Eva Röper

Innovization for Multi-Objective Time-Dependent Route Planning

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG
FACULTY OF COMPUTER SCIENCE

Institute for Intelligent Cooperating Systems
Chair of Computational Intelligence

Innovization for Multi-Objective Time-Dependent Route Planning

Master Thesis

Eva Röper

January 18, 2024

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

Advisor: Dr.-Ing. Jens Weise

Eva Röper: *Innovization for Multi-Objective Time-Dependent Route Planning*
Otto von Guericke University Magdeburg
Institute for Intelligent Cooperating Systems
Chair of Computational Intelligence
Magdeburg, 2024.

Acknowledgements

I would like to sincerely thank Prof. Sanaz Mostaghim for providing opportunity and means for writing this master thesis. Moreover, I am grateful for the entire Chair of Computational Intelligence and all its members for their support in form of a productive working environment and helpful suggestions. Special thanks go to my supervisor Jens, who not only regularly offered valuable feedback but also moral support. Last but not least, I am deeply thankful for my partner's unwavering support through all ups and downs.

Speed data retrieved from Uber Movement, (c) 2023 Uber Technologies, Inc., `https://movement.uber.com`.

Abstract

Multi-objective route planning is a prominent but computationally expensive optimisation problem in everyday life. Reusing knowledge from similar route planning problems could enhance performance and sustainability of routing algorithms. The innovization methodology [18] attempts to extract knowledge from Pareto-optimal solutions of optimisation problems for this purpose. However, applying this methodology to route planning problems leads to some challenges. Therefore, we propose an innovization for route planning which is an adapted version of the original innovization. To this end, we design a multi-objective evolutionary algorithm for routing. Moreover, we introduce a novel local search method for routing problems called Perimeter Mutation Local Search. Lastly, we integrate a detailed analysis step with decision space clustering and correlations between objectives and route characteristics. We evaluate our proposed approach on multi-objective time-dependent routing problems to see what knowledge can be gained and whether this knowledge can improve a multi-objective evolutionary algorithm. Our results show that we are able to extract knowledge using the introduced innovization for route planning. Furthermore, this knowledge is used to improve a multi-objective evolutionary algorithm by reducing computational effort. With about a third of previously necessary function evaluations, we manage to produce similar optimisation results. This is particularly beneficial for mobile applications where limited computational resources are available.

"A straight line may be the shortest distance between two points, but it is by no means the most interesting."

– *The Third Doctor, The Time Warrior*

Contents

List of Acronyms

CSV Comma-separated values

EA Evolutionary algorithm

GPS Global Positioning System

ID Identifier

IGD Inverted Generational Distance

ITS Intelligent Transport System

MOEA Multi-objective evolutionary algorithm

NBX Node Based Crossover

NCM Normal constraint method

NSGA-II Non-dominated Sorting Genetic Algorithm II

OSM OpenStreetMap

OPTICS Ordering Points To Identify the Clustering Structure

PMM-LS Perimeter Mutation Local Search

RQ Research question

SSD Semi-standard deviation

VNS Variable Neighbourhood Search

VRP Vehicle routing problem

1. Introduction

Route planning is a prominent optimisation problem of everyday life, affecting many people and various industries. In the past, front-seat passengers were often the ones to provide direction. Nowadays, this task is mostly performed by routing algorithms of Intelligent Transport Systems (ITSs). Nonetheless, optimising multiple objectives for routes remains a challenging and computationally expensive problem. Ideally, we could reuse knowledge from similar route planning problems to enhance the performance of routing algorithms. The innovization methodology [18] gives instructions for extracting knowledge from optimisation problems for this purpose. The identified, reusable principles could save computation time of routing algorithms which would make the use of multi-objective algorithms for routing more viable in everyday life. This could result in an improvement of quality of life for drivers since they would have additional decision-making support. Moreover, drivers would be able to personalise routes to better suit their needs. In addition, reusing knowledge to accelerate routing algorithms increases the sustainability of algorithms and therefore decreases energy consumption.
However, innovization cannot simply be applied to routing problems because this kind of problems has complex decision variables and constraints, and often non-differentiable objective functions. Coello Coello et al. [12] employed innovization for a related problem, a bi-objective travelling salesman problem. However, their extracted knowledge was too problem-specific. To the best of our knowledge, there are no previous works attempting to use innovization on path or route planning problems. Therefore, we propose an innovization for route planning which is an adapted version of the original innovization. We construct a multi-objective evolutionary algorithm (MOEA) for routing and develop a novel local search method for routing problems called Perimeter Mutation Local Search (PMM-LS). PMM-LS defines a route's neighbourhood and systematically searches for improvements within neighbourhoods. Furthermore, we cluster in decision spaces instead of objective spaces and include an analysis of correlations between objectives and route characteristics.

The main goal of this thesis and its primary contribution is the development of an innovization for route planning. In addition, we evaluate what knowledge can be gained using our new methodology and whether an improvement of a routing evolutionary algorithm can be achieved. For our experiments, we define multi-objective time-dependent routing problems which optimise travel time, travel time variability and ease of driving. Travel times change throughout the day and the week. They are based on an Uber Movement Speeds dataset [62] which contains average hourly speeds recorded for Berlin in January of 2020.
The rest of this thesis is structured as follows. First, we explain all necessary concepts in chapter 2. In chapter 3, we define our research questions and corresponding metrics and give the problem statement. We continue by relating this thesis to other work in chapter 4 before presenting our innovization for route planning methodology in chapter 5. The evaluation of the knowledge extraction and algorithm improvement using our proposed approach follows in chapter 6. Finally, chapter 7 concludes our work and presents ideas for future work.

2. Background

This thesis introduces a methodology for applying innovization to multi-objective path and route planning problems. The concepts of multi-objective path planning problems, multi-objective optimisation and innovization needed for this methodology are explained in this chapter.

2.1. Multi-objective path planning problems

Multi-objective optimisation problems like the route planning problem in this thesis are optimisation problems that minimise or maximise multiple, conflicting objective functions simultaneously. Formally, it is defined as follows [14]:

$$\begin{aligned} \min / \max \quad & f_m(x) && m = 1, 2, \ldots, M \\ \text{s.t.} \quad & g_j(x) \geq 0 && j = 1, 2, \ldots, J \\ & h_k(x) = 0 && k = 1, 2, \ldots, K \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)} && i = 1, 2, \ldots, n \end{aligned} \tag{2.1}$$

Equation 2.1 allows us to define constraints for a multi-objective problem. They can be inequality constraints g_j, equality constraints h_k, lower variable bounds $x_i^{(L)}$ or upper variable bounds $x_i^{(U)}$. An example for bounds of a route planning problem are a restriction of values to node indices in a road network. The variable bounds define a *decision space* X. In the example of route planning, X consist of all possible routes within a defined road network. The set of solutions within the decision space that also satisfy all other constraints are called a *feasible region* S. The space that is bound by the objective functions is named *objective space* Z. The relation of decision and objective space is further illustrated in Figure 2.1.

A solution x of a multi-objective problem is a vector of values for each of the n decision variables. The objective functions are conflicting. This means we cannot simultaneously improve all functions. Therefore, the result of a

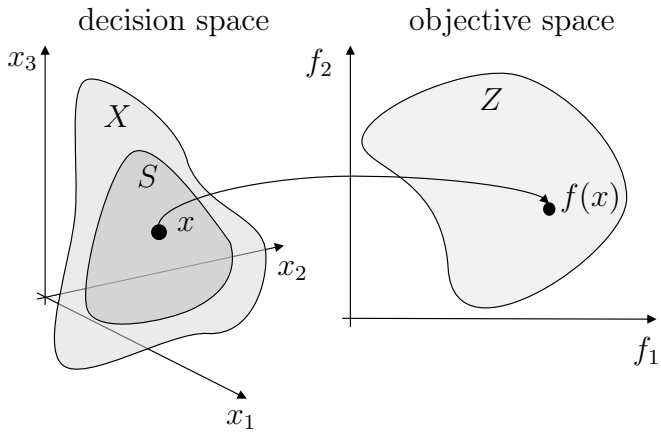


Figure 2.1.: Relation of decision space X with feasible region S to objective space Z. Figure adapted from [14]

multi-objective optimisation is not one single optimal solution but a set of *Pareto-optimal* solutions. This set of solutions, the *Pareto front*, contains trade-off solutions where one is not better than the other. For each Pareto-optimal solution, we cannot improve one objective without worsening another. Formally, the globally Pareto-optimal set is the *non-dominated* set of the entire feasible search space S. A solution is called non-dominated if it is not *dominated* by any other solution. For minimisation problems, a solution x^A dominates another solution x^B if both condition 2.2 and 2.3 hold [14].

$$\forall\, m \in \{1, 2, \ldots, M\} \quad f_m(x^A) \leq f_m(x^B) \tag{2.2}$$

$$\exists\, m \in \{1, 2, \ldots, M\} \quad f_m(x^A) < f_m(x^B) \tag{2.3}$$

Solutions are non-dominated if there exists no other solution which is at least equally as good for all objectives and strictly better for at least one objective. Using a resulting Pareto front from a multi-objective optimisation, we can determine the *nadir point* z^{nad}. This reference point is often estimated as the vector of worst objective values found in a Pareto front per objective [14]. The *ideal point* z^* is another reference point. It is defined independently of any Pareto front as the vector of optimal solutions from optimising each objective individually.

This thesis deals with a particular family of optimisation problems, namely multi-objective *route planning problems*. A specific route planning problem can be identified by a 6-tuple $\big(\{f_1, \ldots, f_M\}, G, n_O, n_D, w_0, t_0\big)$. The goal is to find a set of optimal paths in a road network. Road networks are often represented as graphs $G = (V, E)$ with a set of vertices, or nodes, V and a set of edges E

linking vertices. A path $(n_0, \ldots, n_l)$ is a sequence of nodes that connects a given start point n_O and a destination n_D. A path's optimality is evaluated according to multiple objective functions f_m. Moreover, our route planning problems are time-dependent. This means that the travel times in our road network are determined by a departure time t_0 during the week ($w_0 = 0$) or on weekends ($w_0 = 1$) [19]. A 6-tuple $\big(\{f_1, \ldots, f_M\}, G, n_O, n_D, w_0, t_0\big)$ represents a routing problem of the following form:

$$
\begin{aligned}
\min \quad & (f_1, \ldots, f_M) \\
\text{s.t.} \quad & x \in \big\{ (n_0, \ldots, n_l) \mid l \in \mathbb{Z}_{\geq 1}; \\
& \qquad n_0, \ldots, n_l \in V; \\
& \qquad \forall\, i \in \{0, \ldots, l-1\} \;\; (n_i, n_{i+1}) \in E; \\
& \qquad n_0 = n_O; n_l = n_D \big\} \\
& w_0 \in \{0, 1\} \\
& t_0 \in \{hh : mm : ss \mid hh \in [0, 23]; mm, ss \in [0, 59]\}
\end{aligned}
\tag{2.4}
$$

Notably, route planning problems have some similarities with vehicle routing problems (VRPs). A VRP is a logistical problem where tours for delivering products from depot to customer have to be found [38]. However, start and destination in a route planning problem are distinct and we do not have to visit certain locations on our way to the destination. Moreover, VRPs usually deal with a fleet of vehicles except for single VRP variants which only consider one vehicle like our route planning problems [38].

2.2. Multi-objective optimisation

Solving multi-objective problems a posteriori, meaning without transforming the problem into a single-objective one, can be time-consuming. Nonetheless, it has the advantage that users do not need to specify preferences before the optimisation and can instead choose a solution from multiple alternatives. Especially exact methods are computationally expensive [51] and are not efficient enough for applications such as multi-objective route planning where users expect good results within seconds. This is why we us a metaheuristic to solve our route planning problem. While metaheuristics cannot guarantee finding the global optimal solution, they can generate good-quality solutions in a reasonable amount of time [51]. There are many different metaheuristic algorithms such as

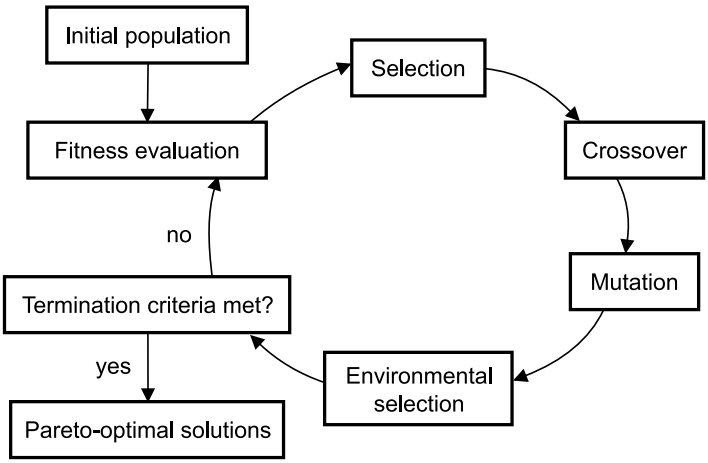


Figure 2.2.: General procedure of an EA

Tabu Search [30] or Simulated Annealing [60]. Evolutionary algorithms (EAs) are the only metaheuristic that can find multiple solutions with one simulation run [14]. For this reason, we employed an EA in our innovization for route planning approach.

EAs are inspired by nature. Their general procedure is depicted in Figure 2.2. EAs work iteratively on a set of individuals, a population. Each individual has a chromosome, which encodes relevant information. Moreover, individuals are assigned a quality measure based on objectives, called fitness. In each iteration, also called generation, offspring is created by applying crossover and mutation to chosen individuals until some termination criterion is reached. After the creation of an initial population, only selected, good individuals are allowed to reproduce and survive as a next generation [14].

Since it cannot be said that one metaheuristic approach is clearly better than another [37], we choose one of the most popular multi-objective EAs for solving our route planning problem. We use Non-dominated Sorting Genetic Algorithm II (NSGA-II) [17] and a variant of it with controlled elitism [16]. In principle, both algorithms follow the procedure of an EA as described above. However, for the selection of surviving individuals *fast non-dominated sorting*, crowding distance sorting and, for NSGA-II with controlled elitism, the geometric distribution from Equation 2.5 is used. The environmental selection of both algorithms is visualised in Figure 2.3. In the standard, elitist NSGA-II, only the best individuals according to the first two sorting methods survive. That means only the individuals from the best fronts in the least crowded areas are kept. This results in a loss of lateral diversity across different fronts which in

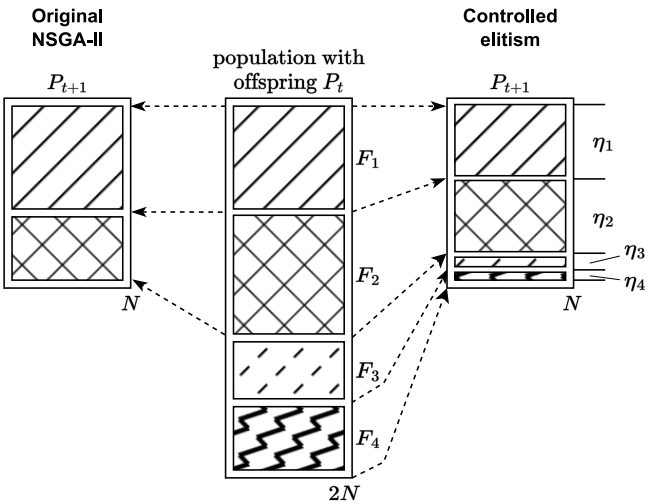


Figure 2.3.: Environmental selection of NSGA-II and NSGA-II with controlled elitism. NSGA-II keeps only the best individuals according to fast non-dominated sorting and crowding distance. With controlled elitism, a geometric distribution is additionally used to select individuals from all fronts. Figure adapted from [16]

turn leads to the algorithm getting stuck in a local Pareto front more easily [16]. NSGA-II with controlled elitism mitigates the loss of lateral diversity by keeping individuals from all fronts. Thereby, this variant of NSGA-II achieves a good diversity as well as a good convergence for the Pareto front it produces. For NSGA-II with controlled elitism, surviving individuals are selected according to three criteria. First, fast non-dominated sorting iteratively finds fronts of non-dominated individuals. From each front i, only η_i individuals are added to the next generation as per the following geometric distribution:

$$\eta_i = N \frac{1-r}{1-r^{|F|}} r^{i-1} \tag{2.5}$$

N refers to the number of individuals that are allowed to survive until the next generation. $|F|$ is the total number of fronts found by fast non-dominated sorting. The reduction rate $r < 1$ determines how much less individuals are kept from the best front. Nevertheless, most of the surviving individuals are from the best front. For $r = 0$, all individuals from the first front survive like in elitist NSGA-II. This would result in little exploration. From each following front increasingly less individuals survive. If not all individuals of a front can be kept, the survivors are chosen in descending order of *crowding distance* as it is done in elitist NSGA-II [16]. The *crowding distance* d_x of an individual x is

computed as the sum of absolute normalised differences in objective function values of its two neighbouring solutions x_{+1}^m and x_{-1}^m per objective m in the same front [17]:

$$d_x = \sum_{m=1}^{M} \frac{|f_m(x_{+1}^m) - f_m(x_{-1}^m)|}{f_m^{max} - f_m^{min}} \tag{2.6}$$

f_m^{max} and f_m^{min} are maximum and minimum value of objective function f_m for the entire population [14]. Boundary solutions are assigned a crowding distance of ∞. In case η_i is larger than the amount of individuals in front i, the rest of survivors is filled up from front $i+1$ [16]. The other steps of NSGA-II and our implementation of them are described in section 5.1 in detail.

