Fakultät für Informatik
Technische Universität München

A User-Centered Approach to Solving the Tourist Trip Design Problem for Individuals and Groups

Daniel Andreas Herzog

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender:
Prof. Dr. Martin Bichler

Prüfende der Dissertation:
1. Prof. Dr. Johann Schlichter
2. Prof. Dr. Nava Tintarev

Die Dissertation wurde am 27.04.2020 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 18.06.2020 angenommen.

Abstract

Tourists exploring a city often look for sequences of points of interest, such as museums, restaurants, and parks, that they can visit along a route. The problem of identifying such tourist trips is called the Tourist Trip Design Problem (TTDP). Many algorithms and heuristics solving the TTDP have been developed in the past, but only few of them try to find the best trip from a user-centered perspective. Recommender Systems (RSs) in tourism have to take into account the user's context and real needs to provide accurate and useful recommendations. In addition, they need to be integrated into practical applications that are a pleasure to use in order to support users in finding desired results. In practice, tourists often travel in groups, which complicates the problem of finding satisfying recommendations. RSs for groups have to consider the preferences of all group members and provide recommendations that are perceived as useful and fair by the whole group.

In this thesis, we show how to solve the TTDP for individuals and groups from a user-centered perspective. For this purpose, we suggest different extensions to a state-of-the-art tourist trip algorithm that allow a more realistic modeling of the TTDP, such as context-aware recommendations. In addition, we present platforms and user interfaces for the integration of TTDP algorithms into practical applications that help tourists in finding the best trip. We demonstrate how to extend our approach to solve the TTDP not only for individuals, but also groups of users. A key contribution of this thesis is the evaluation of all of our proposed solutions in user studies. Furthermore, we have conducted a large user study showing how real groups make travel-related decisions and how they can be supported in finding satisfying recommendations. The results of this thesis are supposed to facilitate the development of RSs in the tourism domain and improve the quality of tourist trip recommendations for individuals and groups.

Zusammenfassung

Touristen, die eine Stadt erkunden, suchen oft nach einer Folge von Sehenswürdigkeiten, wie z.B. Museen, Restaurants und Parks, welche sie entlang einer Route besuchen können. Das Problem der Identifizierung solcher Touristenreisen bezeichnet man als Tourist Trip Design Problem (TTDP). In der Vergangenheit wurden viele Algorithmen und Heuristiken zur Lösung des TTDP entwickelt, doch nur wenige dieser Ansätze versuchen, die beste Reise aus Benutzersicht zu finden. Empfehlungssysteme in der Tourismusbranche müssen den Kontext und die tatsächlichen Bedürfnisse des Benutzers berücksichtigen, um akkurate und nützliche Empfehlungen bereitzustellen. Darüber hinaus müssen sie in praktische Anwendungen integriert werden, die mit Freude benutzt werden können, um die Benutzer bei der Suche nach den gewünschten Ergebnissen zu unterstützen. In der Praxis reisen Touristen oft in Gruppen, was das Problem, zufriedenstellende Empfehlungen zu finden, erschwert. Empfehlungssysteme für Gruppen müssen die Präferenzen aller Gruppenmitglieder berücksichtigen und Empfehlungen generieren, die von der gesamten Gruppe als nützlich und fair empfunden werden.

In dieser Arbeit zeigen wir, wie das TTDP für Einzelpersonen und Gruppen aus Benutzersicht gelöst werden kann. Zu diesem Zweck schlagen wir verschiedene Erweiterungen eines modernen Touristenreise-Algorithmus vor, die eine realistischere Modellierung des TTDP ermöglichen, wie z.B. kontextsensitive Empfehlungen. Darüber hinaus präsentieren wir Plattformen und Nutzerschnittstellen für die Integration von TTDP Algorithmen in praktische Anwendungen, die Touristen bei der Suche nach der besten Reise helfen. Wir zeigen, wie wir unseren Ansatz erweitern können, um das TTDP nicht nur für Einzelpersonen, sondern auch für Gruppen zu lösen. Ein wesentlicher Beitrag dieser Arbeit ist die Evaluierung aller unserer Lösungsvorschläge in Benutzerstudien. Darüber hinaus haben wir eine große Benutzerstudie durchgeführt, die zeigt, wie echte Gruppen reisebezogene Entscheidungen fällen und wie diese bei der Suche nach zufriedenstellenden Empfehlungen unterstützt werden können. Die Ergebnisse dieser Arbeit sollen die Entwicklung von Empfehlungssystemen in der Tourismusbranche erleichtern und die Qualität von touristischen Reiseempfehlungen für Einzelpersonen und Gruppen verbessern.

Acknowledgments

This thesis would not have been possible without the support of many people.

I would like to express my sincere gratitude to my supervisor Prof. Dr. Johann Schlichter who gave me the opportunity to become a doctoral candidate and write this thesis. His consistent support and guidance during the last years steered me through this research. I am also extremely grateful to Asst. Prof. Dr. Nava Tintarev for being my second supervisor and her invaluable advice and feedback on my thesis.

I would like to thank all present and former colleagues at the Chair of Applied Informatics / Cooperative Systems and the Chair of Connected Mobility for their helpful advices and fruitful discussions during coffee breaks and doctoral seminars. I am particularly indebted to Prof. Dr.-Ing. Jörg Ott for his support and the great working atmosphere at the chair. Special thanks also to Dr. Wolfgang Wörndl who introduced me to the field of recommender systems already during my studies. This thesis would not have been possible without his input and the feedback he provided on my work. Furthermore, I would like to express my thanks to all the students and researchers who collaborated with me during the last years. I would like to highlight the contribution of Christopher Laß who supported me as a student assistant and co-author. Working with him was always fun and inspiring. My thanks also go to all participants of my user studies.

Last but not least, many thanks to my family and friends for their patience, support, and love.

Contents

List of Figures

List of Tables

Acronyms

AOP	Arc Orienteering Problem.
AP	Aggregating Profiles of Users.
API	Application Programming Interface.
AR	Aggregating Recommendations.
CAQI	Common Air Quality Index.
CARS	Context-Aware Recommender System.
CBRS	Content-Based Recommender System.
CF	Collaborative Filtering.
COP	Clustered Orienteering Problem.
CTPP	Cycle Trip Planning Problem.
DBDD	Dijkstra-based with Distance to Destination.
DPTOP	Team Orienteering Problem with Decreasing Profits.
DUI	Distributed User Interface.
GOP	Generalized Orienteering Problem.
GRASP	Greedy Randomised Adaptive Search Procedure.
GRS	Group Recommender System.
HCI	Human-Computer Interaction.
HTTP	Hypertext Transfer Protocol.
ILS	Iterated Local Search.
IR	Information Retrieval.
LBSN	Location-based Social Network.
MCMTOPTW	Multi Constrained Multiple Team Orienteering Problem with Time Windows.
MCTOPMTW	Multi Constrained Team Orienteering Problem with Multiple Time Windows.
MCTOPTW	Multi Constrained Team Orienteering Problem with Time Windows.
MTG	Mobile Tourist Guide.
OP	Orienteering Problem.
OPMPC	Orienteering Problem with Maximum Point Categories.
OPSP	Orienteering Problem with Stochastic Profits.

OPSTS	Orienteering Problem with Stochastic Travel and Service Times.
OPTW	Orienteering Problem with Time Windows.
OPVP	Orienteering Problem with Variable Profits.
OR	Operations Research.
PCC	Pearson Correlation Coefficient.
POI	point of interest.
ResQue	Recommender systems' Quality of user experience.
RQ	research question.
RS	Recommender System.
SUS	System Usability Scale.
TDOP	Time Dependent Orienteering Problem.
TDTOPTW	Time Dependent Team Orienteering Problem with Time Windows.
TOP	Team Orienteering Problem.
TOPTW	Team Orienteering Problem with Time Windows.
TTDP	Tourist Trip Design Problem.
UEQ	User Experience Questionnaire.
UI	user interface.
UX	user experience.

1 Introduction

According to the World Travel and Tourism Council [1], travel and tourism is one of the most important sectors for economic development. In 2018, the sector contributed directly or indirectly US$ 8.8 trillion to the global economy and supported 292 million jobs which is equal to 10.4% of the world's gross domestic product and 1 in 10 of all jobs. The number of tourist arrivals is expected to reach 2.2 bn by 2029.

Large cities offer an almost infinite number of points of interest (POIs) that tourists can visit on a trip, such as museums, restaurants, and parks. Visiting all POIs is usually not feasible due to constraints, such as the time or budget available for the trip. Consequently, when planning a trip, tourists are looking for the POIs that are of most interest to them. However, identifying attractive POIs in an unfamiliar area is a challenge. Gathering information from tourist guides, magazines, or travel-related websites is time-consuming. Browsing a lot of data makes it difficult to make decisions and increases the risk of missing attractive POIs. This so-called *information overload* problem justifies the demand for tools to filter large amounts of data when searching for information that is of great interest to the user [2]. Recommender Systems (RSs) are an example of such information filtering tools which are already very popular in the field of tourism [3].

Recommending "physical" items, such as POIs, is more challenging than recommending digital products, such as movies [4]: Users have to physically visit POIs to know and rate them. The cost of visiting a POI is more expensive than watching a movie. If a POI can be visited depends also on many contextual factors, such as the time, location, and weather. For instance, outdoor activities are less suitable on a rainy day. Furthermore, travel preferences can change over time. A family is more interested in visiting parenting-related POIs, such as playgrounds, after having a baby.

Identifying the most attractive POIs is only the first step of travel planning. When exploring a city on a single- or multi-day trip, users usually want to visit POIs along an enjoyable and feasible route. Ideally, this route contains the most attractive POIs and respects user constraints related to travel cost and trip duration while avoiding inappropriate detours. Another critical aspect of trip planning is the order in which POIs are visited. On the one hand, it has been shown that human movement follows reproducible patterns [5]. For instance, many people enjoy going to a bar for a drink after having dinner at a restaurant. On the other hand, travelers want to avoid unpleasant combinations of POIs, such as visiting two restaurants within a short time frame. Likewise, POIs should be recommended at the right time during a trip. A restaurant is more appreciated during midday or in the evening, for example. Nevertheless, the order of POIs in a tourist trip is not as flexible as the order of songs in a playlist, for example, because the location of the recommended POIs becomes a limiting factor.

The problem of finding tourist trips that respect the aforementioned constraints is called the Tourist Trip Design Problem (TTDP) [6]. The current generation of so-called Mobile Tourist Guides (MTGs)[1] is supposed to not only identify travel items of interest to the user, but also solve the TTDP by providing routing features to combine POIs along enjoyable routes. Tourist trip RSs that take into account contextual factors are called Context-Aware Recommender Systems (CARSs) [8].

In practice, tourists often travel in groups. This complicates the problem of finding a tourist trip since group members often have different opinions on what to visit on a trip. The travel preferences and constraints of all group members should be considered when agreeing on a tourist trip to satisfy all group members. Different strategies for aggregating user preferences exist; however, there is no single perfect one. Instead, the group's intrinsic characteristics and the problem's nature have to be considered [9].

Group Recommender Systems (GRSs) support the decision-making of groups of users. They acquire the group members' preferences, use a group recommendation strategy to come up with recommendations, present recommendations to the group, and support the group members in making a final choice [10]. Thereby, not only the selected group recommendation strategy, but also the way how groups interact with the GRS plays an important role on the way to reach a consensus in a group. For instance, a recommendation could be presented on a shared display which may facilitate an open discussion between the group members. However, in this case, the interaction could be dominated by few group members and some people may feel uncomfortable when revealing their preferences to others. An alternative is to keep private data on personal devices, such as smartphones, and use the shared display solely to present the final recommendation to the group. Many factors can influence a group's choice of the preferred GRS configuration, for instance, the group type and the relationship between the group members.

1.1 Problem Statement

Much research has been done to recommend travel items, such as POIs, to users. Today, the focus is shifting towards recommending tourist trips, that is, personalized sequences of POIs along feasible and enjoyable routes. While most TTDP approaches tackle the problem of finding the best tourist trip from a pure Operations Research (OR) perspective, little research has been done to solve the TTDP from a user-centered perspective taking into account typical aspects of RSs. For instance, existing approaches do not consider the fact that the perceived value of a POI or trip can differ between users and highly depends on the context of the recommendation. Furthermore, only few TTDP works have been integrated in practical applications and evaluated in user studies with real users or groups. Consequently, current tourist trip RSs are not necessarily a pleasure to use or do not satisfy the users' true needs.

Until today, TTDP research focuses only on recommendations for individuals and does not consider that tourists often travel in groups. Existing GRSs recommend POIs to

[1] In published literature, MTGs are also known as (personalized) electronic tourist guides and personal navigation systems for tourism [7].

groups, but do not combine them along routes that satisfy all group members. A deep understanding of how groups behave when interacting with GRSs, especially in public places while traveling, is still missing. The result are GRS configurations that either provide a poor user experience (UX) or generate recommendations that are perceived as unfair or inaccurate.

In summary, many approaches exist to combine POIs along routes. However, these approaches often lack the user focus. They either do not tailor the recommendations to the users' needs or consider how individuals and groups interact with tourist trip RSs. User-centered approaches to solving the TTDP are required to overcome these problems. They should be integrated into practical applications and evaluated in user studies to measure their utility when used by real users and groups.

1.2 Goals of the Thesis

In this thesis, we wanted to tackle the aforementioned problems. Our main goal was to increase the satisfaction of individuals and groups interacting with tourist trip RSs. For this purpose, we wanted to answer the following research question (RQ):

How can RSs solve the TTDP for individuals and groups of users from a user-centered perspective?

Based on the description of the identified problems, we broke down the main RQ into smaller subproblems:

RQ 1 How can existing TTDP algorithms be extended to increase the satisfaction of individuals with the recommended trips?

RQ 2 Which platforms and user interfaces (UIs) support tourists the best in solving the TTDP in realistic scenarios with regard to different usability and UX criteria?

RQ 3 Which group recommendation strategies provide the highest user satisfaction when solving the TTDP for groups?

RQ 4 How do different group types agree on decisions when interacting with a GRS for tourist trips and how fair are their decisions?

RQ 5 Which platform-UI configurations for receiving group recommendations support groups the best when looking for a tourist trip with regard to different UX criteria?

In the following chapters, we present our solutions to each of these problems and explain how we verified them in user studies to answer all RQs. The outcome of this thesis are concrete recommendations for the development of practical tourist trip RSs.

1.3 Methodology

The methodology in this thesis is characterized by prototyping and user studies to solve the TTDP from a user-centered perspective and answer the RQs.

As a first step in our research, we developed a framework for the development and evaluation of platform-independent RSs. We used this framework to develop the tourist trip RS TourRec, a practical application that we used as the basis for answering the RQs in this thesis. The modular and scalable architecture of TourRec allowed us to add new recommendation algorithms, clients, and data sources to the RS and facilitated the evaluation of these components in user studies.

With the introduction of TourRec, we developed a general approach of solving the TTDP from a user-centered perspective. We developed different extensions to this general approach to increase the user satisfaction with the recommended tourist trips. Using previous works as baselines, we evaluated our extensions in different user studies. In addition, we conducted an online evaluation with real users over a period of more than one year. The results of our studies allowed us to understand how different extensions can improve the quality of the recommended trips. Furthermore, we conducted user studies to evaluate different UIs for tourist trip RSs for individuals.

The findings we received from the evaluations of our algorithms and UIs for tourist trip RSs for individuals were the basis for solving the TTDP for groups. For this purpose, we adapted existing group recommendation strategies and developed novel approaches. We compared all strategies in a large user study to identify the most promising strategies for different group types.

We extended TourRec's UIs to enable group recommendations. We conducted a user study to determine the best configurations for receiving group recommendations. For this purpose, we observed how different group types agree on travel preferences and to what extent their decisions respect each group member's preferences in a user study. In addition, we evaluated different GRSs configurations with regard to different UX criteria. A major drawback of previous research in the field of GRSs is that studies were either conducted on a small scale, in a contrived setting, or used synthetic groups which can lead to falsified results [11]. This is why we conducted our studies in a user-centered approach with real groups only.

The presented methodology follows the design-science process, as introduced by Hevner et al. [12]. In this thesis, we addressed each of the seven guidelines for design-science research to the following extent:

Guideline 1: Design as an Artifact We produced viable artifacts in form of a framework and multiple prototypes that we evaluated in user studies to answer our RQs.

Guideline 2: Problem Relevance We reviewed a large number of published literature in the relevant research fields (see Sections 2.3, 2.4.3, 3.1, 4.1, and 5.2). This allowed us to identify research gaps and open challenges and confirm the importance and relevance of our RQs.

Guideline 3: Design Evaluation We conducted user studies to evaluate all of the prototypes that we developed in the course of this thesis to answer our RQs. Table 1.1 summarizes the main goals and metrics of all user studies in this thesis. More detailed explanations are provided in the respective sections.

Table 1.1: Overview of user studies conducted within the scope of this thesis.

Section	Main goal	Metrics
5.3.3	Comparison of a context-aware tourist trip algorithm with a context-unaware solution	User satisfaction
5.4.4	Measurement of the impact of route attractiveness attributes on the user's decision of choosing a walking route between two POIs	Travel decision
5.4.4	Evaluation of the quality of recommended routes that consider route attractiveness attributes	Selected recommendation
5.6	Evaluation of additional extensions for tourist trip algorithms with real users	User preferences, user requests, and recommendation characteristics
6.1.2	Evaluation of the usability of a web-based, context-aware tourist trip RS	Usability
6.2.2	Evaluation of the usability of a mobile tourist trip RS	Usability
6.5	Comparison of different platform-UI configurations for tourist trip RSs	UX
7.3	Comparison of different group recommendation strategies for solving the TTDP for groups	User satisfaction
8.2	Analyzing travel preferences of different group types and understanding group behavior and decision making when interacting with GRSs for tourist trips	Group homogeneity, fairness, applied decision making strategies, and observed group behavior
8.2	Comparison of different platform-UI configurations for tourist trip GRSs	UX

Guideline 4: Research Contributions This thesis contributes to the research fields of OR, RSs and Human-Computer Interaction (HCI). Section 1.5 summarizes all of our contributions.

Guideline 5: Research Rigor Our research relies on rigorous methods: Our ideas are based on the results of related work, prototypically implemented, and evaluated in user studies. We used statistical tests and qualitative analyses where applicable to prove our findings.

Guideline 6: Design as a Search Process The design process of our research is a search process. We iteratively developed solutions to answer our RQs, verified them in user studies, and used the findings as input for the subsequent research phases.

Guideline 7: Communication of Research We published the results of this thesis in peer-reviewed journals, conference and workshop papers, and book chapters. Appendix A lists all of our own publications and summarizes all excerpts from these publications.

1.4 Structure of the Thesis

The rest of this thesis is organized as follows:

Chapter 2 presents the fundamentals of RSs that are required to solve the TTDP for individuals and groups from a user-centered perspective. A broad understanding of recommendation techniques, context-aware recommendations, RSs in tourism, GRSs, UIs for RSs, and the evaluation of RSs is necessary to fully comprehend the RQs in this thesis and the solutions that we developed to answer them. This is why we introduce each of these aspects in detail and summarize important related work where applicable to highlight our contributions in the relevant research areas.

In Chapter 3, we introduce the TTDP. We present different route planning problems that serve as models for the TTDP. Furthermore, we present the most important algorithms and heuristics for each of these problems and provide an overview of open challenges in TTDP research.

Chapter 4 introduces the AnyRec framework for the development and evaluation of platform-independent RSs. We explain how the framework can be used to develop practical tourist trip applications and evaluate different components from a user-centered perspective. Furthermore, we introduce the tourist trip RS TourRec that we developed using AnyRec.

In Chapter 5, we present a general approach to generate tourist trip recommendations for individuals and introduce different extensions to improve it. We show how to enable context-aware recommendations that allow a more realistic prediction of POI values when recommending tourist trips. Furthermore, we integrate route attractiveness attributes that consider the quality of the routes between two POIs to recommend more attractive trips. We present the results of user studies that we conducted to evaluate our extensions and suggest further ideas to improve our approaches. In addition, we summarize the insights of the online evaluation that we conducted using the live version of TourRec. Consequently, RQ 1 can be answered.

In Chapter 6, we present different UIs for solving the TTDP for individuals. These UIs include a web-based application, a mobile application, a public display variant, and a Distributed User Interface (DUI) approach combining a smartphone with a public display. We explain how we evaluated the UIs in user studies and summarize the results of these studies to answer RQ 2.

In Chapter 7, we present group recommendation strategies that we developed to recommend tourist trips to groups. We explain every strategy in detail and show how we evaluated them in a user study. The results of this study allow us to answer RQ 3.

RQ 4 and RQ 5 are answered in Chapter 8. We present the results of a user study in which we analyzed how different group types decide on travel preferences when interacting with GRSs for tourist trips and evaluated the fairness of group decisions. Furthermore, we introduce different configurations for receiving group recommendations and the results of the user study that we conducted to evaluate these configurations with real groups.

In Chapter 9, we summarize our results, discuss them, and present the limitations of this thesis. Furthermore, we suggest future work.

1.5 Contributions

This thesis delivers new insights into the research fields of OR, RSs and HCI. More concretely, it shows how connecting relevant aspects of these fields allows us to solve the TTDP from a user-centered perspective. In the following, we present the main contributions of this thesis.

The Combination of Research in Combinatorial Optimization Problems with Current Topics in RSs Research in RSs is mainly based on machine learning and artificial intelligence techniques. However, recent research tries to develop recommendation algorithms utilizing OR methods [13]. Finding optimal routes for tourists is a typical combinatorial optimization problem. In the relevant literature, the so-called Orienteering Problem (OP) is used as a basic model for finding tourist trips composed of multiple POIs [14]. Previous research in this field developed algorithms providing optimal or near-optimal solutions to the problem but rarely integrated these algorithms into practical applications. Furthermore, existing OP solutions cover only very few of the important aspects of RSs, such as preference elicitation, fast algorithms for practical applications, context-aware recommendations, recommendations for groups, and UIs for RSs. We show how practical RS applications taking into account these aspects can utilize OR methods to provide individuals and groups with feasible and enjoyable recommendations that satisfy their needs.

Approaches for Solving the TTDP from a User-Centered Perspective In published TTDP literature, the value of a POI for a user is often described as a fixed profit, and the value of a trip is simply the sum of the POI profits. Novel algorithms are often evaluated by comparing them to optimal solutions. In this case, the researchers' main goal is to find near-optimal solutions with few gaps and quick execution times. However, when utilizing these algorithms in practical applications, this evaluation type is not suitable. The goal of RSs is to suggest items which are not necessarily an optimal solution from a pure mathematical perspective but instead a pleasure for the user [15]. The perceived quality of a tourist trip recommendation depends on many factors, such as the user's previous experiences, the attractiveness of routes between POIs, and the group type when traveling with others. We developed algorithms that take into account the users' personal preferences and these factors to better adapt the recommendations to the users' actual needs. We evaluated our solutions in user studies to gain a better understanding of how the recommendations are perceived by real users.

Strategies to Solve the TTDP for Groups of Users Until today, TTDP research focuses solely on the creation of tourist trips for individuals. In practice, however, tourists often travel in groups. Recently, first approaches to finding tourist trips that satisfy all members of a group were published [7, 16]. Nevertheless, an investigation which preference aggregation strategies work best to find a consensus is still missing [7]. We applied established group recommendation strategies to the TTDP and compared them to novel approaches, such as a strategy that allows groups to split into smaller groups during a trip. We show how well these strategies satisfy users based on a user study that we conducted with real groups.

UIs for Solving the TTDP for Individuals and Groups Practical applications solving the TTDP have to provide UIs that facilitate preference elicitation, present sequences of POIs in an adequate and attractive manner, and allow users to provide feedback on single items as well as on the whole sequence of POIs. The development of such UIs is especially challenging when solving the TTDP for groups because multiple users interact with the same RS at the same time. The goal is to create UIs that facilitate group decision making which still requires more research. We developed and evaluated different prototypes and UIs to solve the TTDP for individuals and groups. We conducted different user studies to evaluate which platforms and UIs individuals and groups prefer and how they rate them with regard to different usability and UX criteria.

A Comparison of Different Platform-UI Configurations of a GRS GRSs have to perform four tasks: Acquiring information about the members' preferences, generating recommendations, presenting the recommendations to the members, and supporting them in finding a consensus [10]. Different configurations of a GRS that solve these tasks for tourists who are looking for a sequence of POIs are possible: connecting multiple smartphones, sharing a public display, and combining both devices in a DUI approach. To the best of our knowledge, we were the first to investigate the usage of public displays in GRSs for tourist trips and compare different configurations of a GRS in a user study. For this purpose, we conducted a user study with real groups and evaluated these configurations with regard to different UX criteria. Furthermore, we examined how the group type influences a group's choice of the preferred configuration and used these results to provide recommendations for the design of GRSs.

An Understanding of the Behavior of Different Group Types when Interacting with a GRS Many factors can influence the process of decision-making in groups, such as the group type and group homogeneity. Group members that know each other very well communicate differently than people with a rather loose relationship. If group members have similar travel preferences, chances are high that they can reach a consensus. However, if discussions are dominated by one person or a subgroup, decision-making can become unfair. We conducted a user study with real groups interacting with a fully working GRS. Thereby, we observed how different group types agree on group preferences and to what extent these preferences respect each group member's personal preferences. We provide new insights into how the behavior of groups and the fairness of decision-making in groups is influenced by group characteristics and provide recommendations for the development of GRSs in tourism.

A Framework for the Development and Evaluation of Platform-Independent RSs Researchers and students often face the problem of implementing the same core components of a RS, such as a client applications, a backend, and user management, every time they want to test an innovation in the field of RS research, such as novel algorithms. This is why we developed AnyRec, a framework supporting the development of practical RSs and their evaluation in user studies. AnyRec reduces implementation overhead but does not limit developers in the selection of a programming language, for example. It can be used to develop any type of RS. We developed the tourist trip RS TourRec to answer the previously presented RQs. It is the first example of a practical RS application that was developed using AnyRec.

2 Fundamentals of Recommender Systems

In this chapter, we introduce the term *Recommender System* and present background knowledge and related work that form the basis of this thesis. We explain how recommendations can be made for individuals and groups, introduce context-aware recommendations, show how RSs support tourists, and present UIs and evaluation strategies for RSs.

2.1 Recommendation Techniques

RSs are "software tools and techniques that provide suggestions for items that are most likely of interest to a particular user" [17]. Items that can be recommended to users can be any type of product, service, or information. RSs have been successfully applied in various domains, such as e-commerce [18], media [19], and tourism [3, 20].

In other words, the goal of a RS is to identify items that are useful for the user in a given context. The utility of an item is often expressed in user ratings. Ratings can be collected implicitly or explicitly [21]: Implicit ratings are derived from the users' actions, such as how long they view or consume a recommendation. Explicit ratings are provided when the user is asked to rate an item on a pre-defined scale. Different types of rating scales exist for this purpose [21, 22]:

- *Unary*: one rating option (e.g., "Like it")
- *Binary*: two rating options (e.g., "Good / Bad")
- *Multi-staged*: numeric or ordinal scales (e.g., five-star scale, 100-point slider)

In an experiment, Sparling and Sen [22] demonstrated that the cognitive load for all types of non-unary scales is similar. Users were, however, most satisfied with the five-star scale.

Ratings can be visualized in a user-item rating matrix [21]. Table 2.1 shows an example of such a matrix with three users and four items. In this example, ratings are expressed on a scale ranging from 0 (lowest rating) to 5 (highest rating). The empty cell indicates that item 4 has not been rated by user C. A RS is supposed to predict the rating of user C for item 4. If the rating is high, the recommendation should be presented to the user.

Formally spoken, a RS tries to estimate the rating function R

$$R : User \times Item \rightarrow Rating$$

Table 2.1: Example of a user-item rating matrix with three users and four items.

	Item 1	Item 2	Item 3	Item 4
User A	2	0	5	5
User B	4	4	2	2
User C	2	0	4	

for all items that have not yet been rated by the user [8]. This prediction can take into account the user's explicit and implicit past ratings. Different recommendation techniques to make these predictions exist. The most popular techniques are content-based recommendations, Collaborative Filtering (CF), knowledge-based recommendations, and demographic recommendations [17]. Furthermore, multiple techniques can be combined to hybrid approaches to overcome the limitations of single techniques [23]. In this section, we present these techniques, show how they predict user ratings, and explain how they can be combined to hybrid approaches.

2.1.1 Content-Based Recommendations

Content-Based Recommender Systems (CBRSs) try to recommend items that are similar to those the user has liked in the past [24]. In the published RS literature, the similarity between two items is often calculated by using item categories or keywords extracted from the items [25]. For example, if a user liked French restaurants in the past, a content-based RS for restaurants will provide the user with other restaurants of the same cuisine.

The idea of content-based recommendations arose from traditional Information Retrieval (IR) [24, 26]. The biggest advantage of CBRSs over previous approaches is the use of user profiles that keep record of the user's tastes and preferences. In general, CBRSs try to predict the utility of an item i for a user u [26]:

$$pred(u,i) = score(ContentBasedProfile(u), Content(i)), \tag{2.1}$$

where $ContentBasedProfile(u)$ is u's user profile and $Content(i)$ is the item profile, i.e., the features characterizing i. Keyword analysis techniques from IR can be used to create user profiles.

An example of a model used in many CBRSs is the vector space model, a model for representing textual documents in a vector space [27]. In RSs that use the vector space model, profiles and items are represented as weighted term vectors. The cosine similarity can be used to determine the similarity between two documents using the weighted term vectors [27]:

$$sim(d_i, d_j) = \frac{\sum_k w_{ki} \times w_{kj}}{\sqrt{\sum_k {w_{ki}}^2} \times \sqrt{\sum_k {w_{kj}}^2}}, \tag{2.2}$$

where w_{kj} denotes the weight for term t_k in document d_j. The keyword weights can be specified using the *Term Frequency-Inverse Document Frequency* weighting function [26].

Other possible techniques for content-based recommendations are Bayesian classifiers and various machine learning techniques, such as clustering, decision trees, and artificial neural networks [26].

A special form of content-based recommendations are case-based recommendations. In case-based RSs, "items or products are represented as cases and recommendations are generated by retrieving those cases that are most similar to a user's query or profile" [28]. Compared to CBRSs, case-based RSs rely on very structured representations of items. This allows the implementation of more sophisticated similarity metrics [28]:

$$Similarity(t, c) = \frac{\sum_{i=1..n} w_i \times sim_i(t_i, c_i)}{\sum_{i=1..n} w_i}, \tag{2.3}$$

where t_i is a feature i of the target query t, c_i the corresponding feature of the candidate case or item c, and w_i the relative importance of feature i. The similarity between two features t_i and c_i is calculated as follows [28]:

$$sim_i(t_i, c_i) = 1 - \frac{|t_i - c_i|}{max(t_i, c_i)}. \tag{2.4}$$

Equation (2.4) is an example of a *symmetric* similarity metric. There is no bias in favor of either higher or lower feature values than specified by the user. For some features, an *asymmetric* similarity metric is more appropriate. In this case, the metric prefers feature values that are lower than the user's specification over feature values that are higher than the user's specification, or vice versa [28]. For example, a user who is willing to pay €300 for a flight will rather accept an offer that costs €200 instead of a recommendation that costs €400.

CBRSs offer several advantages compared to other recommendation techniques: They are built solely on the user's own ratings; hence, recommendations can be made even when only one user is using the system. Furthermore, they can recommend new items which have not yet been rated since recommendations are based on item features, but not ratings. However, CBRSs have some limitations [26, 27]:

- *New User Problem*: Content-based methods require a user profile before a recommendation can be made. A user with no or few ratings cannot receive accurate recommendations.
- *Limited Content Analysis*: CBRSs depend on features that describe the items. A sufficient number of features describing an item is required for accurate recommendations. In addition, different items with the same set of features are indistinguishable.
- *Overspecialization*: Recommendations are similar to items that the user liked in the past. This reduces diversity and serendipity.

2.1.2 Collaborative Filtering

While content-based methods consider only the user's own ratings, the general idea of CF is to recommend items that other users with similar tastes and preferences liked in the past [26]. Different algorithms exist to implement CF. One of the most popular algorithms are *nearest neighbor* algorithms. They can be divided into *user-based nearest neighbor* and *item-based nearest neighbor* algorithms [21].

User-based nearest neighbor algorithms predict the rating of a user u for an item i by analyzing ratings for i from users with similar preferences. Users that are similar to u are called *neighbors* [21].

Different formulas for predicting the rating of a user u for an item i based on the ratings r of all neighbors exist. Equation (2.5) gives more weight to neighbors with higher similarity and takes into account that some users give consistently higher or lower ratings to items than other users [21]:

$$pred(u,i) = \bar{r}_u + \frac{\sum_{n \subset neighbors(u)} userSim(u,n) \times (r_{ni} - \bar{r}_n)}{\sum_{n \subset neighbors(u)} userSim(u,n)}. \tag{2.5}$$

$userSim(u,n)$ measures the similarity between two users u and n. The Pearson Correlation Coefficient (PCC) is one way to calculate the similarity of two users. It measures the linear correlation between two variables and is often used in RSs to identify similar users [19]. PCC ranges from -1.0 (perfect negative linear relationship) to 1.0 (perfect positive linear relationship). Values of 0.10, 0.30, and 0.50 indicate small, medium, and large effect sizes, respectively [29]. Schafer et al. [21] recommend to not use negative correlations to increase prediction accuracy. Equation (2.6) shows the formula for calculating the PCC between user u and a neighbor n, where $CR_{u,n}$ denotes the set of all items rated by u and n [21]:

$$userSim(u,n) = \frac{\sum_{i \subset CR_{u,n}} (r_{uj} - \bar{r}_u) \times (r_{ni} - \bar{r}_n)}{\sqrt{\sum_{i \subset CR_{u,n}} (r_{ui} - \bar{r}_u)^2 \times \sum_{i \subset CR_{u,n}} (r_{ni} - \bar{r}_n)^2}}. \tag{2.6}$$

The ratings of user C in Table 2.1 are more similar to user A's ratings than the ones of user B.[1] Hence, when predicting the rating of user C for item 4, the rating given by user A has a higher impact on the prediction than the rating given by user B.

While *user-based nearest neighbor* predict ratings based on the similarity of users, *item-based nearest neighbor* algorithms consider the similarity of items. Two items that were similarly rated by different users are called *similar items*. Item 3 and 4 in Table 2.1 received equal ratings from all users who rated both items. Hence, these two items are considered to be very similar. Experiments have shown that *item-based nearest neighbor* algorithms provide a higher prediction quality than *user-based nearest neighbor* algorithms [30].

Equation (2.7) shows how to predict the rating of a user u for an item i by considering the similarity of items [21]:

[1] Please keep in mind that Table 2.1 shows only a simplified example of a user-item rating matrix. For more accurate and trustworthy predictions, a higher number of ratings is required.

$$pred(u,i) = \frac{\sum_{j \subset ratedItems(u)} itemSim(i,j) \times r_{uj}}{\sum_{j \subset ratedItems(u)} itemSim(i,j)}. \tag{2.7}$$

$itemSim(i,j)$ measures the similarity between two items i and j. Schafer et al. [21] presented an approach that uses adjusted-cosine similarity to calculate the similarity between two items. Equation (2.8) shows the formula for adjusted-cosine similarity which compares the ratings of all users $RB_{i,j}$ who rated both item i and item j:

$$itemSim(i,j) = \frac{\sum_{u \subset RB_{i,j}} (r_{ui} - \bar{r}_u) \times (r_{uj} - \bar{r}_u)}{\sqrt{\sum_{u \subset RB_{i,j}} (r_{ui} - \bar{r}_u)^2 \times \sum_{u \subset CRB_{i,j}} (r_{uj} - \bar{r}_u)^2}}. \tag{2.8}$$

An advantage of CF is the diversity of the recommendations. CF can recommend any type of item, even items that are not similar to any item the user has consumed in the past. However, CF requires user ratings and therefore comes with some limitations [26]:

- *New User Problem*: As in content-based methods, CF needs to know the user's preferences before a recommendation can be made.
- *New Item Problem*: Items need a substantial amount of user ratings before they can be recommended.
- *Sparsity*: The proportion of ratings in a user-item rating matrix is usually much lower than in Table 2.1. A sparse user-item rating matrix makes it more difficult to identify similar users. Furthermore, items with very few ratings will rarely be recommended.

2.1.3 Knowledge-Based Recommendations

Knowledge-based RSs use domain knowledge to predict if an item can satisfy the user's needs [17]. Two types of knowledge-based RSs can be distinguished: case-based and constraint-based approaches [31]. Both types make recommendations based on knowledge about the items and how well they match the user requirements. The major difference is that case-based approaches use similarity metrics to make recommendations (see Section 2.1.1) whereas constraint-based RSs "predominantly exploit predefined recommender knowledge bases that contain explicit rules about how to relate customer requirements with item properties" [31].

An important interaction style in both types of knowledge-based RSs is critiquing. The idea is that the user can improve a search result by critiquing features of the recommended items. One of the first IR systems that applied critiquing was RentMe, a web interface for a database of classified ads for rental apartments [32]. RentMe allows users to specify a search query and refine the results by using critiques, such as "The apartment could be bigger" or "This neighborhood could be more dynamic". Then, the results are updated. The original search query is kept in mind but some constraints can be relaxed if too few results can be found after the critiquing. Entree and Recommender.com are other examples of knowledge-based RSs that implement critiquing [33].

Figure 2.1: Critiquing in ENTREE [23].

They allow to specify a query and improve the recommendations by modifying the query (e.g., “add feature”) and critiquing the recommendations (e.g., “less $$”) (Figure 2.1).

Knowledge-based RSs do not require any user ratings because their decisions are independent of individual tastes [33]. Furthermore, they are strongly complementary to other types of RSs [23]. However, knowledge of domain experts has to be gathered before a knowledge-based RS can recommend items. This knowledge has to be transformed into a formal and executable representation. This issue is known as *knowledge acquisition bottleneck* [31].

2.1.4 Demographic Recommendations

Demographic RSs generate recommendations based on the user’s demographic profile. For instance, a RS for movies can take into account the user’s age and languages. Similar to collaborative methods, demographic RSs have the ability to “entice users to jump outside of the familiar” [23]. However, compared to the previously presented recommendation techniques, little research has been done on demographic recommendations [17].

2.1.5 Hybrid Techniques

Hybrid RSs combine multiple recommendation strategies to overcome the limitations of single strategies. For example, a CF component can be combined with a CBRS to overcome the *New Item Problem* without decreasing the diversity of recommendations.

Burke [23] identified seven types of hybrid recommendation strategies:

- *Weighted*: Each component calculates a profit for an item. All profits are combined numerically.
- *Switching*: The RS selects one component which is used to generate a recommendation. The decision is based on a selection criterion.
- *Mixed*: Each component generates recommendations. All recommendations are presented in a combined list.
- *Feature Combination*: Elements of one component are integrated into another component. For instance, a CBRS can be extended by a collaborative feature.
- *Feature Augmentation*: One component computes features which are then part of the next component. *Content-boosted CF* is an example where content-based recommendations are used to fill a sparse rating matrix before CF is applied [34].
- *Cascade*: Components are ordered hierarchically. A component with a lower priority cannot change decisions made by components with higher priority, but solve ties.
- *Meta-level*: The RS uses a model learned by another component as input.

2.2 Context-Aware Recommendations

The utility of an item for a user can change under different conditions. For instance, a user who enjoys spending time in parks during summer will most likely not appreciate recommendations for outdoor activities on a rainy day. Hence, a park's utility for this user does not only depend on the user's general attitude towards parks, but also on the weather. Weather is just one example of a contextual factor that has an impact on predicted ratings. Other examples of relevant contextual factors when recommending POIs, such as parks, are the time of the day, the POI's crowdedness, and the user's budget [35].

In general, context describes a broad concept which has been researched in many fields besides computer science, such as linguistics, philosophy, and psychology [8]. Dey defines context as "any information that can be used to characterize the situation of an entity" [36]. An entity in this regard is "a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves" [36].

Wörndl et al. [37] differentiate between four types of context:

- *User context* (e.g., the user's budget)
- *Temporal context* (e.g., the time of the day)
- *Geographic context* (e.g., the user's current location)
- *Social context* (e.g., whether the user is traveling alone)

Baltrunas et al. [35] designed a methodology to assess the relevance of contextual factors in different situations and the influence of contextual conditions on ratings. They applied their methodology to analyze the relevance of tourism-related contextual factors (e.g., weather) on POI categories gathered from a list of POIs in Bolzano, Italy, and how different contextual conditions change the user's rating of a POI. Their results show that, for example, temperature is a relevant factor when deciding upon visiting a nature wonder (0.62 on a scale ranging from 0 to 1), but less important when visiting a castle (0.13). A cold temperature significantly reduces the average rating of a castle POI while other conditions do not have any impact. This methodology has been applied to other domains, such as mobile shopping [38].

After the relevance of contextual factors and their impact on the utility of items has been assessed, the contextual factors have to be integrated into the recommendation process to improve the recommendations. Van Setten et al. [39] explain that contextual factors can either be used as soft or hard criteria in the recommendation process. Items that do not match a hard criterion are no candidates for a recommendation (e.g., a POI that costs more than the user's budget). Items that do not match a soft criterion receive a lower utility, depending on how strongly the criterion is violated. They can, however, still be recommended, especially when no better alternatives are available.

Adomavicius and Tuzhilin [8] presented three different paradigms that explain how to incorporate context at different stages of the recommendation process:

- *Contextual Pre-Filtering*: Context is used to filter the relevant ratings from the user-item rating matrix before a recommendation is made.
- *Contextual Post-Filtering*: At first, recommendations are made without considering context. Then, the recommendations can be adjusted based on the given context. This can be done by either filtering out recommendations that are not suitable (hard citerion) or changing the ranking of the recommendations (soft criterion).
- *Contextual Modeling*: Context is used directly in the recommendation process.

The diagrams in Figure 2.2 illustrate the three different paradigms to incorporate context. While *Contextual Pre-Filtering* and *Contextual Post-Filtering* use the two-dimensional user-item rating matrix as input, *Contextual Modeling* extends the matrix by the third dimension *context*. Formally spoken, *Contextual Modeling* extends the rating function R introduced in Section 2.1 by a third entity *contextual information* [8]:

$$R : User \times Item \times Context \rightarrow Rating$$

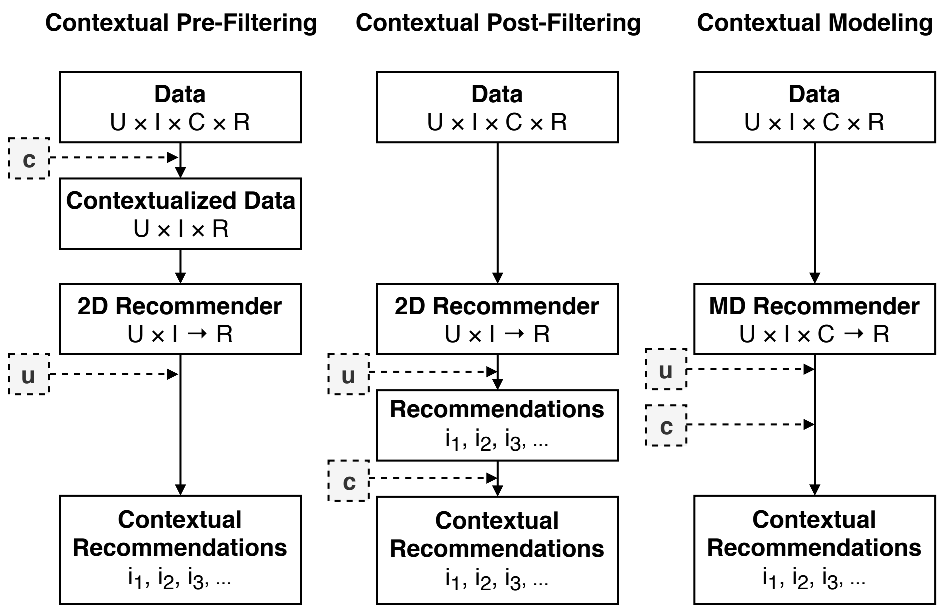


Figure 2.2: Paradigms for incorporating context in RSs (adapted from [8]). *u* illustrates at which stages the recommendation function is used on user *u*, *c* at which stages information about context is applied.

