Lukas Partes

Decision Making for Multi-Objective Pathfinding Problems

OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

INF

FAKULTÄT FÜR
INFORMATIK

Intelligent Cooperative Systems
Computational Intelligence

Decision Making for Multi-Objective Pathfinding Problems

Bachelor Thesis

Lukas Partes

September 8, 2021

Supervisor: Prof. Dr.-Ing. habil. Sanaz Mostaghim

Advisor: Jens Weise, M.Sc.

Lukas Partes: *Decision Making for Multi-Objective Pathfinding Problems*
Otto-von-Guericke Universität
Intelligent Cooperative Systems
Computational Intelligence
Magdeburg, 2021.

Abstract

Multi-objective Pathfinding is an important research field. Although research has been done in multi-criteria decision making, there has been little research on decision making methodologies for the multi-objective pathfinding problem. In this thesis, a decision making approach is proposed that comes in three versions. These versions support decision making according to decision space, according to objective values or using a combination of these two steps. The different steps are then compared by hypervolume and distance-observations. It shows that each of the variant in decision space outperforms the other variants according to distances while the objective space version is superior concerning the hypervolume. The combination of both methods leads to a compromise.

Contents

List of Figures

List of Tables

List of Acronyms

MOP	Multi-objective optimization problem
MOPP	Multi-objective pathfinding problem
MCDM	Multi-criteria decision making
iff	If an only if
RDP	Ramer-Douglas-Peucker algorithm
WSM	Weighted Sum Approach
ELECTRE	Elimination and Choice Translating Reality
TOPSIS	Technique for Order Preference by Similarity to Ideal Solution
HV	Hypervolume
HVC	Hypervolume Contribution

1 Introduction

1.1 Motivation

The multi-objective optimization problem (MOP) covers a wide field of very important research problems. One problem that belongs to this category is the multi-objective pathfinding problem (MOPP). Pathfinding itself surrounds us in our daily lives, each time we navigate between places. Research is done in multiple fields. It is used to find paths across the world in video games [26]. Medical applications include resections [22, 20]. Another field of use is route planning which plays an important rule in logistics [34]. An increasing number of objectives in MOPP leads to huge numbers of optimal solutions that overwhelm the user. In order to select solutions that are most relevant for users, multi-criteria decision making(MCDM) is necessary [37].

Several MCDM-approaches have been proposed for the MOP [27, 35]. However, there has been little research on evaluating these approaches on or specializing them for the MOPP-Problem. Although there has been research conducted on the comparison of trajectories like paths to each other using distance metrics, using the similarity of paths has not been used as a criterion for decision making.

1.2 Research Questions

Before designing a concept, the following main research question has been formulated:

- How can decision making be done in multi-objective pathfinding?

Further, the thesis should answer the following additional questions?

- Which methods can be used?

  - Which methods can be used in decision space?
  - Which methods can be used in objective space?
- How can both methods be combined?

As a result, the goal of this thesis is to find a decision making methodology for the MOPP.

1.3 Structure of the Thesis

This introduction is followed by chapter 2 which explains the basic concepts of multi-objective optimization, clustering and decision making, including techniques that are used in this thesis. After that, chapter 3 gives a summary of some related research that have been proposed for MCDM. In chapter 4, the proposed concept of this thesis is explained. Chapter 5 is used to describe the experiments that have been done for evaluation and explains the results. Finally, chapter 6 concludes this thesis by giving a summary of results.

2 Background

In this chapter, the fundamentals of the multi-objective optimization, pathfinding, clustering and decision making are explained.

2.1 Optimization Problem

An optimization problem is a problem where the best solution of a number of solutions needs to be selected. As defined in [18], an optimization problem (Ω, f) consists of a search space Ω containing all potential solutions and an evaluation function $f : \Omega \rightarrow \mathbb{R}$ which calculates quality scores for each candidate $\omega \in \Omega$. Optimization problems can be divided into minimization and maximization problems. Depending on the type of optimization, a candidate $\omega \in \Omega$ is called solution if and only if either $\forall \omega' \in \Omega : f(\omega') \geq f(\omega)$ or $\forall \omega' \in \Omega : f(\omega') \leq f(\omega)$. In the remainder of this thesis, optimizations are assumed to be minimization problems, if not stated differently.

2.2 Multi-Objective Optimization Problem

As stated in [36], in multi objective optimization, there exists a multi objective evaluation function $\vec{f}(x)$ where each component $f_i : \Omega \rightarrow \mathbb{R}$ is an evaluation function. $\vec{f}(x)$ needs to be optimized. In MOP, objective values are no longer scalars but vectors. As a result, a different definition for solutions needs to be determined in order to compare these objective vectors. The set of solutions is referred to as S with $|S| = n \in \mathbb{N}$,

2.2.1 Pareto-Dominance

A very common way to compare candidates and define solutions is using Pareto-dominance.

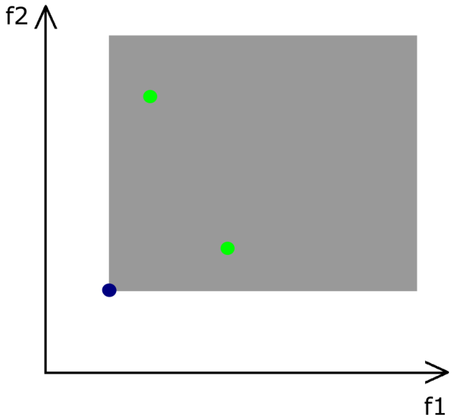

Figure 2.1: Illustration of Pareto-dominance

Definition 2.1 *Pareto-Dominance*

Pareto-dominance is a relation between candidates where $x_1 \in \Omega$ dominates $x_2 \in \Omega$ if and only if (iff)

$$(\forall i, 1 \leq i \leq k : f_i(x_1) \leq f_i(x_2)) \land (\exists i, 1 \leq i \leq k : f_i(x_1) < f_i(x_2)) \quad (2.1)$$

, further denoted as $x_1 \preceq x_2$ [18].

Definition 2.2 *Pareto-Optimality*

A candidate x is called Pareto-optimal iff $\nexists x' \in \Omega, x' \neq x : x' \preceq x$. Figure 2.1 shows a visualization of Pareto-dominance.

There exist different variants of dominance as alternatives to Pareto-dominance.

2.2.2 Epsilon-Dominance

The concept of epsilon-dominance can be used to approximate a Pareto-front. In this thesis, it is used to select solutions from the Pareto-front according to their objectives. The following terms are defined analogously to [19].

Definition 2.3 *(ε-domination)*

A candidate $x_1 \in \Omega$ ε-dominates $x_2 \in \Omega$ iff $\forall i, 1 \leq i \leq k : (1+\varepsilon) \cdot f_i(s_1) \leq f_i(s_2)$ for some $\varepsilon > 0$. This relation is further denoted as $x_1 \preceq_\varepsilon x_2$.

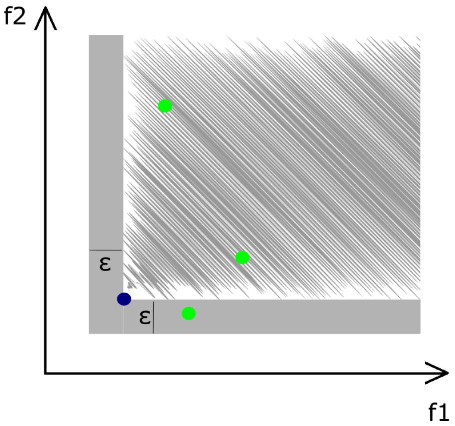

Figure 2.2: Illustration of epsilon-dominance

Definition 2.4 *(ε-optimality)*

Analogously to Pareto-optimality, epsilon-optimality is defined: x is called ε-optimal iff $\nexists x' \in \Omega, x' \neq x : x' \preceq_{\varepsilon} x$. Using epsilon domination, the Pareto Front can be approximated with a smaller number of solutions. Another benefit is that subsequent concepts take less computational effort because operations include fewer solutions. Figure 2.2 shows an example for epsilon-dominance.

2.2.3 Cone Dominance

In [17], Korhonen et al. describe a cone-shaped domination relation between solutions. Analogously to Pareto- and epsilon-dominance, cone dominance-relations are defined.

Definition 2.5 *(Cone-Domination)*

A candidate $x_1 \in \Omega$ cone-dominates $x_2 \in \Omega$ iff $\forall i, 1 \leq i \leq k : f_i(x_1) + \sum_{j=1,j\neq i}^{m} \alpha f_j(x_1) \leq f_i(x_2) + \sum_{j=1,j\neq i}^{m} \alpha f_j(x_2)$ for some angle with radian measure α. This relation is further denoted as $x_1 \preceq_{cone\ \alpha} x_2$.

Definition 2.6 *(Cone-Optimality)*

A non-cone-dominated solution is defined as follows: x is called cone-optimal iff $\nexists x' \in \Omega, x' \neq x : x' \preceq_{cone\ \alpha} x$.

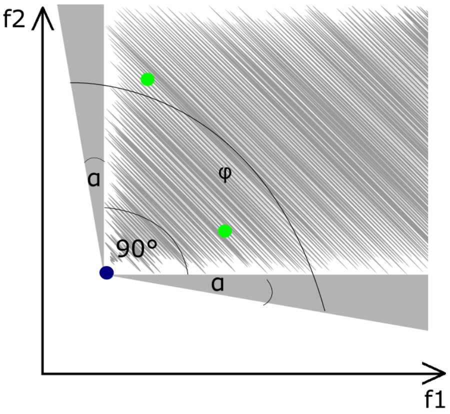

Figure 2.3: Illustration of cone-dominance

The initial angle of 90° which spans across the area of dominated solutions is increased by $2 \cdot \alpha$. Therefor, all solutions in the angle $\varphi = 90^\circ + 2 \cdot \alpha$ are cone-dominated. An illustration can be seen in Figure 2.3

2.3 Multi Objective Pathfinding

In the multi-objective pathfinding problem, there are given a directed Graph $G = (V, E)$, a starting node $n_s \in V$, an end node $n_e \in V$ and a multi objective evaluation function $\vec{f}$. A path is defined as a sequence of nodes $p = (n_1, ..., n_k)$ for some $k \in \mathbb{N}, k \neq 0$ [28]. A feasible solution is a path that connects the starting node and the end node, having the form $p = (n_s, n_2, ..., n_{k-1}, n_e)$. An example of a path can be seen in Figure 2.4 Optimal solutions are paths that optimize $\vec{f}$.

2.4 Clustering

In order to understand large amounts of data, it is useful to divide them into smaller groups. This process is called clustering [23]. In the following paragraph, x_i refers to an instance of S and clusters are referred to as $C_i, i \in \mathbb{N}$.

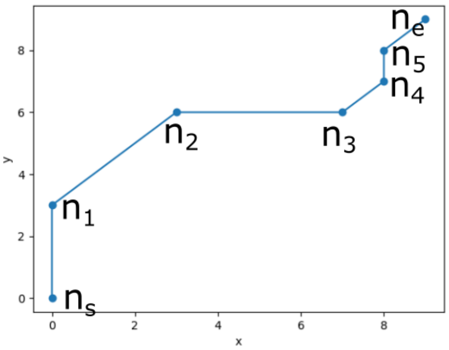

Figure 2.4: Example of a path connecting the points $n_s = (1, 1)$ and $n_e = (10, 10)$

2.4.1 Centroid-Based Clustering

A very often used clustering algorithm is k-Means, originally proposed in [23] and [21]. It calculates k clusters $C_i, 1 < i \leq k$ by their cluster centers c_i, which are called centroids, and assigns all instances to a cluster [39]. First, the number of clusters $k \leq |S|$ needs to be assigned. Then, k random centroids c_i are randomly assigned. All instances are iteratively assigned to their closest cluster and new cluster centers are calculated, until the positions of the centroids stop changing. The position of a centroid c_i is defined as the average of the instances of its cluster $c_i = \frac{\sum_{x \in C_i} x}{|C_i|}$. A disadvantage of this clustering method is that the initial assignment of the centroids influences the clustering result.

