WORKING PAPER SERIES

Dynamic learning-based Search for Multi-criteria Itinerary Planning

Thomas Horstmannshoff/Jan Fabian Ehmke/Marlin Ulmer

Working Paper No. 11/2023

OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

FACULTY OF ECONOMICS
AND MANAGEMENT

Impressum (§ 5 TMG)

Herausgeber:
Otto-von-Guericke-Universität Magdeburg
Fakultät für Wirtschaftswissenschaft
Der Dekan

Verantwortlich für diese Ausgabe:
T. Horstmannshoff, J. F. Ehmke, M. Ulmer
Otto-von-Guericke-Universität Magdeburg
Fakultät für Wirtschaftswissenschaft
Postfach 4120
39016 Magdeburg
Germany

http://www.fww.ovgu.de/femm

Bezug über den Herausgeber
ISSN 1615-4274

Dynamic learning-based Search for Multi-criteria Itinerary Planning

Thomas Horstmannshoff[1*], Jan Fabian Ehmke[2], Marlin Ulmer[1]

[1]Management Science Group, Otto von Guericke University Magdeburg,
Universitätsplatz 2, 39106 Magdeburg, Germany
thomas.horstmannshoff@ovgu.de
[2]Department of Business Decisions and Analytics, University of Vienna,
Kolingasse 14-16, 1090 Vienna, Austria

Abstract

Travelers expect integrated and multimodal itinerary planning while addressing their individual expectations. Besides common preferences such as travel time and price, further criteria such as walking and waiting times are of importance as well. The competing features of these preferences yield a variety of Pareto-optimal itineraries. Finding the set of Pareto-optimal multimodal travel itineraries in efficient run time remains a challenge in case multiple traveler preferences are considered.

In this work, we present a sampling framework to approximate the set of Pareto-optimal travel itineraries that scales well in terms of considered preferences. In particular, we guide the search process dynamically to uncertain areas of the complex multimodal solution space. To this end, we learn the structure of the Pareto front during the search with Gaussian Process Regression (GPR). The GPR sampling framework is evaluated integrating an extensive amount of real-world data on mobility services. We analyze long-distance trips between major cities in Germany. Furthermore, we take up to five traveler preferences into account. We observe that the framework performs well, revealing origin and destination specifics of Pareto fronts of multimodal travel itineraries.

Keywords: Routing, Multi-Criteria Decision Support, Multimodal Mobility, Gaussian Process Regression

Declarations of interest: *none.*

1 Introduction

Travel planning has always been a challenge, searching for travel modes, checking connections, booking tickets, and all this with a firm eye on time, cost, and convenience. Today's digitization automates some of the parts with single-mode planning for trains, planes, buses, and on-demand services but it does not solve the main problem that when planning, the different modes need to be considered in an integrated manner. First platforms such as *GoogleMaps* and *Rome2Rio* aim for integrated and multimodal planning of travel. For such services, the traveler is either offered only one or two, mostly time- or cost-efficient, options, or the traveler is left alone will all itineraries available. Both do not address the multiple expectations travelers usually have when they travel. Besides classical goals of cheap and fast travel, other criteria play important roles, e.g., a limited number of transfers as well as walking or waiting times. All these options are usually competing against each other, e.g. an itinerary can be either cheap but slow or very fast but expensive. Given that the platforms (and often the travelers) do not know the importance of individual preferences, the travelers desire a reasonable-sized set of options covering all different preferences to select

their preferred option, and, ideally, they expect the instant provision of such a set. While this is very convenient for travelers, it poses several challenges for the platform providers. First of all, searching for itineraries over different modes is a challenge in itself. Second, after the search, the platform is confronted with thousands of possible itineraries and now needs to find a small, diverse set to capture potential preferences within a fraction of a second. How to instantly provide a reasonable-sized, diverse set of options in a multimodal context is the focus of this work.

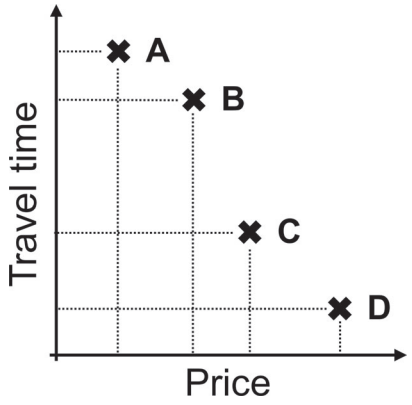


Figure 1: Exemplary Pareto front

Mathematically, we aim on determining a Pareto-set for multiple preferences integrating all available mobility services. Figure 1 depicts exemplary Pareto-optimal itineraries for two different preferences, travel time and price. Itinerary A is the cheapest, but the slowest, itinerary D is the fastest, but most expensive option. Itineraries B and C are non-dominated compromises between time and price, which differ in terms of travel time and price. Delling et al. (2013) and Dib et al. (2017) highlight that taking more than three preferences into account at the same time results in run time issues for this Multi-Criteria Decision Making (MCDM) problem. Since run time is highly important for mobility platforms, multimodal search algorithms have to scale efficiently in terms of the number of individual traveler preferences integrated (Bast et al., 2015; He et al., 2022). To ensure scalability, approximating the Pareto-optimal set of itineraries instead of identifying the full Pareto-optimal set has been proposed in recent literature (He et al., 2022; Horstmannshoff & Ehmke, 2022). In Horstmannshoff & Ehmke (2022), we have proposed an a posteriori systematic sampling framework assuming that the traveler expresses individually relevant preferences with no underlying hierarchy. Furthermore, we require the respective origin, the destination, and the earliest departure time by the traveler. Based on this general information, we approximate the set of Pareto-optimal itineraries by computing multiple two-dimensional sets simultaneously with many iterations of solution sampling equally distributed over the solution space. We have observed efficient scaling integrating up to five traveler preferences. However, we have noticed that uniform sampling across the multimodal solution space might lead to inefficiencies with respect to the approximated Pareto front as there is no focus on relevant regions.

In this work, we enhance our systematic sampling framework by actively learning the Pareto front structure during the search and dynamically guiding the search to promising parts of the complex multimodal solution space. The core idea is to apply Gaussian Process Regression (GPR) to identify the area with the highest uncertainty in the solution space and focus the sampling process on that area. Consequently, we improve the systematic sampling framework where no dynamic guidance occurred while searching the multimodal solution space. The choice set retrieved by the GPR sampling framework can then be presented to the traveler enabling the traveler to take well-informed decisions. Furthermore,

giving additional information about the structure of the multimodal solution space in an expert menu in the integrated mobility application assists the traveler in their decision-making progress. The proposed GPR sampling framework is evaluated in a proof-of-concept study embedding a large amount of real-world data from multiple mobility services. We analyze long-distance trips between major cities in Germany, considering up to five traveler preferences. This includes a comparison with the results of the systematic sampling framework, which serves as a baseline of evaluation. With our extensive experiments, we demonstrate efficient scaling of the GPR sampling framework in terms of considered traveler preferences. We observe that learning the Pareto front structure actively is valuable, in particular, if only a limited number of runs are available. Compared to the baseline approach, improvements are observed mainly for those preferences that are continuous (esp. travel time and waiting time). Furthermore, we also demonstrate how the structure of the Pareto-optimal solutions affects these improvements.

An overview of related work on traveler-oriented route planning, multimodal route planning, and GPR is presented in Section 2. Then, we introduce the GPR sampling framework in detail in Section 3. In Section 4, we present the considered data on mobility services, traveler preferences, and further framework settings. In addition, we introduce relevant metrics for evaluation as well as discuss the parameterization of the GPR. Next, we present results in terms of the examined improvement and the increased effectiveness in comparison to the baseline approach in Section 5. Finally, we give a summary and identify areas for further research in Section 6.

2 Related literature

Multi-criteria itinerary planning is a well-established research field with a large number of publications and practical relevance. First, we discuss requirements traveler have while planning multi-criteria itineraries in Section 2.1. Next, we examine the resulting challenges for mobility platform providers in Section 2.2. In this paper, we propose an active learning sampling framework to identify multimodal traveler itineraries according to multiple individual traveler preferences by actively learning Pareto front structures. Related approaches are discussed in Section 2.3.

2.1 Traveler requirements to multi-criteria itinerary planning

Multiple studies highlight the importance of taking diverse traveler requirements into account while planning multimodal itineraries. Travelers expect a one-stop search for door-to-door travel and do not want to combine and compare results from different mobility applications manually. Consequently, information on all available mobility services has to be integrated into one mobility application (Stopka, 2014; T. Schulz et al., 2020; Valderas et al., 2020). Grotenhuis et al. (2007) emphasize that offering integrated multimodal itinerary planning reduces the cognitive and temporal effort for the traveler. Furthermore, travelers have multiple competing traveler preferences which need to be integrated into the search (Spickermann et al., 2014; Lyons et al., 2020; Musolino et al., 2023). Horstmannshoff (2022) presents an overview of recent studies about individually relevant preferences. Beside common traveler preferences such as travel time, price, and number of transfers, further preferences (e.g. walking distance, waiting time, reliability, and sustainability) can be of importance for the individual traveler as well (Samaranayake et al., 2011; Lyons et al., 2020; Liang et al., 2023). Integrating multiple competing preferences into the search yields a set of Pareto-optimal itineraries. For instance, while one itinerary is fast but very expensive, another is much cheaper but requires the traveler to invest more time. Presenting a diverse, reasonably sized set of Pareto-optimal itineraries enables the traveler to choose according to situational and habitual requirements

(Grotenhuis et al., 2007; Spickermann et al., 2014; Stopka et al., 2016). Wu et al. (2021) propose to infer request-specific traveler preferences based on automatic fare collection data. In this work, we consider five prevalent preferences in an extensive multimodal context to approximate a set of multimodal itineraries. Furthermore, presenting insights into the complex solution space and the itinerary characteristics provides additional information to the traveler.