2.3. Innovization

To aid the optimisation process, Deb and Srinivasan proposed an *innovization* methodology [18] which extracts knowledge about a problem from Pareto-optimal solutions. The similarities, properties of variables or their relation, which were found to ensure the Pareto-optimality of solutions, can be reused when solving similar optimisation problems. This reused knowledge may even speed up the process. The authors introduce the following six step method:

1. Find optimal solutions for each objective separately using a single-objective optimisation technique and note down the ideal point z^*.
2. Compute a Pareto front using NSGA-II and define the nadir point z^{nad} using this front.
3. Normalise all objectives using the ideal point z^* and z^{nad} and cluster a few solutions.
4. Apply a local search to obtain a modified front.
5. Perform the normal constraint method (NCM) [50] with a few different starting points for verification of the found Pareto-optimal solutions.
6. Analyse the solutions for any commonality principles as plausible innovized relationships.

The authors used Benson's method [3] as a local search in step 4.

path	travel times for different departure delays										skewness	mean	SD	SSD
	+0	+5	+10	+15	+20	+25	+30	+35	+40	+45				
A	7	7	2	7	7	2	7	7	7	7	-1.78	6	2	0.89
B	6	8	6	4	2	6	6	6	10	6	0	6	2	1.41
C	5	5	5	10	5	5	10	5	5	5	1.78	6	2	1.79

Table 2.1.: Example of three paths with various travel time distributions over different departure delays in minutes. Mean and standard deviation (SD) fail to distinguish between these paths. The SSD is the only measure which recognizes path A as the most reliable. Example adapted from [69]

2.4. Further concepts

For our innovization variant and the route planning problems, some additional concepts are needed. Therefore, we explain the semi-standard deviation, the discrete Fréchet distance and the *Ordering Points To Identify the Clustering Structure* technique in this chapter.

Semi-standard deviation (SSD) To define one of our objective functions, we need to define SSD first. The SSD σ_{semi} for a discrete random variable with n observations o_i is computed using Equation 2.7 [69].

$$\sigma_{semi} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\big(\max(o_i - \mu, 0)\big)^2} \tag{2.7}$$

The SSD is computed in the same manner as a standard deviation with the exception that it only takes observations greater than the mean μ into account. This is preferable in an objective function, for example for travel time variability, since most drivers only care about minimising the risk of arriving too late at their destination. Compared to the standard deviation, the SSD has the additional advantage that it can differentiate between normal and skewed distributions [69]. Consider the example with three paths from Table 2.1. Path B has a normal distribution of travel times over different departure time delays, path A has a negatively skewed distribution and path C's distribution is positively skewed. Standard deviation and mean are the same for all three paths. The SSD, however,

recognizes that path A with a negatively skewed travel time distribution is the most reliable option and path C with a positively skewed distribution is the most unreliable. Both of these paths mostly exhibit travel times close to their mean but travel time outliers for path A are small values while outliers for path C are high values.

Discrete Fréchet distance The Fréchet distance [25] measures the similarity between two curves. It can be explained intuitively by the example of a person walking their dog on a leash [22]. The person and the dog take different curved paths. Their walking speeds may vary but they cannot backtrack. The Fréchet distance is the shortest possible leash that allows both to walk along their respective path. Since we define paths for routing problems as a sequence of nodes instead of continuously, we can use the discrete Fréchet distance δ_{dF} [22]. It is also referred to as coupling distance. Let $P = (n_{p1}, \ldots, n_{pl})$ and $Q = (n_{q1}, \ldots, n_{ql})$ be two paths which can be mapped to a metric space (Σ, d) such as a coordinate reference system with Euclidean distance. Let Γ be a coupling between P and Q as defined by Equation 2.8 [22].

$$\begin{aligned} &\Gamma(P,Q) = \big((u_1,v_1),\ldots,(u_p,v_q)\big) \\ \text{s.t. } &u_1 = n_{p1}, v_1 = n_{q1}, u_p = n_{pl}, v_q = n_{ql} \\ &u_{i+1} = n_{pi} \vee u_{i+1} = n_{pi+1} \qquad \forall\, i \in 1,\ldots,pl \\ &v_{j+1} = n_{qj} \vee v_{j+1} = n_{qj+1} \qquad \forall\, j \in 1,\ldots,ql \end{aligned} \tag{2.8}$$

A coupling is a sequence of distinct node pairs from P and Q which honours the order of nodes. The length of a coupling $||\Gamma|| = \max\limits_{k=1,\ldots,p} d(u_k, v_k)$ is the length of the longest link in Γ. The discrete Fréchet distance δ_{dF} is defined by Equation 2.9 [22].

$$\delta_{dF}(P,Q) = \min\{||\Gamma|| \mid \Gamma \text{ is a coupling between } P \text{ and } Q\} \tag{2.9}$$

Ordering Points To Identify the Clustering Structure (OPTICS) For the purpose of cluster analysis, Ankerst et al. [2] designed OPTICS. It creates an augmented ordering of the input data which represents its density-based clustering structure. The produced ordering is equivalent to density-based clusterings with various parameter settings. OPTICS takes two input parameters ϵ and $MinPts$. Based on core-distance and best reachability-distance of each point, an ordering is computed. The algorithm defines core points

which have at least $MinPts$ data points, including themselves, in a neighbourhood with radius ϵ. Based on this, each point is assigned a core-distance which is the distance to the closest of the $MinPts$ neighbours inside an ϵ-neighbourhood. The reachability distance from point p to another point o is defined as $max(\text{core-distance}(o), \text{distance}(o, p))$ with respect to some predefined distance function. Core-distance and reachability-distance of a point are undefined if no adequately dense cluster with respect to ϵ is available. For the ordering, OPTICS starts at a random data point and creates a priority queue sorted by smallest possible reachability-distance of each data point in the ϵ-neighbourhood of the starting point [2].
The authors also present a method to automatically extract a clustering. The produced ordering can be visualised in a reachability plot which plots ordering against reachability-distance. Since points in a cluster have low reachability-distances, they show up as valleys in the plot [2]. By detecting these valleys, clusters can be extracted automatically. Any points which are not part of a cluster are labelled as noise.
OPTICS has the advantage that ϵ, which is especially hard to determine for real-world data, does not need to be set by the user. ϵ can simply be fixed to infinity. However, lower ϵ values can be used to limit runtime of the algorithm. Furthermore, since OPTICS can extract a density-based clustering, it has the benefit that it can handle noise and arbitrarily shaped clusters with different densities [2].

3. Problem statement

In this chapter, we specify our research questions in section 3.1 and the problems we are working on in sections 3.2 and 3.3. The goal of this thesis is to develop a method for applying innovization to multi-objective routing problems. The difficulties, that arise when trying to use the original innovization to route planning problems, are detailed in section 3.2. Because of this, we examine how we can adapt the innovization method for route planning problems. The success of our proposed variant is tested on route planning problems as defined in section 3.3.

3.1. Research questions

We aim to explore the following three research questions (RQs) to develop and test an innovization for route planning methodology:

RQ1 How can the innovization methodology [18] be applied to route planning problems?

RQ2 What knowledge is gained from the Berlin Uber Movement dataset when applying RQ1's innovization for route planning?

RQ3 Can the knowledge from RQ2 improve efficiency or results of an MOEA?

An answer to RQ1 is proposed in chapter 5. We apply the method proposed in RQ1 to exemplary routing problems and do a qualitative analysis of the results for RQ2. To evaluate the last research question, we select a metric which is independent of any hardware unlike, for example, runtime. Since we aim to test our method on real-world problems, we additionally cannot use any metric for which the optimal Pareto front has to be known. Therefore, one of the metrics we use for RQ3 is *efficiency*. Our definition for efficiency is based on the one from Gupta, Ong and Feng [28]. We define efficiency of an optimisation algorithm on problem instance P as $Q_t(P)$ which represents the quality of solutions achieved

with regard to multiple objective functions in t time-steps on a designated computer. To compare the efficiency of two multi-objective algorithms, we use normalized hypervolume as a quality measure. We normalize all function values f_m for each objective m with Equation 3.1 using ideal point z^* and nadir point z^{nad} [14].

$$f_m^{norm} = \frac{f_m - z_m^*}{z_m^{nad} - z_m^*} \tag{3.1}$$

Given a set of normalized function values $F^{norm} \subset [0,1]^M$ from a set of non-dominated solutions, the hypervolume (HV) with regard to reference point z^{ref} is computed using Equation 3.2 [27].

$$HV(F^{norm}, z_{ref}) = \Lambda \left(\bigcup_{\substack{f \in F^{norm} \\ f \leq z^{ref}}} [f, z^{ref}] \right) \tag{3.2}$$

Λ refers to the Lebesgue measure and $[f, z^{ref}]$ defines a rectangle delimited by function value f and reference point z^{ref}. We normalize hypervolumes since we test our algorithms on different route planning problems which can return highly different fronts. To calculate hypervolumes, a reference point has to be defined which is not a trivial task. Ishibuchi et al. [34] recommend that the reference point should be worse than nadir point z^{nad} to include extreme solutions in the hypervolume. However, they do not identify how much worse is good for comparison. We estimate z^{nad} as the vector of the worst objective function values for each objective in the Pareto front. Since we normalize our objective values for hypervolume calculations, the nadir point always is 1 in each objective. As per the recommendation, we multiply this nadir point by 1.05 to use as a reference point. The algorithms we compare are two MOEAs. That means to determine their efficiencies, we stop both algorithms after t generations and compare their hypervolume. A larger hypervolume after t iterations indicates better efficiency. Furthermore, we want to compare the final results obtained by running both algorithms with the same stopping criterion. To this end, we analyse the total number of function evaluations and the hypervolumes of final Pareto fronts for each run.

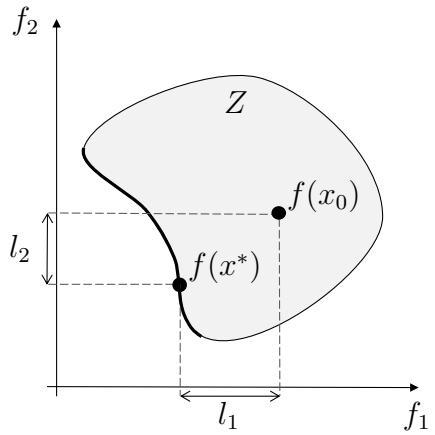


Figure 3.1.: Example of Benson's method on a bi-objective problem. The true Pareto front is indicated by the bold line. Figure adapted from [21]

3.2. Challenges of route planning innovization

The innovization method by Deb and Srinivasan [18] computes a Pareto front for a specific problem. Since this Pareto front should be used to gain insights about the problem, which shall be reused for similar problems, there should be relatively high confidence in the results. Hence, the authors use a local search on a few well-distributed solutions and apply the NCM [50] at several different starting points. When trying to apply innovization to route planning problems, some difficulties arise.

In the fourth step of the innovization, Benson's method is employed as a local search. The goal is to find dominating solutions for a few select individuals or rather to verify that there are no dominating individuals. As we can see in Figure 3.1, Benson performs the search by maximising distances l_m for each objective m between an initial solution x_0 and some efficient, meaning optimal, solution x^* within feasible region S [21]. Mathematically, the local search for an M-objective problem with feasible region S of the decision space is the optimisation of the auxiliary problem in Equation 3.3.

$$\begin{aligned} \max \quad & \sum_{m=1}^{M} l_m \\ \text{s.t.} \quad & f_m(x_0) - l_m - f_m(x) = 0 \quad m = 1, \dots, M \\ & l_m \geq 0 \qquad\qquad\qquad\quad m = 1, \dots, M \\ & x \in S \end{aligned} \tag{3.3}$$

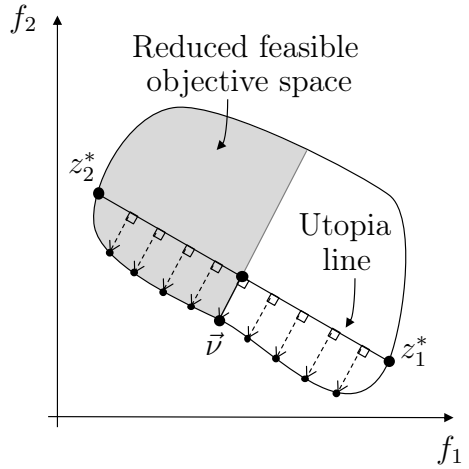


Figure 3.2.: Illustration of the NCM approach on a bi-objective example. Figure adapted from [50]

This auxiliary problem can also be defined for route planning problems. However, route planning problems have extremely complex constraints such as road networks where node indices in possible encodings do not follow a clear system. These types of constraints lead to disconnected feasible regions. Therefore, methods such as algorithms for Integer Programs or gradient-based methods are not applicable to the auxiliary problem for route planning. Moreover, exact methods would not produce efficient solutions within a reasonable time frame. In conclusion, solving Benson's auxiliary problem for route planning is similarly complex as solving the original routing problem which means that it makes sense to use a metaheuristic for the solve. As a result, the guarantees from Benson's method would no longer apply. That means, we cannot assure finding a dominating solution for an initial solution if there is one. Therefore, we propose a new local search for finding dominating solutions that is more suitable for routing use cases. We call our local search Perimeter Mutation Local Search (PMM-LS) and define it in section 5.2.
In the fifth step of the innovization, NCM [50] is used for verifying the obtained Pareto front. The difficulties for route planning problems are similar to those in the fourth step. NCM also defines single-objective auxiliary problems to generate Pareto points. These points are then compared to the solutions in the Pareto front obtained through innovization. A visualisation of how the auxiliary problems from NCM work can be found in Figure 3.2. Each auxiliary problem reduces the feasible objective space Z to one dimension using one of the evenly distributed normals $\vec{\nu}$ along a Utopia line which connects all

single-objectively optimal solutions. Within the reduced objective space, the remaining objective is optimised using a gradient-based method [49]. Due to complex decision variables and often non-differentiable objective functions, gradient-based methods are not applicable. Even solving the single-objective auxiliary problem is only viable with a metaheuristic since brute force would take too long. This means that guarantees about the Pareto-optimality of a set generated by NCM can no longer be made. To the best of our knowledge, there is no other verification method applicable to routing problems that could replace NCM in the fifth step.

3.3. Route planning problem

For this thesis, let a time-dependent route planning problem $\big(\{t_{travel}, ttv, deg_{turn}\}, G, n_O, n_D, w_0, t_0\big)$ be defined as

$$
\begin{aligned}
\min \quad & \big(t_{travel}(x, w_0, t_0), \\
& ttv(x, w_0, t_0), \\
& deg_{turn}(x)\big) \\
\text{s.t.} \quad & x \in \big\{\ (n_0, \ldots, n_l)\ \mid\ l \in \mathbb{Z}_{\geq 1}; \\
& \qquad\qquad n_0, \ldots, n_l \in V; \\
& \qquad\qquad \forall\, i \in \{0, \ldots, l-1\}\ \ (n_i, n_{i+1}) \in E; \\
& \qquad\qquad n_0 = n_O; n_l = n_D\big\} \\
& w_0 \in \{0, 1\} \\
& t_0 \in \{hh : mm : ss \mid\ hh \in [0, 23]; mm, ss \in [0, 59]\}.
\end{aligned}
\tag{3.4}
$$

The goal of such a route planning problem is to find a route which can be taken by some vehicle and which minimises the three objective functions explained below. The objective functions are conflicting which is the reason they have to be optimised simultaneously. Hence, we have a multi-objective route planning problem. The values of our objective functions depend on a route x of variable length l from an origin node n_O to a destination node n_D as well as w_0 and t_0 which define the departure time. w_0 determines whether the departure is during the week ($w = 0$) or on the weekend ($w = 1$) and t_0 is a time specification. The time-dependent road network is defined as a property graph $G = (V, E)$ with a set of vertices, or nodes, V which are connected by a set of edges E. The property graph is directed and allows for attributing

properties to nodes and edges [57]. Nodes are assigned indices as identifiers (IDs). Edges $E \subseteq \big\{\big((u,v), \{s_{uv}^{wh} \mid s_{uv}^{wh} \in \mathbb{R}_{\geq 0}\, \forall w \in \{0,1\}\ \forall h \in [0,23]\}, \{tt_{uv}^{wh} \mid tt_{uv}^{wh} \in \mathbb{R}_{\geq 0}\, \forall w \in \{0,1\}\ \forall h \in [0,23]\}\big) \mid u,v \in V; s_{uv}^{wh} \in \mathbb{R}_{\geq 0}; tt_{uv}^{wh} \in \mathbb{R}_{\geq 0}\big\}$ have hourly average speed attributes s_{uv}^{wh} for during the week and on weekends. These speeds and lengths of edges are the basis for the computation of travel time properties tt_{uv}^{wh} for days during the week and on weekends and each hour of the day h. To get the travel time per minute, hourly travel times are linearly interpolated. Furthermore, each edge is assigned a number of traffic signals sgn_{uv} encountered when traversing the edge in direction from u to v.

First Objective: Travel time One of the main criteria when people plan a trip from one point to another is the time taken to travel along a route. Unlike the shortest path, the fastest route depends on multiple factors such as speed limits, congestion or traffic signals. Therefore, re-evaluating a path in regard to this objective becomes necessary before every new trip. For example, a commuter may take the same shortest path every day but the fastest route might be vastly different depending on the day or just the departure time. We compute the travel time of a route x in seconds using Equation 3.5.

$$\min\ t_{travel}(x, w_0, t_0) = \sum_{i=0}^{|x|-1} tt_{x[i]x[i+1]}^{w_i t_i} + sgn_{x[i]x[i+1]} \cdot 20 \tag{3.5}$$

$|x|$ denotes the length of route x. The variables w_i and t_i calculate the current time and whether it is the weekend when reaching node i depending on the departure time and the travel time already needed to arrive at node i. Like Kanoh [40], who also inspired our third objective, we give a penalty of 20 seconds for each traffic light along the route. This is because signal systems are a factor that significantly affects travel times [46].

