





Finding Sets of Solutions for Temporal Uncertain Problems

Jens Weise^(✉)  and Sanaz Mostaghim 

Faculty of Computer Science, Otto von Guericke University, Magdeburg, Germany
{jens.weise,sanaz.mostaghim}@ovgu.de
<https://ci.ovgu.de>

Abstract. The multi-objective pathfinding problem is a complex and NP-hard problem with numerous industrial applications. However, the number of non-dominated solutions can often exceed human comprehension capacity. This paper introduces a novel methodology that leverages the concept of a Pareto graph to address this challenge. Unlike previous approaches, our method constructs a graph that relates paths where there is potential for change between them and applies a graph community algorithm to identify solution subsets based on specific aspects defined by a decision-maker. We describe the construction of a Route Change Graph (RCG) to represent possible route changes. A matrix is constructed to save the number of possible change opportunities between two routes, which is then used to construct the RCG. We propose using a threshold value for edge weights in the graph construction, balancing between minimising the number of edges and maintaining connectivity. Following the construction of the RCG, we apply a community detection algorithm to identify closely related solutions, using Leiden algorithm due to its efficiency and refinement phase. We propose calculating various metrics on these communities, including Density, Average Cluster Coefficient, Group Betweenness Centrality, and Graph Degree Centrality, to provide insights into the network structure and interconnectivity. This methodology offers a more manageable set of solutions for decision-makers, enhancing their ability to make informed decisions in complex multi-objective pathfinding problems.

Keywords: pathfinding · decision-support · Pareto graph

1 Introduction

Finding a path from one point to another while optimising multiple objectives is known as the multi-objective pathfinding problem and is considered NP-hard [20]. In several industries this technique can be applied, e.g., route planning, aviation, networking or medical applications [20]. For instance, there are multiple

This work is funded by the German Federal Ministry of Education and Research through the 6G-ANNA project (grant no. 16KISK092).

objectives to consider, when planning a logistic trip for a truck, e.g., curvature of the road, ascent and length of a route. All these objectives should be considered simultaneously. In medical applications, inserting a needle to perform a minimal invasive tumour therapy can include objectives such as distance to the vessel system or damaged tissue. Often, these objectives are in conflict. Applying multi-objective optimisation techniques to such problems, can give a decision-maker (DM) a better insight into the problem. The result of such an optimisation is a set of non-dominated solutions, where no solution is better than the other.

However, the cardinality of the obtained set of non-dominated solutions can exceed the number of solutions, a DM can comprehend [20]. According to Miller, humans can comprehend 7 ± 2 information chunks, although more recent research indicates that this number is lower (approx. 3 to 4 chunks) [11, 16]. Various reduction techniques have been proposed that identify important and interesting solutions.

In this paper, we propose a new methodology that utilises the concept of a Pareto graph [13]. In contrast to the original approach, we construct a graph that sets paths into relation when there is the possibility to change between them. Furthermore, we apply various graph community algorithms to identify subsets of solution that comply with various aspects which can be given by a DM. In contrast to other approaches, our proposed methodology does not reduce the whole set of non-dominated solutions, but finds subsets from which DMs can choose.

The paper is structured as follows. In Sect. 2, we describe the necessary background, while Sect. 3 is dedicated to the related work. Section 4 presents our proposed methodology, divided into graph construction and community detection and analysis. In Sect. 5, we evaluate the results and give a conclusion and outlook in Sect. 6.

2 Background

In this section, we present the relevant background, i.e., the multi-objective pathfinding problem, various aspects of graph theory, including community detection.

2.1 Graph Theory

Graphs are used to represent the relations between entities. A graph G consists of a set of vertices that are the representations of such entities and a set of edges that denote the relations. An edge usually consists of an unordered or ordered set of two vertices. Formally, a directed graph G is a pair $G = (V, E)$, where V denotes the set of vertices and E is the set of edges, where $E \subseteq \{(n, n') \mid (n, n') \in V^2, n \neq n', n, n' \in V\}$. Note that E consists of two-element ordered subsets of V^2 , which renders a graph *directed*. In undirected graphs, by contrast, E consists of two-element unordered subsets of V^2 [23].

Connected Components. For a graph $G = (V, E)$, a connected component c of G is a subgraph $c = (V', E')$, where $V' \subseteq V$ and $E' \subseteq E$. For any two vertices $u, v \in V'$, there exists a sequence of vertices (v_1, v_2, \dots, v_n) and a sequence of edges $(e_1, e_2, \dots, e_{(n-1)})$ such that: $v_1 = u$ and $v_n = v$ (i.e., the first vertex is u , and the last vertex is v). For each $i, 1 \leq i \leq n - 1, e_i$ is an edge in E' that connects v_i to $v_{(i+1)}$. Furthermore, we define C as the set of all connected components. Therefore $C = \{c_i\}$, with $i = 1, \dots, k_{comp}$, where k_{comp} is the number of connected components [23].