2.4.2 Agglomerative Clustering

Agglomerative Clustering represents one way of clustering. The clustering is done iteratively [39]. In the beginning, each point has its exclusive cluster. In each step, the two clusters closest to each other are combined. This is done until all entities belong to the same cluster. All of these joins can be visualized in a dendrogram. In Figure 2.5, a dendrogram and the four stages of clustering can be seen.

Distance Metrics

In order to identify how close two clusters are, the similarity of two instances needs to be defined. Distance metrics are used to determine how dissimilar two instances are.

Definition 2.7 *Distance Metric*

As Chen defined in [4], a distance metric $d : S \times S \to \mathbb{R}$ needs to satisfy the following conditions for $s_1, s_2, s_3 \in S$:

1. non-negativity: $d(s_1, s_2) \geq 0$
2. symmetry: $d(s_1, s_2) = d(s_2, s_1)$
3. triangular inequality: $d(s_1, s_3) \leq d(s_1, s_2) + d(s_2, s_3)$
4. $d(s_1, s_2) = 0$ iff $s_1 = s_2$

Euclidean Distance Euclidean distance is a distance measure for points in Euclidean space [32]. The distance for two n-dimensional points x and y is defined as:

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)} \tag{2.2}$$

Hausdorff Distance The Hausdorff distance is a distance measure that can be used on paths and curves in general [5]. For two paths $p1 = (n1_s, n1_2, ..., n1_{k-1}, n1_{e_1})$ and $p2 = (n2_s, n2_2, ..., n2_{k-1}, n2_{e_2})$, it is defined as:

$$d_H = \max\left\{ \sup_{\substack{n1_i; \\ 0<i\leq e_1}} \inf_{\substack{n2_i; \\ 0<i\leq e_2}} d(x, y),\ \sup_{\substack{n2_i \\ 0<i\leq e_2}} ; \inf_{\substack{n2_i; \\ 0<i\leq e_1}} d(x, y) \right\} \tag{2.3}$$

This metric does not take the flow of the curves into consideration.

Fréchet Distance The Fréchet distance is a distance measure that can be used on paths and curves in general. It was first proposed in [12]. Eiter and Manilla use a dog walk analogy to explain the problem [33]: A person walks a dog. Both are walking on their paths and are connected by a leash. Their speed can vary, but they can only move forward.An illustration of one of these dog walks can be seen in Figure 2.6. The Fréchet distance now is the shortest possible leash, depending on the travel speeds of the the dog and the person.

The Fréchet distance of two curves $f : [a, b] \to V$ and $g : [a', b'] \to V$ with $a, a', b, b' \in \mathbb{R}$ and V as metric space is calculated as:

$$\delta_F(f, g) = \inf_{\alpha,\beta} \max_{t\in[0,1]} d(f(\alpha(t)), g(\beta(t))) \tag{2.4}$$

$\alpha : [0, 1] \to [a, b]$ and $\beta : [0, 1] \to [a', b']$ are non-decreasing functions. Hence, $f(\alpha(t))$ and $g(\beta(t))$ calculate the positions of the person and the dog at time t. In [33], Eiter and Manilla proposed the discrete Fréchet distance as an approximation of the Fréchet Distance δ_{dF} that reduces the computational effort to $O(k_f \cdot k_g)$ where k_f and k_g are the number of nodes in each path. In order to calculate the discrete Fréchet distance, the two curves are approximated by polygonal curves P and Q. Their sequences of nodes in those paths are referred to as $\sigma(P) = (u_1, ..., u_p)$ and $\sigma(Q) = (v_1, ..., v_q)$. The authors define a coupling L of P and Q as a sequence $L = (u_{a_1}, v_{b_1}), (u_{a_2}, v_{b_2}), \dots, (u_{a_m}, v_{b_m})$ where the order of points from P and Q is not changed and $a_1 = 1, b_1 = 1, a_m = p, b_m = q$. The discrete Fréchet distance of P and Q is calculated as:

$$d_f(P, Q) = \min\{\|L\| \mid L \text{ is a coupling between } P \text{ and } Q\} \tag{2.5}$$

Here, $||L||$ refers to the length of the coupling L and is defined as the longest length of couples in L.

Linkages

There exist different methods according to which clustering is done. Especially, there are different ways to define distances between clusters. The following are based on [9].

Single linkage defines the distance of two clusters C_1 and C_2 as the minimal distance between two paths of these clusters: $d_{single}(C_1, C_2) = \underset{x\in C_1, y\in C_2}{min} d(x, y)$.

Complete linkage defines the distance of two clusters as the maximum distance between two paths of these clusters: $d_{complete}(C_1, C_2) = \underset{x\in C_1, y\in C_2}{max} d(x, y)$.

Figure 2.7 illustrates both different linkages between two clusters of points.

2.4.3 Center of Clusters

In order to represent a cluster, a medoid can be used [15]. It is defined as the solution with the minimal distance to all other solutions in its cluster. Here, it is defined using discrete Fréchet distance:

$$m_C = \arg\min_{x_1 \in C} \sum_{x_2 \in C} d_f\left(x_1, x_2\right) \tag{2.6}$$

2.4.4 Comparing clusterings

There are different ways to compare clusterings of the same data to each other. In this thesis, only internal measures are discussed. These only use data that has been generated by or was used for the cluster analysis [38].

Average Silhouette Coefficient

The silhouette coefficient was proposed in [30] as a way to compare different clusterings to each other. For every instance i, $a_i = \frac{1}{|C_i|-1} \sum_{j \in C_i, i \neq j} d_F(i,j)$ describes the average distance of an instance i to all other instances in its cluster c_i and $b_i = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d_F(i,j)$ is the average distance of i to that cluster that minimizes this distance. Now, the silhouette coefficient is defined as:

$$S_i = \frac{(b_i - a_i)}{\max\left(a_i, b_i\right)} \tag{2.7}$$

It is used to determine how well an instance is being clustered. To compare different clusterings, the average of the silhouette coefficients needs to be compared. A higher silhouette coefficient might indicate a better clustering. The silhouette coefficient for the different number of clusters can be visualized in a graph, as shown in Figure 2.8. An assignment of a negative silhouette coefficient might indicate a false assignment of the observed individual. As a result, a negative average silhouette coefficient indicates a weak clustering.

Dunn Index

A different approach was proposed by Dunn in [8]. The Dunn index is defined as

$$\alpha(k,P) = \min_{1 \leqslant q \leqslant k} \min_{\substack{1 \leqslant r \leqslant k \\ r \neq q}} \mathrm{d}\left(C_q, C_r\right) / \max_{1 \leqslant p \leqslant k} \operatorname{diam}\left(C_p\right) \tag{2.8}$$

where two clusters C_i and C_j have the inter-cluster separation $d(C_i, C_j) = \min_{\substack{x \in C_i \\ y \in C_j}} d(x, y))$ and cluster C_i has the diameter $diam(C_1) = \max_{x,y \in C_i} d(x, y)$. The Dunn index is challenged by noisy data because the inter-cluster separation then is hardly influenced [38].

Davies-Bouldin index

Davies and Bouldin proposed a separation measure that can be used to compare clusterings of the same data. It consists of operation measure between clusters C_i and C_j $M_{ij} = \left\{ \sum_{k=1}^{N} |c_{ki} - c_{kj}|^p \right\}^{1/p}$ and a dispersion $S_i = \left\{ \frac{1}{|C_i|} \sum_{j=1}^{|C_i|} |x_j - c_i|^q \right\}^{1/q}$. Here, c_i refers to the centroid of the cluster C_i, and c_{ki} to the centroids k-th component. The similarity of two clusters is defined as $R_{ij} \equiv \frac{S_i + S_j}{M_{ij}}$ with maximum values $R_i = \max_{i \neq j} R_{ij}$. These values can be used to calculate the average of cluster similarities $\bar{R} \equiv \frac{1}{N} \sum_{i=1}^{N} R_i$ of N clusters. It needs to be minimized.

2.5 Ramer-Douglas-Peucker algorithm (RDP) for simplifying paths

For the agglomerative clustering, the distances between all paths of the Pareto Front must be calculated. In order to minimize the computational effort of the distance calculation, the paths can be shortened by an algorithm, Ramer, Douglas and Peucker proposed in [7] and [29]. It selects points which are needed to represent the curve using a parameter ε_{rdp}. The algorithm works as follows:

The start and end point of the path are selected as A and B and added to the simplified path. From all points that lie on the initial path between A and B, the point with maximum distance to the line $\overline{AB}$ is detected and referred to as C. If $d(C, \overline{A, B}) > \varepsilon_{rdp}$ then C is added to the simplified path between A and B and the method is repeated recursively between points A and C and between points C and B.

In case $\varepsilon_{rdp} = 0$, all points are added to the simplified path, except for points that are located on straight lines between two other points. Hence, only obsolete points are being removed, and the simplified path is identical to the original path.

An example can be seen in Figure 2.9. This figure shows how the curve stays the same for $\varepsilon_{rdp} = 0$ while using fewer nodes and how $\varepsilon_{rdp} = 0.5$ reduces the amount of nodes even more by simplifying the curve. A consequence of this could be that a clustering or other operations on the paths calculate a different output because the curve of the solutions change.

2.6 Decision Making

Decision making describes the process of selecting solutions, given a set of solutions [11]. In MCDM, this is done whilst taking into account multiple criteria [11]. Methods for MCDM usually calculate a ranking or a selection of the possible solutions. Decision making can be categorized in a priori and a posteriori decision making. The process of a posteriori decision making starts after a Front of optimal solutions has been found [24]. This process tends to be associated with high computational effort. On the other hand, a priori decision making guides the process of finding solutions. For that purpose, the decision maker's preferences must already be specified before finding solutions [24]. There exist also hybrids of these two approaches.