Second Objective: Travel time variability Another important factor for route choice is travel time variability. Gan and Bai [26] report that drivers are less likely to choose paths with higher travel time variability. From different possibilities for measuring variability, we select the upper semi-standard deviation (SSD). It only takes observations greater than the mean into account which is preferable because most drivers only want to minimise the risk of arriving too late at their destination. Since drivers might be unsure when they are actually starting their journey and edges only contain travel times for days

during the week or on weekends, we minimise the travel time variability of a route x over varying departure time. We measure the travel time variability with Equation 3.6 in the interval $\{t_0 + i \cdot 15 \mid i = \text{-}6, -5, \ldots, 6\}$ which is 90 minutes before and after a planned departure time in 15 minute increments.

$$\min\ ttv(x, w_0, t_0) = \sqrt{\frac{1}{13} \sum_{i=\text{-}6}^{6} \Big(\max\big(t_{travel}(x, w_0, t_0 + i \cdot 15) - \overline{tt}, 0\big) \Big)^2} \tag{3.6}$$

$\overline{tt}$ refers to the mean of all travel times in the time interval with the 13 observations. As mentioned above, the hourly travel times of an edge are linearly interpolated depending on the minute of each time $t_0 + i \cdot 15$. If this was not done, the algorithm would be encouraged to minimise travel time variability by taking detours until the next hour starts where lower or no variability was reached for certain edges.

Third Objective: Ease of driving Especially in cities, the fastest path with the most reliable travel time might not be the most comfortable for the driver. For example, by taking smaller roads through residential areas, drivers might be able to avoid congested roads but need to make many more turns. This is not only strenuous but may also negatively affect time and energy consumption on a route. Therefore, our third objective maximises ease of driving which is inspired by Kanoh and Hara [42]. To maximise ease of driving, we minimise the sum of degrees of turning needed to drive along a route x with Equation 3.7.

$$\min\ deg_{turn}(x) = \sum_{i=0}^{|x|-2} 180^\circ - \angle\big(x[i]x[i+1]x[i+2]\big) \tag{3.7}$$

Note that a turn can mean any angle from 0° to 180°. Therefore, it may be a U-turn (180°) or a straightforward crossing of an intersection (0°). We subtract the turning angle from 180° since the sum of turning angles could simply be maximised by an EA by making routes longer. This objective could be expanded by including, for example, incentives to avoid traffic jams, or to use wider streets or roads with more lanes. Avoiding congested streets, for example, is already encouraged by other objectives which is why it is not part of the third objective again. The definition for ease of driving could further be refined by differentiating between right-hand and left-hand turns. When driving on the right side of the road, making left-hand turns is less comfortable. Similarly,

left-hand turns are more time-consuming which could be integrated into the travel time objective. However, since not all countries drive on the right side of the road, we do not include this differentiation in our objectives.

4. Related work

In this chapter, we contextualise our thesis with respect to related work on innovization and multi-objective route planning in time-dependent networks in sections 4.1 and 4.2. Furthermore, subsection 4.2.1 explores related research regarding two of our objectives, travel time variability and ease of driving. Finally, we compiled literature that uses Uber Movement Speeds datasets in section 4.3.

4.1. Innovizing path planning

Our proposed innovization for route planning approach is based on the innovization by Deb and Srinivasan [18]. Both methods share the intention of extracting innovized principles that can be reused in the optimisation of similar problems. As explained in section 3.3, the original innovization cannot simply be applied to route planning problems. Therefore, we propose an adapted version specialised for this kind of problems. To the best of our knowledge, there are no works using innovization for problems in the area of path or route planning. Coello Coello et al. [12] applied innovization to a related problem which is a bi-objective travelling salesman problem. They discovered that there are some edges which are rarely part of Pareto-optimal solutions. As an innovized principle, authors suggest excluding these infrequent edges during optimisation to lower computational effort. In route planning, the value of this type of knowledge is limited since it is not often reusable in complex road networks. There, routes for various route planning problems often use completely different streets even when planning in the same city, for example. Interestingly, there are also efforts to automate innovization [15] so that no human intervention is needed. This could be a future extension of our proposed approach.

4.2. Multi-objective time-dependent route planning

For parts of the innovization for route planning, we also design a MOEA for route planning in time-dependent networks. In this kind of networks, some of its information changes with time. In our case, the travel times of edges depend on the time and whether it is the weekend or not. Planning is generally done offline. This has to be distinguished from dynamic route planning where solutions are computed and recomputed online while driving [24].
For time-dependent route planning, we can also differentiate between sources of time-dependent travel times. The travel times for our experiments are exclusively based on a dataset containing historical speed data. Sometimes, historical data is used to create some kind of estimation model for travel times [61]. A good overview of different prediction methods is given by Lin et al. [46]. Others use real-time traffic information gathered via a vehicular-ad-hoc-network (VANET) in addition to historical data [10, 56]. Another option is modelling travel times stochastically [45, 55, 68]. Since integrating an estimation model for travel times is a complex topic, it is beyond the scope of this thesis. In future, some prediction method could be implemented to improve the accuracy of routing algorithms for real-world applications.
The work most similar to our route planning method is that of Kanoh and Hara [42]. With time-dependent travel time, route length and ease of driving, they define similar objective functions. Unlike us, they use a cellular automata [41] to calculate traffic prediction before the optimisation. Moreover, we utilise NSGA-II while they propose a combination of the Dijkstra algorithm [20] and a genetic algorithm with different operators than ours. Liu et al. [47] also use NSGA-II with Node Based Crossover (NBX) but other steps of their EA differ from ours. Furthermore, they have similar objective functions which are total vehicular emission cost, time-dependent travel time, number of turns, and route length. Although the authors utilise real-world data too, they only test their method on a single path planning problem in a much smaller road network than us.
Other papers only deal with single-objective time-dependent problems [44] or the multi-objective problem is optimised single-objectively using a weighted sum approach [8]. There are also more papers on multi-objective problems in time-dependent networks, but they examine different problem definitions. The problems are usually designed for other use cases such as VRPs [29, 43, 64, 73],

path planning for a set of emergency vehicles [72] or time-dependent multi-hop ride sharing [32].

4.2.1. Objectives

Besides the relatively common travel time minimisation, we optimise two other objectives which are discussed in previous research. In the first paragraph, we compile different approaches for defining travel time variability. The second paragraph shows the inspiration for our ease of driving objective.

Travel time variability Our second objective is to minimise travel time variability, also called travel time reliability. Rajabi-Bahaabadi et al. [55] provide a good overview of literature minimising travel time variability and of different function definitions for reliability. We highlight some papers from this overview that are most relevant to our work and extend the list by some additional works. Mostly, the variability is defined as the expected value of travel time, its variance or a combination of both [33, 35, 58, 66]. These definitions have the disadvantage that they also take arriving too early into account. This is something drivers are usually not concerned by. For this reason, Rajabi-Bahaabadi et al. [55] additionally minimise the probability of travel times exceeding pre-specified travel times. Ishibuchi et al. [36] instead optimise travel time budgets that ensure the driver's on-time arrival with a certain confidence level. Wellman, Ford and Larson [70] on the other hand use a SSD of travel times as an objective function. The latter is the option we choose because of the benefits already explained in section 2.4. Another interesting idea is minimising the congestion probability of routes instead of the travel time variability [71].

Ease of driving Our third objective function, ease of driving, was inspired by Kanoh and Hara [42]. They implement ease of driving as a function which penalises narrower roads and greater numbers of turns, signals and traffic jams. We included some of these aspects in the calculation of our travel time objective. In our thesis, we simplified ease of driving to minimising the degrees of turning needed to drive along a route. Especially in robotics, this is also referred to as path smoothness [1, 39, 65]. Alternatively, the number of turns is minimised instead of the angles [6, 53].

4.3. Uber Movement datasets

For our experiments, we use the Uber Movement Speeds dataset for Berlin [62]. Previous literature related to route planning with Uber Movement Speeds data exists. However, Uber Movement also provides datasets for cities other than Berlin that are mostly used.
Some papers create travel time estimation models with the data but do not use them in a route planning algorithm [48]. Deb et al. [13] additionally analyse patterns in their prediction model to extract knowledge for routing in Mumbai, India. They learned that holidays cause irregular patterns in the data which is something we too observed during the data preparation. In our case, there was a massive increase in recorded rides just after midnight on New Year's Day 2020. However, due to averaging for some weekdays and interpolating for missing edges, this anomaly was no longer noticeable during route planning. Deb et al. furthermore observed that travelling in the evening can take longer than during the PM peak period. Unfortunately, we cannot corroborate this since the authors do not specify which time they consider to be the PM peak period.
Some papers also use Uber Movement data for route planning optimisation. Zheng et al. [74] minimise expected travel time and a travel time budget in Manhattan, New York City. In contrast to our work, their network is not time-dependent. Ch, Krumm and Kun [9] optimise the departure time for a specific route whereas we find a route for given a departure time that minimises the possibility of taking longer than the expected travel time. Overall, we found no papers using any Uber Movement dataset for innovization.

5. Innovization for route planning

To apply innovization to route planning problems and alleviate the challenges described in section 3.2, we propose the following, adapted version of the original innovization method [18]:

1. Compute a Pareto front using a multi-objective evolutionary algorithm (MOEA) for routing problems.
2. Verify extreme solutions using a single-objective algorithm. Add new extreme points discovered during the single-objective runs to the Pareto front and compute the non-dominated set to receive the extended front.
3. Apply Perimeter Mutation Local Search (PMM-LS) to the extended front to obtain the modified front.
4. Prepare the analysis: Cluster solutions of the modified front in the decision space. Plot pairs of objectives for the clusters. Compute correlation coefficients between different route characteristics and objectives.
5. Analyse the Pareto-optimal solutions and materials from the previous step. Check for any commonality principles which can be reused as extracted knowledge. To this end, it might be useful to combine the analysis of modified fronts from multiple runs, for example with different start and end points or with different departure times.

A detailed explanation of our MOEA for the first step follows in section 5.1 and for PMM-LS in section 5.2. As suggested in the original innovization method, the MOEA was reused for single-objective runs in the second step by defining only one objective function at a time. It is possible that single-objective algorithms return new extreme points instead of verifying the ones in the front. This is because finding extreme points for routing problems is hard due to the complexity of decision variables and constraints. In some cases, a single-objective

algorithm outputs multiple individuals with the same optimal value. Trivially, only non-dominated extreme solutions should be added to the extended Pareto front. Moreover, newly discovered extreme solutions might dominate solutions in the extended front. It is therefore necessary to compute the non-dominated set of the Pareto front from the first step along with any newly found extreme points.
For the decision space clustering in the fourth step, we apply OPTICS [2] with automatic cluster extraction. The advantage of this method is that there is no need to define a number of clusters or an ϵ-value for the neighbourhood size. In addition, OPTICS is able to handle arbitrarily shaped clusters with different densities and noise. The only parameter which has to be set is $MinPts$ for identifying core points. We employ the heuristic $MinPts =$ number of objectives $+2$ by Ester et al. [23] which they discovered for DBSCAN, another density-based clustering algorithm similar to OPTICS. As the distance function for routes in the decision space, we utilise the discrete Fréchet distance δ_{dF} [22] with Euclidean distance between node coordinates. We recommend visualising clusters of routes on a map for analysis. The route characteristics mentioned in the fourth step are the number of traffic signals along a route and the percentages of street types. As street types, we distinguish between motorways, main roads, residential streets and other or unclassified streets.
For route planning problems, analysing fronts separately in the fifth step may result in the extraction of knowledge which is too problem-specific. For example, we could find from one run with specific start and end points that routes always pass through one specific node or always use a certain edge. However, this knowledge is not applicable to most other route planning problems with other start and end points. An in-depth discussion of our entire proposed innovization for route planning is contained in section 5.3.

5.1. Multi-objective evolutionary algorithm

We use NSGA-II [17] as our multi-objective evolutionary algorithm (MOEA). NSGA-II is an EA which iteratively applies crossover and mutation to a population of individuals until a given termination criterion is reached. We customised some of the steps for route planning which are explained in the following subsections. To encode our route planning problems, we want to employ a natural encoding which is also suitable for real-world data. Since our road network is represented as a graph $G = (V, E)$, chromosomes are a sequence of node indices I_{n_i}

from start n_O to destination n_D of our route planning problem. Formally, an individual $x \in \{(I_{n_0}, \ldots, I_{n_l}) \mid l \in \mathbb{Z}_{\geq 1}; n_0, \ldots, n_l \in V; n_0 = n_O; n_l = n_D\}$. This way, similar chromosomes have similar fitness values. Feasible individuals must have an edge between each pair of consecutive nodes. As paths may have different lengths in reality, our encoding also has variable length.
In the first step of our MOEA, an initial population is created using guided random walks as described in subsection 5.1.1. The size of our population is chosen dynamically for every route planning problem based on its difficulty. One factor which influences the difficulty is the length of solutions. Therefore, we choose the population size N for a routing problem based on the length of the shortest path l_{sp} between start and end point. Equation 5.1 computes the population size as 2.5 times the length of a shortest path rounded to the nearest ten.

$$N = \left\lfloor \frac{2.5 \cdot l_{sp}}{10} + 0.5 \right\rfloor \cdot 10 \tag{5.1}$$

Individuals of the initial population are then evaluated according to a fitness function. In our case, the fitness is a vector of the objective functions described in section 3.3. Additionally, we give a penalty of $100,000$ to each objective for any missing edge between consecutive nodes in a chromosome. Based on this fitness function, NSGA-II employs binary tournament selection to choose solutions for creating new individuals with crossover and mutation. To fill each spot in the mating pool, two random individuals compete in a tournament [14]. The winner is the dominating solution or, if neither solution dominates the other, it is the solution with greater crowding distance. We customise crossover and mutation operators as we explain in subsections 5.1.2 and 5.1.3. Survivors are then selected from the population including newly created individuals based on fast non-dominated sorting and crowding distance sorting.
These steps are repeated until our stopping criterion is reached. We follow the recommendation from the original innovization [18] to let the MOEA run for a large number of generations. The goal of this is to reach a high confidence in innovization results. The understanding of a large number of generations varies per routing problem. Therefore, we terminate if the change in ideal point z^*, nadir point z^{nad} and Inverted Generational Distance (IGD) is less than 0.0025 for 25 consecutive generations. This set of indicators ensures that the algorithm is stopped after diversification and converging of the algorithm are finished [5]. Stopping criteria are evaluated over multiple generations to make the termination more robust. As a fall-back, we stop the algorithm after 200 generations.

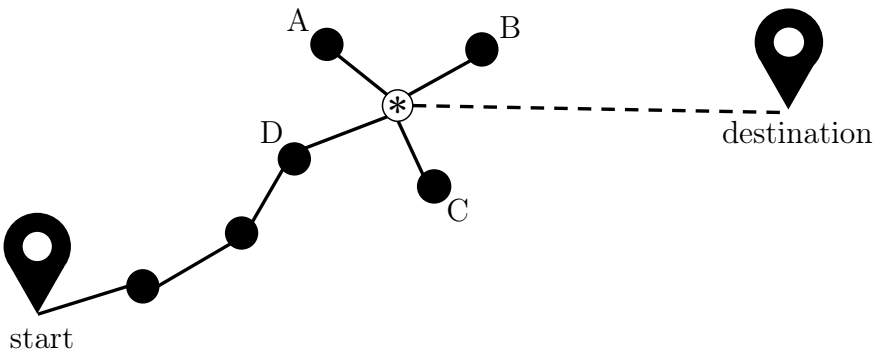


Figure 5.1.: Generation of an example individual. From the neighbours A, B, C and D of current node $*$, B is chosen with a 75 % chance. A and C are selected with a 12.5 % chance each

5.1.1. Generation of initial population

The initial population is created using guided random walks since standard random walks take too long in complex graphs like road networks. A random walk starts at the given start point and ends when the predefined destination is reached. Consider the example in Figure 5.1 where we are currently at node $*$. If present, we remove any visited neighbours from the selection pool. In this case, D would be excluded. For simplicity, self-loops are always disallowed. We move in direction of the node that is closest to the direction of our destination with a probability of 75 %. In the example, this applies to node B. The other 25 % chance are evenly distributed among the rest of the neighbours or, in this instance, A and C. This guidance is a trade-off between diversity of initial populations and runtime. However, the direction of a neighbour towards the goal can sometimes be misleading and the random walk gets stuck in a circle. If the current node has been visited at least ten times and all neighbours have already been visited, we choose a successor uniformly at random from all neighbours to get out of any loops.
For future work, one could consider removing any loops from individuals after their generation or disallowing revisiting nodes altogether. However, the latter would mean that random walk and reproduction get more complex as multi-edge U-turns are sometimes necessary to prevent individuals from becoming infeasible.