2.2 Challenges of platform providers for multi-criteria itinerary planning

Several challenges arise from multiple traveler requirements for mobility platform providers. Core of each mobility platform is a multimodal routing algorithm applicable for MCDM problems. Bast et al. (2015) present a detailed overview of route planning algorithms as well as the extension of these into a multimodal context. Herzel et al. (2021) highlight that most methods for MCDM problem settings assume at most three objectives. Delling et al. (2013) introduce a label-based algorithm to retrieve a Pareto-optimal set of itineraries in a multimodal context while integrating three preferences at the same time. Dib et al. (2017) proposes a multi-criteria planning algorithm taking travel time, walking time and number of transfers into account. Potthoff & Sauer (2022) extend established bi-criteria routing algorithms by focusing on relevant solutions only during the search. The vast majority of the multi-criteria multimodal routing algorithms introduced in the literature – including the above approaches – consider only two or three preferences simultaneously. Delling et al. (2013); Dib et al. (2017); Potthoff & Sauer (2022) emphasize that significant run time issues arise when integrating more preferences into the search while identifying the full Pareto-optimal set.

To ensure scalability and to make the solution space more accessible to travelers, approximating the set of Pareto-optimal itineraries has recently been proposed. We have introduced a systematic sampling framework in our recent paper Horstmannshoff & Ehmke (2022). By breaking down the high-dimensional problem setting into multiple problems of smaller dimensions, we ensure scalability even for a large number of considered traveler preferences. In this work, we propose a scalable multi-criteria itinerary planning framework. In particular, we actively guide the sampling process during the search by dynamically learning the structure of the Pareto front. Thereby, we tackle a shortcoming of our recent paper of sampling the complex multimodal solution space uniformly.

2.3 Dynamically learning Pareto front structures

Guiding the search to promising areas of the complex multimodal solution space is challenging. In our sampling framework, we aim at learning the function describing the structure of the Pareto front as accurately as possible with a limited number of iterations available, hence ensuring efficient run time. One common approach to actively explore unknown functions with high accuracy is Gaussian Process Regression (GPR) (E. Schulz et al., 2018). The core idea of GPR is to learn multiple possible functions according to a preset underlying kernel. Then, once we have some data points $x_1, \dots, x_N$, we learn a scalar function $f(x)$. Hereby, it is assumed that the function $p(f(x_1), \dots, f(x_N))$ is jointly Gaussian, with some mean $\mu(x)$ and covariance $\sum(x)$ provided by $\sum_{ij} = k(x_i, x_j)$, where k is a positive definite kernel function. Using the kernel, we control that x_i and x_j are seen as similar to each other in case we expect these points close to each other in the output function as well (Rasmussen & Williams, 2006; Murphy, 2012). The applied configuration of the GPR kernel for our GPR sampling framework will be discussed in Section 4.3.

GPR is applied in many areas of MCDM. For instance, Palm et al. (2022) propose a GPR based

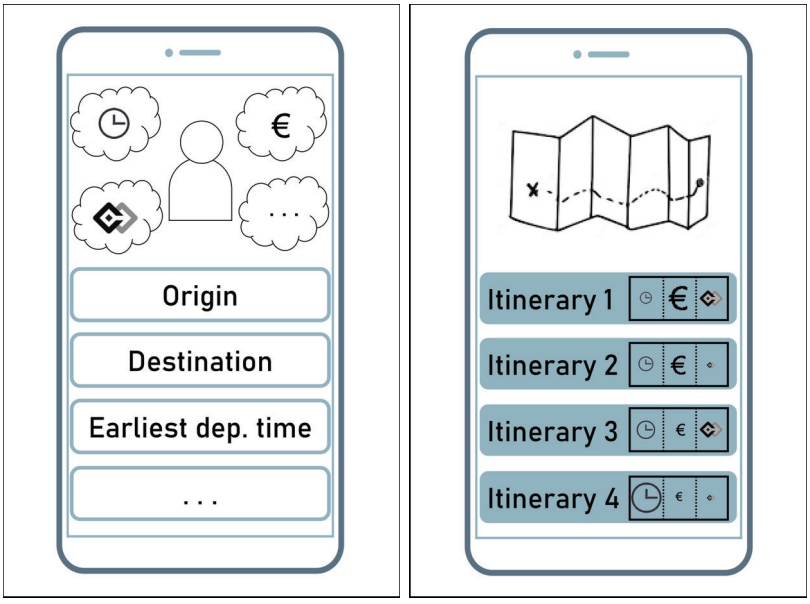
multi-objective Bayesian optimization for designing power systems. Deringer et al. (2021) present a detailed overview of how GPR can be used in the area of computational materials science and chemistry taking multiple criteria in parallel into account. However, GPR has yet not been utilized in the area of multimodal itinerary planning.

We apply GPR in the area of multi-criteria multimodal mobility planning to steer our sampling framework to promising parts of the complex solution space.

3 Gaussian process regression sampling framework

In the following, we present our GPR sampling framework enhancing the systematic sampling framework introduced by Horstmannshoff & Ehmke (2022). The idea is to learn the Pareto front structure actively during the search and dynamically guide the search to promising parts of the complex multimodal solution space. We refer to the enhanced framework outlined here as *GPR sampling*. We refer to the framework presented in Horstmannshoff & Ehmke (2022), in which no dynamic guidance takes place while searching the multimodal solution space, as *systematic sampling*. Based on the problem description in Section 3.1, we introduce the enhanced GPR framework in Section 3.2. Finally, we present the multimodal network model and the multimodal shortest-path algorithm in Section 3.3.

3.1 Problem description



(a) Traveler request (b) Presentation of solution set

Figure 2: Individual traveler request to an integrated mobility application

Figure 2 depicts the overall procedure for a new traveler request. As common for a posteriori settings, we assume that basic information such as the individual origin O, the destination D, and the earliest departure time t_{dep} is given by the traveler (Figure 2(a)). Furthermore, the traveler has an individual set of preferences in mind without being able to judge the respective importance of these. Based on the provided information and using an integrated mobility application, the traveler expects a diverse set of solutions to choose from (Figure 2(b)). Additional information about the solution preference values assists the traveler in their decision-making progress.

To identify the choice set in reasonable run time, several challenges result for the platform provider of the integrated mobility application. First of all, the provider has to integrate a set of ms available mobility services $MS := \{MS_1, \ldots, MS_{ms}\}$ such as ridesharing services and railway into the multimodal network model. Furthermore, a set of traveler preferences P has to be considered. The set $P := \{P_1, \ldots, P_p\}$ contains p traveler preferences such as price, travel time, and the number of transfers. Knowledge of these parameters is a common assumption for a posteriori settings.

To solve the problem setting at hand, we propose a framework that is capable of retrieving multiple itineraries quickly while integrating multiple traveler preferences simultaneously in a large multimodal network. In the following, we will refer to an itinerary as a solution. To ensure scalability with respect to the number of considered traveler preferences – hence ensure fast run times – we approximate high-dimensional Pareto fronts by sampling many lower-dimensional Pareto fronts in parallel resulting in a set of Pareto-optimal multimodal solutions S_{all}^{opt}. In particular, we rerun a multimodal routing algorithm SPM multiple times with different configurations for each lower-dimensional set. These configurations are actively learned during the search (details in Section 3.2.2). Each run of SPM returns an optimal solution Sol, which contains information about the solution as well as its preference values. Hereby, Sol_i represents the solution value for traveler preference $i \in P$. The set S_{all}^{opt} is composed of different Pareto-optimal solutions. We create several sets of solutions for two and three dimensions. Here, the term dimension refers to how many traveler preferences are taken into account simultaneously when creating the respective sets. Two-dimensional sampling means that we take two preferences into account when approximating the Pareto front for each set. For three-dimensional sampling, we extend this to three preferences considered at the same time. We will repeat sampling pairwise (for two-dimensional sampling) or threefold (for three-dimensional sampling), respectively, to approximate sets of solutions considering all preferences.

3.2 Framework

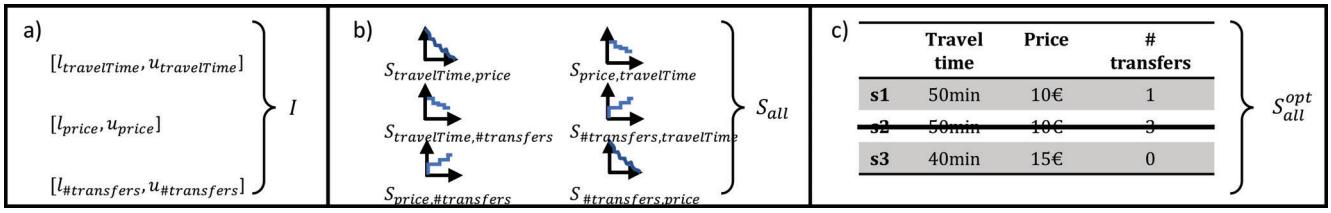


Figure 3: Main steps in sampling framework

3.2.1 Overview

Figure 3 shows the basic procedure of our GPR sampling framework. For a new traveler request, we first identify a min-max-interval $[l_i, u_i]$ for each preference $i \in P$ resulting in a set of intervals I indicating the worst and best value for the respective preference. The intervals are determined by a single-objective run of the multimodal routing algorithm for each traveler preference and setting the obtained minimum and maximum values of the respective preferences as interval boundaries. This guarantees that only reasonable parts of the multimodal solution space are searched. Figure 3(a) presents an example of identifying the set of intervals I in case three preferences $P := \{$travel time, price, number of transfers$\}$ are considered.

Next, we approximate the set of Pareto-optimal solutions by many runs of the multimodal routing algorithm SPM. Therefore, we compute multiple two-dimensional sets $S_{i,j} | i \neq j \in P$ simultaneously.