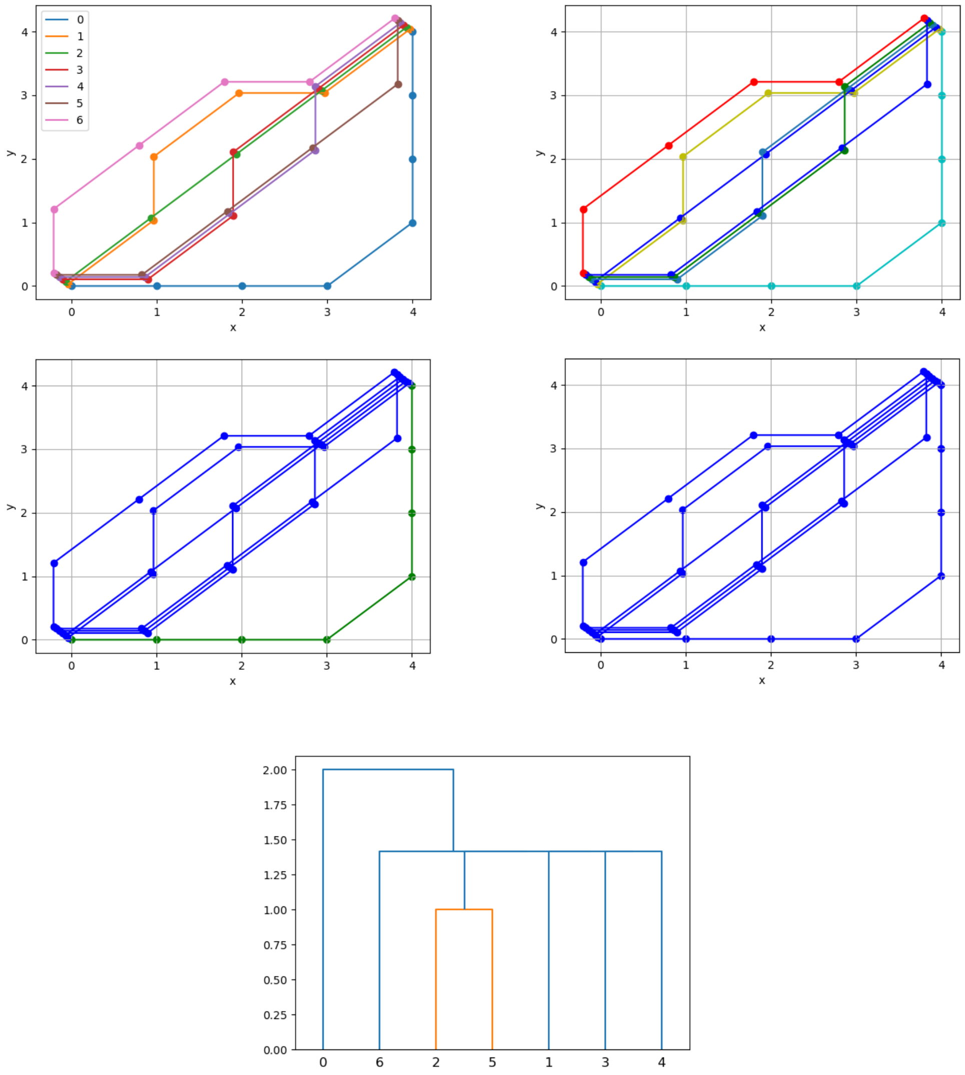

Figure 2.5: Different clusterings of a set of paths with seven (top left), six (top right), two (left middle) and one (right middle) clusters and the dendrogram (bottom)

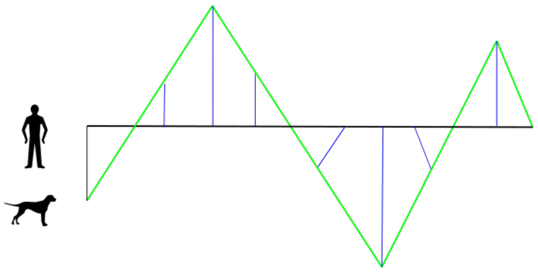

Figure 2.6: Example of a dog walk; leash illustrated as red

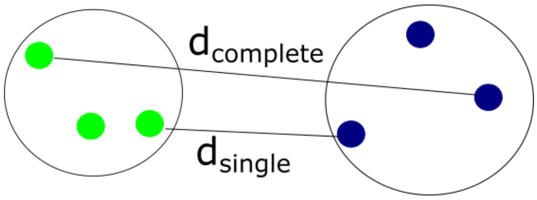

Figure 2.7: An illustration of single and complete linkage

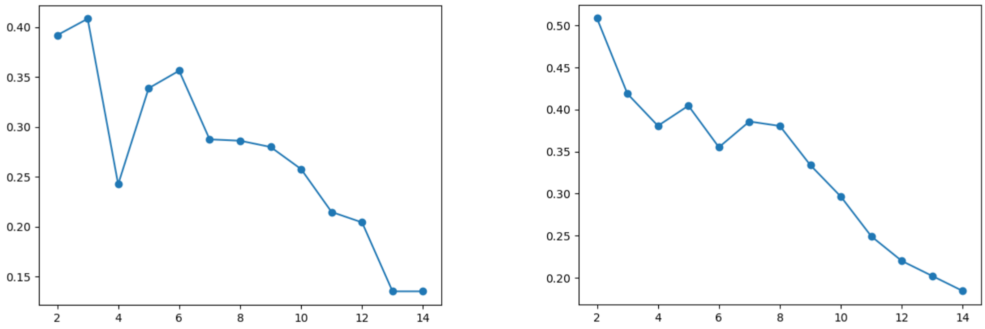

Figure 2.8: Example of curves for silhouette coefficient with single (left) and complete (right) linkage clustering

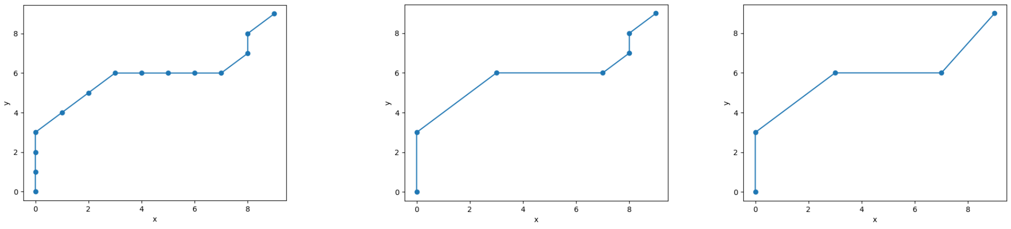

Figure 2.9: An example path (left) and two simplifications using RDP wit $\varepsilon_{rdp} = 0$ (middle) and $\varepsilon_{rdp} = 0.5$ (right)

3 Related Research

There has been a lot of research on multi-objective pathfinding. This section will explain some of the most common general decision making methods.

3.1 Weighted Sum Approach (WSM)

A very common and simple approach for MCDM, proposed in [10], uses a weight w_j for each criterion j [35]. For all solutions A_i, a WSM-score is calculated:

$$WSM_{A_i} = \sum_{j=1}^{n} a_{ij} w_j, \ i \in \{1, ..., N\} \tag{3.1}$$

The best solution then is the one with the Maximum WSM-score. This method can be used in problems where all criteria have the same dimensions and units. On the other hand, in multi-dimensional problems, WSM can not be used because the different criteria of different dimensions can not be compared to each other.

3.2 Analytic Hierarchy Process

In [31], Saaty describes a different approach to solve the MCDM, called the Analytic Hierarchy Process. First, weights for the different criteria are calculated, followed by weights for the different solutions. In both cases, this is done by comparing the importance of the elements, which can refer to criteria or solutions. A progressing scale from 1 to 9 ranges from equal importance to one element being extremely important compared to the other one. The scale is expanded by the reciprocal values. This can be represented in a matrix, where a value $a_{ij} = 9$ means that element i is extremely important compared to element

	criteria1	criteria2	criteria3
criteria1	1	3	7
criteria2	1/3	1	5
criteria3	1/7	1/5	1

Table 3.1: Example for the comparison matrix.

j, while $a_{ij} = 1$ means that both solutions have the same importance. It holds that $a_{ij} = 1/a_{ji}$.

The weights w_i for the criteria i are identical to the normalized principal eigenvector. Afterward, the same method is used to get weightings $w_{ij}$for each solution j in respect to each criterion i. That means that for every criterion, a matrix is created that compares the importance of all solutions. Finally, the priority of a solution is equal to $p_j = \sum_i w_{ij} \cdot w_i$. This can be used to get a ranking of solutions.

A different method of obtaining priorities is to only use the weights obtained for the different criteria. Those can be used to get a ranking analogously to WSM but with relative attribute values [35].

3.3 ELECTRE

The *Elimination and Choice Translating Reality* (ELECTRE)-Method uses pairwise comparisons between solutions [35]. Initially, the values of all solutions are represented by a matrix X in which an entry x_ij holds the weighted, normalized value of the j-th criterion of the i-th solution. The normalization can be formulated as $x_{ij} = \frac{a_{ij}}{|a_i|} \cdot w_i$.

For each pair of solutions (k, l), the concordance set $C_{kl} = \{j|y_{kj} \leq y_{lj}\}$ and discordance set $D_{kl} = \{j|y_{kj} > y_{lj}\}$ are calculated. These sets are being used to create a concordance matrix C with entries $c_{kl} = \sum\limits_{j \in C_{ki}} w_j$ and a discordance matrix D with entries $d_{kl} = \frac{\max_{j \in D_{k1}} \left| y_{kj} - y_{lj} \right|}{\max_j \left| y_{kj} - y_{lj} \right|}$. Entries for $k = l$ are not defined in both matrices. A concordance threshold $\underline{c} = \frac{1}{m(m-1)} \sum\limits_{\substack{k=1 \\ \text{and } k \neq l}}^{m} \sum\limits_{\substack{l=1 \\ \text{and } k \neq l}}^{m} c_{kl}$ is used to determine a concordance dominance matrix F with entries

$f_{kl} = \begin{cases} 1; & c_{kl} \geq \underline{c} \\ 0; & c_{kl} < \underline{c} \end{cases}$. Analogously, a discordance dominance matrix G with entries $g_{kl} = \begin{cases} 1; & d_{kl} \geq \underline{d} \\ 0; & d_{kl} < \underline{d} \end{cases}$ is created based on a discordance threshold $\underline{d} = \frac{1}{m(m-1)} \sum\limits_{\substack{k=1 \\ \text{and } k \neq l}}^{m} \sum\limits_{\substack{l=1 \\ \text{and } k \neq l}}^{m} d_{kl}$.

The aggregate dominance matrix E holds entries $e_{kl} = f_{kl} \times g_{kl}$. These entries can be used to identify a domination according to ELECTRE: $e_{kl} = 1$ means that solution A_k dominates solution A_l according to concordance and discordance. As a result, all columns with at least one entry that is equal to one can be discarded.

3.4 TOPSIS

"Technique for Order Preference by Similarity to Ideal Solution" (TOPSIS) was proposed by Yoon and Hwang in [14]. The main idea is that the distance to the ideal solution for a problem should be minimal and the distance to the worst problem should be maximal. The first step of their approach is to create a matrix with weighted, normalized attribute values, analogously to ELECTRE. Additionally, the ideal and worst possible solutions $A^* = \{(\min\limits_i v_{ij} | j \in C) | i = 1, 2, ..., m\}$ and $A^- = \{(\max\limits_i v_{ij} | j \in C) | i = 1, 2, ..., m\}$ are calculated. C refers to the set of criteria that need to be minimized.

The relative closeness of solution A_i to the ideal solution is calculated as $closeness_{i^*} = d_{i^-} / (d_{i^*} + d_{i^-})$ with distances d_{i^-} and d_{i^*} to the worst and ideal solution. Finally, the solutions A_i can be ranked in the decreasing order of $closeness_{i^*}$.

3.5 Knee Points

In the MOP, a knee refers to a convex "bulge" in the curve of the Pareto-front [6]. Those points that are located in this knee tend to be chosen as the preferred solutions. An example can be seen in Figure 3.1.

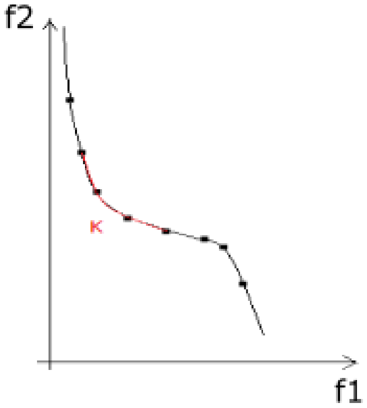

Figure 3.1: An Example of a knee K for a MOP with two objectives

There have been multiple MCDM-approaches that use knees in solutions for MOP.