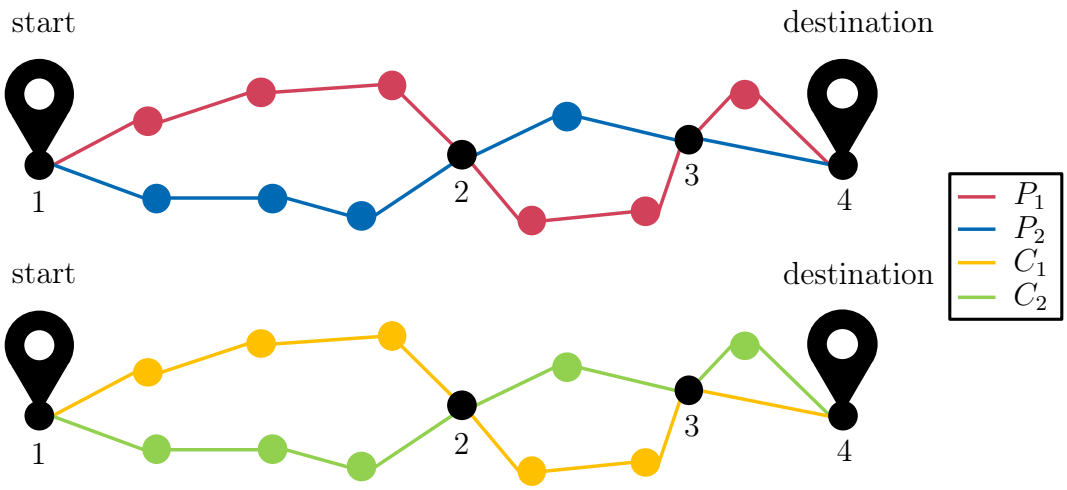


Figure 5.2.: Crossover of two example solutions P_1 and P_2 at common node 3 resulting in children C_1 and C_2

5.1.2. Node Based Crossover

Recombining two paths can be intuitively done as a one-point crossover when paths share common nodes like in Figure 5.2. To the best of our knowledge, this kind of crossover was first described by Munetomo et al. [52] and has later been termed Node Based Crossover (NBX) [11]. In the example, the two parent solutions P_1 and P_2 have common nodes 1, 2, 3 and 4. Since a crossover at start (1) or end point (4) would not produce a new individual, we choose any other common node for a one-point crossover. Say we randomly select node 3, the crossover produces children C_1 and C_2 by giving children a different part of each parent. If two parents share no common points, a one-point crossover at a randomly placed cut is performed. The position of the cut is random but it is restricted such that a crossover at the start or end point is prevented. The resulting individuals are repaired with shortest path between the node before the cut and the one after it.

5.1.3. Perimeter Mutation Operator

After the creation of offspring using crossover, a mutation operator is applied to them. Our mutation operator is largely inspired by the Perimeter Mutation Operator from Weise and Mostaghim [67]. The idea is to replace a random part of the path by an alternative route. A visualisation of our mutation operator can be found in Figure 5.3. First, we choose random mutation start and end indices.

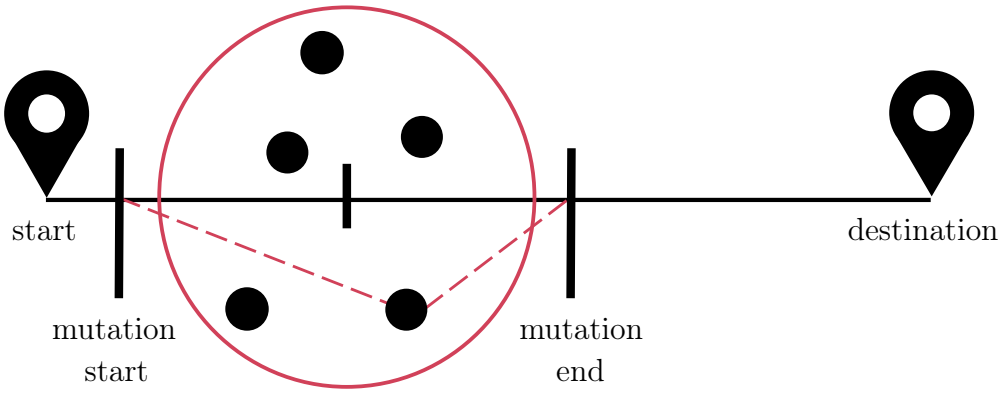


Figure 5.3.: Visualisation of our mutation operator which replaces a random, small part of the path with an alternative path

To ensure that the mutation is only a small change but that an alternative route can be found, the indices are within 10 % to 20 % of chromosome length apart. Secondly, we determine the middle point between the two nodes at the chosen mutation indices. We find all nodes within a circle around this middle point with a radius of 0.8 times the distance between mutation start and end node. In Figure 5.3, this is represented by the red circle. We randomly select one of the nodes which is not already on the path. Finally, we replace the path between mutation start and end node with the shortest path between mutation start and end via the chosen alternative node. Since there might be no alternative nodes available within a radius, not every mutation is successful. Therefore, we set our mutation probability a little higher than usual at 25 %.

5.2. Perimeter Mutation Local Search

Perimeter Mutation Local Search (PMM-LS) is a local search method based on the mutation from previous subsection 5.1.3. It is more systematic than the small mutations of few individuals during the course of an EA. The goal of PMM-LS in this particular case is finding solutions dominating individuals in the extended front P_{res} from step 2 of our innovization for route planning. The pseudocode of PMM-LS is detailed in Algorithm 1. For every individual in P_{res}, alternative routes are created using the mutation from subsection 5.1.3. This mutation is applied for every possible mutation window of size 20 % of chromosome length. Instead of randomly selecting only one alternative node as for the MOEA, all possible alternative routes are generated. After each set

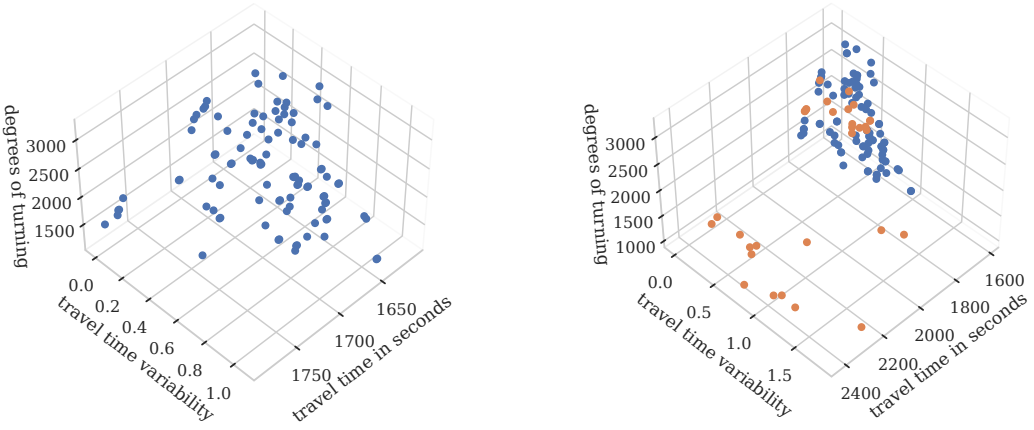
Algorithm 1 PMM-LS(P_{res})

```
function DIST(A, B)
    return GREAT-CIRCLE_DISTANCE(A, B)

function CONCAT(x, w_start, w_end, node_alt)
    route_alt = x[0, ..., w_start]
    route_alt = route_alt : SHORTEST_PATH(x[w_start], node_alt)
    route_alt = route_alt : SHORTEST_PATH(node_alt, x[w_end])
    route_alt = route_alt : x[w_end, ..., |x|]
    return route_alt

P = P_res
X_new = P_res
step = 1
while X_new ≠ ∅ do
    P_old = P
    for x ∈ X_new do
        w_start = 0
        w_end = w_start+ROUND(|x| · 0.2)
        while w_end ≤ |x| do
            centre = MIDDLE_POINT(x[w_start], x[w_end])
            radius = {v ∈ V | DIST(v, centre) ≤
                                0.8·DIST(x[w_start], x[w_end])}
            for node_alt ∈ radius do
                P = P ∪ CONCAT(x, w_start, w_end, node_alt)
            w_start = w_start + step
            w_end = w_end + step
    P = NON-DOMINATED_SET(P)
    X_new = P\P_old
return P
```

of mutations, the sliding window is moved by *step* size of 1. This parameter can be changed to a higher value for a trade-off between computation time and completeness of the search. After all alternative routes for all individuals have been computed, only the set of non-dominated individuals of P_{res} and alternative routes is kept. These steps are repeated for any newly found individuals until no more new individuals were discovered. Finally, the method returns a locally improved Pareto front.

5.3. Discussion

In this chapter, we have answered RQ1 by proposing an adapted innovization method which is specialised for handling route planning problems. The specialisation was achieved particularly by developing a new local search method for routing problems, PMM-LS. This local search defines a route's neighbourhood and systematically searches for improvements within these neighbourhoods. Since route planning problems have complex decision variables and constraints, and often non-differentiable objective functions, applying the local search and the verification method from the original innovization [18] is only viable when using a metaheuristic. However, that means that the guarantees of these methods no longer hold. Therefore, the biggest difference of our innovization variant in comparison to the original is the introduction of a specialised local search and the omission of the NCM verification step. For a more comprehensive assessment of the challenges of the original innovization, refer to section 3.2. Additionally, the order of steps has been revised to include improvements from newly discovered extreme points and PMM-LS when clustering. Inserting new extreme points from single-objective runs into the front before PMM-LS helps improve the results of the local search even further. The reason for this is that routing problems are deceptive. This means that extreme points with the same optimal value for one objective are not necessarily close in the decision space. Because of this, PMM-LS often reveals previously undiscovered parts of the Pareto front. An example of this can be seen in Figure 5.4. In the example, new points close the MOEA front are found along with completely new parts of the front. These new routes dominate many routes of the MOEA front. Overall, PMM-LS proved useful in our experiments. The local search lead to a median normalised hypervolume improvement of about 0.02 when comparing the fronts produced by our MOEA and by PMM-LS. In future, other design choices for PMM-LS such as different search strategies could be explored. For example,



(a) Result of MOEA

(b) After PMM-LS

Figure 5.4.: Pareto fronts of an exemplary routing problem as a result of the MOEA step (a) and after inserting extreme points and performing PMM-LS (b). Orange points are newly discovered

more exploration could be included instead of the current greedy approach where we only check neighbourhoods of newly found, improved individuals. One idea is to gradually decrease the size of the mutation window and thus decrease the radius in which alternative routes are examined. This allows for more exploration in the beginning of the search though a larger mutation window increases computation time. Another idea is to integrate PMM-LS into a Variable Neighbourhood Search (VNS) [31] to break out of potential local optima after PMM-LS converges. VNS applies local search within a set of larger neighbourhood structures, moving from one neighbourhood to another if no improvement was found.

PMM-LS is one of the steps of our innovization variant that can be slow, especially for Pareto fronts with many individuals. However, a trade-off between computation time and completeness of the search can be achieved by adjusting the parameters for window size or step size. Moreover, parallelising multi-objective and single-objective optimisations accelerates the process. This has also been suggested by the authors of the original innovization [18]. Another noteworthy difference of our proposed innovization variant is that we choose the population size dynamically. Equation 5.1 computes population size depending on the difficulty of each routing problem which is indicated by the length of the shortest path from start to end node. Thereby, we prevent an unnecessarily

large excess of function evaluations. Like the original innovization method, innovization for route planning overall remains a time-consuming procedure. However, it is intended to be applied only once to a set of problems. Its benefit arises from using extracted knowledge for optimisations of similar problems where this knowledge ideally reduces computational effort.
In contrast to the original innovization, we only perform the clustering after inserting new extreme points and conducting a local search. The reason for this is that PMM-LS often returns dominating solutions which would have to be assigned to already existing clusters. Furthermore, dominated solutions would have to be deleted from clusters. Overall, this could result in a totally different cluster structure. Additionally, we do not cluster in the objective space but in the decision space. As we do not cluster in the objective space and we omit NCM, there is no need to compute the ideal and nadir point or to normalise objectives. Since we use the clustering for analysis, individuals of the same cluster need to possess some commonalities. However, finding meaningful similarities in objectives space clusters of route planning problems is difficult due to the deceptiveness of route planning and the complexity of decision variables such as lists of node indices. Moreover, principles, that can be found for objective space clusters, are often too problem-specific. Deb and Srinivasan [18] only use clustering to identify some well-distributed solutions for which they run their local search. Similarly, we could use representative solutions from decision space clusters to speed up PMM-LS. Future work for innovization for route planning could also include advancing the clustering step by testing other clustering algorithms or by finding a heuristic for OPTICS' ϵ parameter. Setting ϵ to less than infinity would decrease the runtime of the clustering.

6. Analysis

For our experiments, we implement the method from the previous chapter in Python using the pymoo framework [4] for our MOEA. Due to time restrictions and since the MOEA just serves as a step in our innovization, we do not conduct a systematic hyperparameter optimisation. Instead we set parameters to values which we observed to work well during implementation. A hyperparameter optimisation, especially for mutation probability and dynamic population size, should be run in future to try to improve the MOEA. To calculate time-dependent travel times for a network, we use the Uber Movement Speeds dataset [62] for Berlin. This dataset is explained and discussed in the following section. Section 6.2 thereafter covers the knowledge extraction from innovization experiments run for RQ2. Innovized principles from this section are used for the experiments for the following section 6.3 where we examine whether an improvement of our MOEA is achieved. We conclude this chapter with a discussion of our results.

6.1. Uber Movement Speeds dataset

Defining route planning problems from section 3.3 for our experiments requires a graph representing a road network. Additionally, each edge (u, v) needs a number of traffic signals sgn_{uv} and a speed attribute s_{uv}^{wh} for every hour h during the week and on weekends w. Road networks are readily available as OpenStreetMap (OSM) [54] graphs with the OSMnx package [7] in Python. The edges of these OSM graphs also contain relevant information such as the number of traffic signals. As a source for speed attributes s_{uv}^{wh}, we choose the Uber Movement Berlin Speeds dataset [62]. This dataset contains mean hourly speeds and their standard deviation recorded for Berlin street segments in January of 2020. Start and end points of each street segment are identified by OSM node IDs. Each data entry is an average traversal speed from a sufficient number of Uber trips on a specific street segment in one hour of a certain

day. Travel time attributes tt_{uv}^{wh} are calculated based on interpolated hourly speeds attributes s_{uv}^{wh} and length of edges provided by OSM. The determination of speed attributes for each edge in the OSM graph of Berlin is described in subsection 6.1.1. Afterwards, we discuss the dataset in subsection 6.1.2.

6.1.1. Determination of speed attributes

Since our routing problems only pertain to motor vehicle, we only load the OSM graph with the drive network type for feasibility and runtime reasons. Uber maps GPS data from trips to internal node IDs first before providing a matching to OSM node IDs. Unfortunately, the OSM IDs provided by Uber often do not match node IDs in the drive network. Sometimes the size of segments are not equivalent. In many other cases, nodes are not even part of the drive network but instead represent, for example, footpaths or lifts. We hypothesise that this is because Uber Movement only matches internal IDs to coordinate-wise closest nodes in a complete OSM network of Berlin. Therefore, we provide our own matching from OSM street segment IDs to OSM node IDs in the drive network graph.

Matching For each data entry, we check whether start and end points of the segment are nodes in the drive network. If both are contained in the OSM graph and they share an edge (u, v), we assign s_{uv}^{wh} for the respective hour h and weekend or not w of the data entry. In case both nodes are part of the graph but they do not share an edge, this usually means that there is some interstitial node on the street. These are nodes which are not part of any intersections. Therefore, we update correspondent speed attributes for all edges on the shortest path from OSMnx. Unfortunately, these two cases only cover around half of the data entries. For the rest, we get coordinates of the affected nodes from the complete OSM graph of Berlin and match them to the closest edges in the drive network. Still, around 7 % of the data points cannot be matched. This is mostly due to deprecation of nodes or changes in the road network such as construction. Moreover, for some speed attribute s_{uv}^{wh} multiple data entries exist, for example on different dates. In that case, we take the mean of all speeds recorded for the respective speed attribute. Since the unsimplified drive network of Berlin with 211,838 nodes and 385,194 edges is incredibly large, we simplify the network in the next step to improve computation times.

Simplification We observe that the dataset for Berlin is sparse and disconnected. This is the case even though we already aggregate the data for days during the week and days on the weekend. Data in the center of Berlin is a little less sparse covering approximately 14 % of edges in comparison to only around 3.9 % edge coverage for the entirety of Berlin. Therefore, we limit our road network to a polygon which contains the center of Berlin up to ring roads. Since there are only main roads and residential streets in the center of Berlin, the inclusion of circular motorways allows for more diverse solutions. The used polygon is marked by a dashed line in Figure 6.1. This reduces the graph to 58,518 nodes and 96,989 edges, which is roughly only a quarter of edges from the entire graph. We now use the OSMnx simplification module that decreases the size of the graph even further by removing interstitial nodes. These nodes only add irrelevant information to chromosomes as individuals would have to visit these nodes anyway when traversing the corresponding edge from one intersection to another. Theoretically, this simplification could have been triggered before node matching. However, the reduced number of nodes would make matching more difficult and less accurate. Hence, we simplify the graph after matching, reducing it to 6,557 nodes and 15,928 edges. The OSMnx simplification method merges edges by creating lists when different values are encountered for the same attribute. We simply take the mean of lists for speed attributes. Unfortunately, some interstitial nodes include traffic signal information. Therefore, we save the number of traffic signals for each edge before the simplification and reinsert it on the closest edge afterwards. The direction of the edge, on which the information is inserted, depends on the assumption that traffic lights are close to the end node of an edge. Note, however, that this assumption may not hold for other countries, such as the United States of America. Additionally, we delete multiples of edges between node pairs since edges in our road network must be uniquely identifiable by a pair of node IDs. The deletion affects some parallel edges but also alternative streets such as U-shaped roads. Lastly, we delete dead ends, meaning nodes with no ingoing or no outgoing edges. These nodes hinder optimisation and could only be used as start or end points of routes anyway. Since there are still many unassigned street attributes in our simplified graph, we interpolate in the next step.