Based on the ϵ-constraint method (Deb, 2011), we set preference $i \in P$ as a single objective and iteratively alter the upper-bound constraint of preference $j \in P$ for each set $S_{i,j}$. Figure 3(b) depicts a simplified example for identifying six two-dimensional sets in parallel. The reduction of a high-dimensional problem setting into multiple two-dimensional problem sets reduces the computational complexity of calculating each of these sets significantly as we merely have a single-objective problem at hand, and the different upper-bound configurations are only impacted by one preference at the same time. This guarantees scalability even for a larger number of preferences since only the number of parallelly determined sets grows, but not the complexity of calculating each of these two-dimensional sets.

To ensure a focused search on promising areas of the solution space, we learn actively during the search process where to sample next to guide the search and make our sampling framework more effective by applying GPR. The overall procedure how we identify each two-dimensional set $S_{i,j}$ depends on whether preference j is subject to an integer constraint (e.g. number of transfers) or not (e.g. price). For the latter, we dynamically decide which area of the complex multimodal solution space is to be sampled next by actively learning the area with highest uncertainty. Details are presented in Section 3.2.2. For preferences to which the former applies, only a few possible upper-bound constraints can be set. Following, we set each possible integer value in the identified min-max-interval $[l_j, u_j]$ as an upper-bound constraint and run the multimodal routing algorithm SPM with preference i as a single-objective. Details are presented in Section 3.2.3. After each two-dimensional set $S_{i,j}$ has been calculated, these are merged together into a joint set $S_{all} = \bigcup_{i,j|i\neq j\in P} S_{i,j}$.

Finally, as shown in Figure 3(c), dominated solutions are removed from S_{all} as they are of no relevance to the traveler. A solution Sol_1 dominates a solution Sol_2 if Sol_1 is strictly superior to Sol_2 regarding at least one preference $i \in P$ and not inferior in terms all other preferences (Delling et al., 2013). All remaining Pareto-optimal solutions shape the travelers' choice set S_{all}^{opt}. For a comprehensive introduction to the systematic sampling framework (without GPR applied) including detailed pseudo codes, we refer to Horstmannshoff & Ehmke (2022).

Algorithm 1 GPR solution space sampling framework algorithm

1: $I \leftarrow$ IdentificationOfMinMaxIntervals(O, D, t_{dep}, P)
2: **for all** $i, j | i \neq j \in P$ **do** ▷ Parallelized Execution
3: **if not** j s.t. integer condition **then**
4: $S_{i,j} \leftarrow$ GPR$(O, D, t_{dep}, i, j, I, k)$ (details in 3.2.2)
5: **else**
6: $S_{i,j} \leftarrow$ SystematicSampling(O, D, t_{dep}, i, j, I) (details in 3.2.3)
7: **end if**
8: **end for**
9: $S_{all} = \bigcup_{i,j|i\neq j\in P} S_{i,j}$
10: $S_{all}^{opt} \leftarrow$ RemovalOfDominatedSolutions(S_{all})

Algorithm 1 presents the procedure of our sampling framework in detail. After identifying a set of min-max-intervals I in line 1, we compute multiple two-dimensional sets $S_{i,j}$ in parallel (line 2). In case preference $j \in P$ is not subject to an integer constraint, we apply the feature of GPR where to sample next and thereby focus on promising parts of the complex multimodal solution space (line 4). Otherwise, we systematically sample the solution space (line 6). After each two-dimensional set has been calculated, we merge all sets (line 9) and remove dominated solutions from the merged choice set (line 10).

Algorithm 2 GPR learning sampling

```
1: function GPRSampling(O, D, t_dep, i, j, I, k)
2:     for ĵ ∈ range(1, k) do
3:         if ĵ = 1 then
4:             NextSamplingPoint ← 0.5
5:         else
6:             NextSamplingPoint ← GaussianProcessRegression(S_{i,j}, Blocked_{i,j})
7:         end if
8:         C_{restrValue_j} = NextSamplingPoint * (u_j − l_j) + l_j
9:         Sol ← SPM(O, D, t_dep, i, C)
10:        S_{i,j} = S_{i,j} ∪ Sol
11:        Blocked_{i,j} ← AddBlockedArea()
12:        if not AreaAvailable(Blocked_{i,j}) then
13:            terminate()
14:        end if
15:    end for
16:    return S_{i,j}
17: end function
```

3.2.2 Gaussian process regression sampling

Next, we introduce how to dynamically decide where to sample next by applying GPR sampling. Algorithm 2 details the overall procedure. Given basic information of the individual request, we sample k times (line 2 in Algorithm 2). In the first iteration of the sampling process, we set the next sampling point in the middle of the identified min-max-interval as we only have knowledge of the min-max-interval boundary solutions (line 4). For the further iterations, we use GPR to identify the area with the highest uncertainty in the solution space and set the next sampling point (line 6). The core idea of GPR is to actively learn the complex structure of the function describing the Pareto front taking already found solutions as well as areas marked as "blocked" into account.

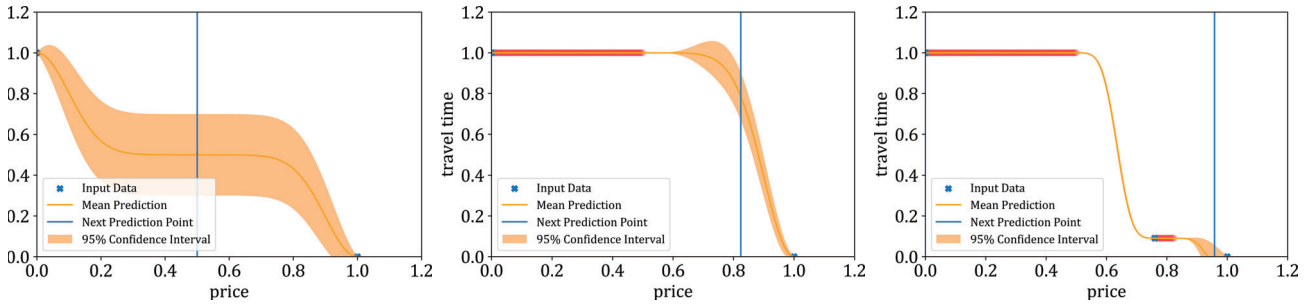
The underlying assumptions of the complex structure (e.g. linear, quadratic, and periodic) can be controlled by a positive definite kernel function, which describes the covariance of the Gaussian process random variables. Using the mean and standard deviation retrieved by GPR, we identify the area with the highest uncertainty in the complex multimodal solution space. This area is chosen as the next sampling point. Thereby, we guide the sampling process dynamically during the search. Please note that we 0-1-normalize the already found solution values to avoid scaling issues in GPR (Rasmussen & Williams, 2006; E. Schulz et al., 2018). Details about the configuration of GPR will be discussed in Section 4.3.

Next, we convert the next sampling point into the actual upper-bound constraint value for preference j (line 8) and run the multimodal routing algorithm SPM with the current setting (line 9). Then, we add the retrieved solution Sol to the set $S_{i,j}$ (line 10).

Furthermore, we mark certain areas as blocked. Blocked areas indicate that we have already retrieved the respective solution that we would retrieve when running the sampling framework with upper-bound constraints within the areas marked as blocked. Therefore, there is no added value in running the sampling framework with the respective setting: the area between the value for preference j (for which the upper-bound constraints are dynamically adjusted) of the newly found solution Sol and the respective upper-bound constraint set for j is marked as blocked since any upper-bound constraint would lead to the same solution. Furthermore, we block the area between the new found solution Sol and an already identified solution in $S_{i,j}$, which both have the same objective function value. In case all solutions between the lower-interval bound l_i and the upper-interval bound u_i are marked as blocked (line 12), we know

that no additional solutions will be found in the respective two-dimensional set and thus we terminate the sampling process for the respective set (line 13).

An example for the set $S_{travelTime,price}$ is presented in Figure 4(a). Already found solutions (here only the boundary solutions) are depicted by blue crosses. While the mean prediction retrieved by GPR is shown by the orange line, the orange highlighted area indicates the 95% confidence interval. The area with the highest uncertainty (largest confidence interval) is then used as the next sampling point and highlighted by a blue vertical line. The multimodal routing algorithm is started with the upper-bound constraint for the price set in the middle of the corresponding interval. As shown in Figure 4(b), the resulting solution is equal to the lower-bound solution of the previously identified min-max-interval. Consequently, this range of the multimodal solution space is marked as blocked (highlighted in red), since no new solutions can be received with upper-bound constraints in this part of the solution space. Subsequently, the next sampling point is determined in iteration 2 by means of GPR resulting in a normalized value of about 0.8. With this configuration, the multimodal routing algorithm is started again in the third iteration (Figure 4(c)), and the area between the set upper-bound constraint for price and the value of the solution found for price is marked as blocked. This process is repeated until k iterations have been passed or all solutions ranging from the lower-interval bound l_i to the upper-interval bound u_i are marked as blocked.



(a) Iteration 1 (b) Iteration 2 (c) Iteration 3

Figure 4: GPR sampling for set $S_{travelTime,price}$

3.2.3 Systematic sampling for integer constraints

Algorithm 3 Systematic sampling for interval constraints

1: **function** SystematicSampling(O, D, t_{dep}, i, j, I)
2: $\hat{j} = u_j - l_j$
3: **while** $\hat{j} \geq 0$ **do**
4: $C_{restrValue_j} = l_j + \hat{j}$
5: $Sol \leftarrow \text{SPM}(O, D, t_{dep}, i, C)$
6: $S_{i,j} = S_{i,j} \cup Sol$
7: $\hat{j} = \hat{j} - 1$
8: **end while**
9: **return** $S_{i,j}$
10: **end function**

In the following, we introduce the systematic sampling framework applied in case preference j is subject to an integer constraint. Given the traveler's origin O and destination D, the earliest departure time t_{dep}, preference $i \in P$ set as a single-objective, preference $j \in P$, which upper-bound constraint is systematically altered, and the interval set I, we create the two-dimensional set $S_{i,j}$ as shown in Algorithm 3. For each

value in the respective min-max-interval $[l_j, u_j]$ (line 3 to line 8), we systematically alter the upper-bound constraint for preference j (line 4), then run the multimodal routing algorithm SPM (line 5), and finally add the retrieved soluion to the two-dimensional set $S_{i,j}$ (line 6).