3.5.1 Angle-Based Method

In [2], Branke et al. proposed two methods for a posteriori decsion making that work by detecting knee points. The first method is based on the observation that angles are spanned by neighbors of a knee point tend to be larger than the angle around other points. To detect knee points, for each objective vector $\vec{f}(x_i)$, lines are spanned to connect them with its two neighbors $\vec{f}(x_{i-1})$ and $\vec{f}(x_{i+1})$. The angle between these lines is calculated. In an advanced version, four angles between $\vec{f}(x_i)$ and its four neighbors $\vec{f}(x_{i-2})$, $\vec{f}(x_{i-1})$, $\vec{f}(x_{i+1})$ and $\vec{f}(x_{i+2})$ are calculated and the largest one is selected. In both variations, the points that span the largest angles can be assumed to be knee points. Although this method could be used to solve the MCDM in MOP in general, the authors recommend this approach for bi-objective problems only.

3.5.2 Utility-Based Method

The second method proposed in [2] can also be used for problems with more than two objectives. It uses utility functions $U(x, \lambda) = \sum \lambda_i f_i(x)$ with randomly chosen $\sum \lambda_i = 1$. The marginal utility

$$U'(x_i, \lambda) = \begin{cases} \min_{j \neq i} U(x_j, \lambda) - U(x_i, \lambda) & \text{for } i = \arg\max_j U(x_j, \lambda) \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

calculates the cost for discarding the value x_i with the highest utility and having to rely on the second best value x_j. Its expected value can be approximated, applying random utility functions on each solution and calculating the average value. Finally, the values with a higher expectation of marginal utility values are more likely to be knee points.

4 Proposed Approach

The goal of Decision Making is to provide the user with a ranking or selection of "good" solutions. The proposed approach selects these solutions according to their objective values or according to their representation in decision space. In both spaces, the methodologies are supposed to select solutions that represent the set of solutions. The algorithm consists of two steps. One of them is done in objective space. Here, the amount of solutions is reduced by using either epsilon- or cone-dominance. The other step is done by clustering the data and calculating output candidates in decision space. The algorithm works a priori, which means that it selects solutions from an already generated set of Pareto-optimal solutions.
There are three versions of the algorithm:

- Only make decision according to objective values
- Only make decision according to clustering in decision space
- Combination of both methods

The combination starts by reducing the solutions according to their objective values, followed by applying the clustering algorithm. The different versions are visualized in Figure 4.1

4.1 Preparation

The algorithm takes Pareto-optimal solutions represented by their paths and objective values as input by reading them from files. The paths are simplified with the RDP-algorithm and $\varepsilon_{rdp} = 0$ in order to keep their course while using fewer nodes. This safes up computational time in later steps. Additionally, all objective values are normalized to prepare for the evaluation.

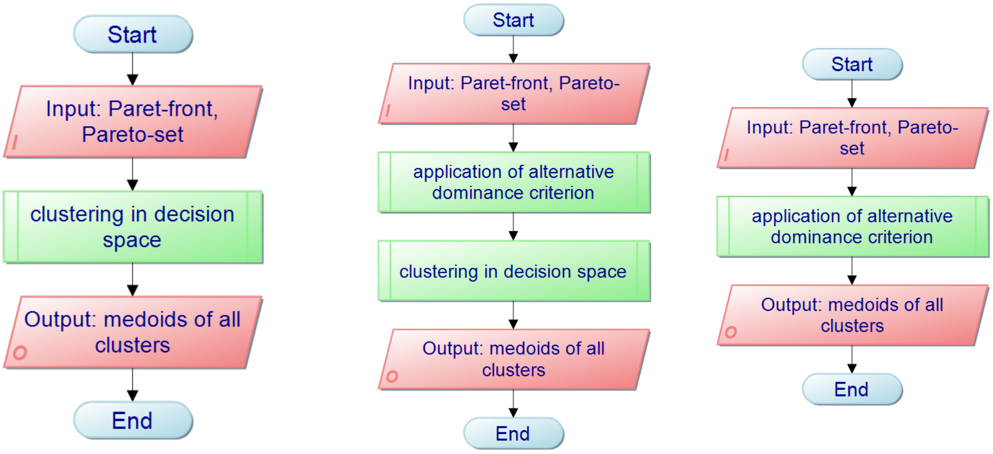
Start
Input: Paret-front, Pareto-set
clustering in decision space
Output: medoids of all clusters
End
Start
Input: Paret-front, Pareto-set
application of alternative dominance criterion
clustering in decision space
Output: medoids of all clusters
End
Start
Input: Paret-front, Pareto-set
application of alternative dominance criterion
Output: medoids of all clusters
End

Figure 4.1: possible sequences of both steps of the algorithm

4.2 Objective Space

This step is supposed to make a decision according to the objective values of solutions. Therefor, a relevant subset of solutions from the Pareto-front needs to be selected. In order to that, a more general form of domination is used, namely epsilon- or cone-dominance. This allows the algorithm to further select non-dominated solutions out of the non-Pareto-dominated solutions. In order to find representative solutions, the algorithms either uses as approximation of the Pareto-front by using epsilon-dominance or tries to detect knee points by using cone-dominance. First, the value of ε or φ is assigned. This is followed by the selection of non-dominated solutions in the sense of epsilon- or cone-dominance. For this selection process, a solution archive is maintained. All solutions are inserted iteratively. On each insertion, all currently stored solutions are checked for domination by the inserted one. Additionally, the inserted solution is checked for domination by all other solutions. In case of domination, the dominated solutions are discarded. This process written in pseudocode can be seen in Algorithm 1. The input $solution\_front$ represents the front of solution, $dominates(objectives1, new\_objectives2, domination\_parameter)$ returns true iff the vector of objective values $objectives1$ dominates $objectives2$. This can either refer to epsilon- or cone-dominance. The function $delete(list, value)$ is used to delete the value from the list $list$.

Algorithm 1 DM in Objective Space

Input: $solution\_front, domination\_parameter, dominates$

```
$archive = list[]$
for $new\_objectives$ in $solution\_front$ do
    $is\_dominated = FALSE$
    for $objectives$ in $archive$ do
        if $dominates(objectives, new\_objectives, domination\_parameter)$ then
            $is\_dominated = TRUE$
            BREAK
        end if
    end for
    for $objectives$ in $archive$ do
        if $dominates(new\_objectives, objectives)$ then
            $delete(archive, objectives)$
        end if
    end for
end for
```

4.3 Decision Space

In decision space, the algorithm is supposed to select solutions that differ in their course. Centroid-based clustering algorithms are difficult to use because they need to calculate an average of the instances. For paths, this proves difficult, although an approach for centroid-based clustering has been proposed in [3]. Hence, an agglomerative clustering is performed on the set of solutions with the intention of finding clusters of similar paths. In order to take the flow of the different paths into consideration, the distance between paths is calculated using discrete Fréchet distance. As linkage, complete and single linkage are used. The clustering is applied to calculate numbers of clusters ranging from two to half of the number of paths in the set of solutions. The "best" clustering is defined by the maximal silhouette coefficient. In order to find the best number of clusters, the clustering algorithm calculates numbers of clusters ranging from two to half of the number of paths in the set of solutions using single linkage and complete linkage separately. Then, the silhouette coefficient for each clustering is calculated. The clustering with the maximal value is supposed to be the best choice and consequently selected. This is done for both linkages independently.

Algorithm 2 Clustering in Decision Space

```
Input: $solution\_set$
$scores = \left\lceil \frac{|soultion\_set|}{2} \right\rceil$
$best\_clustering = 0$
$best\_score = 0$
for linkage in {single, complete} do
    for $number\_of\_clusters = 2, 3, \ldots, \left\lceil \frac{|solution\_set|}{2} \right\rceil$ do
        $clustering = clustering(solution\_set, number\_of\_clusters,$
          $discrete\_fr\_dist, linkage)$
        $scores[number\_of\_clusters] = calculate\_silhouette\_score(clustering)$
    end for
    if $argmax(scores) > best\_score$ then
        $best\_clustering = clustering(argmax(scores) + 1)$
        $best\_score = \max(scores)$
    end if
end for
for $cluster$ in $best\_clustering$ do
    Output: $medoid(cluster, discrete\_fr\_dist)$
end for
```

Finally, one path per cluster is used as output. This path should represent the cluster. Therefor, the medoid of each cluster is used. A pseudocode of the algorithm can be seen in Algorithm 2. The *solution_set* refers to the set of solutions that need to be clustered. To calculate the discrete Frechét distance, *discrete_fr_dist* is used. The function *clustering* performs a hierarchical clustering. As arguments, it gets a set of solutions that need to be clustered, the number of clusters, that need to be calculated, the distance functions and the preferred linkage. Further, *calculate_silhouette_score* and *medoid* are used to calculate the silhouette score and geometrical medoid of a clustering. The terms *single* and *complete* refer to linkages for clustering.

5 Evaluation

In this chapter, the evaluation of the methodologies is described. Therefor, the problems that were used for evaluation are explained. The different evaluation metrics are portrayed in order to present the evaluation results. Finally a summary is given.

5.1 Problems to evaluate

For evaluation, the proposed decision making is applied to Pareto fronts of different MOPs. These fronts have been provided by [37]. The Problems consist of two-dimensional grids with sizes $[1, x_{max}] \times [1, y_{max}]$ for $x_{max}, y_{max} \in \mathbb{N}$ where grid positions are referred to as (x, y). In each map, a start node $n_S = (1, 1)$ and an end node $n_e = (x_{max}, y_{max})$ are defined for path finding.

5.1.1 Features of Different Map Types

The different Problems differ in their size, but also in the type of the used map [37]. The three different features are explained according to [37] in this section.

Obstacles

Weise and Mostaghim [37] implemented different kinds of obstacles in each map: no obstacles, checkerboard pattern, lake obstacle. In the checkerboard pattern, every second cell is an obstacle. The lake is a circle-shaped obstacle with radius $r = x_{max}/4$. The velocity of an obstacle cell is $v(x, y) = 0$. The lake and checkerboard obstacles for a 12×12 map are shown in Figure 5.1

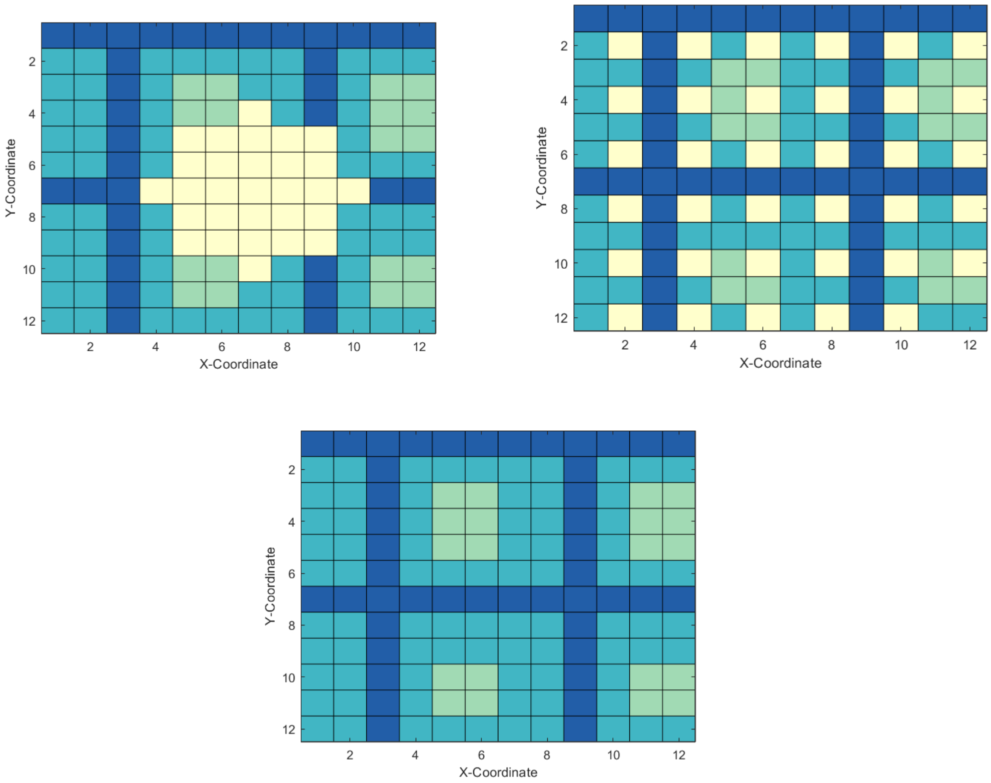

Figure 5.1: An example of the different obstacle-types: no obstacles (left), lake (middle) and lake Obstacles (right) for a map size of twelve; higher velocities are represented by darker colors