Communities. Communities in graphs refer to subsets of nodes within a larger network that exhibit higher intra-connectivity compared to interconnectivity. The detection and analysis of communities play a crucial role in understanding the structure and function of complex systems, including social networks, biological networks, and information networks. Various algorithms and methods have been developed to uncover communities in graphs, with the common objective of identifying densely connected subgraphs. A fundamental concept used in community detection is modularity, which measures the quality of a partition of nodes into communities. The modularity of a graph partition is defined as:

$$Q = \frac{1}{2|E|} \sum_{i,j} \left(A_{ij} - \frac{deg(i)deg(j)}{2|E|} \right) \delta(\mathfrak{C}_i, \mathfrak{C}_j) \quad (1)$$

where A_{ij} represents elements of the adjacency matrix, $deg(i)$ and $deg(j)$ are the degrees of node i and j , m is the total number of edges, \mathfrak{C}_i and \mathfrak{C}_j are the communities of nodes i and j , and $\delta(\mathfrak{C}_i, \mathfrak{C}_j)$ is the Kronecker delta function that equals 1 if $\mathfrak{C}_i = \mathfrak{C}_j$ and 0 otherwise.

The modularity optimization problem aims to find the partition that maximizes Q , indicating strong community structure. Beyond modularity, other methods like spectral clustering, hierarchical clustering, and random-walk-based approaches have been developed to uncover communities. The study of communities in graphs has provided valuable insights into the organization of networks and has practical applications in recommendation systems, information diffusion modelling, and network analysis [12].

2.2 Multi-objective Pathfinding

The multi-objective route planning problem, hereafter called the *pathfinding problem*, can be defined as a network flow problem [14, 15]. The goal is to find a set of optimal paths (routes) $P^* = \{p_1, \dots, p_L\}$ in a graph

$$G = \left(V, E, \phi, \vec{f}, \iota_V(\mathfrak{P}), \iota_E(\mathfrak{P}), n_s, n_e \right) \quad (2)$$

where V is the set of vertices or nodes, E represents the set of edges and ϕ represents a function mapping every edge to an ordered pair of nodes n and n' ; hence $\phi : E \rightarrow \{(n, n') \mid (n, n') \in V^2\}$. A path p_i is the sequence of nodes

from a starting node $n_S \in V$ to a predefined end node $n_{End} \in V$, i.e., $p_i = (n_i, n_{i+1} \dots, n_k)$, where $n_S = n_i$ and $n_{End} = n_k$ and $n_i \in V$ for $i = 1, 2, \dots, k$ and $\exists \phi(e_{i,i+1}) = (n_i, n_{i+1}) \in E$ for $i = 1, 2, \dots, k - 1$. Such a path p is called a path of length $k - 1$ from n_1 to n_k . A path p_i is here represented as a list of nodes in a graph. Another representation is a list of edges to traverse; hence $p_i = (e_1, \dots, e_{k-1})$ where $n_S = \phi(e_1)(1)$ and $n_{End} = \phi(e_k)(2)$ and $e_i \in E$ for $i = 1, 2, \dots, k$. Following the definition of a multi-objective optimisation problem, the decision variable \mathbf{x} is a path p in search space Ω [20].

3 Related Work

In this section, we present the related work about decision support systems (DSSs) that is used to decrease the number of solutions a DM has to choose from. Furthermore, we give a short overview on methodologies related to the concept of a Pareto graph, in which non-dominated solutions are put into relation using a graph structure.

3.1 Pareto Set Reduction as a DSS

In real-world applications, the Pareto set can be vast, making it challenging for decision-makers to analyse and select a preferred solution. To address this challenge, Pareto set reduction techniques have been developed as a decision support tool, aiming to reduce the size of the Pareto set while preserving its essential characteristics [9].

Pareto set reduction methods, utilized to provide decision-makers with a more manageable set of solutions, are divided into clustering-based and representative selection approaches. Clustering-based methods amalgamate similar solutions within the Pareto set into clusters, selecting a representative solution from each cluster, and have been further explored through various subsequent works focusing on the clustering of non-dominated solutions and the application of graph-based representations in Multi-Objective Optimization (MOO) [2, 20]. For instance, a graph-theoretical clustering approach has been proposed to identify a reduced set encapsulating extreme solutions of Pareto optimal solutions for MOO problems [8]. Another technique employs clustering in both the objective and decision spaces to find intersection sets, aiding a DM in electing the optimal solution [20]. Conversely, representative selection strategies try to directly select a subset of solutions embodying the diversity and distribution of the entire Pareto set [8]. Through these methodologies, both approaches facilitate simplified analysis and decision-making by rendering a condensed yet diverse set of solutions for evaluation.