2.3 Recommender Systems in Tourism

The previously presented techniques can be used to recommend items of various types. During the last years, an increasing number of RSs has been used in the field of e-tourism to support tourists at different stages of their trip planning or when already traveling. These e-tourism RSs are mainly used to recommend one of the following travel-related items [3, 20]:

- Ranked lists or sets of POIs.
- Travel plans[2] combining coherent travel items, such as destinations, activities, and other services, in one recommendation. For example, a travel plan for a skiing trip could contain a destination, a nearby ski resort, and a hotel [40].
- Sequences of POIs along an enjoyable route for a single or multi-day trip (also called tourist trips).

In the following, we present important examples from published literature for each of these three recommendation goals. The goal of this section is to provide a comprehensive list of research projects and prototypes in the field of tourist trip RSs. We use this list to classify the goals of this thesis in relation to research in e-tourism and RSs and present related research areas which are not covered by the results of this thesis. Further

[2]also called travel bags or travel bundles in published literature

extensive surveys of tourism RSs have been published by Borràs et al. [20] and Gavalas et al. [3].

Parts of this overview have been published in [41, 42].

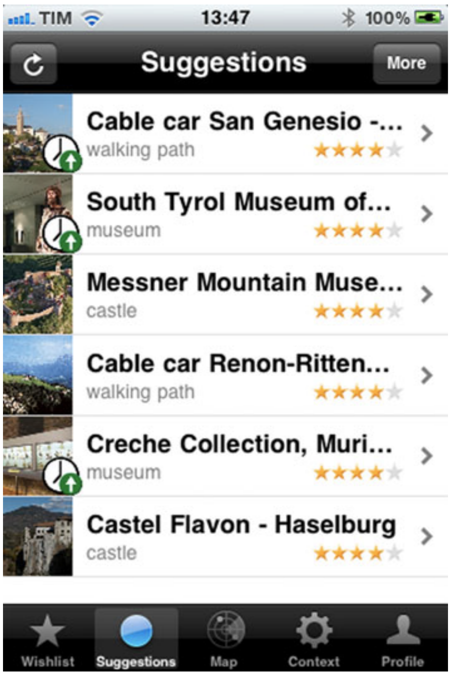
2.3.1 Recommendation of Lists or Sets of Points of Interest

The majority of RSs in tourism suggest lists or sets of POIs to users without combining them to coherent travel plans or tourist trips. In this section, we list some of the most important RSs for this recommendation goal and briefly present their main characteristics and key features. Even though we focus on recommending tourist trips in this thesis, these examples served as inspiration for our own solutions as a tourist trip RS has to identify attractive POIs that can be merged to a composite trip. Thus, the following list summarizes ideas and technical solutions for realizing such POI recommendations.

GUIDE is a MTG developed for early mobile devices [43]. It takes into account different personal and environmental contextual factors, such as the user's age and the time of the day, to recommend appropriate POIs. A user study with 60 participants showed that the system is appreciated by the vast majority of users. Another mobile CARS is COMPASS [39]. In a user survey with 57 participants, the users confirmed that context-aware tourism recommendations are perceived as useful. However, some participants emphasized that they want to be the ones who decide which factors are relevant for a recommendation; hence, a CARS should not take away the full responsibility from the user. Baltrunas et al. [35] developed the CARS ReRex, an iPhone application which utilizes relevant contextual factors for POI recommendations in the city of Bolzano, Italy (Figure 2.3). ReRex allows users to switch on/off contextual factors and to specify them, if necessary (e.g., the type of companion). A usability test with 20 participants confirmed that context-aware recommendations are more effective than context-unaware recommendations. South Tyrol Suggests is a novel CARS for POIs in South Tyrol, Italy [44]. It considers various contextual factors, such as the weather at a POI. Furthermore, a personality questionnaire is used to overcome the cold-start problem. In a user study with 54 participants, the authors showed that including the weather factor increases the user's satisfaction with the selected recommendation. The application is available for download on Google Play[3]. MobyRek recommends travel products when the user is already traveling and when the current situation is appropriate for a recommendation, such as a nearby restaurant [45]. It uses a conversational approach to improve the recommendations iteratively and reduce the user's effort. Xie et al. [46] proposed a novel system called CompRec-Trip for recommending sets or sequences of POIs. In addition, they developed a graphical UI that allows users to customize the recommendations. Benouaret and Lenne [47] presented a novel RS for travel packages whereby each package is composed of a set of different POIs. The authors used a real-world dataset to demonstrate the quality of the recommendations and are planning to do a further study with a mobile application that they develop. Baral and Li [48] presented another promising approach to find POIs but it has not yet been

[3] `https://play.google.com/store/apps/details?id=it.unibz.sts.android` (accessed February 16, 2020)

implemented in a practical application. Their approach combines different aspects of check-in information in Location-based Social Networks (LBSNs), such as categorical, temporal, social, and spatial information, in one model to predict the most potential check-in locations.



(a)

(b)

Figure 2.3: Context-aware POI recommendations in ReRex [35]. The application (a) provides a list of recommendations and (b) marks recommendations that are particularly suited for the specified context.

2.3.2 Recommendation of Travel Plans

While the previously presented examples of RSs suggest independent items which are not directly connected to each other, travel plan RSs combine multiple travel items, such as destinations, activities, and other services, to one coherent recommendation. Hence, they can be understood as an extension of RSs for POIs as they also answer travel-related questions that go beyond the choice of POIs, such as accommodation. In the following, we list and briefly describe important travel plan RSs that were presented in published literature.

Lenz [49] was the first to develop a case-based RS for holiday trips. The case description of CABATA contains features such as the type of holiday, the travel region, and means of transport. The case solutions present recommendations that fulfill all user requirements and others that are at least similar to the user query. CABATA is a prototypically implemented part of an architecture for travel agent systems called

IMTAS and was presented to the public in 1994 [50]. Ricci and Werthner [51] extended the idea of CABATA in different directions. For example, the recommendations do not only consider the user query but also the personal case base of the user or other, similar users. Other examples of case-based travel RSs are DieToRecs and Trip@dvice. DieToRecs allows the recommendation of single items, such as destinations or hotels, and bundling of travel items for a personalized travel plan [52]. The case base of Trip@dvice contains travel plans created by the community [53]. It has been selected by the European Union and by the European Travel Commission as a travel RS in the European tourism destination portal visiteurope.com. Ricci [54] presented TripMatcher and VacationCoach, two of the early travel RSs using content-based approaches to match the user preferences with potential destinations. VacationCoach explicitly asks the user to choose a suitable traveler type, such as *culture creature* or *beach bum*. TripMatcher uses statistics on past user queries and guesses the importance of attributes not explicitly mentioned by the user to come up with recommendations. A conversational RS for travel planning was introduced by Mahmood et al. [55].

2.3.3 Recommendation of Sequences of Points of Interest along Routes

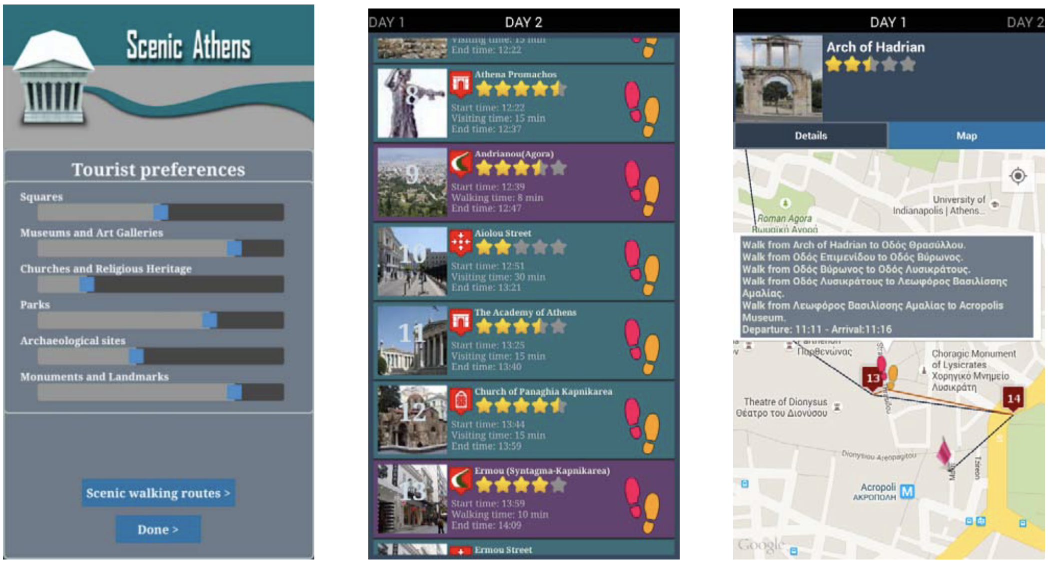
The previously presented examples of RSs in tourism recommend POIs and some of them enrich the recommendations with other travel-related items or information to propose so-called travel plans. However, they are not able to plan complete sightseeing trips as they do not recommend a chronological sequence of POIs and do not come with recommended durations of stay or suggest walking routes between the recommended POIs. Tourists exploring a city on single- or multi-day trip are often looking for such sequences of POIs along feasible and enjoyable routes to facilitate travel planning. These sequences are known as tourist trips [6]. Tourist trip recommendations have to respect several user requirements and constraints, such as the time available for the trip, opening hours of POIs, and desirable breaks [3].

Tourist trips RSs are often developed with the goal of providing solutions to different combinatorial optimization problems. In Section 3, we introduce these problems and summarize algorithms and heuristics solving them to generate routes. In the following, we present a list of practical tourist trip applications and alternative approaches for recommending sequences of POIs. We explain their key features and describe briefly how they come up with personalized trip recommendations. The purpose of this section is to provide a broad understanding of existing approaches and technical solutions for recommending tourist trips, which is the focus of this thesis. Furthermore, it allows us to highlight our own contributions in the field of tourist trip recommendations by explaining the key differences from our approaches to the presented examples at the end of this section.

The aforementioned CARS GUIDE was also one of the first applications that combines recommended POIs to tourist trips [43]. For this purpose, users have to choose POIs they would like to visit. Then, the system creates a route taking into account relevant factors, such as the opening hours of the selected POIs. GUIDE is also able to update the recommended routes dynamically when the user decides to stay longer at a

POI as planned. Hence, it takes into account temporal contextual factors due to a reduced remaining time for the trip. e-Tourism [56] is a web application that recommends personalized tourist trips in the city of Valencia, Spain. It first recommends a list of POIs by considering different criteria, such as the user's travel preferences, demographics, and former trips. e-Tourism uses a taxonomy to classify the user's profile information and uses a hybrid of demographic, content-based, and preference-based filtering to generate recommendations. Then, it creates a route by scheduling the recommended POIs. The City Trip Planner is a web application that recommends multi-day tourist trips [57]. It respects certain limitations, such as opening hours, and can include a lunch break into the trip. A mobile web application for multi-day trips is dailyTRIP [58]. Besides user preferences, it takes into account opening hours, time available for visiting attrations, and average visiting times. Rodríguez et al. [59] developed SAT, a tourist support system which includes a multi-criteria model considering tourist wishes and needs, desired activities, and characteristics of the target area. They implemented a practical application of the system and demonstrated it by recommending personalized trips in the Autonomous Region of Andalusia. Tanahashi and Ma [60] performed two user studies to test their mashup system OnMyWay that allows designing road trips. Their work focuses on the design of UIs to facilitate the exploration of data relevant for the itinerary planning, but does not present algorithms for an automatic route generation. myVisistPlanner^{GR} is another web-based application for trip planning [61]. It supports travelers in planning trips in the region of Northern Greece. It considers the user's demographics and interests, and user-selected criteria, such as visit duration and geographical areas of interest. Gavalas et al. [62] developed eCOMPASS, a context-aware web and mobile tourist trip planner. They developed the SlackRoutes algorithm which integrates multi-modal route planning into the recommendations and suggests lunch breaks. A pilot study in Berlin showed that the recommended trips are attractive, feasible, and relevant to the user's preferences. Scenic Athens is another context-aware MTG for personalized tour trip recommendations developed by Gavalas et al. [63] (Figure 2.4). Compared to similar applications, Scenic Athens can also incorporate scenic routes into the recommended trips. The authors developed an Android application and conducted several performance tests as well as a small user study with locals and tourists to evaluate their approach. c-Space is a novel tourist trip RS which takes into account the travel and time-use implications of visiting POIs already when selecting a set of candidate POIs [64]. The authors conducted a Social Choice experiment to create a user model for their RS and a latent-class analysis to segment the participants with regard to their travel preferences. Results identified three traveler segments. The authors evaluated their user model with recommendations in Trento, Italy. The feedback from 35 users confirmed the usefulness of their approach.

Photos and LBSNs have become a popular data source to generate tourist trip recommendations. De Choudhury et al. [65] used photo streams to estimate where users were and how much time they spent at a POI and for traveling between POIs. Based on this information, their approach creates a POI graph and recommends tourist trips. In a user study, the authors showed that their recommendations are as good as ground truth trips provided by bus tour companies in terms of overall usefulness and POI sat-



(a) (b) (c)

Figure 2.4: Tourist trip recommendation in Scenic Athens [63]. The app allows (a) users to rate POI category preferences (a similar UI is available for scenic route categories) and display recommendations on (b) a list view that illustrates POIs (green background) and scenic routes (purple background) and (c) on a map with walking instructions.

isfaction. Lu et al. [66] developed Photo2Trip, a trip planning framework that uses data extracted from 20 million geo-tagged photos to recommend trips. The data gathered from the photos allowed them to identify not only popular destinations, but also the order of locations, typical travel paths between destinations, and the recommended duration of stay for each location. Another solution using photos was presented by Brilhante et al. [67]. Their application TripBuilder uses unsupervised learning for mining common patterns of movements of tourists in a given geographic area. Therefore, TripBuilder mines public photos from Flickr and collects POI data from Wikipedia to create a POI database including patterns of movement of tourists that visited a POI in the past which is used to generate trips. The authors created datasets of three cities to evaluate their approach. Results showed that their method outperforms two baselines. The PersTour algorithm, recently presented by Lim et al. [68], personalizes POI visit durations on recommended trips. The authors used real-life travel sequences and POI popularity extracted from geo-tagged photos to train their model. In an evaluation, they compared their approach against various baselines using a Flickr dataset across ten cities. Their approach outperformed the baselines with regard to different metrics, and it was able to recommend trips that better reflect real-life travel sequences of tourists. Quercia et al. [69] introduced a different approach for route recommendation. Instead of recommending shortest paths between two directions or maximizing POI profits, their trip recommender suggests routes that are perceived as pleasant. The authors collected crowd-sourced ratings to identify pleasant routes and showed that this type of information can be computed from Flickr metadata. Their user studies confirmed that their approach can recommend beautiful, quiet, and happy routes adding only a few extra walking minutes to the trip. Yu et al. [70] developed a tourist trip RS which uses check-in data from the Chinese LBSN Jie Pang to generate travel packages. A route planning algorithm is used to find routes containing appropriate POIs from the recommended travel packages. The authors implemented a first prototype of their system which is composed of a mobile client and a recommendation server. Other approaches extract POI sequences from travel blogs [71]. The content of travel blogs often provides much crowdsourced geospatial data, such as the locations of POIs and the spatial relation between POIs. These data can be transformed to POI graphs that show popular POIs and recommended sequences of POIs.

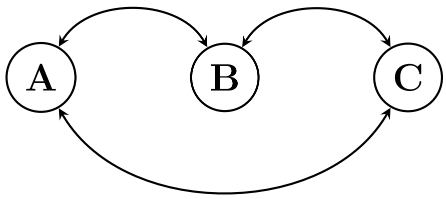
These examples of tourist trip RSs pursue a similar goal as the prototypes that we developed in this thesis: they recommend sequences of POIs along routes for single- or multi-day trips. This section provided an overview of different approaches for achieving this goal. It also revealed research gaps that we want to close with our work. Only few practical applications use the advanced OR methods that we present in Chapter 3 to recommend tourist trips. In this thesis, we show how to extend these methods to solve the TTDP from a user-centered perspective. For this purpose, we integrated aspects from RS research which have not yet been considered in such methods, such as context-aware recommendations, and evaluated our proposed solutions in user studies. Section 3.2 explains these open challenges in TTDP research and our motivation to tackle them in more detail. Furthermore, in contrast to the aforementioned examples of tourist trip RSs, we also solved the TTDP for groups of users (see Chapters 7 and 8).

2.4 Group Recommender Systems

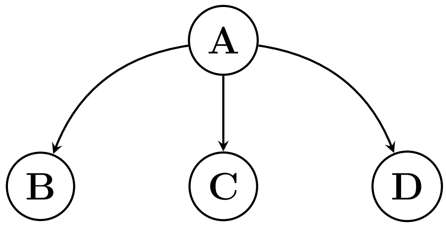
The aforementioned recommendation techniques are tailored to recommendations for individuals. In addition, the majority of the presented RSs generates recommendations only for one person at a time. In some situations, however, groups of users are looking for a mutual recommendation, such as a restaurant for dinner or a movie. GRSs support groups in making a decision and finding a consensus that satisfies all group members. In this section, we define the term *Group*, explain how GRSs make recommendations, and present examples of GRSs.

2.4.1 Social Groups

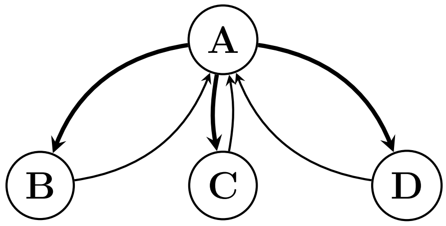
A wide range of definitions for the term *Group* exist [72–74]. For example, Bonner defines a group as “a number of people in interaction with one another, and it is this interaction process that distinguishes the group from an aggregate” [75]. According to this definition, interaction among group members is a key characteristic that makes an aggregate of individuals a group. Most existing definitions emphasize the importance of interaction among group members when defining groups [73]; however, a large number of



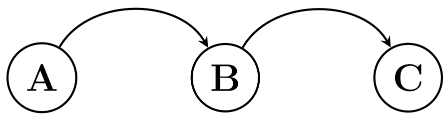
(a) Symmetric interdependence with reciprocity.



(b) Hierarchical interdependence without reciprocity.



(c) Hierarchical interdependence with (unequal) reciprocity.



(d) Sequential interdependence without reciprocity.

Figure 2.5: Examples of group interdependencies (adapted from [72]). Letter represent persons, arrows indicate the direction of influence.

definitions exist that use one or more of the following characteristics to describe groups: goals, interdependence, structure, and cohesiveness [72].

Group members can perform a wide range of interactions which can either focus on the relationships among the group members (e.g., supporting each other) or on the tasks and goals of the group [72]. Goals that groups pursue can be very different. McGrath [76] describes four main types of group goals: generating, choosing, negotiating, and executing. Groups can generate new ideas and plans, choose between alternatives, solve conflicts, and perform action tasks. Tasks and goals of a group are highly related. Steiner [77] defines a task as “a set of specifications identifying the goal that is to be achieved and the procedures that an individual or group may employ when attempting to achieve it”. In other words, tasks are what a group must do to achieve its goals [73]. Steiner, furthermore, distinguishes between divisible and unitary tasks. Divisible tasks can be divided into subtasks and therefore make a division of labor feasible.

Interdependence describes dependencies between group members [72]. Figure 2.5 illustrates four examples of interdependencies in groups. In Figure 2.5a, the influence among all members is equal and reciprocal. Figure 2.5b, however, illustrates the interdependency of a group which is hierarchically but not reciprocal. Person A influences all group members, but not vice versa. There are also examples where hierarchical groups are reciprocal. In the example of Figure 2.5c, however, A still influences the other group members to a greater extent than they can influence A. The fourth example (Figure 2.5d) shows a sequential interdependence. A has no direct influence on C, but B’s actions (which influence C) are influenced by A.

The group structure are features, such as roles, norms, and relationships in a group, that organize a group [72]. The group cohesion, furthermore, describes the integrity, solidarity, and social integration of groups and the "sticking-togetherness" of the group members [72, 77]. Group members of a cohesive group are more attracted to the group than to other options with regard to factors, such as expected payoffs and costs of membership [77]. Consequently, "members of cohesive groups are generally better satisfied with the group than members of noncohesive group" [73] .

According to Forsyth [72], four group types can be distinguished: *Primary Groups*, *Social Groups* (also known as *Secondary Groups*), *Collectives*, and *Categories*. A primary group is a "small, long-term group characterized by frequent interaction, solidarity, and high levels of interdependence among members that substantially influences the attitudes, values, and social outcomes of its members" [72]. Members of primary groups share close relationships with the other group members and feel very committed to their group; the group is an important part in their life. Examples of primary groups are families and close friends. Secondary groups, however, are usually larger than primary groups and characterized by rather loose relationships. Thus, it is easy for group members to leave a secondary group and join another one. Secondary groups are often created in goal-focused situations. Coworkers and fellow students are examples of secondary groups. Collectives are usually larger groups of people that are often created spontaneously. They do not exist for a long time and their members do not feel very committed to the group. People waiting in a line or watching the same concert can be called collectives. Categories are groups whose members are similar to one another, often in terms of demographic factors, such as gender, age, or nationality. Even though members in a category do not know each other, they still can feel very connected to their category. Examples of categories are U.S Americans and supporters of the same football team.

Groups can also be characterized in terms of homogeneity. The degree of homogeneity in groups can refer to different variables, such as the group members' needs, their personality attributes, and their perception of the group goals [73]. Steiner [77] explains that in competitive situations, group members hold different goals for the group, whereas in cooperative situations, the goals are homogeneous. However, with regard to task-relevant abilities, heterogenous group promise a high productivity when the task is disjunctive, but a low productivity for conjunctive tasks.

Experiments using groups as subjects can choose between *experienced groups* (also called *natural groups*) and *naïve groups* (also called *artificial groups*) [73]. Experienced groups have already established relationships which can influence study results. In naïve groups, the members are either randomly assigned to the group or selected by the investigator. Using naïve groups allows studying groups that do not exist in the real world, e.g., groups composed of only highly dominant individuals. However, using naïve groups makes it more complicated to transfer results to the real world.

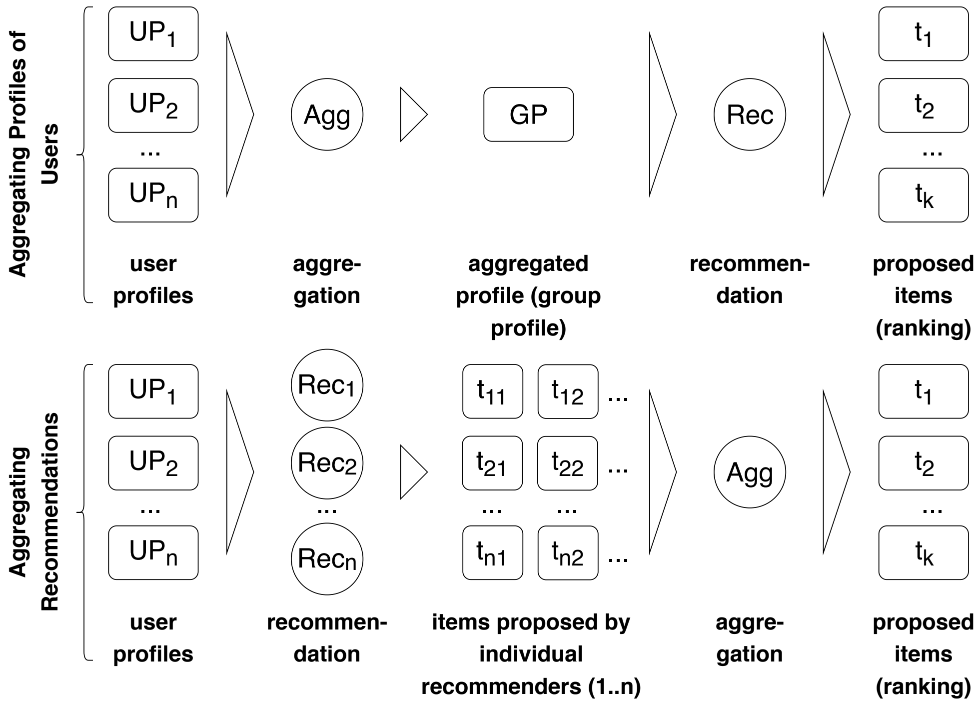


Figure 2.6: Group recommendation strategies (adapted from [25]).

2.4.2 Group Recommendation Techniques

Two main principles for generating a group recommendation exist (Figure 2.6): On the one hand, the user profiles of all group members can be aggregated to create a group profile [78]. This group profile is then used to request a recommendation (*Aggregating Profiles of Users (AP)*). On the other hand, a recommendation can be made for every user individually before the recommendations are combined into one group recommendation (*Aggregating Recommendations (AR)*) [10, 79].

2.4.2.1 Aggregating Profiles of Users

Many strategies that can be used to aggregate profiles of group members to a group profile were inspired by Social Choice Theory [78]. In the following, we present examples for some of the preference aggregation strategies that were discussed in published literature [78, 80]. Table 2.2 shows the ratings of a group of three users. Each user rated the six same items. We use these ratings to illustrate the following group recommendation strategies.

AP strategies can be categorized into majority-based, consensus-based, and borderline strategies [81]. Table 2.3 exemplifies four consensus-based strategies. These strategies consider the preferences of all group members. The AVERAGE strategy calculates the average of the individual ratings of all group members for every item. The MULTIPLICATIVE strategy is a similar strategy. It uses the product of the group members' ratings

Table 2.2: Ratings of a group of size 3.

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
User A	3	1	2	1	4	5
User B	3	4	5	4	3	1
User C	3	1	5	4	2	3

to decide upon a recommendation. One disadvantage of both strategies is that they allow manipulation. For example, people can give very low ratings to all items that they would like to avoid to increase the chances that one of their favorite items is selected by the RS. The issue of manipulation is less relevant when ratings are inferred from user behavior and when the group members are unaware of the ratings of others [11]. The MEDIAN strategy allows overcoming this problem even when using explicit ratings [80]. In the example of Table 2.3, the median ratings of items 3 and 4 are high even though user A's ratings for both items are low. A problem of the AVERAGE, MULTIPLICATIVE, and MEDIAN strategies is that they can assign high ratings to items which can be unsatisfying for some group members. If we do not assume manipulation, user A will most likely be unhappy with the recommendation of item 4. The AVERAGE WITHOUT MISERY strategy overcomes this problem by removing all items that received at least one rating below a pre-defined threshold. In the example of Table 2.3, the threshold t is set to 2. All items with ratings below this threshold are removed. Therefore, only items 1, 3, and 5 remain candidates for a recommendation.

Table 2.3: Group ratings generated by consensus-based strategies based on the user ratings in Table 2.2.

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
Average	3	2	4	3	3	3
Median	3	1	5	4	3	3
Without Misery ($t = 2$)	3	-	4	-	3	-
Multiplicative	27	4	50	16	24	15

Borderline strategies take into account only a subset of the user preferences. Table 2.4 shows three examples of borderline strategies. The assumption behind the LEAST MISERY strategy is that a group is as satisfied as its least satisfied member. Therefore, it uses the minimum rating among all group members as group rating. This is why item 1 in Table 2.4 receives the highest group rating. The MOST PLEASURE strategy is the opposite of the LEAST MISERY strategy. It assumes that a group is as satisfied as the most satisfied group member. Therefore, items 3 and 6 receive the highest group rating, while item 1 receives the lowest rating of all items. In groups which apply the DICTATORSHIP strategy, one group members decides upon the group preferences. In the example of Table 2.4, user A's ratings are the group ratings. Therefore, item 6 receives the highest rating.

Table 2.4: Group ratings generated by borderline strategies based on the user ratings in Table 2.2.

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
Least Misery	3	1	2	1	2	1
Most Pleasure	3	4	5	4	4	5
Dictatorship (User A)	3	1	2	1	4	5

Whilst all of the previously presented strategies take into account the strengths of the preferences, majority-based strategies take only into account the relative position of items in each individual's preference list [78]. Table 2.5 shows how to apply the Borda Count strategy. For every user, it assigns 0 points to the lowest rated item, 1 point to the second lowest item, and so on. In case of ties, points are distributed. Another example of a majority-based strategy is Approval Voting, a strategy where the group members vote for items. If a user votes for an item, this item receives 1 point. The group rating is determined by the sum of the votes. Table 2.6 shows an example where the group members voted for items they somewhat liked (in this example, every item of Table 2.2 with a rating of 3 or higher).

Table 2.5: Example of the Borda Count strategy based on the user ratings in Table 2.2.

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
User A	3	0.5	2	0.5	4	5
User B	1.5	3.5	5	3.5	1.5	0
User C	2.5	0	5	4	1	2.5
Group	7	4	12	8	6.5	7.5

Table 2.6: Example of the Approval Voting strategy based on the user ratings in Table 2.2.

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
User A	1	0	0	0	1	1
User B	1	1	1	1	1	0
User C	1	0	1	1	0	1
Group	3	1	2	2	2	2

2.4.2.2 Aggregating Recommendations

The idea of AR is to generate a recommendation for every group member individually before all recommendations are combined into one group recommendation. The simplest approach to create a group recommendation is to recommend a set containing the top item(s) of all group members [10]. Then, it is up to the group members to select their favorite item(s) from the recommended set. Felfernig et al. [25] presented the concept of *aggregated predictions* which extends the idea of AR. In this case, the group members' predictions for candidate items are aggregated which leads to a ranking of candidate items. Social Choice strategies from AP can be used to aggregate the predictions (see Section 2.4.2.1).

The recommendation techniques for individuals introduced in Section 2.1 can be used when aggregating predictions. For instance, CF can be applied to predict ratings of group members before aggregating them in the next step [25].

2.4.3 Group Recommender Systems in Published Literature

In the following, we present examples of GRSs from different domains. These examples illustrate how to apply the previously presented group recommendation techniques in practical applications. In this thesis, we used similar techniques for our own solutions and developed novel techniques based on these approaches (see Chapter 7).

2.4.3.1 Overview of Existing Group Recommender Systems

MusicFX was one of the first GRSs using a preference aggregation strategy inspired by Social Choice to recommend items to groups [82]. It aggregates the music preferences of gym members to select the music played in the gym. It uses a variant of the Average strategy. The users can rate music stations, such as *Alternative Rock*, on a scale from +2 ("I love this music") to -2 ("I hate this music"). These ratings are increased by 2 to convert them to positive numbers and then squared to widen the gap between well rated and poorly rated music stations. After that, the list of music stations is sorted by the sum of ratings so that the most popular categories are on top. The candidate set is limited to a defined number of music stations. The probability of selecting a music station is then calculated by dividing the rating of this music station by the sum of the ratings of the other music stations in the limited candidate set. MusicFX was installed in a gym and according to a poll conducted six week after the installation, the majority of the respondents thought that MusicFX improved the music selection. However, 15% of the respondents complained about occasional bad music which is a downside of the Average strategy.

Let's browse recommends webpages to groups [83]. It uses active badges worn by the participants to identify the current group members and updates the recommendation when a new user enters or leaves the group. The individuals' user profiles are created by performing a keyword frequency analysis on their webpages. Let's browse uses a simple linear combination of the user profiles to recommend the page which scores the best according to the aggregated profile.

O'Connor et al. [84] were the first to use CF in a GRS. They developed PolyLens, an extension of the movie RS MovieLens[4]. PolyLens uses the Least Misery strategy to merge the recommendation lists of all group members. A field study showed that PolyLens was mainly used by small groups of two or three users. 77% of the participants mentioned that they find group recommendations more useful than individual recommendations when deciding upon a movie. Another GRS for movies was presented by Quijano-Sanchez et al. [85]. Their approach uses CF to generate recommendations while taking into account aspects such as personality and trust.

Popescu [86] developed a voting mechanism for a playlist RS which encourages the group members to state their preferences truthfully. First, the users rate songs and the ratings are normalized so that all ratings given by a user sum to 1. The rating of a song is computed as the sum of the ratings given by the individual users. The probabilistic weighted sum method does not select the item with the highest rating. Instead, the rating is normalized by the sum of the ratings and the result is a probability distribution which determines the probability of the song to be chosen. The results of a small user study showed that the probabilistic weighted sum method is a promising strategy to aggregate user preferences.

2.4.3.2 Group Recommender Systems in Tourism

A few RSs suggest travel-related items to groups of users. In the following, we present important examples of GRSs in the field of e-tourism. RSs in the tourism domain differ from those of other domains, such as movie RSs, as they recommend physical items. Hence, for the success of a GRS in the tourism domain it is even more critical to make recommendations that satisfy all group members as the costs of visiting physical items, such as POIs, are more expensive for every group member than watching a movie, for example [4].

INTRIGUE [87] recommends lists of POIs to groups; however, it does not combine POIs along a route. Recommendations are tailored to the needs of subgroups, such as children and disabled (Figure 2.7). It applies a weighted Average strategy to generate group recommendations. In this strategy, every subgroup is assigned a weight that takes into account the size and relevance of the subgroup.

Pocket Restaurant Finder is a GRS for restaurants [88]. The group members can rate different features, such as the cuisine, the distance they are willing to travel, and the budget. The user's individual preference for each restaurant is determined by calculating the ratings of the single features and adjusting them according to the relative weights specified by the user. The group profit for a restaurant is the average of all individual preferences for this restaurant.

The Travel Decision Forum supports users in agreeing on attributes of a trip, such as desired room facilities and leisure activities [80]. It allows users to view and copy the preferences of the other group members which may reduce the effort of specifying own preferences and support the learning from other users. However, revealing preferences

[4] `https://movielens.org/` (accessed February 16, 2020)

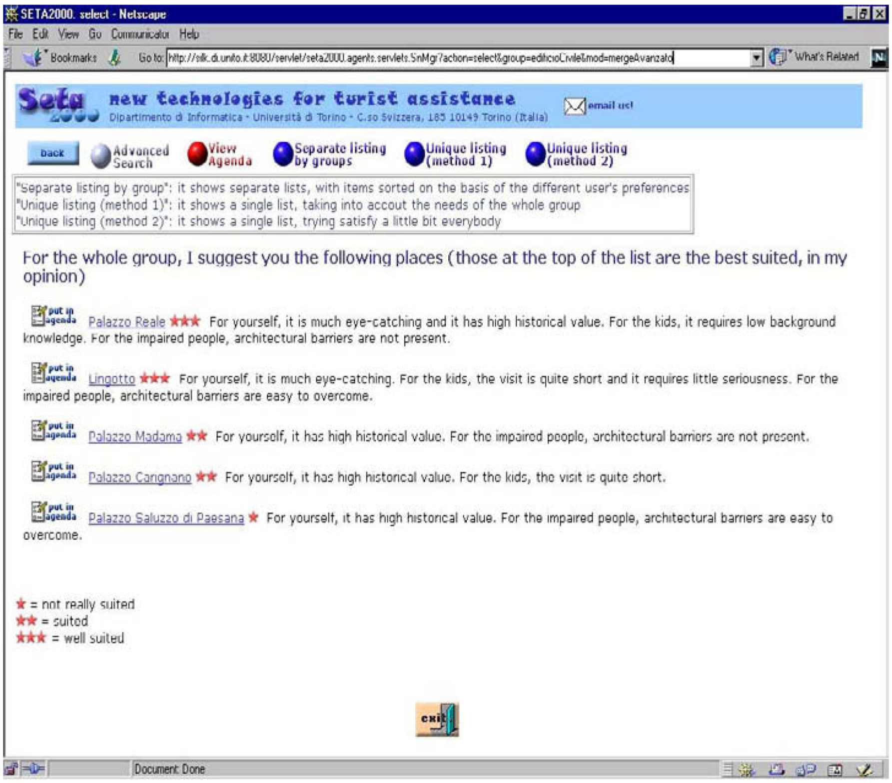


Figure 2.7: The GRS INTRIGUE [87]. In the unique listing of tourist attractions, recommendations are ordered by the predicted rating for the whole group (red stars) while explanations are tailored to subgroups.

allows manipulation. Users can give low ratings to items that are liked by other group members to prevent these items from showing up in a recommendation. The author implemented the MEDIAN strategy to overcome this problem. The TRAVEL DECISION FORUM also offers a random choice as well as an automatically generated mechanism which is non-manipulable as well.

CATS is a case-based RS proposing ski packages that consist of several attributes, such as a location, ski lift systems, and hotel features [40]. It allows group members to critique features of the cases which updates the recommendations accordingly. Furthermore, a case can be added to a stack area which indicates that the user is interested in this vacation. Users can also discard cases which makes a recommendation of this vacation impossible. Hence, the recommendation strategy in CATS can be seen as a variant of the LEAST MISERY strategy [11]. The final recommendation is drawn from the stack area.

Another GRS that applies critiquing is WHERE2EAT. It recommends restaurants in Bolzano, Italy [89]. Each restaurant is composed of the five features price, type, location, facilities, and cuisine. Recommendations can be made for pairs of users. One user can request a recommendation by specifying desired restaurant features. Then, the system suggests restaurants similar to the user query. The user picks one recommendation and forwards it to the other user who can either accept or refuse the suggestion or critique features to receive a new list of recommendations. The user can then send the counterproposal which comes with an automatically generated explanation. This procedure continues until the group agrees on a restaurant.

Nguyen and Ricci [90] developed STSGroup, an extension of the previously presented mobile application South Tyrol Suggests. STSGroup supports groups in finding a POI to visit. The focus of their work is the discussion stage of the group decision making. For this purpose, they integrated a chat-based interface into the mobile application that allows group members to communicate with each other. When using the chat, users can suggest POIs which can be rated and commented by other group members. In addition, users can ask for group recommendations to automatically suggest POIs to the group. POIs that can be recommended are represented by 84 features. These features are weighted according to the importance of the features for the users, which are collected through the group members' actions during the group discussion. The individuals' weights are then aggregated by using a weighted Average strategy which favors users who provide a lot of feedback in the chat or who are in a vulnerable context, such as bad mood.

Recently, Benouaret and Lenne [91] presented a package-to-group recommendation framework which recommends sets of POI packages to tourist groups. They developed two models that incorporate the user impact in selecting items. They compared their models to two baseline approaches in an experiment using the Yelp challenge dataset[5]. Their study was conducted with artificial groups from the dataset and the presented models were not integrated in a practical application.

2.5 User Interfaces for Recommender Systems

The previous sections describe how RSs generate recommendations. Individuals and groups who want to receive a recommendation have to interact with the RS through its UIs to specify preferences, send queries, and view results.

Over the course of the last decades, practical RS applications have been developed for different types of platform-UI configurations: desktop and web-based UIs, mobile applications, and DUIs that distribute UIs over multiple devices. Each of these UI types comes with different advantages and limitations. The selected configuration has a large impact on the success of a RS, especially in the tourism domain. The requirements of tourists towards a RS strongly depend on the context of use. For instance, tourists may prefer wide screens with large UIs when planning trips in advance but also want to receive information on their mobile devices when already on the move [20]. Furthermore, the requirements can change when interacting with a RS in a group instead of traveling alone as additional aspects, such as privacy and embarrassment, become more important [11].

[5] `https://www.yelp.com/dataset/challenge` (accessed February 16, 2020)

In this section, we present the most important types of platform-UI configurations for RSs and highlight their advantages and limitations. This overview served as input for the development of our own prototypes which we present in Sections 6 and 8.

2.5.1 Desktop and Web-Based User Interfaces

RSs started to become successful on e-commerce websites which used them to draw the user's attention to potentially interesting items, such as movies, books, and news [3]. In the early 1990s, the history of RSs started as manual filtering systems which allowed users to query for items they were looking for [92]. RSs were soon developed for operating systems with graphical UIs. They were integrated into news clients, such as Emacs Gnus and NN for UNIX machines and NewsWatcher for Macintoshes [93], or came with world wide web interfaces [32, 33, 83, 94].

Web-based UIs became particularly popular among tourism RSs as they enable a user-friendly travel planning. They allow "displaying a large amount of data extended with maps, images or even high quality videos. Moreover, the mouse permits to interact easily with the computer and move through maps, perform zoom actions, select items or even drag and drop them" [20]. Many web-based RSs that support tourists in their travel planning have been published in the last years (see Section 2.3).

Since web-based RSs are designed to be accessed via personal computers or notebooks with large screens and often provide many interaction options, they are usually not optimized for the usage while on the move [20].

2.5.2 Mobile Recommender Systems

With the widespread adoption of mobile devices, such as smartphones, tablets, and wearables, people can now access RSs anywhere, anytime. They hence overcome the limitations of traditional web-based RSs which are not designed to be used while already on the move. For instance, tourists can use mobile RSs to discover new POIs and update their planned trips whenever necessary [20]. Another advantage of mobile devices is that they usually come with a large number of sensors. Data collected by these sensors can be used to make context-aware recommendations (see Section 2.2). For example, mobile devices equipped with GPS can use the current location of the user to provide recommendations in the vicinity. Such sensors can also determine whether the current situation is appropriate for a recommendation, that is, the user is not busy with other, critical tasks and can benefit from a recommendation in the current context. If this is the case, the RS can proactively suggest items. The concept of recommending items without explicit user request is called *proactive recommendations* [37].

Mobile devices, however, come with some limitations compared to desktop computers. Screen size is usually smaller and input is limited since many mobile devices come only with virtual keyboards or small physical keyboards. This makes it more difficult and time consuming to specify queries, browse large amounts of data, and search for the right information [95, 96]. Also, the availability of mobile internet can vary and users can be completely offline when traveling abroad without a suitable data plan. Nevertheless, the rapid advancement of mobile devices helps overcoming some of these limitations [20].

2.5.3 Public Displays

Public displays are mediums deployed in public spaces that bring digital content to the general public [97]. They can be found in shopping malls, airports, and at any public place that is of interest to locals and tourists to display relevant content to passersby (Figure 2.8). Many of the existing public display applications, such as digital timetables, have an information-only purpose, but advances in technology allow shifting more towards interactive public displays. Interactive public displays have the great advantage that they can tailor their content to the users' needs. Imagine an interactive display in a shopping mall or at a touristic area which does not only show a static map, but also highlights shops or POIs the user might be interested in.

Figure 2.8: A public display integrated into a so-called mobility station of the *Münchner Verkehrsgesellschaft* (Munich Transport Company). The selected application shows a map with POIs in the vicinity.

Public displays vary in size from small television screens to display static information, such as visitor information in museums, to large and interactive multi-user wall displays [98]. Besides their size, public displays can be differentiated based on offered input types and interaction techniques [99]. For instance, users can directly interact with the touch screen or use keys that are attached to the display. More sophisticated methods include voice commands and gestures that are captured by cameras, for example.

Social embarrassment is a factor that often prevents people from interacting with a public display. Brignull and Rogers [100] examined how people behave in front of public displays, how they approach them, how they interact with them, and how they socialize around them. The authors noticed a *honey-pot* effect around the public display. This effect describes the progressive increase of people near the display. Progressive increase in this context means that people standing around the display and open for discussions give a tacit signal to others in the vicinity. Consequently, the number of people around the display and thereby also the number of people tacitly inviting others to join the gathering increases gradually. Furthermore, the authors identified three *activity spaces* when examining the flow of public interaction around public displays:

- *Peripheral awareness activities*: Activities not related to the display, such as eating or socializing.
- *Focal awareness activities*: Activities associated with the display, such as talking about it and watching the screen being used by others.
- *Direct interaction activities*: People in this activity space are using the public display.