Elevation

The height of the grid cells is defined by an elevation function which uses a hill function in the interval $[-3, 3]$. The coordinates of the grid cells need to be scaled to spread across the interval. As hill function, one of the following functions are used with scaled coordinates (x_s, y_s):

$$\begin{aligned} h_m(x_s, y_s) = 3(1-x_s)^2 e^{-x_s^2-(y_s+1)^2} &- 10e^{-x_s^2-y_s^1}(-x_s^3 + x_s/5 - y_s^5) \\ &- 1/3e^{-(x_s+1)^2-y_s^2} \end{aligned} \tag{5.1}$$

$$h_1(x_s, y_s) = 5e^{-(x_s+1.5)^2-(y_s+1.5)^2} \tag{5.2}$$

$$h_2(x_s, y_s) = 5e^{-(x_s-1.5)^2-(y_s-1.5)^2} \tag{5.3}$$

$$h_3(x_s, y_s) = 5e^{-(x_s-1.5)^2-(y_s+1.5)^2} \tag{5.4}$$

The elevation is defined by combining those hill functions using a parameter n_h as follows:

$$h(x,y) = \begin{cases} \sum_{i=1}^{n_h} h_i; & n_h \in 2,3 \\ h_3; & n_h = 1 \\ h_m; & n_h = M \end{cases} \tag{5.5}$$

Neighborhood Relation

The 2^k - neighborhood relation defines between which nodes an agent is allowed to travel. The relation is denoted by a parameter $k \in \{2,3\}$ [37]. Therefore, $k = 2$ denotes a 4-neighborhood which means that an agent is only allowed to move north, east, south and west. An 8-neighborhood which is denoted by $k = 2$ additionally allows the agent to move north-east, south-east, south-west and north-west.

Backtracking is not allowed in any of the maps. As a result, the agent is only allowed to move north, east and north-east.

5.1.2 Objectives

The Pareto fronts were determined for problems with five objectives which Weise described in [37] as follows.

Euclidean length

The length of a path is calculated by adding the sums of the Euclidean distance between all adjacent pairs of nodes (n_i, n_{i+1}) of a path p.

$$f_1(p) = \sum_{i=1}^{K-1} d\left(n_i, n_{i+1}\right) \tag{5.6}$$

Expected delays

The second objective is the sum of delays between two adjacent nodes in the path. The *delay* function takes into consideration the velocity values of two adjacent nodes:

$$\text{delay}\left(n_i, n_{i+1}\right) = \begin{cases} 2; & v_{\max}\left(n_i\right) \neq v_{\max}\left(n_{i+1}\right) \\ 3; & v_{\max}\left(n_i\right) = v_{\max}\left(n_{i+1}\right) = 50 \\ 1; & v_{\max}\left(n_i\right) = v_{\max}\left(n_{i+1}\right) = 100 \\ \frac{1}{5}; & \text{otherwise} \end{cases} \tag{5.7}$$

Therefore, the second objective is defined as follows:

$$f_2(N) = \sum_{i=1}^{K-1} \text{delay}\left(n_i, n_{i+1}\right) \tag{5.8}$$

Elevation

The Elevation describes the sum of ascends between two adjacent nodes in the path $e(n_i, n_{i+1})$ using the elevation function $h(n_i)$:

$$\begin{aligned} f_3(p) &= \sum_{i=1}^{K-1} e\left(n_i, n_{i+1}\right) \\ e(m,n) &= \begin{cases} h(n) - h(m), & \text{if } h(n) > h(m) \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{5.9}$$

Traveling Time

The time which is needed by the agent to travel a path is used as the fourth objective. The traveling time for a single path segment is calculated by dividing the distance of the two adjacent nodes $d(n_i, n_{i+1})$ by their average velocity $\frac{v_max(n)+v_max(n+1)}{2}$:

$$f_4(p) = \sum_{i=1}^{K-1} \frac{2d\left(n_i, n_{i+1}\right)}{v_{\max}(n) + v_{\max}\left(n_{i+1}\right)} \tag{5.10}$$

Smoothness

The fifth objective describes the curvature of a path. It is the sum of all angles between two path segments. Said angle between two angles is calculated as $cos(\varphi) = a \cdot b / ||a||\ ||b||$. The smoothness, then, is the sum of all angles in the path:

$$f_5(p) = \sum_{i=2}^{K-1} \arccos\left(\frac{\overline{n_i n_{i-1}} \cdot \mid \overline{n_{i+1} n_i}}{|\overline{n_i n_{i-1}}| \cdot |\overline{n_{i+1} n_i}|}\right) \tag{5.11}$$

5.1.3 Names of Problems

The different problem instances are named according to their features. An example is *CH_X9_ Y9_P1_K3_BF*. The names contain six sections, separated by a backslash. The first section stands for the obstacle type. The possible abbreviations, *NO, CH, LA* refer to the obstacle types, no obstacles, checkerboard pattern and lake obstacles. The second and third section name the size of the map in X- and Y-direction. The fourth section indicates the elevation function. Here, the letter P is followed by a character that refers to the parameter n_h of the hill function. In the fifth section, the digit after K refers to the parameter k of the neighborhood. The last section indicates the backtracking. BF and BT stand for disabled and enabled backtracking. The example refers to a problem with checkerboard obstacles, a size of nine by nine, an elevation of $n_h = 1$, a $k = 3$ neighborhood and no backtracking.

5.2 Evaluation Criteria

In this section the metrics used for evaluation are discussed. It needs to be mentioned that a qualitative evaluation of the output is difficult because the quality of results might be a subjective observation.

5.2.1 Metrics

Hypervolume Indicator

The hypervolume (HV) is a quality indicator for Pareto-fronts. It is calculated as the combined volumes that are being dominated by the individuals of the

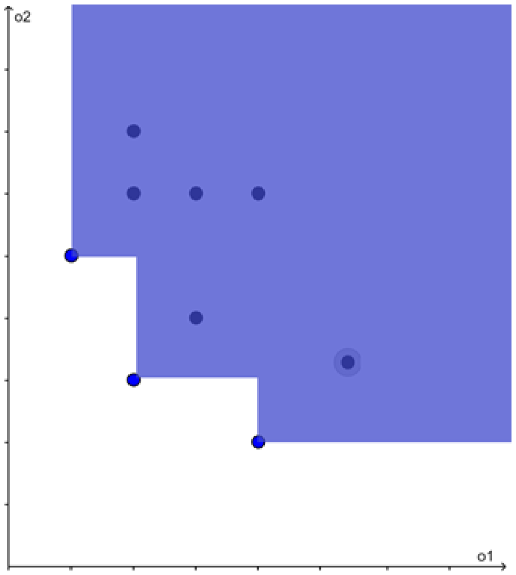

Figure 5.2: Visualization of hypervolume

Pareto front [16]. It is limited by a reference point r [13]. In the illustration in Figure 5.2, the HV can be described as the blue area. It is calculated as

$$HV(S,r) = \lambda(\bigcup_{i=1}^{|S|} v_{i,r}) \tag{5.12}$$

where $v_{i,r}$ refers to the set of all points in the region dominated by x_i, limited by the reference point r. The Lebesgue measure λ assigns a volume-like measure to a set.

In the experiments, hypervolume is used to compare the quality of the initial front with the output of the algorithm. All initial solutions are normalized as stated in Section 4.1. This is done because the objectives are of different dimensions.

Hypervolume Cotribution (HVC)

While the HV can be used to compare different Pareto-fronts, the HVC can be used to rank different solutions according to their importance for the HV

[1]. For solution x_i, it is defined as the part of the HV, the solutions dominates exclusively

$$HVC(x,r) = \lambda \left(v_{i,r} / \bigcup_{j=1, j \neq i}^{|S|} v_{j,r} \right) \tag{5.13}$$

In this evaluation, the HVC is used to compare the importance of selected output solutions. This s done by setting the average HVC of all output solutions in relation to the average HVC of all initial solutions.

Distances between paths

The decision making algorithm needs to put out paths that are different to each other. In order to do that, the Fréchet distances between output paths are used to indicate the spread. The minimal, maximal and average distances of the output are used as indicators.

5.2.2 Further Evaluation Criteria

Optimal Number of Alternatives

The main goal of the proposed MCDM-approach is to provide the user with a selection of representable solutions. On the other hand, it is very important to define a maximum number of solutions, that should be presented. This can be illustrated with the following example: a user is provided with a selection 100 out of 1000 solutions. Obviously, this selected output is smaller than the input of the MCDM-algorithm. However, it would serve little to no purpose because the user would still be overwhelmed by the amount of solutions. Hence, the question arises, what number of solutions a user is still able to process.

Psychological research has been done on how many instances a human can handle in his short-term memory. In [25], Miller found out that this number is in the range of 5 to 9. This number seems to depend on the type of information. Therefor, the lower bound of 5 will be used as a threshold for the number of paths that can be processed by the decision maker.

5.3 Experiments and Remarks

The experiments are performed on problems from [37]. More specifically, the problem instances cover all combinations of problems with:

- a size ranged from three to 14,
- objectives of all different types: no objectives, checkerboard pattern and lakes,
- all different elevation functions
- no backtracking

From the resulting problems, all problems with at least 4 paths are selected. This is done, because the clustering does not output solutions for less than four paths. This leads to 206 problem instances. The results are observed under the assumption that a good output of the DM-algorithm consists of a maximum of five solutions. In decision space, the average distance between paths should be maximized in order to represent the set of solutions. The HV serves to compare objective values of the output with objective values of the initial solutions.

In Section 5.4, the metric values for the output solutions will be represented by relative values like relative HV. This means that the HV of the output is set into relation to the HV of the initial solutions. There will be different visualizations of solutions in decision space. In order for the solutions not to overlap each other, solutions are shifted to the upper left. That way, it is still possible to see the flow of all paths. An example is shown in Figure 5.3.

5.4 Results

In this section, the results of the experiments are presented and discussed.

