEEE.

Roshanov, P., Fernandes, N., and Wilczynski, J. (2013). Features of effective computerised clinical decision support systems: Meta-regression of 162 randomised trials. *BMJ (Online)*, 346(7899):1–12.

Roy, B. (1990). The Outranking Approach and the Foundations of Electre Methods. In *Readings in Multiple Criteria Decision Aid*, pages 155–183. Springer Berlin Heidelberg.

Saaty, T. (1987). The analytic hierarchy process-what it is and how it is used. *Mathematical Modelling*, 9(3-5):161–176.

Saner, L., Bolstad, C., Gonzalez, C., and Cuevas, H. (2009). Measuring and Predicting Shared Situation Awareness in Teams. *Journal of Cognitive Engineering and Decision Making*, 3(3):280–308.

Sardain, A., Sardain, E., and Leung, B. (2019). Global forecasts of shipping traffic and biological invasions to 2050. *Nature Sustainability*, 2(4):274–282.

Sarraf, R. and McGuire, M. (2020). Integration and Comparison of Multi-Criteria Decision Making Methods in Safe Route Planner. *Expert Systems with Applications*, 154:113399.

Selmi, M., Kormi, T., and Bel Hadj Ali, N. (2016). Comparison of multi-criteria decision methods through a ranking stability index. *International Journal of Operational Research*, 27(1/2):165.

Siraj, S., Mikhailov, L., and Keane, J. (2015). Contribution of individual judgments toward inconsistency in pairwise comparisons. *European Journal of Operational Research*, 242(2):557–567.

Steiner, D., Macdonald, R., Liu, Y., Truszkowski, P., Hipp, J., Gammage, C., Thng, F., Peng, L., and Stumpe, M. (2018). Impact of Deep Learning Assistance on the Histopathologic Review of Lymph Nodes for Metastatic Breast Cancer. *American Journal of Surgical Pathology*, 42(12):1636–1646.

Tomsett, R., Preece, A., Braines, D., Cerutti, F., Chakraborty, S., Srivastava, M., Pearson, G., and Kaplan, L. (2020). Rapid Trust Calibration through Interpretable. *Patterns*, 1(4):100049.

Triantaphyllou, E. (2000). *Multi-criteria decision making methods*. Springer.

Triantaphyllou, E. and Mann, S. H. (1989). An examination of the effectiveness of multi-dimensional decision-making methods: A decision-making paradox. *Decision support systems*, 5(3):303–312.

Vasey, B., Nagendran, M., Campbell, B., Clifton, D., Collins, G., Denaxas, S., Denniston, A., Faes, L., Geerts, B., and Ibrahim, M. (2022). Reporting guideline for the early-stage clinical evaluation of decision support systems driven by artificial intelligence: Decide-ai. *Nature medicine*, 28(5):924–933.

Wallenius, J., Dyer, J., Fishburn, P., Steuer, R., Zionts, S., and Deb, K. (2008). Multiple criteria decision making, multiattribute utility theory: Recent accomplishments and what lies ahead. *Management Science*, 54(7):1336–1349.

Webber, W., Moffat, A., and Zobel, J. (2010). A similarity measure for indefinite rankings. *ACM Transactions on Information Systems*, 28(4):1–38.

Zanakis, S., Solomon, A., Wishart, N., and Dublish, S. (1998). Multi-attribute decision making: A simulation comparison of select methods. *European Journal of Operational Research*, 107(3):507–529.