Interactive public displays and their applications have to be designed in a way that motivates people to cross the thresholds to *focal awareness* and *direct interaction* and overcome social embarrassment [100]. Michelis and Müller [101] presented the *audience funnel* framework which is based on the aforementioned *activity spaces*. It adds the phase of *subtle interaction* which is specific to gesture-based displays. In this phase, the user is still some meters away from the display but tries to cause some reaction by it. Furthermore, multiple interactions and follow-up actions, such as taking a picture, are added to the framework. Based on the observations of people interacting with the Magical Mirrors, a set of four large public displays with gesture-based interaction installed in Berlin, Germany, the authors found out that the biggest challenge for public displays is initiating subtle interaction.

Privacy is another issue that prevents people from interacting with a public display. A six-month field evaluation of the digital public notice area Digifieds in Oulu, Finland confirmed that even though people showed interest in interacting with a public display, some of them felt uncomfortable when entering personal information [102]. Passersby could have a look on personal data that the user is inputting or sensitive content which is not meant for the eyes of strangers, such as the next location that the user will visit. Brudy et al. [103] call this phenomena *shoulder-surfing*. It has been shown that using a mobile device to enter personal information is one promising solution to overcome privacy issues [104] (see Section 2.5.4). Other solutions to protect users from *shoulder-surfing* are flashing borders when a passerby enters a defined area around the display and blacking out parts of the visible content [103].

Huang et al. [105] provide recommendations for the design of large, public displays based on another field study they conducted by observing the behavior of people towards 46 public displays located in three cities in Western Europe. The authors recommend to minimize text and present informative content to arouse the interest of passersby which do not to spend more than a few seconds to determine whether the public display is of their interest. Displays should be positioned close to eye-height to encourage glances while dynamic content can prolong the user's attention. Another important recommendation is to give the user at least some control over what information is presented, that is, offer a personalization of content. Their study, however, was limited to non-interactive displays presenting non-urgent content. Further studies are required to show which of these recommendation can be applied to interactive public displays.

The work of Alt et al. [106] summarizes important RQs, study types, and methods for evaluating public displays in field studies and laboratory studies. Furthermore, the authors provide a set of guidelines for designing public display studies.

2.5.4 Distributed User Interfaces

The limitations of the previously mentioned UIs motivated researchers to distribute them across multiple devices, allowing them to overcome the limitations of each device. A distribution of UI elements can be done across different dimensions, not only devices. Elmqvist [107] included five dimension into his definition of DUIs:

> "A distributed user interface is a user interface whose components are distributed across one or more of the dimensions input, output, platform, space, and time."

He defines the dimensions as follows:

- *Input*: Managing user input can be distributed across several different devices.
- *Output*: Graphical output can be distributed across several different devices.
- *Platform*: Distributing across different platforms affects different architectures, operating systems, and networks, for example.
- *Space*: UIs can be distributed geographically.
- *Time*: UIs elements distributed in time work asynchronously.

Another definition, which focuses on different aspects that have to be considered in a distribution, is provided by Vanderdonckt [108]:

> "A UI distribution concerns the repartition of one or many elements from one or many user interfaces in order to support one or many users to carry out one or many tasks on one or many domains in one or many contexts of use, each context of use consisting of users, platforms, and environments."

The key aspects in Vanderdonckt's definition are elements, UIs, users, tasks, domains, and the context of use. Vanderdonckt presents a set of questions that have to be answered to distribute UIs. The presented aspects provide the answers to these questions:

- *Distribute what?* Any UI element can be distributed. Pixels are the most atomic level where distribution can occur.
- *Distribute from what?* All distributed elements should belong to one or many clearly identified UIs.

- *Distribute for who?* DUIs are often used by different users who can be co-located or work at different places.
- *Distribute for which?* Tasks can distributed by dividing them into subtasks that are carried out on different platforms, for example.
- *Distribute on what?* Tasks cannot only be attached to one single domain model, but also several, potentially distributed domain models.
- *Distribute across what?* According to Vanderdonckt, the platform is a parameter that significantly influences the design of DUIs. UIs can be produced for several devices simultaneously, migrated from one device to another, or divided across devices, displays, or platforms.
- *Distribute where?* The environment is the social and physical setup in which a user is working on a task. Different UIs can be offered to the users when the environment changes to adapt to the current situation.

Migratory UIs describe a similar concept. DUIs are distributed across one or more of the aforementioned dimensions, whereas the migration of UIs describes "the action of transferring a UI from a device to another one, for example from a desktop computer to a handheld device" [109]. While the original idea covered only migrating whole applications between systems, more recent migratory UIs also distribute at an UI component level [107]. This is why we use only the term *Distributed User Interface* in this work to describe migratory UIs and DUIs.

DUIs promise to overcome the limitations of the previously presented platform-UI configurations. In the tourism scenario, users can use a website to plan a trip and a mobile application to access relevant information while already on the move. Kenteris et al. [110] developed the myMytileneCity guide which allows users to select interesting content, such as lodging, sightseeing, and entertainment, on a website. Then, the system generates an application which can run on mobile devices. The application does not need internet access; content can be updated when the user is online again.

DUIs can also facilitate interaction with public displays and surfaces and thereby increase the user's privacy. Schmidt et al. [111] showed how to use mobile devices in a stylus-like fashion to interact with surfaces. Their approach facilitates authentication on surfaces, for example. Users can touch a password field on the surface to activate it and then enter the password on the mobile device hidden from the eyes of others. Alt et al. [104] developed the aforementioned digital public notice area Digifieds and used it to compare a wide range of different interaction techniques to create new content, post content on Digifieds, and retrieve content from the display. Content can be created directly on the display using a virtual keyboard, a mobile client, or a web browser on a remote PC. While content created on the display appears directly on the screen, users of the mobile or web client can either choose an alphanumeric code or QR code captured by a camera and attached to the display to place content on the screen. Another option for mobile devices is the *phone/display touch feature* which allows users to touch the screen with the mobile device at the position where they want the content to appear. Content can be retrieved on the mobile device, the website, as printout, or via e-mail. 20 participants tested all combinations except for the alphanumeric code technique which was not part of this study. Results showed that there is no single perfect interaction technique. The usability is best for creating content directly at the display but creating content on a PC is significantly faster than at the display or on a mobile device. However, young and technology-savvy users and users on-the-go prefer the mobile device interaction techniques. Vepsäläinen et al. [112] examined which methods work best to redirect the web browser of a mobile device to a webpage acting as the screen controller. In a small laboratory usability study, they compared NFC, QR code, typing an URL, and connecting to a WiFi access point. The usability of the URL method is higher than of NFC and WiFi at the 10% significance level. QR code is ranked second with regard to usability. Other approaches in published literature use changing bluetooth device names [113] or manipulate live video images on mobile devices to update content on remote or public displays [114].

Only a few RSs for DUIs have been developed in the last years. Abdrabo and Wörndl [115] developed DiRec, a DUI for video recommendations. DiRec distributes parts of the application across a mobile device and a larger display (Figure 2.9). This allows users to rate a recommendation on the mobile device while consuming it on the large display, for example. The authors compared the UX of DiRec with a version that runs only on the mobile device. DiRec outperformed the mobile version with regard to stimulation and novelty, but it was also more difficult to get familiar with the DUI approach than the mobile application.

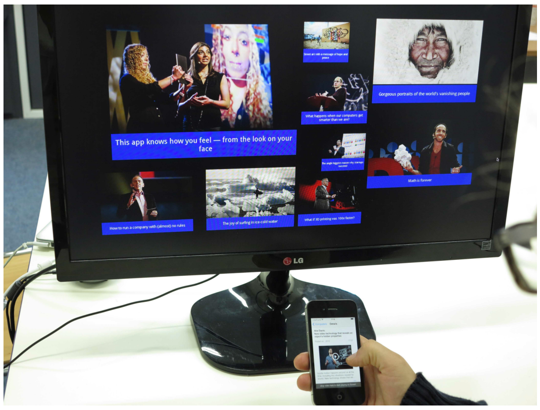


Figure 2.9: The mobile and large display UIs of the distributed RS DiRec [115].

2.6 Evaluating Recommender Systems

The process of evaluating a RS helps us to measure its success. Gunawardana and Shani [116] differentiate between three types of RS evaluations: Offline experiments, online evaluations, and user studies. In the following, we describe these evaluation types.

2.6.1 Offline Experiments

Offline experiments use pre-collected data, such user ratings, to evaluate the performance of a RS. One of the most popular metrics in such evaluations is the RS's accuracy [117]. Accuracy indicates how close a RS's prediction for a user, e.g., the ranking or exact rating of an item, differs from the user's actual ranking of preference or rating [118]. Accuracy can be measured by using some parts of the data to train the recommender model and hide other parts which are then predicted using the trained model. Other metrics for offline experiments are precision, recall, and F-measure [116]. Precision measures the rate of false positives, that is, items that are predicted to be a good recommendation but actually not appreciated by the user. Recall, on the other hand, measures how good a RS is in not leaving out recommendations that should have been presented to the user [119]. F-measure is the harmonic mean of the equally weighted precision and recall [116]. Compared to the more popular accuracy, precision, and recall, it has rarely been used in RS evaluations [119].

The advantage of offline experiments is that they do not require any interaction with the actual users; only pre-collected data are required. However, offline experiments can cover only certain aspects, such as accuracy, which is not always a suitable metric for measuring the success of a RS [15]. For example, a user who receives a set of accurate but similar recommendations may not be satisfied with the RS because of the low diversity of the recommendations. If a user is looking for novel or positively surprising recommendations, accurate recommendations which lack serendipity can have a negative impact on the user's satisfaction with the RS. Other aspects that have an impact on the success of RSs are the usability and the perceived usefulness of the RS [120]. Besides providing accurate recommendations, RSs should also be a pleasure to use [15]. It is hence important to evaluate RSs from a user's perspective instead of relying solely on offline experiments. Online evaluations and user studies are two evaluation types that allow overcoming the limitations of offline experiments.

2.6.2 Online Evaluations

Online evaluations are conducted with real users on real systems solving real tasks [116]. Hence, they allow to evaluate the true value of a RS from different perspectives. With online evaluations, the overall system goals, such as profit generated by the users and retention, can be evaluated. Different aspects of the RS can be varied in an A/B test setting to evaluate the impact of an updated recommendation algorithm or UI, for example. However, online evaluations are more expensive than offline experiments. They require real users, a fully working system, and can have a negative impact on the users' perception of the product when the tested modifications lead to a significant downgrade for the users.

2.6.3 User Studies

User studies overcome some of the limitations of offline and online evaluations [116]. They allow to evaluate the user interaction with a RS but are conducted with users specifically selected for the user study. A user study allows observing participants that solve tasks on a RS, taking notes, and measuring qualitative data, such as how long it takes for the participants to solve a task. In addition, questionnaires and qualitative questions can be used to gather additional data, such as the system's usability.

Two established questionnaires for measuring the usability of interactive systems are the System Usability Scale (SUS) and the User Experience Questionnaire (UEQ). The SUS is a questionnaire that consists of ten usability statements with five response options on a scale ranging from strongly agree to strongly disagree [121]. The SUS is calculated by converting every response to a score from 0 to 4 where 4 is the best score. The common response to half of the statements is strong agreement, and to the other half, strong disagreement, to avoid response biases. The scores are summed up and multiplied by 2.5 to get a total score between 0 and 100. An SUS score above 68 is considered as above average [122].

The UEQ is a questionnaire “that allows a quick assessment done by end users covering a preferably comprehensive impression of user experience” [123]. It is a semantic differential with 26 items grouped into six UX aspects:

- *Attractiveness*: Measures the overall impression.
- *Perspicuity*: Measures whether the system is easy to use and understand.
- *Efficiency*: Measures whether the system helps the users to accomplish their tasks without unnecessary effort.
- *Dependability*: Measures the level of control that the users feel while interacting with the system.
- *Stimulation*: Measures whether the users are excited and motivated to use the system.
- *Novelty*: Measures the level of interest that users feel about the system and whether they think that it is an innovative system.

The UEQ comes with a benchmark dataset that allows comparing the performance of each aspect to other systems. It classifies the six UX scales of the tested system into five categories (compared to the benchmark dataset): excellent, good, above average, below average, and bad.

Compared to the previously mentioned types of evaluations, user studies are the only method that allows collecting qualitative data which can be used to interpret quantitative results. However, user studies are expensive; a large number of participants is required to draw statistically significant conclusions. Finding participants can be challenging. Participants should resemble the potential users of the tested RS as closely as possible, which is hard to achieve [117].

Knijnenburg and Willemsen [117] differentiate between user studies, which are smaller observational studies used to improve the usability of a RS, and user experiments, which “denote the use of experimental conditions and formal measurement as a means of testing theories about users interacting with” [117]. They emphasize that user experiments are mandatory for a proper evaluation of the UX of a RS. Two frameworks that can be used to evaluate the UX of RSs in user experiments were introduced by Pu et al. [120] and Knijnenburg et al. [124].

Recommender systems' Quality of user experience (ResQue) is a framework which measures different user-centered aspects, such as the quality of the recommendations, the RS's usability, and the user's satisfaction with the RS [120]. It is based on existing usability-oriented research and uses principles from other, established usability evaluation models.

ResQue comprises 43 questions clustered into 15 constructs. These constructs are, moreover, structured into four layers of higher-level constructs:

- *Perceived System Qualities*, such as the recommendation accuracy, interface adequacy, and interaction adequacy.
- *Beliefs*, such as the perceived usefulness, perceived ease of use, and transparency.
- *Attitudes*, which are more long-lasting than beliefs, e.g., the overall satisfaction and trust.
- *Behavioral Intentions*, which express whether or not the user is willing to use the system and consume the recommendations.

The framework allows researchers to understand how the users' perception of physical features of a RS influences their beliefs, attitudes, and behaviors [120]. Consequently, not only the recommendation accuracy of a RS can be evaluated; furthermore, it is also possible to evaluate the effects of changing system aspects. For example, an updated UI could have a positive impact on the user's satisfaction with the RS and thereby increase the user's interest in using the RS in future.

The Knijnenburg et al. [124] framework has a similar purpose. It is composed of the following components [117]:

- *Objective System Aspects*: Aspects of the system that are currently being evaluated, such as algorithms, rating scales, and layouts.
- *Subjective System Aspects*: The user's perceptions of the *Objective System Aspects*, such as the perceived recommendation quality. Can be measured with questionnaires.
- *User Experience*: The qualities of the RS, such as the satisfaction with the chosen items. Is also measured with questionnaires.
- *Interaction*: The user's interaction with the system, e.g., the number of clicks or time spent with the RS.
- *Personal and Situational Characteristics*: Characteristics, such as domain knowledge and choice goals, measured with questionnaires.

Knijnenburg et al. [124] explain that the framework can be used as a guideline for controlled experiments. For example, it allows measuring the effects of a changing *Objective System Aspect* on the user's perceptions (*Subjective System Aspects*), experience (*User*

Experience), and behaviors (*Interaction*) [117]. Therefore, the framework "allows for a better understanding of why and how certain aspects of the system result in a better user experience" [124].

2.7 Summary

RSs are software tools and techniques that identify products, services, or informations that best satisfy the user's needs. How well an item matches a user's preferences is often expressed in ratings. Different techniques exist to predict ratings of a user u for an item i. CBRSs recommend items that are similar to those the user has liked in the past. They use item categories or keywords extracted from the items to measure the similarity between a user query and an item. Case-based RSs are a variant of CBRSs that rely on very structured representations of items. The goal of CF is to identify items that other users with similar tastes and preferences like. *Nearest neighbor* algorithms can be used to identify similar users. *User-based nearest neighbor* algorithms predict the rating of a user u for an item i by analyzing ratings for i from similar users. *Item-based nearest neighbor* algorithms, however, predict ratings based on other items that were similarly rated by other users. Knowledge-based RSs and demographic RSs are two other types of RSs. Different techniques can be combined to so-called hybrid approaches.

More sophisticated RSs consider not only items and users to predict ratings, but also the context of the recommendation. Different contextual conditions change predicted ratings. For example, outdoor POI are assigned a lower profit on rainy days. Context can be integrated into RSs at different stages, either before or after a recommendation is made or directly within the recommendation process. Contextual factors can either be soft or hard criteria in the recommendation process. If an items does not match a hard criterion, it is completely removed from the list of recommendation candidates. Items that do not match a soft criterion can still be recommended, but the probability of a recommendation decreases.

RSs and context-aware recommendations are particularly popular in e-tourism. RSs in tourism are mainly used to recommend POIs, travel plans, and sequences of POIs. Sequences of POIs along enjoyable routes are also called tourist trips. Recommending tourist trips is a challenging task. Tourists can usually not visit all POIs during their trip. Tourist trip RSs have to identify the most attractive POIs and combine them along a route. The recommendation has to respect several constraints, such as time and budget constraints, opening hours of POIs, and desirable breaks.

In practice, tourists often travel in groups. Groups are aggregates of individuals that are characterized by interactions, mutual goals, interdependencies between the group members, a group structure, and cohesion. Different techniques exist to recommend items to groups. User profiles of group members can be aggregated using Social Choice strategies to create a group profile which is used to request a recommendation. An alternative approach is to make a recommendation for every user individually before the recommendations are combined into one group recommendation.

Users and groups interact via UIs with RSs. Originally, RSs were developed for operating systems with graphical UIs and web browsers. Novel mobile devices, such as smartphones, allow to access recommendations even when on the move. In addition, their sensors promise highly context-aware recommendations. However, their small screen size and dependence on mobile internet can be significant limitations. An alternative to smartphones are displays that are deployed in public spaces, but privacy concerns and social embarrassment can prevent people from interacting with them. Distributing applications over multiple interfaces allows overcoming the limitations of single devices. UIs can be distributed among many dimensions, such as user input and graphical output.

The evaluation of RSs is done in offline experiments, online evaluations, or user studies. A disadvantage of many previous RS evaluations is that they focus on prediction accuracy only. However, RSs should also be a pleasure to use and hence, evaluated from the user's perspective. ResQue and the Knijnenburg et al. framework are two frameworks that can be used to measure the UX of RSs in user experiments.

3 The Tourist Trip Design Problem

The problem of finding a sequence of POIs along a route that respects different user requirements and constraints, such as opening hours of POIs and the time available for the trip, is called the TTDP [6]. Many route planning problems have been introduced to model realistic variants of the TTDP that consider these user requirements and constraints. The integration of such problems into practical tourist trip applications allows the recommendation of personalized POI sequences, as introduced in Section 2.3.3. However, only few state-of-the-art RSs try to solve complex but realistic variants of the TTDP and hence are not able to recommend tourist trips that are entirely tailored to the user's needs. On the other hand, we believe that route planning problems should be extended by established recommendation techniques, such as context-aware recommendations and group recommendations, to be more suitable in practical tourism applications.

In this chapter, we provide a summary of route planning problems that serve as basic models for the TTDP. Extensive overviews of existing algorithms and heuristics solving the described problems have been published [14, 125, 126]. Therefore, we focus on explaining the main idea of each problem and show how it can be applied to model the TTDP. In addition, we summarize open challenges in TTDP research. We used the findings of this chapter to come up with novel solutions to solve the TTDP for individuals and groups.

3.1 Models for the Tourist Trip Design Problem

The majority of literature in the field of tourist trip recommendations uses the so-called OP to model the TTDP [14]. This is why our work is based on the OP and some of its variants which we present in this section. Furthermore, we summarize the most important algorithms and heuristics that have been published to solve the presented problems.

Parts of this overview have been published in [41, 42].

3.1.1 Orienteering Problem

A well-known optimization problem that formulates a simple version of TTDP is the OP. The name *orienteering* originally described a running and navigation sport where the participants have to find and arrive at fixed locations [127]. In one variant of this sport, profits[1] are assigned to the locations and the participants do not have to visit all

[1] In this thesis, we use the term *profit* to describe the predicted value of a POI in a tourist trip for a user. In published literature, the term *score* is often used for the same purpose.

locations but are supposed to maximize the collected profits during the playing time. Hence, this sport requires not only a good stamina but also path finding skills.

More generally speaking, in the OP, several locations with an associated profit have to be visited within a given time limit. Each location may be visited only once, while the aim is to maximize the overall profit gained on a single tour [127]. The problem can be applied to other domains, such as transportation logistics and tourism [6].

The OP can be formulated as an integer problem [125]:

$$Max \sum_{i=2}^{N-1} \sum_{j=2}^{N} s_i \times x_{ij}, \tag{3.1}$$

$$\sum_{j=2}^{N} x_{1j} = \sum_{i=1}^{N-1} x_{iN} = 1, \tag{3.2}$$

$$\sum_{i=1}^{N-1} x_{ik} = \sum_{j=2}^{N} x_{kj} \leq 1; \quad \forall k = 2, \dots, N-1 \tag{3.3}$$

$$\sum_{i=1}^{N-1} \sum_{j=2}^{N} t_{ij} \times x_{ij} \leq T_{max}, \tag{3.4}$$

$$2 \leq u_i \leq N; \quad \forall i = 2, \dots, N, \tag{3.5}$$

$$u_i - u_j + 1 \leq (N-1)(1 - x_{ij}); \quad \forall i = 2, \dots, N, \tag{3.6}$$

$$x_{ij} \in \{0, 1\}; \quad \forall i, j = 1, \dots, N, \tag{3.7}$$

where N is the number of locations than can be visited. s_i is the profit of location i and $x_{ij} = 1$ if a visit to location i is followed by a visit to j, 0 otherwise. t_{ij} is the cost of traveling from i to j and T_{max} is the time budget. u_i denotes the place of location i in the path. The objective function (3.1) is to maximize the sum of the collected profits. The constraints guarantee that the path starts at location 1 and ends at location N (3.2), the connectivity of the path and that every location can only be visited once (3.3), that the total costs of the path do not exceed the time budget (3.4), and prevent subtours (3.5, 3.6).

The OP is NP-hard [128]. Hence, when using the OP as a model for the TTDP to recommend tourist trips in practical applications with a large dataset of locations, heuristics are necessary to ensure reasonable computation times.

In the relevant OP literature, it is differentiated whether a directed or undirected graph is given [14]. In the tourist trip scenario, the locations are POIs that a user can visit during a single-day or multi-day trip. A tourist can access all of these POIs from any point in the city. The cost of traveling between two POIs, which can be expressed by the walking or driving time, for example, remains the same regardless of the direction of movement. Hence, for this scenario, an undirected graph with edges connecting all pairs of vertices should model the travel area. However, there may be exceptions where a directed graph is more appropriate. For instance, if a museum gift shop can only be

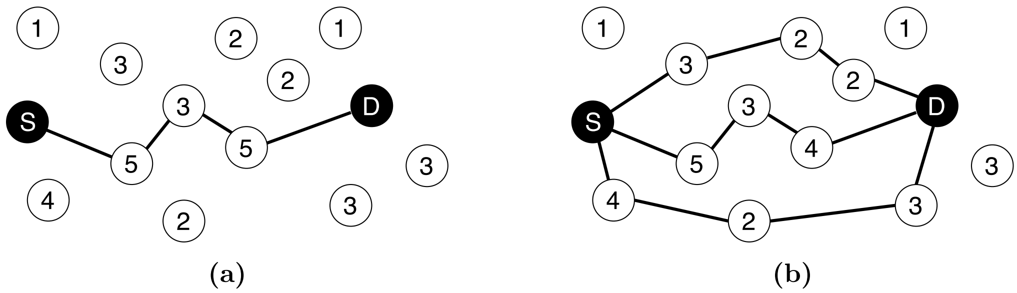


Figure 3.1: Illustrations of the a) OP and b) Team Orienteering Problem (TOP) with $k = 3$ teams. Numbers denote location profits. Edge costs are removed for readability.

reached after visiting a museum, the edge connecting these two vertices is only in one direction. Another differentiation is whether the starting and destination locations are fixed points or not [14]. In the tourist trip scenario, a fixed starting point could either be the user's current location or a desired starting point, such as a hotel. If a fixed destination is required depends on whether the user has a specific destination in mind. Another option is a round trip in which the user wants to return to the starting point at the end of the trip. Figure 3.1a illustrates an example path from a starting point S to a destination D solving the OP.

A large number of exact algorithms and heuristics solving the OP have been developed. Tsiligirides [127] started with suggesting two algorithms based on approximate methods: a stochastic algorithm using a Monte Carlo method and a deterministic algorithm, similar to Wren and Holiday's [129] approach for a vehicle-scheduling problem, which partitions the given area into several sectors. Chao et al. [130] presented a heuristic for the OP which is composed of two steps: initialization and improvement. Only locations within an ellipse over the entire set of points with the start and end points as the two foci of the ellipse are considered for the suggested solution. In the initialization phase, multiple solutions are created by inserting points in a greedy way. The path with the highest profit is selected as the initial solution. Improvements can then be made by exchanging, moving and removing points in the paths. They showed the good performance of this approach by applying it to 107 problems. The heuristic of Golden et al. [128] calculates centers of gravity to find the best route. This approach outperforms both algorithms presented by Tsiligirides. Gendreau et al. [131] described a tabu search heuristic for the OP. By testing this approach on randomly generated instances, they showed that it always yields optimal or near-optimal solutions.

Some recent work also tackles the traditional OP [132–134]. Taylor et al. [135] recently introduced an algorithm solving the OP as an integer linear program which can also take into account must-see POIs [135]. However, the research focus is shifting towards more complex variants of the OP which have been developed to serve as more sophisticated models for the TTDP.

3.1.2 Team Orienteering Problem

The goal of the TOP is to find k routes at the same time maximizing the total profit of all routes [136]. The name of this problem is derived from a team in which each team member selects one route in an attempt to avoid overlaps in the points visited by each team member. In the tourism scenario, a team member could be interpreted as one day in a multi-day trip, for example. Figure 3.1b shows an example solution to the TOP with $k = 3$ teams.

A few exact algorithms solving the TOP have been proposed [137–139]. The first heuristic, MAXIMP, was introduced by Butt and Cavalier [140]. It is composed of four steps: First, weights are assigned to vertex pairs, estimating how valuable it is to include both vertices in the same tour. Then, vertex pairs which are not feasible are removed before vertices are assigned to tours. In the end, a final check tries to find out if adding single vertices to a tour can increase the total profit of that tour. Another solution was presented by Chao et al. [136]. They extended their approach introduced in [130] to find k paths instead of only one. The authors showed that their approach is computationally efficient and outperforms a concurrent version of Tsiligirides's algorithm. Archetti et al. [141] presented metaheuristics for the TOP: two variants of a generalized tabu search algorithm and a variable neighborhood search algorithm. All of these three solutions outperform previous heuristics, however, the variable neighborhood search algorithm turned out to be more efficient and effective for this problem than the two tabu search algorithms they presented. Souffriau et al. [142] presented a Greedy Randomised Adaptive Search Procedure (GRASP) for the TOP. GRASP is a metaheuristic first introduced by Feo and Resende [143]. It performs a number of independent iterations to eventually return the best result. A Path Relinking extension that recombines solutions from different iterations of the original GRASP approach to find better solutions to the TOP was presented by Souffriau et al. [144]. The authors developed a fast and a slower but more accurate variant of the Path Relinking metaheuristic. They showed that the path relinking extension significantly outperforms the basic GRASP algorithm.

Friggstad et al. [145] presented an algorithm solving a problem that can be understood as an extension of the TOP: avoiding low-quality trips in multi-day recommendations. They evaluated their algorithm by using an anonymized Google historical visit dataset and Foursquare public check-in data. Results showed that their algorithm significantly improves the quality of the worst day compared to a state-of-the-art multi-tour algorithm. Another evaluation with human raters showed that the recommended trips score only slightly below trips created by human travel experts.

3.1.3 Orienteering Problem with Time Windows

In the Orienteering Problem with Time Windows (OPTW), each location can only be visited within a defined time window [146]. These time windows can represent the opening hours of POIs, for example. If the TOP is extended by time windows, it is called the Team Orienteering Problem with Time Windows (TOPTW) [147].

Kantor and Rosenwein [146] were the first to solve the OPTW [125]. They developed the tree heuristic and compared it to a simple insertion heuristic. The tree heuristic is based on a depth-first search which creates paths by iteratively adding vertices until a complete path is generated or one of six rules forces the abandonment of the current path. Tests showed that the tree algorithm delivers significantly better results than the insertion heuristic. Vansteenwegen et al. [147] developed an Iterated Local Search (ILS) heuristic solving the TOPTW. It is composed of an insertion step and a shake step. In the insertion step, a new vertex is inserted into the path after verifying that all scheduled visits after the new vertex still satisfy their time windows. In the shake step, vertices can be removed or shifted to improve the solution by escaping local optimums. The authors used a large test set to show that their approach can find high quality paths in a very short time, making the heuristic optimal for usage in practical applications. However, according to Gavalas et al. [148], the heuristic comes with two major weaknesses: (i) vertices with high profits can be left out if they are to time-expensive to reach and (ii) topology areas with a high density of vertices can be left out when vertices with high profits isolated from these areas are included into the solution. They developed two cluster-based heuristics CSCRatio and CSCRoutes extending the ILS to overcome these weaknesses. The main idea behind both heuristics is to organize vertices into clusters based on topological distance criteria. The heuristics try to combine vertices in the same cluster to reduce travel duration. Compared to CSCRatio, CSCRoutes constructs routes that visit each cluster at most once. Both approaches outperform ILS with respect to solutions quality and can reduce the frequency of long transfers between vertices. A recent approach was presented by Hu and Lim [149], who developed a three-component heuristic for the TOPTW. It executes a local search procedure and a simulated annealing procedure to explore the solution space. Then, routes can be recombined to identify high quality solutions. This heuristic outperforms existing approaches in published literature with regard to average performance.

3.1.4 Time Dependent Orienteering Problem

The Time Dependent Orienteering Problem (TDOP) assumes that the time needed to travel between two locations depends on the time the traveler leaves the first location [150]. This extension can be used to model different modes of transportation in a tourist trip recommendation. For example, a tourist can leave a POI later than planned when a bus connection to the next POI is available as the traveling time between the two POIs decreases. Combining the TDOP with time windows and multiple routes leads to the Time Dependent Team Orienteering Problem with Time Windows (TDTOPTW) [151].

Fomin and Lingas [150] introduced the TDOP and provided a $(2 + \epsilon)$-approximation algorithm to solve it. Garcia et al. [151] were the first to develop a heuristic solving the TDTOPTW. It starts with calculating an average travel time between each pair of POIs. Using these averages, a heuristic, which is based on the ILS metaheuristic implemented by Vansteenwegen et al. [147], creates a solution solving the TOPTW. In the end, a repair procedure introducing the real travel times between the POIs is executed.

The authors implemented their heuristic into desktop and mobile prototypes which have been initialized with real data from the Spanish city of San Sebastian. Garcia et al. [152] developed a second approach to solving the TDTOPTW using real travel time between POIs instead of calculating average values. The main concept of this approach is a fast, local evaluation of each possible insertion which involves only the POI that is inserted and the POIs directly before and after the new POI. Again, the authors used a test set with around 50 POIs from San Sebastian to evaluate their approach. Results showed that the real travel time approach outperforms the approach with average travel times. Gavalas et al. [153] extended the previous TOPTW heuristics in [148] to solve the TDTOPTW. They presented two heuristics: TDCSCRoutes and SlackCSCRoutes. TDCSCRoutes modifies CSCRoutes's insertion step to handle time dependent travel times among different vertices. The decision if a vertex is inserted is based on the insertion cost, whereas SlackCSCRoutes takes into consideration the effect of an insertion in the whole route. The authors compared their algorithms with two other algorithms using a dataset with POIs from Athens, Greece. With respect to the total profit of a trip, TDCSCRoutes performs marginally better than the three other algorithms.

3.1.5 Multi Constrained Team Orienteering Problem

The Multi Constrained Team Orienteering Problem with Time Windows (MCTOPTW) introduces additional thresholds besides the time budget which a path is not supposed to exceed [154]. A common constraint when traveling is money. Tourists usually have a limit on how much they want to spend for entrance fees or food, for example. In this case, the vertices come with a fixed cost and the routing algorithm has to find a path which does neither exceed the financial threshold nor the time budget. Souffriau et al. [155] extended the MCTOPTW to the Multi Constrained Team Orienteering Problem with Multiple Time Windows (MCTOPMTW) which allows defining different time windows on different days and more than one time window per day.

Garcia et al. [154] were the first to solve the MCTOPTW. Their metaheuristic is based on the ILS metaheuristic implemented by Vansteenwegen et al. [147]. It takes into account every constraint while checking the feasibility of a vertex insertion when comparing vertices that can be inserted. The authors proved the appropriateness of their approach on different test sets. Souffriau et al. [155] developed a hybrid of the ILS metaheuristic implemented by Vansteenwegen et al. [147] and GRASP and adapted it to solve the MCTOPMTW. Tests on a large dataset showed that the average run of the algorithm has a total profit gap of 5.19% and an average execution time of 1.5 s. Sylejmani et al. [156] developed a Tabu Search approach for solving the MCTOPTW. In this approach, neighborhood exploration is done by using three general moves: insert a new POI into the path, replace a POI from the path with another POI not in the path, and swap two POIs inside the path. In a test set of 148 instances, the heuristic average performance has a gap of 4% from the state-of-the-art approach in [155] and an average execution time of 6 s.

3.1.6 Orienteering Problem with Maximum Point Categories

Bolzoni et al. [157] introduced the Orienteering Problem with Maximum Point Categories (OPMPC), an extension of the OP that considers location categories and introduces the maximum number of locations per category as additional constraint. The authors developed different variants of their algorithm CLIP and evaluated them using two real-world datasets. They proved that CLIP is able to generate trips that are close to the optimal solution. The TourRec problem, introduced by Gionis et al. [158], is a similar extension of the OP which also considers location categories and the order in which the user wants to visit the categories [158]. The authors introduced two different satisfaction functions for the TourRec problem: an additive satisfaction function, which is the sum of each location's profit, and a coverage satisfaction function, which represents the number of locations and activities in the vicinity that the user could visit or join during the trip. The authors developed algorithms for both variants and showed their practical utility and efficacy using a Foursquare dataset.

3.1.7 Arc Orienteering Problem

In the Arc Orienteering Problem (AOP), profits are associated with edges (*arcs*) instead of vertices [159]. As in the traditional OP, AOP algorithms try to find a route from a starting point to a destination that maximizes the total profit while not violating given constraints. However, the goal is to find a trip that is composed of the most attractive routes. Souffriau et al. [159] used GRASP to solve the AOP. Their approach finds a near optimal solution in only 1 s. The AOP is suitable in scenarios where the user mainly benefits from the routes between locations. For instance, Souffriau et al. integrated their approach into two real-life applications that support cyclists in finding attractive routes.

Verbeeck et al. [160] extended the AOP to the Cycle Trip Planning Problem (CTPP) in which a vertex can be visited multiple times. A combination of the OP and the AOP is the Mixed OP [125]. In this variant, profits are assigned to routes as well as locations.

3.1.8 Orienteering Problems with Flexible Location Profits

The aforementioned problems assume that each vertex or edge is assigned a fixed profit. A few works introduced problems where the profit of a location is flexible and depending on certain events. These problems allow a more realistic modeling of the TTDP since the value of a POI can be influenced by the presence or absence of other POIs in a trip, for example [6]. In the following, we present some important models for the TTDP with flexible location profits.

3.1.8.1 Generalized Orienteering Problem

In the Generalized Orienteering Problem (GOP), every location is assigned multiple profits representing different goals of the visitor [161]. Hence, the user's travel purpose can be modeled. If a traveler is planning a trip with sports activities only, a museum will have a low profit even if this user likes museums in general.

The objective function in the GOP is nonlinear. Therefore, the GOP can penalize paths that include two similar attractions, for example. This is important for a realistic modeling of the TTDP to avoid unpleasant combinations of POIs, such as two restaurant in a row [162].

3.1.8.2 Orienteering Problem with Variable Profits

The Orienteering Problem with Variable Profits (OPVP) assumes that the vertex values depend on a number of discrete passes or the time spent at the vertex [163]. This problem allows modeling a version of the TTDP where it is necessary to visit a POI multiple times or stay at POIs for at least a minimum amount of time to fully benefit from the visits.

The authors presented programming models for the case of discrete passes and the case of a continuous amount of time to be spent at the vertex. They developed a branch-and-cut algorithm for both cases and showed that the pass model can be solved for about 200 vertices within two hours of computing time. The continuous time model, however, is beyond the computational reach for more than 75 vertices.

3.1.8.3 Team Orienteering Problem With Decreasing Profits

In the Team Orienteering Problem with Decreasing Profits (DPTOP), the profit of each vertex decreases with time [164]. Hence, the DPTOP can be used to model a variant of the TTDP where the profit of a POI is lower the later the traveler arrives at this POI. The decreasing function of time can, for example, represent the fitness state of travelers if we assume that they enjoy POIs less when being more tired at a later point of the trip.

3.1.8.4 Clustered Orienteering Problem

In the Clustered Orienteering Problem (COP), the profit of a vertex can only be gained if all vertices of a group of vertices are part of the path [165]. Travel-related scenarios where this problem may be used for modeling are complementary exhibitions, events, and plays which become only interesting if all parts are visited.

The authors presented an exact branch-and-cut algorithm and a tabu search heuristic to solve the COP. The exact algorithm is able to solve instances with up to 318 vertices in one hour of computing time. The simple heuristic approach, however, is able to calculate high quality solutions in a short computing time.

3.1.8.5 Orienteering Problem with Stochastic Profits

The Orienteering Problem with Stochastic Profits (OPSP) assumes that the locations' profits are stochastic with a known distribution and their values are not revealed before the locations are visited [166]. The crowdedness of a POI is an example of a stochastic factor in tourism. RSs should suggest POIs which are not too crowded and they can use historical data to predict the number of visitors. The actual number of visitors at a

POI without a mandatory ticketing or a fixed number of places is stochastic, hence the full benefit for visitors cannot be estimated before they arrive at the POI.

The authors suggested an exact parametric solution technique and a Pareto-based bi-objective genetic algorithm to solve the OPSP.

3.1.8.6 Orienteering Problem with Stochastic Travel and Service Times

Campbell et al. [167] introduced the Orienteering Problem with Stochastic Travel and Service Times (OPSTS) in which travelers are punished if they do not reach a location before a deadline. They exemplify the problem using the example of a company providing deliveries or services to their customers: a reward is received for all customers which can be reached on the planned route before the deadline and a penalty is received for those not reached. The problem can also be used to model the TTDP if we assume that a user's satisfaction decreases with every POI missed in a given area and time frame. In this case, the total profit of a trip is the sum of the POIs' profits minus the penalty for every POI not being visited during the trip.

3.1.9 Orienteering Problems for Groups of Users

The goal of the aforementioned problems is to create one or more routes for individuals. Recently, a few works proposed variants of the OP which try to find routes for a group of users. This is an important extension when modeling the TTDP since tourists often travel in groups.

Lim et al. [168] introduced the GroupTourRec problem which has three objectives: (i) cluster users into groups, (ii) recommend a tourist trip to the group, and (iii) recommend a tour guide to the group. Hence, the goal of GroupTourRec is not to make recommendations to pre-defined, individual groups. Instead, it connects people with similar interests.

Anagnostopoulos et al. [16] introduced TourGroup, an extension of the OP, to create tours that satisfy all members of a group by finding a compromise route. They presented three different formulations of the problem: TourGroupSum, TourGroupMin, and TourGroupFair. The objective functions of these problems are derived from preference aggregation strategies which are based on Social Choice Theory. TourGroupSum tries to maximize the sum of the individuals' satisfactions, TourGroupMin tries to make the least satisfied person as happy as possible, and TourGroupFair tries to optimize the overall group satisfaction while penalizing overly unfair solutions. The authors presented a set of algorithms solving the TourGroup problem: a dynamic programming heuristic, multiple greedy heuristics, BUMA (an algorithm that can use any of the presented algorithms to recommend a route to each group member and then selects the one which is best for the whole group) and an ant-colony optimization algorithm. Furthermore, they implemented an exhaustive search algorithm, which is an exponential-time algorithm returning the optimal solution, as benchmark. The authors created datasets of POIs and users in the Italian cities of Pisa, Rome, and Florence using different sources, such as Wikipedia and Flickr, to evaluate their algorithms. Results showed that the ant-colony heuristic always gives high-quality solutions with reasonable execution times. Its running times are significantly faster than the ones of the dynamic programming heuristic and BUMA which also find the optimal solution most of time. The greedy heuristics should be chosen if very quick route creation is required.

Sylejmani et al. [7] introduced the Multi Constrained Multiple Team Orienteering Problem with Time Windows (MCMTOPTW), a problem to model the case of multiple trips with multiple tours while taking the social relations between the different group members into account. They presented four different approaches to solving this problem:

- Solo: individual users conduct the trip alone.
- Subgroups: the group is split, users with common preferences and mutual social relations travel together.
- All together: all users stay together.
- Combined: users are together at times and separated at other times.

The authors developed a new algorithm based on a tabu search metaheuristic. The fast mode of this algorithm takes about 20 s to obtain better personalized trips for tourist groups than when scheduling the whole group together. They used a newly generated test set to compare the solutions obtained by different approaches. Results showed that the combined approach performs better than any of the other approaches. The all together approach performs better than the solo and the subgroups approaches and is four times faster than the combined approach. However, when applying a single tour algorithm on averaged user preferences, the personal preferences of the individual group members get lost.

3.2 Open Challenges

The OP and its variants have been researched for over 30 years and became very popular models for the TTDP. Various extensions allow modeling more complex variants of the TTDP that consider multi-day trips, opening hours of POIs, and multiple constraints, for example. Table 3.1 summarizes all problems presented in this chapter, their key characteristics, and how they can be applied in the tourism scenario.

We identified three important challenges that research in TTDP still has to meet: context-aware tourist trips, a user-centered perspective on TTDP algorithms, and tourist trips for groups of users.

Some of the existing TTDP models, such as the OPTW, introduce aspects which can be understood as contextual factors from a RS perspective: if a location is recommended depends on its opening times, which is an example of a temporal contextual factor [8]. The influence of different contextual factors on single POI recommendations has been researched in the last years [35]. However, there are contextual factors specifically relevant for sequences of POIs, such as the order of POIs in a trip and how visiting a POI influences the perceived quality of the remaining trip. The attractiveness of routes

Table 3.1: Summary of models for the TTDP from published literature with their most relevant characteristics and relevance for the tourism scenario. (Note: for some of the listed problems, team variants for finding k tours have been introduced.)

Problem	Constraints & parameters	Relevance for tourism scenario
OP	Time budget	Recommend POIs along a route
OPTW	Time windows	Respect opening hours
TDOP	Travel time dependency	Integration of different modes of transportation
MCTOPMTW	Multiple budgets, time windows	Multiple travel constraints (e.g., time and budget)
OPMPC	Location categories	Limit number of POI categories
TourRec	Total distance covered, location categories	Specify number of POI categories
AOP	Profits assigned to arcs	Recommend routes instead of POIs
CTPP	Profits assigned to arcs, vertices can be visited multiple times	Recommend routes instead of POIs
Mixed OP	Profits assigned to vertices and arcs	Recommend routes and POIs
GOP	Multiple profits, nonlinear objective function	Model travel purpose, penalize similar attractions
OPVP	Profit depends on number of discrete passes or time spent at vertex	Visit POIs multiple times, specify minimum duration of stay
DPTOP	Profit decreases with time	Penalize late arrivals at POIs
COP	Profit of a vertex can only be gained if all vertices of a cluster are visited	Recommend complementary POIs and activities
OPSP	Stochastic profits (not revealed before vertex is visited)	Model stochastic factors, such as crowdedness of a POI
OPSTS	Penalty for vertices not visited before a deadline	Decreasing user satisfaction if POIs are missed
GroupTourRec	Multiple users and profits, collective group interest, tour guides	Cluster travelers into groups, recommend trips and guides to groups
TourGroup	Multiple users and profits	Recommend tourist trips to groups
MCMTOPTW	Multiple users, profits, and constraints	Multiple trips with multiple tours

between POIs can also have a large influence on the perceived quality of a recommended trip and therefore be considered as another type of contextual factor. TTDP algorithms have to integrate such contextual factors to make the recommendations more suitable for practical applications. In Chapter 5, we show how to use combinations of different OP variants to meet this requirement. We introduce a novel problem that enables flexible location profits to model all types of contextual factors that are relevant in a tourist trip RS. It considers the user's preferences for location categories, such as the OPMPC and TourRec problem, but without the requirement to specify fixed category bounds. Furthermore, we present a solution that integrates ideas from the Mixed OP to recommend tourist trips that also consist of attractive routes.

