Natural Computing (2023) 22:315–328
https://doi.org/10.1007/s11047-022-09908-z

A comparison of distance metrics for the multi-objective pathfinding problem

Jens Weise[1] · Sanaz Mostaghim[1]

Accepted: 25 July 2022 / Published online: 19 August 2022

Abstract
Pathfinding, also known as route planning, is one of the most important aspects of logistics, robotics, and other applications where engineers must balance many competing interests. There is a significant challenge in pathfinding problems with multiple objectives because many paths can map to the same objective value. Such multi-modal solutions cannot easily be found in multi-objective optimisation algorithms, which are typically geared towards selection mechanisms in the objective space. A niching approach for preserving good diverse solutions in the decision space is proposed in this paper, which is tailored for pathfinding problems. The criteria used to compare the solutions within the decision space are path similarity metrics, which we extend from a previous study, and are used instead of the well-established crowding distance. In two variations, we investigate the proposed meta-heuristic approach on a range of benchmark instances and compare the methodology to a deterministic optimisation approach.

Keywords Pathfinding · Multi-objective optimization · Evolutionary algorithm · Path similarities · Shortest path · Niching

1 Introduction

Optimal route planning (or pathfinding) is among the most challenging tasks for industrial and logistical applications (Koenig and Likhachev 2005). Any improvement in the results can significantly impact several factors, such as fuel consumption and the environment. Current state-of-the-art route planning algorithms typically consider the travel time, and the distance in the optimisation. In a recent work, we have proposed a scalable benchmark suite for pathfinding problems (Weise and Mostaghim 2021b) and define five objective functions. In a recent study (Weise and Mostaghim 2021c), we investigated a new niching methodology for the pathfinding problem and this paper extends our previous work.

There is an extensive amount of literature in the field of route planning and pathfinding in general, especially for vehicle route planning which uses evolutionary algorithms (Ahmed and Deb 2013). The most important feature concerns the solution and problem representation, which can define the size of the search space and influence the efficiency of the algorithms (Weise et al 2021). However, there is a limited amount of literature using evolutionary algorithms for many-objective pathfinding, i.e., when more than three objectives are to be optimised. Tozer et al (2017) provide an overview of existing approaches and use reinforcement learning to address the problem with six objective functions. Pulido et al (2015) introduce a dimensionality reduction technique to minimise dominance checks during the optimisation and tested their algorithm on a map from a real-world application. In addition, they extended the NAMOA* algorithm, which was first introduced by Mandow and De La Cruz (2010) and is a multi-objective extension to the well-known A* algorithm (Hart et al 1968). In our previous work, we have proposed the *ASLETISMAC* benchmark suite (Weise and Mostaghim 2021b) which includes several maps with various configurations for elevation profiles, neighbourhood metrics and

✉ Jens Weise
jens.weise@ovgu.de

Sanaz Mostaghim
sanaz.mostaghim@ovgu.de

[1] Faculty of Computer Science, Otto von Guericke University, Universitätsplatz 2, 39106 Magdeburg, Germany

backtracking options to test and assess multi-objective pathfinding algorithms. In one of our recent studies (Weise and Mostaghim 2020), we have found out that many of the obtained optimal paths have a considerable overlap with each other, and it is very challenging to diversify the search in the decision space while solving the many-objective pathfinding problem.

This paper aims to address this problem by introducing a specialised similarity measurement of paths into the multi-objective optimisation algorithms. We propose using three different metrics to measure similarities between two paths and incorporate this measurement in the algorithms in two different ways, as a replacement for crowding distance. Instead of applying the niching methodology in the objective space, we apply a customised methodology in the decision space.

In a previous study, we have evaluated how the usage of Fréchet distance can help in the diversification during the search process (Weise and Mostaghim 2021c). We proposed using a dissimilarity matrix to find dense areas in the decision space. In this paper, we extend the approach and evaluate two additional path similarity metrics. Furthermore, we implement the crowding measurement in two variations, extending our previous work, which proposed using one of them. In addition, we propose a methodology to solve the used benchmark using classical multi-objective shortest path techniques, such as a multi-objective version of the well-known Dijkstra algorithm. We create a larger set of true Pareto-fronts for the used benchmark, compared to the original study in Weise and Mostaghim (2021b). Eventually, we compare the exact approach with our meta-heuristic approach with a new performance indicator, that combines runtime and quality of solutions.

The remainder of the paper is structured as follows: In Sect. 2 we introduce the pathfinding problem, its characteristics and niching, while Sect. 3 presents the similarity measurements. Section 4 is dedicated to the proposed multi-objective algorithm, and Sect. 5 describes the deterministic and exact approach. Finally, Sect. 6 describes the experiments and discusses the results, and the paper is concluded in Sect. 7.

2 Background

In this section, we describe the pathfinding problem, its various representations and a benchmark for the evaluation of multi-objective pathfinding algorithms. Furthermore, we discuss niching techniques.

2.1 The multi-objective pathfinding problem

Typically, there are two main ways of representing paths: The first is a variable-length chromosome representation of a solution which is often used with the graph-based problem representation (Jun and Qingbao 2010; Weise and Mostaghim 2021b). This approach represents a solution as a list of nodes varying in length when computing a path. The second approach is a fixed-length chromosome, representing the directions of travel, together with a list of nodes in a graph or a list of grid cells (Besada-Portas et al 2013; Beke et al 2020; Weise et al 2021). Grid-based representations for pathfinding problems are shown to be very practical for evolutionary algorithms (Ahmed and Deb 2013; Weise and Mostaghim 2021a). Such grid representations can be refined depending on the required resolution of the problem. Besides, they are often used for benchmarking purposes (Sturtevant 2012). Also, they can represent real-world problems by discretising the problem representation (Anguelov 2011). Grids typically consist of units with adjustable sizes (Anbuselvi 2013). A solution encoding can consist of a linked-list of units (Xiao and Michalewicz 1999), the directions (Weise et al 2021), or the coordinates of several waypoints (Weise et al 2020). Considering grids, we can introduce several constraints for movements on specific paths by defining an upper limit for movements (such as speed) on each unit. In this way, we can easily incorporate various linear and non-linear constraints on each unit to represent speed, ascent, obstacles, and others.

It is comparatively easy to convert a grid into a graph (also called lattice-graphs) by considering units as nodes and their contact-edges as the graph's edges. This conversion is performed in several applications, e.g. the game industry. The commonly used A* algorithm is an example of pathfinding on a grid which is transferred to a graph (Yap 2002), as its heuristic function can be defined as the Euclidean distance that can measured on a grid between the cell centres.

In general, compared to grid-based representations, graph-based representations allow higher flexibility in representing real-world problems, which can be considered heterogeneous, while grids are usually homogeneous. In this paper, we represent the pathfinding problem as a grid transferred to a graph. By employing the *property-graph-data-model*, we can assign various properties to each node (Rodriguez and Neubauer 2010)

The multi-objective pathfinding problem can be defined as follows. The goal is to find a set of k Pareto optimal paths $P^* = \{p_1, \ldots, p_k\}$ in a graph $G(V, E)$ from a starting node $n_S \in V$ to a pre-defined end node $n_{End} \in V$, i.e., $p_i = (n_S, \ldots, n_{End})$, where each path is of a variable length

K that is evaluated concerning a set of objective functions $\vec{f}(p)$ which are to be minimised. Given this definition, we can use a variable-length chromosome representation to apply the proposed next distance metrics. Figure 1 depicts a graph which is superimposed on a grid (lattice-graph) in the left figure and two paths p_i and p_{i+1} in the right figure.

In Weise and Mostaghim (2021b), we proposed a benchmark suite, where we have defined five objective functions for a multi-objective pathfinding problem. Note that problems with more than three objectives are considered as *many-objective* (Köppen and Yoshida 2007).

The benchmark instances themselves are defined on a grid with a size of (x_{max}, y_{max}); however, in the original study, the authors used a graph-based representation, where each grid-cell was represented by a node n in the set of vertices V in a graph $G = (V, E)$. Neighbourhood relations between cells, set by a benchmark's instance settings, are reflected in the set of edges E. Each edge defines if the traversal from one cell to one of its neighbours is allowed, as a directed graph is used.

Therefore, following the original study Weise and Mostaghim (2021b), the objective functions are defined for a list of nodes $n \in N$, $N \subseteq V$. A list of nodes, of variable length K, $N_i = (n_1, \ldots, n_K)$ represents a path p_i, that consists of a list of subsequent coordinates, i.e. $p_i = (x_1, \ldots, x_K)$, where $x_i \in \mathbb{R}^2$. Therefore, each node's n_i location is defined by a two-dimensional coordinate.