Interpolation To calculate missing speed attributes, we cannot simply take the average of speed attributes from neighbouring edges since speeds may vary,

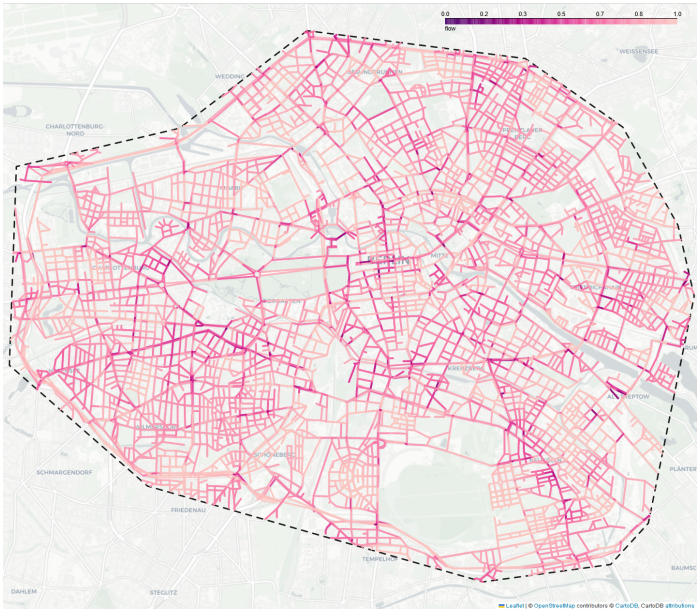


Figure 6.1.: Flows of the interpolated road network for 8 pm (median average flow) during the week. The dashed line marks the polygon in which speed data was used

for example due to inhomogeneous speed limits. Only interpolating streets with the same street type or speed limit also does not work. The reason for this is that for some street types, particularly residential roads, there is almost no data available. Instead, we interpolate all speed attributes per hour and per weekend categorisation using, what we call, flow. The flow ρ_{uv}^{wh} of an edge (u, v) at hour h with weekend categorisation w is computed using Equation 6.1 where $s_{uv}^{\max}$ is the maximum allowed speed on edge (u, v).

$$\rho_{uv}^{wh} = \frac{s_{uv}^{wh}}{s_{uv}^{\max}} \tag{6.1}$$

Speed attributes, to which no entry in the dataset was matched, are assumed to have 100 % flow in the beginning of the interpolation. For each edge, we only interpolate the flows from ingoing and from outgoing edges to preserve speeds for varying directions. To update speed attributes, we just multiply flow ρ_{uv}^{wh} by the speed limit $s_{uv}^{\max}$. Speed attributes which were assigned using the dataset are fixed which means they are not updated. The interpolation runs through the set of edges E in iterations until speed attributes converge. However, convergence for interpolating $2 \cdot 24$ speed attributes for all edges is very slow. Therefore, we only update speed attributes if the difference is greater

or equal to 0.01. For the Uber Movement Speeds dataset for Berlin, this results in an interpolated network graph for which flows are visualised in Figure 6.1. Flows in the graph are shown based on speed attributes at 8 pm during the week, with the average flow in the graph being the median.

6.1.2. Dataset discussion

The two main advantages of the Uber Movement Speeds dataset are that it is publicly available and in an easily machine-readable CSV format. Unfortunately, matching the provided OSM IDs to ones in an OSM drive network is not trivial. The biggest limitation of this dataset, however, is the sparseness of the data. Especially for smaller roads such as residential streets, speeds are missing. To compensate for the lack of data, aggregating speeds of different weekdays and interpolating becomes necessary. For this reason, most speed attributes do not directly come from entries in the dataset but are estimates based on aggregation, averaging and interpolation. Moreover, our interpolation assumes 100 % flow for speed attributes not in the dataset. Future work needs to assess whether this is a valid assumption. These aspects likely mean that the resulting road network is not a realistic representation of Berlin's streets in January of 2020. Moreover, one could even question whether the speeds provided by the dataset themselves are an accurate reflection of reality. For example, it could be possible that the dataset is biased since Uber drivers might use expert knowledge to avoid frequently congested streets by themselves. That would also mean that the assumption of free flow on roads without data would be inaccurate in some cases. Another aspect which is unclear from the dataset is how much traversal speeds are affected by waiting times at traffic signals and signs. In some cases, the added traffic signal penalty of 20 seconds could already be included. Another abnormality of this dataset is that a fraction of speed attributes exceed the respective speed limit. Uber Movement assert that no conclusions about road safety can be drawn from this [63]. They argue that speeds are not instantaneous but instead calculated as segment lengths divided by traversal times. In conclusion, these potential problems have to be investigated further in future work to assess whether this dataset is a good basis for time-dependent travel time estimations. Nevertheless, the speed data does provide some insights into travel times and congestion, which is why we used it.

In general, using real-world data comes with more difficulties compared to

assigning, for example, stochastic distributions of travel times to edges or to entire routes which is often done in related work on travel time variability. Aside from the additionally necessary preparation of real-world data discussed in the previous subsection, the main complications stem from the discreteness of travel times. While stochastic distributions are continuous, real-world data is mostly recorded in larger time steps. Hourly travel times lead to problems with our travel time variability objective where routes would take detours until the next hour to minimise travel time variability of certain edges. Therefore, our objective functions need to interpolate again between hourly travel times.

6.2. Knowledge extraction

This section examines RQ2 which questions what knowledge can be gained from applying our innovization variant to the Berlin Uber Movement dataset. To extract knowledge which is representative for the whole family of routing problems in Berlin, we compare classes of innovization experiments for different sets of route planning problems:

1. medium-length routes at different times of the day during the week
2. medium-length routes during the week versus on the weekend
3. long versus short routes

The classification of routes by length is based on the polygon in which we use speed data from Uber Movement. A circle with a 10.5 km diameter around the Deutscher Dom, a central place in the polygon, includes most of the polygon. Based on this, routes are separated into three classes according to the straight line distance between their origin and their destination:

- *short:* distance $\leq$ 3.5 km
- *medium-length*: 3.5 km $<$ distance $\leq$ 7 km
- *long:* distance $>$ 7 km

As described in our innovization for route planning approach, each experiment should return a decision space clustering, a plot of objective values of clusters and correlation coefficients between different route characteristics and objective functions. However, some experiments only output a single optimal solution or just one cluster. As explained in chapter 5, we selected percentages of street

type of experiment	travel time	travel time variability	degrees of turning
during the week, rush hour	0.620	0.014	-0.680
during the week, medium flow	0.645	0.202	-0.617
during the week, best flow	0.687	0.173	-0.737
on weekends	0.762	0.280	-0.736
short	0.421	-0.097	-0.459
long	0.605	0.040	-0.740
all	0.672	0.152	-0.687

Table 6.1.: Median Spearman correlation between main roads percentage and each objective for different types of route planning problems

types and number of traffic signals along a route as route characteristics. We analyse their correlation among each other and with each of our objective functions. Since the median correlation between main roads percentage and number of signals across all experiments was relatively high at 0.817, we only consider the main roads percentage in the analysis. For the latter characteristic, the Spearman correlations with each objective for different types of experiments are listed in Table 6.1. Overall, we see a relatively strong positive correlation of 0.672 in relation with the first objective and a relatively strong negative correlation of -0.687 in relation with the third objective. However, for some types of experiments these correlations are considerably weaker.
The different classes of experiments listed at the beginning are analysed in detail in the following two subsections. After reporting our findings, we evaluate some additional experiments on excluding our second objective from the optimisation in subsection 6.2.3. Finally, we formulate innovized principles to use for algorithm improvement based on our findings.

6.2.1. Medium-length routes

The first two sets of experiments use a representative set of ten medium-length routes in Berlin. Their shortest paths are visualised in Figure 6.2. Their

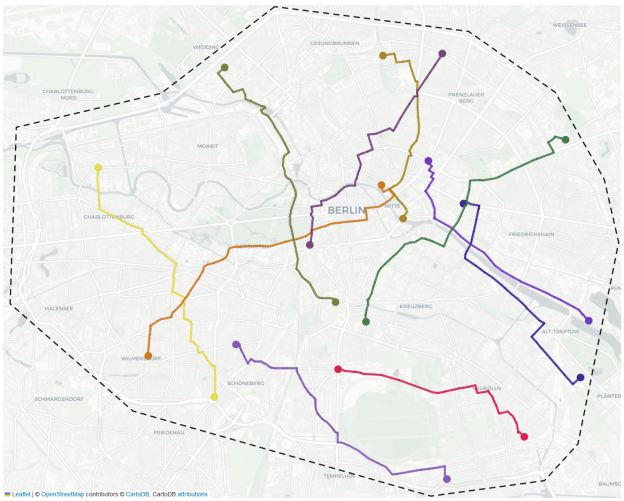


Figure 6.2.: Shortest paths of ten medium-length routes used in the innovization experiments

formal problem definitions with OSM start and end point indices and the route numbering are detailed in Table A.1 in Appendix A. We will look at individual results of route planning problems at different times of the day during the week and at medium flow on weekends before discussing knowledge extracted from comparative maps. Comparative maps visualise which solutions overlap and which do not for the different times of the day and during the week versus on the weekend.

Different times of day We select three different times of the day during the week for the first class of experiments on medium-length routes. The selection is based on the best, median and worst average flow during the week in our network. Rush hour is at 4 pm with an average flow of 75.3 %, best flow is at 4 am with 97 % flow. Medium flow is between 7 pm and 8 pm. We randomly select 8 pm as the time with medium flow at 85.5 %. For all times of the day, the majority of Pareto-optimal solutions more or less follow the straight line from start to end point. An example of this can be seen in Figure 6.3a. This knowledge is already exploited in our original MOEA since we use guided random walks as initialisation. However, the guiding in our current initialisation only loosely follows the straight line. A subset of initial populations could be created by closely following the linear path.
At best flow in the network, there are some exceptions as in Figure 6.3b and

(a) R9 at rush hour (b) R10 at best flow

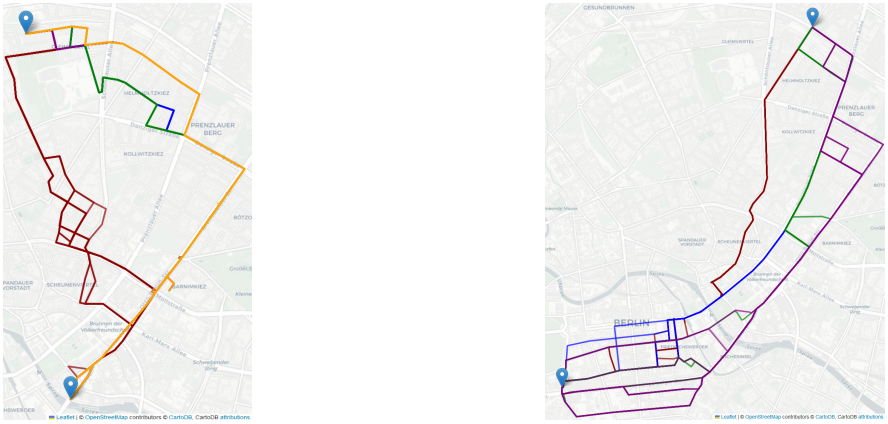
(c) R6 at best flow (d) R3 at medium flow (e) R7 at medium flow

(f) R7 at best flow (g) R1 at best flow (h) R7 at rush hour

Figure 6.3.: Decision space clustering for various medium-length problems during the week. Different clusters have different colours but routes classified as noise are not included. The R-numbering refers to the route numbering in Table A.1 in Appendix A

in Figure 6.3c where solutions take relatively long detours to use main roads or motorways to achieve lower degrees of turning. However, this usually also leads to longer travel times, which is unsurprising since the routes with detours are longer. The same trade-off can be observed at medium flow, for example in Figure 6.3d. Notably, there is one case at medium flow where the exact opposite compromise is achieved. For Figure 6.3e, many solutions closer to the straight line between start and end point have higher travel times and lower ease of driving. The reason for this is that clusters closer to the linear path can also follow a main road but are closer to the city centre where the flow is worse. Nevertheless, solutions taking detours away from the straight line path via main roads or motorways provide important trade-off options. Innovized principles should encourage the generation or the discovery of this kind of routes.
Another observation that can be made at best flow is that not all compromise solutions might be useful to a decision maker. In Figure 6.3f, most clusters which take longer detours are worse on average for both the first and third objective but exhibit a slightly better travel time variability. Similarly, half of the clusters in Figure 6.3g, that take the more eastern routes, have similar average travel times to the rest of the clusters but insignificantly lower travel time variabilities. The differences in travel time variability here are between 0.1 and 0.4 seconds. Everyday drivers are unlikely to care about this minor trade-off. As an innovized principle, we could implement a different dominance criterion for the second objective so that solutions, which are only better for travel time variability, are dominated.
During rush hour, there is one route planning problem depicted in Figure 6.3h where none of the solutions follow the straight line path. This is because it is dominated by more eastbound options with higher main roads percentage and therefore lower degrees of turning or lower travel time variability.

During week versus on the weekend In the second class of experiments, we run our innovization method for the medium-length routes depicted in Figure 6.2 during the week and on the weekends at medium flow. Medium flow is at 8 pm during the week which is why we also choose this time for experiments on weekends. The runs during the week are the same as those for medium flow in the previous paragraph and do not need to be rerun or separately analysed again.
The majority of Pareto-optimal routes follow the straight line between start and destination. Interestingly, there are a few clusters that make an exception

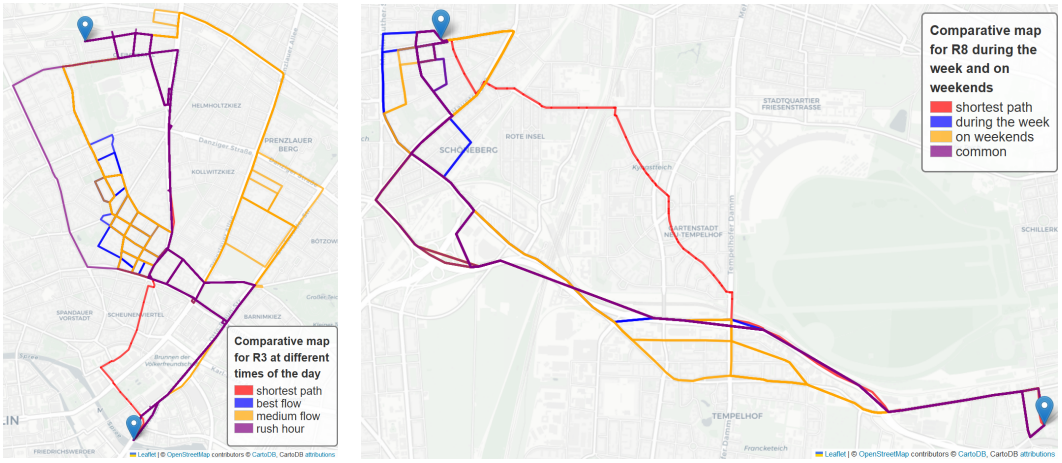


(a) R3 (b) R7

Figure 6.4.: Decision space clustering for various medium-length problems on weekends. Different clusters have different colours but routes classified as noise are not included.

from this to achieve different combinations of function values. In Figure 6.4a, only one cluster more or less follows the straight line route. The other clusters take more eastern routes which result in a slightly higher travel time but an insignificantly lower travel time variability. This is most likely not in the interest of a decision maker. In other cases like in Figure 6.4b, straying from the linear path yields a more meaningful trade-off. Most clusters in Figure 6.4b take detours using main roads for higher travel times but lower degrees of turning. This is supported by the Spearman correlations in Table 6.1 which are 0.762 on weekends between the main roads percentage and the first objective and -0.736 for the third objective. These compromise routes should be discovered more easily with our innovized principles.

Comparative maps For further analysis, we created comparative maps for each medium-length route for the three different times of the day during the week and for medium flow during the week versus on the weekend. Examples are shown in Figure 6.5. Generally, almost no solutions overlap completely. Only a median of 0.36 % overlap at all three times of the day and only a median of 2.7 % both during the week and on the weekend. The overlap between sets of edges used in solutions on the other hand is larger. Some patterns are noticeable for the travel times of common routes. Unsurprisingly, the travel time increases as the flow worsens during the week. There is an increase of

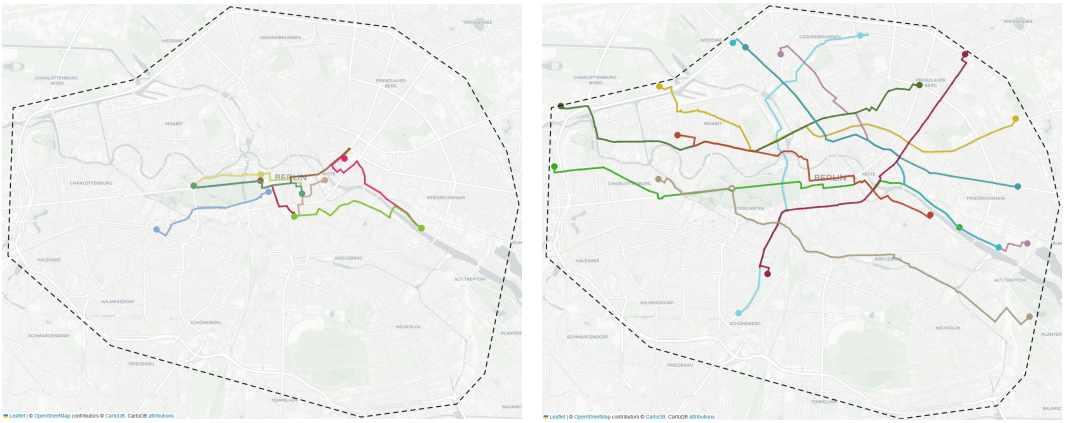


(a) R3 at different times of the day

(b) R8 during the week versus on weekends

Figure 6.5.: Two exemplary comparative maps depicting sets of solutions at different times of the day (a) or on the weekend versus during the week (b), common routes and the respective shortest path.