One main focus in published OP literature is the development of heuristics with little gaps to optimal solutions and quick execution times in order to utilize them in practical applications. The optimal solution in this case is the maximum possible sum of location profits that can be reached without violating the given constraints. In practical applications, the satisfaction with a recommended trip is not necessarily the sum of POI profits. Many individual aspects influence how a tourist perceives the quality of a trip. Hence, evaluating a tourist trip from a pure OR perspective is not suitable to determine the most satisfying tourist trip recommendation. Instead, tourist trips generated by OP algorithms should be evaluated from a user-centered perspective to find trips that are not only a near-optimal solution from a pure mathematical perspective, but instead a pleasure for the user. One approach to achieve this goal is to integrate TTDP algorithms into RSs. A RS can take into account many aspects influencing the perceived quality of a recommendation, such as context, and therefore be used for a user-centered approach to generate tourist trips. This is why we integrated all of our solutions to the TTDP into practical applications and evaluated them in user studies with real users and groups in this thesis.

The vast majority of OP studies focus solely on the creation of tourist trips for individuals. Since tourists often travel in groups, approaches solving the TTDP for a group of users are necessary. GRSs apply preference aggregation strategies from Social Choice Theory to create a user profile representing the whole group. This approach can be adapted to solve the TTDP for a group of users, however, an investigation on which strategies work best to find a consensus is still missing [7]. Such an investigation should also evaluate the influence of social-psychological aspects, such as group type, on how well a preference aggregation strategy can support a group in finding a consensus. In Chapter 7, we present different group recommendation strategies for solving the TTDP and compare them to novel approaches, such as a solution that allows groups to split temporarily during a trip. Until today, very few works have solved the TTDP for user groups. Both [16] and [7] evaluated their approaches in experiments using datasets but created synthetic groups. To the best of our knowledge, we are the first to evaluate TTDP strategies for groups in a user study using a practical application and with real groups.

3.3 Summary

The problem of finding a sequence of POIs along a route is called the TTDP. A large number of algorithms and heuristics solving the TTDP exist. The majority of them use the OP to model the TTDP. In the OP, several locations with an associated profit have to be visited. Each location may be visited only once. The travel time between locations and the maximum time available for the trip limits the number of locations that can be visited. The goal is to find a route that maximizes the overall profit. Team variants, such as the TOP, have been introduced to find multiple routes and maximize the total profit of all routes.

Further variants of the OP allow a more realistic modeling of the TTDP. The OPTW introduces time windows which can represent opening hours of POIs, for example. The TDOP assumes that the time needed to travel between two locations depends on the time the traveler leaves the first location. It can hence be used to model public transport in the TTDP. The MCTOPTW introduces multiple constraints (e.g., time, budget) that limit the number of locations in a trip. In the OPMPC, location categories are considered. The maximum number of locations per category can be specified as additional constraint. Other variants introduce flexible location profits. For instance, the profit of a location can change depending on the presence or absence of other locations in the same trip or when the user does not spend a minimum time at a location.

Very few works have solved the TTDP for user groups. This is an important issue as tourists often travel in groups. However, recommending tourist trips to groups remains an open challenge in research. Besides recommendations for groups, TTDP algorithms should also be context-aware and evaluated from a user's perspective instead of using datasets only. In this thesis, we combine different variants of the OP, such as the OPTW, the Mixed OP, and variants with flexible location profits, to model all types of contextual factors in a RS. We apply existing group recommendation strategies to the TTDP, develop novel approaches, and compare them in user studies with real groups to solve the TTDP for groups from a user-centered perspective.

4 A Framework for the Development of Practical Tourist Trip Recommender Systems

In this chapter, we introduce AnyRec, a framework that we developed to facilitate the development of practical RSs and the evaluation of TTDP algorithms and UIs from a user-centered perspective. AnyRec aims to be the starting point of any generic RS. It provides common components that are required in most RSs, such as user management, data gathering and dispatching, Application Programming Interface (API) interfaces, and evaluation from an end user's perspective. The framework makes it convenient to implement new or improve existing user clients, recommendation algorithms, and data sources.

We used the AnyRec framework to develop the tourist trip RS TourRec. In this chapter, we introduce AnyRec and show how it can be used to develop and evaluate novel recommendation algorithms and UIs. Furthermore, we explain the general idea of TourRec and present all of its components. The outcome of this chapter is an architecture that supports the development of practical applications solving the TTDP from a user-centered perspective. The TourRec RS that we developed based on this architecture serves as the basis for answering all of our RQs in the following chapters.

4.1 Existing Tools and Frameworks for the Development and Evaluation of Recommender Systems

Many existing machine learning services support developers in creating and training models. Popular commercial examples are Amazon Machine Learning[1], Google Cloud Machine Learning (ML) Engine[2], and Azure ML Studio[3]. These services can often be combined with different machine learning frameworks, such as Tensorflow[4], and can be used for different machine learning tasks, such as predicting ratings for recommendations. An example of an open-source machine learning server is Prediction IO[5]. Apache Mahout[6] is a machine learning framework facilitating the development of RSs by providing a set of CF algorithms, for example.

[1] https://aws.amazon.com/aml/ (accessed February 16, 2020)
[2] https://cloud.google.com/ml-engine/ (accessed February 16, 2020)
[3] https://studio.azureml.net/ (accessed February 16, 2020)
[4] https://www.tensorflow.org/ (accessed February 16, 2020)
[5] https://predictionio.apache.org/ (accessed February 16, 2020)
[6] https://mahout.apache.org/ (accessed February 16, 2020)

Research has been focusing on providing tools and frameworks for supporting research, development, and evaluation of RSs. LensKit for Python is such an example of an open-source toolkit [169]. It is a set of Python tools for RS research and development which provides various modules for splitting data for cross-validation, algorithm APIs for training models, and top-N and prediction accuracy metrics for evaluation purposes. Guo et al. [170] developed LibRec, an open-source Java library that implements more than 70 recommendation algorithms and a set of evaluation metrics. Another framework for the implementation and evaluation of recommendation algorithms is RankSys [171]. It is implemented in Java 8 and targets the ranking task problem with a focus on novelty and diversity. Mihelčić et al. [172] presented a RS extension for RapidMiner, a data science platform for machine learning. The goal of their extension is to simplify and speed up the creation of new RSs.

These examples help researchers to develop and evaluate RSs. Their focus, however, are often machine learning tasks, such as predicting ratings by supporting developers in creating and training models, offering recommender algorithms, and supporting the evaluation of the recommendations using established metrics, such as accuracy. The framework that we present in this chapter has a different purpose and should rather be understood as a complement for the mentioned tools. It does not provide a set of recommendation algorithms. Instead, it focuses on the overall developer experience of the RS. By providing a RS skeleton, the framework facilitates the development and deployment of novel user clients, recommendation algorithms, and data sources. Furthermore, it supports the evaluation of every component from a user's perspective. Consequently, RSs developed using our framework do not only focus on high prediction accuracy, but support users in the whole process of finding best items. We used these strengths of our framework to develop TourRec, a practical tourist trip RS that solves the TTDP from a user-centered perspective.

4.2 System Overview

AnyRec is a multi-tier architecture that is partitioned into three tiers: presentation tier, application logic tier, and data tier (Figure 4.1). In addition, external services can be integrated into the RS. External services are typically third party data providers that provide the items that can be recommended.

AnyRec is domain-independent, any kind of item can be recommended depending only on the available data sources. One main advantage of AnyRec is its modularity. Through well-defined interfaces and communication via the Hypertext Transfer Protocol (HTTP) protocol, novel data sources, clients, and recommendation algorithms can easily be added independent of programming language or runtime environment. Furthermore, the framework enables the evaluation of RSs from a user's perspective. For example, multiple recommendation algorithms can be tested in an A/B testing approach and evaluated according to UX criteria, such as the perceived quality of recommendations in the desired client application.

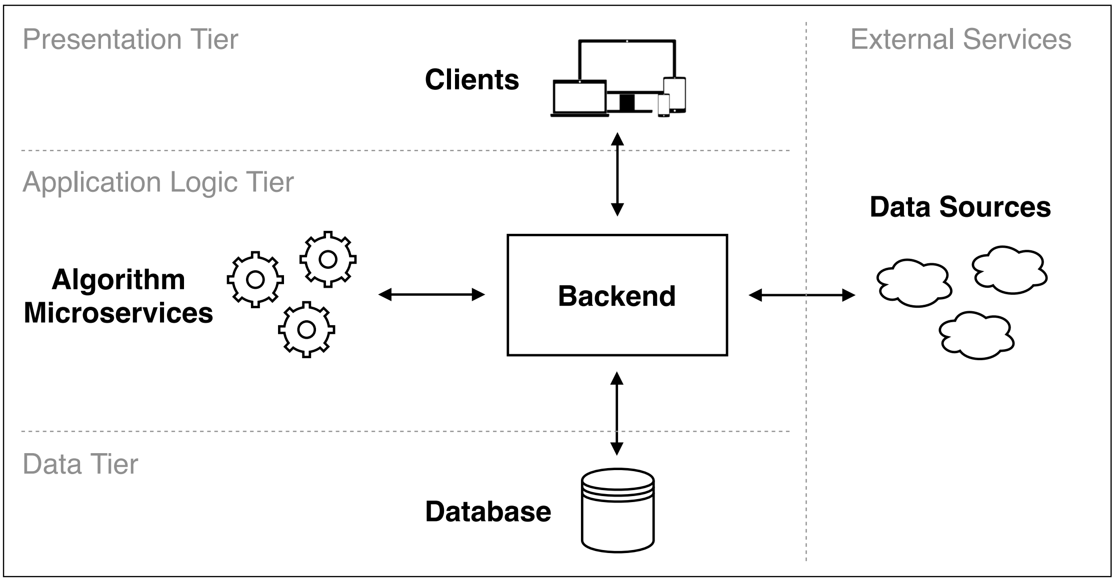


Figure 4.1: The AnyRec architecture.

In the following, we present each of the three tiers in detail. We demonstrate the capabilities of our framework by presenting TourRec, a RS for tourist trips that we developed using AnyRec. We explain how we can use TourRec to solve the TTDP from a user-centered perspective and answer the RQs of this thesis.

A brief description of the presented architecture has been published in [173].

4.2.1 Presentation Tier

The entry point for all user requests are the clients. They are part of the presentation tier and should be the front facing UIs of the RS. Clients can be any suitable UI for RSs (see Section 2.5) and developed using any programming language. A client is responsible for gathering user preferences and other relevant user input data, such as the user's current location. The user request is then sent to the backend which is part of the application logic tier. Users eventually receive and view recommendations in the client application.

We developed and evaluated several client applications for TourRec within the scope of this thesis (see Chapters 6 and 8). Individuals and groups can use these clients to specify their travel preferences and all of the important constraints for solving the TTDP. The recommended trips are displayed on the clients and can be rated by the users. Consequently, TourRec's presentation tier allows us to answer the RQs *"Which platforms and UIs support tourists the best in solving the TTDP in realistic scenarios with regard to different usability and UX criteria?"*, *"How do different group types agree on decisions when interacting with a GRS for tourist trips and how fair are their decisions?"*, and *"Which platform-UI configurations for receiving group recommendations support groups the best when looking for a tourist trip with regard to different UX criteria?"*

4.2.2 Application Logic Tier

The backend is the core of the AnyRec framework. It acts as the connecting link between the clients, the data sources, and the recommendation algorithms. The framework provides well-defined HTTP endpoints for user management, authentication, recommendation queries, and feedback that each client must adhere to.

The backend receives incoming recommendation requests from the clients via HTTP requests. The framework defines a list of middleware that all requests must pass through before being further handled by the application. These middleware validate the enveloped data against a JSON schema provided by the developer, for example. This step ensures that the following components can operate on the expected data structure.

After the request passed the middleware, necessary data from external services are gathered. A custom adapter for each external service is required. The adapter is responsible for building the request and defines the output data structure for its corresponding external service. Once AnyRec gathered the required data for a recommendation from the clients and all external services, the request is handed over to the recommendation algorithm. The recommendation algorithm receives the full user request including the user preferences and all data from the external data source. AnyRec allows integrating multiple recommendation algorithms into the RS. All algorithms are extracted into their own dedicated microservice. They communicate with AnyRec over well-defined HTTP interfaces. Only one recommendation algorithm receives the recommendation request. By default, AnyRec chooses an algorithm mircoserivce randomly, but developers can also assign selection probabilities. This facilitates the A/B testing approach where multiple algorithms can be evaluated. The selected algorithm microservice responds back to AnyRec with the generated recommendation.

The backend is based on the PHP framework Laravel 5.8[7]. AnyRec provides skeletons for algorithm microservices written in PHP and Java; however, any programming language or framework can be used to built recommendation algorithms. The whole framework is built with mircoservices in mind and the AnyRec components are containerized with Docker[8] which consist of a complete and isolated run time environment.

The TourRec backend fetches POIs that can be recommended in a trip from the Foursquare Places API[9]. Contextual data are fetched from other external service. For instance, weather data are provided by the OpenWeatherMap API[10]. In this thesis, we developed multiple tourist trip algorithms in Java and PHP which we present in the following chapters. The results of the studies that we conducted to evaluate these algorithms allow us to answer the RQs *"How can existing TTDP algorithms be extended to increase the satisfaction of individuals with the recommended trips?"* and *"Which group recommendation strategies provide the highest user satisfaction when solving the TTDP for groups?"*

[7] `https://laravel.com/` (accessed February 16, 2020)
[8] `https://www.docker.com/` (accessed February 16, 2020)
[9] `https://developer.foursquare.com/` (accessed February 16, 2020)
[10] `https://openweathermap.org/` (accessed February 16, 2020)

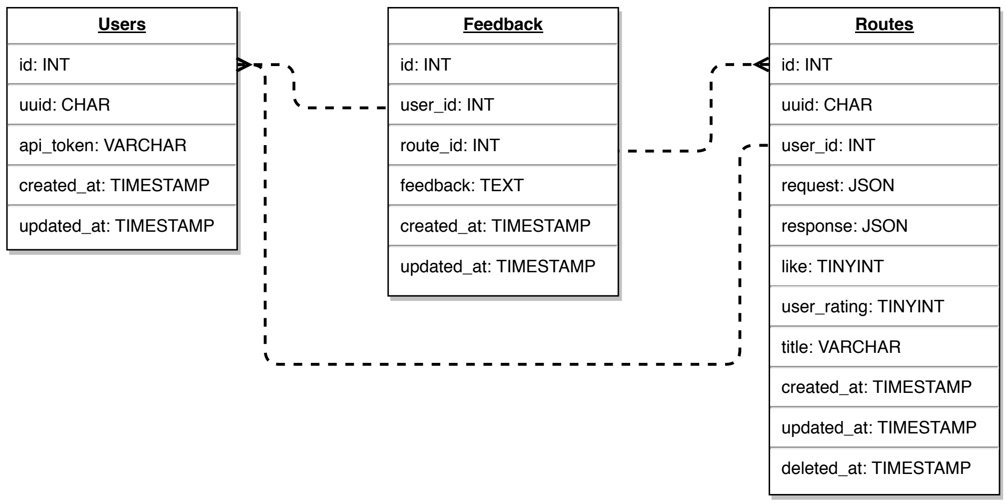


Figure 4.2: TourRec database schema.

4.2.3 Data Tier

AnyRec stores all relevant data from the user requests to the actual recommendation in the data tier and assigns them to unique identifiers for later reference. The data tier comprises a database which is also used to store user profiles, API credentials, and evaluation data, such as user ratings of recommendations. Note that in our framework, the items to recommend are not stored in our database but instead provided by the external data sources. The framework's default database is MySQL[11], but PostgresSQL[12] and MariaDB[13] are also supported.

TourRec stores all relevant trip data in a MySQL database. This includes the user request with travel preferences, the recommended trip, the algorithm which was used to generate the recommendation, and the user ratings and feedback for the recommended trip (Figure 4.2). These data allow us to gather quantitative and qualitative feedback when evaluating different TourRec components in this thesis. The data tier therefore supports us in answering the previously mentioned RQs.

4.3 Summary

In this chapter, we presented AnyRec, a framework supporting the development of practical RSs and their evaluation from a user's perspective. It is a multi-tier architecture that is partitioned into three tiers: presentation tier, application logic tier, and data tier. The client applications are part of the presentation tier. They allow users to specify queries, receive and view recommendations, and provide feedback on the recommended trips. The application tier consists of the backend and the recommendation

[11] `https://www.mysql.com/` (accessed February 16, 2020)
[12] `https://www.postgresql.org/` (accessed February 16, 2020)
[13] `https://mariadb.org/` (accessed February 16, 2020)

algorithms. The backend receives user requests, fetches relevant data from external services, and forwards all data to the algorithms. One of the algorithms is then selected to make a recommendation which is returned to the client by the backend. The data tier is used to store important data, such as user requests and corresponding recommendations. The modular architecture of the framework facilitates the implementation of novel user clients, recommendation algorithms, and data sources. Furthermore, it supports researchers in evaluating these components by enabling A/B tests and storing qualitative and quantitative user feedback.

We used the AnyRec framework to develop TourRec, a RS for tourist trips. TourRec serves as the basis for answering the RQs in this thesis. In the following chapters, we present the different recommendation algorithms and UIs that we developed to solve the TTDP for individuals and groups.

5 User-Centered Solutions to the Tourist Trip Design Problem for Individuals

Many algorithms and heuristics solving the OP and its variants for individuals have been developed (see Chapter 3). However, existing approaches solve these problems only partly from a user-centered perspective, as they take into account only basic user constraints, such as time and budget. As described in Section 3.2, next-generation MTGs should also consider different types of contextual factors, such as the weather, time of the day, and previously visited POIs, to better adapt tourist trip recommendations to the user's needs. For instance, outdoor POIs should be avoided on rainy days and some restaurants are only suitable for lunch or dinner. Another factor that can be understood as contextual factor is the attractiveness of routes between POIs in a trip. In some scenarios, such as relaxing city walks, the quality of routes between POIs is considered at least as important as the selection of POIs. Furthermore, only few works integrated solutions to the TTDP into practical applications and evaluated them in user studies. Consequently, it is not known if solutions that are optimal from a mathematical perspective also satisfy real users in realistic scenarios.

In this chapter, we describe the problem of recommending tourist trips to individuals and explain the additional requirements that result when solving the problem from a user-centered perspective. We present an algorithm to solve the TTDP for individuals that is based on Dijkstra's algorithm for shortest paths and was developed in close collaboration with this project. We introduce several extensions to this algorithm that allow a more realistic modeling of the TTDP. These extensions include the integration of contextual factors that have an impact on the profits of POIs as well as route attractiveness attributes that influence the selection of routes between POIs in a trip. We present the user studies that we conducted to evaluate our extensions and summarize the results of an online evaluation that we conducted using the live version of the TourRec application. The findings of this chapter allow us to answer our first RQ: *"How can existing TTDP algorithms be extended to increase the satisfaction of individuals with the recommended trips?"* The results serve, furthermore, as a basis for the following chapters.

5.1 Problem Description

The formulation of the TTDP that we solved in this thesis is a variant of the traditional OP, as introduced in Chapter 3. A user is looking for a single-day trip recommendation. The trip starts at a specified starting point and ends at a different destination. Furthermore, the user can specify the maximum duration of the trip. Every location that the user could visit on the trip has a profit. The goal is to find a trip from the starting point to the destination that maximizes the total profit for the user without exceeding the time budget. This problem can be described as a graph problem with the POIs as vertices and the connection between the POIs as edges [174]. The graph is undirected and every vertex in the graph is connected to all other vertices because a traveler can choose to visit all the POIs from any location. The number of POIs in a trip depends on the distances between the POIs, which are denoted by the edge weights, and the specified time budget.

We extended the described problem by further constraints to better adapt the recommendations to the user's needs. In our scenario, POIs are not only vertices with a fixed profit. Instead, every location is assigned a realistic POI category, such as *Food*. As explained in Section 3.2, our solution should consider the user's preferences for POI categories and contextual factors, such as the previously visited POIs, the weather, and the time of the day, to better adapt the recommendations to the user's needs. Consequently, users should be able to specify their interests by rating categories. The ratings and the impact of the contextual factors determine the profit of a POI. Another important requirement is that the recommended trip should be suitable for a walking tourist. Hence, the distance between the starting point and the destination has to be limited. Furthermore, the routes between the recommended POIs should be perceived as attractive.

We tackled the described problem by iteratively improving a tourist trip algorithm that is based on Dijkstra's algorithm [175]. In the following, we present different variants of this algorithm. Then, we introduce a context-aware variant of the algorithm and show how to integrate route attractiveness attributes. All variants were evaluated in user studies.

5.2 A Tourist Trip Recommendation Algorithm based on Dijkstra's Algorithm

In this section, we present an algorithm solving the TTDP from a user-centered perspective. It is based on Dijkstra's algorithm and extends a preliminary solution which has been introduced by Iltifat [174]. While Dijkstra's algorithm is an iterative algorithm that determines the shortest path between two vertices in a graph with non-negative edge weights, in our scenario, we are looking for a path that maximizes the total profit of the trip. We explain two variants of this approach: a constraint-free and a constrained-based variant. Both variants are based on the same abstract model to solve the TTDP [41]:

1. Retrieving and scoring of items, based on user preferences and context.

2. Combining and grouping the items to form a composite trip.

The tourist trip algorithm that we describe in this section has been developed by Wörndl and Hefele [176] and in close collaboration with this research project. It served as a basis for all of our own approaches in this chapter. Furthermore, we used this algorithm as a basis for solving the TTDP for groups of users in the following chapters. This is why we describe the idea, implementation, and evaluation of the Dijkstra-based tourist trip algorithm in detail in this section. The following description is adapted from [41], which is an extended and updated version of [176].

5.2.1 Retrieving and Scoring Points of Interest

According to the problem description in Section 5.1, a tourist trip recommendation is made from a starting point to a different destination. The recommended trip has to contain POIs that are located in the area around and between both locations. Hence, potential POIs in this area have to be identified and their profits determined before an algorithm can propose a tourist trip that consists of some of these POIs.

The solution introduced by Wörndl and Hefele used Foursquare Places API as data source which consists of 105M places around the world. Developers can search for POIs and ask for recommended places in an area via a RESTful API. Preliminary tests showed that the Foursquare API returns more diverse POIs in various categories compared to other, similar APIs, such as Google Places [174]. This is why we also decided to use Foursquare as POI data source in this thesis.

The Foursquare API allows specifying a circular or elliptical region for the POI search. In their initial solution, Wörndl and Hefele set the maximum distance between starting point and destination to 5 km to make the trips suitable for walking tourists. The search region for the Foursquare API request is determined as follows: First, the midpoint between starting point and the destination is determined. Then, a circle is drawn around this midpoint with the distance between starting point and midpoint multiplied by 1.2 as radius. The Foursquare API limits the number of places returned per request. Consequently, multiple requests with an offset parameter have to be sent if there are more POIs in the specified region than returned by one request.

Foursquare POIs are assigned one or many categories. The categories can be very specific and follow a hierarchical structure to define subcategories.[1] For example, a *Food* POI can be assigned to the following hierarchy of categories: *Food* → *German Restaurant* → *Bavarian Restaurant*. In their initial solution, Wörndl and Hefele used only a few number of Foursquare top level categories that are typical for travel-related activities: *Sights and Museums*, *Night Life*, *Food*, *Outdoors and Recreation*, *Music and Events*, and *Shopping*.

Users specify their travel preferences by rating every POI category on a scale ranging from 0 (not interested in this category) to 5 (strongly interested in this category). The final profit of a POI, furthermore, considers the Foursquare rating of the POI and the number of votes, i.e., how many people rated the POI. The approach applies a logarithmic scale of votes for the final profit of a POI to put more weight on the POI rating:

$$profit = rating \times log_2(number\ of\ votes + 1). \tag{5.1}$$

Only POIs that fulfill the following requirements are candidates for a recommendation:

[1] The full list of Foursquare categories is available under `https://developer.foursquare.com/docs/resources/categories` (accessed February 16, 2020).

- The POI has a Foursquare rating, that is, a sufficient number of Foursquare users have checked in or left reviews for the POI,
- the POI is overall well rated,
- there are not more than ten venues in the POI's category (otherwise some of the worst rated POIs of this category are removed), and
- the user did not rate the POI's category with 0 (otherwise all POIs of this category are removed).

The set of candidate POIs is used to build a weighted graph with the POIs as vertices and the connection between the POIs as edges (see Section 5.1). The edge weight is defined as the distance (beeline) between two POIs which can be calculated using the latitude and longitude information of two POIs.

5.2.2 Constraint-Free Algorithm

The constraint-free variant of the Dijkstra-based algorithm does not consider any user constraints related to travel cost and trip duration when recommending tourist trips from a starting point to a destination. Instead of choosing the subpath with the shortest distance in each iteration of the Dijsktra-based algorithm, this variant prefers paths that maximize the fraction *entertainment/distance*. *entertainment* in this scenario is defined as the sum of the profits of all POIs in the recommended trip. *distance* is the total path length.

It has been shown that the number of POIs in each of the six categories differs greatly. For instance, *Food* is often the most returned POI category. This can lead to unsatisfying results that contain too many restaurants, even when the category *Food* has a low user preference. The presented approach uses the PCC to better correlate the POIs in a recommendation to the user preferences. For example, if a user rates *Shopping* with a 4 and *Sights and Museums* with a 2, the recommended tourist trip should contain roughly twice as many POIs in the category *Shopping* than in *Sights and Museums*. Consequently, the algorithm wants to maximize:

$$\frac{r(preferences, number\ of\ places\ per\ category\ in\ path\ so\ far) \times entertainment^2}{distance^2},$$

where the correlation coefficient r increases the number of places in a category that the user likes, but that is also underrepresented in the set of discovered places. Preliminary tests showed that the algorithm's performance can be improved if *entertainment* and *distance* values are weighted more than the PCC, which is why they are squared.

5.2.3 Constraint-Based Algorithm

The basic principle of the constraint-based variant of the Dijkstra-based algorithm is the same as in the constraint-free variant. The only difference is that the constraint-based variant takes into account time and budget constraints for the trip. The user can specify these constraints with each request. Furthermore, each POI is assigned values for cost and time to spend at the POI. Then, a weighted graph is created, as explained in Section 5.2.1. Every time a subpath is compared against another path in the Dijkstra-based algorithm, it is verfied that the subpath does not violate time or budget constraints. In addition, a trip recommended by the constraint-based variant cannot contain more than one restaurant or more than one nightlife POI.

If all of these conditions are met, the subpath is compared to the priorly found best solution. Similar to the constraint-free variant, the algorithm tries to maximize:

$$r(preferences, number\ of\ places\ per\ category\ in\ path\ so\ far) \times entertainment.$$

Compared to the constraint-free variant, the constraint-based algorithm uses only the profits of POIs to compare subpaths, but not the distance between the POIs. The distance, however, is used for creating the subpath and as an overall trip constraint.

The recommended duration of stay at each POI is based on the findings of Melià-Seguí et al. [177]. They derived heuristics for some of the six POI categories based on a real world Foursquare dataset with 3.7M users and 300M check-ins. For instance, they found out that users spend 41 min on average for breakfast, 53 min for lunch, and 1 h 39 min for dinner. In their work, Wörndl and Hefele used 45 min as a rough estimation of the time to spend at POIs in the category *Food*. Some of the values identified by Melià-Seguí et al. were less suitable for the presented tourist trip scenario because the observed categories did not match any of the six POI categories or because of disproportionately long durations of stay for a single-day tourist trip. For instance, the average duration of stay for *Arts & Entertainment* is around 5 h. In order to allow users to do multiple activities on a single-day tourist trip, more realistic estimations with no value greater than 60 min were used as suggested durations of stay. In addition, the suggested durations of stay are adjusted with regard to user preferences. The assumption is that tourists who like certain POI categories a lot want to spend more time than average at such POIs. Table 5.1 summarizes how the suggested durations of stay are adjusted.

Table 5.1: Adjustment of suggested durations of stay based on the user's rating for a POI category.

Rating	Adjustment
0 (worst)	No recommendation
1	−15 min
2	−5 min
3	No adjustment
4	+5 min
5 (best)	+15 min

The cost estimation for a POI is based on Foursquare's price categories. Foursquare assigns POIs to one of the four categories *cheap*, *moderate*, *expensive*, and *very expensive*. Wörndl and Hefele converted this categorization into concrete values: € 8 for *cheap*, € 16 for *moderate*, € 24 for *expensive*, and € 32 for *very expensive*.

Some Foursquare venues are not assigned to a price category. For instance, a fixed price assignment is often not possible for POIs in the category *Outdoors and Recreation*.

5.2.4 Evaluation

Wörndl and Hefele implemented both variants of the Dijsktra-based algorithm in PHP. Furthermore, they developed a web-based client application to evaluate the algorithms in a user study. Users were able to specify their travel preferences and optional time and budget constraints on the client application. If constraints were enabled and entered, the constraint-based algorithm was used, otherwise the constrained-free variant.

In the following, we describe the user study setup and summarize the most important results, as presented by Wörndl and Hefele.

5.2.4.1 Setup

Both the constraint-free and constraint-based variants were compared to the preliminary solution of Iltifat [174] which did not integrate the PCC into the path-finding algorithm and used different estimations for durations of stay and costs. For this purpose, the web-based client application presented two tourist trip recommendations for each user query: one tourist trip for the preliminary version, one for the improved approach. The two trip recommendations were presented to the user as red and blue paths on a map and as a list with the POI categories. The users were not aware that different variants of a tourist trip algorithm were used and the assignment to the two colors was randomized. At the bottom of the recommendation UI, a short survey composed of the following five questions was displayed for both trip recommendations:

1. The total number of places was...? (too low / low / perfect / high / too high)
2. The length of the path was...? (too short / short / perfect / long / too long)
3. How well did your received places match your preferences? (not at all / rather not / fairly well / quite well / perfectly)
4. Would you consider taking this route yourself? (no / maybe / yes)
5. How satisfied are you with the overall result? (not satisfied / rather not satisfied / rather satisfied / quite satisfied / very satisfied)

Table 5.2 lists the answer scales and best possible value when mapping the response options to scales from 1 to 5 and 1 to 3, respectively.

The survey also contained a question asking the users if they preferred the red or blue trip recommendation. In addition, an input text field for optional comments was

Table 5.2: Answer scales and best possible response options for the Dijkstra-based algorithm user study.

Question	Answer scale	Best value
1	1 – 5	3
2	1 – 5	3
3	1 – 5	5
4	1 – 3	3
5	1 – 5	5

provided. The link to the publicly available web application was shared via e-mail lists and Facebook groups.

5.2.4.2 Results

The web application was accessed over 600 times during the evaluation period. In total, 533 recommendations were made. Feedback was submitted 123 times for tourist trips in cities all over the world. Hence, the collected dataset contained feedback for 123 tourist trips generated by the preliminary solution and 123 tourist trips generated by the improved algorithm. 85 of these 123 data records for the improved algorithm were made for the constraint-free variant, and 38 for the constraint-based variant. The results for both variants are aggregated in the following.

The average distance between the specified starting points and destinations was about 2.5 km ($s = 1.2$). The actual trip length for both trips was about 5.6 km on average. Furthermore, trips generated by the preliminary solution contained 14.3 POIs on average and trips generated by the improved algorithm 15.4 POIs on average.

Outdoors & Recreation was the most popular POI category with an average user rating of 3.48. In contrast, *Shopping* was the least popular category with 28 users rating the category with 0, which means that they wanted no POIs from this category in their recommendation. Table 5.3 shows all average category ratings and the detailed distribution of the ratings given by the users.

Figure 5.1 illustrates how the preliminary solution and the improved algorithm performed with regard to each of the five questions of the questionnaire. Both algorithms performed equally with regard to the first two questions. When analyzing the responses for the improved version in detail, however, the authors found out that about 50% of respondents stated that the total number of places was *perfect*, about 12% thought the number was *too low*, and about 16% found it *too high*. Furthermore, more than 70% of the users found the length of the recommended trips *perfect* and only in less than 5% of the cases, the users rated the trip length as *too short* or *too high*. This indicates that the improved approach can recommend reasonable trips. The responses for the third question confirmed that the modified representation of user preferences in the improved approach led to higher user satisfaction. Slightly more than 20% stated that they would

Table 5.3: User preferences by categories.

Category	0	1	2	3	4	5	Mean
Sights & Museums	8	10	20	25	31	29	3.2
Nightlife	23	21	21	24	18	16	2.33
Food	9	14	14	29	37	20	3.07
Outdoors & Recreation	6	3	12	38	33	31	3.48
Music & Events	13	9	20	33	29	19	2.92
Shopping	28	29	20	27	14	5	1.88

not use the recommendation, while around 40% answered with *maybe* and around 40% with *yes* to question 4. Slightly more users favored the improved algorithm. The results for the fifth question show that the users were more satisfied with the improved approach. 62% of the users were *quite satisfied* or *very satisfied* with the results.

38% of the users liked the recommendation from the preliminary version better than that of the improved algorithm. 47% preferred the recommendations made by the improved approach and 15% preferred no algorithm.

33 participants submitted anonymous, textual comments. The majority of comments were positive. Many users liked the idea of the tourist trip RS or a particular recommendation. Some users commented that there were too many restaurants in the recommendations made by the constraint-free alternative; for example, one user commented:

> "I have the feeling, that the system should not recommend more than 2 restaurants without explicitly marking them as alternatives, just because I can't eat in all 10 suggested one's. I think it would be good to make this dependent on the category since for shopping 10 shops might be a good fit." [41]

Based on the results of this study, we decided to use the constraint-based variant of the improved algorithm as input for our own work. In the following, we explain how to extend this algorithm to a context-aware variant and introduce a solution that considers the attractiveness of routes between POIs. Finally, we present additional ideas to improve the Dijkstra-based tourist trip algorithm and summarize the results of an online evaluation that used the live version of the TourRec application to evaluate our ideas.

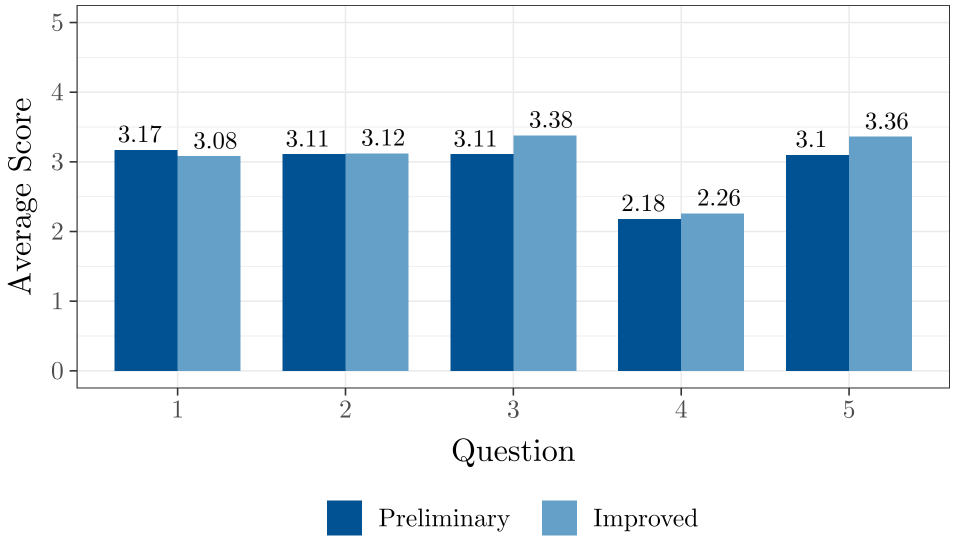


Figure 5.1: Average results for each of the five questions in the evaluation of the Dijkstra-based tourist trip algorithm.

5.3 Context-Aware Tourist Trip Algorithm

The previously presented approach for recommending tourist trips does not consider any contextual factors. In Section 2.2, we explained the concept of CARSs and how they can improve recommendations. In this section, we propose a novel, context-aware tourist trip recommendation algorithm that enhances the previous, Dijkstra-based solution. It incorporates various kinds of contextual information, including two contextual factors that are especially relevant for POI sequences. We present the contextual factors that our CARS observes and explain how the respective contextual conditions ratings have been acquired. Then, we show how we integrated the contextual information into our novel algorithm. We compared our algorithm to the previous, context-unaware approach in a user study.

The content of this section has been published in [178] with some revisions.

5.3.1 Acquiring Context Relevance

We designed an online questionnaire to acquire quantitative measures of how selected contextual factors influence a user's decision of going to a POI. The following approach assesses the context relevance and is based on the Baltrunas et al. [35] methodology (see Section 2.2).

For each of the six top level POI categories introduced in Section 5.2.1, we asked the participants to rate how a given condition would influence the decision of going to this POI. The questionnaire covered the following contextual factors and conditions:

- *Previously visited POI (category)*: Arts & Museum, Food, Music Event, Nightlife, Outdoors & Recreation, Shopping
- *Time of the day*: Morning (8am - 12pm), Midday (12pm - 2pm), Afternoon (2pm - 6pm), Evening (6pm - 10pm), Night (past 10pm)

Consequently, the participants rated eleven conditions for each of the six POI categories. We selected twelve POIs in Munich, Germany as visual examples for the six predefined categories. Figure 5.2 shows an extract of the questionnaire in which participants were asked whether they would visit a certain POI after they have been to a different POI.

Using this methodology, we were able to observe contextual factors that are especially crucial for sequences of POIs and have not yet been observed in related work. For other contextual factors, such as the day of the week, weather, and temperature, which are also relevant for single POIs, we can rely on [35]. Opening hours were also considered. They are a hard criterion, which is why they do not require a preliminary user study. We incorporated all mentioned contextual factors into the context-aware recommendation algorithm.

The aim of this study was to evaluate the influence of the selected contextual factors on the users' decisions to visit a category represented by a selected POI as well as the change of POI popularity precipitated by contextual conditions. In total, we received 324 responses by 27 participants. Participants were recruited via mailing lists and mainly composed of students.

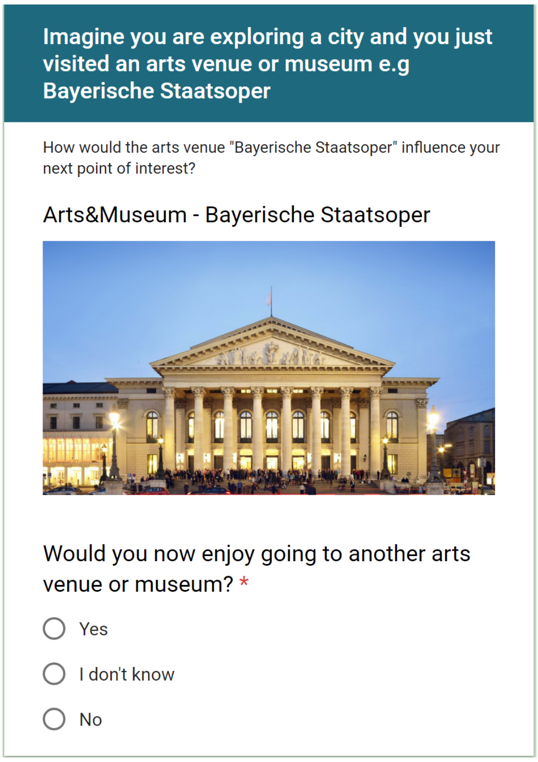


Figure 5.2: Extract from the online questionnaire to acquire context relevance [178].

We calculated the measured relevance U for each contextual factor for all POI categories. The values are listed in Table 5.4. U is normalized to an interval $[0, 1]$, where $U = 0$ means that the contextual factor does not have any influence for this POI category. U is also relevant for the actual context-aware tourist trip recommendation algorithm and is thereby utilized as a weighting factor for the context assessment in Equation 5.2.

In addition to the measured relevance U of a contextual factor, our context-aware approach also depends on ratings for POIs under different contextual conditions. The dataset resulting from the previous conducted questionnaire can also be utilized to determine such a rating. To make the responses quantifiable, *Yes*, *I don't know*, and *No* are mapped to the values 2, 1, and 0. A simple approach would be to use the mathematical expectation value as a rating of a POI category for each contextual condition. However, this does not respect the variation of the rating for a POI when a contextual condition holds or not. Informally speaking, if a POI category is typically very popular except during night, the expectation value would not reflect the real value of the contextual condition *Night*. For example, the expectation value for POIs in the category *Food* is 1.3. However, if we consider only ratings for POIs in the category *Food* under the contextual condition *Night*, the expectation value is 0.749. Hence, we must present a comparison between the average ratings of POIs and ratings of the same items assuming a certain contextual condition holds. We achieve this by dividing the expectation value for a specific contextual condition by the expectation value over all ratings for this POI category. For the category *Food* during night time, the normalized rating is therefore: $\frac{0.749}{1.3} = 0.58$. All computed ratings for POI categories under different contextual conditions are listed in Table 5.5.

Table 5.4: Measured relevance of the contextual factors by POI categories.

	Previously visited POI	Time of the day
Arts & Museums	0.52	0.48
Nightlife	0.26	0.74
Food	0.49	0.51
Outdoors & Recreation	0.33	0.67
Music & Music Event	0.31	0.69
Shopping	0.42	0.58

5.3.2 Incorporating Context into a Tourist Trip Algorithm

Our collected dataset includes two types of data that can be utilized to calculate context-awareness for a tourist trip. First, ratings for categories under different contextual conditions, as displayed in Table 5.5. A rating $r_{TC_1 \dots C_k}$ indicates the evaluation for the POI category T made in the context $C_1, ..., C_k$ and must be in the interval $[0, 2]$ to reflect the aforementioned *Yes*, *I don't know*, and *No* mappings. Second, the relevance $U_{C_1 \dots C_k}$ of each contextual factor $C_1, ..., C_k$, as listed in Table 5.4. As explained in Section 5.3.1, the measured relevance must be in the interval $[0, 1]$.

Given these data, we can calculate a context-awareness factor $\overline{C}$ with a simple weighted arithmetic mean:

$$\overline{C} = \frac{\sum_{i=1}^{k} U_{C_i} r_{TC_i}}{\sum_{i=1}^{k} U_{C_i}} \tag{5.2}$$

Table 5.5: Ratings for POI categories under different contextual conditions.

Contextual Condition	Arts	Nightlife	Food	Outdoors	Music	Shopping
Previously visited POI						
Arts & Museum	1.36	1.16	1.43	1.25	1	0.72
Food	1.4	1.77	0.19	1.28	1.06	1.18
Nightlife	0	1.43	1.04	0.13	1.45	0
Outdoors & Recreation	1.63	0.86	1.37	0.76	1.42	1.52
Music Event	0.04	1.69	1.1	0.6	1.32	0.11
Shopping	0.91	0.79	1.45	0.97	0.52	1.25
Time of the day						
Morning	1.56	0.19	0.3	1.36	0.1	1.82
Midday	1.56	0.07	1.29	1.41	0.19	1.78
Afternoon	1.48	0.15	0.85	1.41	0.68	1.71
Evening	0.64	0.79	1.4	0.76	1.71	0.8
Night	0.42	2	0.58	1.07	1.55	0.11

$\overline{C}$ can now be used to extend the 2D recommender baseline algorithm introduced in Section 5.2.3 by scaling the result of its comparison function:

$$P = r(preferences, number\,of\,places\,per\,category\,in\,path\,so\,far) \times entertainment \times \overline{C}. \tag{5.3}$$

$\overline{C}$ is in the interval $[0, 2]$. A value of $\overline{C} = 0$ zeroes the profit P of a POI while $\overline{C} = 2$ doubles the profit.