The paths are evaluated by: the length of a path, travel time, delay, smoothness (or curvature) of a path, and ascent (or elevation) of a path. In the following, the equations to compute each objective are shown:

Objective 1: Euclidean length. In the first objective function, the Euclidean lengths between two neighbours are summed.

$$f_1(N) = \sum_{i=1}^{K-1} d_{euc}(n_i, n_{i+1}) \tag{1}$$

Objective 2: Expected delays. The second objective can be considered as a measurement of running in an accident or any other unexpected event. It is defined by the differences in v_{max} of two consecutive cells or nodes.

$$f_2(N) = \sum_{i=1}^{K-1} delay(n_i, n_{i+1}) \tag{2}$$

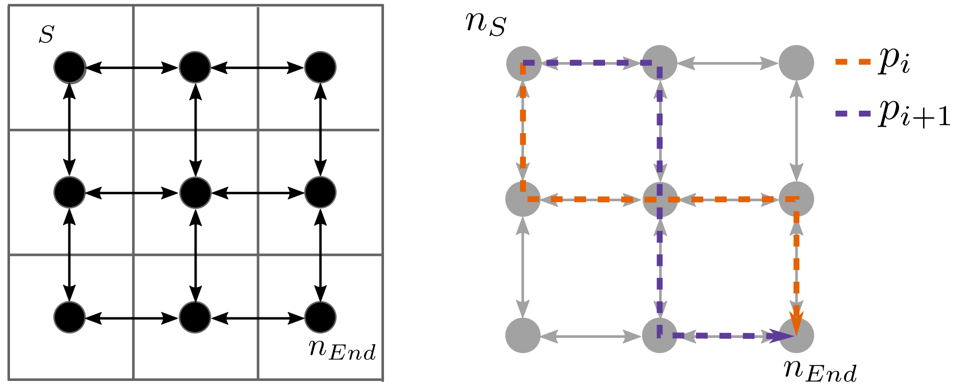
Objective 3: Elevation. The third objective corresponds to the total positive ascent when traversing a path. Only positive values are considered, since this objective can represent fuel consumption to a certain extent.

$$\begin{aligned} f_3(N) &= \sum_{i=1}^{K-1} e(n_i, n_{i+1}) \\ e(m, n) &= \begin{cases} h(n) - h(m), & \text{if } h(n) > h(m) \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{3}$$

Objective 4: Travelling time. The fourth objective corresponds to the total time needed to traverse a path, and is defined by the distance between two nodes and the average v_{max}.

$$f_4(N) = \sum_{i=1}^{K-1} \frac{2 \cdot d(n_i, n_{i+1})}{v_{max}(n) + v_{max}(n_{i+1})} \tag{4}$$

Objective 5: Smoothness. The last objective corresponds to the smoothness of the path. However, since we intend to minimise the objectives, a smaller objective value represents a more straight path. Similar to Oleiwi et al (2014) and Jun and Qingbao (2010), $a \cdot b = \|a\|\|b\| cos(\theta)$ is inverted:



(a) Superimposed graph on a grid

(b) Two paths p_i and p_{i+1} which go along nodes and edges.

Fig. 1 A grid is transferred to a graph (grid or lattice graph), to enable graph-based pathfinding algorithms to find solutions (Weise and Mostaghim 2021c)

$$f_5(N) = \sum_{i=2}^{K-1} arccos\left(\frac{\overrightarrow{n_i n_{i-1}} \cdot \overrightarrow{n_{i+1} n_i}}{\|\overrightarrow{n_i n_{i-1}}\| \cdot \|\overrightarrow{n_{i+1} n_i}\|}\right) \tag{5}$$

For our evaluation, we take the same definitions for the obstacle-defining functions, $v_{max}(n)$, and $delay(n_i, n_{i+1})$ as in the original study (Weise and Mostaghim 2021b):

$$v_{max}(x, y) = \begin{cases} 130, & \text{if} w(x, y) > 0.9 \\ 50, & \text{if} w(x, y) < -0.4 \\ 100, & \text{else} \end{cases} \tag{6}$$
$$w(x, y) = \max(sin(x-1), cos(y-1))$$

$$delay(n_i, n_{i+1}) \begin{cases} 2 & \text{if } v_{max}(n_i) \neq v_{max}(n_{i+1}) \\ 3 & \text{if } v_{max}(n_i) = v_{max}(n_{i+1}) = 50 \\ 1 & \text{if } v_{max}(n_i) = v_{max}(n_{i+1}) = 100 \\ \frac{1}{5} & \text{otherwise} \end{cases} \tag{7}$$

The function v_{max} uses the node n's coordinates that are denoted by x and y, $n_i = (x_i, y_i)$. For more in-depth information and how $h(n)$ is computed, the reader is referred to the original study (Weise and Mostaghim 2021b).

In analogy to the original study, we, furthermore, also compute paths from the cell or node with the coordinates of (1, 1) to cell with the coordinates of (x_{max}, y_{max}), which are given by the specified benchmark instance problem.

It is noteworthy that all objectives, except the *Smoothness*, i.e. f_5, can be computed for a path of length $k \geq 2$, while at least three nodes are needed to compute the fifth objective. It is important, as this restricts several other pathfinding algorithms to solve the problem. Usually, edge weights are taken into account, which is only possible for the first four objectives, as the function value can be used as an edge weight in the underlying graph representing the problem.

2.2 Niching

As Shir noted, genetic algorithms typically suffer from loss of diversity within populations, resulting in a local optimum (Shir 2012). Niching methods address this problem by preserving diversity. In state-of-the-art algorithms, there are various methods for increasing or maintaining diversity. Crowding distance is an example of using the objective space to identify crowded areas. Individuals are measured by the average distance between their two neighbouring solutions (Deb et al 2002). More isolated solutions can be emphasized by using this measurement during the NSGA-II algorithm's replacement and selection phase. Another technique is to use reference vectors. The solutions along such vectors are generally preferred, and their distribution within the objective space allows diversity to be maintained (Deb and Jain 2014). When optimizing more than three objectives, the latter methodology often serves as a good solution to problems with many objectives. However, study results indicate that performance can be affected not only by the number of objectives but also by the type of problem, requiring careful consideration when choosing an algorithm (Cai et al 2018; Weise and Mostaghim 2021b).

Moreover, if more than one solution can map to the same objective values in specific problems, such as multimodal problems, other diversity measurements such as combining different metrics in the objective and decision space can be useful (Javadi et al 2020; Deb and Tiwari 2005). For instance, the many-objective pathfinding problem has close solutions in the objective space, though they are far apart in the decision space. The use of diversity-preserving measurements in decision space can, as a result, improve an algorithm's performance (Shir et al 2009; Weise and Mostaghim 2021c). Shir states that diversity along the Pareto front does not necessarily result in the same diversity in the decision space among the Pareto set. Futhermore, a decision space-diverse set is of interest for a potential decision maker (Shir et al 2009; Ulrich et al 2010). In their work, Ulrich et al (2010) proposed a methodology to integrate the decision space diversity into a hypervolume based search. However, there are numerous difficulties in calculating a measure of diversity in the decision space, which is especially true if a variable-length representation represents the solution. The crowding distance, in addition to other metrics, such as the harmonic mean, can be used when the chromosomes are fixed-length (Javadi and Mostaghim 2021).

3 Paths similarities

Path, or curve, similarity measurement is found in several fields. For instance, in handwriting recognition, curves are compared to match letters or words (Sriraghavendra et al 2007). Other fields include morphing (Efrat et al 2002) and protein structure alignment (Jiang et al 2008).

There are different methods of path similarity measurements. This article evaluates three different metrics, i.e. Hausdorff distance, Fréchet distance, and dynamic time warping (DTW). With Hausdorff distance, the distance of two subsets (curves) in a metric space can be measured (Munkres 2000), while Fréchet distance also takes the flow of curves into account. Fan et al. used the Fréchet distance on road networks for measuring the resemblance of road tracks (Fan et al 2011). Assuming two subsets in the same metric space, they can have a short Hausdorff distance but a rather large Fréchet distance. Originating from signal theory, DTW finds a warping path between two curves or signals to align them, and its path length determines the similarity between the signals, and therefore curves. In the following, we describe each of the distance metrics in more detail.