9.0 % in travel time from best to medium flow and another 10.5 % from medium to worst flow. Moreover, the travel time for common routes is 5.4 % better on weekends. Interestingly, the travel time variability does not follow the same pattern and differences in values are negligibly small. The degrees of turning are independent of departure times. Overlapping solutions are the ones that follow the straight line path the closest. They are almost identical to the respective shortest path with a median discrete Fréchet distance δ_{dF} of 0.175 in both cases. We considered the inclusion of the shortest path of a route planning problem in the initial population as an innovization. However, we believe that this would lead to the algorithm getting stuck in a local optima since the shortest path is initially significantly better than individuals generated by guided random walks. The shortest path is even misleading in some cases where all solutions deviate from it. For example in Figure 6.5a, no route follows the shortest path in the beginning since it uses edges with bad flow. In Figure 6.5b, no solutions match the middle part of the shortest path. Instead, routes take the motorway which has a similar flow but allows a higher maximum speed.

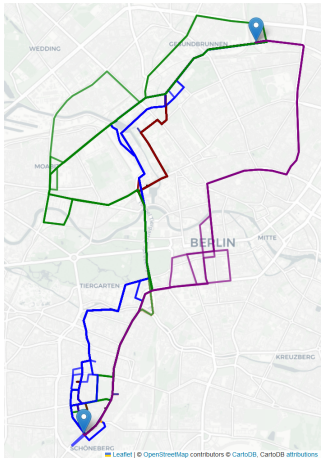


(a) Short routes (b) Long routes

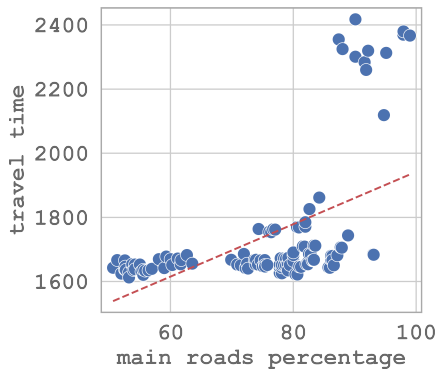
Figure 6.6.: Shortest paths of ten short (a) and ten long (b) routes used in the innovization experiments

6.2.2. Short versus long routes

In this section, we share the results of the last class of innovization experiments on short and long routes during the week at medium flow. We randomly selected a set of ten representative problems each. Their shortest paths are shown in Figure 6.6. Their formal problem definitions with OSM start and end node indices and the route numberings are listed in Table A.2 and A.3 of Appendix A. Start and end points of short routes are randomly chosen from a list of points of interest in the centre of Berlin based on [59]. The short route planning problems are generally easy to solve. Two of them return only a single optimal solution and four others output only one decision space cluster. In the other cases, most clusters have higher average travel times and lower average degrees of turning, the higher their average main road percentage is.
The route planning problems with long routes are generally harder and produce more Pareto-optimal solutions and more clusters. Overall, we observe a strong correlation between clusters' average main road percentage and travel times and degrees of turning respectively. This is consistent with the median correlations over all solutions in Table 6.1. Interestingly, we can see that the correlation with travel times can additionally be grouped by length. In Figure 6.7 for example, we can see that two clusters, which take longer detours, are separate from the trend between main roads percentage and travel times. Note that on the left of this example almost all clusters have to go through a bottleneck which is the Tiergarten Spreebogen tunnel. While this is interesting for this particular



(a) Long R1



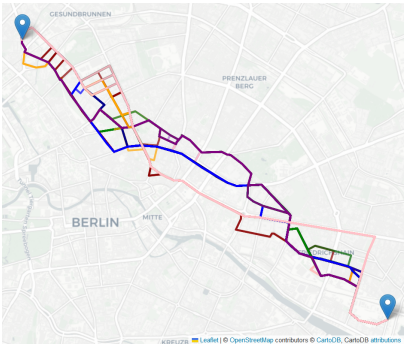
(b) Relationship between main roads percentage and travel time for all individuals from (a)

Figure 6.7.: (a) Decision space clustering of a route planning problem where most routes have to use a bottleneck. The R-numbering refers to the route numbering in Table A.3 in Appendix A. (b) Relationship between main roads percentage and travel time for the same route planning problem. Individuals in the green and the purple cluster in (a) have higher travel times separating them from the rest of the trend

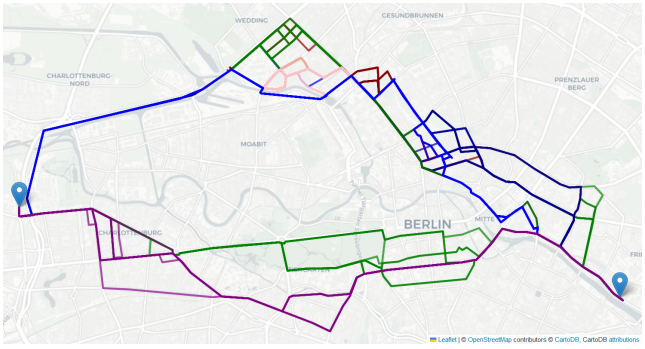
routing problem, it is not useful for other route planning problems in Berlin as mentioned above.
Many solutions such as in Figure 6.8a somewhat follow the straight line from start to destination like we have seen in other classes of experiments. Nevertheless, many clusters accept longer detours to use main roads or motorways. In some cases such as in Figure 6.8c, this is true for all clusters. Figure 6.8b illustrates that detours for long routes can be much longer than for short or medium-length routes. This should be taken into account when formulating innovizations in subsection 6.2.4.

6.2.3. Exclusion of travel time variability

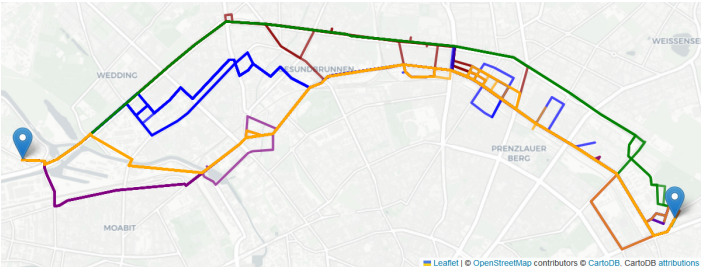
While analysing the results of our innovization experiments, we noticed that our second objective, travel time variability, returns relatively low values. As shown in the middle column of Table 6.2, the maximum function value of Pareto-optimal solutions from all experiments is 2.217 seconds. Furthermore, solutions



(a) Long R3



(b) Long R4



(c) Long R2

Figure 6.8.: Decision space clustering for various long route planning problems. Different clusters have different colours but routes classified as noise are not included

	with 2nd objective	without 2nd objective
minimum	0.0	0.0
maximum	2.217	2.055
maximum median	0.653	0.701
median standard deviation	0.188	0.147
median number of solutions found	36.0	8.5

Table 6.2.: Statistics for second objective function values of Pareto-optimal solutions in seconds and for number of solutions with the second objective included in the optimisation and without it

return similar travel time variabilities with a median standard deviation of only 0.188 seconds. In our scenario, an everyday driver probably does not care about a risk of being roughly two seconds late in the worst case. Moreover, some optimisations find routes which are worse in both the first and third objective but only insignificantly better in travel time variability. These routes are not really relevant for drivers and only hinder their decision making process. For these reasons, we perform some additional MOEA runs where we use the same set of experiments but leave out the second objective during optimisation. Afterwards, we compute travel time variability function values for the Pareto-optimal solutions. We aim to determine whether excluding the second objective negatively affects function values of solutions. The right column in Table 6.2 shows that the minimum and maximum travel time variability function values, the median maximum value as well as the median standard deviation only slightly change when excluding the second objective from the optimisation. Furthermore, a two-tailed Mann–Whitney U test results in a p-value of around 0.8 which shows that the change in worst case maximum values is not significant between both sets of experiments. Therefore, this objective can be left out. However, it seems that the objective helps to preserve lateral diversity during the optimisation since the average hypervolume decreases by 912.16 in comparison to the original RQ2 experiments. Some diversity preserving mechanism should be included in our innovizations in the following subsection to counteract this phenomenon.
Interestingly, the additional MOEA runs also show that the travel time variability objective is responsible for small protrusions in routes that are clearly visible, for example in Figure 6.8c. It seems that it is more reliable, for instance, to make a right turn and a U-turn to get back on the same road instead of just crossing straight through an intersection.

6.2.4. Innovized principles

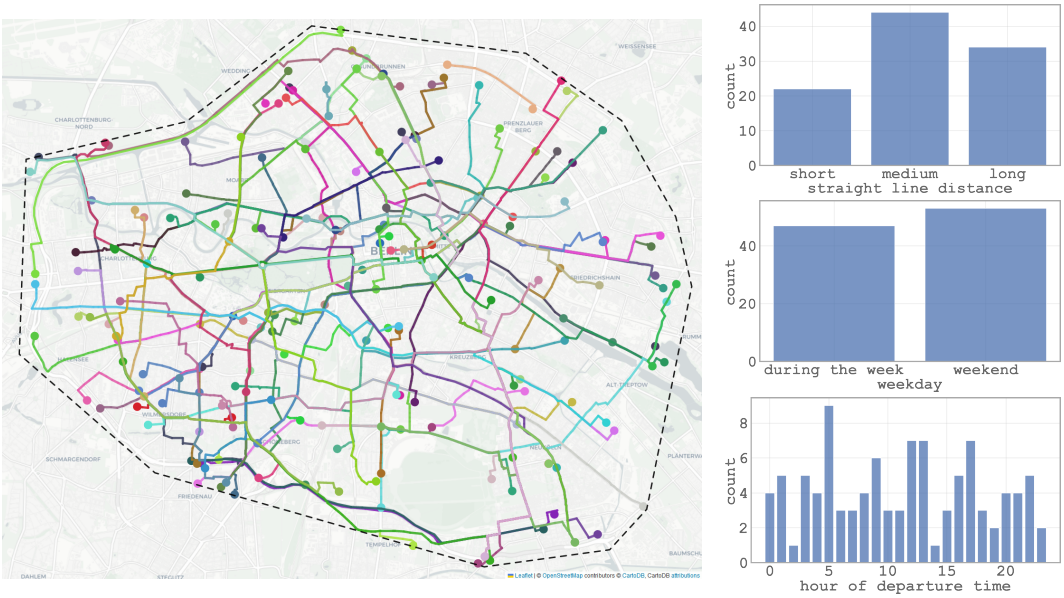
From the results of these innovization experiments, we can extract four principles which can be reused in future route planning optimisations.
When analysing the correlation between the main roads percentage and our objectives in Table 6.1, we generally see a positive correlation with the travel time and a negative correlation with the turning degrees. Pareto-optimal routes can usually be separated into two groups, for which these trends can also be observed. The first closely follows the straight line from start to destination and is often similar to the shortest path. This group exhibits lower travel

times but higher turning degrees. The opposite normally holds for the second group which uses a high percentage of main roads and motorways. Considering the first group, which uses more residential roads, another objective could be added in the future to balance the comfort of drivers and residents. In some experiments, only one of the two groups exists, such as for Figure 6.3h, or both groups overlap. An overlap occurs, for example, if the straight line routes mostly use main roads as in Figure 6.3a. Otherwise, the routes in the second group might take detours to use main roads and motorways. We observe that the longer the route is, the longer are the possible detours. With this knowledge, we can adapt the generation of the initial population. Originally, the entire population was initialised using guided random walks where the edge in closest direction to the destination is chosen with a probability of 75 %. Now we split the initialisation in half. The first half is still created using guided random walks but the individuals should follow the straight line more closely. Therefore, the probability of going in direction of the goal is increased to 98 %. Nonetheless, we do not seed the initial population with the shortest path since the optimisation then gets stuck in local optima more easily. We also no longer insert the shortest path as a replacement, if the generation gets stuck. The other half of individuals prefer main roads and motorways. When choosing the next edge, the outgoing edges are sorted first by whether they are main roads or motorways, and second by their orientation towards the destination. The probability of choosing the most preferred edge depends on the categorisation of a route by length defined in the beginning of section 6.2. The probability is 95 % for short routes, 90 % for medium-length routes and 85 % for long routes. This way, we allow longer detours for longer routes. However, since some destinations are only reachable via residential roads, we switch to guided random walk as in the first half, when the destination is less than 250 m away. Nonetheless, some paths still get stuck in a loop during generation. If a path is five times the length of the shortest path from start to destination, we finish it by inserting the shortest path from the current node to the destination. This prevents excessively long, initial routes.

An analysis of function values of the travel time variability objective show that leaving the second objective function out of the optimisation has virtually no impact on the travel time variability of the Pareto-optimal solutions found. The worst recorded function value is a travel time variability of around two seconds either way. Since everyday drivers are unlikely to be concerned about a risk of being a couple of seconds too late, we can leave the second objective function

out of the optimisation. Additionally, the first or the third objective could be omitted for some of the route planning problems. For example, there is a relatively strong negative median Spearman correlation of -0.857 between these objectives for long routes. However, we did not utilise this since the correlation is not strong enough and routing problems are deceptive. This means that parts of a Pareto front may not be discovered if one of the objectives out is omitted because routes with similar fitness values can be relatively far apart in the decision space. If the correlation was higher, either the first or the third objective function could have been left out for a trade-off between computational effort and quality of Pareto-optimal solutions.
Even leaving only the second objective function out, makes the route planning problems less complex and easier to solve. As we can see in Table 6.2, the number of Pareto-optimal solutions found by our EA is reduced to less than a quarter of Pareto-optimal solutions for the same optimisation problems with the second objective. For this reason and to avoid unnecessary function evaluations, we decrease the populations size from 2.5 times the shortest path rounded to the nearest ten to 1.5 time the shortest path rounded to the nearest ten.
Exemplary experiments have also shown that optimisations easily get stuck in local optima without the travel time variability objective. Getting stuck in local optima is a known problem for NSGA-II. While the algorithm has mechanisms to ensure diversity along the Pareto-optimal front, the selection mechanisms destroy lateral diversity across fronts since only the best individuals are chosen for reproduction and survival. To mitigate this problem, NSGA-II with controlled elitism [16] is used. A disadvantage of this algorithm is that another hyperparameter, the reduction rate r, has to be set. In our case, setting this parameter $r = 0.2$ has worked well in exemplary tests. The same tests revealed a higher mutation probability of 35 % instead of 25 % for optimisations on the weekend as favourable.
In conclusion, the following four innovizations are applied in the experiments for RQ3:

- Update of generation of initial population
- Exclusion of second (travel time variability) objective
- Decrease in population size
- Increase in exploration through controlled elitism and higher mutation probability on the weekends



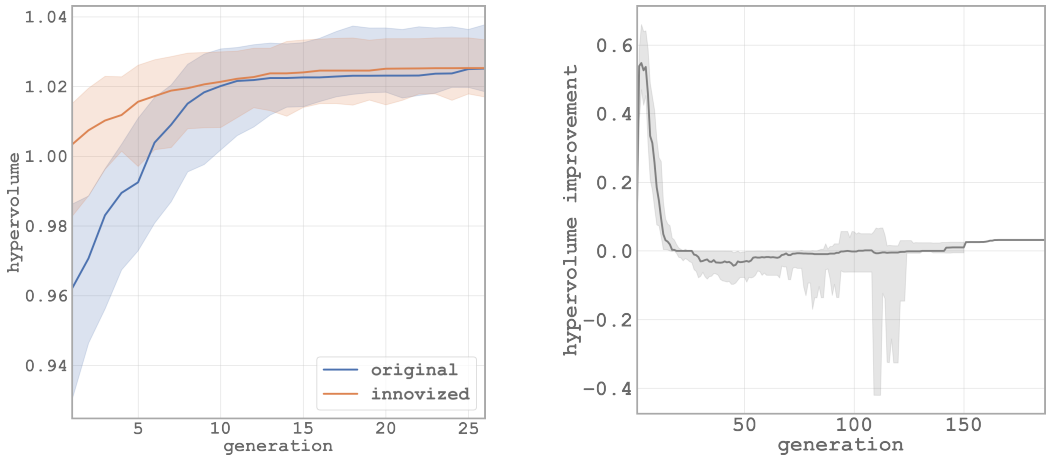
(a) Shortest paths

(b) Parameter distribution

Figure 6.9.: Map with shortest paths (a) and distributions per different categories (b) of the 100 route planning experiments for RQ3

6.3. Algorithm improvement

As stated in our third research question, we now want to examine whether the innovized principles from the previous section can improve the efficiency or the results of our MOEA. We compare our route planning MOEA from section 5.1 to an adapted version which implements the innovized principles. From now on, we will be referring to the former as original MOEA and to the latter as innovized MOEA. Both algorithms are run on 100 randomly generated route planning problems. The parameter settings of these experiments are listed in Appendix B. A map of the shortest path and the distribution of experiments among different randomised parameters can be found in Figure 6.9. The randomised parameters are relatively balanced with a slight overrepresentation of medium-length routes and problems with a departure time between 5 am and 6 am. To evaluate the third research question, we compute efficiency, quality of the final results and total number of function evaluations for both algorithms. The efficiency measures the quality of solutions as the hypervolume after a certain number of generations. To evaluate the final results, we measure hypervolumes after each run terminates. The termination criterion is the same as for the original MOEA



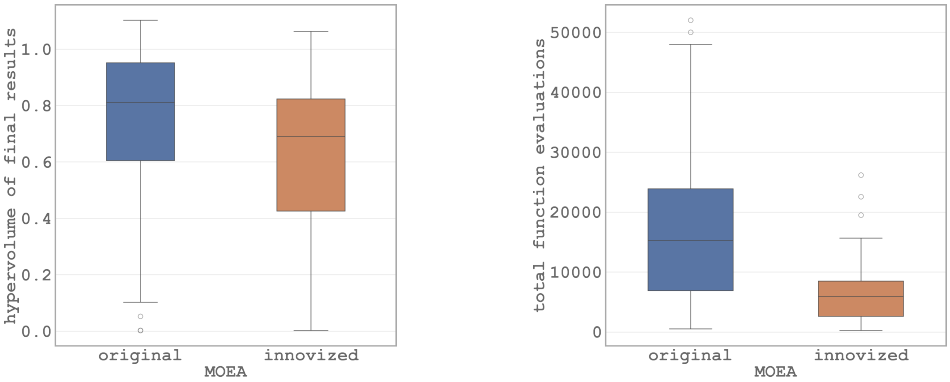
(a) Median hypervolume

(b) Median hypervolume improvement

Figure 6.10.: Evolution of the median hypervolume (a) and the median hypervolume improvment (b) over generations for all experiments. Translucent areas show the respective 95 % confidence intervals

which stops an algorithm if there is no significant change for 25 consecutive generations or after 200 generations at the latest. Since the knowledge extraction in the previous section revealed that the travel time variability is not useful in an everyday routing use case, we omit the second objective from the optimisation in the innovized MOEA and from any hypervolume computations. The original MOEA still uses travel time variability in its fitness function but we compute the set of non-dominated solutions before any hypervolume calculations after leaving out the second objective.