5.4.1 Results of Clustering

The clustering in decision space is applied to the set of solutions in three variations concerning the used linkage: single linkage only, complete linkage only and best linkage as a combination of both. In the combination, clustering

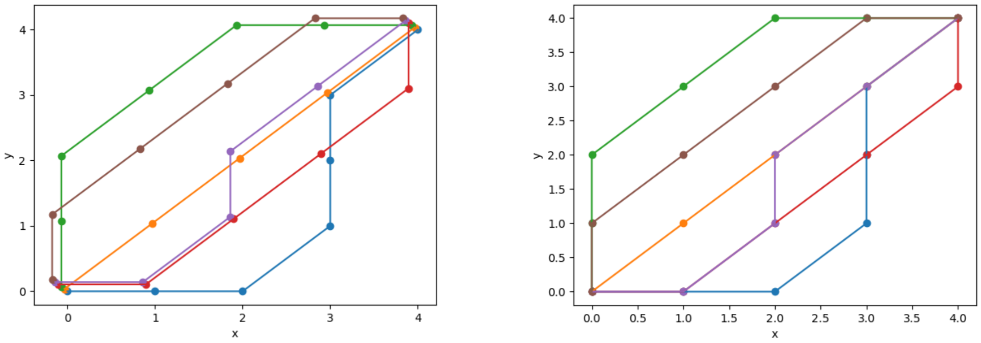

Figure 5.3: Visualization of a set of solutions without shift (left) and with shift (right)

is done once with each linkage and the clustering with the maximal silhouette coefficient is calculated.

As can be seen in Table 5.1, the silhouette coefficient tends to choose two as the preferred number of clusters for all linkages, followed by three, four and five clusters. As a result, for 91.7%, 97.6% and 98.1% of the problems, the algorithm calculated an output with a maximum of 5 solutions using single, complete and best linkage. The silhouette coefficient seems to assign higher scores for a higher number of clusters when single linkage clustering is used. This indicates that single linkage is inferior concerning the reduction of solutions. This might make processing difficult for the decision maker. An extreme example can be seen in Figure 5.4. In this example and other problem instances with a high number of output clusters, the silhouette coefficient falls for a smaller number of clusters till it has reached its minimum and then starts to rise again. While the scores for complete linkage stay positive in these situations, the score for single linkage tends to reach negative values. This indicates a wrong assignment of clusters. For some problems, single linkage clustering does not reach a positive score at all. An example can be seen in Figure 5.5. In this problem, the silhouette score is the highest for two clusters for both linkages. However, for single linkage, it stays negative while it stays positive for complete linkage as it does in all other problems too. As a result, single linkage seems to be worse than complete linkage according to the silhouette coefficient. On the other hand, in some cases it calculates the better solution according to the silhouette coefficient, as Figure 5.6 shows. This shows that using both methods and using the best method according to the score might be a good solution.

number of clusters	single linkage	complete linkage	best linkage
2	150	148	152
3	22	36	34
4	12	12	10
5	5	5	6
6	3	1	1
7	1	1	1
8	2	0	0
9	1	0	0
11	0	1	1
12	1	1	1
14	1	0	0
15	0	1	0
16	1	0	0
18	1	0	0
27	1	0	0
28	1	0	0
30	1	0	0
40	1	0	0
42	1	0	0
50	1	0	0

Table 5.1: Distribution of the number of calculated clusters for the different linkages used

According to Table 5.2, the average count of output solutions for single linkage is 31% higher than for complete linkage and 33% higher than for best linkage. The average relative comparison metrics concerning the path distance of the output are the lowest for sinlge linkage. However, the average relative path distance is still 96.9% of the one for complete and 96.6% of the one for best linkage. This means that the additional solutions that come from single linkage are important to represent the data. It should be kept in mind that too many selected paths might overwhelm the decision maker. In cases with an especially large number of output solutions, the output could be further reduced, either by setting a fixed limit and choosing the best clustering according to that limit and silhouette score, or reducing solutions according to their objective values. An example where single linkage calculates a larger output with a

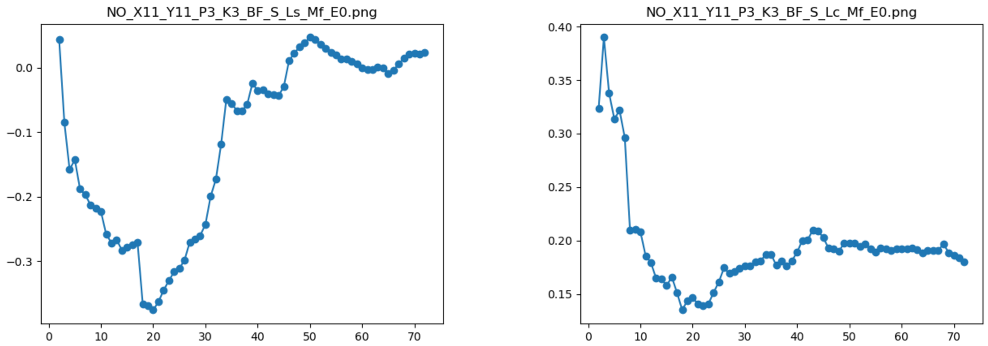

Figure 5.4: silhouette scores for clustering of problem NO_X11_Y11_P3_K3_BF with single linkage(left) and complete linkage(right)

	relative HV		**discrete Fréchet dist**			**#solutions**	
linkage	**HV**	**HVC**	**min**	**max**	**avg**	**total**	**relative**
single	0.3913	0.9034	4.7874	0.761	1.419	3.7184	0.211
complete	0.3893	1.0298	5.0184	0.7679	1.4642	2.5631	0.1987
best	0.3709	0.9756	5.0689	0.7647	1.4692	2.4854	0.1973

Table 5.2: Benchmarks for clustering in decision space, values for output are set in relation to initial values

higher average distance than complete linkage can be seen in Figure 5.7. Here, for complete linkage, the best clustering consists of 2 clusters. In the best clustering for single linkage, an additional cluster consists of one path of each of those two clusters. The initial output candidates remain, while a new candidate increases the average distance between output paths from 6.32 to 6.97.

For single, complete and best linkage, the HV decreases to 39.1%, 38.9% and 37.1%. This is a result of the reduction of solutions. The HVC is only decreased by 3% for complete linkage clustering while it even decreases up to 9.7% for the other linkages. A reason for this might be that the clustering is only done according to decision space and does not consider objective values.

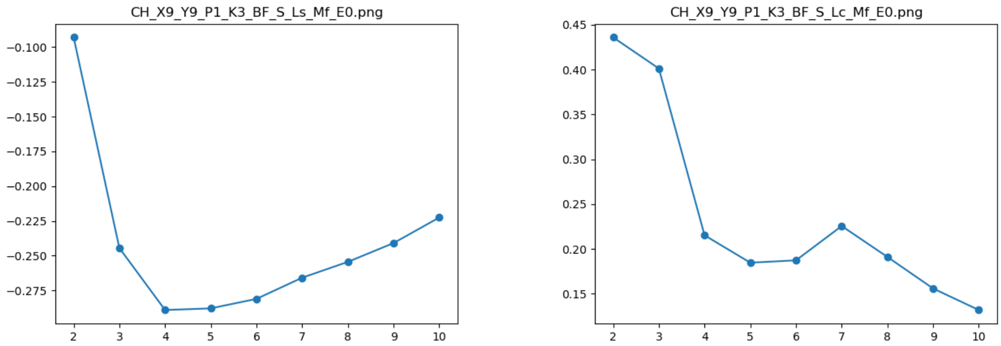

Figure 5.5: silhouette scores for clustering of problem CH_X9_Y9_P1_K3_BF with single linkage(left) and complete linkage(right)

5.4.2 Epsilon- and Cone-Dominance

In order to find good values for ε and φ, different values were used. A value bigger than $\varepsilon = 3.2$ leads to the reduction of solutions to one for most problems. As a result, the values $\varepsilon \in \{0.025, 0.05, 0.1, 0.2, 0.4, 0.8, 1.6, 3.2\}$ were used. As shown in Table 5.3, the number of paths calculated using epsilon-dominance exceeds the threshold of 5 in many cases for $\varepsilon \in \{0.05, 0.1, 0.2\}$ and for greater values it tends to reduce the number of paths to one. However, it needs to be mentioned that for all problems, at least one ε-value created an output with a number of solutions between one and five. For cone-dominance, there were two cases, in which the minimal number of output solutions was 6. Hence, for observations concerning the HV, all ε-value were considered for all problems where their output consisted of a number of paths in the said range. For further evaluation and for each problem, the best solution according to HV was chosen.

Table 5.5 shows the average of HV, HVC, path distances and solution-count of the output of problem-instances, set in relation to the initial values of the problem and the total solution count. As could be inspected, the minimal and average distance values are worse than the values for the different linkages in clustering. In contrast to that, the maximum value is larger. It needs to be considered that a higher number of output solutions was generated. The interesting criterion in objective space, however, is the HV which is decreases compared to the initial Pareto-Front due to the reduction of solutions. In contrast to that, it increases compared to the clusterings. With 74.5% and 80%

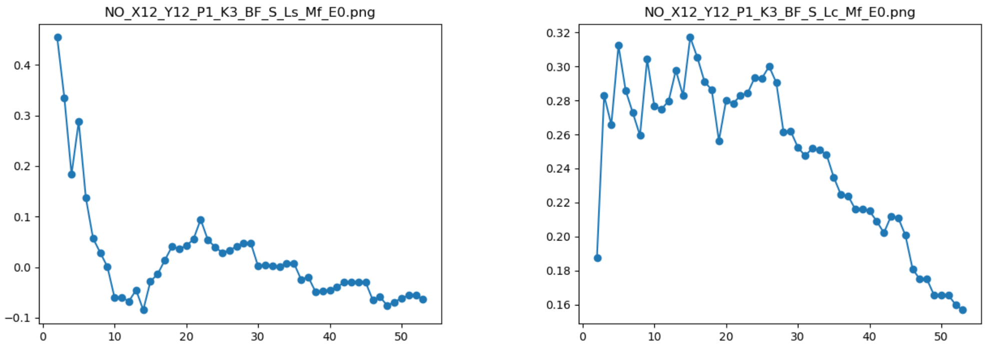

Figure 5.6: silhouette scores for clustering of problem NO_X12_Y12_P1_K3_BF with single linkage(left) and complete linkage(right)

ε	**|output>5|**	**|output=1|**
0.025	153	0
0.05	136	1
0.1	103	3
0.2	40	8
0,4	2	15
0.8	0	80
1.6	0	154
3.2	0	184

Table 5.3: Number of occurrences of output paths for different epsilon-values

of the initial value, the original Pareto-fronts seem to be represented while reducing the output to a size that is processable. In general, epsilon-domination seems to be better concerning the HV compared to cone-dominance. A look at the HVC-values shows an increase to 310% for epsilon-dominance and 507% for cone dominance. This shows that both methods selected important solutions although cone-dominance was superior.