3.1 Hausdorff distance

The *Hausdorff distance*, can be used to compute the distance between two sets of points without taking their flow into account. In Eq. 8 its formula is depicted. It is the greatest distance of all distances between points in one set and their nearest points in the other.

$$\begin{aligned} D(x,K) &:= \min\{d(x,k) \mid k \in K\} \\ \delta_{hd}(A,B) &:= \max\{\max\{D(a,B) \mid a \in A\}, \max\{D(b,A) \mid b \in B\}\} \end{aligned} \tag{8}$$

3.2 Fréchet distance

The Fréchet distance is a measurement of similarity for curves in a metric space. Eiter and Mannila described it by using a dog walk analogy (Eiter and Mannila 1994). A dog and its owner walk on two different paths, and both can vary their speed but cannot backtrack. Since there is a leash attached to both, the Fréchet distance can be defined as the shortest length of a leash, which is required for both to follow their paths as shown in Fig. 2. The dashed lines indicate the leash.

Fréchet distance (Eiter and Mannila 1994) is mathematically defined as follows:

$$\delta_F(A,B) = \inf_{\alpha,\beta} \max_{t\in[0,1]} \left\{ d\Big(A(\alpha(t)),\ B(\beta(t))\Big) \right\} \tag{9}$$

Here, A and B are curves as a continuous mapping in a metric space S, defined as $A : [0,1] \rightarrow S$ and $B : [0,1] \rightarrow S$. The reparametrisations α and β are non-decreasing functions from [0, 1] to [0, 1]. In other words, $A(\alpha(t))$ and $B(\beta(t))$ are the positions of the owner and the dog in timestep t and they are non-decreasing as the two entities cannot move backwards. The function d is the distance metric in S, e.g. Euclidean distance, and hence it describes the length of the leash between the dog and the owner. Eventually, the Fréchet distance is obtained by computing the infimum (greatest lowest bound) of all parameterizations α and β of [0, 1] of the maximum over all t of the distance d in S between $A(\alpha(t))$ and $B(\beta(t))$. Alt and Godau studied the computational properties of the measurement (Alt and Godau 1995), initially introduced by Fréchet (1906). They specified an algorithm to compute the distance δ_F in time $\mathcal{O}(ab\log^2 ab)$, where a and b are the number of segments of two curves (Eiter and Mannila 1994). Eiter and Mannila proposed a variation of the distance metric, i.e., the coupling distance, or discrete Fréchet distance δ_{dF}, which provides a good approximation of δ_F in time $\mathcal{O}(ab)$. They also show that δ_{dF} is an upper bound of δ_F. They approximate the two curves A and B by polygonal curves, which is, intuitively, a list of supporting points. Those points specify the endpoints of line segments. The sequence of segment endpoints of a polygonal curve P is denoted as $\sigma(A) = (u_1, \ldots, u_a)$. A *coupling* L between A and B is defined as a sequence $L = (u_{c_1}, v_{d_1}), (u_{c_2}, v_{d_2}), \ldots, (u_{c_m}, v_{d_m})$ of distinct pairs of $\sigma(A) \times \sigma(B)$, with respect to $c_1 = 1, d_1 = 1, c_m = a, d_m = b$. The coupling respects the order of points. Eventually, the length of the coupling is defined as $||L|| = \max_{i=1,\ldots,m} d(u_{c_i}, v_{d_i})$; hence the longest connection in L and the discrete Fréchet distance as shown in Eq. 10 (Eiter and Mannila 1994).

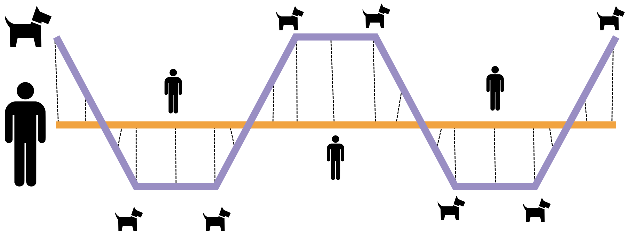


Fig. 2 An example of a dog walk (Weise and Mostaghim 2021c)

$$\delta_{dF}(A,B) = \min\{||L|| | L \,\text{is a coupling between}\, A \,\text{and}\, B\} \tag{10}$$

In more recent works, subquadratic algorithms were developed to approximate the discrete Fréchet distance (Bringmann and Mulzer 2016). The authors developed an algorithm that runs in $\mathcal{O}(n\log n + \frac{n^2}{\alpha})$, where n is the number of points and $\alpha \in [1, n]$ is the approximation. The two curves have to have the same number of nodes. Chan and Rahmati (2018) also proposed an approximation algorithm resulting in $\mathcal{O}(n\log n + \frac{n^2}{f^2})$, where f denotes the approximation and is $f \in [1, n]$. In this paper, we use all points of the two curves. We can therefore use the original methodology by Eiter and Mannila (1994). Using the more recent approaches with all points would result in $\alpha = f = n$, resulting in $\mathcal{O}(n\log n)$. However, the polygonal curves in this paper are not constituted of the same number of nodes. We, therefore, use Eiter and Mannila's approach.

Compared to the Hausdorff distance, the Fréchet distance accounts for the flow of the curves, while the Hausdorff distance measures the distance from one point in one curve to the closest on the second curve. For the remainder of the paper, we refer to the discrete Fréchet distance. As we define a path in this work as $p_i = (n_s, \ldots, n_{End})$, we can represent it by analogy to the definition of δ_{dF}; hence $p_i = \sigma(A)$ and

$(n_s, \ldots, n_{End}) = (u_1, \ldots, u_a)$, where $n_s = u_1$, $n_{s+1} = u_2$, $n_{End} = u_a$. In other words, A represents a continuously defined curve and $\sigma(A)$ are the segment endpoints. Since a path in a graph is defined by its nodes, every node $n_i \in p_i$ is, in fact, a segment endpoint.

3.3 Dynamic time warping

Although the Fréchet distance is a metric that results in a dissimilarity measurement of two curves, there is another metric that can find the best match between two signals and determine their distance (Müller 2007). The Dynamic Time Warping (DTW) measures similarities between temporal sequences. However, we can use the approach to determine a distance between two paths, which are sequences of locations. In Eq. 11 the formal definition of the DTW distance is shown. For its computation, a local cost measurement $d(a, b)$ is needed that describes the similarity between two points $a \in A$ and $b \in B$, e.g., the Euclidean distance. A cost matrix $C(A, B)$ can be built by computing $d(a, b)$ for each pair of a and b. The goal is to find an optimal warping path p^* of the two curves A of length R and B of length S that the overall cost (Eq. 11a) of a warping path p is minimal (Eq. 11c). An (R, S)-warping path p is a path in the cost matrix that runs along a *valley* of low cost (Müller 2007). As the matrix' values represent cost, the warping path is the one obtaining the least combined costs, if it can go through the matrix' cells (Eq. 11c). In this path, the element $a_{r_\ell} \in A$ is assigned to the element $b_{s_\ell} \in B$. To compute δ_{dtw}, dynamic programming has been used.

$$c_p(A,B) := \sum_{\ell=1}^{L} d(a_{r_\ell}, b_{s_\ell}) \tag{11a}$$

$$\delta_{dtw}(A,B) := c_{p^*}(A,B) \tag{11b}$$

$$\delta_{dtw}(A,B) = \min\{c_p(A,B) \mid p \,\text{is an}\, (R,S)\text{−warping path}\} \tag{11c}$$

Figure 3 shows two curves and their respective distance values using each of the three proposed metrics. In all three metrics, the function $d(\cdot,\cdot)$ represents the Euclidean distance between two points.

4 Path similarity-based NSGA-II

To preserve the diversity of solutions (paths) in the decision space, we use each of the proposed path similarity measurements in the well-known NSGA-II algorithm (Deb et al 2002). Although NSGA-II has some drawbacks on many-objective problems, we saw in our previous study (Weise and Mostaghim 2021b) that it outperformed NSGA-III (Deb and Jain 2014) in the majority of our problem instances. The problem's Pareto-fronts are usually irregular and degenerate. Due to the use of crowding distance, NSGA-II is more robust to these types of problems, as evenly distributed reference vectors which are used in NSGA-III can lead to the same solution (Cai et al 2018). Furthermore, the pathfinding problem is partially deceptive, which can lead the algorithms to get stuck in local optima. We use the three proposed distance metrics and aim to understand their impact on the algorithm results.