In the following, we illustrate the presented methodology with an example comparison considering the two contextual factors *Time of the day* and *Previously visited POI*:

It is 5 pm and the user has just been to a restaurant. The CARS should now predict the user rating for another restaurant. In this scenario, the 2D comparison algorithm would calculate a profit of 4.5 on a scale ranging from 0 to 5. To calculate the context-awareness factor $\overline{C}$, we use the values 0.49 and 0.51 as relevance of the contextual factors from Table 5.4 and the values 0.19 and 0.85 as ratings for the category *Food* in the current contextual condition from Table 5.5. After calculating $\overline{C}$, the 2D profit of 4.5 is downscaled to 2.37:

$$\overline{C} = \frac{0.19 \times 0.49 + 0.85 \times 0.51}{0.49 + 0.51} = 0.526 \tag{5.4}$$

$$P = 4.5 \times 0.526 = 2.37 \tag{5.5}$$

According to Section 2.2, one could assume that this algorithm adheres to *Contextual Post-Filtering*. However, the definition explicitly states that the traditional RS must be executed on the entire dataset first. Since this is not the case, the paradigm *Contextual Modeling* was utilized to incorporate context into the baseline algorithm.

The contextual factors that are considered in the current implementation of the context-aware tourist trip algorithm and their values (contextual conditions) are:

- *Previously visited POI (category)*: Arts & Museum, Food, Music Event, Nightlife, Outdoors & Recreation, Shopping
- *Time of the day*: Morning (8am - 12pm), Midday (12pm - 2pm), Afternoon (2pm - 6pm), Evening (6pm - 10pm), Night (past 10pm)
- *Day of the week*: Working day, Weekend
- *Weather*: Sunny, Cloudy, Clear Sky, Rainy, Snowing
- *Temperature*: Hot, Warm, Cold
- *Opening hours*: Open, Closed

One benefit of the weighted arithmetic mean is the independence of the number of contextual factors. This list can easily be extended. Also the number of contextual factors applied on POIs within a tourist trip can vary. For example, designing contextual factors only known for a specific POI category, e.g., *Nightlife*, is not a concern. On the other hand, one disadvantage resulting from considering multiple contextual factors for $\overline{C}$ is that a supposedly drastic condition, e.g., the POI is closed, can be balanced out by a different condition such as *Sunshine*. We present a solution to this problem in Section 5.5.1.

5.3.3 Evaluation

We developed a web-based application to evaluate our context-aware recommendations (see Section 6.1 for more information about the client application). The RS is based on the TourRec system architecture that we introduced in Chapter 4. Again, POI data was fetched from Foursquare and weather data from OpenWeatherMap. We implemented our algorithm in PHP.

5.3.3.1 Setup

The RQ that we wanted to answer in this user study was whether a context-aware algorithm distinguishing between several contextual conditions can improve tourist trip recommendations generated by a context-unaware algorithm. For this purpose, we conducted an A/B test to measure the effect of the novel approach presented in this section on the user's satisfaction compared to the previously presented Dijkstra-based approach that does not exploit context at all. In this study, only one tourist trip recommendation was displayed to the participants after every request; thereby, the algorithm for the recommendation was randomly chosen. Apart from the recommended trip, participants were not able to distinguish between the algorithms. The recommendation screen displayed relevant contextual information (e.g., the weather) whether or not the context was actually considered.

After every recommendation, a questionnaire was presented to the user. It was composed of the following six statements that reflect the dependent variables of the study. Their purpose was to cover all contextual factors that we investigated in this study. Every statement came with with five possible response options on a scale ranging from 1 (strongly disagree) to 5 (strongly agree).

1. Overall, I am satisfied with the recommended tour.
2. The number of places in my route is well chosen.
3. The selection of different categories in the trip is satisfying.
4. Places are suggested at the right times during the tour.
5. The tour is feasible for a walking tourist.
6. I consider taking this route myself.

We spread the link to the publicly available TourRec application via e-mail and added the questionnaire that the users were asked to complete to the bottom of the application's UI. The participants were mainly composed of students.

5.3.3.2 Results

In total, 15 forms were completed for the baseline algorithm and 9 for the context-aware approach. The conditions were not balanced because of the random selection of an algorithm.

Figure 5.3 illustrates the performance of both algorithms for each of the six statements. Our novel approach for context-aware route recommendations performed somewhat better with regard to the overall satisfaction with the recommended trip and the right number of POIs in the recommendations. In terms of *Feasible Walking Route* and *Consider Taking the Route*, the context-aware algorithm was rated slightly lower than the baseline. However, for these four mentioned statements, the difference was not significant. The biggest difference between the context-aware and context-unaware recommendation algorithms occurred with regard to statements 3 and 4. The selection of categories in the recommended trips was rated better by the participants in the context-aware algorithm. Furthermore, the participants believed that the context-aware algorithm suggests POIs better at the right times of day.

A Mann-Whitney U test showed that the difference for statement 4 is significant at $p < 0.01$. We conclude that our novel approach leads to improved recommendations. However, due to the low number of observations, some of the results were not significant. We recommend conducting a larger user study with more participants in the future to

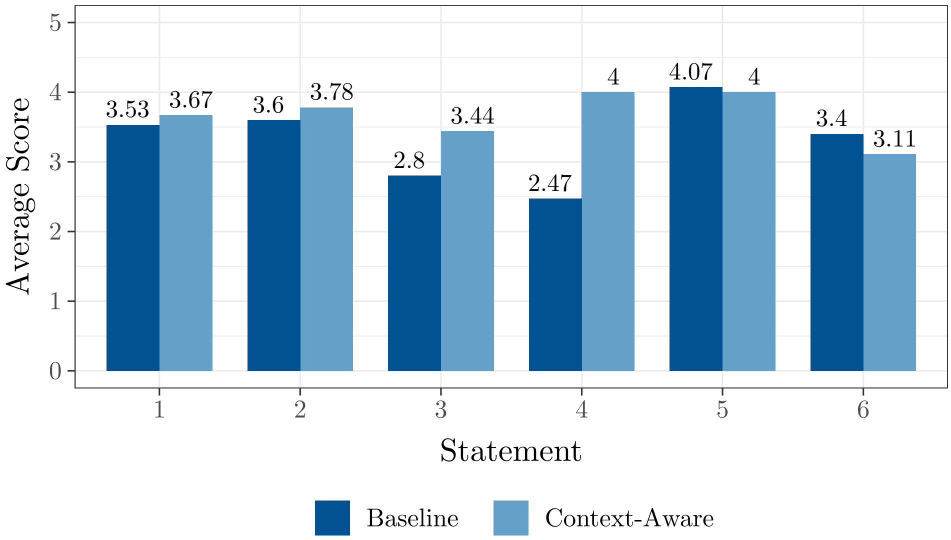


Figure 5.3: Context-aware algorithm user study results (adapted from [178]).

verify these results. Furthermore, our study was conducted with a random sample of students. Repeating the study with participants from different backgrounds could lead to different results as well.

5.4 The Integration of Route Attractiveness Attributes into Tourist Trip Recommender Systems

In the previously presented approaches to solving the TTDP, the total profit of a trip is the sum of the profits of the visited POIs. The profit of a POI is influenced by many factors, such as previously visited POIs. However, we argue that tourists do not always want to take the shortest route between two POIs. Rather, the perceived quality of a tourist trip depends also on the attractiveness of the routes between the POIs. For example, a trip becomes more attractive when the route between two POIs is a relaxing walk in a green area instead of a walk by a loud street, even if this means taking a detour. Furthermore, when too many tourists take the same routes, the recommendations can be adapted to better balance the tourist flows in the city, thereby helping reduce the crowd and pollution in these areas. Consequently, tourist trip RSs should be able to determine the profits of routes by considering all relevant route attractiveness attributes and adapt the recommendations accordingly. In Section 3.1.7, we presented different problems that can be used as underlying models for this variant of the TTDP. They all consider profits for routes; however, they have not been implemented in CARSs to recommend personalized tourist trips.

In this section, we present a solution to integrate route attractiveness attributes into a tourist trip RS for walking tourists. Our approach allows the integration of any data influencing the quality of routes and adjusting the importance of each attribute according to the user's needs. We illustrate our approach by explaining the integration of three attributes as an example. In addition, we present the results of a user study to evaluate the recommended routes.

The content of this section has been published in [179] with some revisions.

5.4.1 Route Attractiveness

The influence of environmental attributes on people's decisions to participate in outdoor activities, such as walking, has been explored earlier. We analyzed previous studies [180], literature reviews [181, 182], developed frameworks [183], and models [184, 185] that dealt with the influence of environmental attributes on the choice of walking routes. Some of these publications also investigated the people's decisions to cycle; however, they all focused on the physical activity of walking. It is not possible to directly compare the results of all works as they used different methodologies or were limited to specific target groups, such as elderly people. Based on our findings, we thus devised a novel, subjective list of route attractiveness attributes that should be considered when evaluating routes in a tourist trip RS. Table 5.6 lists all attributes that we identified and their probability of affecting the route attractiveness. The attributes are classified into three categories: attributes that are likely to affect the route, attributes that are somewhat likely to affect the route, and attributes that are less likely to affect the route or for which the data are insufficient to determine the probability.

All the earlier studies that we reviewed agree that aesthetics attributes are the most important route attractiveness attributes. These attributes include the number of trees along the path, pollution, and cleanliness. Traffic speed is an important safety hazard influencing the attractiveness of a route while the impact of traffic noise is unclear. Other important route attractiveness attributes are the presence of pavements, maintenance of the walking surface, and personal safety in form of surveillance (people around, avoiding empty streets, etc.).

We suggest that smart tourism applications, such as tourist trip RSs, should consider all attributes in the first category and examine whether attributes from the secondary category should be considered before recommending routes to people. Further research is necessary to evaluate the impact of the attributes from the last category, and their impact may also highly depend on the use case.

5.4.2 Edge Weight Calculation of Exemplary Attributes

The previously presented approaches to recommend tourist trips are executed on graphs with the POIs as vertices and the connection between the POIs as edges. The extension that we introduce in this section, however, does not use only the distance between two vertices to determine the edge weight (*cost*). It also considers the presence or absence of the relevant route attractiveness attributes listed in Table 5.6.

In the following, we explain the calculation of the edge weights in tourist trip algorithms taking into account route attractiveness attributes for three examples: trees,

Table 5.6: Route attractiveness attributes for tourist trip RSs. The listed attributes in every category are not ordered. The (+) sign indicates a positive impact on a walking route, and the (-) sign, a negative impact.

Attribute	Impact	Probability
Aesthetics: Trees	(+)	High
Aesthetics: Pollution	(-)	High
Aesthetics: Cleanliness	(+)	High
Permeability: Pavements	(+)	High
Traffic: Speed	(-)	High
Walking Surface: Maintenance	(+)	High
Personal Safety: Surveillance	(+)	High
Aesthetics: Landscaping	(+)	Medium
Permeability: Intersection Distance	(-)	Medium
Traffic: Volume	(-)	Medium
Personal Safety: Lightening	(+)	Medium
Traffic: Crossings	(+)	Medium
Traffic: Crossing Aids	(+)	Medium
Streets: Width	(+)	Medium
Aesthetics: Parks	(+)	Medium
Permeability: Slopes	(-)	Medium
Permeability: Stairs	(-)	Medium
Traffic Control Devices	(+)	Low / Unclear
Walking Surface: Continuity	(+)	Low / Unclear
Traffic: Verge Width	(+)	Low / Unclear
Destination: Shops	(+)	Low / Unclear
Traffic: Noise	(-)	Low / Unclear
Facilities: Places to rest	(+)	Low / Unclear
Environment: Walking Trails	(+)	Low / Unclear
Facilities: Shops	(+)	Low / Unclear
Personal Safety: Blind Walls	(-)	Low / Unclear
Aesthetics: Green strips	(+)	Low / Unclear

pollution, and cleanliness. These are the most relevant attributes for our scenario. We then present the final edge weight calculation taking into account all three attributes. Our goal is to present a flexible and extendable solution for integrating attractiveness attributes. Providers of tourist trip RSs should be able to easily add or replace attributes, depending on the available data sources. Users should be able to individually adjust the importance of each attractiveness attribute while exploring a city.

5.4.2.1 Aesthetics: Trees

The tree edge weight is based on the tree density on a route. The greater the number of trees on the edge, the higher is the tree density and vice versa. A higher tree density corresponds to a lower edge weight in the graph.

In order to weight the different tree densities, we utilize a weighting method described by Giles-Corti et al. [186]. For our purpose, we normalize the weights to a scale from 0 to 5 (Table 5.7).

Table 5.7: Weights for tree density according to [186] and after normalization (exact values in brackets).

Tree Density	Weight	Normalized Weight
Many trees touching	14.3	5
Some trees touching	11.4	4 (3.986)
Trees close but do not touch	8.6	3 (3.007)
Trees spread apart	5.7	2 (1.993)
Sparse trees	2.86	1
No trees	0	0

We use the horizontal spread of the tree when viewed from the top, i.e., the crown spread, to define tree density. For the sake of simplicity, we use a generic crown radius of 5 m for every tree in our algorithm. This is roughly the average crown size of the *Tilia cordata* Mill. species, which is the species of trees most commonly planted in Berlin, Germany (35 %) [187]. We use the following function to estimate tree density:

$$DensityScore(Edge_i) = \frac{EdgeLength(Edge_i)}{NumberOfTrees(Edge_i)}, \tag{5.6}$$

where $Edge_i$ is the ith edge in the graph; $EdgeLength$ is the distance of the edge in meters; and $NumberOfTrees$ is the total number of trees assigned to the edge. Using this equation and assuming a generic crowd radius of 5 m, we estimated tree density as specified in Table 5.8.

The tree edge weight is eventually calculated by dividing the density score by the normalized weight of the respective tree density category in Table 5.8. When there are no trees, the edge length is divided by 0.5.

Table 5.8: Tree density categorization based on density score.

Tree Density	Density Score (x)
Many trees touching	$x < 5$
Some trees touching	$5 \leq x < 15$
Trees close but do not touch	$15 \leq x < 25$
Trees spread apart	$25 \leq x < 35$
Sparse trees	$35 \leq x$
No trees	0

5.4.2.2 Aesthetics: Pollution

Air quality is measured differently by different countries. Pollutants are measured over a certain defined period, and the density of the pollution is usually measured in micrograms per cubic meter. Some of the most common air pollutants are nitrogen dioxide (NO2), ozone (O3), fine particulate matter (PM2.5), and coarse particulate matter (PM10). In order to standardize the concentration values of different pollutants, we use the Common Air Quality Index (CAQI) (Table 5.9) [188]. The CAQI is an index that compares air quality across different European countries. It has been used in the data on the website airqualitynow.eu since 2006.

Table 5.9: CAQI values corresponding to different pollutant concentrations [188].

Index Class	Grid	Pollutant (hourly) density in μ/m^3			
		NO2	PM10	O3	PM2.5 (opt.)
Very high	>100	>400	>180	>240	>110
High	75–100	200–400	90–180	180–240	55–110
Medium	50–75	100–200	50–90	120–180	30–55
Low	25–50	50–100	25–50	60–120	15–30
Very low	0–25	0–50	0–25	0–60	0–15

If two or more pollutants have different CAQI values for a region, the higher value is considered to be the overall CAQI value as the worst grid value determines the overall index class.

We use the CAQI values to map air pollution values to edges. The air pollution weight is calculated by dividing the CAQI value of the edge by 25 and multiplying it with the edge length. For very low air pollution CAQI values, we divide the CAQI value of the edge by 50 to reduce the costs of edges with very low pollution even more strongly.

5.4.2.3 Aesthetics: Cleanliness

There is no standard way of measuring cleanliness as it greatly depends on the public perception of littering. In this work, we measure cleanliness as littering on a scale of 0 to 10. The littering value could be reported by citizens through mobile applications, for example. Values less than 5 are considered as low or medium littering while values greater than or equal to 5 are considered as high littering. The value of 0 represents either no littering or unavailability of data.

The littering weight for high littering is calculated by multiplying the littering value by the edge length. For low littering values, we divide the littering value by 2 before multiplying it by the edge length. For no littering or no data, we multiply the edge length by 0.1.

5.4.2.4 Final Edge Weight

The final edge weight e takes the edge length and the attractiveness attributes into account. Our proposed calculation assigns weights to the attributes:

$$e = (x \times treesWeight) + (y \times airpollutionWeight) + (z \times litterWeight), \tag{5.7}$$

where x, y, and z determine the importance of each attractiveness attribute. They can either be fixed, provided at runtime by the user, calculated from user preferences, or learned through user behavior. The dynamic nature of these values makes the algorithm adjustable, allowing us to meet the requirement of a flexible and extendable solution to integrate attractiveness attributes. For example, if the presence of trees is more important than no littering for a user, x should be greater than z. If the user wants to completely ignore an attribute, the corresponding value is set to 0. Furthermore, additional attributes with the desired weight can be added at any time. In this work, we initially set x, y, and z to 1. We evaluated the importance of each attribute, as described in Section 5.4.4.1.

5.4.3 Implementation

As explained, the calculated final edge weight can be interpreted as the cost for traveling between two POIs in a tourist trip RS. Tourist trip algorithms, such as the previously presented approaches, can be used to incorporate the costs when recommending tourist trips: The RS first determines the profits for POIs that the user could visit on a trip based on the user's interests and contextual factors, such as the weather. Then, a tourist trip algorithm tries to recommend a sequence of POIs along a route. The algorithm uses the calculated costs to determine the exact route between two POIs in the trip.

In order to execute a tourist trip algorithm that incorporates route attractiveness attributes, we need map data to build a weighted graph of vertices and edges that is required in our presented approaches. In the following, we explain the pre-processing of map data and our example attributes, and the mapping of these attributes to the graph.

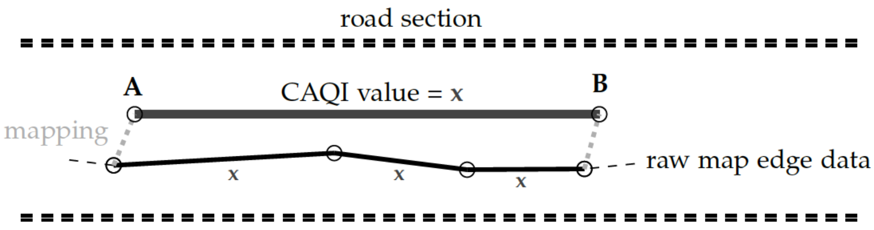


Figure 5.4: Mapping a CAQI vector to edges [179].

For our prototype, we used the open-source vector map data source OpenStreetMap[2] to access the required data. We implemented our approach using only a small extract of OpenStreetMaps since loading all map data for a large city or region and performing route operations on it is a memory-intensive task. In this work, we used an extract of the city center of Munich, Germany.

In OpenStreetMap, ways are essentially a collection of connected nodes. We can consider ways as edges for simplicity of discussion. A node in OpenStreetMap can be a street intersection, a bench, or any other point specific information. Nodes, such as benches, trees, or ways that are only used to define boundaries of a park, for example, are not connected to other nodes, and hence, they do not specify streets or footpaths. We need to extract only those nodes that are surely connected. We use these ways from our OpenStreetMap extract to eventually build a graph. Given the latitude and longitude information of two neighboring nodes, we can calculate the distance between them, and eventually create a weighted graph.

In the next step, we map the attractiveness attributes to the graph edges so that they can be used in the tourist trip algorithm.

Tree data are available in OpenStreetMap files and represented by latitudes and longitudes. We assign every tree to the edges of the nearest node. This approach increases the actual number of trees assigned to the graph; however, this tree count is only used to estimate the tree density for an edge.

Air pollution data are represented by a vector line with a starting point and an end point. Since we do not have access to real pollution data, we used random CAQI values in this work. First, we find two nodes: the nearest node for the starting point and that for the end point of the CAQI vector. Within the length of a single CAQI vector, multiple edges of the graph might be present. The CAQI value needs to be translated to all the edges lying between two points A and B. Hence, we use a shortest path calculation between two nearest nodes to find all such edges. This approach is illustrated in Figure 5.4. Nodes A and B represent the starting and end points of a CAQI vector. They are mapped to the nearest nodes, and the shortest path between these nodes has three edges. If multiple CAQI values are mapped to the same edge, the highest CAQI value is assigned.

[2] `https://www.openstreetmap.org/` (accessed February 16, 2020)

The litter data are also represented as a point; however, the impact is considered to be within a certain radius. Since we do not have access to real littering data, we used random values in this work. First, we find the nearest node for every litter position. Then, we determine all the edges that are within or intersect with a 10 m radius around this nearest node. For each edge, we add the current litter value to the total litter value of this edge. It is important to note that multiple littering spots near the same edge add up, and hence the final littering value of an edge can be greater than 10.

After assigning all the weights to the edges, the final graph can be created and used by a tourist trip RS or any type of routing application to find routes taking into account the integrated attractiveness attributes. We developed a web application incorporating the aforementioned attractiveness attributes to visualize our approach and the routes that can be generated. The applications was developed using ReactJS 16.3.0[3], a JavaScript library for building web applications. Figure 5.5 shows the recommendation made by our application regarding the shortest path between two POIs and different alternatives depending on the considered attribute.

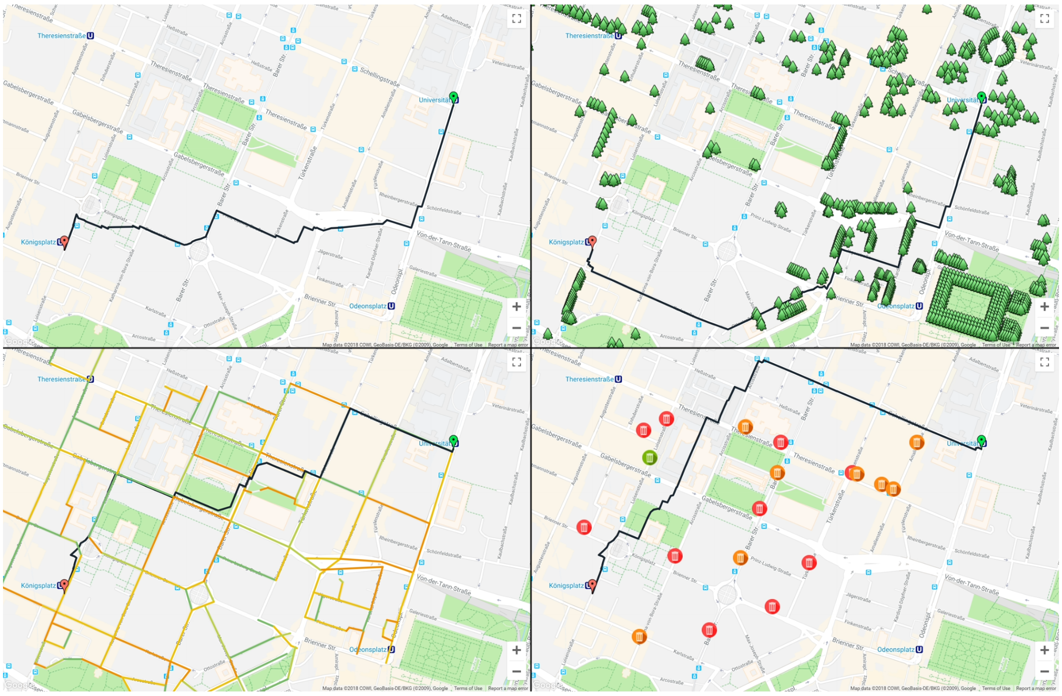


Figure 5.5: Visualization of route attractiveness attributes in our application and the recommended routes between two POIs [179]. Top left: shortest path, top right: trees, bottom left: air pollution (green: very low pollution, yellow: medium pollution, red: very high pollution), and bottom right: cleanliness (green: littering value < 2, orange: littering value < 5, red: littering value ≥ 5).

[3] `https://reactjs.org/` (accessed February 16, 2020)

5.4.4 Evaluation

The main goal of the user study is to determine the impact of route attractiveness attributes on the user's decision of choosing an alternative walking route between two POIs instead of the shortest path. The user study is divided into two parts:

1. Determination of the impact of the three considered attributes on route attractiveness.
2. Evaluation of the recommended routes, including the route attractiveness attributes of the examples.

In all, 16 users participated in the user study. Most of the participants were students from Munich in the age range 22 and 29 years.

5.4.4.1 Attributes Questionnaire

A relevant route attractiveness attribute either has a positive impact or a negative impact on route attractiveness. We asked the participants about their decision to walk a route between two POIs if an attribute is (a) present and (b) absent.

Table 5.10 summarizes the results of our questionnaire. Air pollution is obviously a very critical attribute for travelers. In fact, 100% of our participants claimed that they will avoid routes with at least little air pollution (Table 5.9), while the absence of air pollution is the reason for choosing an alternative route over the shortest path. A similar, less unanimous behavior can be observed when asking participants about littering: 81.25% of the participants mentioned that they will avoid routes that are characterized by visible littering and 87.5% of the participants will choose an alternative route over the shortest path to avoid littering. Greenery, however, seems to be a bonus for travelers. Having no greenery (e.g., no trees) does not influence their decision of choosing an alternative route.

Table 5.10: Influence of the presence or absence of route attractiveness attributes on travel decisions. (Note: highest value marked in bold)

	Travel	Avoid	No influence
No air pollution	**100%**	0%	0%
Little air pollution	0%	**100%**	0%
No greenary	12.5%	18.75%	**68.75%**
Greenary	**93.75%**	0%	6.25%
No littering	**87.5%**	0%	12.5%
Littering	6.25%	**81.25%**	12.5%

We conclude that air pollution and littering are two dominant factors influencing the choice of a route. Hence, they should receive very high weight in a tourist trip

RS. Missing greenery does not necessarily decrease the recommendation probability of a route if the route is clean. However, almost a fifth of travelers prefer routes with greenery when traveling. Greenery can therefore be used to choose between different routes with similar pollution. This also validates our assumption that users should be able to adjust the weights, if necessary.

5.4.4.2 Evaluation of the Recommended Trips

The participants of our user study were then asked to use the application. Each participant had to specify two POIs, a starting and ending point, five times. For each POI pair, the shortest path and a route for each of the four test conditions (tree attribute only, air pollution attribute only, litter attribute only, and all the attributes combined) were calculated. For each test condition, the participants saw the attributes visualized on the application's map, as illustrated in Figure 5.5. Since some of our data were assumed, the participants were asked to assume that all data are real.

After every recommendation, the participants were given a comparison of the time taken in minutes for the detour compared to that for the shortest path and were asked if they would prefer the recommended route over the shortest path.

Table 5.11 summarizes the results. For each test condition, we conducted a binomial test to find out if the integration of route attractiveness attributes had a significant effect on the users' decision. Results show that integrating trees, air pollution, and all the attributes taken together prompt tourists to significantly more often select the recommended route over the shortest path. Less littering, however, does not seem to make people choose the recommended route more often than the shortest path.

Table 5.11: Ratio of users choosing the more attractive route over the shortest path. (Note: $^{*}p < 0.05$; $^{**}p < 0.01$; $^{***}p < 0.001$)

Algorithm	Mean	Sig.
Air pollution only	0.7	***
Trees only	0.76	***
Littering only	0.59	
Combined	0.66	**

5.4.4.3 Discussion

Previous research showed that many environmental factors have a significant influence on the perceived attractiveness of routes. The presence or absence of some of these factors is prompting people to choose routes with the given characteristics even at the cost of taking a detour. For tourists, in particular, route attractiveness plays an important role when planning tourist trips. Many tourists do not want to only visit as many POIs as possible. The walking time between two locations is also part of the pleasure and can increase a traveler's happiness. Hence, attractive routes are often preferred over the shortest path if they promise a more pleasant journey. RSs for tourists should incorporate route attractiveness attributes to better adapt their recommendations to the users' individual needs.

We presented a list of attributes that influence people's decisions on the choice of walking routes. We demonstrated the integration of three such attributes into a tourist trip RS. In our preliminary user study, a significant majority chose recommended routes considering attractiveness attributes over the shortest path.

Route recommendations considering environmental data not only promise improved support of tourists while traveling in cities, but also help cities to become smarter from many perspectives. Recommendations can be made to avoid currently polluted areas until the air quality improves. The congestions of routes and means of transport can be integrated as route attractiveness attributes to support tourists, locals, and commuters, helping them avoid congested areas. Thus, better distribution of travelers and, in turn, a higher satisfaction of all players in a city can be ensured. Furthermore, cities can analyze how people move between POIs and start initiatives to make alternative routes more attractive by adopting measures such as planting more trees and improving pavements along unpopular routes.

Our user study came with some limitations. We had to use random data for air pollution and littering since we did not have access to real data. Furthermore, our sample size was limited to 16 participants from a similar background and in the same age range. Finally, the prototype that we used for the evaluation was not integrated in a fully working tourist trip RS that allowed the users to specify own travel goals and preferred POI categories or adjust the importance of attractiveness attributes. For a better understanding of the influence of route attractiveness and detours on a traveler's satisfaction with a recommended trip, we propose evaluating our approach in a larger user study with real tourists and a fully working application.

5.5 Proposed Extensions

Incorporating context and route attractiveness attributes into tourist trip algorithms improves the quality of the recommended trips; however, our user studies also revealed some shortcomings and potentials for improvement. In the following, we suggest extensions to our algorithms that promise better recommendations and show how to integrate our approach into a different TTDP algorithm. We implemented all of our ideas prototypically. They were preliminarily evaluated in an online evaluation that we conducted using TourRec. The main findings of the online evaluation are summarized in Section 5.6.

5.5.1 Counteracting the Equalization of two or more Extreme Contextual Conditions

As previously mentioned, one disadvantage of our context-aware approach is the possible equalizing of two or more extreme contextual conditions due to the weighted arithmetic mean. This is why we suggest a modified version of our context-aware algorithm that sets an item profit to 0 if one or more context ratings are below a threshold t. We implemented a first version of this extension in which we set the threshold $t = 0.3$. Using this threshold, a restaurant cannot be recommended directly after another restaurant, even when the time is perfect for a *Food* recommendation (e.g., for lunch), as the profit of a *Food* POI when recommended after another *Food* POI is below the threshold, as shown in Table 5.5.

However, it is possible that an algorithm that implements this approach terminates too early and does not use the whole time budget when no suitable POIs without context ratings below the threshold t can be found. We present statistical evidence that supports this assumption in Section 5.6.2.3.

5.5.2 Item Dependencies

Our results and previous experiments [78] showed that consuming recommended items can have a large impact on the profits of other items in the same sequence. However, the contextual factor *Previously visited POI* that we observed in Section 5.3 considers only the influence of one POI on another POI's profit. We argue that consuming an item does not only have an impact on the utility of the subsequent item, but on a sequence of items. For instance, after visiting a restaurant, a user will most likely not be interested in another restaurant within the next few hours, even when doing some other short-time activities in the meantime. Only after a few hours or activities, the user might be interested in going to a restaurant again.

We developed the idea of item dependencies to determine how the presence or absence of an item in a recommendation influences the utility of the subsequent items in the same recommendation. Item dependencies are based on the *Previously visited POI* ratings, as presented in Table 5.5. The underlying concept of item dependencies is that the longer the period between two POIs in a sequence, the lower the impact of the prior POI on the other POI. This means that the item rating in Table 5.5 approaches 1. The concept is similar to the OPSP (see Section 3.1.8.5). The main difference is that in the OPSP, each vertex has a normally distributed random profit and the profit is not revealed before the user arrives at the POI. The idea of item dependencies is that the influence of one POI on another POI is already known before the trip is generated.

Initial values of item dependencies can be taken from Table 5.5. Item dependencies can follow a general pattern (e.g., limiting restaurants in a trip to a reasonable number) but usually differ between users because of personal preferences. This is why a RS should learn which combinations of POIs the user appreciates or rejects. Critiquing can be used to achieve this goal (see Section 2.1.3). For example, users can be presented with two or more alternatives for concrete POI recommendations and indicate their preference for one POI over the other. Other options are suggestions for adding or removing POIs. Users should not be overwhelmed with interactions, this is why implicit feedback should play an important in practical applications. If, for example, a user spends a lot of time at a POI, it is likely that the user is interested in similar POIs.

The following example illustrates the basic idea of item dependencies: A tourist visits a restaurant R_1. Visiting another restaurant R_2 right after R_1 would decrease the profit of R_2 by 81% according to Table 5.5. If the tourist visits other POIs after R_1, visiting

R_2 will become more interesting again. The item dependency $d(R_1, R_2)$ approaches 1 over the number of visited POIs between R_1 and R_2. That is, if the period between two restaurants is long enough, R_1 will have no impact on R_2 anymore.

We use the following formula to describe how an item dependency d evolves over visited POIs:

$$d(POI_1, POI_2) = 1 + (d' - 1) \times e^{-tx}, \tag{5.8}$$

where POI_1 is the prior POI and POI_2 the new POI that is added to the sequence. d' is the initial item dependency value from Table 5.5. It is in the interval $[0, 2]$, as explained in Section 5.3.2. $t > 0$ determines how quickly d approaches 1. x is the number of other POIs between POI_1 and POI_2. For example, $x = 0$ if there are no other POIs between POI_1 and POI_2. Using this formula, $d(POI_1, POI_2)$ will never be exactly 1. Consequently, one can define a threshold (e.g., after 10 other POIs or a certain amount of time) after which a previous POI has no influence on a subsequent POI, that is $d(POI_1, POI_2) = 1$.

The following example of a sequence of three POIs and with $t = 0.2$ illustrates the evolvement of item dependencies:

$$Restaurant\ R_1 \rightarrow Shopping\ Mall\ S_1 \rightarrow Restaurant\ R_2$$

The user already visited R_1 and S_1. Now, the RS is about to add R_2 to the sequence. The item dependencies $d(S_1, R_2)$ and $d(R_1, R_2)$ have to be determined to calculate the profit of R_2:

$$d(S_1, R_2) = 1 + (1.45 - 1) \times e^{-0.2 \times 0} = 1.45 \tag{5.9}$$

$$d(R_1, R_2) = 1 + (0.19 - 1) \times e^{-0.2 \times 1} = 0.34 \tag{5.10}$$

As there is already another POI between the two restaurants, R_1 decreases the profit of R_2 by 66% and not 81%.

If there are, for example, five other POIs between R_1 and R_2, $d(R_1, R_2)$ will change accordingly:

$$d(R_1, R_2) = 1 + (0.19 - 1) \times e^{-0.2 \times 5} = 0.70 \tag{5.11}$$

Figure 5.6 illustrates the curve progressions of item dependencies with factors 0.19 and 1.45. Both curves approach 1. The more POIs are between two restaurants in a trip, the more likely it is that a second restaurant is appreciated by the user and becomes part of the recommended sequence.

Our approach has two main advantages over previous solutions: It does not only update the previously introduced *Previously visited POI* contextual factors, it also makes the usage of PCC (see Section 5.2) to improve the selection of POI categories obsolete. Since item dependencies allow to punish unwanted combinations of POIs in a trip, such as two restaurants in a row, a suitable number of POIs in a category is chosen automatically.

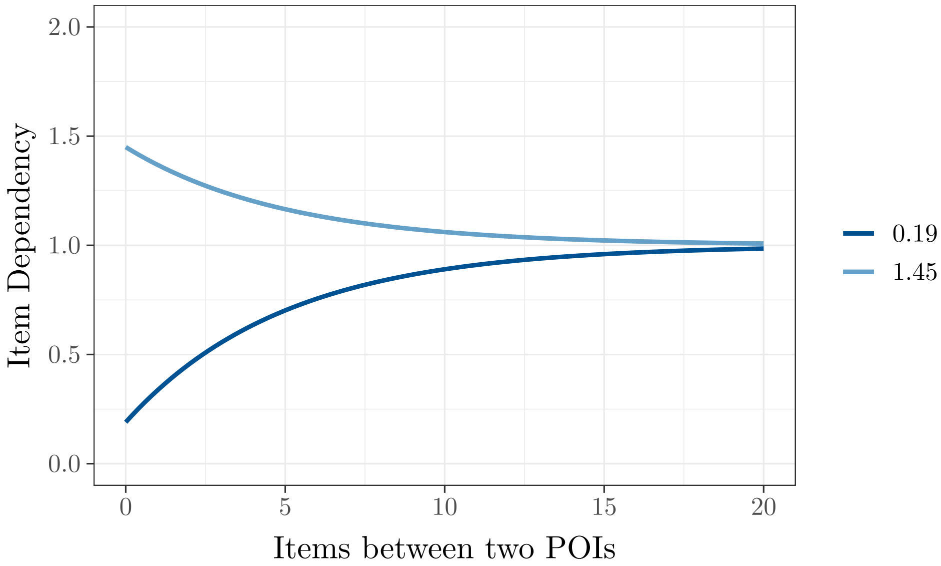


Figure 5.6: Two examples of the development of item dependency d between two items over the number of items between both items with the initial profits $p_1 = 0.19$ and $p_2 = 1.45$. All item dependencies approach 1.

The number of desired categories, such as *Food*, can still be increased by specifying a high user rating for this category.

When a routing algorithm tries to add a POI to a recommendation, the item dependency between the new POI and all previous POIs has to be calculated. The final profit of the new POI when added to the recommendation is the product of all item dependencies and the POI's original profit (which considers the user preferences and other contextual factors).

For the example sequence *Restaurant* R_1 → *Shopping Mall* S_1 → *Restaurant* R_2 the final profit p of R_2 is:

$$p(R_2) = d(S_1, R_2) \times d(R_1, R_2) \times p'(R_2), \tag{5.12}$$

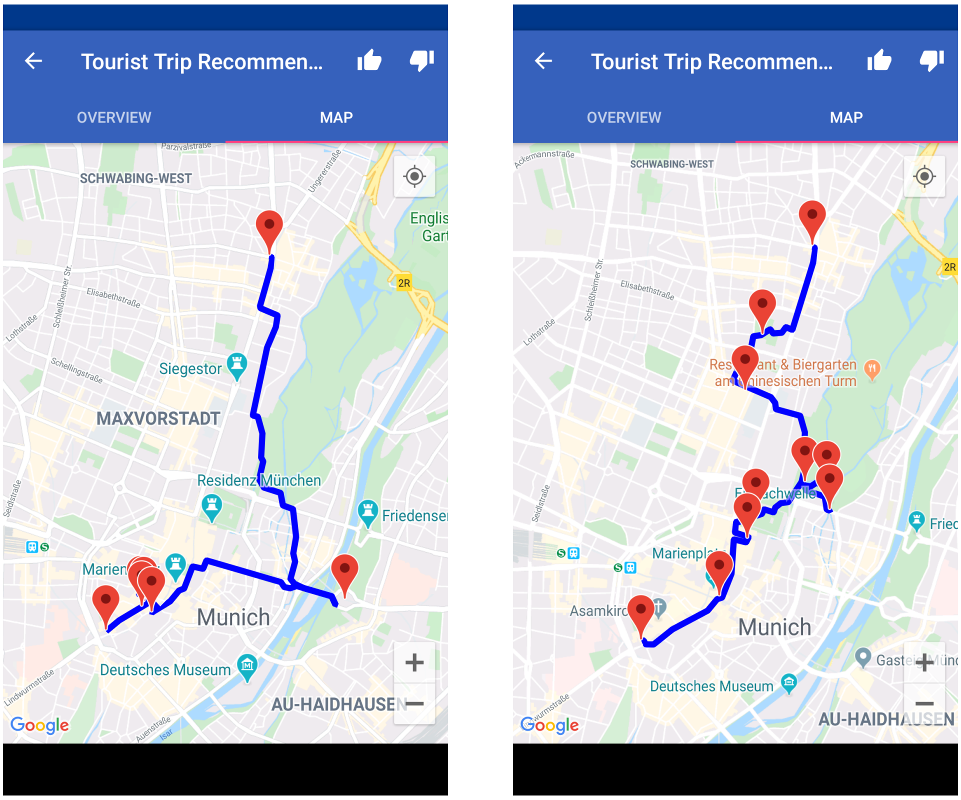
where $p'(R_2)$ is the original profit of the second restaurant. If the original profit of R2 for the user is 4 on a scale ranging from 0 to 5, the profit in this sequence will be updated to:

$$p(R2) = 1.45 \times 0.34 \times 4 = 1.97 \tag{5.13}$$

5.5.3 Minimizing the Distances to the Final Destination

The previously presented Dijkstra-based algorithm tends to recommend tourist trips in which the majority of POIs is concentrated in a small area. This is often the case when the requested route passes a touristic area with many POIs. In this case, the algorithm reaches the specified time limit without recommending POIs close to the destination.

Hence, the connection from the last POI to the final destination can be perceived as relatively large with a disproportionate walking distance (Figure 5.7).



(a) **(b)**

Figure 5.7: Examples of (a) a tourist trip that is concentrated in a touristic area and characterized by a large walking distance to the destination and (b) a tourist trip whose POIs are evenly distributed along the route.

To overcome this problem, we propose considering the distance to the final destination when adding a new location to a path in the Dijkstra-based algorithm. Hence, the extended algorithm tries to maximize *entertainment/(distance + distance to destination)* for every subpath. In Section 5.6.2.3, we present statistical evidence that this approach reduces the probability of a disproportionate walking distance to the final destination.

5.5.4 Integration into other Tourist Trip Algorithms

The context-aware recommendation techniques that we introduced in this thesis can also be integrated into other types of routing algorithms. We suggest the integration into established TTDP algorithms. In this section, we show how to implement GRASP, a metaheuristic that performs a number of independent iterations until it returns the best result found [143]. It has been used by Souffriau et al. [142] to solve the TOP and extended in other works to solve related problems (see Section 3.1). GRASP selects next

POIs randomly to some extent; hence, recommendations can become more serendipitous compared to other tourist trip algorithms. It has been shown that modifications of GRASP, such as GRASP with Path Relinking post-processing (see Section 3.1.2), improve the performance of GRASP but lead to a significantly higher running time [133]. This is why we rely on the original GRASP method in the following as it is more suitable for practical tourist trip applications. We extended this implementation of GRASP by context-aware recommendations and item dependencies and used it in the online evaluation of TourRec in Section 5.6.

The following description of a GRASP algorithm for tourist trip recommendations is based on [142]: A GRASP algorithm performs a number of iterations until a stopping criterion is met. For each iteration, a parameter between 0 and 1 is randomly chosen, prescribing a ratio between greediness and randomness. Next, a candidate list with all possible insertions is created. For each POI in the insertion list, a heuristic value is calculated. A threshold between the minimum and maximum heuristic values of the candidate list and based on the greediness parameter is calculated. Only POIs with a heuristic value exceeding the threshold are taken into account for insertion in this round. For this purpose, one POI of the restricted candidate list is randomly picked. At the end of each iteration, a path containing multiple POIs is created and another iteration with a different greediness parameter is started. In the end, the best iteration result is returned.

In our scenario, the heuristic value of a POI j is calculated by dividing the profit p of the POI by the distance from the previous POI i:

$$h_j = \frac{p_j}{dist(i,j)} \tag{5.14}$$

The threshold is computed by multiplying the greediness parameter g with the difference between the maximum and minimum heuristic values of the candidate list and adding the product to the minimum heuristic value:

$$t = h_{min} + g \times (h_{max} - h_{min}) \tag{5.15}$$

Calculating the threshold ensures that only POIs with a sufficient high profit per distance are taken into account for the recommended trip despite the random selection of a POI.

5.6 Insights from an Online Evaluation of TourRec

We conducted an online evaluation to test our proposed extensions with real users. For this purpose, we integrated the extensions into different variants of the previously presented Dijkstra-based algorithm. In addition, we implemented a variant of GRASP, as explained in Section 5.5.4. The online evaluation was conducted using an updated version of the TourRec application that is publicly available on the Google Play Store (see Section 6.2).