For both domination alternatives, there are problems for which the reduction of solution is especially large compared to loss of HV. An example is LA_X13_Y13_P2_K8 with a HV-loss of 8.1% while reducing the amount of solutions to 8.5% with $\varepsilon = 0.05$. It is illustrated in Figure 5.8. Here the

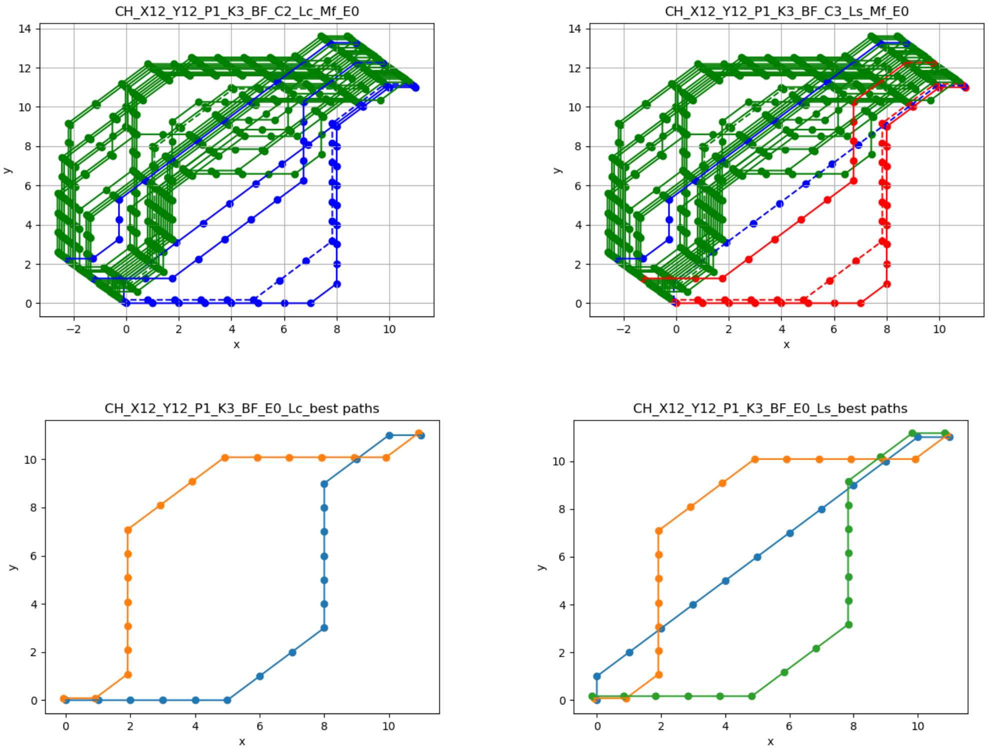

Figure 5.7: Clustering (top pictures) and output candidates (bottom pictures) for clustering problem CH_X12_Y12_P1_K3_BF; complete linkage (left); single linkage (right)

relative HVC is 10.76. On the other hand, there are also problems, for which HV-loss is very high compared to the reduction of solutions. An example, LA_X11_Y11_P2_K3, can be seen in Figure 5.9. Here, the HV is decreased to 9.47% with $\varepsilon = 0.1$ while the number of paths is reduced to 25%. This is also an example of a bad selection of paths concerning the decision space, as the paths initially go in two different directions around the lake-obstacle and the output solutions only go in one direction.

φ in °	|output>5|	|output=1|
91	156	0
92	151	0
93	151	0
95	150	0
100	137	1
110	106	3
120	67	8
150	2	83

Table 5.4: Number of occurrences of output paths for different angles

	relative HV		discrete Fréchet dist			#solutions	
dominance	**HV**	**HVC**	**min**	**max**	**avg**	**total**	**relative**
epsilon	0.7448	3.0968	1.9882	0.8668	1.1852	4.0874	0.3655
cone	0.8	5.0653	1.6125	0.7763	1.0222	3.8529	0.3674

Table 5.5: Benchmarks for domination alternatives, values for output are set in relation to initial values

5.5 Combination of alternative Dominance and Clustering

Table 5.6 shows the benchmark for both variations of the combined version of the algorithm. The selected solutions were obtained by first applying a dominance alternative with previously used parameters, then clustering using best linkage followed by choosing the solution with maximum HV. It shows that the relative HV for both combinations is higher on average than for all clusterings and smaller than for the dominance-alternatives themselves. The same holds for HVC. Interestingly, the HVC for the cone-dominance-combination is reduced to 64.2% compared to only using cone-dominance while it is only reduced to 98.3% for epsilon-dominance. According to relative values for HV and HVC, cone-dominance still shows superior results.

For relative minimal, maximal and average path distance, the combination with epsilon-dominance shows superior results. Compared to best linkage, the results for these metrics change to 96.9%, 99.7% and 99% for epsilon-dominance and to

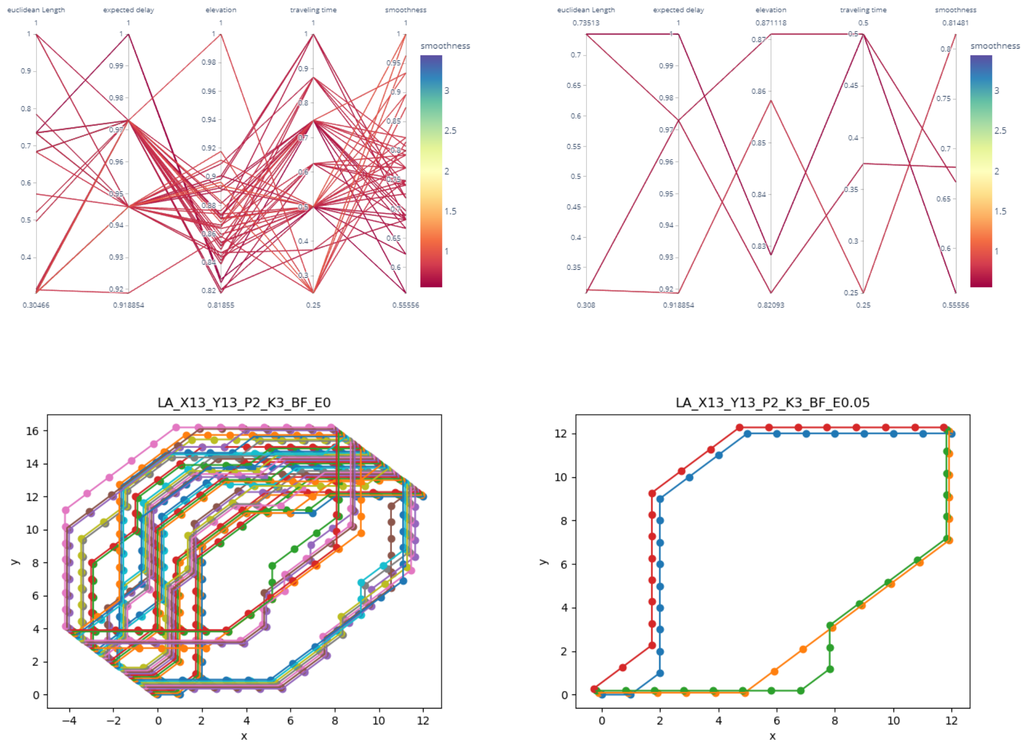

Figure 5.8: Objective values (top pictures) and paths (bottom pictures) for clustering problem LA_X13_Y13_P2_K8; initial paths (left) and for $\varepsilon = 0.05$ (right)

89.2%, 97% and 93.4% for cone-dominance. For both combinations, the minimal and average path distances increase compared to only using the dominance alternatives. However, the maximal distance decreases.

The combined algorithm could not select solutions for 16 problems with cone-dominance and six problems using epsilon-dominance. The reason for this is that the usage of Pareto-alternatives lead to less than four solutions and the clustering step was not possible. NO_X13_Y13_P2_K3_BF was one of the two problems, the cone-dominance reduction put out more than five solutions. For said problem, it selected 6 out of 139 solutions with $\varphi = 150$. Further applying best linkage-clustering returned three solutions. As a result of the reduction of solutions, the HV was reduced to 0.0094%. Illustrations can be seen in Figure 5.10.

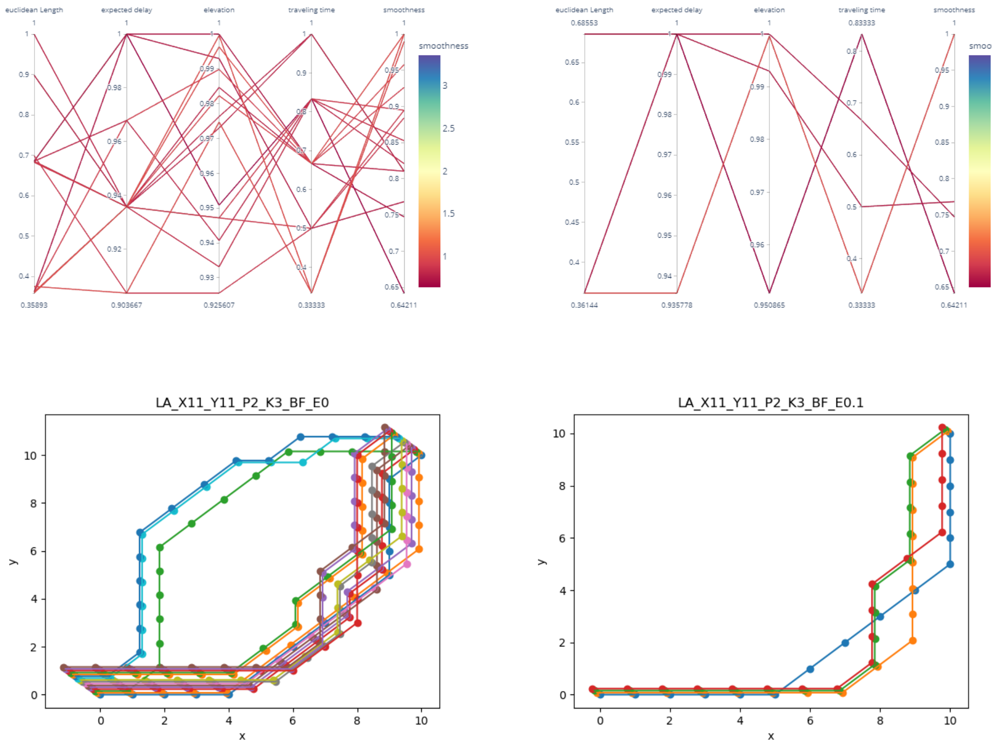

Figure 5.9: Objective values (top pictures) and paths (bottom pictures) for clustering problem LA_X11_Y11_P2_K3; initial paths (left) and for $\varepsilon = 0.1$ (right)

5.6 Summary

The results show that the algorithm is able to reduce the amount of solutions t in all its three variants although the combination of clustering and alternative dominance relations did find solutions for all problems. The results were interpreted under the assumption that an user can process 5 paths. The clustering in decision space was not always able to reduce the amount of solutions below 6. Concerning this criterion, complete linkage was superior to single linkage, while best linkage was even better.

The clustering step is able to increase the average and minimal distances between paths and therefor finds a representative representation of the problems in decision space. For this metric, the dominance-alternatives lead to worse results than clustering. In respect to the path distances, the clustering outper-

	relative HV		**discrete Fréchet dist**			**#solutions**	
dominance	**HV**	**HVC**	**min**	**max**	**avg**	**total**	**relative**
epsilon	0.5583	3.045	4.9108	0.7624	1.455	2.4975	0.1881
cone	0.5792	3.2515	4.5207	0.7419	1.3726	2.5612	0.1893

Table 5.6: Benchmarks for combinations of clustering and alternative dominace metrics, values for output are set in relation to initial values

forms the dominance alternatives. An exception is maximal path distance. On other hand, the step in objective space outputs solutions with higher HV- and HVC-values than the clustering. These metrics were used as indicators for the output quality in objective space. According to them, cone-dominance is superior to epsilon-dominance. A ranking of the results puts the combination of clustering and decision space between the separate use of these steps.