Figure 5 shows a fictitious example for determining set $S_{price,numberOfTransfers}$ with "price" set as the single-objective and "number of transfers" set as the systematically altered upper-bound constraint which is subject to an integer constraint. For each value in the interval $[l_{numberOfTransfers}, u_{numberOfTransfers}] := [0, 3]$ set as an upper-bound constraint, the multimodal routing algorithm is run, and different solutions are retrieved.

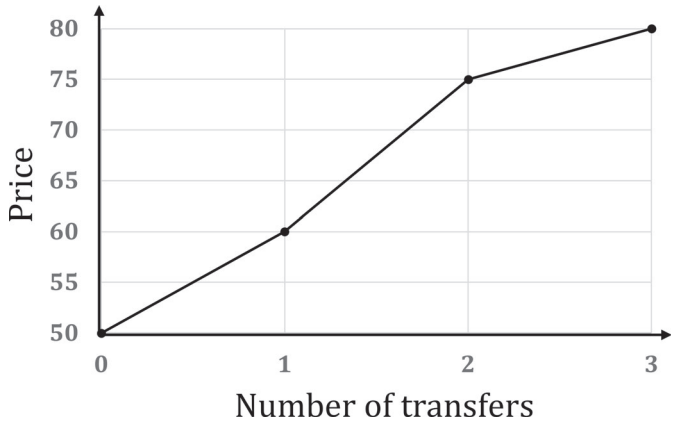


Figure 5: Systematic sampling for set $S_{price,numberOfTransfers}$

3.2.4 Extension to three-dimensional sampling

We also investigate our sampling framework for three dimensions aiming at identifying additional solutions which cannot be found in a two-dimensional setting. This results in multiple three-dimensional sets $S^3_{i,j,h}$ for each considered preference $i, j, h | i \neq j \wedge i \neq h \wedge j < h \in P$. In this case, we set traveler preference i as the objective and dynamically alter the upper-bound constraints for preferences j and h by our framework. The extension of the sampling framework to a three-dimensional setting works analogously to the two-dimensional setting. The identification of the next sampling point by GPR is then conducted in three dimensions.

3.3 Multimodal network model and routing

We require a multimodal network and a multimodal routing algorithm for our sampling framework. We build a multimodal network model G that incorporates various mobility services following Pajor (2009). First, for each considered mobility service $ms \in MS$, we create a unimodal network $G_{ms} = (V_{ms}, A_{ms})$ modeled as a time-expanded network. The set of vertices V_{ms} represents arrival and departure events. The set of arcs A_{ms} models a valid trip/subsequence of a trip between two vertices, which are assigned to the same trip. In the next step, we merge all unimodal networks into one multimodal network $G = (V, A)$. The set of vertices V (arcs A) is created by merging all unimodal sets V_{ms} (A_{ms}) with $ms \in MS$ into one set. In addition, we add transfer and link arcs. While transfer arcs provide feasible transfers between the same mobility service at the same stop location at different arrival and departure events, link arcs represent transfers between different mobility services. These transfers have to be in a given walking distance wd^{max} assuming a predefined walking speed ws.

In addition, we enable car usage by adding an arc from O to D assuming that an itinerary cannot

be partially driven by car. Furthermore, we use the concept of Contraction Hierarchy (CH) to establish shortcuts in the network between relevant vertices and therefore speed up the optimization (Geisberger et al., 2008). Finally, to avoid unrealistic solutions, e.g., using a bus service in between two car sections, we use non-deterministic finite automata fa, which represent the travelers' mode choice in the network model (Pajor, 2009; Bast et al., 2015). The core idea is that all solutions retrieved by SPM have to fulfill the solution structure as defined by fa. Building on the multimodal network as well as the finite automata, we create a product network $G^{\times} = (V^{\times}, A^{\times})$.

Given the network, we rerun a multimodal shortest-path algorithm SPM as presented by Pajor (2009) multiple times with different settings as described above. For an in-depth explanation of creating the multimodal network and the resource-constrained multimodal shortest-path algorithm including a detailed pseudo code, we refer to Horstmannshoff & Ehmke (2022).

4 Computational design

In the following, the experimental setup for the evaluation of the GPR sampling framework (Section 4.1), relevant metrics (Section 4.2), and the configuration of the GPR (Section 4.3) will be introduced.

4.1 Experimental setup

To evaluate the GPR sampling framework, we examine 300 origin-destination-combinations (ODs) between the main railway stations of major cities in Germany assuming the earliest departure time t_{dep} at 9 am on October 8, 2018. We have chosen this date as data for all integrated mobility services are available and it represents a regular working day (Monday). The resulting multimodal network $G = (V, A)$ for this day of operation consists of about 40,000 vertices and 10,000,000 arcs.

The set of mobility services MS is based on a large amount of real-world data. We include available General Transit Feed Specification (GTFS) data for German Railways, Flixbus, and local transit services. In addition, we use publicly available service data for flights and long-distance ridesharing service like "BlaBlaCar". We consider major airports with more than 50,000 aircraft movements per year (Berlin-Schönefeld, Cologne/Bonn, Berlin-Tegel, Dusseldorf, Frankfurt am Main, Munich, Stuttgart, Hamburg, Hanover, Leipzig/Halle, Nuremberg). Furthermore, we assume that the traveler has to arrive at the respective airport one hour prior to the flight departure, and requires 15 minutes after landing to get out of the airport. To enable car usage, we integrate information on the road network using the open-source routing library GraphHopper (*www.graphhopper.com*). For a more realistic estimation of travel times during peak hours, we multiply the retrieved travel times by 1.25. Furthermore, we set the walking speed ws to $5\frac{km}{h}$ and the maximum walking distance wd^{max} to $0.5km$ for adding the link and transfer arcs as described in Section 3.3.

Since German Railways and Flixbus' prices are not disclosed in the GTFS data, we estimate them as follows. For German Railways, the estimation is based on the train type chosen as well as the distance covered with the respective train type. We consider three different types of trains, namely *regional trains* (slowest train), *intercity trains* as well as *intercity express trains* (fastest train). We assume that intercity trains are 50% more expensive and intercity express trains are 100% more expensive in comparison to regional trains. Based on preliminary empirical investigations on *www.bahn.de*, the price for regional trains will be €17 per 100-kilometer distance traveled. For Flixbus, we assume €10 per 100-kilometer distance traveled. For individual road mobility, we assume 30 cents per kilometer following the flat-rate depreciation allowance in the German tax system. Note that all these values are only rough estimates and

can be adapted as needed in a real-world scenario to, e.g., incorporate tariff fare structures and advance booking periods (Randelhoff, 2022; Schöbel & Urban, 2022). The prices for flights are based on real data, which have been sampled for the respective day of the experiment.

The set of traveler preferences P consists of: *travel time* (tt), *price* (pr), *number of transfers* (nt), *overall walking distance* (wd), and the overall *waiting time* (wt). We examine the following combinations as stated in Table 1:

Table 1: Examined combinations of traveler preferences

Combination	Set P consists of
tt, pr	{travel time, price}
tt, pr, nt	{travel time, price, number of transfers}
tt, pr, nt, wd	{travel time, price, number of transfers, walking distance}
tt, pr, nt, wd, wt	{travel time, price, number of transfers, walking distance, waiting time}

In the first set, we only consider travel time and price. These two preferences are perceived as essential decision criteria for the traveler (Grotenhuis et al., 2007). Subsequently, we integrate additional prevalent exemplary preferences of high importance for the traveler into the search (Grotenhuis et al., 2007; Esztergár-Kiss & Csiszár, 2015; Alt et al., 2019). In general, it is simple to add further preferences that can be modeled as cost values of an arc in the multimodal network.

In addition, we analyze the impact of different sampling granularities on the quality of the Pareto front approximation, which is controlled by the sampling density parameter k. We evaluate the GPR sampling framework by setting k to $8, 16, 32, 64$ and 128, respectively. With this, we can assess if investing additional effort into the search is advantageous. We also evaluate the impact of extending the search from two to three dimensions, as we expect that considering three dimensions at once improves the solution quality, but also results in larger computational effort.

The framework has been implemented in Java 12. The experiments are run on a multi-core environment with 16 core processors (AMD Epyc 7351 Processors) and 256GB of DDR4-2666 RAM.

4.2 Metrics

The results are evaluated using the following metrics:

Run time [s]: The total run time in seconds provides information about the total run time the GPR sampling framework requires to approximate the set of Pareto-optimal itineraries S_{all}^{opt}.

# of Pareto-optimal solutions: The number of Pareto-optimal solutions reflects the size of set S_{all}^{opt} retrieved by the framework to form the choice set for the traveler.

Improved solutions [%]: This metric compares the Pareto-optimal sets retrieved by our GPR sampling framework to the systematic sampling framework proposed by Horstmannshoff & Ehmke (2022), which serves as a baseline for evaluation. While a value of 0% indicates that no solution at all could be improved by GPR sampling, a value of 100% means that all identified solutions have been improved and hence dominate those solutions found by applying systematic sampling.

Iterations to retrieve Pareto-optimal solution: This metric shows in which iteration a Pareto-optimal solution has been retrieved while sampling the respective two-dimensional set $S_{i,j}$ ($S_{i,j,h}^{3}$ for a

three-dimensional setting). As multiple sets are computed simultaneously, a Pareto-optimal solution can be identified in more than one set. In that case, the smaller iteration is taken for this metric.

4.3 Configuration of Gaussian process regression

The GPR used to actively learn the Pareto front structure during the search and dynamically decide where to sample next requires specific configurations, which are as follows:

Python package: We use the Python package *scikit-learn* with default settings if not stated otherwise to apply the GPR.