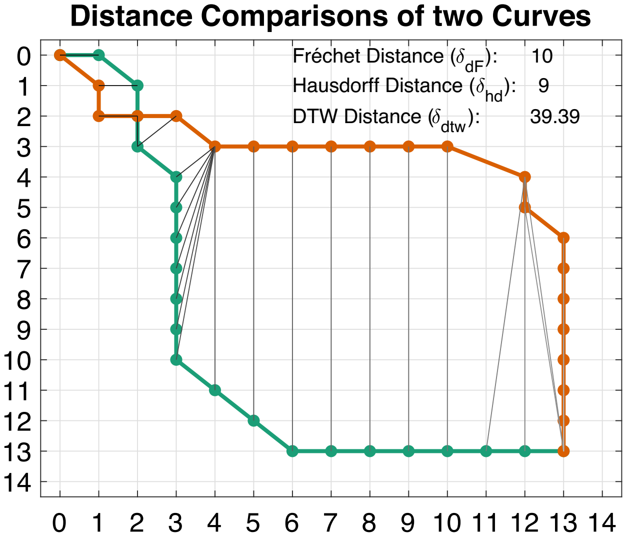


Fig. 3 Comparison of the three used distance metrics on two curves (orange and green)

In our algorithm's approach, we replace the crowding distance used in the NSGA-II algorithm with the proposed distance metric in the decision space. In other words, when the next parent population is filled and the last front cannot be added completely, we compare the solutions by using δ_{hd}, δ_{dF}, or δ_{dtw}. Moreover, we use this distance metric in the selection process. For this purpose, we implement a path similarity sorting method. We compute a dissimilarity matrix for N paths in a population and assign either the lowest distance of all $N-1$ values to each path or the median distance to $N-1$ paths as described in algorithm 1. The function ψ describes if to use $min(\cdot)$ or $median(\cdot)$ as the crowding measurement.

Algorithm 1: Crowding Path similarity (Weise and Mostaghim, 2021c), adapted

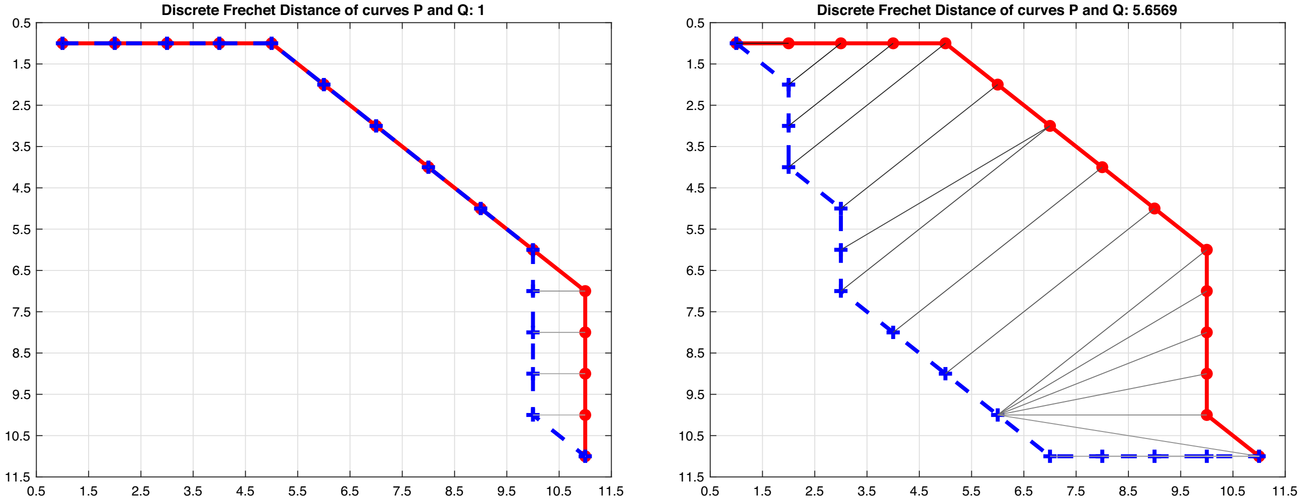
```
Input: List of Paths $P = \{p_1, \cdots, p_k\}$
// Map is a key-value store. A path $p$ is the key with the
   minimum distance as its value.
Result: Map(Path,minimum distance)
// Every Path $p_i \in P$ gets $\infty$ assigned
Map results = Map(Paths,Infinity);
// The distance $\delta$ from every path to every other is
   computed
// $\{(p_i, p_j) | p_i, p_j \in P, p_i \neq p_j\}$
for $i$=1 to $k$ do
    for $j$=$i$+1 to $k$ do
        $\delta_{old}(p_i)$ = Map($p_i$);
        $\delta_{current}(p_i)$=$\delta$($p_i$,$p_j$);
        Map($p_i$)=$\psi$($\delta_{old}(p_i)$,$\delta_{current}(p_i)$);
    end
end
```

In this way, paths with less similarity have a higher distance value. Figure 4 illustrates an example with several possible paths on a benchmark instance. Figure 4a shows two of the paths and their respective discrete Fréchet distance of 1, while Fig. 4b shows two rather distinct paths with large distance values of 5.65. After computing every pair, the paths are sorted by their distance value in descending order. This algorithm is called *NSGA-II-CR-* δ-ψ, where δ and ψ are exchanged with their respective implementation when running the algorithm. For instance, if $\delta = \delta_{FD}$ and $\psi = min()$, then is algorithm is called *NSGA-II-CR-FD-MIN*.

5 Deterministic and exact approaches

We also compare our approach in terms of performance to an exact approach, i.e., the multi-objective Dijkstra shortest path algorithm (Martins 1984). However, a comparison with exact methods is not trivial, since algorithms, such as Dijkstra or A* usually use edge-weights to evaluate a path's cost. In the classical multi-objective shortest path problem, a path p of length k is defined by a sequence of edges in the graph, i.e., $p = (e_1, \ldots, e_k)$, where $e_i \in E$, for $i = 1, \ldots, k$ and its cost is defined by the sum of the edges' weight vectors, i.e., $z(p) = (z_1(p), \ldots, z_m(p))$, where $z_j(p) = \sum_{l=1}^{k} w_j(e_l)$ and $\vec{w(e)} = (w_1(e), \ldots, w_m(e))$, where m is the number of cost values or objectives (Gandibleux et al 2006).

In the benchmark from Weise and Mostaghim (2021b), the objectives *cannot* be represented as a weight vector $\vec{w(e)}$ on each edge $e \in E$. The reason is the smoothness objective that depends on the location of three nodes in a sequence. In Eq. 5, it is visible, that this objective takes three nodes as its input parameters. It, therefore, cannot be expressed as a weight value assigned to an edge connecting only two nodes. It always depends on three nodes. Therefore, an edge's weight vector $\vec{w(e_i)}$ depends on the previous edge on the path, i.e., $\vec{w(e_{i-1})}$. To apply conventional multi-objective pathfinding methodologies to these problems, the problem must be reduced to a regular multi-objective pathfinding problem by transferring the graph $G = (V, E)$ to a new graph $G' = (V', E')$. A node is denoted by $(\cdot)$, and a directed edge between two nodes is denoted by an arrow, i.e., $(\cdot) \rightarrow (\cdot)$. In graph G', each pair of nodes, i.e. $(q, r), q, r \in V$, with $(q) \rightarrow (r) \in E$ in graph G, is represented by a new node $(q, r) \in V'$. Furthermore, an edge from $(q, r) \rightarrow (r, s)$ is added to E', that represents the traversal from r to s, assuming q was the traversed node before r. The pathfinding problem has pre-determined start and target nodes, i.e., $s, t \in V$. Therefore, we add the nodes



(a) Fréchet distance of two close paths

(b) Fréchet distance of two distinct paths

Fig. 4 Different discrete Fréchet distance values (Weise and Mostaghim 2021c)

$s^+ \in V'$ and $t^+ \in V'$ and edges $\gamma = (s^+) \rightarrow (s,q) \in E'$, with $w_i(\gamma) = f_i(s,q)$, for $i = 1,\ldots,4$, but $w_i(\gamma) = 0$, for $i = 5$, and $\eta = (q,t) \rightarrow (t^+) \in E'$ with $w(\vec{\eta}) = \vec{0}$, i.e, the zero vector.

Although the graph transformation enables conventional multi-objective shortest path algorithms to solve the problem, the number of nodes and edges in graph G' can be substantially higher compared to those in G, depending on the specified problem instance. For K3 instances with enabled backtracking, the number of nodes is defined by $|V'| = 4(x_{max} - 1)(2x_{max} - 1) + 2$ and the number of edges $|E'| = 4(x_{max} - 1)(16x_{max} - 23) + 6$. Given these functions, it becomes clear that the number of nodes and edges grows quadratically, and influences the algorithms' performance. In Fig. 5, an example is visible, showing the original graph G in Fig. 5a and the transferred graph G' in Fig. 5b.