5.6.1 Setup

We used TourRec's A/B testing feature to compare the context-aware algorithm that we introduced in Section 5.3 (Baseline) to four more variants that implemented the suggested extensions that we presented in Section 5.5[4]:

- Dijkstra: The context-aware algorithm extended by item dependencies.
- Dijkstra-based with Distance to Destination (DBDD): The context-aware algorithm extended by item dependencies which tries to minimize the distance to the destination.
- Dijkstra Plus: The context-aware algorithm extended by item dependencies and with counteracting the equalization of contextual conditions.
- GRASP: The proposed GRASP algorithm extended by item dependencies.

TourRec randomly chose one of these algorithms for every trip request. The users were not able to see which algorithm was selected. The goal of this online evaluation was not to collect individual feedback for each of the presented extensions. Instead, we wanted to learn the users' travel preferences when using a tourist trip RS in a realistic scenario and understand the impact of our proposed extension on the generated recommendations. We suggest conducting additional user studies to evaluate the user satisfaction with each of the presented algorithms in future work.

The online evaluation was conducted from May 23, 2018 until July 6th, 2019. We promoted the mobile application via various Facebook groups and mailing lists. However, we had no impact on who downloaded and used the application. The application was available for download in 52 countries. In total, 373 trip requests were made by 135 real users. Note that if a user deleted the application and re-installed it or used different devices to generate tourist trips, a new user ID was assigned to the user. Hence, duplicate users were possible in this dataset.

5.6.2 Results

We used the collected dataset to analyze the travel preferences of the TourRec users, the requests that they formulated, and the recommendations generated by the five variants of the tourist trip algorithm.

5.6.2.1 Travel Preferences

Users were able to rate POI categories on a scale ranging from 0 to 5. With the start of the online evaluation, we introduced subcategories to allow users to specify their travel preferences more precisely. The user rating for a subcategory overwrites the rating of the corresponding main category. For instance, users can rate all Food POIs with a 3 but rate French Restaurants with a 5 if they like restaurants in general but prefer going to a

[4]We published the results of a preliminary user study to evaluate some of these algorithms in [189].

French restaurant. Every trip request contained the full set of rated user preferences, i.e., ratings for the 64 subcategories that are listed in Appendix B. The mobile application allowed the users to generate multiple trips with the same travel preferences or update the preferences whenever desired. In the following, we explain how to determine the popularity of each of the 64 POI categories based on all user requests that we collected.

Not every user requested the same number of trips. However, since every user should have the same weight when identifying the most popular categories, we firstly have to calculate each user's average preferences for every subcategory. For instance, if a user requested two trips and rated *French Restaurant* once with 1, once with 5, the average user preference for *French Restaurant* is 3. Furthermore, when calculating the average user preferences, we have to remove sets of preferences in which all categories were rated with the default value of 3. In this case, these users did not specify their personal travel preferences before requesting a recommendation. This approach resulted in 135 sets of user preferences. The average values of these user preferences are illustrated in Figure 5.8. *Italian Restaurant* received the highest ratings while *Shopping Mall* received the lowest ratings on average.

Table 5.12 shows the average rating per main category. The values are very similar to the results obtained by Wörndl and Hefele (see Section 5.2.4). *Arts & Entertainment*, *Food*, and *Outdoors & Recreation* are the most popular categories while *Shopping* received the lowest ratings on average.

Table 5.12: Average user preferences per main category in the online evaluation.

Category	Mean
Arts & Entertainment	3.20
Nightlife	1.99
Food	3.44
Outdoors & Recreation	3.38
Shopping	1.79

5.6.2.2 Route Requests

Figure 5.9 illustrates the locations of all 373 trip requests. The majority of the requests were made for trips in Germany. In total, trips were requested in 31 countries all over the world. Table 5.13 lists the ten most frequent countries for tourist trip requests.

The average time budget specified by the users was 7.36 h ($s = 1.87$). 69.7% of the requests kept the default value of 8 h. When removing all requests with the default maximum duration of 8 h, the average time budget was 5.9 h ($s = 2.92$).

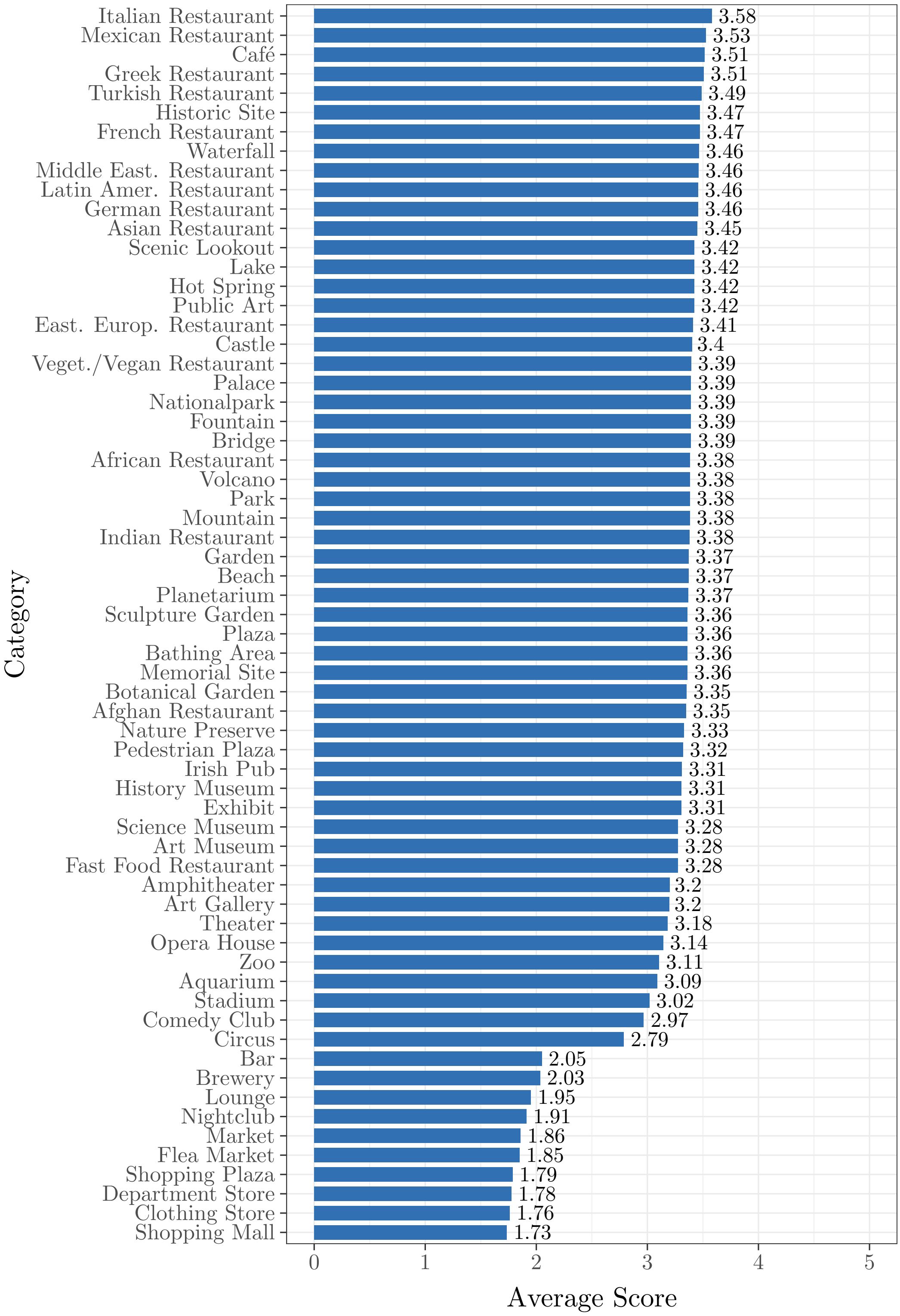


Figure 5.8: Average user ratings for TourRec's POI categories ordered by popularity.

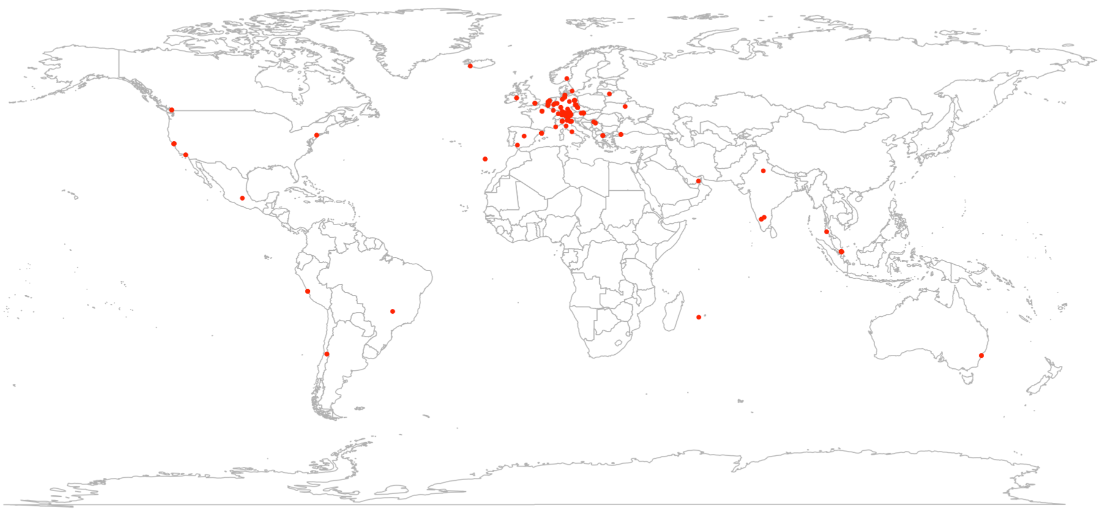


Figure 5.9: Locations of requested tourist trips in the online evaluation.

5.6.2.3 Recommendations

Table 5.14 shows that the actual average duration of a trip was 6.3 h ($s = 2.23$). A significant number of the recommended trips took between 7 and 8 h which is a result of the high ratio of users that did not change the default time budget of 8 h. Compared to the average maximum duration specified by the users, the duration of recommended trips deviated by 1.06 h on average. Table 5.14 reveals that especially trips generated by Dijkstra Plus are on average shorter than the user's time budget. On the other hand, GRASP deviates only by 34 min on average.

We conducted Shapiro-Wilk tests to test if the trip duration deviations are normally distributed and rejected the null hypothesis ($p < 0.001$); hence, there is evidence that the data tested are not normally distributed. The result of a Kruskal–Wallis test confirms that there is a significant difference between the average trip duration deviations ($p < 0.001$).

Dunn's post-hoc tests [190] reveal that the average duration of trips generated by Dijkstra Plus deviates significantly more from the user request than the duration of trips generated by Baseline ($p < 0.001$), GRASP ($p < 0.001$), Dijkstra ($p = 0.004$) and DBDD ($p = 0.005$). Dijkstra Plus is more selective than the other algorithms when POIs do not fully fit the contextual conditions which makes a recommendation impossible.

On average, the recommendations contained 4.8 POIs between start and destination. GRASP, which was able to generate the trips with the lowest unused time, contained most POIs on average. However, a Kruskal–Wallis test does not confirm that there are significant differences between the numbers of POIs. 18 of the 373 route recommendations contained no POIs but only the direct route between starting point and destination. This could happen when the dataset contains no POIs in the recommendation area or

Table 5.13: The ten most frequent countries for tourist trip requests in TourRec.

Rank	Country	Requested trips
1	Germany	202
2	Italy	20
3	Austria	18
4	Spain	17
4	France	17
6	Singapore	14
7	United States	9
7	Netherlands	9
7	Belgium	9
10	Switzerland	8

Table 5.14: Tourist trip characteristics by algorithm.

Algorithm	n	Duration (Avg.)	Deviation (Avg.)	POIs (Avg.)	No POIs
Baseline	82	6.57 h	0.77 h	5.05	3
Dijkstra	80	6.08 h	1.05 h	4.78	4
DBDD	60	6.23 h	0.92 h	4.70	4
Dijkstra Plus	75	5.75 h	2.01 h	4.24	7
GRASP	76	6.85 h	0.57 h	5.17	0
Total	373	6.30 h	1.06 h	4.80	18

when the selected algorithm cannot fulfill the user request (e.g., Dijkstra Plus cannot recommend any POIs when the user is looking for shopping recommendations during the night). Dijkstra Plus recommended 7 trips without any POIs while GRASP was the only algorithm that was always able to include at least one POI into the recommended trip. A recommend trip contained 1.25 POIs in the category *Food* on average. The result of a Kruskal–Wallis test confirms that trips generated by Dijkstra (0.98) and DBDD (0.97) contained significantly less POIs in the category *Food* than trips recommended by the Baseline algorithm (1.76). This proves that the concept of item dependencies reduces the number of POIs in the category *Food* in an average trip of 6.3 h to a more reasonable value.

We conducted additional tests to verify whether DBDD can distribute POIs more evenly along the route. The goal of this approach was to avoid disproportionate walking distances to the final destinations. For this purpose, we calculated how long the walking time from the last POI to the final destination is compared to the overall walking time of

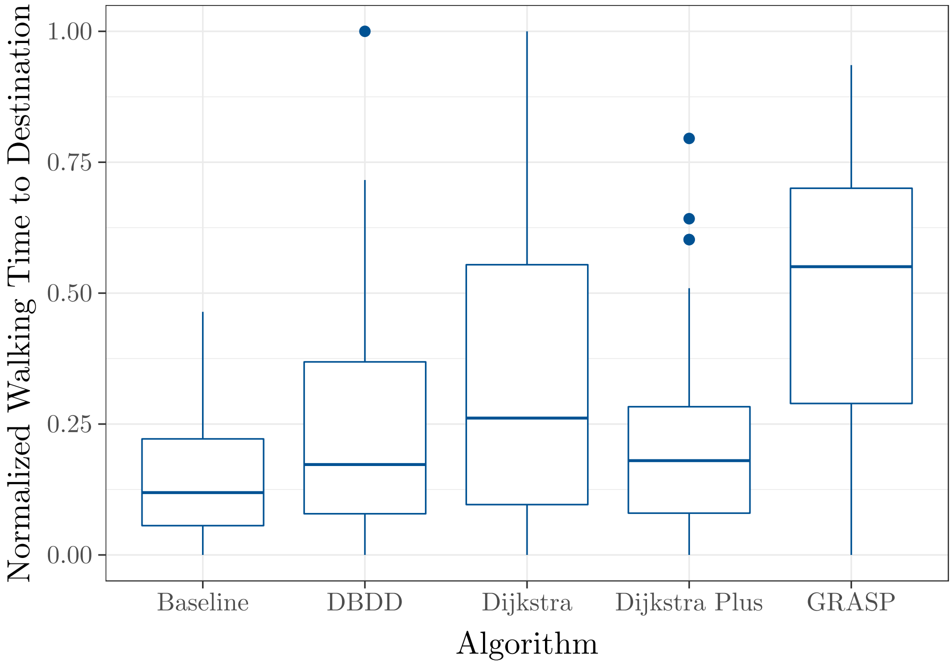


Figure 5.10: Boxplots visualizing the distribution of the normalized walking times to destination for all algorithms.

the trip for every recommended trip with at least one POI. We call this ratio *normalized walking time to destination*. If, for instance, a user walks for two hours during the whole trip including one hour to arrive at the final destination, the normalized walking time to destination is 0.5. Boxplots for all algorithms are illustrated in Figure 5.10.

We conducted statistical tests to verify whether the normalized walking time to destination differs between algorithms. Shapiro-Wilk tests show that there is evidence that the data are not normally distributed ($p < 0.001$). The result of a Kruskal–Wallis test confirms that there is a significant difference between the normalized walking times to destination ($p < 0.001$). Dunn's post-hoc tests show that:

- GRASP recommends trips with a significantly higher normalized walking time to destination than all the other strategies ($p < 0.001$),
- Dijkstra recommends trips with a significantly higher normalized walking time to destination than Baseline ($p < 0.001$) and Dijsktra Plus ($p = 0.015$)
- DBDD recommends trips with a significantly lower normalized walking time to destination than Dijkstra ($p = 0.04$), but only after removing the outlier (Figure 5.10). The outlier is a trip that contained only one POI that was located next to the starting point. However, not enough time was left to visit any other POI. Hence, the normalized walking time to destination was 1.

The results indicate that the introduction of item dependencies increased the likelihood of trips with a relatively long walking time to destination. The idea of the DBDD algorithm as well as counteracting the equalization of contextual conditions turned out to be solutions to this problem. Our GRASP implementation did not integrate any of these two extensions which is why it generated trips with a high normalized walking time to destination. Furthermore, the randomness factor of GRASP can lead to longer walking times to destination.

5.6.3 Discussion

The online evaluation confirmed the findings presented in Section 5.2.4.2: users of a tourist trip RS are mainly interested in going to restaurants, visiting arts & entertainment venues, and doing outdoor activities while traveling. It is important to keep in mind that the popularity of POI categories strongly depends on many factors, such as the user's travel goals. A tourist looking for a relaxing beach vacation will rate POI categories most likely differently than when planning a cultural trip to Paris, for example.

The results of the online evaluation revealed that the extensions suggested in Section 5.5 promise to improve tourist trip recommendations from a user's perspective. For instance, we introduced the concept of item dependencies to avoid unwanted combinations of POIs, such as too many restaurants, in a trip. We showed that the number of restaurants decreases significantly to around one restaurant during a half-day tourist trip. Additional user studies are required to evaluate how the users perceive this result and other effects of item dependencies.

It has been shown that our approaches to solving the TTDP for walking tourists can lead to a disproportionate walking distance to the final destination of the trip. Keeping in mind the final distance to the destination during the execution of our Dijkstra-based tourist trip algorithm promises to distribute POIs more evenly along the route. We have empirical evidence that our proposed approach reduces the walking time from the last POI to the final destination compared to the overall walking time.

Another problem of our tourist trip algorithm that we identified in Section 5.3.3 is the possible equalizing of two or more extreme contextual conditions. We developed an extension of the algorithm that sets the profit of a POI to 0 if one or more context ratings are below a threshold. However, this extension can make it more difficult to find suitable trips and enough POIs that can be combined along a route. Hence, the recommended trips are significantly shorter than the user's time budget which may lead to unsatisfying results.

The online evaluation delivered insights into user preferences and expectations when interacting with tourist trip recommendations. It also showed how several extensions to tourist trip algorithms can improve the quality of recommended trips. Furthermore, we showed how to integrate these extensions into an algorithm from published literature. However, we did not collect quantitative data to evaluate the quality of the recommended trips. Therefore, future work should conduct additional user studies to evaluate how satisfied real users are with recommendations made by each algorithm.

5.7 Summary

In this chapter, we tackled the problem of recommending tourist trips to individuals. More concretely, we wanted to present extensions to existing TTDP solutions to improve the quality of tourist trip recommendations.

Generally speaking, the problem of finding a tourist trip can be formulated as a graph problem. In an undirected graph, all POIs that can be visited are the vertices and the connection between the POIs are the edges. The distance between two POIs is denoted by the edge weight.

We firstly presented a solution to the TTDP that is based on Dijkstra's algorithm. This solution is composed of two steps: (i) retrieving and scoring POIs, and (ii) grouping the POIs to a composite trip. External services and APIs can be used to gather POIs of travel-related categories. The profit of a POI determines the value of the POI for the user. It can take into account the user's travel preferences and the number of votes provided by other users in a LBSN.

Wörndl and Hefele [176] presented a constraint-free and a constraint-based variant of their improved approach of the Dijkstra-based tourist trip algorithm. The algorithm calculates the PCC between the user preferences for categories and the number of POIs per category in the trip. The idea is that the number of POIs in a category in a trip should be appropriate in relation to other categories and the user's preferences. The constraint-based algorithm takes into account time and budget constraints for the trip. In addition, it adjusts the suggested durations of stay based on the user's preferences for categories. The authors compared both approaches to a preliminary solution in a user study. The total amount of POIs and the length of the trips was equally well rated for both approaches. Trips generated by the improved approach matched better the users' preferences. Slightly more users would actually make the recommended trip and the users were more satisfied with the overall result when the improved approach was used. Overall, more users preferred trips made by the improved approach compared to the preliminary solution. We used the constraint-based variant of the improved algorithm as input for the following extensions.

CARSs can increase the quality of recommendations compared to traditional 2D RSs. A lot of research has been done to identify contextual factors relevant for tourism recommendations. Every contextual factor has a different relevance for different POI categories and a different impact on the predicted rating of an item. We integrated some of the most important contextual factors for POI recommendations into the Dijkstra-based tourist trip algorithm including two that are especially relevant for POI sequences: *Time of the day* and *Previously visited POI*. An A/B test confirmed that the context-aware variant of the Dijkstra-based algorithm outperforms the previous algorithm which does not consider context. The participants were overall slightly more satisfied with the recommended trips and the POIs are recommended at significantly more suitable times during the trip. The results also showed that two or more extreme contextual conditions could be equalized in the proposed approach. This is why we suggest that items should be discarded if one or more context ratings are below a threshold t. Furthermore, instead of only considering the previously visited POI, a tourist trip RS should consider all the previously visited POIs as we believe that the absence or presence of POI can have a significant impact on other POIs in the same trip. We call this influence item dependencies. Item dependencies are in the interval $[0, 2]$. Hence, the presence or absence of an item can increase or decrease another item's profit by up to 100%. Item dependencies approach 1 when more items are recommended between the two items.

In many scenarios, the quality of a trip is not only determined by the sum of the profits of the recommended POIs. Moreover, the attractiveness of the routes between the POIs can have a high impact on the user's satisfaction. A wide range of route attractiveness attributes has an impact on people's decisions to participate in outdoor activities. Some of them, such as tree density, pollution, and littering, have a high probability to affect the quality of a walking route and should definitely be considered in tourist trip RSs. This is done by updating the final edge weight in the POI graph. In addition, attribute weights can be assigned to determine the importance of each attractiveness attribute. We implemented this approach using OpenStreetMap data and developed a web application to evaluate the impact of the attributes and the recommended routes. The results showed that having no air pollution and littering motivates people to choose an alternative route over the shortest path. Greenery, however, seems to be a bonus for travelers and missing trees does not influence their decision of choosing an alternative route. The evaluation of four test conditions for recommending routes showed that integrating trees, air pollution, and all the attributes taken together prompt tourists to significantly more often select the recommended route over the shortest path. Less littering, however, does not make people choose the recommendation more often than the shortest path between two locations.

Besides item dependencies and counteracting the equalization of two or more extreme contextual conditions, we also proposed an extension of our context-aware tourist trip algorithm that minimizes the distance to the final destination when recommending a tourist trip. Furthermore, we showed how to integrate our context-aware recommendation techniques into GRASP, an established algorithm from published literature. We conducted an online evaluation over a period of more than one year to test all of these extensions with real users. For this purpose, we implemented our proposed extensions into different variants of the Dijkstra-based, context-aware algorithm and GRASP. All algorithms were integrated in the publicly available TourRec Android application which used TourRec's A/B testing feature to randomly choose one algorithm for every request. Overall, 373 tourist trips were requested during the evaluation period. The results of the online evaluation showed that real users downloaded the application to generate tourist trip recommendations in cities all over the world. They were especially looking for trips containing POIs in the categories *Arts & Entertainment*, *Food*, and *Outdoors & Recreation*. An average trip contained 4.8 POIs and had a duration of 6.3 h. The concept of item dependencies allows travelers to avoid unwanted combinations of POIs within a trip, such as too many restaurants. When keeping in mind the distance to the final destination, our tourist trip algorithm was able to ensure a better distribution of POIs along the walking route, that is, it reduced the probability of a disproportionate walking time to the final destination. Counteracting the equalization of two or more extreme contextual conditions can avoid inappropriate recommendations; however, the trip length in our approach deviated significantly from the specified time budget. We recommend larger user studies in future work to collect additional quantitative data about the quality of the recommended trips.

6 Platforms and User Interfaces for Tourist Trip Recommender Systems

The success of a RS does not only depend on recommendation accuracy, it should also provide a positive UX and be a pleasure to use (see Section 2.6). RSs can run on different platforms and users interact with these platforms and its applications via UIs; hence, the choice of a platform and the design of UIs is a critical task when developing tourist trip RSs.

In this chapter, we show how the TTDP can be solved on different types of platform-UI configurations, as introduced in Section 2.5. While web-based applications enable a user-friendly travel planning, mobile applications are more suitable for receiving recommendations while already traveling. We developed prototypes of the TourRec application for both variants and evaluated the usability of our solutions in user studies. In addition, we introduced two additional platform-UI configurations that allow travelers to receive recommendations while already traveling and in public spaces: a public display application and a distributed approach that combines a mobile application and a public display. We conducted an additional user study to compare the usability and UX of the mobile application to these two variants. The results of this chapter allow us to answer our second RQ: *"Which platforms and UIs support tourists the best in solving the TTDP in realistic scenarios with regard to different usability and UX criteria?"*

The content of this section has been published in [173, 178, 191, 192] with some revisions.

6.1 Web Application for Context-Aware Tourist Trip Recommendations

The first TourRec application that we developed is a web-based prototype. This application was used to compare our context-aware, Dijkstra-based tourist trip algorithm to the context-unaware variant in an A/B test in Section 5.3.3.

6.1.1 Description

The web application is structured into three segments: search, recommendation, and feedback. In the search segment, users can enter their preferences for all six POI categories, the starting point and destination, as well as the time frame for the trip, which is translated to the user's time budget. (Figure 6.1). The input is validated on both client side as well as server side.

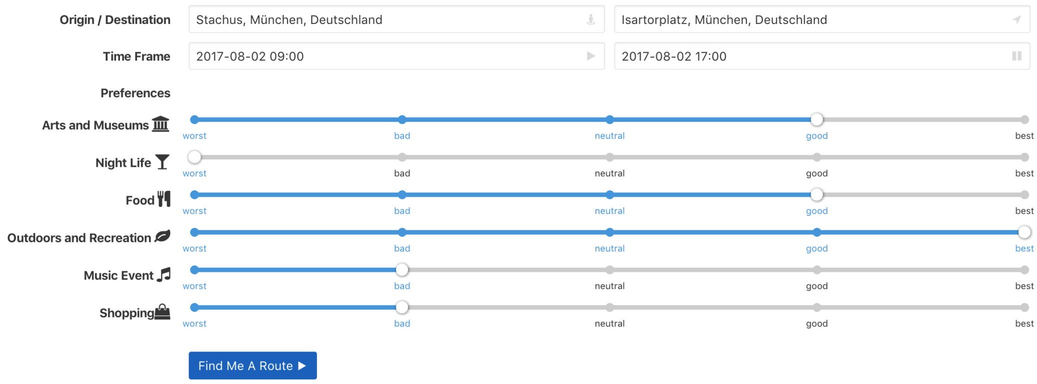


Figure 6.1: Preference elicitation in the context-aware, web-based RS [178].

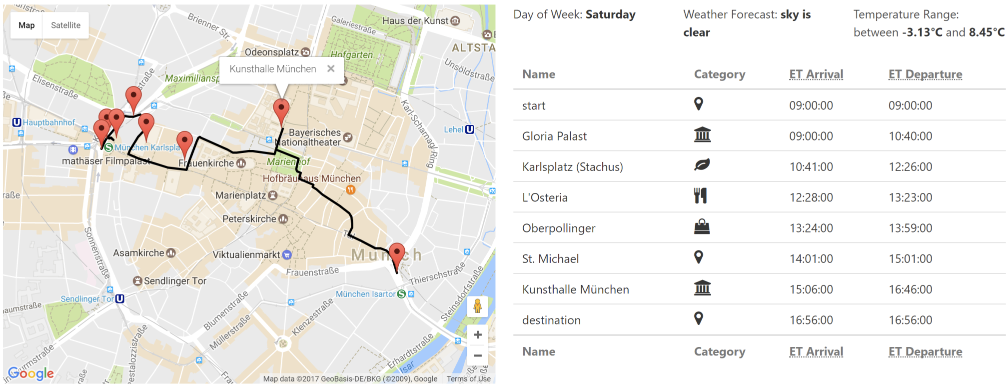


Figure 6.2: Context-aware recommendation in the context-aware, web-based RS [178].

The recommendation segment is structured as follows (Figure 6.2): A map with the suggested POIs and a rendered walking path on it is located on the left-hand side. Note that in this prototype, only one recommendation per request gets displayed. This allowed us to conduct an A/B test in which the users were not aware of the chosen algorithm. On the top right-hand side, contextual information that has been acquired by the system and is relevant for the user for this situation, such as the weather forecast, is displayed. Finally, an ordered list of POIs and their estimated arrival and departure times, respectively, can be found below the contextual information.

The feedback segment is located at the bottom of the prototype. It provided the user with a short introduction into the feedback process. In essence, it is a simple table with multiple statements which can each be answered with radio buttons on a five-point Likert scale ranging from strongly disagree to strongly agree (see Section 5.3.3.1).

The web application was built with help of the JavaScript framework Vue.js 2.2.0[1] and the CSS framework Bulma 0.4.0[2].

[1] https://vuejs.org/ (accessed February 16, 2020)
[2] http://bulma.io/ (accessed February 16, 2020)

6.1.2 Evaluation

We asked the users to complete the SUS questionnaire (see Section 2.6.3) after every tourist trip recommendation to evaluate the usability of the context-aware prototype. The questionnaire was part of the application's feedback segment, similar to the previously presented questionnaire for evaluating the quality of the recommended trips (see Section 5.3.3). The link to the application and the questionnaire were spread via e-mail.

19 participants completed the SUS questionnaire after using the prototype. The participants were mainly composed of students. The average SUS score was 84.17. This score approximately converts to a percentile rank of 94% [122]. This means that the web-based, context-aware TourRec prototype performs better than about 94% of systems tested in terms of perceived usability. However, since our application was accessible from virtually any device, the actual system usability could vary for different screen sizes, operating systems, and browser vendors.

6.2 Android Application

As explained, web-based tourist trip applications allow a user-friendly travel planning, but they are less suitable for receiving recommendations while already traveling. This motivated us to develop a mobile TourRec application for Android devices. The initial version of the mobile application has been introduced in [173] and extended over the course of the last years. The current version of TourRec is available for download in the Google Play Store[3]

6.2.1 Description

The biggest difference to the web-based application is the smaller screen size. In addition, it is optimized for devices with touch screens and supports gestures to zoom in and out on the map that displays the recommended trip, for example. The user's general movement through the mobile application is similar to the web-based application. The starting page of the mobile application allows users to requests a new tourist trip recommendation by specifying the starting point (e.g., the current location), a destination, the starting time and the maximum duration of the trip (Figure 6.3a). If desired, users have the option to specify travel preferences by rating the POI categories on a scale from 0 to 5, as in our other prototypes (Figure 6.3b). The current version of the TourRec Android application allows users to rate subcategories to specify travel preferences more precisely, as introduced in Section 5.6.2.1 (Figure 6.3c).

Recommendations are displayed on a map (Figure 6.3d) or as a list of POIs with additional information, such as predicted arrival times and suggested durations of stay (Figure 6.3e). The list view also summarizes relevant trip data, such as the overall duration and the weather forecast. Clicking on a POI in the list shows all relevant information about the POI on the Foursquare website.

[3] `https://play.google.com/store/apps/details?id=de.tum.in.cm.tourrec` (accessed February 16, 2020)

Users can rate recommendations on a five-star scale (Figure 6.3e). In addition, they can reject a recommendation by clicking the *Thumbs Down* icon on the top right corner of the recommendation interface. After clicking the icon, a pop-up window appears that allows the users to specify the reason for rejecting the recommendation (Figure 6.3f). User can select one reason or decide to not specify any. The rejection and the selected critique are stored in TourRec's data tier.

Note that all UIs were designed using Material Design, a design language introduced by Google[4].

6.2.2 Evaluation

We conducted a usability test to evaluate the initial mobile application in [173] which did not contain subcategories and the critiques in Figure 6.3f. The average SUS score among 39 participants was 84.64. The score converts approximately to a percentile rank of 95%, meaning that the application performs better than about 95% of tested systems in terms of perceived usability. In addition, we separated the participants into two study groups based mobile operating system the user usually uses (iOS and Android). Our results showed that Android users were slightly more satisfied than iOS users (SUS score of 88.06 vs. 81.25).

6.3 Public Display Application

The mobile prototype was the first TourRec application that we developed to receive recommendations while already traveling. In this section, we propose two additional platform-UI configurations for this purpose: a public display variant and a DUI approach that combines a mobile application with a public display. We present a user study to compare all three variants in Section 6.5.

A tourist trip RS on public displays comes with some advantages compared to a mobile application: Users do not need their own devices with internet connection while traveling. Larger displays can facilitate orientation in an unknown area and support the selection of a suitable recommendation when all relevant data, such as POI information, a map, and context data, are displayed on a single UI. Furthermore, a public display can facilitate decision making when used by a group because the recommendation can be viewed by all members of the group. More advanced approaches allow users to modify the recommendation directly on the public display and send it to their personal devices.

These advantages represent our motivation to integrate public displays into our TourRec application. For this purpose, we adapted the mobile TourRec application to public displays and compared it to the smartphone application and a DUI approach in a user study (Section 6.5). We tried to keep the changes to the smartphone's UIs to a minimum so that the only independent variable tested in our user study was the interaction type rather than other changes in the layout. Thus, the public display application applies the same layout but attempts to benefit from the larger display area

[4] `https://material.io/guidelines/` (accessed February 16, 2020)

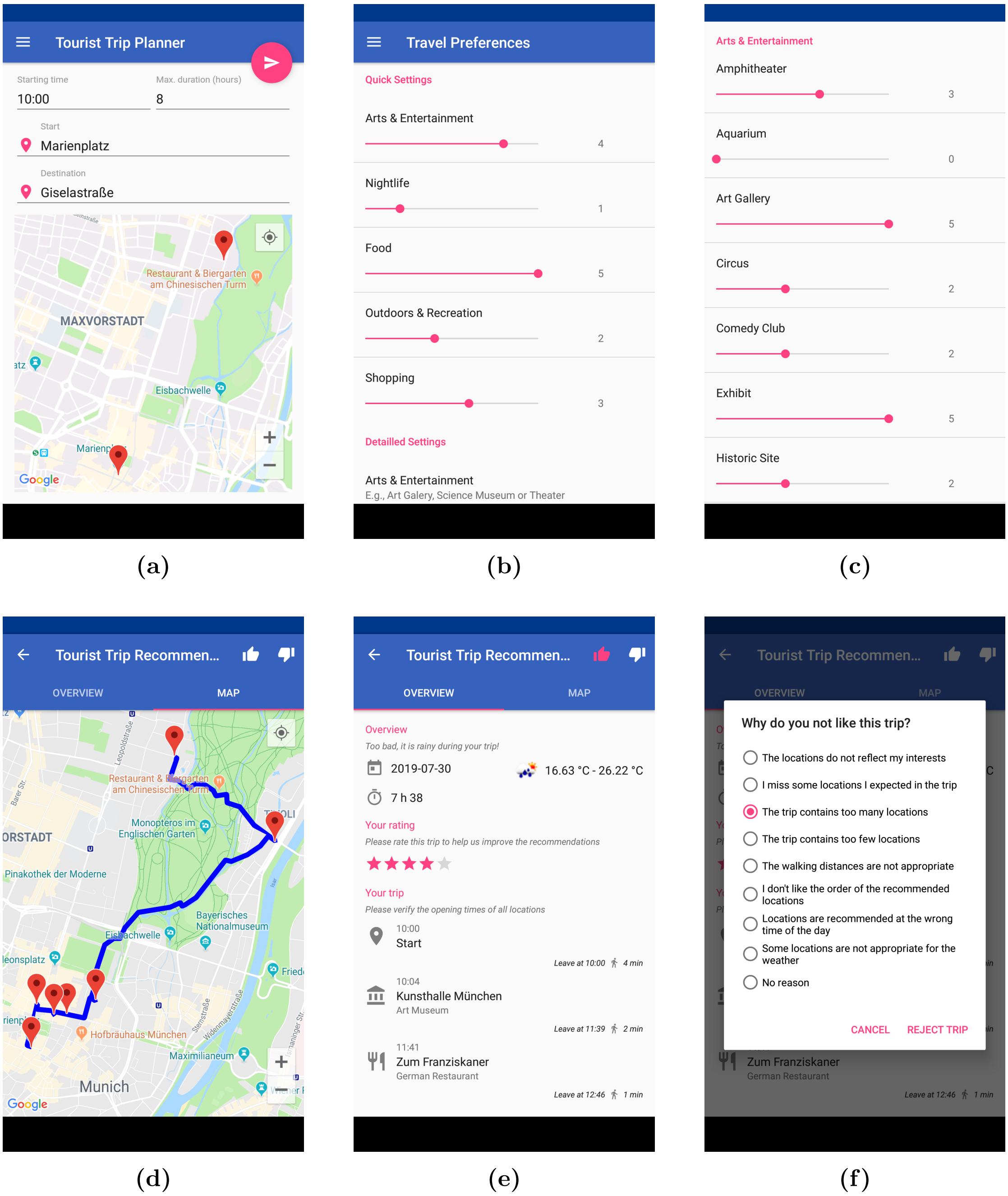
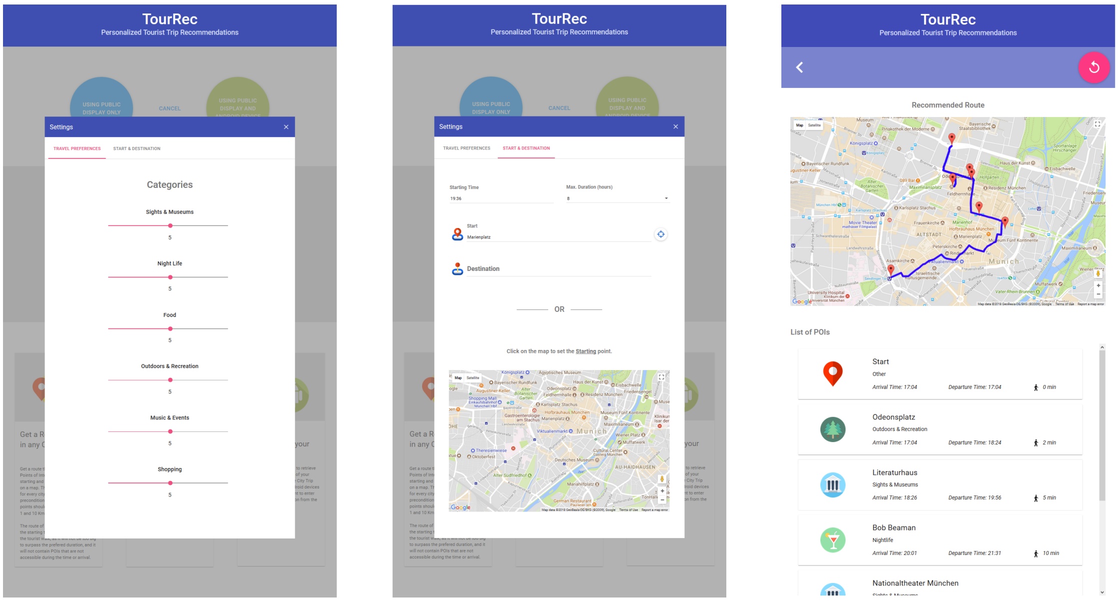


Figure 6.3: UIs of the TourRec Android application for (a) formulating a user request, (b) rating main categories, (c) rating subcategories, (d) viewing the recommendation on a map, (e) viewing the recommendation as a list with additional information, and (f) critiquing the recommendation.



(a) (b) (c)

Figure 6.4: Public Display UIs for (a) specifying travel preferences, (b) formulating a user request, and (c) viewing the recommendation on a map and as a list of POIs [193].

wherever possible. Figure 6.4 shows the most important public display UIs. Again, the final tourist trip recommendation is presented both on a map and as a list of POIs (Figure 6.4c). However, the public display variant takes advantage of the larger screen and displays both modes simultaneously. The map and list are displayed on the top and bottom of the screen, respectively.

We used the AngularJS 1.5.5[5] framework to implement the public display application. In this work, we used a kiosk system equipped with a 55-inch multi-touch screen in portrait orientation as public display (Figure 6.5). Similar tourism information kiosks can be found in many touristic areas. The kiosk system ran Windows 10 and the application could be accessed via any web browser.

6.4 Distributed User Interface Approach

The DUI approach distributes the recommendation process among a smartphone and a public display. The two main reasons for this approach are: (i) users can keep sensitive data on their private device but view the recommendation on a large display, and (ii) users can prepare a trip request prior to traveling and display a recommendation on a public display as required. We decided to use a QR code for the pairing between the smartphone and public display because it has been shown that this method provides high usability in similar scenarios (see Section 2.5.4). Furthermore, QR codes are

[5] `https://angularjs.org/` (accessed February 16, 2020)

already used in common software, such as WhatsApp[6], to pair a desktop client and a smartphone.

After the user formulated a request, the extended smartphone application allows the user to send it to a public display. The user has to scan the QR code on the public display using the smartphone's camera to transmit the request to an intermediary server application that we developed for this purpose. The public display fetches the trip request from the intermediary server application. To identify the correct smartphone, each request is labeled with a unique ID that is also encoded in the QR code. After the public display received the request, it forwards it to the backend and receives a recommendation, which is then presented to the user on the public display. The smartphone and public display applications are the same as previously presented; however, they are extended by the pairing feature. The intermediary server application is a web service implemented in Node.js 8.9.3[7].

Figure 6.5: TourRec running on a kiosk system [42].

6.5 A User Study to Compare Different User Interfaces for Tourist Trip Recommender Systems

We compared the three variants that are suitable for receiving recommendations while already traveling in a user study: the smartphone application, the public display variant, and the DUI approach.

6.5.1 Setup

We evaluated the three variants relative to UX, execution time of the selected tasks, and comfortability of use in a public space. The user study followed a within-group design. We allowed the participants to test the prototypes in random order to avoid biased results due to the learning effect. The participants were asked to execute three tasks for each platform-UI configuration:

1. Request a tourist trip recommendation between two predefined POIs.
2. Request a tourist trip recommendation between two predefined POIs with their own travel preferences.

[6] `https://www.whatsapp.com/` (accessed February 16, 2020)
[7] `https://nodejs.org/en/` (accessed February 16, 2020)

3. Request a tourist trip recommendation from the current location to a predefined destination.

The participants were asked to fill out a UEQ after every interaction with one of the tree platform-UI configurations (see Section 2.6.3). In addition, we included one extra question asking the user how comfortable they felt using the prototype in a public place.

In total, 16 people in age ranges from 18–24 years (25%) and 25–34 years (75%), 8 female and 8 male, participated in the user study. All participants were bachelor or master's degree students or had recently graduated. 62% of the participants had a technological, educational background while 38% stated that they have a background in other disciplines. Overall, the participants had rather limited experience with interactive public displays, For instance, 50% of the participants had never used a similar system previously.

6.5.2 Results

We performed statistical tests where applicable to determine whether the performance of the platform-UI configurations differed significantly relative to any of the aforementioned aspects. We used analysis of variance when the results were distributed normally and the Friedman test in other cases. The Shapiro-Wilk test for normality was performed to select the correct significance test. In case of a significant difference, we performed a post-hoc test to identify where the difference occurred, i.e., between platform-UI configurations.