3.2 Pareto Graphs

In multi-objective optimization (MOO), obtaining a well-distributed set of non-dominated solutions is a crucial goal. Paquete and Stützle extended the concept

of Pareto graphs [5] (also known as *efficient graphs*) to represent relationships among solutions in the objective space. Each node in the Pareto graph corresponds to a solution, and each directed edge represents whether one solution can be reached from another within a certain distance. They conducted an experimental analysis on the properties of the Pareto graph induced by the set of efficient solutions for multi-objective combinatorial optimization problems, observing that the Pareto graph contains clusters of non-dominated solutions which are tightly connected subsets of solutions [13]. Furthermore, Liefvooghe et al. proposed to use a graph in which edges represent the potential ability of a search algorithm to jump from one solution to another [10].

4 Finding Related Paths

In this section, we describe how pairs of paths can be identified that share common sub paths and how a respective graph from this information can be constructed. Furthermore, we propose to use community detection algorithms to find interesting subsets of paths. These communities can help a DM to make a more informed decision.

4.1 Constructing the Route-Change-Graph

To represent possible changes of routes, we can construct a Route-Change-Graph (RCG), that is a graph $G = (V, E)$ where each $v \in V$ represents a single path from the designated start to the goal node and each $e \in E$ represents a change opportunity between two routes (two nodes).

Such a graph is constructed by analysing a set of possible routes and identifying their pairwise common contiguous nodes (excluding start and end). For each pair of routes (r_i, r_j) , where a route $r = (n_s, \dots, n_e)$, we construct the intersection of their subsets of contiguous nodes, excluding n_s and n_e . Let r_i and r_j be the ordered sets of their respective points. Therefore, $I_{ij} = r_i \setminus \{n_s, n_e\} \cap r_j \setminus \{n_s, n_e\}$ is the intersection of the two sets without the start and end nodes. We create a such an intersection for each route pair. Each set I_{ij} contains nodes and, therefore, subroutes, that are present and shared in two routes. However, instead of obtaining the cardinality of the intersection set I_{ij} , i.e. $|I_{ij}|$, we save the number of common contiguous subroutes $|\mathfrak{S}_{ij}|$ (between r_i and r_j , in a matrix M , where each column and each row represents a route. Therefore, the $n \times n$ matrix M is symmetric.

To obtain $|\mathfrak{S}_{ij}|$, we consider two ordered sets r_i and r_j , where each r_i consists of a sequence (n_1, \dots, n_k) with k being variable. The task is to find the number of common contiguous subsequences between r_i and r_j . A contiguous subsequence of r_i is any sequence $(n_{i_a}, \dots, n_{i_b})$ where $1 \leq a < b \leq k$, and the indices a and b form a contiguous range.

Let's denote by $S(r_i)$ the set of all contiguous subsequences of r_i , i.e., $S(r_i) = \{s | s \text{ is a contiguous subsequence of } r_i\}$. We are interested in finding the cardinality of the intersection of $S(r_i)$ and $S(r_j)$, denoted $|\mathfrak{S}_{ij}| = |S(r_i) \cap S(r_j)|$.

In Algorithm 1, we show pseudocode to compute the common contiguous subsequences.

Algorithm 1. Common Contiguous Subsequences of Ordered Sets r_1 and r_2

```

1: function COMMONCONTIGUOUSSUBSEQUENCES( $r_1, r_2$ )
2:    $S(r_1) \leftarrow \text{GetContiguousSubsequences}(r_1)$ 
3:    $S(r_2) \leftarrow \text{GetContiguousSubsequences}(r_2)$ 
4:    $common\_count \leftarrow 0$ 
5:   for each  $s_1$  in  $S(r_1)$  do
6:     for each  $s_2$  in  $S(r_2)$  do
7:       if  $s_1 = s_2$  then
8:          $common\_count \leftarrow common\_count + 1$ 
9:       end if
10:    end for
11:  end for
12:  return  $common\_count$ 
13: end function

14: function GETCONTIGUOUSSUBSEQUENCES( $r$ )
15:   $subsequences \leftarrow \emptyset$ 
16:  for  $i = 1$  to  $\text{length}(r)$  do
17:    for  $j = i + 1$  to  $\text{length}(r) + 1$  do
18:       $subsequences \leftarrow subsequences \cup \{(r_i, \dots, r_{j-1})\}$ 
19:    end for
20:  end for
21:  return  $subsequences$ 
22: end function

```

Each element in M contains then the number of possible change opportunities between two routes. In the following, we only consider one half of the matrix, as it is symmetric. The intersection of a route to itself is the route itself and is not considered in the following analysis (the respective matrix cells are set to 0). The matrix M looks as follows.