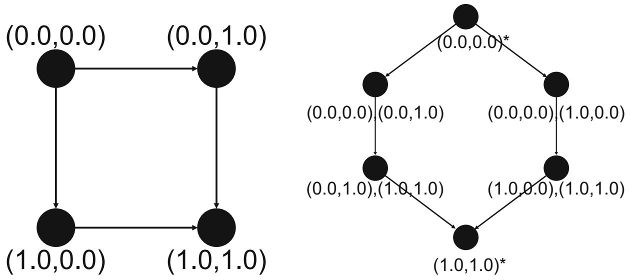
To set the edges' weight vectors, we can employ the objective functions from the benchmark to compute the weights for each $e \in E'$. The proposed functions are path-based and have an arbitrary path p of length $k > 1$ as an input, where k is the number of nodes in p_i. Again, all objective functions, except the *smoothness* objective, need two consecutive nodes to be computed. The *smoothness* (f_5, Eq. 5), however, is based on three nodes to compute the angle between them. The edge's weight vectors are set as shown in Eq. 12, where the inputs of the objective functions are node lists.

$$\vec{w}((q,r) \rightarrow (r,s)) = (f_1((r,s)), f_2((r,s)), f_3((r,s)), f_4((r,s)), f_5((q,r,s))) \tag{12}$$

We have employed the multi-objective Dijkstra on a variety of instances of different sizes and different properties, to find the true Pareto fronts and sets.

6 Experiments and results

In this section, we describe the experimental settings and present and discuss the results.



(a) Original Graph G (b) Transferred Graph G'

Fig. 5 Comparison of the original problem graph G and the transferred graph G'. In brackets: the respective grid coordinates

6.1 Settings

In the experiments, we examine the proposed approaches on the benchmark test suite ASLETISMAC (Weise and Mostaghim 2021b). We consider a two-dimensional space with three obstacle types (NO, LA, CH), with K3 neighbourhood, enabled backtracking, and grid sizes of $\{15, 20, 24, 26, 28, 30\}$. NO indicates no obstacles, while LA and CH introduce bulk and chequerboard obstacles, as shown in Fig. 6. In this way, CH constraints the decision space to a few feasible paths. The K3 neighbourhood restricts the number of possible neighbours to eight, i.e., all surrounding cells. All these combinations result in 84 test instances. Given a solution represented by a path N of length K as $N = (n_i, n_{i+1}, \ldots, n_K)$, i.e., is a list of nodes, we evaluate it by five objectives to be minimised: (1) Euclidean length, (2) Delays, (3) Elevation, (4) Travelling time and (5) Smoothness (Curvature). The details can be found in Weise and Mostaghim (2021b).

We use the same operators for pathfinding as in Weise and Mostaghim (2021b) i.e., a one-point crossover, which creates new offspring chromosomes by crossing two parent paths at one common point. We also use the proposed perimeter mutation for the mutation operator, which mutates the middle point of two arbitrary points within a specific network distance inside a given maximum radius and reconnects the paths afterwards. Consequently, we compare the algorithms with the three incorporated distance metrics to each other. Although it is a many-objective optimisation problem, we consider a smaller computational budget than our previous studies, as we have observed that the quality of results changes only marginally after 100 generations. With that decision, we want to take time performance considerations into account by sacrificing quality. Furthermore, we use a population size of 100, to further account for fewer function evaluations. In real-world applications, obtaining results in a short amount of time is often a requirement. For the experiments, we calculate the IGD$^+$ indicator that is a distance measurement between the obtained front of non-dominated solutions and the known true Pareto reference front (Ishibuchi et al 2015). Furthermore, we report the respecting wins, losses and ties of each algorithm of all problem instances.

To assess the quality of solutions in the decision space, we also employ the IGDX indicator (Aimin Zhou et al 2009), which measures the distance between the known Pareto-set and the found solutions. Again, we compare the results of the algorithms to each other and test for statistical significance using Bonferroni correction, as we perform multiple comparisons. Our null hypothesis states, that the populations have equal medians.

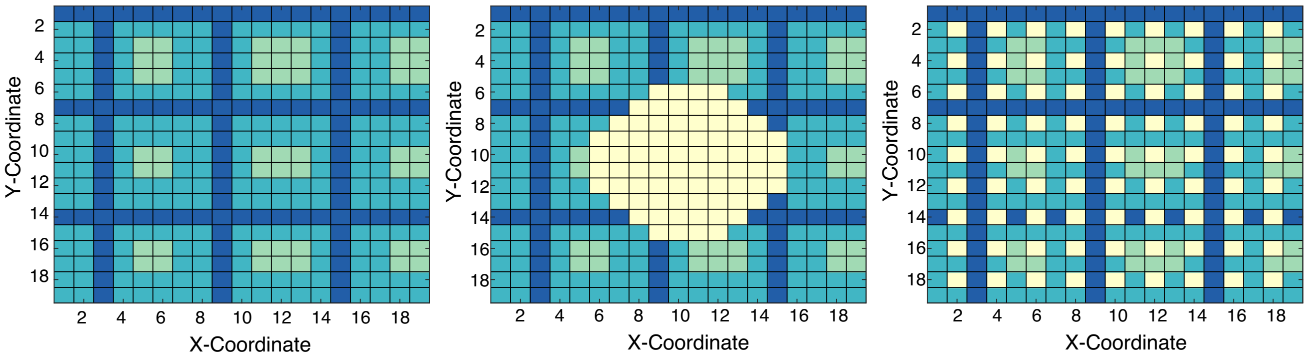


Fig. 6 Examples for various maps with different properties for units representing NO, LA and CH (from left to right) (Weise and Mostaghim 2021c)

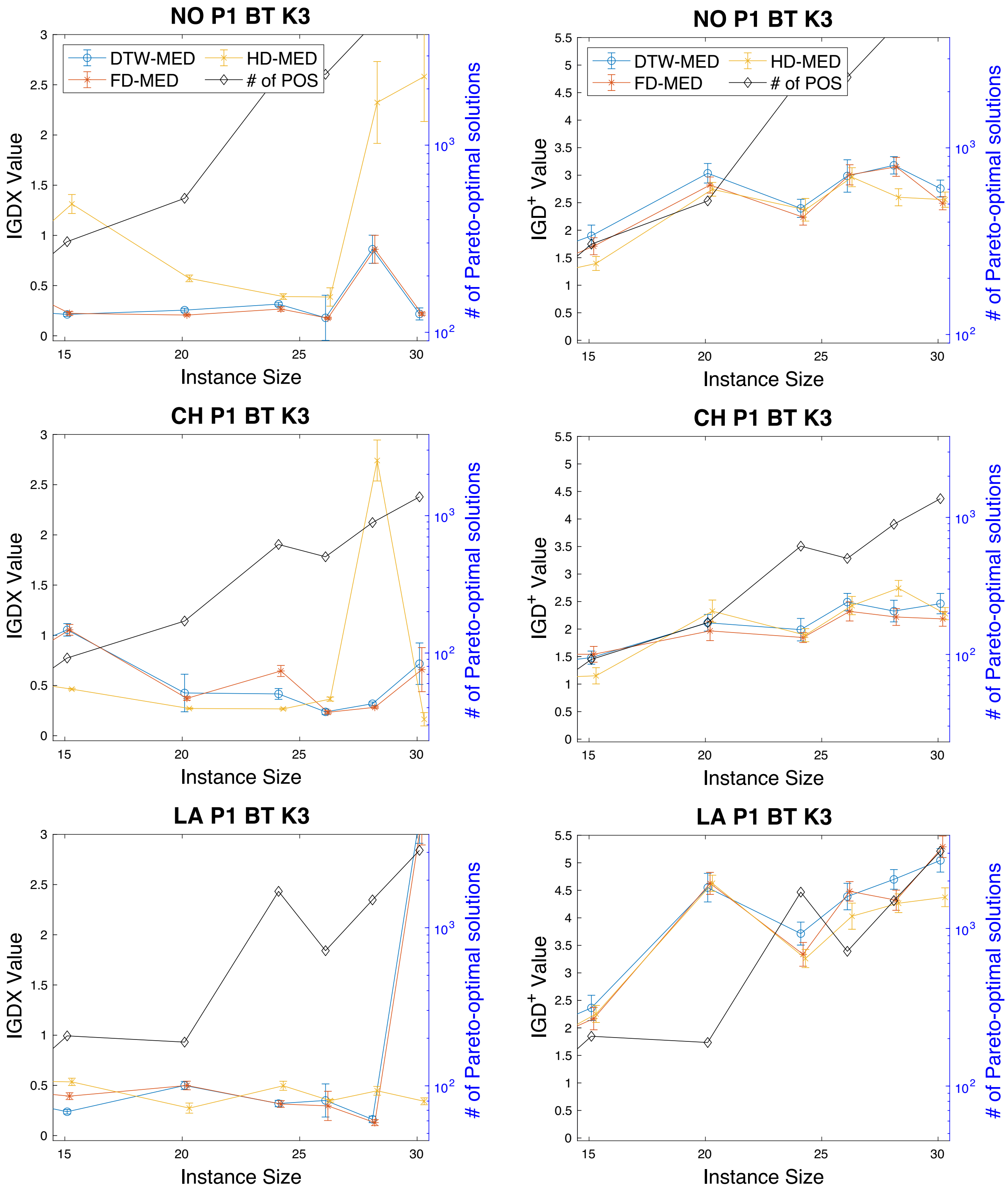
Naturally, it is not trivial to compare an exact algorithm to a meta-heuristic. However, since the running time of the multi-objective Dijkstra is directly related to the number of nodes and edges, we have decided to combine running times with the quality of results of the algorithms to compare them. Therefore, we make use of the indicator values. Our comparison metric is $\lambda_I = median(t_I) \cdot (1 + median(\mathrm{I}))$, i.e., the median run time t_I of an algorithm on a problem times the median indicator value I added to 1. With such a metric, we take the suboptimality of meta-heuristics into account. However, the indicator value I for the optimal algorithm is naturally 0, where I is the IGD$^+$ or IGDX indicator.