Figure 6.6 shows the results of all prototypes relative to the six UEQ aspects. The attractiveness of all prototypes is considered *excellent*, which means that it is among the 10% best results of the benchmark dataset. However, perspicuity, which determines how easy it is to get familiar with the application, is significantly higher for the stand-alone smartphone mode compared to the DUI approach ($p = 0.002$). Many people are familiar with using smartphone applications. Hence, it was easier for them to learn how to use the stand-alone smartphone variant than the DUI approach. For dependability, the difference between the stand-alone smartphone and public display modes is significant ($p = 0.006$), which means that the participants felt more in control of the interaction when using a smartphone than a public display. Moreover, the public display's dependability score is *below average* compared to the benchmark dataset that comes with the UEQ because the public display scored very low for the *Secure vs Insecure* item in the questionnaire. Thus, further effort to protect user data and prevent *shoulder-surfing* is required. This was also confirmed by many participants who expressed their concerns about privacy during the study as their data would be publicly visible on the large screen. However, half of the participants stated verbally that the large screen of the public display is a great advantage. In addition, 25% emphasized that the public display would be the ideal choice when used by groups of travelers, as the larger screen can facilitate discussions among group members. Our DUI approach appears to be a promising solution because its dependability is similar to the stand-alone smartphone variant. Furthermore, the DUI approach demonstrates the highest novelty, which means

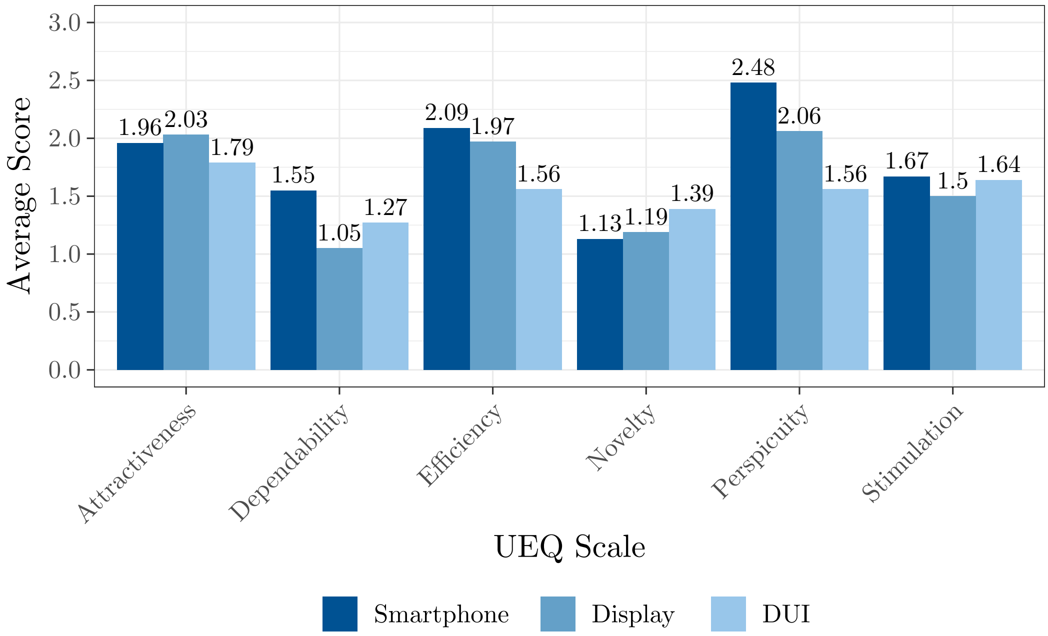


Figure 6.6: UEQ results for the three TourRec applications (adapted from [192]).

that this approach felt the most innovative and creative. However, this difference is not significant.

Table 6.1 shows the average execution times for each task and prototype. The execution times of Task 1 are significantly shorter for both the stand-alone smartphone mode ($p = 0.007$) and the stand-alone public display mode ($p = 0.015$) than the DUI approach. Task 3, which required the users to give the system access to their current location, is significantly faster on the smartphone than on the public display ($p = 0.002$) and the DUI approach ($p = 0.003$). Interacting with the pop-up window to grant location permission was faster on the smartphone than on the public display because this window was relatively large on the smartphone but difficult to see on the public display. There is no significant difference between the execution times of Task 2 which included entering the travel preferences before requesting a recommendation. Hence, specifying preferences using sliders can be done as quickly on mobile screens and large touch screens. Many people are more familiar with using smartphones but the preference sliders were larger on the public display which made it easier to select the desired ratings.

The analysis of execution time shows that there is nearly no difference between the public display and smartphone configurations. This is surprising because many participants had no previous experience with interactive public displays.

Comfortability using a smartphone in a public place is significantly higher than when using a public display ($p = 0.005$) or the DUI approach ($p = 0.005$). During the study, 75% of participants explained that using two devices is a disadvantage and too complex because they could obtain the same recommendation using a single device. However,

Table 6.1: Average execution times for each task and prototype.

	Task 1	Task 2	Task 3
Smartphone	33.38 s	77.44 s	25.56 s
Public Display	34.69 s	73.06 s	33.63 s
DUI approach	43.81 s	81.19 s	36.81 s

25% of participants emphasized that preparing the trip recommendation in advance, e.g., by entering trip parameters on the smartphone while waiting in line to use the public display, could be a significant advantage in practical use.

6.6 Summary

In this chapter, we introduced different platform-UI configurations for a tourist trip RS. The basic idea is the same among all variants that we developed: Users can rate POI categories on scales ranging from 0 to 5. A tourist trip request is composed of a starting point, a destination, a starting time, and the maximum duration of the trip. The recommended route is displayed as a list with additional information and on a map.

We developed a web-based TourRec client that is suitable for planning trips in advance. We used this prototype to compare our context-aware, Dijkstra-based tourist trip algorithm to a context-unaware solution in an A/B test in Section 5.3.3. Another user study with 19 participants proved the high usability of the web-based solution.

In addition, we introduced platform-UI configurations that allow receiving recommendations while already traveling: a mobile application, a public display variant, and a DUI approach which distributes TourRec among both smartphone and public display. The mobile application for Android devices has been developed and extended over the course of the last years. For instance, we introduced subcategories in the mobile application that allow users to specify their travel preferences more precisely. Users can also rate recommendations and critique them. The feedback is stored in TourRec's data tier. In a user study with 39 participants, the mobile application received a high usability score. We adapted the mobile application to public displays. In this work, we used a kiosk system as public display. One main advantage of the public display variant was the large screen which allowed us to display the recommendation as a list and on a map simultaneously. In addition we developed a DUI approach that allows users to specify their preferences and a tourist trip request on the mobile device but view the recommendation on the large screen.

We compared the UX of the mobile application, the public display variant, and the DUI approach in a user study. In this study, very high attractiveness was demonstrated by all approaches. The results also showed that integrating public displays into a tourist trip RS is perceived as an advantage by some users; however, the feedback received also indicates that public displays could become more valuable when a group of users attempts to agree on a tourist trip. This is why we show how to extend TourRec to enable group recommendations in the following chapters. Furthermore, privacy concerns relative to using a public display remain. The DUI approach allows the user to keep sensitive data, such as travel preferences, on the private device while benefiting from the public display. This is particularly important in a group recommendation scenario where the users want to share a mutual display but not reveal their personal preferences to other group members. However, further efforts are needed to protect the data on the public display and prevent *shoulder-surfing*.

7 Recommending Tourist Trips to Groups of Users

Many people want to travel in groups, such as families, friends, and colleagues. It is challenging to recommend good sequences of POIs to groups of users: Group members can have different expectations towards a tourist trip but a group recommendation is supposed to consider the travel preferences of all group members. Consequently, GRSs have to find compromises that are appreciated by all group members to guarantee a positive experience for the whole group.

In the previous chapters, we introduced the TTDP for individuals and presented algorithms and platform-UI configurations to solve it from a user-centered perspective. In this chapter, we show how to solve the TTDP for groups of users. This variant of the TTDP comes with an additional constraint: instead of scoring POIs based on the travel preferences of only one user, a GRS uses multiple sets of travel preferences as input to generate a tourist trip recommendation. As explained in Section 2.4.2, two types of group recommendation techniques can be distinguished for this purpose: The user profiles of all group members can be aggregated using a Social Choice strategy. The group profile is then used to request a recommendation (AP). Furthermore, a recommendation can be made for every user individually before the recommendations are combined into one group recommendation (AR).

A number of studies have been conducted on group recommendation strategies in various domains. In the tourism domain, GRS research focuses on recommending single items or travel plans (see Section 2.4.3). Furthermore, to the best of our knowledge, there are no user studies that evaluate different group recommendation strategies to solve the TTDP. In this section, we show how to apply established AP and AR group recommendation strategies to the TTDP. Furthermore, we introduce two novel approaches, including a strategy called Split Group, which extends the AP approach and allows groups to split into smaller groups during a trip. We compared all strategies in a user study with 40 real groups. The results of this user study allow us to answer our third RQ: *"Which group recommendation strategies provide the highest user satisfaction when solving the TTDP for groups?"*.

The content of this section has been published in [194] with some revisions.

7.1 Problem Description

The problem that we want to solve in this study is an extension of the formulation of the TTDP that we described in Section 5.1. Instead of recommending tourist trips to individuals, the goal in this study is to generate recommendations that satisfy groups of users. For this purpose, a group has to specify a mutual starting point, a mutual destination, and the maximum duration of the trip. Every group member can have different interests. For instance, one group member could be interested in visiting museums, while the rest of the group prefers outdoor activities. The goal is find a tourist trip from the starting point to the destination that takes into account the interests of all group members.

We solve the described problem as follows: All group members can specify their individual travel preferences separately by rating POI categories, as in our previously presented approaches for individuals. Then, the RS collects the preferences of all group members. The group members' preferences and relevant contextual factors are used to determine the POI profits. The profit of a POI can either be the same or different for each group member, depending on the selected group recommendation strategy. Using this information, the RS generates a tourist trip recommendation from the mutual starting point to the mutual destination that is presented to all group members. We developed different group recommendation strategies that all implement the Dijkstra Plus algorithm. We showed that the context-aware variant of the Dijkstra-based tourist trip algorithm recommends satisfying tourist trips to individuals (see Section 5.3). Moreover, the Dijkstra Plus variant of this algorithm counteracts the equalization of contextual conditions but avoids disproportionate walking times to the final destination (see Section 5.6).

7.2 Group Recommendation Strategies for the Tourist Trip Design Problem

In this section, we present the group recommendation strategies that we developed to solve the TTDP for groups. We show how to apply AP and AR strategies and introduce our two proposed strategies termed Split Group and Connect Segments.

7.2.1 Aggregating Profiles of Users

As explained, the goal of AP is to create a common user profile that reflects all preferences of all group members. Social Choice strategies can be used to aggregate the profiles of group members.

We are interested not only in the relative positions of the ratings in each individual's category preferences, but also in the strengths of preferences. Therefore, majority-based strategies are not suitable for solving the described problem (see Section 2.4.2.1). This is why we implemented consensus-based and borderline AP strategies in this study: Average, Average without Misery, and Most Pleasure. These strategies performed well in previous experiments, whereas other strategies, such as Least Misery, performed poorly [78]. These strategies work as follows when used to solve the previously described formulation of the TTDP:

- The Average strategy calculates the average of all group member ratings for every POI category.
- The Average without Misery strategy works as the Average strategy but sets the profit of all POI categories with at least one user rating below a threshold to zero.
- For every POI category, the Most Pleasure strategy uses the maximum individual rating as group rating.

The group profile is then used together with the context ratings to calculate the profit of every POI before executing the Dijkstra-based, context-aware tourist trip algorithm.

7.2.2 Split Group

One disadvantage of AP is that it can undermine individual preferences because every group member has to use the same recommendation. Therefore, we present an extension of the AP approach, which allows every user to visit important POIs based on their personal recommendations. Figure 7.1 visualizes our proposed approach. First, the POI profits for all users are determined and aggregated to recommend a mutual trip for the group. Our implementation uses the Average strategy; however, it is also possible to use other suitable Social Choice strategies. Then, an individual recommendation is made for each user. The algorithm checks if POIs from the mutual trip could be replaced with the POIs from the individual trip for every group member. To determine the best replacement for a POI in the mutual trip, the profit of every candidate POI (i.e., every POI in the individual trip) is divided by the overall distance that the user needs to walk from the previous POI to the candidate and to the following POI. If the profit of a candidate is higher than the current POI's profit, it replaces the POI, that is, the user leaves the group to visit this POI. Only POIs with a profit below a threshold t can be replaced because we believe that no group member should leave the group if the mutual recommendation is already satisfying for the group member. Furthermore, our approach avoids waiting times for subgroups. For this purpose, the proposed duration of stay of a replacement is adapted to the POI that it replaces. A POI can only be replaced by another POI if the replacement's default duration of stay is not less than a quarter and not more than four times the original POI's duration of stay. This avoids replacing a POI by another POI with a completely different optimal duration of stay.

Figure 7.5 shows a trip generated by Split Group.

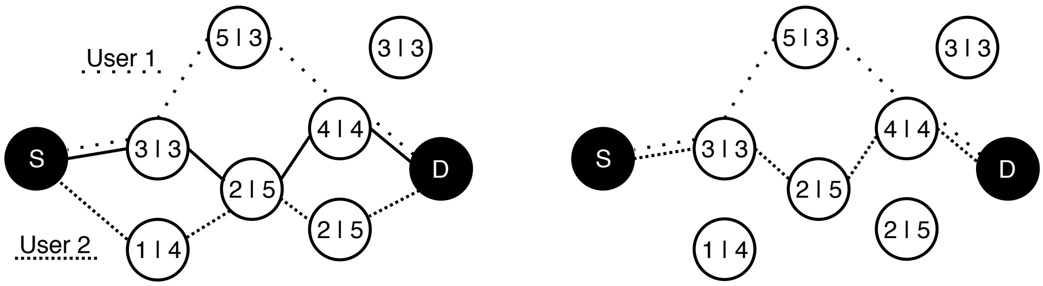


Figure 7.1: Visualization of Split Group with two users and threshold $t = 3$ for replacing POIs [194]. The profits for each user are displayed in the vertices in the format (user 1|user 2). The left side shows the mutual trip (solid line) and the individual trips (dotted lines). The final recommendation on the right side is the mutual trip for both users; however, user 1 visits one POI from the individual recommendation before rejoining user 2 at the last POI.

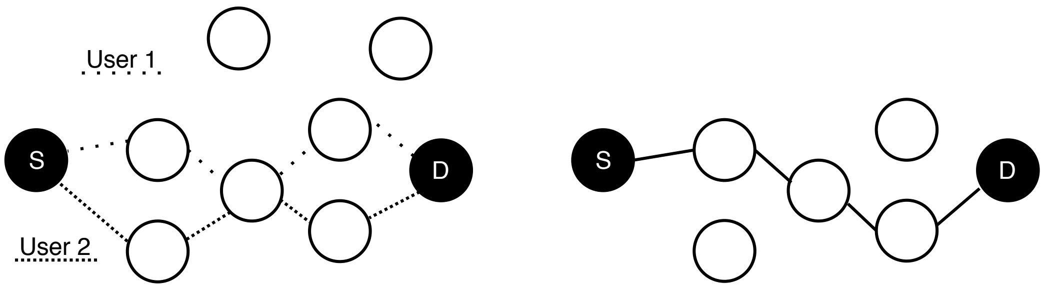


Figure 7.2: Visualization of our AR approach with two users. Only POIs that are part of at least one of the individual recommendations (left side) are candidates for the group recommendation. A Social Choice strategy is then executed to generate the trip recommendation on the right side.

7.2.3 Aggregating Recommendations

AR means that a recommendation is generated for every group member individually before the recommendations are combined into one group recommendation [79]. Figure 7.2 visualizes our proposed approach which is based on the idea of *aggregated predictions* (see Section 2.4.2.2): We apply a Social Choice strategy on the POIs that are part of at least one of the individual trips to aggregate recommendations. The profit of a POI is increased by the factor of n^2, where n is the number of individual routes that contain the POI. The idea is to make it more likely that the POIs that are part of multiple individual recommendations appear in the group recommendation. In this study, we used the Average strategy to test this approach.

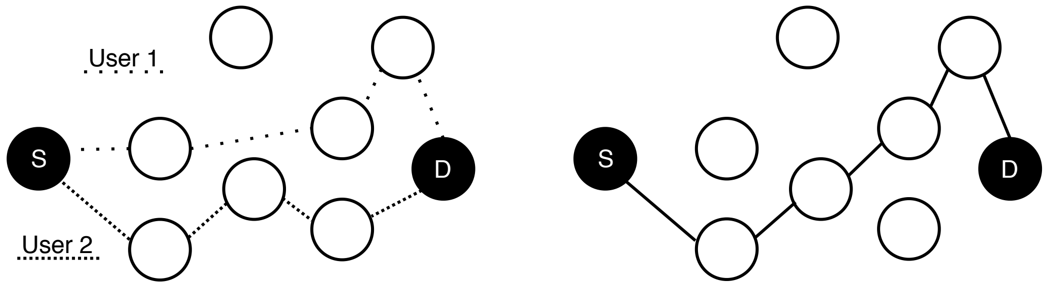


Figure 7.3: Visualization of Connect Segments with two users. The first two POIs in the recommended trip on the right side are taken from user 2's, the last two POIs from user 1's individual recommendation (left side).

7.2.4 Connect Segments

Connect Segments is a variation of AR that follows the idea that during a trip, every group member can visit their favorite POIs for a specified period. For instance, in the morning, the group visits a museum that user A likes the most, then they have lunch at user B's favorite restaurant, and so on.

Connect Segments does not require aggregated preferences. It only calculates individual recommendations and picks single POIs from these sequences to generate a new recommendation. In this study, groups visit a segment of two POIs from a group member's individual recommendation before the next two POIs are taken from another group member's individual recommendation. The order of the group members in this process is determined randomly. This procedure continues until either the end of all individual recommendations is reached or no more time is left.

Figure 7.3 illustrates an example of Connect Segments with a group of two users.

7.3 User Study

We evaluated all of the presented recommendation techniques for groups in a user study. Consequently, we compared the following six techniques for solving the TTDP for groups in this user study:

- Average (AP)
- Average without Misery (AP)
- Most Pleasure (AP)
- Split Group (AP)
- Aggregating Recommendations (AR)
- Connect Segments (AR)

The user study was conducted as a laboratory study. It was part of a larger study that was composed of two parts: a user study to evaluate recommendation techniques (this section) and a user study to evaluate different GRS configurations and group interaction with tourist trip RSs in public spaces (see Chapter 8).

7.3.1 Participants

The participants were made to register for the study as groups because we wanted to conduct our user study with real, non-synthetic groups. We limited the study to groups of three to reduce the number of experimental conditions. In total, 120 participants (40 groups) participated. Our goal was to cover all user types of a tourist trip GRS in this study. Hence, we were looking for participants with different backgrounds, including different experiences with tourism applications and different travel frequencies. All participants completed a demographic and group-related questionnaire at the end of the user study. The questionnaire can be found in Appendix C.

The participants were in the age ranges of 18–24 years (60 %) and 25–34 years (40 %), 50.8 % were females and 48.3 % were males. One participant preferred to not specify the gender. The participants were mainly composed of students and alumni. 23.3% hold a high school diploma or equivalent degree, 63.3% a bachelor's degree, and 13.3% a master's degree or higher.

We asked participants for self-assessment of their group's type: 62.4% called their group "close friends" or "family", 31.7% "student fellows", and 3.3% did not know the other group members prior to the study. The rest gave multiple answers. Using this self-assessment, we clustered participants into two group types (see Section 2.4.1): If all group members chose close friends or family as their group type, we categorized the group as primary group. If at least one group member chose another group type, the group was characterized as secondary group. Collectives and categories were not part of this study since these group types do not plan mutual trips. In total, 17 primary groups and 23 secondary groups participated in our user study.

Our study covered participants with different experiences with tourism-related applications on smartphones and tablets. Almost half of the participants (49.2%) uses such mobile applications less than once per month or never. However, 23.4% of the participants uses tourism applications on smartphones or tablets weekly or daily. Our study also covered participants with different travel frequencies. Most participants (57.5%) travel more than three times per year. A small share (5%) travels rarely, i.e. not more than one time per year.

We asked the participants to rate statements about general travel preferences on a 5-point Likert scale ranging from 1 (strongly disagree) to 5 (strongly agree). The statements and average responses can be found in Table 7.1.

For our participants, visiting the attractions with the highest personal interest and that satisfied a majority of the group members was more important than visiting the favorite attractions of every group member and avoiding unwanted attractions. The question whether or not the participants appreciate splitting during a trip received contradictory responses (Figure 7.4). Many participants stated that splitting was accept-

Table 7.1: General travel preferences in groups.

Statement	Mean	SD
When traveling in a group, I think it is okay to split up for a while.	3.58	1.31
When traveling in a group, I want to avoid attractions that do not interest me.	3.76	1.02
When traveling in a group, I want to visit the attractions that I am most interested in.	4.38	0.70
When traveling in a group, I want to visit the favorite attractions of every group member.	3.67	1.07
When traveling in a group, I want to visit attractions that satisfy the majority of the group.	4.22	0.81

able, even when traveling with close friends. However, 25 % of the participants felt that splitting should be avoided (response 2 or lower).

7.3.2 Setup

During the user study, we generated three trip recommendations for each of the six recommendation strategies presented in Section 7.2. Consequently, the participants received and rated 18 trips. The order of the strategies was randomly chosen for every group to reduce the learning effect on the results. To reduce the number of independent variables, the trips came with fixed conditions. Every recommendation strategy was used to generate trips with three pre-defined start and destination pairs in the city center of Munich, Germany, a touristic area that offers of a wide range of POIs. The participants were living in Munich or its suburbs and were hence familiar with the city and its attractions. All the trips had the same maximum duration (8 h). The weather during each trip was set to sunny, and the group size was set to three, similar to previous group recommendation research [78].

Every participant was equipped with an Android smartphone with the extended TourRec application that enabled group recommendations. The participants entered their travel preferences separately in their own devices. Again, travel preferences were specified by rating POI categories on a scale ranging from 0 (not interested in this category) to 5 (strongly interested in this category). In this study, only subcategories that were available in the test area in Munich, Germany, were provided to the users. Consequently, a user profile was composed of 42 subcategories (see Appendix B). The connection between the smartphones to exchange travel preferences automatically and display the recommended trip on all devices simultaneously was hard coded. One device collected the user preferences of all smartphones and requested a recommendation using the selected group recommendation strategy which was not revealed to the group members. The recommended trip was eventually displayed on all smartphones on a map and as a list with additional information, such as arrival times (Figure 7.5).

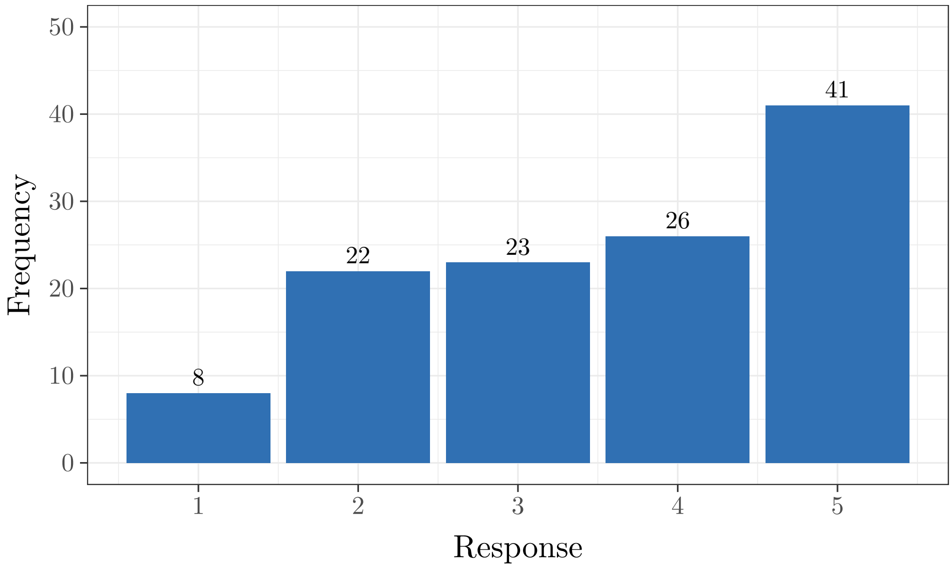


Figure 7.4: Responses to the statement *"When traveling in a group, I think it is okay to split up for a while"*.

After examining the recommendations, the participants were asked to rate five statements that we adapted from the ResQue questionnaire on a 5-point Likert scale ranging from 1 (strongly disagree) to 5 (strongly agree) (see Section 2.6.3):

(S1) The recommended trip matched my personal interests.

(S2) The attractions in the recommended trip are diverse.

(S3) The order of attractions in the trip is satisfactory.

(S4) The recommended trip is feasible for a walking tourist.

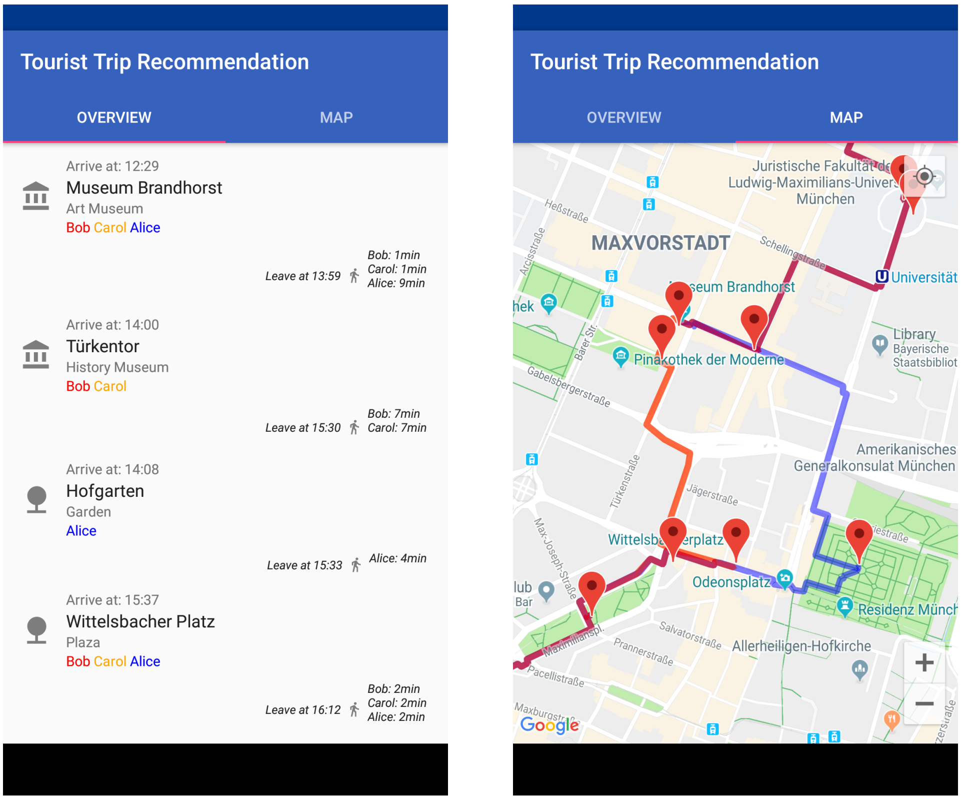
(S5) I would make this trip when traveling with my group.

In this study, we were interested in the individual satisfaction with the recommended trips and therefore did not ask for actual group decisions.

7.3.3 Results

Table 7.2 shows the average responses for all recommendation strategies and whether there is a significant difference between the strategies based on the Friedman tests that we conducted.

The results show that there is a significant difference between the strategies with regard to each of the five criteria. Split Group performed the best in three out of five criteria, and it had a score similar to the Average strategy and the Average without Misery strategy for S4 (*feasible for walking tourists*) and S5 (*intention to make the trip*). Conover's post-hoc tests [195] show that Split Group:



(a) (b)

Figure 7.5: Extract from a recommendation generated by Split Group [194]. After visiting an art museum, Bob and Carol visit another museum while Alice spends some time in a garden before rejoining Bob and Carol at a plaza.

- matches the personal interests of the participants significantly more than the Most Pleasure strategy ($p < 0.001$), AR ($p < 0.001$), and Connect Segments ($p < 0.001$),
- generates a significantly higher diversity of the trips than all the other strategies ($p < 0.001$) except for the Average strategy.
- ensures a significantly better ordering of items in the trip than the Most Pleasure strategy ($p = 0.004$), AR ($p = 0.025$), and Connect Segments ($p < 0.001$),
- creates trips that are significantly more feasible for walking tourists than Connect Segments ($p = 0.023$), and
- creates trips that the participants would much rather make when traveling with their groups than the trips generated by the Most Pleasure strategy ($p < 0.001$) and Connect Segments ($p < 0.001$).

Table 7.2: Average responses for all scenarios on a 5-point Likert scale (Note: $*p < 0.05$; $**p < 0.01$; $***p < 0.001$. Highest value marked in bold.)

Q	Avg	AwMi	MoPl	Split	AR	CoSe	Sig.
S1	3.74	3.73	3.48	**3.88**	3.56	3.52	***
S2	3.71	3.44	3.46	**3.85**	3.52	3.49	***
S3	3.64	3.60	3.51	**3.72**	3.56	3.39	***
S4	3.98	**4.03**	3.96	3.99	4.02	3.84	**
S5	**3.49**	3.43	3.19	3.48	3.32	3.21	***

Only in the Average strategy our tests did not reveal any significant difference from Split Group in any of the five statements. The trips generated by the Average without Misery strategy were similarly rated by the participants; however, the diversity of these trips (S2) was significantly less than the diversity of the trips generated by the Average strategy ($p = 0.005$) and Split Group ($p < 0.001$). The worst strategies in our experiment were Most Pleasure and Connect Segments. Average ($p < 0.001$), Split Group ($p < 0.001$), and Average without Misery ($p = 0.003$) created trips that the users would much rather make (S5) than the trips generated by the Most Pleasure strategy. Trips generated by the Connect Segments strategy had the worst performance with regard to the order of POIs (S3). This was expected because our first implementation of the Connect Segments strategy combined parts of different trips without a post-hoc optimization of the order of the POIs in the new trip.

We analyzed whether the willingness to split and the trip ratings generated by Split Group depended on the group type. Our results indicated that there was no significant difference between primary and secondary groups. Thus, we did not find an effect of group type on the willingness to split during a trip. This was also confirmed by comments received from many participants after the study; these participants explained that they would split temporarily, even when traveling with a very close person, if this satisfies everyone's needs.

Finally, we compared the ratings of Split Group provided by people who were willing to split during a trip with the ratings given by people who thought that splitting should be avoided or is not an option at all; the result of the Wilcoxon rank sum test shows that there is a greater possibility that the former group would make a trip generated by Split Group rather than the latter ($p = 0.007$).

7.3.4 Discussion

Our user study revealed that many people want to visit the attractions that they are most interested in when traveling in groups. Furthermore, many of the participants were willing to split for some time during a daily trip, even when traveling with a primary group, such as close friends and relatives. The option to split during a trip allows every user to visit their favorite POIs and consequently, the quality of the recommended trips can be improved. However, 25 % of the participants wanted to avoid splitting, or they completely rejected the idea. UIs in practical RSs should allow users to specify if they are willing to split during a trip. For those who prefer to travel together, another strategy, such as the AVERAGE strategy, could be used to generate recommendations of a similar quality.

The CONNECT SEGMENTS strategy that we proposed is another way to ensure that every group member can visit their preferred POIs. However, our user study revealed that groups were less satisfied with trips that were generated using this strategy because it led to a suboptimal order of POIs in a trip. To overcome this problem, we suggest extending this algorithm by using a post-hoc optimization phase and evaluating the extension in a subsequent study.

In this study, we set the group size to three to reduce the number of variables. Although many people were open to the idea of splitting during a trip, certain participants stated that they would not be willing to split into smaller groups because they did not want to travel alone. Therefore, we believe that the SPLIT GROUP algorithm would perform even better in large groups; this should be verified in future studies. A limitation of our study was the integration of the AVERAGE strategy into SPLIT GROUP. We did not test combinations with other Social Choice strategies, which could have led to different results. Furthermore, our SPLIT GROUP approach considers only single users that are willing to leave their group during a trip. An alternative approach could identify subgroups based on the users' travel interests. In this case, all members of a subgroup would stay together when leaving the main group.

Our algorithms did not consider POI categories when suggesting a group to split. In a few cases, a group was supposed to split for lunch or for dinner. The feedback we received from these groups was that splitting was not an option during such activities, even when the group members had different food choices. In addition, the groups did not prefer to split when the categories that the subgroups were supposed to visit were similar, for example, a garden and a park. We suggest determining the categories that would be optimal for splitting in future work. The findings can be used to optimize the SPLIT GROUP algorithm.

7.4 Summary

In this chapter, we described the problem of solving the TTDP for groups of users. In our scenario, the goal is to recommend a tourist trip from a mutual starting point to a mutual destination that considers the preferences of all group members. For this purpose, we adapted AP and AR strategies to solve TTDP for groups and introduced two novel strategies: SPLIT GROUP and CONNECT SEGMENTS.

Social Choice strategies can be used to aggregate the preferences of different group members. We implemented the AVERAGE, AVERAGE WITHOUT MISERY, and MOST PLEASURE strategies as examples of AP. These strategies create a group profile which is used in the context-aware, Dijkstra-based tourist trip algorithm. SPLIT GROUP is an extension of the AP approach which replaces POIs from the group recommendation by POIs from individual recommendations. AR strategies generate recommendations for every group member individually before they are combined into one group recommendation. Our proposed solution applies a Social Choice strategy on the POIs that are part of at least one of the individual trips to aggregate recommendations. Our Connect Segments approach follows the idea that during a trip, two POIs are based on one user's preferences before the next two POIs are taken from another group member's individual recommendation.

We compared all group recommendation strategies in a user study with 40 real groups. Our study revealed that many participants preferred to visit the attractions that they are most interested in or that satisfied a majority of the group members. Recommendations made by Split Group best achieved this goal. It matched the personal interests of the participants significantly more than most of the other strategies and ensured a higher diversity of the trips. Furthermore, it ensured a significantly better ordering of items in the trip than other strategies and many participants were willing to make trips proposed by Split Group. Only in the Average strategy our tests did not reveal any significant differences from Split Group. Connect Segments performed worse in our study, especially with regard to the order of POIs in a trip. It combines parts of different trips in one recommendation and therefore requires an additional post-hoc optimization of the order of POIs in the generated trip.

The results of our study indicated the willingness to split during a trip does not depend on the group type; many participants would split temporarily, even when traveling with families or close friends. However, some participants wanted to avoid splitting, or they completely rejected the idea. Practical GRSs should allow groups to specify whether or not they are willing to split temporarily during a trip and under which circumstances. For instance, some participants did not want to split when traveling in small groups or when the POI categories that the subgroups were supposed to visit were similar.

8 Group Interaction with Tourist Trip Recommender Systems in Public Spaces

Recommendation algorithms that take into account the user's preferences and context are not the only requirement for successful RSs. In Chapter 6, we showed that a RS's platform and UIs have a high impact on the user experience. The same applies when a group of users interacts with a tourist trip RS: On the one hand, group recommendation strategies have to come up with recommendations that satisfy all group members. On the other hand, a GRS's UIs must support groups in finding the best recommendation [84]. The tourism domain poses a particular challenge for UIs in a GRS: Because tourists often change their plans or look for new attractions while traveling, they need to interact with the system when moving and in public spaces. Different configurations for receiving a group recommendation in this scenario are possible: Group members can use their mobile devices independently to specify their travel preferences and then leave preference aggregation to the GRS. However, integrating a large, shared display, such as a public display, into the RS promises a more open discussion about group members' preferences and thus higher satisfaction with the group recommendation.

We presented the TTDP for groups and recommendation strategies to solve it in Chapter 7. In this chapter, we extend our findings from Chapter 6 by introducing three platform-UI configurations for using a GRS in public spaces: a smartphone variant, a public display application, and a DUI approach combining both devices. We conducted a large user study on groups interacting with each of these prototypes. While most previous works used synthetic groups to evaluate GRSs [11], we conducted our user study with real groups, such as groups of friends or colleagues. This allowed us to analyze travel preferences of different group types, understand their behavior and decisions when interacting with GRSs for tourist trips, and learn how they can be supported in finding the best recommendation considering all group members' preferences. Our contribution is a better understanding of the needs of primary and secondary groups and the challenges they face when interacting with GRSs in public spaces. Consequently, the results of this study allow us to answer our fourth and fifth RQs: *"How do different group types agree on decisions when interacting with a GRS for tourist trips and how fair are their decisions?"* and *"Which platform-UI configurations for receiving group recommendations support groups the best when looking for a tourist trip with regard to different UX criteria?"*.

The content of this section has been published in [196] with some revisions.

8.1 Developed Group Recommender System Configurations

We developed three configurations of a GRS for tourist trips based on the TourRec prototypes that we developed for individuals (see Chapter 6). We extended the TourRec RS with a critiquing feature that allows users to improve the recommended trip in an iterative manner even after openly discussing the first proposal. Critiquing can be done both on the POI and trip levels. On the one hand, users can either "pin" or reject single POIs. Pinning means that a POI has to be part of the recommendation in the next iteration; rejecting means that the POI will be discarded. On the other hand, users can critique the whole trip. Critiques that our prototypes offer are: recommend a longer or shorter trip, spend more or less time at all POIs on average, or recommend a different trip (consider pinned and rejected POIs but do not change the trip duration or average durations of stay).

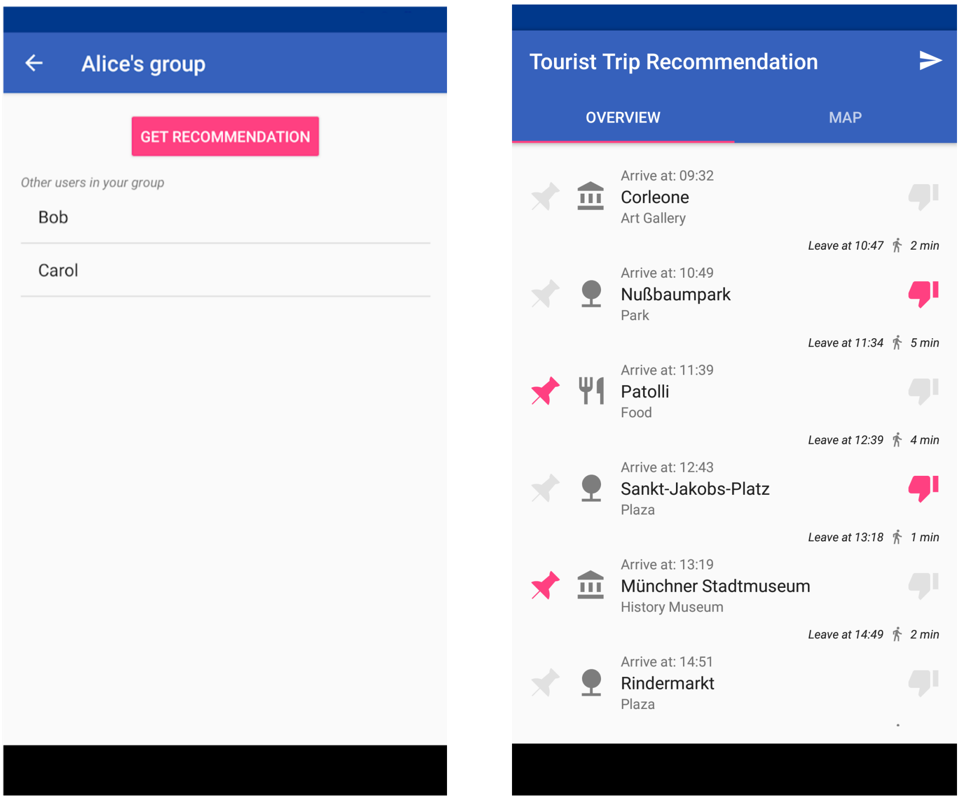
In the following, we introduce all of our configurations for group recommendations.

8.1.1 Smartphones Only

The first configuration uses multiple smartphones, one device for each group member. This configuration has the advantage that all users can specify their preferences and critique recommendations individually and privately hidden from other group members to avoid social embarrassment and manipulation [80].

After all group members have entered travel preferences individually on their smartphones, a group recommendation can be requested. This process is adapted from multiplayer video games: Within the application, one person can create a new group that others nearby can find and join (Figure 8.1a). By joining the group, the user's preferences are automatically aggregated with the other group members' preferences. In this work, the connection between devices is hard-coded using a Node.js 8.9.3 server application (see Section 6.4). We used the Average strategy to aggregate preferences in this work. This Social Choice strategy performed well in our experiment in Section 7.3. Furthermore, there is evidence that it is used by humans to make decisions and was already applied in similar GRSs in tourism [78, 80]. The request with aggregated preferences is sent to the backend, and the recommendation is then displayed on all devices.

Every group member can critique POIs independently. For this prototype, we chose a veto approach: A reject from one user is enough to remove a POI. If a POI is pinned, but not rejected by anyone, it will stay in the recommendation. Hence, this approach could also be interpreted as a variant of the Average without Misery strategy. Figure 8.1b shows an example in which one group member wants to keep a restaurant and a museum but rejects two outdoor activities. After every group member submits feedback, a pop-up window for trip-level critiquing is displayed on the group creator's device. The updated trip is again displayed on all devices.



(a) (b)

Figure 8.1: UIs for (a) creating a new group on the smartphone and (b) viewing and critiquing a recommendation on the smartphone [196].

8.1.2 Public Display

In the second configuration, to receive a recommendation, all group members share one display. This means that the group has to agree verbally on group preferences and that their individual preferences are not automatically aggregated. Therefore, the group members must reveal their preferences to other group members and discuss them in the group, but doing so might be uncomfortable for some. Furthermore, some group members risk feeling left out during the preference elicitation when part of the group dominates the interaction with the display. Another problem of discussing preferences in a group are *anchoring effects*: Group members who first express their preferences have a stronger influence on the decision made by the group than other group members [197]. Conversely, as explained in Section 2.4.3.2, knowing other group members' preferences can sometimes facilitate preference elicitation due to reduced effort in coming up with one's own preferences. In addition, group members can learn from each other when openly talking about their preferences.

Like the smartphone variant, our public display application applies the Material Design, but attempts to benefit from the larger display area wherever possible. Subcategories, as introduced in Section 5.6.2.1, can be shown below main categories and switched using tabs. The recommendation list and map are displayed on one screen (Figure 8.2a). POIs can be critiqued in the same manner as in the smartphone application. Again, the group has to agree verbally on their feedback before entering it into the display. After submitting feedback, a pop-up window asks the group to provide trip-level feedback prior to a new recommendation.

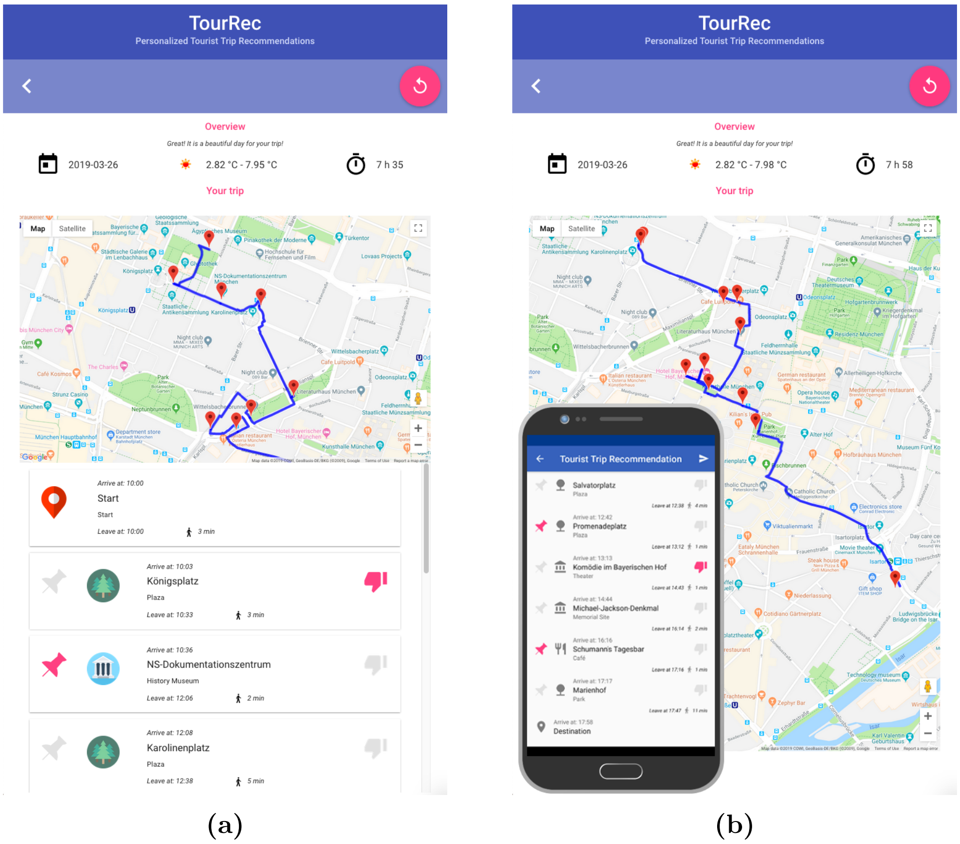


Figure 8.2: UIs for viewing and critiquing a recommendation (a) on the public display and (b) in the DUI approach [196].