First, we compare the efficiency of both algorithms. For this, we only evaluate hypervolumes up to the 26th generation because that is the maximum number of generations computed for all runs. We can see the development of the median hypervolumes over generations for all experiments for both algorithms in Figure 6.10a. Figure 6.10b shows the evolution of the median hypervolume improvement. We compute the hypervolume improvement as the difference in hypervolume from the original MOEA to the innovized MOEA for each experiment in every generation. As we can see in these figures, the innovized MOEA starts with significantly higher hypervolumes in comparison to the original MOEA. A median hypervolume improvement of about 0.11 after the first generation implies a positive effect of the new initialisation. Afterwards, the median hypervolume of the innovized MOEA increases until about the



(a) Hypervolumes of final results

(b) Total function evaluations

Figure 6.11.: Normalised hypervolumes of final results per algorithm (a) and total number of function evaluations needed to achieve the final results (b)

10th generation. The median hypervolume of the original MOEA converges towards the median hypervolume of the innovized MOEA. After 19 generations, the median hypervolume improvement stagnates at 0.0 for a few generations. After the 26th generation, the median hypervolume improvement is 0.0 which means that, according to our definition, the efficiency of both algorithms is the same. The median hypervolume improvement then worsens and before it increases again. However, this gradient is questionable since experiment runs finish after different generations. This is due to our stopping criteria which allow earlier termination based on changes of multiple indicators. Therefore, we compare the final results from all runs instead.

Figure 6.11 shows the normalised hypervolumes of the final results and the total number of function evaluations needed per algorithm. In Figure 6.11a, we can see that hypervolumes of final results are better when using the original MOEA. The median hypervolume of the original MOEA is around 0.81 while the one for the innovized MOEA is worse at 0.69. Both algorithms also produce extremely low hypervolumes. When both algorithms return only a single optimal solution, the hypervolume improvement is 0.0. A two-tailed Mann-Whitney U test with significance level of 0.01 results in a p-value of 0.0002, indicating that the difference in hypervolume is significant. However, if we compare hypervolumes for each experiment separately, the difference between both algorithms is smaller with a median hypervolume improvement of only about -0.01. This means that although the difference in hypervolumes is significant, the quality of final results

per routing problem is not substantially worse.
Remarkably, Figure 6.11b demonstrates that the number of total functions evaluations for runs of the innovized MOEA is much lower than that of the original MOEA. Median total function evaluations of 5,909 for the innovized MOEA are only 38.6 % of the median from the original MOEA at 15,300. Likewise, when comparing the sum of total evaluations for all experiments, the innovized MOEA with 643,239 evaluations needs only about a third of the original MOEA with 1,801,385 evaluations.
To see what results we would get if we did not exclude the second objective, we run our RQ3 experiments on a third variant which is the same as the original MOEA except for the innovized generation of the initial population. For this version, the median hypervolume improvement in comparison to the original MOEA is 0.0002. The total number of function evaluations is $1,926,124$ which is slightly higher than those for the original MOEA.
In conclusion, we have seen that an improvement of the original MOEA can be achieved with the innovized principles. While the efficiency remains the same, the final results are not substantially worse and the innovized MOEA requires only approximately a third of the function evaluations. That means a similar quality of solutions can be produced with far less computational effort. This is particularly valuable for route planning problems where many optimisations run, for example, on mobile devices that typically have limited computational resources. Alternatively, a better quality of final results can be achieved when using a variant of the original MOEA with only the innovized initialisation of populations. However, this results in a slightly higher number of function evaluations. Furthermore, a positive effect of our innovized initialisation has been observed in form of better hypervolumes for earlier generations. Future work could improve operators and exploration mechanisms of the MOEAs to make better use of the advantage created by the innovized initialisation.

6.4. Discussion

In this chapter, we successfully extracted knowledge from route planning problems using our proposed innovization for route planning. Moreover, the extracted knowledge helped us improve our original MOEA by reducing the computational effort. Since we set hyperparameters of the MOEAs to values that proved to be reasonable during exemplary test, it could be argued that an algorithm improvement could also be attained by a hyperparameter optimisation. Nevertheless,

this would improve both algorithms which would result in a level playing field again. However, performing the hyperparameter optimisation before applying our innovization variant, may improve the results from the multi-objective optimisation step. In spite of this, we argue that this does not change the end results of our innovization significantly because any shortcomings in the first, multi-objective step are compensated by the improvements from single-objective optimisations and PMM-LS. For example, better solutions from the first step might only mean that we discover fewer new extreme solutions during the single-objective runs or that the PMM-LS needs fewer iterations. Since hyperparameter optimisations themselves are time-consuming, we recommend doing a hyperparameter optimisation to speed up the process if innovization for route planning is frequently used or if the employed MOEA is intended to be reused.

Another point of discussion is the travel time variability. One of our innovized principles excludes this objective since its values are negligibly small. However, we hypothesise that this is not due to the design of the objective but rather due to the nature of the dataset used. The Uber Movement Speeds dataset only contains a relatively small subset of recorded speeds from all edges at all hours on every day of the week. That is why we averaged speeds during the week and on weekends, and interpolated the data for any missing edges or times. Because of this, peaks in the data are levelled out and edges with high travel time variability are already avoided due to high travel times. We still believe that the travel time variability objective is useful in combination with a more detailed dataset or a sophisticated travel time prediction model. This may be especially true for certain use cases such as for emergency services. Even the small values we deem negligible for everyday drivers could make a crucial difference when emergency vehicles are trying to reach their destination.

Leaving out the second objective moreover leads to problems with the comparability of the original and the innovized MOEA. We exclude the second objective from all hypervolume calculations, even though the function values for the second objective could be computed after running the innovized MOEA. When including the second objective in the hypervolume computation, there are some cases where the original MOEA has a significantly better hypervolume due to a better diversity in its front. However, the solutions, that lead to the significantly better diversity in these cases, are also the solutions that are dominated when excluding the second objective. As we have already mentioned, these additional options are more of a hindrance than a help to decision makers because they

only have a slightly better travel time variability but are worse for all other objective functions. This is why we believe that omitting the second objective from hypervolume computations is more reasonable. Nevertheless, a different dataset might make the travel time variability a useful objective so that it would make sense to include it in calculations, or innovized principles would not exclude it in the first place.

7. Conclusion

In this thesis, we presented innovization for route planning which is an adapted version of the original innovization. To this end, we introduced PMM-LS, a local search method for routing problems. This novel search method systematically explores the neighbourhoods of routes by generating alternative routes. PMM-LS was able to achieve a median normalised hypervolume improvement of 0.018 in our experiments. Additionally, we designed a routing MOEA and described a detailed analysis step.

Using innovization for route planning, we successfully extracted four innovized principles from time-dependent route planning problems. One of the major discoveries was that Pareto-optimal routes can typically be separated into two groups. The first group consists of routes that are close to the linear path from start to end point. These solutions usually exhibit lower travel times but higher degrees of turning. The second group is a set of longer routes that take faster roads and that have higher travel times but lower degrees of turning. This knowledge was used to update the generation of the initial population. The other main finding was that the travel time variability objective can be omitted. The reason for this is that the differences in travel time variability between solutions are insignificant for decision makers. We suspect that the overall small values are due to limitations of the chosen dataset. In future, the impact of the travel time variability objective should be re-evaluated on a dataset that has sufficient speed data for a road network graph.

Lastly, we have shown that we are able to improve a MOEA using extracted knowledge. While the efficiency remained the same in our experiments, the quality of the final results was not substantially worse and we managed to drastically decrease the number of necessary function evaluations to one third. That means similar quality solutions can be produced with far less computational effort using the knowledge extracted with our innovization for route planning. This is particularly valuable for applications, where limited computational resources are available, such as mobile devices which are often utilised for routing. Additionally, we demonstrated that choosing between computational effort and

quality of final results is possible by using only the innovized generation of the initial population.
In conclusion, our primary contribution is the development of an innovization for route planning and PMM-LS. Outside of academic research, our proposed methodology can be used to improve drivers' quality of life by being able to offer more personalisation and decision support with faster multi-objective routing algorithms. Moreover, the reuse of knowledge increases the sustainability of algorithms.
One limitation of innovization for route planning is its computation speed. Especially PMM-LS is slow. However, a trade-off between computation time and completeness of search is possible via the step and window size parameters. Generally, the slow computation times are not a drawback since innovizations are intended to be used only once for knowledge extraction. Nevertheless, future work could speed up the innovization process by parallelising some steps or by employing a heuristic for setting problem-specific ϵ-values for OPTICS. Moreover, only applying PMM-LS to a few well-distributed solutions as in the original innovization is possible. Additionally, PMM-LS might be further improved by other search strategies such as an integration into VNS. A verification method for the Pareto-optimality of solutions, that can handle route planning problems, would also be a beneficial extension. The data analysis step could be improved by analysing more route characteristics such as road width or the usage of one-way streets. In general, using innovization for route planning with other datasets or for differently defined route or path planning problems are interesting future research opportunities. Furthermore, automating knowledge extraction, as it has been done in related work, is a possible research area of the proposed approach.
In case our MOEA is intended to be reused for routing optimisations, improvements should be made to make use of the advantage from the innovized generation of the initial population. We recommend changing operators or exploration mechanics, or executing a hyperparameter optimisation.

A. Problem definitions of RQ2 experiments

Experiments with medium-length routes include all route planning problems $\{(\{t_{travel}, ttv, deg_{turn}\}, G, n_O, n_D, 0, t_0) \mid n_O, n_D$ as in Table A.1; $\forall\, t_0 \in \{4, 16, 20\}\}$ and $\{(\{t_{travel}, ttv, deg_{turn}\}, G, n_O, n_D, 1, 20) \mid n_O, n_D$ as in Table A.1$\}$.

route number	index of start node n_O	index of end node n_D
R1	26784424	21487259
R2	26953628	2769140801
R3	29207861	260053605
R4	28794056	1544797640
R5	2796846632	8894045297
R6	28096041	581883334
R7	26869273	61771109
R8	27543579	351868926
R9	26913884	2612436579
R10	31032157	28096372

Table A.1.: Route numbering and OSM start and end node indices for medium-length routes

Experiments with short and long routes are comprised of all route planning problems $\{(\{t_{travel}, ttv, deg_{turn}\}, G, n_O, n_D, 0, 20) \mid n_O, n_D$ as in Table A.2 and A.3$\}$.

route number	index of start node n_O	index of end node n_D	points of interest
R1	1447899974	27011222	Reichstag to Victory Column
R2	1447899974	283039346	Reichstag to Gendarmenmarkt
R3	3463621770	27011222	Brandenburg Gate to Victory Column
R4	25663420	27011222	Checkpoint Charlie to Victory Column
R5	863119413	3463621770	Berlin TV Tower to Brandenburg Gate
R6	25663420	262479941	Checkpoint Charlie to Museum Island
R7	283039346	27011222	Gendarmenmarkt to Victory Column
R8	26763015	29063088	Kaiser Wilhelm Memorial Church to Holocaust Memorial
R9	295600226	863119413	East Side Gallery to Berlin TV Tower
R10	25663420	295600226	Checkpoint Charlie to East Side Gallery

Table A.2.: Route numbering and OSM start and end node indices for short routes. Origins and destinations are based on points of interest in Berlin

route number	index of start node n_O	index of end node n_D
R1	27785303	625736296
R2	1828366924	9169023886
R3	26750963	20246171
R4	2296386929	4839921236
R5	3015494652	281814186
R6	27197252	26960746
R7	28253030	1410036225
R8	29269508	29318894
R9	244430254	27409724
R10	26984400	29276213

Table A.3.: Route numbering and OSM start and end node indices for long routes

B. Problem definitions of RQ3 experiments

RQ3 experiments consist of all route planning problems $\{(\{t_{travel}, ttv, deg_{turn}\}, G, n_O, n_D, w_0, t_0) \mid n_O, n_D, w_0, t_0 \text{ as in Table B.1}\}$.

route number	index of start node n_O	index of end node n_D	weekday w_0	departure time t_0
1	26765651	773318474	1	13:39:52
2	338795310	270737259	0	18:30:21
3	29063088	29785881	1	05:41:10
4	28300064	21432815	0	15:47:17
5	3131745415	26748231	0	21:33:08
6	394582779	3015494652	1	15:59:03
7	28252568	761343334	1	01:25:16
8	29276959	10073921795	0	07:09:59
9	26758719	26916065	1	05:21:59
10	26726690	26745948	1	10:57:30
11	26881976	1411289036	0	03:51:27
12	442032872	26807748	0	06:15:14
13	299937279	4505398576	0	05:06:22
14	29063251	28300100	0	02:11:35
15	1675108861	1599974865	0	22:08:11
16	26751311	7217029008	0	00:18:26
17	260053613	26738414	0	06:21:53
18	271030233	760570150	1	09:40:16
19	601409797	26646273	1	09:18:54
20	26763057	29788864	1	12:31:01
21	26822979	304510227	0	08:50:44
22	305244365	2928629712	1	23:37:46

Continued on next page

route number	index of start node n_O	index of end node n_D	weekday w_0	departure time t_0
23	26731247	26969265	0	11:48:12
24	27555119	10554821413	0	01:03:36
25	575727837	28253982	1	06:01:56
26	20246267	10648314047	0	16:11:51
27	26866793	29266243	1	00:13:47
28	697441444	38457826	1	20:50:23
29	2959830776	26704108	0	09:40:03
30	27540223	224023546	1	04:58:05
31	28254057	29271700	0	17:21:12
32	29785483	295706299	0	17:37:00
33	604944182	27484580	1	16:46:20
34	271739554	26704450	0	08:37:38
35	27785489	1864428537	0	17:41:30
36	27540475	29785878	1	22:09:04
37	26745596	26734234	1	13:50:02
38	27785150	26731180	1	14:55:33
39	26761234	1837610629	1	17:59:07
40	268523016	26751256	1	22:49:19
41	26704087	26960767	1	12:24:06
42	87828184	26754191	0	07:49:33
43	28252019	26908826	1	18:44:01
44	26952874	283035794	1	01:25:34
45	270182847	508308244	0	00:02:41
46	7702993650	26726531	1	22:53:41
47	29221583	21487242	0	05:55:23
48	664798253	727333481	0	12:14:35
49	1837885779	26784982	1	17:39:25
50	9177686437	26785760	1	03:16:19
51	243993487	26682638	1	21:23:39
52	26740510	271400982	1	22:50:15
53	26682651	26757492	1	20:57:33
54	26752863	26750492	1	12:30:51
55	1835539162	26952905	0	04:19:38
56	28097275	27787547	0	20:18:23

Continued on next page

route number	index of start node n_O	index of end node n_D	weekday w_0	departure time t_0
57	29804217	10554290231	1	13:30:53
58	29985590	1711812125	0	05:30:55
59	27186878	28794519	1	05:31:31
60	21677321	271649075	1	12:45:07
61	26749751	637672657	0	12:46:39
62	26980893	29221506	1	17:07:05
63	28302186	678901775	0	13:27:38
64	1935829068	29208253	0	19:23:53
65	26868108	29269506	1	01:20:25
66	1893660715	309345035	1	03:42:36
67	300128994	28252304	1	17:11:32
68	26750963	274977646	0	03:08:39
69	29790216	26747878	0	03:58:52
70	25663498	26749747	1	13:34:28
71	27432786	26731182	0	07:32:48
72	263703832	95728073	1	05:45:38
73	6206501300	31372375	1	11:27:10
74	31259202	697323164	0	13:51:26
75	26852202	26708477	1	18:21:09
76	31357356	26726686	0	20:14:43
77	805463430	26727583	0	09:23:54
78	3336091663	349904567	0	05:42:15
79	218739803	2212847786	1	15:41:39
80	26765068	26785770	1	13:54:59
81	2476048346	272438052	1	05:50:08
82	1666487146	26822448	1	08:33:56
83	58571017	4377815950	0	11:09:33
84	26735641	4348866539	1	09:42:36
85	27541390	359390910	0	21:57:55
86	29276014	27005148	0	10:55:44
87	31259202	28302080	0	19:05:18
88	27541390	26876583	0	04:03:32
89	34812302	287650997	0	12:53:23
90	26785772	564640084	1	16:47:42

Continued on next page

route number	index of start node n_O	index of end node n_D	weekday w_0	departure time t_0
91	29421278	26871569	1	23:01:58
92	303069220	27008019	0	08:31:44
93	324368407	416010793	0	21:16:43
94	34812302	1837885734	1	04:11:26
95	388557132	29326011	0	10:37:48
96	5450916417	10002955547	1	16:37:40
97	26731303	27011994	1	16:33:28
98	8874096289	3168358968	1	00:39:38
99	415919772	26852187	0	01:03:07
100	205312472	26731270	1	09:38:19

Table B.1.: Parameters of route planning problems for RQ3 experiments

Bibliography

[1] Faez Ahmed and Kalyanmoy Deb. Multi-objective optimal path planning using elitist non-dominated sorting genetic algorithms. *Soft Computing*, 17:1283–1299, 2013.