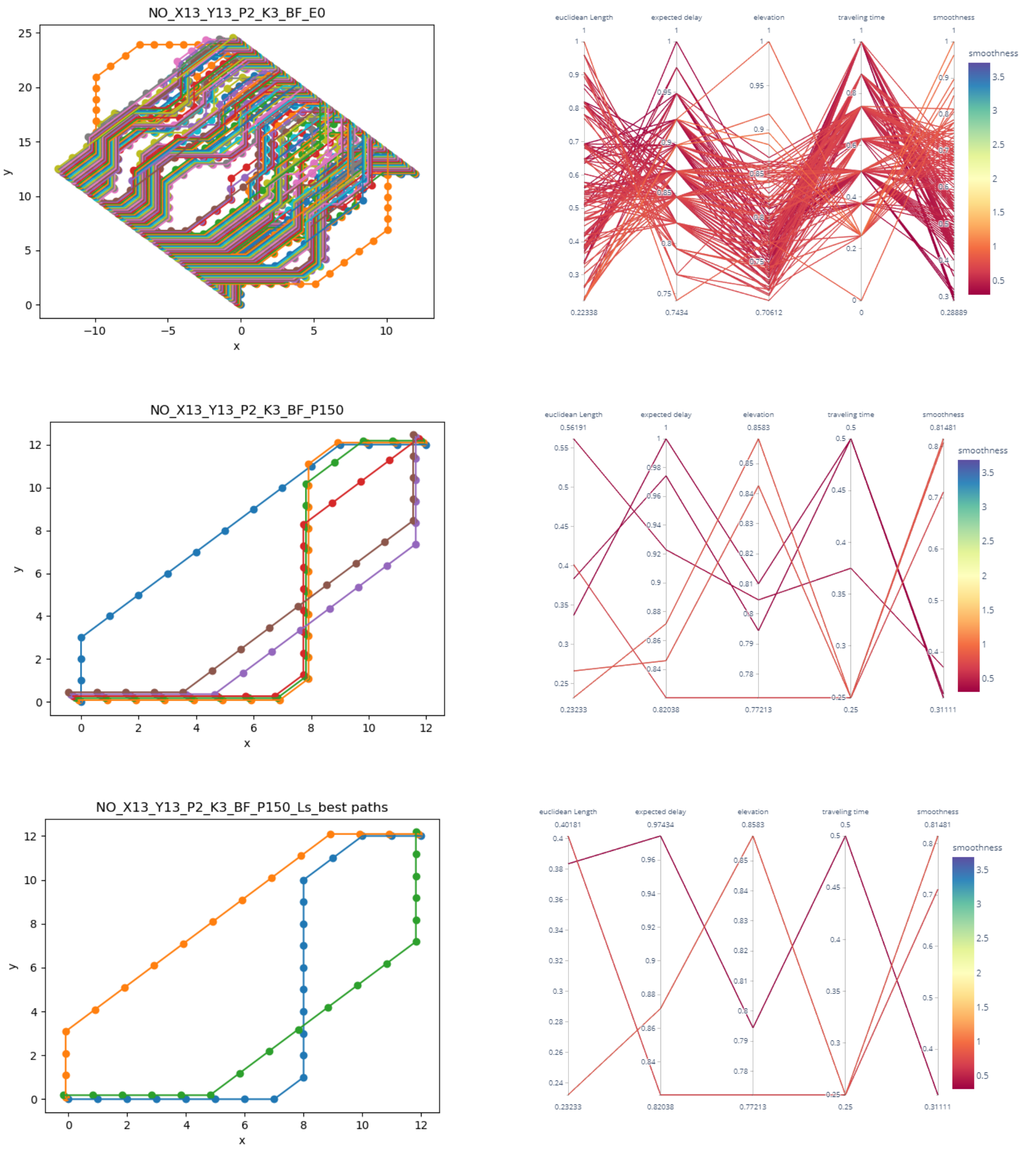

Figure 5.10: Initial solutions (top pictures), solutions obtained with cone-dominance (middle pictures) and solutions obtained by combination of cone-dominance and best linkage (bottom pictures) for clustering problem NO_X13_Y13_P2_K3_BF; paths (left) and objective values(right)

6 Conclusion and Future Work

For the MOPP, little research has been done to find a decision making algorithm. In this thesis, an algorithm for Decision Making for the MOPP has been proposed that supports the user in making decisions according to decision space and objective space. It comes in three versions: DM according to objective values, DM according to the solutions in decision space and a combination of both versions. The results show that the version in decision space seems to select a good representation of solutions according to their representation in decision space for the decision maker that he is able to process. The use of alternatives to Pareto-domination is able to represent more of the initial information in objective space. The combination of both methods is a compromise concerning decision and objective space.

For some problems, the clustering returned more than five solutions. In the future, it could be analyzed how to further reduce those solutions. Additionally, further techniques for objective space can be proposed.

In this thesis, average silhouette score was used to select the best number of clusters although there are alternatives. Future research might compare different techniques for clustering of paths.

Concerning the used evaluation metrics, further research might examine which results can be considered as "good solutions".

Bibliography

[1] Johannes Bader, Kalyanmoy Deb, and Eckart Zitzler. Faster hypervolume-based search using monte carlo sampling. In Matthias Ehrgott, Boris Naujoks, Theodor J. Stewart, and Jyrki Wallenius, editors, *Multiple criteria decision making for sustainable energy and transportation systems*, volume 634 of *Lecture Notes in Economics and Mathematical Systems*, pages 313–326. Springer, Heidelberg u. a., 2010.

[2] Jürgen Branke, Kalyanmoy Deb, Henning Dierolf, and Matthias Osswald. Finding knees in multi-objective optimization. In Xin Yao, Edmund K. Burke, José A. Lozano, Jim Smith, Juan Julián Merelo-Guervós, John A. Bullinaria, Jonathan E. Rowe, Peter Tiňo, Ata Kabán, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature - PPSN VIII*, Lecture Notes in Computer Science, pages 722–731, Berlin and Heidelberg, 2004. Springer.

[3] Kevin Buchin, Anne Driemel, Natasja van de L'Isle, and André Nusser. klcluster. *27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM SIGSPATIAL GIS 2019*, pages 496–499, 2019.

[4] Shihyen Chen, Bin Ma, and Kaizhong Zhang. On the similarity metric and the distance metric. *Theoretical Computer Science*, 410(24-25):2365–2376, 2009.

[5] Stefan Cobzaş. Completeness with respect to the probabilistic pompeiu-hausdorff metric.

[6] I. Das. On characterizing the "knee" of the pareto curve based on normal-boundary intersection. *Structural optimization*, 18(2-3):107–115, 1999.

[7] David H. Douglas and T. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *undefined*, 1973.

[8] J. C. Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*, 4(1):95–104, 1974.

[9] Brian Everitt. *Cluster analysis.* Wiley series in probability and statistics. Wiley, Chichester, 5. ed. edition, 2011.

[10] Peter C. Fishburn. Additive utilities with incomplete product sets: Application to priorities and assignments. *Operations Research*, 15(3):537–542, 1967.

[11] Peter C. Fishburn. *Utility theory for decision making*, volume no. 18 of *Operations Research Society of America. Publications in operations research.* Wiley, New York, 1970.

[12] M. Maurice Fréchet. Sur quelques points du calcul fonctionnel. *Rendiconti del Circolo Matematico di Palermo (1884-1940)*, 22(1):1–72, 1906.

[13] Andreia P. Guerreiro, Carlos M. Fonseca, and Luís Paquete. The hypervolume indicator: Problems and algorithms.

[14] Ching-Lai Hwang and Kwangsun Yoon. *Multiple Attribute Decision Making: Methods and Applications A State-of-the-Art Survey*, volume 186 of *Lecture Notes in Economics and Mathematical Systems.* Springer, Berlin and Heidelberg, 1981.

[15] Leonard Kaufman and Peter J. Rousseeuw. *Finding groups in data: An introduction to cluster analysis.* Wiley series in probability and mathematical statistics. Wiley-Interscience, Hoboken, NJ, 2005.

[16] J. D. Knowles, D. W. Corne, and M. Fleischer. Bounded archiving using the lebesgue measure. In *The 2003 Congress on Evolutionary Computation, 2003. CEC '03*, pages 2490–2497 Vol.4, 2003.

[17] Pekka Korhonen, Jyrki Wallenius, and Stanley Zionts. Solving the discrete multiple criteria problem using convex cones. *Management Science*, 30(11):1336–1345, 1984.

[18] Rudolf Kruse, Christian Borgelt, Christian Braune, Sanaz Mostaghim, and Matthias Steinbrecher. *Computational intelligence: A methodological introduction.* Texts in Computer Science. Springer London, London and s.l., 2nd ed. 2016 edition, 2016.

[19] Marco Laumanns, Lothar Thiele, Kalyanmoy Deb, and Eckart Zitzler. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary computation*, 10(3):263–282, 2002.

[20] A. Littmann, A. Schenk, B. Preim, G. Prause, K. Lehmann, A. Roggan, and H.-O Peitgen. Planning of anatomical resections and in situ ablations in oncologic liver surgery. *International Congress Series*, 1256:684–689, 2003.

[21] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[22] Andreas Lubienski. Ways to the target. In Andreas H. Mahnken and Jens Ricke, editors, *CT- and MR-Guided Interventions in Radiology*, pages 55–68. Springer-Verlag, s.l., 2008.

[23] MacQueen, J., author. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297. University of California Press, 1967.

[24] Kaisa Miettinen. *Nonlinear Multiobjective Optimization*, volume 12 of *International Series in Operations Research & Management Science*. Springer, Boston, MA, 1998.

[25] George A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63(2):81–97, 1956.

[26] Nathan R. Sturtevant, Robert Geisberger. A comparison of high-level approaches for speeding up pathfinding.

[27] Muneendra Ojha, Krishna Pratap Singh, Pavan Chakraborty, and Shekhar Verma. A review of multi-objective optimisation and decision making using evolutionary algorithms. *International Journal of Bio-Inspired Computation*, 14(2):69, 2019.

[28] Francisco-Javier Pulido, Lawrence Mandow, and José-Luis Pérez-de-la Cruz. Dimensionality reduction in multiobjective shortest path search. *Computers & Operations Research*, 64:60–70, 2015.

[29] Urs Ramer. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1(3):244–256, 1972.

[30] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[31] Thomas L. Saaty. Decision making with the analytic hierarchy process. *International Journal of Services Sciences*, 2008.

[32] John Tabak. *Geometry: The language of space and form.* The history of mathematics. Facts On File, New York, 2004.

[33] Thomas Eiter and Heikki Mannila. *Computing Discrete Frechet Distance.* 1994.

[34] Paolo Toth and Daniele Vigo. *Vehicle routing: Problems, methods, and applications*, volume 18 of *MOS-SIAM series on optimization.* SIAM Society for Industrial and Applied Mathematics and Mathematical Optimization Society, Philadelphia, second edition edition, 2014.

[35] Evangelos Triantaphyllou. *Multi-criteria Decision Making Methods: A Comparative Study*, volume 44 of *Applied Optimization.* Springer, Boston, MA, 2000.

[36] Lihui Wang, Amos H. C. Ng, and Kalyanmoy Deb. *Multi-objective evolutionary optimisation for product design and manufacturing.* Springer, London, 2011.

[37] Jens Weise and Sanaz Mostaghim. A scalable many-objective pathfinding benchmark suite. *IEEE Transactions on Evolutionary Computation*, page 1, 2021.

[38] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu. Understanding of internal clustering validation measures. In *2010 IEEE International Conference on Data Mining*, pages 911–916, 2010.

[39] Xin-She Yang and Xing-Shi He. *Nature-inspired computation in data mining and machine learning*, volume 855 of *Studies in computational intelligence.* Springer, Cham Switzerland, 2020.