$$M = \begin{array}{c|ccccc} & r_1 & r_2 & \cdots & r_{n-1} & r_n \\ \hline r_1 & 0 & |\mathfrak{S}_{12}| & |\mathfrak{S}_{13}| & |\mathfrak{S}_{14}| & |\mathfrak{S}_{15}| \\ r_2 & - & 0 & |\mathfrak{S}_{23}| & |\mathfrak{S}_{24}| & |\mathfrak{S}_{25}| \\ \vdots & \vdots & - & 0 & |\mathfrak{S}_{34}| & |\mathfrak{S}_{35}| \\ r_{n-1} & \vdots & \vdots & - & 0 & |\mathfrak{S}_{45}| \\ r_n & - & \cdots & \cdots & - & 0 \end{array}$$

With the obtained route change matrix M , we can now construct the RCG, i.e. $G_{RCG} = (V_{RCG}, E_{RCG})$. We assume a bidirectional possibility to change between two routes. Each route r is represented by a node $v_{r_i} \in V_{RCG}$. Each element in the matrix M represents an edge in E_{RCG} between two routes (column

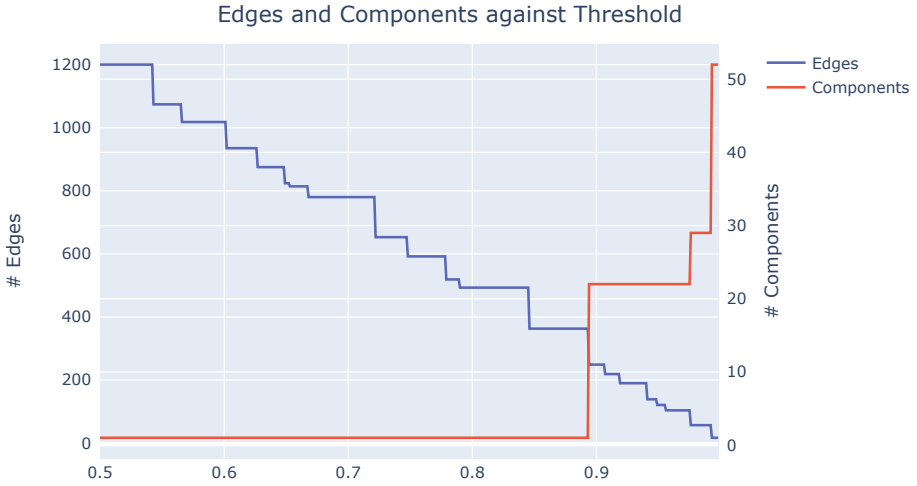


Fig. 1. $|E|$ and $|C|$ in relation to τ for $G_{RCG,\tau}$

and row), and the value represents the edge’s weight. As the matrix has as many rows and columns as there are routes, the resulting graph can be substantially large. Therefore, we propose to use a threshold value τ for the edges’ weights that are constructed in the graph. The threshold value τ is determined using quantiles on the matrix’ values. An edge is constructed if the respective value is over the specified threshold. However, constructing fewer edges can result in a disconnected graph and, therefore, in having multiple connected components. Nevertheless, our proposed RCG should have the least possible number of connected components while also having the least possible number of edges, keeping it less dense. The graph can be constructed for various thresholds, and the value that maintains both properties low can then be identified. In Fig. 1, such an analysis is shown. With an increasing threshold, the number of edges decrease while there are more connected components. In the given example, we can decide on a threshold quantile of 0.891, which results in one connected component and 363 edges in E_{RCG} . However, for a different set of routes, it can happen that the possibility of having only one connected component is not given. Then, a τ should be chosen, that minimizes $|C|$.

4.2 Community Detection and Analysis

After constructing the RCG, which nodes represent paths, which edges represent change possibilities and which edges’ weights represent how often a route can be changed, we can apply community detection algorithms to identify closely related solutions. Furthermore, we propose to use various metrics of these communities to identify subsets of solutions that are presented to a DM. In addition, we propose three strategies a DM can utilise to identify a feasible and fitting community.

Community Detection. We propose to use the Leiden algorithms, which is an extension of the Louvain algorithm, to ensure well-connected communities [18].

The Leiden algorithm is a highly efficient algorithm for community detection in networks. It is an improvement over the Louvain method, which is known for its high performance but has certain limitations. The Leiden algorithm addresses these limitations by incorporating a refinement phase to improve the quality of partitions.

Mathematically, the Leiden algorithm optimizes the modularity function, shown in Equation (1).

We furthermore propose to set the settings of the algorithm to find as many different communities that can be comprehended by a DM, i.e., according to [16], 3 to 4. Furthermore, we propose to choose the graph's modularity as the quality function of the Leiden algorithm [12]. We set the number of iterations to 2, since this is the default setting of the implementation we use to compute the partition [19].