Kernel settings: The underlying kernel, which describes the covariance of the Gaussian process random variables, can be customized to embed the underlying assumptions of the complex structure of the Pareto front into the search. We compose our kernel of two additive components: a radial basic function kernel and a linear kernel. The radial basic function kernel is a widely used kernel that makes the assumption that closer points in the solution space are more similar to each other than farther points (E. Schulz et al., 2018). Following Rasmussen & Williams (2006), the radial basic function kernel is also referred to as squared exponential kernel and is defined as follows:

$$k(x_i, x_j) = \exp(-\frac{1}{2}|x_i - x_j|^2). \tag{1}$$

The linear kernel takes into account that the boundary solutions of each min-max-interval are in opposite corners of the solution space:

$$k(x_i, x_j) = {x_i}^T \sum\nolimits_p x'. \tag{2}$$

To ensure reproducible results across multiple runs of the GPR sampling framework, we fix the random number generation used to initialize the centers of the regression to 1234.

5 Computational results

In this section, we present computational results for our GPR sampling framework. First, we show aggregated results across 300 OD pairs in Section 5.1. Then, we perform a detailed analysis of the number of improved solutions to the systematic sampling framework, which serves as the baseline, in Section 5.2. Furthermore, we analyze if these improvements are dependent on the structure of the respective Pareto-optimal solutions. Next, we examine the effectiveness retrieving relevant solutions fast of the GPR sampling framework in Section 5.3. Finally, we conclude with a detailed example of a traveler request from Stuttgart to Erfurt in Section 5.4.

5.1 Summary results

Table 2 shows aggregated results across all analyzed ODs and number of considered traveler preferences P differentiated by two- and three-dimensional setting and the applied sampling density k. These are analyzed in-depth in the following subsections. The creation of two-dimensional sets requires about 5.32 seconds on average. The run time differs in terms of the applied sampling density k. Applying $k = 8$, hence sampling eight times per two-dimensional set $S_{i,j}$, requires 4.11 seconds. If we apply a sampling

Table 2: Summary results for GPR sampling

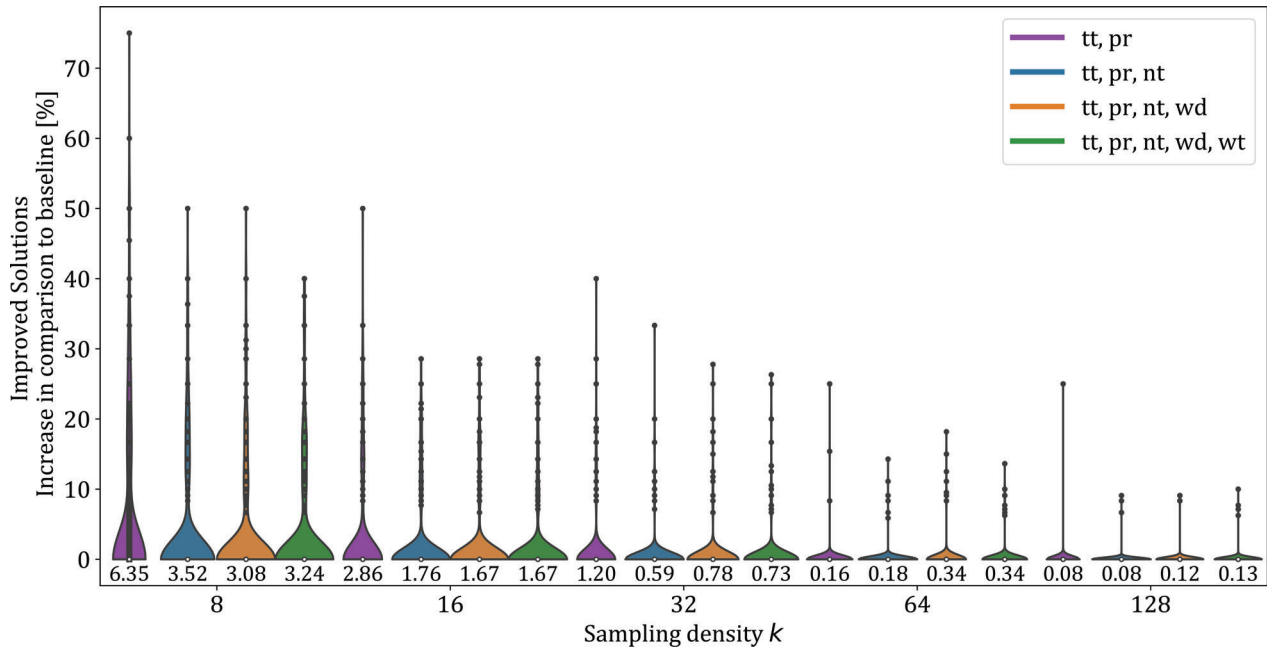
Dim.	Sampling density k	avg. run time [s]		avg. # Pareto-opt. solutions		avg. improved solutions [%]	
two-dim. setting	8	4.11	(114.36%)	4.89	(104.30%)	4.05	(-)
	16	4.98	(113.84%)	5.01	(102.38%)	1.99	(-)
	32	5.57	(111.96%)	5.05	(100.95%)	0.83	(-)
	64	5.91	(111.35%)	5.05	(100.13%)	0.26	(-)
	128	6.05	(110.34%)	5.05	(99.90%)	0.10	(-)
	∅	5.32	(112.37%)	5.01	(101.49%)	1.44	(-)
three-dim. setting	8	12.13	(108.57%)	5.49	(105.85%)	3.76	(-)
	16	15.87	(108.12%)	5.56	(103.07%)	1.78	(-)
	32	19.40	(107.55%)	5.58	(101.15%)	0.53	(-)
	64	22.77	(107.33%)	5.58	(100.24%)	0.09	(-)
	128	28.85	(107.12%)	5.58	(99.96%)	0.01	(-)
	∅	19.80	(107.74%)	5.56	(101.99%)	1.23	(-)

density 16 times as high with $k = 128$, the average run time increases only by about 47% compared to $k = 8$. This is due to the fact that the sampling framework terminates early at high sampling densities as we already have information that no additional solutions can be found with further iterations. Hence, the entire approximated Pareto front is marked as blocked when calculating the respective two-dimensional set. Compared to the systematic sampling framework, the run time increases by approximately 10% resulting from the additional computational effort of actively learning the complex structure of the Pareto front during the search. Investing the additional effort pays off compared to the baseline framework. Applying $k = 8$, with 4.89 Pareto-optimal solutions, about 4% additional solutions are found on average. Doubling the sampling density leads to a decrease of additionally found (102.38%) as well as improving solutions (1.99%). This pattern is also valid with each further doubling of the sampling density. This observation indicates that applying the GPR sampling framework is, in particular, promising if only a limited number of runs is available.

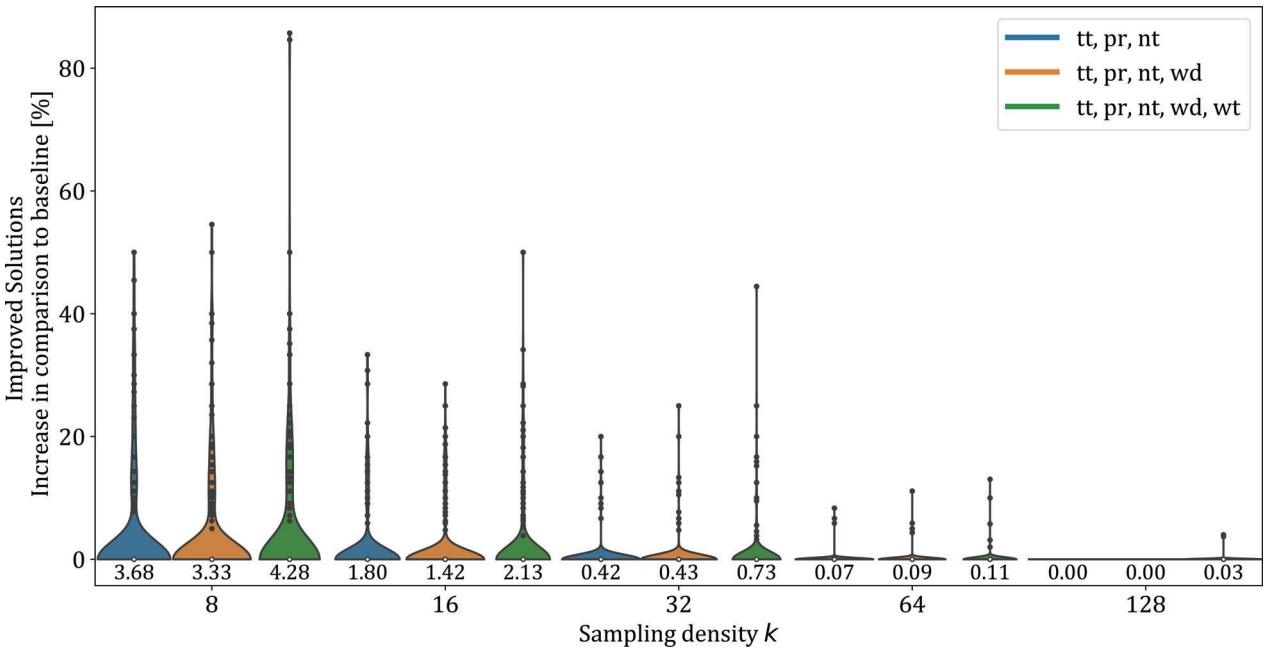
Extending the search to three dimensions results in a significant increase in run time compared to two-dimensional sampling as we now alter upper-bound constraints for two preferences j and h simultaneously when creating the three-dimensional sets $S^3_{i,j,h}$. While we require about 12 seconds on average for $k = 8$, this metric increases to about 29 seconds for a very fine-grained search with $k = 128$. In contrast to the systematic sampling framework, the required run time increases by about 7 to 8 percent. In terms of the average number of retrieved Pareto-optimal solutions and the proportional increase in improved solutions a comparable pattern to the two-dimensional analysis can be examined. Especially for smaller sampling densities k, these metric values increase, while almost no change at all can be observed for high $k = 128$.