[2] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. OPTICS: Ordering points to identify the clustering structure. *ACM Sigmod record*, 28(2):49–60, 1999.

[3] Harold P. Benson. Existence of efficient solutions for vector maximization problems. *Journal of Optimization Theory and Applications*, 26:569–580, 1978.

[4] Julian Blank and Kalyanmoy Deb. pymoo: Multi-Objective Optimization in python. *IEEE Access*, 8:89497–89509, 2020.

[5] Julian Blank and Kalyanmoy Deb. A running performance metric and termination criterion for evaluating evolutionary multi- and many-objective optimization algorithms. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE, 2020.

[6] Stanislav Bochkarev and Stephen L. Smith. On minimizing turns in robot coverage path planning. In *2016 IEEE international conference on automation science and engineering (CASE)*, pages 1237–1242. IEEE, 2016.

[7] Geoff Boeing. OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, 65:126–139, 2017.

[8] Maurizio Bruglieri, Alberto Colorni, Federico Lia, and Alessandro Luè. A Multi-objective Time-dependent Route Planner: A Real World Application to Milano City. *Transportation Research Procedia*, 3:460–469, 2014.

[9] Nabil Al Nahin Ch, John Krumm, and Andrew Kun. Speed Anomalies and Safe Departure Times from Uber Movement Data. In *Proceedings of the 9th SIGKDD International Workshop on Urban Computing*, volume 24, San Diego, CA, USA, 2020.

[10] Ing-Chau Chang, Hung-Ta Tai, Feng-Han Yeh, Dung-Lin Hsieh, and Siao-Hui Chang. A VANET-based A* route planning algorithm for travelling time- and energy-efficient GPS navigation app. *International Journal of Distributed Sensor Networks*, 9(7):794521, 2013.

[11] C. Chitra and P. Subbaraj. A nondominated sorting genetic algorithm for shortest path routing problem. *International Journal of Computer and Information Engineering*, 4(8):1227–1235, 2010.

[12] Carlos A. Coello Coello, Kerstin Dächert, Kalyanmoy Deb, José Rui Figueira, Abhinav Gaur, Andrzej Jaszkiewicz, Günter Rudolph, Lothar Thiele, and Margaret M. Wiecek. Usable knowledge extraction in multi-objective optimization: An analytics and "innovization" perspective (WG3). *Personalized Multiobjective Optimization: An Analytics Perspective*, page 70, 2018.

[13] Bilash Deb, Salehin Rahman Khan, Khandker Tanvir Hasan, Ashikul Haque Khan, and Dr. Md. Ashraful Alam. Travel time prediction using machine learning and weather impact on traffic conditions. In *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, pages 1–8. IEEE, 2019.

[14] Kalyanmoy Deb. *Multi-objective optimization using evolutionary algorithms*. Wiley-Interscience series in systems and optimization. Wiley, Chichester, repr. edition, 2004.

[15] Kalyanmoy Deb, Sunith Bandaru, David Greiner, António Gaspar-Cunha, and Cem Celal Tutum. An integrated approach to automated innovization for discovering useful design principles: Case studies from engineering. *Applied Soft Computing*, 15:42–56, 2014.

[16] Kalyanmoy Deb and Tushar Goel. Controlled Elitist Non-dominated Sorting Genetic Algorithms for Better Convergence. In Eckart Zitzler, Lothar Thiele, Kalyanmoy Deb, Carlos Artemio Coello Coello, and David Corne, editors, *Evolutionary Multi-Criterion Optimization*, pages 67–81, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[17] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[18] Kalyanmoy Deb and Aravind Srinivasan. Innovization: Innovating design principles through optimization. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 1629–1636, 2006.

[19] Daniel Delling, Peter Sanders, Dominik Schultes, and Dorothea Wagner. Engineering route planning algorithms. In *Algorithmics of large and complex networks: design, analysis, and simulation*, pages 117–139. Springer, 2009.

[20] Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[21] Matthias Ehrgott. *Multicriteria Optimization*, volume 491. Springer Science & Business Media, Berlin, Heidelberg, second edition, 2005. First edition as Lecture notes in economics and mathematical systems, volume 491.

[22] Thomas Eiter and Heikki Mannila. Computing discrete fréchet distance. Technical report CD-TR 94/64, Christian Doppler Labor für Expertensysteme, 1994.

[23] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD-96 Proceedings*, pages 226–231, 1996.

[24] Stefano Fontanelli, Enrico Bini, and Paolo Santi. Dynamic route planning in vehicular networks based on future travel estimation. In *2010 IEEE Vehicular Networking Conference*, pages 126–133. IEEE, 2010.

[25] Maurice Fréchet. Sur quelques points du calcul fonctionnel. 1906.

[26] Hongcheng Gan and Yang Bai. The effect of travel time variability on route choice decision: a generalized linear mixed model based analysis. *Transportation*, 41:339–350, 2014.

[27] Andreia P. Guerreiro, Carlos M. Fonseca, and Luís Paquete. The hypervolume indicator: Computational problems and algorithms. *ACM Computing Surveys (CSUR)*, 54(6):1–42, 2021.

[28] Abhishek Gupta, Yew-Soon Ong, and Liang Feng. Insights on transfer optimization: Because experience is the best teacher. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(1):51–64, 2017.

[29] Ali Haghani and Soojung Jung. A dynamic vehicle routing problem with time-dependent travel times. *Computers & operations research*, 32(11):2959–2986, 2005.

[30] Michael Pilegaard Hansen. Tabu search for multiobjective optimization: MOTS. In *Proceedings of the 13th international conference on multiple criteria decision making*, pages 574–586. Citeseer, 1997.

[31] Pierre Hansen and Nenad Mladenović. Variable neighborhood search. In *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. Springer, 2014.

[32] Wesam Herbawi and Michael Weber. A genetic local search algorithm for multiobjective time-dependent route planning. In *2012 IEEE Congress on Evolutionary Computation*, pages 1–7. IEEE, 2012.

[33] Farah F. Hussein, Gürsel A. Süer, and Bhaven Naik. Travel time modeling using non-linear multi-objective fuzzy optimization approach. In *International Conference on Transportation and Development 2020*, pages 104–118. American Society of Civil Engineers Reston, VA, 2020.

[34] Hisao Ishibuchi, Ryo Imada, Yu Setoguchi, and Yusuke Nojima. How to specify a reference point in hypervolume calculation for fair performance comparison. *Evolutionary computation*, 26(3):411–440, 2018.

[35] Zhaowang Ji, Anthony Chen, and Kitti Subprasom. Finding multi-objective paths in stochastic networks: a simulation-based genetic algorithm approach. In *Proceedings of the 2004 congress on evolutionary computation (IEEE Cat. No. 04TH8753)*, volume 1, pages 174–180. IEEE, 2004.

[36] Zhaowang Ji, Yong Seog Kim, and Anthony Chen. Multi-objective α-reliable path finding in stochastic networks with correlated link costs: A simulation-based multi-objective genetic algorithm approach (SMOGA). *Expert Systems with Applications*, 38(3):1515–1528, 2011.

[37] Dylan F. Jones, S.K. Mirrazavi, and M. Tamiz. Multi-objective meta-heuristics: An overview of the current state-of-the-art. *European journal of operational research*, 137(1):1–9, 2002.

[38] Nicolas Jozefowiez, Frédéric Semet, and El-Ghazali Talbi. Multi-objective vehicle routing problems. *European Journal of Operational Research*, 189(2):293–309, 2008.

[39] Yutaka J. Kanayama and Bruce I. Hartman. Smooth local-path planning for autonomous vehicles. *The International Journal of Robotics Research*, 16(3):263–284, 1997.

[40] Hitoshi Kanoh. Dynamic route planning for car navigation systems using virus genetic algorithms. *International Journal of Knowledge-based and Intelligent Engineering Systems*, 11(1):65–78, 2007.

[41] Hitoshi Kanoh, Takeshi Furukawa, Souichi Tsukahara, Kenta Hara, Hirotaka Nishi, and Hisashi Kurokawa. Short-term traffic prediction using fuzzy c-means and cellular automata in a wide-area road network. In *Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005.*, pages 381–385. IEEE, 2005.

[42] Hitoshi Kanoh and Kenta Hara. Hybrid genetic algorithm for dynamic multi-objective route planning with predicted traffic in a real-world road network. GECCO '08, pages 657—664, New York, NY, USA, 2008. Association for Computing Machinery.

[43] Adrianus Leendert Kok, Erwin W. Hans, and Johannes M.J. Schutten. Vehicle routing under time-dependent travel times: the impact of congestion avoidance. *Computers & operations research*, 39(5):910–918, 2012.

[44] Thomas Liebig, Nico Piatkowski, Christian Bockermann, and Katharina Morik. Dynamic route planning with real-time traffic predictions. *Information Systems*, 64:258–265, 2017.

[45] Sejoon Lim, Hari Balakrishnan, David Gifford, Samuel Madden, and Daniela Rus. Stochastic motion planning and applications to traffic. *The International Journal of Robotics Research*, 30(6):699–712, 2011.

[46] Hong-En Lin, Rocco Zito, and Michael Taylor. A review of travel-time prediction in transport and logistics. In *Proceedings of the Eastern Asia Society for transportation studies*, volume 5, pages 1433–1448. Bangkok, Thailand, 2005.

[47] Ying Ying Liu, Fatemeh Enayatollahi, and Parimala Thulasiraman. Traffic Aware Many-Objective Dynamic Route Planning. In *2019 IEEE Sym-*

posium Series on Computational Intelligence (SSCI), pages 1241–1248, 2019.

[48] C. Ludwig, J. Psotta, A. Buch, N. Kolaxidis, S. Fendrich, M. Zia, J. Fürle, A. Rousell, and A. Zipf. Traffic speed modelling to improve travel time estimation in openrouteservice. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 48:109–116, 2023.

[49] Achille Messac, Amir Ismail-Yahaya, and Christopher A. Mattson. The normalized normal constraint method for generating the pareto frontier. *Structural and multidisciplinary optimization*, 25:86–98, 2003.

[50] Achille Messac and Christopher A. Mattson. Normal constraint method with guarantee of even representation of complete pareto frontier. *AIAA journal*, 42(10):2101–2111, 2004.

[51] Seyedali Mirjalili, Jin Song Dong, and Andrew Lewis. Nature-inspired optimizers. *Studies in Computational Intelligence*, 811:7–20, 2020.

[52] Masaharu Munetomo, Yoshiaki Takai, and Yoshiharu Sato. A migration scheme for the genetic adaptive routing algorithm. In *SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 98CH36218)*, volume 3, pages 2774–2779. IEEE, 1998.

[53] Aydin Nassehi, Wesley Essink, and Jack Barclay. Evolutionary algorithms for generation and optimization of tool paths. *CIRP Annals*, 64(1):455–458, 2015.

[54] OpenStreetMap contributors. OpenStreetMap. `https://www.openstreetmap.org`, 2023.

[55] Mojtaba Rajabi-Bahaabadi, Afshin Shariat-Mohaymany, Mohsen Babaei, and Chang Wook Ahn. Multi-objective path finding in stochastic time-dependent road networks using non-dominated sorting genetic algorithm. *Expert Systems with Applications*, 42(12):5056–5064, 2015.

[56] Younes Regragui and Najem Moussa. Investigating the impact of real-time path planning on reducing vehicles traveling time. In *2018 International Conference on Advanced Communication Technologies and Networking (CommNet)*, pages 1–6. IEEE, 2018.

[57] Marko A. Rodriguez and Peter Neubauer. Constructions from dots and lines. *Bulletin of the American Society for Information Science and Technology*, 36(6):35–41, August 2010.

[58] Suvrajeet Sen, Rekha Pillai, Shirish Joshi, and Ajay K Rathi. A mean-variance model for route guidance in advanced traveler information systems. *Transportation Science*, 35(1):37–49, 2001.

[59] State of Berlin. Top 10 Berlin Sights and Attractions. `https://www.berlin.de/en/attractions-and-sights/top-10/`. [Online; accessed 2023-10-06].

[60] A. Suppapitnarm, Keith A. Seffen, Geoff T. Parks, and P.J. Clarkson. A simulated annealing algorithm for multiobjective optimization. *Engineering optimization*, 33(1):59–85, 2000.

[61] Bogdan Tatomir, Leon J.M. Rothkrantz, and Adriana C. Suson. Travel time prediction for dynamic routing using ant based control. In *Proceedings of the 2009 Winter Simulation Conference (WSC)*, pages 1069–1078. IEEE, 2009.

[62] Uber Technologies, Inc. Historical Speeds, Hourly Time Series (January 2020). `https://movement.uber.com/cities/berlin/downloads/speeds?lang=en-US&tp[y]=2020&tp[q]=1`. [Online; accessed: 2023-04-14].

[63] Uber Technologies, Inc. Uber Movement: Speeds Calculation Methodology. `https://movement.uber.com/_static/97e6e916ed8e8176.pdf`. [Online; accessed: 2023-05-03].

[64] Tom Van Woensel, Laoucine Kerbache, Herbert Peremans, and Nico Vandaele. Vehicle routing with dynamic travel times: A queueing approach. *European journal of operational research*, 186(3):990–1007, 2008.

[65] Haichao Wang, Jinlong Wang, Guoru Ding, Jin Chen, and Jian Yang. Completion time minimization for turning angle-constrained UAV-to-UAV communications. *IEEE Transactions on Vehicular Technology*, 69(4):4569–4574, 2020.

[66] Shuihua Wang, Jianfei Yang, Ge Liu, Sidan Du, and Jie Yan. Multi-objective path finding in stochastic networks using a biogeography-based optimization method. *Simulation*, 92(7):637–647, 2016.

[67] Jens Weise and Sanaz Mostaghim. A Scalable Many-Objective Pathfinding Benchmark Suite. *IEEE Transactions on Evolutionary Computation*, 26(1):188–194, 2022.

[68] Michael P. Wellman, Matthew Ford, and Kenneth Larson. Path planning under time-dependent uncertainty. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 532–539, 1995.

[69] Xu Zhang and Mei Chen. Genetic algorithm–based routing problem considering the travel reliability under asymmetrical travel time distributions. *Transportation Research Record*, 2567(1):114–121, 2016.

[70] Xu Zhang and Mei Chen. Bi-objective routing problem with asymmetrical travel time distributions. *Journal of Intelligent Transportation Systems*, 22(2):87–98, 2018.

[71] Yudong Zhang, Yan Jun, Geng Wei, and Lenan Wu. Find multi-objective paths in stochastic networks via chaotic immune PSO. *Expert Systems with Applications*, 37(3):1911–1919, 2010.

[72] Jiandong Zhao, Yujie Guo, and Xiaohong Duan. Dynamic path planning of emergency vehicles based on travel time prediction. *Journal of advanced transportation*, 2017, 2017.

[73] P.X. Zhao, W.H. Luo, and X. Han. Time-dependent and bi-objective vehicle routing problem with time windows. *Advances in Production Engineering & Management*, 14(2):201–212, 2019.

[74] Wanzheng Zheng, Pranay Thangeda, Yagiz Savas, and Melkior Ornik. Optimal routing in stochastic networks with reliability guarantees. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 3521–3526. IEEE, 2021.

Statement of Authorship

Thesis	"Innovization for Multi-Objective Time-Dependent Route Planning"
Name	Eva Röper
Date of birth	18.01.1998
Matriculation no.	214782

I herewith assure that I wrote the present thesis independently, that the thesis has not been partially or fully submitted as graded academic work and that I have used no other means than the ones indicated. I have indicated all parts of the work in which sources are used according to their wording or to their meaning.

I am aware of the fact that violations of copyright can lead to injunctive relief and claims for damages of the author as well as a penalty by the law enforcement agency.

Magdeburg, January 18, 2024

Eva Röper