Community Analysis. After finding a good number of communities, we propose to compute various metrics on these. These metrics should reflect how intra-connected the communities are, but also if they are interconnected to other communities. We have decided to compute four metrics for each community:

1. **Density** [3]. The density of a graph structure, denoted as $\rho(G)$, is a measure that provides insight into how many edges are present in the graph relative to the maximum possible number of edges. For an undirected simple graph with $|V|$ vertices, the maximum number of edges is $\frac{|V|(|V|-1)}{2}$. Thus, the density is defined as:

$$\rho(G) = \frac{2|E|}{|V|(|V|-1)} \quad (3)$$

where $|E|$ represents the number of edges in the graph. For directed graphs, the maximum number of edges is $|V|(|V|-1)$, and the density is calculated as:

$$\rho(G) = \frac{|E|}{|V|(|V|-1)} \quad (4)$$

Consequently, a graph's density ranges from 0 (for an empty graph) to 1 (for a complete graph).

2. **Average Cluster coefficient** [17]. The average clustering coefficient, $\langle \mathcal{C} \rangle$, quantifies the degree of clustering in a network. It's calculated as:

$$\langle \mathcal{C} \rangle = \frac{1}{|V|} \sum_{v_i \in G} \mathcal{C}(v_i) \quad (5)$$

Where:

– $|V|$ is the total number of nodes

- v_i represents each node in the graph G
- $\mathcal{C}(v_i)$ is the clustering coefficient of node v_i

For a given node v_i , $\mathcal{C}(v_i)$ is the node’s clustering coefficient, i.e., a proportion of existing links between its neighbours over the total possible links.

Given a node v with k_v neighbours, the cluster coefficient $\mathcal{C}(v)$ for that node can be calculated using the following equation:

$$\mathcal{C}(v) = \frac{2\mathcal{T}_v}{k_v(k_v - 1)}$$

where \mathcal{T}_v represents the number of edges between the neighbours of v . This equation calculates the ratio between the number of actual edges \mathcal{T}_v and the maximum possible number of edges between k_v nodes. In other words, it is the ratio of actual triangles that node involved in and the number of possible triangles.

If a node has less than two neighbours, its clustering coefficient is 0. This measure provides an overall sense of the network’s *cliquishness*.

3. **Group Betweenness Centrality** [6]. Everett and Borgatti proposed the concept of Group Betweenness Centrality as a measure to identify the most central group within a network. It extends the idea of individual node centrality to encompass groups of nodes. The Group Betweenness Centrality of a group of nodes is defined as the sum of the fraction of shortest paths between all pairs of nodes in the network that pass through at least one node in the group. This measure reflects the extent to which a group collectively acts as a bridge or gatekeeper between other nodes in the network. Given a group of nodes, \mathcal{V} , the betweenness centrality of this group, denoted as $bc(\mathcal{V})$, is given by:

$$bc(\mathcal{V}) = \sum_{n_{\text{from}} \neq v \neq n_{\text{to}}} \left(\frac{\sigma(n_{\text{from}}, n_{\text{to}} | \mathcal{V})}{\sigma(n_{\text{from}}, n_{\text{to}})} \right) \tag{6}$$

Where:

- $\sigma(n_{\text{from}}, n_{\text{to}})$ is the total number of shortest paths from node n_{from} to node n_{to}
- $\sigma(n_{\text{from}}, n_{\text{to}} | \mathcal{V})$ is the number of those paths that pass through some node in group \mathcal{V}

Notice that $n_{\text{from}} \neq v \neq n_{\text{to}}$ means that we take all pairs of nodes except those pairs where either node is in the group \mathcal{V} . In contrast to the other three metrics, we compute the Group Betweenness Centrality for a community in the scope of the whole graph, while the other metrics are calculated using solely the nodes and edges of the respective community.

4. **Graph Degree Centrality** [7]. The degree centrality of a graph is a measure of the overall connectivity of the graph. It is an average of the degree centralities of all nodes in the graph.

It is defined as:

$$dc(G) = \frac{\sum_{i=1}^{|V|} [dc(v^*) - dc(v_i)]}{|V|^2 - 3|V| + 2} \quad (7)$$

Where:

- $dc(G)$ represents the degree centrality of the graph G
- $dc(v^*)$ and $dc(v_i)$ denote the degree centralities of the node with the highest degree (v^*) and each other node (v_i) respectively
- $|V|$ is the total number of nodes in the network

This formula calculates the sum of differences between the degree centrality of the node with the highest degree and that of every other node. This sum is then normalized by dividing it by $|V|^2 - 3|V| + 2$, which is derived from the maximum possible sum of differences.

In this case, a higher degree centrality indicates that one node (the one with the highest degree) is significantly more connected than others, while a lower degree centrality suggests a more evenly distributed network where no single node dominates in terms of connections.

Community Selection. After computing the four different metrics for each community, we can use one or more of these measurements to select a community that is being presented to a DM. With our approach, we shift the DM's task from the objective space (and where possible interesting areas are) to a space where they have to decide on specific properties of subsets of solutions. Especially for problems similar to the multi-objective pathfinding problem, that can be highly uncertain from a temporal perspective, it can be beneficial to choose a subset of solutions rather than a single solution to have alternatives ready when the solution is executed but not feasible any more. For instance, when traversing a path, a DM may get the information that a chosen segment on a later stage of the path is not traversable any more. With a pre-computed set of alternative solutions, the DM can still choose from various non-dominated solutions. As follows, we present three strategies to use the proposed community metrics. We propose to apply non-dominated sorting of the set of communities using their respective metric values, and then to use a combination of the four metrics.