6.2 Results

Figure 7 illustrates the median values and standard error of the IGDX and IGD$^+$ indicators respectively, concerning the instance sizes and obstacle settings. Each row in the figure shows a different obstacle profile, i.e. *NO*, *CH*, or *LA*.

We also report the number of Pareto-optimal solutions for each problem instance on the right axis, which we obtained using the exact approach. From a graphical perspective, it is visible that the results vary depending on the type and size of the problem instance. It is noteworthy that in the instance of LA P1 BT K3 (bottom left) of size 28, the algorithms using Fréchet and DTW distance obtain a small IGDX value, compared to a relatively high IGD$^+$ value. This indicates that solutions near the optimal solutions in the decision space have been found, that are, however, of low quality in the objective space. For the multi-objective pathfinding problem, we have shown that solutions close to the optimum in decision space are not necessarily close to the optimum in terms of the respective objective values. However, the same two algorithms obtain a worse IGDX value on instance size 30 for the same map type, while the algorithm incorporating the Hausdorff distance can still obtain a low value. Nevertheless, in the other two obstacle settings, using the Hausdorff distance is not as stable as the other two distance metrics regarding IGDX values, as backtracking is allowed, and therefore, the metrics taking the flow of a curve into account give better results. If a path gets closer to its origin, whereas another path does not, the Hausdorff distance is not always able to compute their similarity. As a result, the outcome can be small although the Fréchet and DTW distance give higher values, as the paths are more distinct. Concerning the IGD$^+$ indicator, all three algorithms obtained similar values. The indicators' results show that using the proposed niching methodology can improve the quality of solutions in terms of closeness to the true Pareto-set, while there is still room for improvement regarding the objective values. However, as we have limited the search to 10,000 function evaluations, which is comparatively low for many-objective optimisation problems, the results in terms of IGD$^+$ were expected. In this study, the search process took particularly measurements in the decision space to minimise the objective functions. We, therefore, do not see much improvement in the objective space that is measured using IGD$^+$. The underlying problem is deceptive, which can result in paths that are close to an optimal solution in the decision space (measured using IGDX) but far from the optimum in the objective space. For a real-world application, the impact would be, that a slight perturbation when executing or traversing a path can result in a severe deterioration in terms of objective functions. Future research can work on more advanced methodologies, that focus on local optimisation, as there is only a small portion of the path that needs to be changed to result in better objective values.

Table 1 shows the wins, losses, and ties of each algorithm on all the 84 test problems. Again, the differences concerning the two performance indicators can be seen, as the majority of outcomes regarding IGD$^+$ are ties, while several instances have a definite winner, comparing the IGDX values.



(a) IGDX results

(b) IGD$^+$ results

Fig. 7 IGDX and IGD$^+$ over instance sizes. Top, middle and bottom rows illustrate NO, CH and LA obstacles

Table 1 Wins, Losses and Ties of each algorithm pair (rows vs. coloumn) with statistical significance at $p < 0.01$, Bonferroni correction applied, IGDX and IGD^{+} indicator

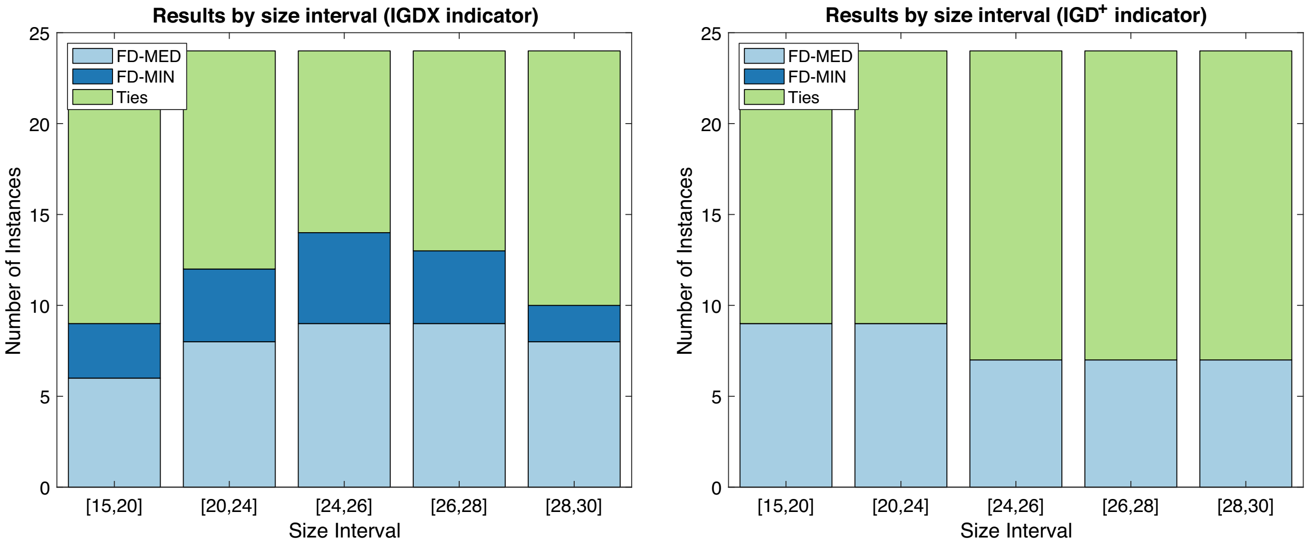
	FD-MED	HD-MED
DTW-MED		
IGDX	3/6/75	23/16/45
IGD^{+}	0/0/84	0/6/78
FD-MED		
IGDX		23/14/47
IGD^{+}		0/2/82

In Fig. 8 we compare two variants of the algorithm using δ_{dF}, i.e. (1) taking the median value of all distances to other paths, denoted by *FD-MED*, and (2) taking the minimum value of the distances, denoted by *FD-MIN*. The two variants are again compared concerning IGDX (Fig. 8a) and IGD^{+} (Fig. 8b). Interestingly, while in terms of IGD^{+} *FD-MED* wins or is of the same quality as *FD-MIN*, the latter wins in several instances when comparing regarding IGDX. The reason is that the *min*() approach can work better for local optimisation since the closest paths are used as a reference.

A more detailed view can be seen in Fig. 9, where the respective indicator values over different sizes for a specific instance type are depicted. Clearly, *FD-MIN* obtains better or similar values when comparing the IGDX values, while it gets outperformed by *FD-MED* with respect to IGD^{+}.

When analysing the results, it becomes clear that the high number of ties indicates room for improvement. The high number of ties are a result of the path similarity sorting. Sorting, given two subsets of distinct paths with relatively short distance values inside the subset, will result in a short distance value for each path, although the two sets can be apart from each other.