5.2 Improved solutions

In the following, we deepen the analysis with respect to improved solutions. Results aggregated across all 300 ODs are shown in Figure 6 with sampling density k in increasing number along the x-axis. In addition, we differentiate the results by the respective set of considered preferences P. The violin plots can be interpreted as follows: Each black dot represents a separate data point, while the average is given as a number under the respective plot. The width in the violin plot represents the number of data points for the respective y-value area.



(a) Improved solutions: Two-dimensional setting



(b) Improved solutions: Three-dimensional setting

Figure 6: Improved solutions in comparison to the baseline

Figure 6(a) shows how many improved solutions have been found in comparison to the systematic sampling framework for two-dimensional sampling. For instance, when setting the sampling density to $k = 8$ and considering two preferences in the search (tt, pr), on average 6.35% of the identified Pareto-optimal solutions by the GPR sampling framework are dominating those found by the systematic sampling framework (left purple plot). It is apparent that with each doubling of the sampling density k fewer improved solutions in percentage terms are found with decreasing impact. Limiting the set of considered preferences P yields a higher average number of improved solutions. While, for instance, on average 2.86% improved solutions can be seen for $k = 16$ and $P := \{\text{tt, p}\}$, merely 1.67% improved solutions occur in case five preferences are considered simultaneously (tt, pr, nt, wd, wt). Comparable observations can be made for three-dimensional sampling in Figure 6(b). Again, the number of analyzed improved

solutions decreases with every doubling of the configured sampling density value. For instance, when taking three preferences at the same time into account (tt, pr, nt), on average 3.68% improved solutions result while no improved solutions at all occur when sampling fine-grained with $k = 128$. Furthermore, extending the sampling process to three dimensions yields slightly more improved solutions in comparison to two dimensions only as more potential configurations are analyzed in the multimodal solution space. In general, we observe that improved solutions are found in particular for small sampling densities. In this context, large differences can be observed depending on the respective OD. While we do not find any improving solutions at all for the majority of the analyzed ODs, up to 85.71% improved solutions can be examined for other ODs. We conclude that actively learning during the search makes sense and is competitive with the baseline approach.

Table 3: Baseline solutions enhanced by improved solutions differentiated by preference

Dim.	Sampling density k	Travel time	Price	Number of transfers	Walking distance	Waiting time
two-dim. setting	8	11.92%	7.71%	2.91%	6.88%	9.46%
	16	7.30%	5.18%	1.17%	4.07%	5.35%
	32	3.17%	2.48%	0.86%	2.21%	2.43%
	64	1.38%	0.96%	0.73%	1.44%	1.79%
	128	0.60%	0.43%	0.55%	0.69%	0.74%
	∅	4.87%	3.35%	1.24%	3.06%	3.95%
three-dim. setting	8	11.36%	6.82%	2.55%	7.38%	10.20%
	16	6.83%	4.36%	1.31%	4.68%	6.06%
	32	2.55%	1.49%	0.54%	2.00%	2.57%
	64	0.79%	0.34%	0.29%	0.82%	0.80%
	128	0.15%	0.05%	0.09%	0.12%	0.40%
	∅	4.34%	2.61%	0.96%	3.00%	4.01%

Next, in Table 3, we analyze if the observed improved solutions enhance those solutions from the set of baseline solutions with respect to certain preferences. The results are aggregated across all analyzed OD pairs as well as considered sets of traveler preferences P and differentiated by the underlying dimension as well as the applied sampling density k. For instance, for two-dimensional sampling and $k = 8$, on average 11.92% of the baseline solutions could be improved with respect to travel time. This value decreases with every doubling of the configured sampling density. Aggregated across all parameter values for k, 4.87% of the baseline solutions could be improved in terms of travel time. On average, the second highest improvement is seen in waiting time with about 4%, followed by price and walking distance with 3.35% and 3.06%, respectively. Only for number of transfers as the solely considered traveler preference, which is subject to an integer constraint, merely 1.24% solutions could be improved. Similar trends can be observed in the three-dimensional setting. While on average 4.34% of the baseline solutions have been improved in terms of travel time, merely about 1% could be improved regarding the number of transfers. We conclude that applying the GPR sampling framework is in particular promising to retrieve improving solutions with additional value to the traveler in terms of continuous traveler preferences, esp. travel and waiting times.

5.3 Effectiveness of GPR sampling

In the following, we examine the effectiveness of the GPR sampling framework in comparison to the systematic sampling framework. Table 4 presents the required iterations to retrieve Pareto-optimal

Table 4: Avg. iteration to retrieve Pareto-optimal solutions: Two-dimensional setting

sampling density k	GPR: avg. iteration to retrieve Pareto-optimal solutions	Systematic sampling (baseline): avg. iteration to retrieve Pareto-optimal solutions
8	1.81	2.38
16	2.12	2.88
32	2.31	3.66
64	2.33	5.10
128	2.33	7.93

solutions averaged across all analyzed 300 OD pairs as well as considered traveler preferences P. For $k = 8$, we require on average 1.81 iterations to find a Pareto-optimal solution while creating two-dimensional sets $S_{i,j}$. To find the respective solution applying the systematic sampling approach, we need on average 2.38 iterations. Hence, Pareto-optimal solutions are approximated faster when actively learning the area with the highest uncertainty in the complex multimodal solution space. When increasing the sampling density to $k = 16$, the average iteration to approximate Pareto-optimal solutions increases to 2.12 when applying the GPR sampling framework. With every further doubling of k, it increases only slightly up to 2.31 (for $k = 32$) and remains equal at 2.33 for $k = 64$ and $k = 128$, respectively. Hence, we examine that putting additional effort has no value anymore, while the average iteration to approximate Pareto-optimal solutions when applying the systematic sampling framework still increases with every doubling of the sampling density k. We conclude that actively learning characteristics of the Pareto front structure enhances the systematic sampling framework significantly.

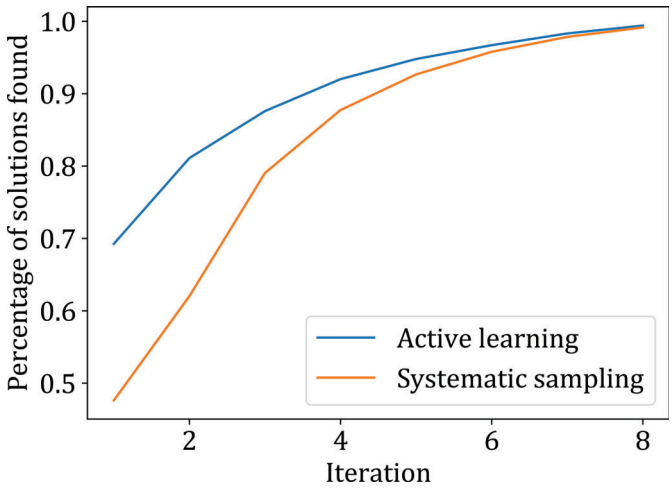
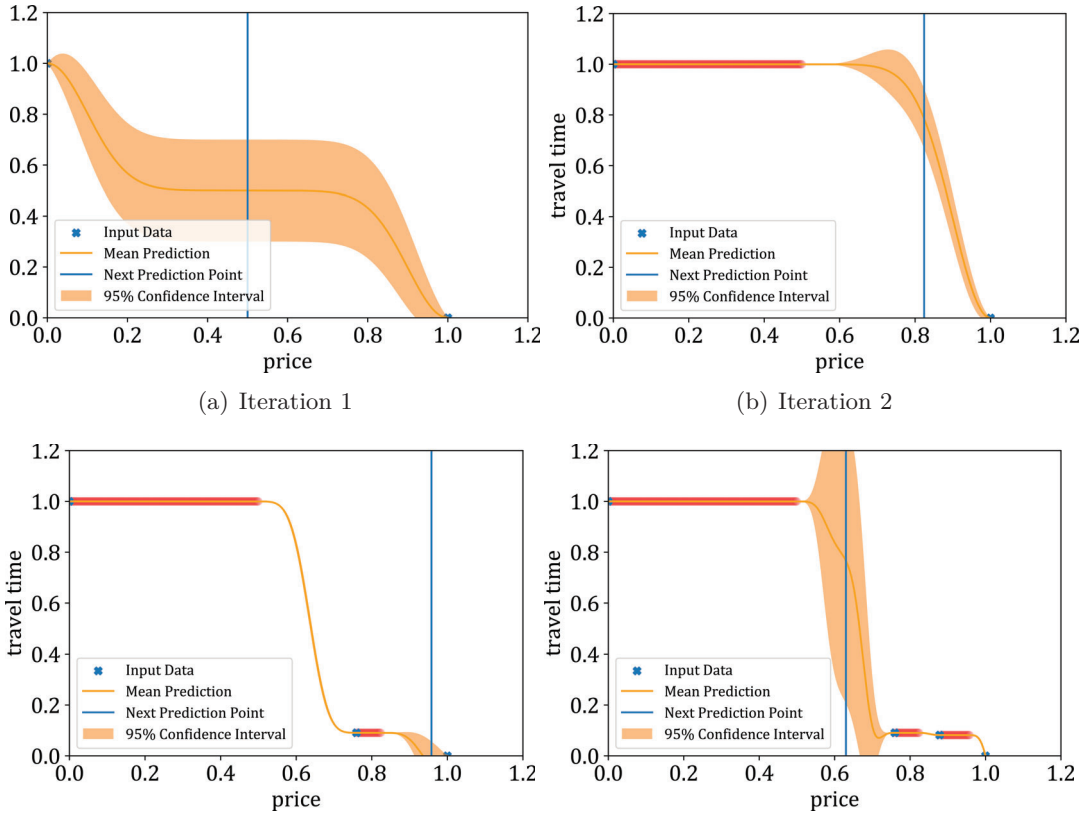


Figure 7: Accumulated average iteration to retrieve Pareto-optimal solutions

Figure 7 presents further insights into the average number of required iterations to approximate Pareto-optimal solutions for a sampling density of $k = 8$ applied in a two-dimensional setting. The number of iterations is shown on the x-axis. Accumulated information in which iteration proportionally how many Pareto-optimal solutions are found are depicted on the y-axis. When applying systematic sampling (orange line), about 50% of the Pareto-optimal solutions are retrieved already in the first iteration. For the GPR sampling framework, even about 70% of the Pareto-optimal solutions are found in the first iteration (blue line). In the next iterations, we examine that applying the GPR sampling framework results in faster convergence of approximating the Pareto-optimal solutions. Comparable observations are applicable for further sampling densities of $k = 16, 32, 64, 128$ as shown in Figure 10 in Appendix A.