Always alternatives. If a DM aims for solutions that have always alternatives when being traversed, we propose to choose a community with a high density and low graph degree centrality. An example of such a community is presented in *Community 1* in Fig. 2.

Main route, but possible dead ends in alternatives. A star shape community represents a set of alternatives with one main route and adjacent solutions. Choosing such a set may result from a high priority on a specific route. However, depending on the number of rays of the star, i.e., the alternatives, a different route might be available with the sacrifice of having no more alternatives afterwards. Nevertheless, a return to the main route can be possible. An example of such a community is presented in *Community 2* in Fig. 2.

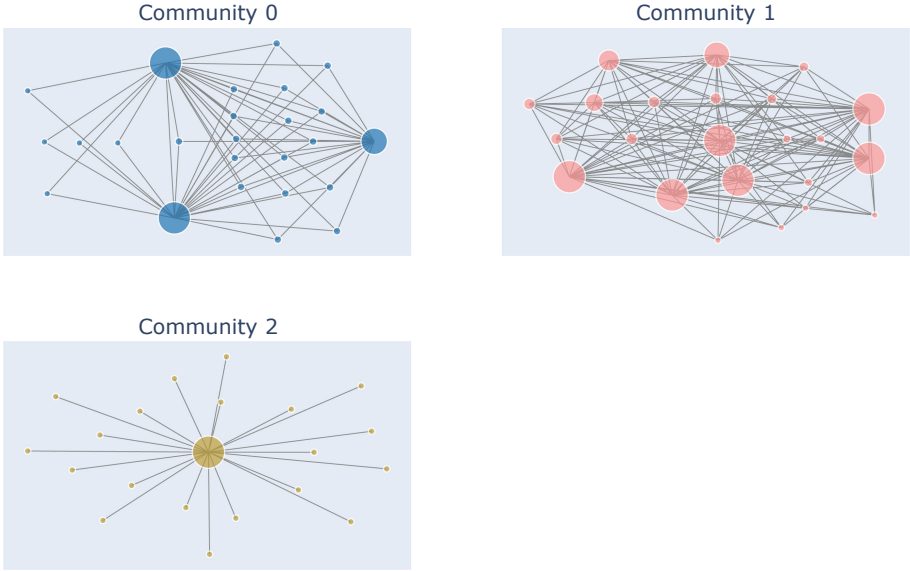


Fig. 2. The three obtained communities. Force-directed layout for visualisation.

Few central solutions, few alternatives. A community can exist with multiple central solutions, where each solution has a substantially high number of alternatives, and that community has also a few additional solutions with a lower number of available alternatives. An example of such a community is presented in *Community 0* in Fig. 2.

5 Evaluation and Discussion

In this section, we apply the proposed methodology to an instance of the pathfinding problem that has been proposed in [20] and which has been published in [22].

The instance of the problem represented the task of finding the set of Pareto optimal routes within the European road network from Warsaw to Madrid. The final network consisted of 1.14×10^8 nodes and 1.46×10^8 edges and a variation of the NSGA-II algorithm [4, 20] was applied to optimise (minimization) four objectives, i.e., length of the route, time to traverse it, positive ascent and the curvature. For a detailed description, the interested reader is referred to [20]. The authors have obtained 69 different and non-dominated routes that are shown in Fig. 3. Although the routes are very similar from a visual perspective, there are small differences in various locations.

We can now construct all intersection sets \mathfrak{S}_i , using our proposed methodology, and build the matrix M from it. The result is a 69×69 matrix, which elements represent possible changes between routes and the value related to the number of possible changes. From this adjacency matrix, we construct the



Fig. 3. All obtained Pareto-optimal routes for four objectives [20]

respective RCG, shown in Fig. 4. The graphical representation was created using a force directed layout [1].

As described in Sect. 4.1, we use the 0.891-quantile as a threshold so that our graph has exactly one connected component. In Fig. 4, we have also coloured the communities, that have been found when applying the Leiden algorithm. In Fig. 2, we show each community separately, also arranged using a force-directed layout. From a visual approach, the structural differences of the communities are already visible.

To compare the communities, we can now compute the proposed metrics, i.e., density, average cluster coefficient, group betweenness centrality and graph degree centrality. In Fig. 5, we show these metrics for each community. It should be noted, that, in terms of these metrics, all communities are non-dominated. We propose to only use non-dominated communities. From a visual perspective, community 2 is structurally different compared to the other two. It has a rather high graph degree centrality and group betweenness centrality, but also a low density and an average cluster coefficient of 0, as the community does not contain any triangles.