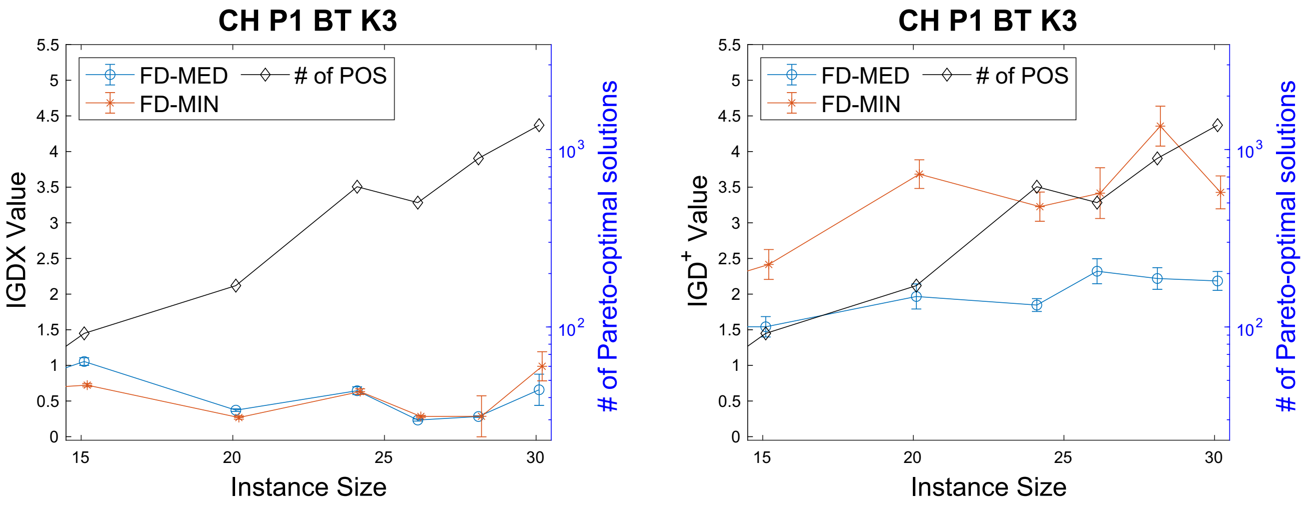
We also compared the performance of the algorithms to an exact approach as we want to evaluate the quality and running time, when we only allow a small computational budget of 10,000 evaluations. Figure 10 shows the values of the proposed λ-performance for the two employed quality indicators. The y-axis is on a logarithmic scale. In the left graph, λ_{IGDX} shows that the proposed approach has a better value from instance sizes of 26 onwards, while the values are lower from size 28 for $\lambda_{IGD^{+}}$, compared to the exact approach. Nevertheless, a trend is visible that shows lower λ values for larger map sizes. That is to be expected as the running time for the exact approach grows quadratically. Interestingly, the DTW approach achieved the best performance in most runs. Although it is computationally expensive, it can diversify the solution set better since it is more sensitive to path differences. Nevertheless, the λ indicator is an approximation that cannot be used to make decisions about significant differences between algorithms.



(a) IGDX results

(b) IGD^{+} results

Fig. 8 Wins/losses/ties of the algorithm incorporating Fréchet distance with respect to the IGD^{+} and IGDX indicators over different instance size intervals



(a) IGDX results

(b) IGD^+ results

Fig. 9 Indicator values of the instance CH P1 K3 BT for different sizes, comparing *FD-MED* and *FD-MIN*

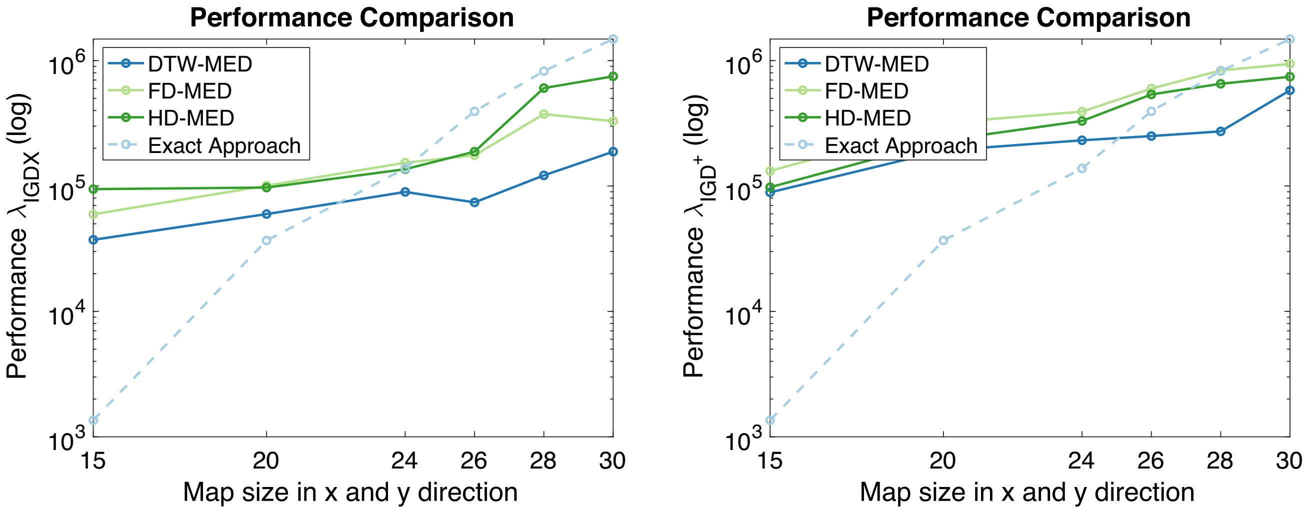


Fig. 10 Results of the λ performance measurement

7 Conclusion

In this paper, we have extended our study in Weise and Mostaghim (2021c) to evaluate different path similarity metrics. In the previous study, we have already shown that using such a metric can positively influence the quality of results in the objective space. We implemented two further distance metrics for path similarity calculation and incorporated them into the NSGA-II algorithm in two different ways, i.e. specifying the usage of either the minimum or median of all distances. We replaced the crowding distance measurement in objective space with the new distance metric in decision space. When examining the results in the objective space, all three algorithms gave similar results compared to each other. However, when comparing the quality of solutions in terms of closeness to the true Pareto-set in the decision space, we have found that all three distance metrics perform well, with a few exceptions, depending on the benchmark instance type and size. Comparing the two crowding measurement type, i.e. *min* and *median*, we have found that the *median* variant is at least as good as the *min* variant in terms of objective values; however, there are differences regarding the IGDX indicator, where *FD-MIN* outperformed *FD-MED* in several instances.

Additionally, we have compared our approach to a multi-objective version of the well-known Dijkstra algorithm. A significant drawback of the comparison to exact approaches is that more recent approaches have not been tested. Pulido et al.'s dimensionality reduction technique (Pulido et al 2015) is one of the approaches that should be compared to the meta-heuristic approach in the future. However, on larger maps with more than three objectives, a meta-heuristic can be beneficial to compute good results in a short amount of time. In Pulido's article, the authors set a time limit of eight hours, which was exceeded in some experiments, mainly when their algorithm was applied to a large real-world map considering three objectives.

There is still room for improvement and research by modifying the usage of the path similarity metric in the algorithms. For instance, the proposed path similarity sorting can be developed more in the future by altering the way it is calculated and used. Furthermore, as shown in Weise and Mostaghim (2021a), using an ordering of paths can lead to better results. Nevertheless, different similarity metrics should be evaluated as well. Maheshwari et al. proposed a Fréchet distance measurement, incorporating certain speed limits, which could lead to better results (Maheshwari et al 2011). Moreover, combining both crowding distance in objective space and a path similarity metric in decision space could lead to better results, which was already shown by Javadi et al (2019). Moreover, computing the hypervolume in the decision space, as shown in Ulrich et al (2010), is an interesting path. However, using a path similarity measurement function to compute the hypervolume remains future research. In Deb and Tiwari (2005), the *Omni-Optimizer* methodology was proposed. For multi-objective multi-modal problems, a combination of the crowding distances in the objective and the decision space is presented, i.e., to use the maximum of both values as a solution's final crowding distance. Multi-modal problems create solutions that have the same objective values but are different in their respective decision variables. Deb's methodology preserves such solutions and emphasises isolated solutions in both spaces. This approach is of interest to our proposed problem and analysed in future research. In the future, we want to investigate other benchmark instances, e.g. with a larger size and actual real-world data. Additionally, we want to incorporate the proposed distance measurement in other state-of-the-art many-objective algorithms. Furthermore, we want to evaluate adaptive approaches of decomposition-based algorithms, e.g., an adaptive version of the NSGA-III algorithm.