5.4 OD-specific example



(a) Iteration 1 (b) Iteration 2

(c) Iteration 3 (d) Iteration 4

Figure 8: GPR for set $S_{travelTime,price}$

Finally, we demonstrate the GPR sampling framework for the specific OD pair of Stuttgart to Erfurt, Germany, in detail. In this two-dimensional setting, we take five preferences into account and set the sampling density to $k = 8$.

Figure 8 provides visualizations for the first four iterations when calculating the set $S_{travelTime,price}$ for the exemplary OD pair. Already retrieved solutions are depicted by blue crosses. The orange line indicates the mean prediction of the GPR, while the orange highlighted area indicates the 95% confidence interval. The area with the largest confidence interval – thus the highest uncertainty – is then used as the next sampling point and marked with a blue vertical line. Areas in which no additional information is expected are marked as blocked and highlighted in red. Figure 8(a) presents the GPR for the first iteration. As only the boundary solutions of the min-max-interval are known, we set the next sampling point in the middle of the identified interval for price. Rerunning the multimodal routing algorithm with this setting yields a solution that equals the left boundary solution already retrieved while identifying the min-max-interval. Hence, the area between between 0.0 and 0.5 is marked as blocked as can be seen in Figure 8(b). Then, GPR is used to identify the area with the highest uncertainty in the multimodal solution space. The respective mean prediction as well as the resulting 95% confidence interval are highlighted in orange in Figure 8(b). Following, the next sampling point is set to about 0.8 of the 0-1-normalized price interval. Subsequently, this process is repeated until either all solutions ranging from the respective lower-interval bound to the respective upper-interval bound are marked as blocked or k iterations have been run through.

Figure 9 presents the solutions identified using the GPR sampling framework in comparison to the systematic sampling framework. While the solution labeled “Solution 1” to “Solution 5” are retrieved

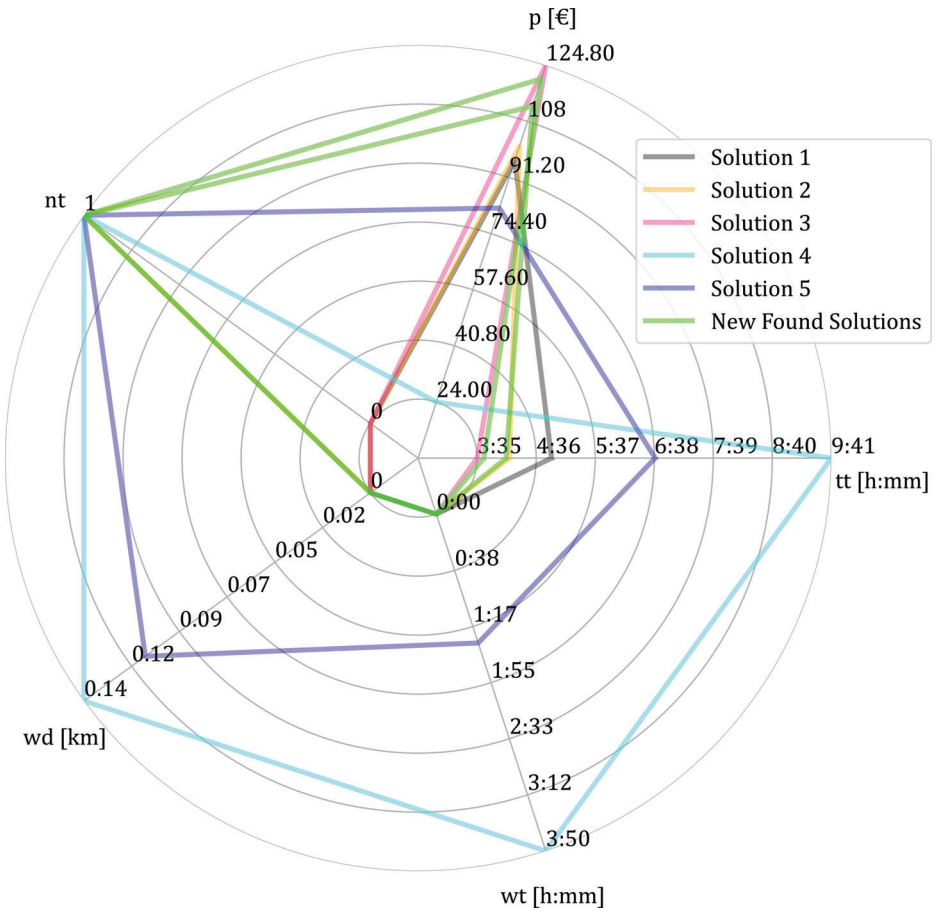


Figure 9: Radar plot for the request from Stuttgart to Erfurt

by both frameworks, two additional solutions are found by actively learning the structure of the Pareto front during the search (highlighted in light green in Figure 9). These two additional solutions represent a trade-off compared to the solution with the lowest travel time (highlighted in red); both solutions are slightly cheaper (up to €12.25), but require a bit more travel time and one more transfer.

6 Conclusion

In recent years, the importance of decision support for the planning of multimodal door-to-door itineraries has increased. However, one major challenge when identifying a choice set for the traveler is the integration of multiple individual traveler preferences into the search. Modern MCDM approaches struggle to scale efficiently in terms of run time when more than three traveler preferences are considered. In this work, we have enhanced a recent systematic sampling framework proposed by Horstmannshoff & Ehmke (2022). By sampling many lower-dimensional sets simultaneously, we approximate the high-dimensional Pareto fronts efficiently and hence ensure scalability. In particular, we learn the structure of the Pareto front actively during the search and thereby guide the search dynamically to promising parts of the complex multimodal solution space.

The GPR sampling framework has been evaluated analyzing long-distance itineraries between major cities in Germany embedding a large amount of real-world data from multiple mobility services. To evaluate the scalability of the proposed framework, we have integrated up to five traveler preferences – travel time, price, number of transfers, walking distance, and waiting time – into the search. In addition, we have analyzed the impact of different sampling densities.

We have examined significant improvements compared to the baseline framework by Horstmannshoff & Ehmke (2022). In particular, if only a few iterations (small sampling density) are available, dynamically guiding the sampling process to areas with highest uncertainty is promising. The potential improvement is thereby highly OD-dependent. Therefore, we conclude that multi-criteria itinerary planning with GPR shows added value for the traveler.

In future work, we plan to integrate more realistic price information such as tariff fare structures and advance booking periods to design the framework even more realistically. In addition, analyzing the impact of integrating additional traveler preferences such as reliability and sustainability into the framework adds additional value for the traveler. Furthermore, we plan to evaluate the proposed framework against MCDM frameworks from the literature that determine the full Pareto-optimal set of multimodal itineraries. These algorithms do not scale effectively when multiple traveler preferences are taken into account, as stated in Section 2, whereas our introduced GPR sampling framework ensures scalability while approximating the set of Pareto-optimal itineraries. This comparison would give us insights how close we are to the computation expensive full Pareto-optimal set.

References

Alt, R., Ehmke, J. F., Haux, R., Henke, T., Mattfeld, D. C., Oberweis, A., ... Winter, A. (2019). Towards customer-induced service orchestration - requirements for the next step of customer orientation. *Electronic Markets*, *29*(1), 79–91. doi: 10.1007/s12525-019-00340-3

Bast, H., Delling, D., Goldberg, A., Müller-Hannemann, M., Pajor, T., Sanders, P., ... Werneck, R. F. (2015). Route planning in transportation networks. In *Algorithm Engineering.* doi: https://doi.org/10.48550/arXiv.1504.05140

Deb, K. (2011). Multi-objective optimisation using evolutionary algorithms: An introduction. In L. Wang, A. H. C. Ng, & K. Deb (Eds.), *Multi-objective Evolutionary Optimisation for Product Design and Manufacturing* (pp. 3–34). London: Springer London. doi: 10.1007/978-0-85729-652-8_1

Delling, D., Dibbelt, J., Pajor, T., Wagner, D., & Werneck, R. F. (2013). Computing multimodal journeys in practice. In D. Hutchison, T. Kanade, & J. Kittler (Eds.), *Experimental Algorithms* (Vol. 7933, pp. 260–271). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-642-38527-8_24

Deringer, V. L., Bartók, A. P., Bernstein, N., Wilkins, D. M., Ceriotti, M., & Csányi, G. (2021). Gaussian process regression for materials and molecules. *Chemical Reviews*, *121*(16), 10073–10141. doi: 10.1021/acs.chemrev.1c00022

Dib, O., Manier, M.-A., Moalic, L., & Caminada, A. (2017). A multimodal transport network model and efficient algorithms for building advanced traveler information systems. *Transportation Research Procedia*, *22*, 134–143. doi: 10.1016/j.trpro.2017.03.020

Esztergár-Kiss, D., & Csiszár, C. (2015). Evaluation of multimodal journey planners and definition of service levels. *International Journal of Intelligent Transportation Systems Research*, *13*(3), 154–165. doi: 10.1007/s13177-014-0093-0

Geisberger, R., Sanders, P., Schultes, D., & Delling, D. (2008). Contraction hierarchies: Faster and simpler hierarchical routing in road networks. In C. C. McGeoch (Ed.), *Experimental Algorithms* (pp. 319–333). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-540-68552-4_24