We assume that a DM should decide on one specific community. However, also the linking between communities can be of interest. The DM can utilise the Group Betweenness Centrality to estimate how well a change between commu-

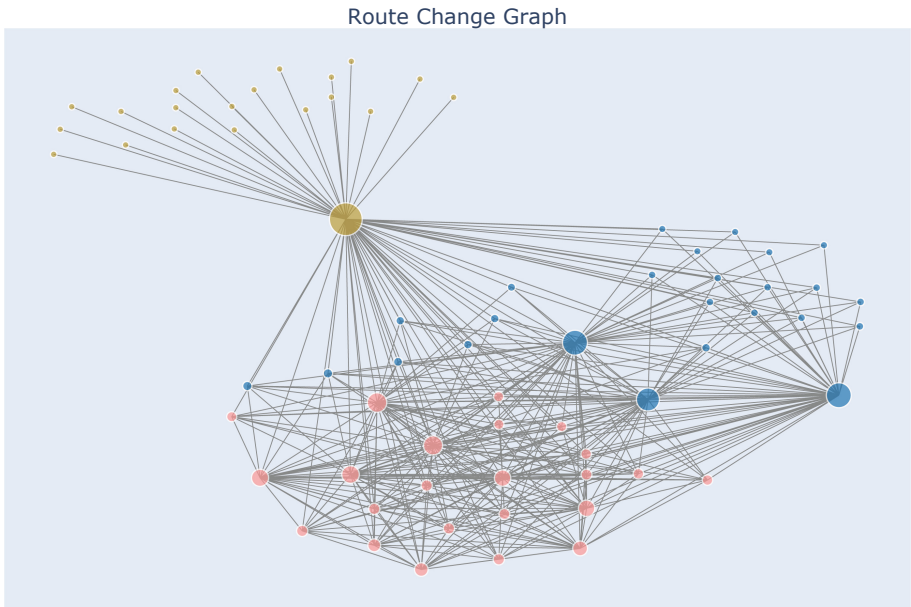


Fig. 4. The obtained RCG from the real-world example. Layout obtained by applying a force directed algorithm

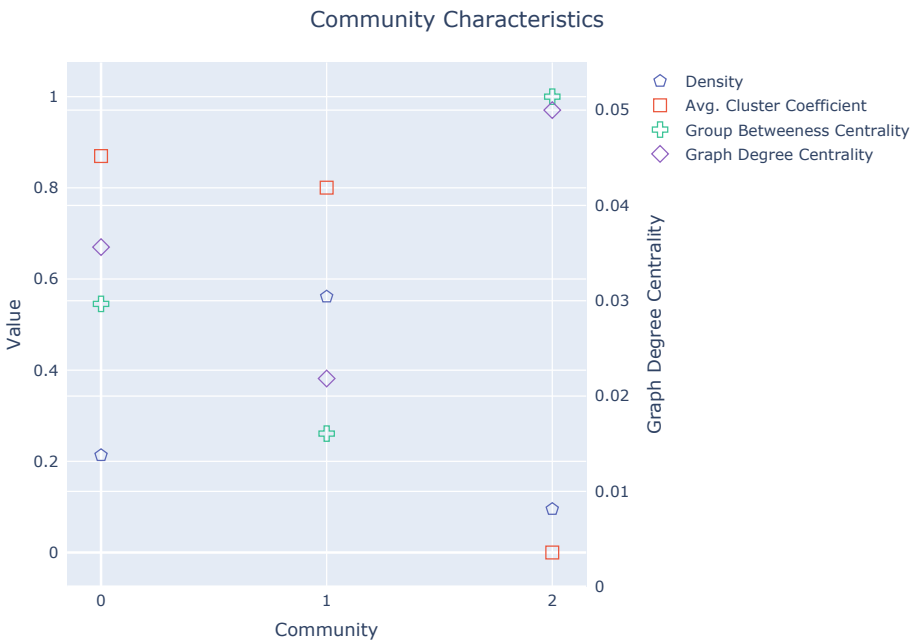


Fig. 5. Various characteristics of each community. The right y-axis shows Graph Degree Centrality, the left axis shows the other metrics.

nities can be done. In other words, a community with a high group betweenness centrality enables to easily change to other communities.

To provide an easier access to our proposed methodologies, we have published the code that we used. In addition, we provide an easy-to-use UI that uses the artificial and real-world data [21].

6 Conclusion and Outlook

In this paper, we have proposed a novel DSS that can identify a comprehensible number of subsets of solutions for decision-makers to choose from. The approach is especially suitable for problems where there are possibilities to switch between solutions, as they are temporal uncertain and alternatives are available. With our approach, an Route-Change-Graph (RCG) is generated using a problem specific threshold to keep the number of edges low, then communities are identified and finally, the communities are analysed using various graph metrics to help a DM choose the most fitting subset of solutions. In addition, we have evaluated the methodology on a real-world problem. However, an empirical analysis with actual DMs is missing and should be carried out in the future.