Author contributions JW: Conception and design of the work; programming, acquisition, analysis, interpretation of data; writing. SM: Conception, analysis, editing, supervision

Funding Open Access funding enabled and organized by Projekt DEAL. This work was financially supported by the German Federal Ministry of Education and Research through the MOSAIK project (Grant No. 01IS18070B)

Data availability Pareto-fronts, sets, codes and and a problem generator can be found at https://ci.ovgu.de/Publications/Scalable+Many_Objective+Pathfinding+Benchmark+Suite-p-910.html

References

Ahmed F, Deb K (2013) Multi-objective optimal path planning using elitist non-dominated sorting genetic algorithms. Soft Comput 17(7):1283–1299

Zhou Aimin, Zhang Qingfu, Jin Yaochu (2009) Approximating the set of pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm. IEEE Trans Evol Comput 13(5):1167–1189

Alt H, Godau M (1995) Computing the Fréchet distance between two polygonal curves. Int J Comput Geom Appl (01n02) 05:75–91

Anbuselvi R (2013) Path finding solutions for grid based graph. Adv Comput Int J 4(2):51–60

Anguelov B (2011) Video game pathfinding and improvements to discrete search on grid-based maps. PhD thesis. University of Pretoria

Beke L, Weiszer M, Chen J (2020) A comparison of genetic representations for multi-objective shortest path problems on multigraphs, vol 8. Springer International Publishing, Berlin, pp 35–50

Besada-Portas E, de la Torre L, Moreno A et al (2013) On the performance comparison of multi-objective evolutionary UAV path planners. Inf Sci 238:111–125

Bringmann K, Mulzer W (2016) Approximability of the discrete Fréchet distance. J Comput Geom 7(2):46–76

Cai X, Sun H, Fan Z (2018) A diversity indicator based on reference vectors for many-objective optimization. Inf Sci 430–431:467–486

Chan TM, Rahmati Z (2018) An improved approximation algorithm for the discrete Fréchet distance. Inf Process Lett 138:72–74

Deb K, Jain H (2014) An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. IEEE Trans Evol Comput 18(4):577–601

Deb K, Tiwari S (2005) Omni-optimizer: a procedure for single and multi-objective optimization. Lect Notes Comput Sci 3410:47–61

Deb K, Pratap A, Agarwal S et al (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197

Efrat A, Guibas LJ, Har-Peled S et al (2002) New similarity measures between polylines with applications to morphing and polygon sweeping. Discrete Comput Geom 28(4):535–569

Eiter T, Mannila H (1994) Computing discrete Fréchet distance. Tech. Report CD-TR 94/64, Christian Doppler Lab. Expert Sys., TU Vienna, Austria

Fan C, Luo J, Zhu B (2011) Fréchet-distance on road networks. In: Akiyama J, Bo J, Kano M, Tan X (eds) Computational geometry, graphs and applications. CGGA 2010. Lecture Notes in Computer Science, vol 7033. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-24983-9_7

Fréchet MM (1906) Sur quelques points du calcul fonctionnel. Rendiconti del Circolo Matematico di Palermo (1884–1940) 22(1):1–72

Gandibleux X, Beugnies F, Randriamasy S (2006) Martins' algorithm revisited for multi-objective shortest path problems with a MaxMin cost function. 4OR 4(1):47–59

Hart PE, Nilsson NJ, Raphael B (1968) A formal basis for the heuristic determination of minimum cost paths. IEEE Trans Syst Sci Cybern 4(2):100–107

Ishibuchi H, Masuda H, Nojima Y (2015) A study on performance evaluation ability of a modified inverted generational distance indicator. In: Proceedings of the 2015 on genetic and evolutionary computation conference—GECCO '15. ACM Press, New York, pp 695–702

Javadi M, Mostaghim S (2021) Using neighborhood-based density measures for multimodal multi-objective optimization. In: Evolutionary multi-criterion optimization. EMO 2021. Lecture Notes in Computer Science, vol 12654. Springer, Cham. https://doi.org/10.1007/978-3-030-72062-9_27

Javadi M, Zille H, Mostaghim S (2019) Modified crowding distance and mutation for multimodal multi-objective optimization. In: Proceedings of the genetic and evolutionary computation conference companion. ACM, New York, pp 211–212

Javadi M, Ramirez-Atencia C, Mostaghim S (2020) Combining Manhattan and crowding distances in decision space for multimodal multi-objective optimization problems. Springer ECCOMAS book series on computational methods in applied sciences. ACM, Guimarães

Jiang M, Xu Y, Zhu B (2008) Protein structure–structure alignment with discrete Fréchet distance. J Bioinform Comput Biol 06(01):51–64

Jun H, Qingbao Z (2010) Multi-objective mobile robot path planning based on improved genetic algorithm. In: 2010 International conference on intelligent computation technology and automation, vol 2. IEEE, Changsha, pp 752–756

Koenig S, Likhachev M (2005) Fast replanning for navigation in unknown terrain. IEEE Trans Robot 21(3):354–363

Köppen M, Yoshida K (2007) Substitute distance assignments in NSGA-II for handling many-objective optimization problems. LNCS. Evolutionary multi-criterion optimization, vol 4403. Springer, Berlin, pp 727–741

Maheshwari A, Sack JR, Shahbaz K et al (2011) Fréchet distance with speed limits. Comput Geom 44(2):110–120

Mandow L, De La Cruz JLP (2010) Multiobjective A* search with consistent heuristics. J ACM 57(5):1–25

Martins EQV (1984) On a multicriteria shortest path problem. Eur J Oper Res 16(2):236–245

Müller M (2007) Dynamic time warping. Information retrieval for music and motion. Springer, Berlin, pp 69–84

Munkres JR (2000) Topology. Featured titles for topology. Prentice Hall, Incorporated, New York

Oleiwi BK, Roth H, Kazem BI (2014) Modified genetic algorithm based on A* algorithm of multi objective optimization for path planning. J Autom Control Eng 2(4):357–362

Pulido FJJ, Mandow L, Pérez-De-La-Cruz JLL (2015) Dimensionality reduction in multiobjective shortest path search. Comput Oper Res 64:60–70

Rodriguez MA, Neubauer P (2010) Constructions from dots and lines. Bull Am Soc Inf Sci Technol 36(6):35–41

Shir OM (2012) Niching in evolutionary algorithms. In: Handbook of natural computing, vol 1–4. Springer, Berlin, pp 1035–1069

Shir OM, Preuss M, Naujoks B et al (2009) Enhancing decision space diversity in evolutionary multiobjective algorithms. Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics) LNCS, vol 5467, pp 95–109

Sriraghavendra E, Karthik K, Bhattacharyya C (2007) Fréchet distance based approach for searching online handwritten documents. In: Ninth international conference on document analysis and recognition (ICDAR 2007), vol 1. IEEE, Curitiba, Paraná, pp 461–465

Sturtevant NR (2012) Benchmarks for grid-based pathfinding. IEEE Trans Comput Intell AI Games 4(2):144–148

Tozer B, Mazzuchi T, Sarkani S (2017) Many-objective stochastic path finding using reinforcement learning. Expert Syst Appl 72:371–382

Ulrich T, Bader J, Zitzler E (2010) Integrating decision space diversity into hypervolume-based multiobjective search. In: Proceedings of the 12th annual conference on genetic and evolutionary computation—GECCO '10. ACM Press, New York, p 455

Weise J, Mostaghim S (2020) A many-objective route planning benchmark problem for navigation. In: Proceedings of the 2020 genetic and evolutionary computation conference companion, GECCO '20. ACM, New York, pp 183–184

Weise J, Mostaghim S (2021) A customized Niching methodology for the many-objective pathfinding problem. In: 2021 IEEE symposium series on computational intelligence (SSCI). IEEE, Orlando, pp 1–8

Weise J, Mostaghim S (2021) A scalable many-objective pathfinding benchmark suite. IEEE Trans Evol Comput 26:188–194

Weise J, Mostaghim S (2021c) Many-objective pathfinding based on Fréchet similarity metric. In: 11th International conference, EMO 2021, Shenzhen, China, 28–31 Mar 2021, proceedings. 01, pp 375–386

Weise J, Mai S, Zille H et al (2020) On the scalable multi-objective multi-agent pathfinding problem. In: 2020 IEEE congress on evolutionary computation (CEC). IEEE, pp 1–8

Weise J, Zille H, Mostaghim S (2021) A comparative study of different encodings on the multi-objective pathfinding problem. In: 2021 IEEE symposium series on computational intelligence (SSCI). IEEE, Orlando, pp 1–8

Xiao J, Michalewicz Z (1999) An evolutionary computation approach to robot planning and navigation. Soft Comput Mechatron 32:117–141

Yap P (2002) Grid-based path-finding. Lecture notes in computer science (including subseries Lecture notes in artificial intelligence and Lecture notes in bioinformatics), vol 2338, pp 44–55