Grotenhuis, J.-W., Wiegmans, B. W., & Rietveld, P. (2007). The desired quality of integrated multimodal travel information in public transport: Customer needs for time and effort savings. *Transport Policy*, *14*(1), 27–38. doi: 10.1016/j.tranpol.2006.07.001

He, P., Jiang, G., Lam, S.-K., Sun, Y., & Ning, F. (2022). Exploring public transport transfer opportunities for pareto search of multicriteria journeys. *IEEE Transactions on Intelligent Transportation Systems*, 1-14. doi: 10.1109/TITS.2022.3194523

Herzel, A., Ruzika, S., & Thielen, C. (2021). Approximation methods for multiobjective optimization problems: A survey. *INFORMS Journal on Computing*. doi: 10.1287/ijoc.2020.1028

Horstmannshoff, T. (2022). Mobility-as-a-Service-Plattformen – Berücksichtigung von komplexen Reisendenanforderungen mittels nutzerorientierter Algorithmen. In M. Bruhn & K. Hadwich (Eds.), *Smart Services* (pp. 523–546). [S.l.]: GABLER. doi: 10.1007/978-3-658-37346-7_19

Horstmannshoff, T., & Ehmke, J. F. (2022). Traveler-oriented multi-criteria decision support for multimodal itineraries. *Transportation Research Part C: Emerging Technologies*, *141*, 103741. doi: https://doi.org/10.1016/j.trc.2022.103741

Liang, J., Zang, G., Liu, H., Zheng, J., & Gao, Z. (2023). Reducing passenger waiting time in oversaturated metro lines with passenger flow control policy. *Omega*, *117*(1), 102845. doi: 10.1016/j.omega.2023.102845

Lyons, G., Hammond, P., & Mackay, K. (2020). Reprint of: The importance of user perspective in the evolution of maas. *Transportation Research Part A: Policy and Practice*, *131*, 20–34. doi: 10.1016/j.tra.2019.11.024

Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. Cambridge, Mass. and London: MIT Press.

Musolino, G., Rindone, C., Vitale, A., & Vitetta, A. (2023). Pilot survey of passengers' preferences in Mobility as a Service (MaaS) scenarios: a case study. *Transportation Research Procedia*, *69*, 328–335. doi: 10.1016/j.trpro.2023.02.179

Pajor, T. (2009). *Multi-modal route planning* (Diploma Thesis). Karlsruhe Institute of Technology, Karlsruhe.

Palm, N., Landerer, M., & Palm, H. (2022). Gaussian process regression based multi-objective bayesian optimization for power system design. *Sustainability*, *14*(19), 12777. doi: 10.3390/su141912777

Potthoff, M., & Sauer, J. (2022). Efficient algorithms for fully multimodal journey planning. In *22nd Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS 2022). Ed.: Mattia D'Emidio* (Vol. 106, p. 14). Schloss Dagstuhl - Leibniz-Zentrum für Informatik GmbH (LZI). doi: 10.4230/OASIcs.ATMOS.2022.14

Randelhoff, M. (2022). *Innerdeutsches Reisen: Reisezeit- und Preisvergleiche zwischen Bahn, Pkw, Flugzeug und Fernbus, url = https://www.zukunft-mobilitaet.net/172255/analyse/preisvergleich-bahn-flugzeug-fernbus-pkw-auto-reisezeit-dauer-kosten/.*

Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian processes for machine learning*. Cambridge, Mass. and London: MIT. doi: 10.7551/mitpress/3206.001.0001

Samaranayake, S., Blandin, S., & Bayen, A. (2011). A tractable class of algorithms for reliable routing in stochastic networks. *Procedia - Social and Behavioral Sciences*, *17*(1), 341–363. doi: 10.1016/j.sbspro.2011.04.521

Schöbel, A., & Urban, R. (2022). The cheapest ticket problem in public transport. *Transportation Science*, *56*(6), 1432–1451. doi: 10.1287/trsc.2022.1138

Schulz, E., Speekenbrink, M., & Krause, A. (2018). A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions. *Journal of Mathematical Psychology*, *85*, 1–16. doi: 10.1016/j.jmp.2018.03.001

Schulz, T., Gewald, H., Böhm, M., & Krcmar, H. (2020). Smart mobility: Contradictions in value co-creation. *Information Systems Frontiers*. doi: 10.1007/s10796-020-10055-y

Spickermann, A., Grienitz, V., & von der Gracht, H. A. (2014). Heading towards a multimodal city of the future? *Technological Forecasting and Social Change*, *89*, 201–221. doi: 10.1016/j.techfore.2013.08.036

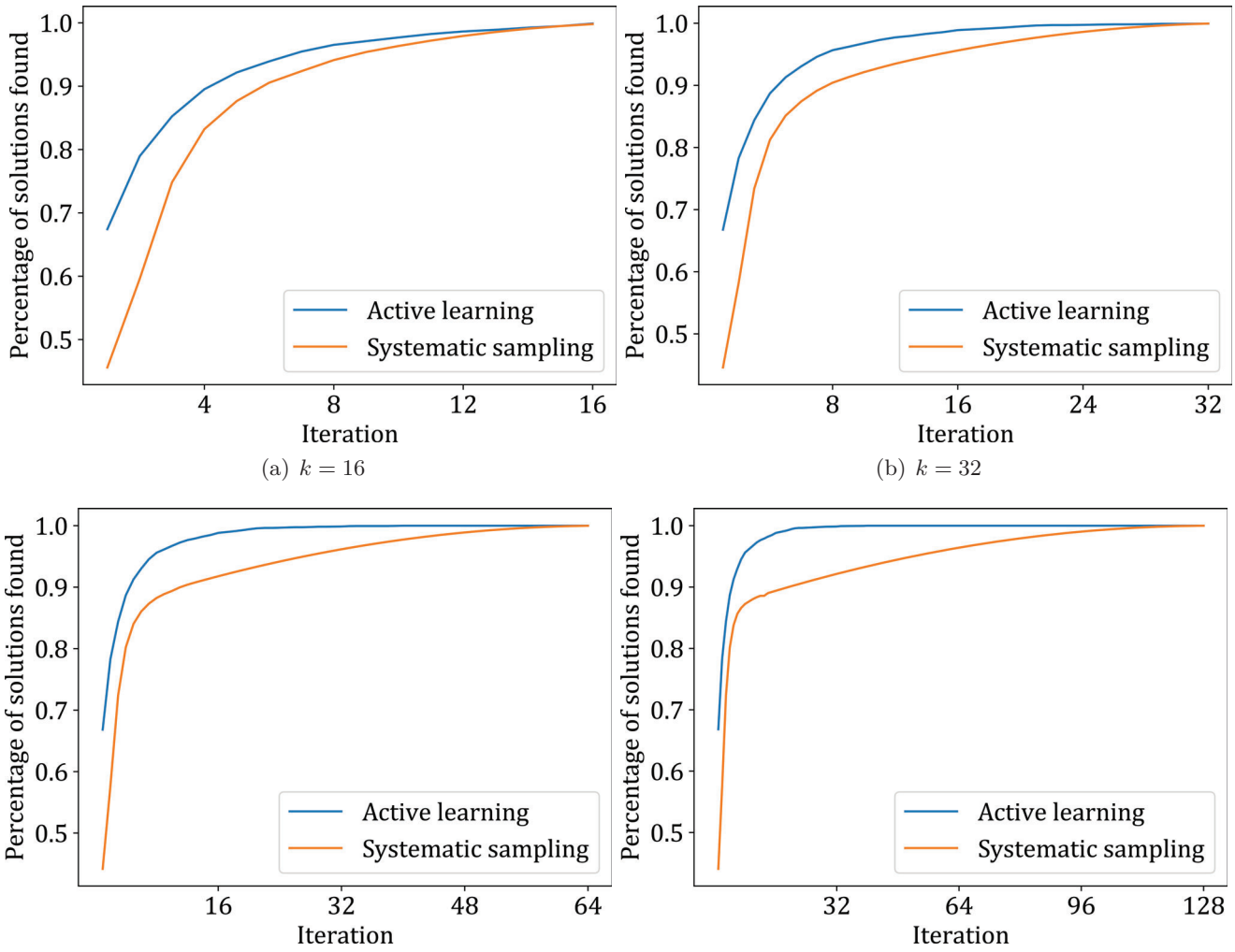
Stopka, U. (2014). Identification of user requirements for mobile applications to support door-to-door mobility in public transport. In D. Hutchison, T. Kanade, & J. Kittler (Eds.), *Human-Computer Interaction. Applications and Services* (Vol. 8512, pp. 513–524). Cham: Springer International Publishing. doi: 10.1007/978-3-319-07227-2_49

Stopka, U., Fischer, K., & Pessier, R. (2016). Evaluation methods and results for intermodal mobility applications in public transport. In M. Kurosu (Ed.), *Human-Computer Interaction. Novel User Experiences* (pp. 343–354). Cham: Springer International Publishing. doi: 10.1007/978-3-319-39513-5_32

Valderas, P., Torres, V., & Pelechano, V. (2020). Towards the composition of services by end-users. *Business & Information Systems Engineering*, *62*(4), 305–321. doi: 10.1007/s12599-019-00617-z

Wu, L., Kang, J. E., Chung, Y., & Nikolaev, A. (2021). Inferring origin-destination demand and user preferences in a multi-modal travel environment using automated fare collection data. *Omega*, *101*(4), 102260. doi: 10.1016/j.omega.2020.102260

A Accumulated average iteration to retrieve Pareto-optimal solutions



(a) $k = 16$ (b) $k = 32$

(c) $k = 64$ (d) $k = 128$

Figure 10: Accumulated average iteration to retrieve Pareto-optimal solutions

Otto von Guericke University Magdeburg
Faculty of Economics and Management
P.O. Box 4120 | 39016 Magdeburg | Germany

Tel.: +49 (0) 3 91 / 67-1 85 84
Fax: +49 (0) 3 91 / 67-1 21 20

www.fww.ovgu.de/femm

ISSN 1615-4274