Furthermore, in the future, we want to test the approach on different problems than route planning on maps, e.g., network routing or also medical applications. Moreover, other graph related metrics than the four that we have utilised, should be evaluated in the future. We see our proposed methodology as a starting point to more problem-centric DSS instead of general applicable approaches.

References

1. Bastian, M., Heymann, S., Jacomy, M.: Gephi: an open source software for exploring and manipulating networks. In: Proceedings of the International AAAI Conference on Web and Social Media, vol. 3, issue 1, pp. 361–362 (2009)
2. Bejarano, L.A., Espitia, H.E., Montenegro, C.E.: Clustering analysis for the pareto optimal front in multi-objective optimization. *Computation* **10**(3), 37 (2022). Publisher: Multidisciplinary Digital Publishing Institute
3. Bollobás, B.: *Graph Theory*, vol. 63. Springer, New York, New York, NY (1979), series Title: Graduate Texts in Mathematics. <https://doi.org/10.1007/978-1-4612-9967-7>
4. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002)
5. Ehrgott, M., Klamroth, K.: Connectedness of efficient solutions in multiple criteria combinatorial optimization. *Eur. J. Oper. Res.* **97**(1), 159–166 (1997)
6. Everett, M.G., Borgatti, S.P.: The centrality of groups and classes. *J. Math. Sociol.* **23**(3), 181–201 (1999)
7. Freeman, L.C.: Centrality in social networks conceptual clarification. *Soc. Netw.* **1**(3), 215–239 (1978)
8. Kahagalage, S., Turan, H.H., Jalalvand, F., El Sawah, S.: A novel graph-theoretical clustering approach to find a reduced set with extreme solutions of Pareto optimal solutions for multi-objective optimization problems. *J. Global Optim.* **86**(2), 467–494 (2023)

9. Korhonen, P., Wallenius, J.: A pareto race. *Naval Res. Logist. (NRL)* **35**(6), 615–623 (1988)
10. Liefoghe, A., Derbel, B., Verel, S., López-Ibáñez, M., Aguirre, H., Tanaka, K.: On pareto local optimal solutions networks. In: Auger, A., Fonseca, C.M., Lourenço, N., Machado, P., Paquete, L., Whitley, D. (eds.) *Parallel Problem Solving from Nature - PPSN XV*. LNCS, pp. 232–244. Springer International Publishing, Cham (2018)
11. Miller, G.A.: The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychol. Rev.* **63**(2), 81–97 (1956)
12. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. *Phys. Rev. E* **69**(2), 026113 (2004), publisher: American Physical Society
13. Paquete, L., Stützle, T.: Clusters of non-dominated solutions in multiobjective combinatorial optimization: an experimental analysis. In: Barichard, V., Ehrgott, M., Gandibleux, X., T'Kindt, V. (eds.) *Multiobjective Programming and Goal Programming*. LNEMS, pp. 69–77. Springer, Berlin, Heidelberg (2009)
14. Pulido, F.J.J., Mandow, L., Pérez-De-La-Cruz, J.L.L.: Dimensionality reduction in multi objective shortest path search. *Comput. Oper. Res.* **64**, 60–70 (2015)
15. Raith, A., Ehrgott, M.: A comparison of solution strategies for biobjective shortest path problems. *Comput. Oper. Res.* **36**(4), 1299–1331 (2009)
16. Rouder, J.N., Morey, R.D., Cowan, N., Zwilling, C.E., Morey, C.C., Pratte, M.S.: An assessment of fixed-capacity models of visual working memory. *Proc. Natl. Acad. Sci.* **105**(16), 5975–5979 (2008)
17. Schank, T., Wagner, D.: Approximating clustering coefficient and transitivity. *J. Graph Algorithms Appl.* **9**(2), 265–275 (2005)
18. Traag, V.A., Waltman, L., van Eck, N.J.: From Louvain to Leiden: guaranteeing well-connected communities. *Sci. Rep.* **9**(1), 5233 (2019). Publisher: Nature Publishing Group
19. Traag, V., et al.: *vtraag/leidenalg: 0.10.0* (2023)
20. Weise, J.: Evolutionary many-objective optimisation for pathfinding problems. Doctoral Thesis, Otto von Guericke University Magdeburg, Magdeburg (2023). Accepted: 2023-03-14T10:25:49Z. ISBN: 9781839107955
21. Weise, J.: Pareto graph analysis (2023). <https://doi.org/10.5281/zenodo.10044244>
22. Weise, J., Mostaghim, S.: Dataset (Pareto fronts and sets) for the Multi-objective pathfinding problem (2023). <https://doi.org/10.5281/ZENODO.10008219>
23. Wilson, R.J.: *Introduction to Graph Theory*. Longman (2010)