Our public display prototype is a web application developed with AngularJS 1.5.5 that runs on any web browser. The public display we used in this study is again a kiosk system equipped with a 55-inch multi-touch screen in portrait orientation (Figure 6.5).

8.1.3 Distributed User Interface Approach

The third configuration we suggest, the DUI, is a combination of the previous prototypes. Our DUI approach's general idea is that group members can enter preferences independently in smartphones, but send them to a public display, if available. In this way, group members can specify their personal travel preferences before traveling, but generate and discuss recommendations on a shared display. This reduces the risk of *shoulder-surfing.*

Every group member can send preferences to a nearby display by selecting corresponding menu items on the mobile app and public display. Again, a hard-coded connection was used for this study. The public display shows how many user preferences were sent to the display and aggregates them automatically. Then, the group can request a new trip using the public display. The recommendation view is distributed over both devices (Figure 8.2b). The public display shows a large map with which users can interact. The POI list with critiquing options remains on every user's smartphone. Consequently, critiquing can again be done individually. Feedback on the whole trip, a decision that concerns the whole group, is given mutually on the public display. Therefore, a pop-up window appears after all group members submit their individual feedback.

8.2 User Study

In this user study, we wanted to discover which of the three aforementioned GRS configurations best supports which types of groups in finding the optimal recommendation: a smartphone-only variant, a public display application, or a DUI approach. For this purpose, we wanted to answer the following questions:

- How do different group types agree on group preferences?
- To what extent do group preferences respect each group member's preferences?
- How do the presented GRS configurations perform according to different UX criteria?

In order to answer these questions, we had to analyze how homogeneous different types of groups are in their travel preferences (see Section 8.2.3.1). With the user study's help, we want to provide recommendations for the development of GRSs used by tourists in public spaces.

8.2.1 Participants

The user study was conducted as a laboratory study and was composed of a usability test and different questionnaires. It was conducted in conjunction with the user study presented in Section 7.3. Therefore, the participants were the same 40 real groups of three users (120 participants) in both user studies.

We asked the participants how frequently they interact with public and interactive displays or information kiosks in public spaces, such as tourist areas, shopping malls, and train stations (see Appendix C). The majority of the participants (55%) use such displays less than once a month. 16.7% use them monthly, 15.8% never. However, a small share uses public and interactive displays and information kiosks weekly (10%) or daily (2.5%). Furthermore, we asked them how comfortable they feel in general when using a large display in a public space. Most of the participants (62.5%) stated that they feel very comfortable (response 4 or higher on a 5-point Likert scale ranging from 1 to 5); however 21.7% feel rather uncomfortable (response 2 or lower) when using such systems in public spaces.

8.2.2 Setup

During the usability test, each group interacted with all three prototypes, which were tested in random order to avoid biased results due to learning effect. Before interacting with a prototype, a moderator asked the group to imagine the following situation: *"It is a beautiful day in Munich. Your group just arrived in the city center, and you have the whole day left for sightseeing before you return to your hotel in the evening. However, you do not know which attractions to visit on your trip."*. Then, the group was asked to complete the following tasks on the current prototype in the order given:

1. All users specify travel preferences as individuals on smartphones or as a group on the display.

2. One user creates a group and requests a group recommendation with a fixed starting point and destination on the group creator's smartphone or on the display.

3a. All users modify the recommendation (POI level) as desired on a smartphone or as a group on the display.

3b. All users modify the recommendation (trip level) as desired on the group creator's smartphones or as a group on the display.

After each prototype, every participant was asked to complete a ResQue questionnaire for the tested prototype. Table 8.5 shows the questions that we adapted to our scenario. Additional questions we asked on participants' general travel preferences and preferred modes for different tasks are shown in Tables 8.4 and 8.6. During the entire user study, we video recorded the groups for later behavioral analysis. Additionally, in unstructured interviews, we asked the participants for their opinions on the GRS.

8.2.3 Results

In the following, we present our user study's important findings.

8.2.3.1 Group Homogeneity

Again, users were able to rate 42 categories to specify their travel preferences (see Appendix B). The *beautiful day* scenario is likely why outdoor activities were highly rated, while time-consuming indoor POIs, such as *Circus*, *Theater*, and *Opera House*, were among the least popular categories. Since all participants tested every prototype, we gathered every user's individual travel preferences from the smartphones.

A user's travel preferences are represented by a vector of length 42. We used the PCC to determine the similarity of two user's travel preferences. Figure 8.3 illustrates PCCs between all user pairs in each group (*user-user PCC*). Ideally, all pairs of group members have strong correlation of travel preferences. Group 2 exemplifies a very homogenous group. In other groups, there is no correlation between group members (group 28), or a negative correlation between two group members, as in group 34.

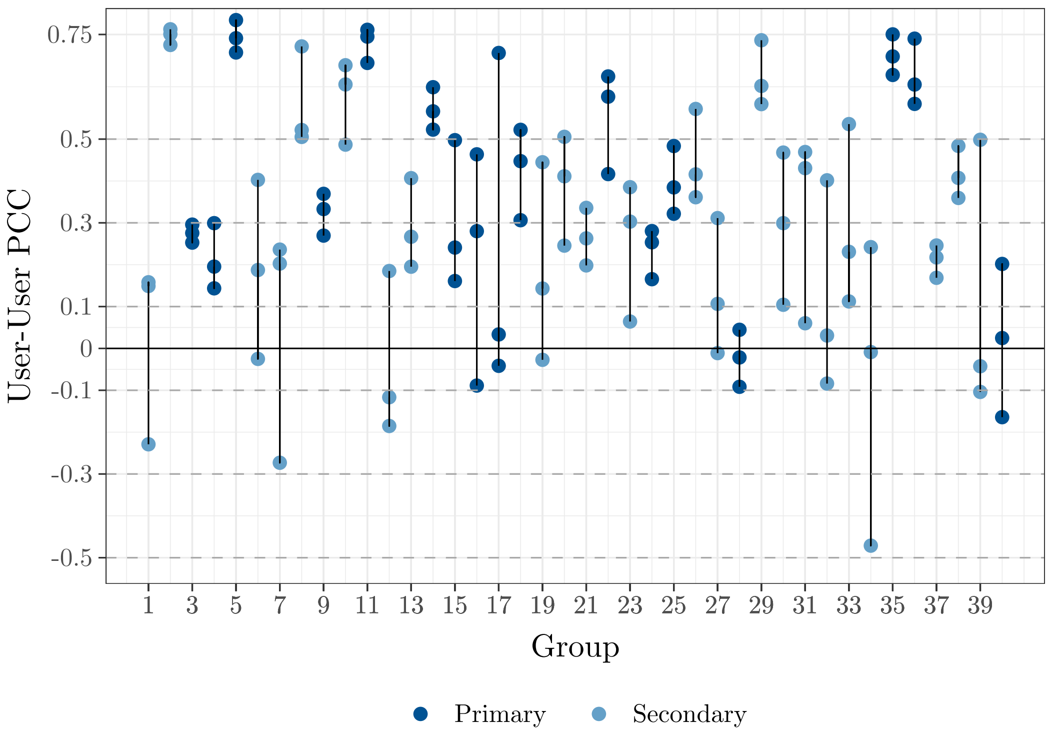


Figure 8.3: User-user correlations per group (adapted from [196]). Every dot illustrates the PCC of a user-user pair in a group (three pairs per group). Vertical lines illustrate the range in each group, dashed lines the effect size borders.

We used a group's average user-user PCC to express the group's homogeneity. A Fisher's z transformation is required before calculating an average PCC [198]. After calculating all averages and back-transforming them to PCCs, we received the distribution presented in Figure 8.4 ("User-User"). Most groups had a weak positive average correlation, but a local maximum of groups also had a strong positive average PCC.

Using this data, we can discover whether group homogeneity differs between group types. The average PCCs for primary groups ($r = 0.42$) was higher than for secondary groups ($r = 0.3$); however, a t-test on the Fisher's z transformations of the PCCs revealed that the difference was not significant ($p = 0.157$). Hence, we cannot conclude that groups with close relationships are more homogeneous in their travel interests.

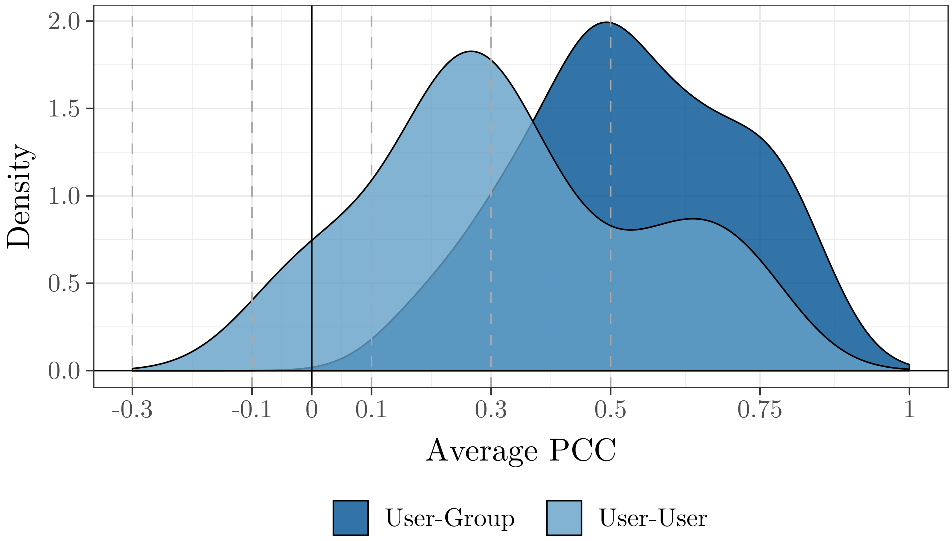


Figure 8.4: Densities of average user-user PCCs and user-group PCCs (adapted from [196]). Dashed lines illustrate the effect size borders.

8.2.3.2 Group Travel Preferences

Every group verbally agreed on their group preferences when using the public display for preference elicitation. How they decided on group preferences was completely up to participants. The comparison between the individual preferences and the group preferences allows us to analyze how different group types agree on decisions when interacting with a GRS for tourist trips and determine the fairness of their decisions.

Concordance Between Group Members' Individual Preferences and Group Preferences Figure 8.5 shows the PCC between a user's individual preferences entered on the smartphone and group preferences entered on the display for each user in each group (*user-group PCC*). Every dot shows how much this user's preferences complied with group preferences. If a user's individual preferences and the group preferences were exactly the same, the user-group PCC would be 1 for this user. We calculated the average user-group PCC per group to find out how well the group preferences reflected all group members' individual preferences. Figure 8.4 ("User-Group") presents the distribution of the groups' user-group PCC.

Obviously, the average user-group PCC per group is always higher than the average user-user PCC per group since group preferences are supposed to reflect a compromise among all group members. There is also a strong linear correlation ($r = 0.91$) between the average user-user PCC and the average user-group PCC. The average user-group PCC is usually high when a group is homogenous.

A t-test on the Fisher's z transformations revealed that the average user-group PCC for primary groups ($r = 0.62$) was significantly higher than for secondary groups ($r = 0.51$)

($p = 0.048$). Hence, we can conclude that groups with close relationships can find better compromises reflecting all group members' travel preferences than can secondary groups, even though they do not have more similar travel preferences (see Section 8.2.3.1). Support in finding optimal group preferences, e.g., by automatically applying Social Choice strategies on the individuals' preferences, is thus more critical when secondary groups use GRSs.

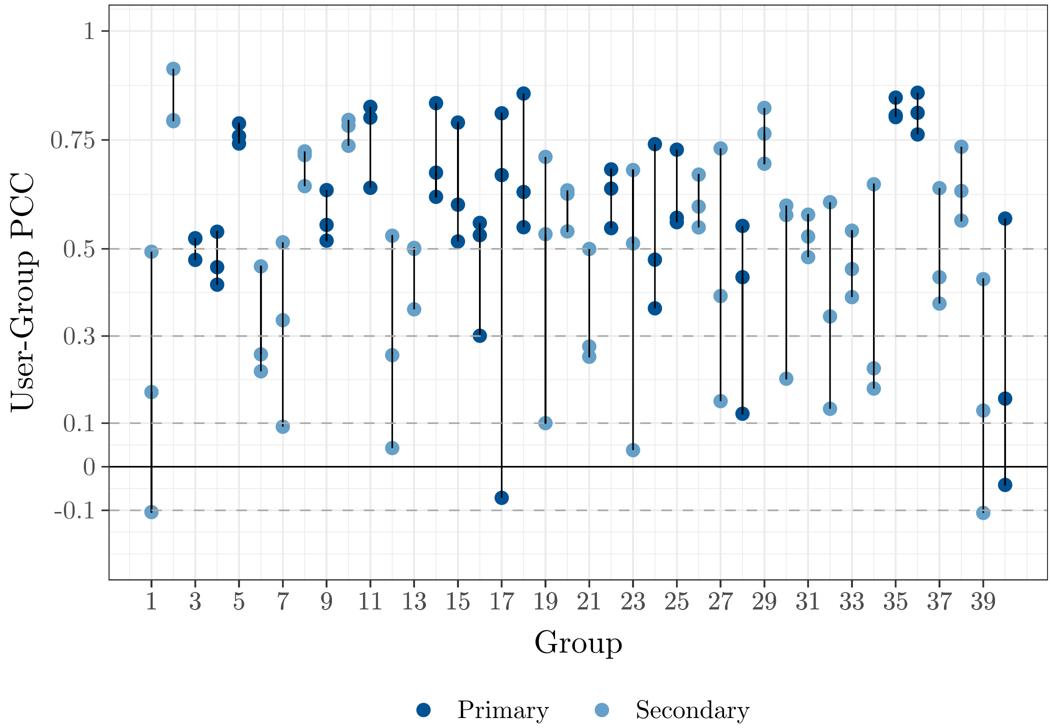


Figure 8.5: User-Group correlations per group (adapted from [196]). Every dot illustrates the user-group PCC of a user (three users per group). Vertical lines illustrate the range in each group, dashed lines the effect size borders.

Fairness Another metric for analyzing group preferences is fairness. Ideally, all group members have high correlation with group preferences entered on the display, meaning that their individual preferences are represented equally. Group 5 is one example for such an ideal result (Figure 8.5). Possibly, however, preferences are dominated by one person, e.g., in group 40, group preferences had strong correlation with one user's individual preferences, but no or only a weak correlation with other users' preferences. Group 30 exemplifies two users dictating group preferences.

In our scenario, we suggest two definitions of fairness: a) a loose definition in which all users have at least a weak positive correlation with group preferences (i.e., all members'

PCCs are ≥ 0.1) or b) a strict definition in which all group members' PCCs have the same effect size. Table 8.1 shows distributions of groups according to these definitions.

Table 8.1: Ratio of fair to unfair group preferences according to both definitions of fairness.

Definition	Fair	Unfair
Loose definition	82.5%	17.5%
Strict definition	42.5%	57.5%

When applying the strict definition of fairness, the majority of groups did not have travel preferences equally representing group members' preferences. Instead, part of the group dominated group preferences. Primary groups' fairness was higher than that of secondary groups (loose: 88% vs. 78%; strict: 59% vs. 30%). This is another indicator that primary groups can find better compromises than secondary groups when discussing preferences on a shared display. Hence, secondary groups should receive more support from RSs in finding optimal group preferences. One solution to this problem is intelligent UIs that notify the group in real-time when preferences are too biased by one or two group members (see Section 8.2.4).

Furthermore, we investigated group preferences' fairness when one person was clearly leading preference elicitation, i.e., was talking most of the time and clicked the most on the display. This was the case in 15 (of 40) groups, and these groups' fairness was higher than in groups without a clear leader (loose: 93% vs. 76%; strict: 47% vs 40%). This shows that a person leading interaction with public displays can increase group recommendations' fairness by preventing multiple users from trying to optimize their personal values (see Section 8.2.3.3).

Preference Aggregation Strategies Analysis of video records seemed to show that, to come up with group preferences, most groups applied strategies similar to the Social Choice strategies presented in Section 2.4.2.1. To reveal which strategy a group most likely chose, we applied every strategy to the group members' individual preferences we gathered from smartphones to calculate "optimal" group preferences and calculated correlation with preferences that the group entered on the public display. The strategy with the highest PCC was the one the group most likely chose.

Table 8.2 shows how often a strategy was applied by both group types. For most primary and secondary groups, preferences entered on the public display most resembled to the Average strategy. The second most applied strategy was Dictatorship, meaning that group preferences correlated most with a member's individual preferences. More secondary than primary groups applied this strategy, confirming the assumption presented in Section 8.2.3.2 that agreeing on fair preferences is especially difficult for secondary groups. However, this results does not necessarily indicate that one group member had selfish reasons for dominating the preference elicitation. It is possible that the dominant person tried to make the best decisions for the whole group (see Section 8.2.3.3), thereby accidentally giving more weight to own preferences.

Strategies in which categories were consistently rated by the most enthusiastic (Most Pleasure) or most unsatisfied (Least Misery) person or in which group members could have a veto (Without Misery) were barely applied. In practice, however, groups often seemed to use combinations of these strategies. For example, a group could apply the Average strategy, but allow vetoes for a small subset of categories.

Table 8.2: Most likely applied preference aggregation strategy.

Strategy	Primary	Secondary
Average	52.9%	52.2%
Dictator	35.3%	43.5%
Most Pleasure	5.9%	4.3%
Least Misery	5.9%	0%
Without Misery	0%	0%

8.2.3.3 Group Behavior

We observed groups interacting with the display to understand better how they came up with group preferences.

Discussions Among Group Members We were interested in understanding whether group members talk to each other when they have to complete a task. Most groups did not discuss their decisions when an individual device was used for a task. For instance, only 4 out of 40 groups discussed their travel preferences when specifying them individually using smartphones. This is particularly critical when a task concerning the whole group has to be completed, for instance, providing trip-level feedback using the group creator's smartphone, as explained in Section 8.1.1. In this case, fewer than 50% of group creators asked other group members their opinion instead of deciding themselves. When using a shared display to specify preferences or modify POIs or the entire trip, groups had very vivid discussions.

User Interaction with the Display Table 8.3 summarizes how many people interacted with the public display. In most cases, all group members entered preferences and modified the trip. In three groups, only one person interacted with the display. These groups' average user-group PCC was higher than the PCC of groups in which multiple members interacted with the display; however, this difference was not significant due to the low number of observations. Nevertheless, this again raises the question of whether group preferences can be elicited in a fairer manner when only one person enters all users' preferences instead of multiple users trying to maximize their personal benefit.

After comparing the 15 groups with a clear leader and the groups with no leader, we concluded that groups with a leader had a higher average user-group PCC, but the difference was not significant.

Table 8.3: Users per group interacting with the display.

# Users	Frequency	∅ User-Group PCC
1	3	0.71
2	8	0.54
3	29	0.55

General Preferences Further insights can be derived from the post-study questionnaire (Table 8.4). Results of Mann-Whitney U tests show that primary groups felt significantly more comfortable when sharing a public display and when revealing their preferences to other group members. All types of groups much appreciated knowing other group members' preferences.

Table 8.4: General GRS preferences (Note: $*p < 0.05$; $**p < 0.01$; $***p < 0.001$. Highest value marked in bold.)

Question	Primary	Secondary	Sig.
I felt comfortable when sharing a display to plan a trip with my group	**4.51**	4.23	*
I felt comfortable when revealing my travel preferences to the other group members	**4.59**	4.30	*
I appreciated knowing the preferences of the other group members	**4.49**	4.36	

8.2.3.4 User Experience

We conducted different tests to evaluate the UX of the three GRS configurations with regard to relevant criteria from the ResQue questionnaire.

ResQue Questionnaire Results of Friedman tests and Conover post-hoc tests [195] (Mann-Whitney U test for Q5 as preference elicitation in the DUI approach is also done on the smartphone) on the ResQue questions (Table 8.5) show that specifying travel preferences using multiple smartphones was perceived as easier than using a shared display (Q5). However, the public display variant was considered to have the most attractive layouts (Q2). Participants became familiar with the stand-alone variants of the GRS faster than with the DUI approach (Q8).

When looking into different group types' responses, we learned that secondary groups believed that their personal interests were considered more when not sharing a display (Q1). Reasons for this were that the primary groups could find better compromises for group preferences when using the public display than secondary groups (see 8.2.3.2). Another important finding is that satisfaction with the three prototypes (Q11) and the intention to use the recommendation (Q13) showed no difference. We conclude that all configurations are similarly suitable for receiving group recommendations when traveling.

Task Times We measured how much time groups took to complete each task. On average, they needed 2 min 53 s to specify preferences on smartphones compared to 4 min 13 s to agree on travel preferences on the public display. This strongly significant difference was to be expected since preference aggregation in the smartphone configuration is done automatically by the GRS, not verbally by group members. While requesting a new recommendation from a smartphone takes the same amount of time as a public display, modifying a recommendation on POI and trip levels is faster in the DUI approach (1 min 15 s) than on smartphones only (1 min 31 s) and the public display (2 min 16 s). We conclude that providing feedback on single items using multiple smartphones means substantial time saving; however, tasks concerning the group as a whole, such as feedback on the entire trip, can be distributed to a public display without loss of time.

Preferred Modes for Each Task (Specify Preferences, View Recommendation, Modify Recommendation) Table 8.6 shows that the majority of participants preferred using smartphones to enter travel preferences and modify the recommendation. Particularly, secondary groups preferred to use individual devices for these data-sensitive tasks. Viewing the trip was most popular in the DUI approach that distributes the UI over two devices. In total, a small majority chose the DUI recommender as their favorite system; however, secondary groups were rather undecided, with many preferring the smartphone-only variant. This confirms our assumption that primary groups are more willing to share a display for the GRS interaction.

8.2.3.5 Qualitative Feedback

At the end of the user study, we had short discussions with each group in unstructured interviews. We asked questions such as *"How did you feel when discussing group preferences in front of the display?"* to start discussions. Further questions resulted from the first answers.

In general, most groups liked the idea of TourRec and wanted to use it when traveling the next time. They emphasized that finding the most interesting attractions is often complex, even when traveling with friends with different travel preferences.

The majority of participants found the DUI prototype the most interesting and fun to use; however, many participants stated that they preferred the smartphone variant when traveling because they are used to mobile apps. Critical for integrating public displays was a) that they are optional, i.e., they can add value to UX when available, but all functionality is available without an additional display, and b) they have a “send back to smartphone” feature allowing transfer of selected content, such as recommended POIs, to users’ devices for consumption at any time.

Using a public display for the whole recommendation process was the least appreciated approach, especially among secondary groups. However, many users can imagine using it as a backup solution when they have no smartphones with internet access while traveling.

Table 8.5: ResQue questionnaire comparing the UX of the smartphone-only (SO), public display (PD) and DUI approaches (Note: *$p < 0.05$; **$p < 0.01$; ***$p < 0.001$. Highest value marked in bold.)

	Question	SO	PD	DUI	Sig.
1	The final trip recommended to us matched my interests	**3.75**	**3.75**	3.67	
2	The layouts of this recommender system interfaces are attractive	3.89	**4.17**	4.02	*
3	The layouts of the interfaces are adequate for a tourist trip recommender system	3.98	**4.15**	4.02	
4	I found it easy to find a trip for our group using this recommender system	**4.11**	4.07	4.10	
5	I found it easy to specify my travel preferences in this recommender system	**4.31**	4.02	-	*
6	I found it easy to view the recommended trip in this recommender system	4.27	4.24	**4.29**	
7	I found it easy to modify the recommended trip in this recommender system	**3.93**	3.85	3.78	
8	I became familiar with this recommender system very quickly	**4.46**	4.42	4.26	*
9	This recommender system helped me to find the ideal trip	3.51	3.50	**3.53**	
10	I feel in control of telling this recommender system what trip I want	3.68	**3.69**	3.64	
11	Overall, I am satisfied with this recommender system	3.75	3.79	**3.85**	
12	I will use this recommender to find a trip when traveling in a group	**3.91**	3.83	3.87	
13	As a tourist, I would make the recommended trip with my group	3.89	**3.92**	3.85	

Table 8.6: Most preferred prototype for each task (Note: comparison between smartphone-only (SO), public display (PD) and DUI approach. Highest value marked in bold.)

	Primary (n=51)			Secondary (n=69)		
Task	SO	PD	DUI	SO	PD	DUI
Specify Preferences	**66.7%**	33.3%	-	**75.4%**	21.7%	-
View Recommendation	27.5%	19.6%	**52.9%**	34.8%	15.9%	**46.4%**
Modify Recommendation	**45.1%**	19.6%	35.3%	**55.1%**	17.4%	27.5%
Favorite RS	31.4%	23.5%	**45.1%**	**40.6%**	17.4%	39.1%

Some participants also had the feeling that mandatory discussions for agreeing on group preferences made them more confident about their decisions. Still, privacy was a main concern for many participants. Some who liked the public display variant during the user study were not sure if they would use it in crowded areas when people can possibly shoulder-surf. They asked for mechanisms to hide sensitive data. Furthermore, a few admitted they are often too shy to use an unknown system in front of strangers.

8.2.4 Discussion

The results of our user study show that the general idea of a GRS for tourist trips was appreciated by all group types. Groups with close relationships were more satisfied with recommended trips when sharing a display throughout the recommendation process than groups with looser connections because primary groups found fairer compromises when specifying travel preferences as a group and also felt more comfortable when sharing a display and revealing their preferences to other group members. Recommendations to secondary groups were more often unfair and biased towards preferences of part of the group. Therefore, these groups preferred to use separate devices to specify their preferences individually and to leave the preference aggregation to the GRS.

Integration of public displays in GRSs motivated groups to discuss travel-related decisions. However, group tasks on a public display can be time-consuming, and some people felt uncomfortable entering private data on a public device. This is why we suggested a DUI configuration of our GRS. Groups can specify their preferences on their personal devices even before traveling and switch to public displays, when necessary. Our results showed all group types appreciated this idea. DUIs were especially appreciated for consuming complex recommendation items, such as tourist trips, as a group because of the possibility of interacting with the content on a larger display. Distributed GRSs should give the user control over which content is distributed and also allow group members to send selected content back to their smartphones, so they can consume the recommendation even after leaving the display.

Shoulder-surfing and social embarrassment prevent people from using public displays. Our DUI configuration allows keeping sensitive data on a smartphone, but further effort is necessary to protect content on public displays. First approaches, for instance blacking out parts of the display or mirroring a passerby's position, have been developed [103], but future work should evaluate how to integrate them into GRSs.

When groups have to agree on travel preferences on a shared display, the outcome can be unfair, especially when group members do not have close relationships. We suggest implementation of intelligent UIs that promote fairness in group decisions. For example, interfaces can invite group members who have less interaction time or whose preferences are underrepresented to interact more with the display. For this purpose, the display might need to know group members' individual preferences in advance. This can be achieved by automatically sending preferences on users' devices to the display, as suggested in our DUI approach.

Another suggestion resulting from our user study is the improvement of the GRS's public display application in a user-centered approach, so it will appeal more passersby and tourists looking for recommendations.

A limitation of our study was the fixed group size of three users. The behavior of group members can differ in larger groups and hence, other platform-UI configurations could be required to ensure fair recommendations and a positive UX for the group. Furthermore, our user study was conducted as a laboratory study. We suggest repeating our experiment with different group sizes and in a public area crowded with real passersby in future work.

8.3 Summary

In this chapter, we presented GRS configurations for receiving tourist trip recommendations while already traveling: a smartphone-based GRS, a public display variant, and a DUI approach.

A smartphone-based GRS uses multiple smartphones, one device for each group member, to request a recommendation. For this purpose, one user creates a new group that others can join to connect the smartphones. After all group members have specified their travel preferences individually, the preferences are aggregated and a recommendation can be made. Every group member receives the recommendation on the smartphone and can critique POIs independently. Furthermore, a pop-up window for trip-level critiquing is displayed on the group creator's device. The advantage of this approach is that it reduces social embarrassment and manipulation.

In the public display configuration, all group members share the same display. Therefore, the group members have to reveal their personal preferences to the group to agree verbally on group preferences. POIs and the whole trip can be critiqued in the same manner as in the smartphone application. Sharing a public display can facilitate preference elicitation due to reduced effort in coming up with one's own preferences. Furthermore, group members can learn from each other when openly talking about travel preferences. However, revealing preferences might be uncomfortable for some. Another disadvantage of this approach are *anchoring effects*.

The DUI approach promises to overcome these limitations. Group members specify their personal travel preferences individually on their smartphones, but generate and discuss recommendations on a shared display. Critiquing POIs is done individually on the smartphones while trip-level feedback is done mutually on the public display.

We conducted a user study with 40 real groups to evaluate which configuration performed the best according to various UX criteria and recommendation fairness. Our analysis indicated that primary groups are not necessarily more homogenous in their travel preferences than secondary groups. However, they are able to come up with fairer travel preferences that reflect all group members' travel preferences when sharing a mutual display. When applying a strict definition of fairness, the majority of groups came up with unfair group preferences. More primary groups came up with fair preferences than secondary groups. Fairness was also higher when only one person was leading preference elicitation on the public display. Most groups seemed to apply the AVERAGE strategy to come up with group preferences. Group preferences resembled more to the DICTATORSHIP strategy for more secondary than primary groups, confirming that agreeing on fair preferences is more challenging for secondary groups.

When observing the group behavior during the interaction with the GRS configurations, we learned that most groups did not discuss decisions when using individual devices for a task. This is especially critical when decisions concern the whole group. When using a public display, in most cases, all group members interacted with the display. Furthermore, our user study showed that primary groups felt more comfortable sharing a display and revealing their travel preferences to group members than secondary groups.

The evaluation of the UX of each of the three configurations revealed that it is easier to become familiar with stand-alone variants of the GRS than with the DUI approach and specifying travel preferences is perceived easier on a smartphone than a public display. The public display variant was considered to have the most attractive layouts but secondary groups believed that their personal interests were considered more when using multiple smartphones instead of sharing a display. Specifying travel preferences and providing feedback was significantly faster on the smartphone prototypes than on the public display. Tasks concerning the group as a whole, however, can be distributed to a public display without loss of time. The majority of participants preferred using smartphones to enter travel preferences and modify the recommendation. Viewing the trip was most popular in the DUI approach. The majority of primary groups chose the DUI recommender as their favorite configuration while many secondary groups preferred the smartphone-only variant.

Unstructured interviews at the end of the user study revealed that most groups liked the idea of a tourist trip RS. Many participants liked interacting with a public display but emphasized that the integration should be optional. Furthermore, they want to be able to send back content from the display to the smartphones. The interviews also confirmed that public display applications have to increase privacy and avoid social embarrassment to become more attractive when used in public spaces. First approaches to overcome *shoulder-surfing* have already been developed but have to be adapted to the GRS scenario.

9 Conclusion

In this chapter, we sum up the findings of our work, discuss them, and suggest future work.

9.1 Thesis Summary

In this thesis, we showed how to solve the TTDP for individuals and groups of users from a user-centered perspective.

We summarized relevant fundamentals and related work that form the basis of this thesis in Chapter 2.

In Chapter 3, we presented route planning problems that serve as basic models for the TTDP. Furthermore, we identified open challenges in TTDP research that motivated our own work: context-aware tourist trips, evaluation from a user-centered perspective, and recommendations to groups.

In Chapter 4, we introduced AnyRec, a domain-independent framework supporting the development of practical RSs and the evaluation of recommendation algorithms and UIs from a user-centered perspective. It is a multi-tier architecture that is partitioned into three tiers: presentation tier, application logic tier, and data tier. This type of architecture allows AnyRec to provide a RS's most common components which makes it easier for developers to implement and evaluate novel user clients, recommendation algorithms, and data sources. We demonstrated AnyRec's capabilities by developing TourRec, a tourist trip RS for individuals and groups. It is a fully working application that is publicly available for download. The TourRec application has been used to evaluate various recommendation algorithms and user clients that we developed within the scope of this thesis, thereby solving the TTDP for individuals and groups from a user-centered perspective. Consequently, the presented architecture and the TourRec application served as a basis for answering the following five RQs of this thesis.

RQ 1: How can existing TTDP algorithms be extended to increase the satisfaction of individuals with the recommended trips? As shown in Chapter 3, many algorithms and heuristics have been developed to solve the OP and its variants for individuals. However, the majority of these approaches tackles the problem of finding a tourist trip from a pure mathematical point of view and does not take into account personal user preferences or the context of the recommendation.

In Chapter 5, we introduced several extensions that allow a more realistic modeling of the TTDP. We extended an existing tourist trip algorithm that is based on Dijkstra's algorithm to enable context-aware recommendations. Furthermore, we showed which route attractiveness attribute have an impact on a user's choice of walking route and how these attributes can be integrated into a tourist trip RS. We evaluated the extensions in user studies and showed that they can increase the user satisfaction with the recommendations. Furthermore, we presented ideas to improve these extensions, such as the concept of item dependencies, and implemented these ideas. The feedback that we gathered from an online evaluation with TourRec indicates that these ideas can further improve the quality of the recommended trips.

RQ 2: Which platforms and UIs support tourists the best in solving the TTDP in realistic scenarios with regard to different usability and UX criteria? Different types of platform-UI configurations are suitable for tourists who are looking for tourist trip recommendations. In Chapter 6, we presented a web application, a mobile application, a public display application, and a DUI approach for the tourist trip RS TourRec. We evaluated all prototypes in usability tests and conducted a user study to compare the UX of all applications that can be used while traveling: the mobile application, the public display variant, and the DUI approach. The results showed that all of our prototypes were perceived as very attractive. However, for most participants it was easier to get familiar with the stand-alone smartphone variant than the DUI approach. In addition, using a public display raises privacy issues triggered by *shoulder-surfing*, for example. The DUI approach can be a promising solution to this problem because it allows users to keep sensitive data on the smartphone. Moreover, 25% of the participants emphasized that a public display could be the ideal choice when used by groups instead of individuals. Nevertheless, the majority of the participants felt more comfortable using a smartphone than a public display or the DUI approach when using a tourist trip RS individually in public spaces.

RQ 3: Which group recommendation strategies provide the highest user satisfaction when solving the TTDP for groups? While recommendations for individuals consider only the preferences of one user, GRSs have to generate recommendations that satisfy a group of users. Two types of group recommendation techniques have been presented in related work for this purpose: On the one hand, the user profiles of all group members can be aggregated using a Social Choice strategy. Then, the aggregated user preferences are used to request a recommendation (AP). On the other hand, a recommendation can be made for every user individually before the recommendations are combined into one group recommendation (AR). Both techniques have been used in related work to recommend travel-related items, such as restaurants.

We adapted existing AP and AR strategies to solve the TTDP for groups in Chapter 7. Furthermore, we introduced two novel approaches, including a strategy called Split Group that extends the idea of AP, but allows groups to split during a trip. All of our approaches integrated the context-aware tourist trip algorithm for individuals that we introduced in Chapter 5. We compared all group recommendation strategies in a user study with 40 real groups. Results showed that the Average Social Choice strategy and Split Group work best for solving the TTDP for groups. Both strategies matched the personal interests of the participants significantly more than most of the other strategies and many participants were willing to make the generated trips. We received feedback that SPLIT GROUP could perform even better in larger groups. However, 25 % of our participants felt that splitting should be avoided. In this case, users should be able to specify whether or not they are willing to split during a trip and under which circumstances.

RQ 4: How do different group types agree on decisions when interacting with a GRS for tourist trips and how fair are their decisions? In Chapter 8, we presented the results of a user study that we conducted with 40 real groups to observe different group types interacting with GRSs. For this purpose, we let the group members specify their travel preferences individually using smartphones and as a team using a public display.

The user study allowed us to examine how different group types agree on group preferences and to what extent these group preferences respect each group member's individual preferences. Our observations revealed that groups with close relationships are not necessarily more homogeneous in their travel interests than groups characterized by rather loose relationships. However, we found evidence that primary groups can find better compromises reflecting all group members' travel preferences when sharing a public display than can secondary groups. Specifying travel preferences as a group often leads to unfair results in which only a part of the group dominates group preferences. Our results showed that this is more likely when secondary groups are traveling together. For this purpose, UIs should be designed in a way that they promote fairness in group decisions. For most primary and secondary groups, the strategy to come up with group preferences most resembled to the AVERAGE and DICTATORSHIP strategies. Our user study also revealed that in most cases, all group members interacted with the shared display. However, in 15 out of 40 groups, one person clearly led the preference elicitation. Fairer results are possible when only one person interacts with a shared display instead of multiple users trying to maximize their personal benefit.

RQ 5: Which platform-UI configurations for receiving group recommendations support groups the best when looking for a tourist trip with regard to different UX criteria? The results of Chapter 6 indicated that the integration of public displays could become more attractive when a group of users attempts to agree on a tourist trip. For this purpose, we extended the TOURREC prototypes to enable group recommendations. We presented the three resulting GRS configurations in Chapter 8: a smartphone-only, a public display, and a DUI variant. As part of the large user study that we conducted with 40 real groups, we evaluated all of these GRS configurations. The participants were equally satisfied with all configurations and the intention to use the recommendations generated by any of the three configurations showed no significant difference. Many participants called the DUI prototype the most interesting and fun to use; however, the participants found it easier to specify travel preferences using smartphones only. The public display variant was considered to have the most attractive layouts and participants became familiar with the smartphone-only and public display variants faster than with the DUI approach. The majority of primary groups called the DUI approach their favorite configuration while many secondary groups preferred the smartphone-only variant.

9.2 Limitations

The methodology that we applied to answer our RQs followed the design-science process (see Section 1.3). It was characterized by viable artifacts that we evaluated in user studies. In the following, we summarize important limitations that should be noted. Limitations specific to single user studies are described in the respective sections.

All user studies in this thesis were conducted as laboratory studies, with the exception of the online evaluation in Section 5.6. The results of these studies could differ when conducted as field experiments. For instance, users could feel more stressed and be more worried about *shoulder-surfing* when interacting with public displays in crowded areas with many passersby. Consequently, interaction times could be shorter than presented in Section 6.5.2, and the public display approach could provide a worse UX than the stand-alone smartphone variant and the DUI approach. However, using smartphones while walking in crowded areas is difficult. We did not evaluate the consequences of this in our experiments. Furthermore, our participants rated recommendations without visiting the recommended POIs, which was another disadvantage of our laboratory studies. Collecting implicit feedback, such as how long users stay at recommended POIs, is more accurate than only asking participants if they could imagine visiting the POIs. We suggest repeating our experiments in public areas with real tourists who are traveling and looking for recommendations.

Another limitation of our work was the limitation to three members per group in our group recommendation studies in Chapters 7 and 8. Couples, for instance, could feel more comfortable with sharing a display and revealing preferences. In our experiment in Section 8.2, three users could easily stand in front of the kiosk system and interact with it simultaneously. Larger groups, however, cannot interact with such a device at the same time. This increases the probability of excluding some group members completely from the decision-making. On the other hand, larger groups could be more willing to split during a trip if no group member has to travel alone. For this purpose, subgroups with similar interests can be identified before a recommendation is made. We suggest verifying these assumptions in experiments with different group sizes.

The participants in our studies were not fully representative. For instance, no participant in our group recommendation studies was younger than 18 years or older than 34 years. The majority of the participants were students or holding a university degree. Hence, our studies miss feedback from people with different backgrounds. For instance, a family traveling with children or elderly could prefer different recommendation strategies that give more weight to the preferences of these group members.

The tourist trip algorithms that we developed in this thesis and used in our experiments are based on the Dijkstra-based algorithm that we presented in Section 5.2. We explained how to integrate our proposed extensions into other types of algorithms, such as GRASP, in Section 5.5.4. Our experiments should be repeated using GRASP and other algorithms, such as algorithms that use photos and LBSNs as input (see Section 2.3.3), and the results compared to our findings.

9.3 Discussion and Future Work

RSs are ubiquitous today. They are an important feature in many online services that allows users to easily identify products, services, or information that best satisfy their needs. It is not surprising that tourism became one very popular domain for RSs. When integrated in tourism applications, they help users to avoid browsing large amounts of travel-related data. Moreover, they can facilitate all steps of travel planning: First of all, RSs identify travel destinations, activities, and POIs that users might miss otherwise, especially when relying on printed media. Then, state-of-the-art tourism applications combine travel items to coherent travel plans and tourist trips. Finally, they support travelers with updated recommendations even when already on the move. It should be stressed that RSs in tourism are not only attractive for tourists. Travel agencies, tour operators, and public authorities can use them to better advertise their offers by directly approaching their target groups.

The tourist trip scenario, as presented in this thesis, is a complex example of a tourism RS. On the one hand, a set of attractive POIs has to be identified. On the other hand, a routing algorithm has to find a route that consists of the most attractive POIs without violating constraints, such as the time available for the trip. Until today, most works have solved the TTDP from a pure mathematical perspective: POIs are vertices in a graph with a fixed profit. A route has to be found that maximizes the sum of the profits without violating predefined constraints. However, such a mathematical approach provides only little benefit in practical applications. The context of a recommendation and personal travel preferences have a large impact on the user satisfaction with recommended tourist trips. A user could be interested in completely different POIs when traveling on a rainy instead of sunny day. Furthermore, tourists often travel in groups which complicates the search for a good tourist trip recommendation, especially when traveling in heterogenous groups or groups characterized by loose relationships.

There is a consensus today that a good recommendation is not only defined by an accurate user rating prediction. Recommendations should also be diverse, serendipitous, and trustworthy, and a good RS should be a pleasure to use. Additional requirements arise when traveling in groups. For instance, recommendations should be fair and cover the interests of all group members. Consequently, in order to determine the real value of RSs in practical applications, they should not only be evaluated in offline tests, but also from a user-centered perspective with real users and groups.

We evaluated our solutions for solving the TTDP for individuals and groups in different user studies and an online evaluation. Our results confirmed that taking into account the user perspective increases satisfaction with tourist trip recommendations. Integrating contextual factors and route attractiveness attributes make tourist trip recommendations more attractive for travelers. The choice of UIs has an impact on how easily users can get familiar with RSs and if they are willing to use them in public spaces. Our user studies revealed additional challenges that developers of practical tourist trip applications have to meet. For instance, social embarrassment and privacy concerns have been identified as main concerns of public display users in published literature [100, 102]; our studies confirmed that users can feel uncomfortable when interacting with large displays in crowded areas and in front of strangers. Furthermore, secondary groups felt significantly less comfortable when sharing a public display and when revealing their preferences to other group members than primary groups. We also showed that many groups can benefit from splitting temporarily during a trip, but in some situations, splitting could become less appropriate. This is why GRSs that implement our suggested Split Group approach have to know if and under which circumstances a group is willing to split. Other solutions could cluster group members into subgroups based on mutual interests before splitting a group. Furthermore, group decisions tend to be often unfair, especially when the group members do not know each other very well. Consequently, GRSs have to promote fairness in group decisions to satisfy all group members.

Solutions to these problems have to be found in future work to improve tourist trip recommendations for individuals and groups. Furthermore, we identified four additional research areas that were only considered marginally in this thesis but are important for the development of practical tourist trip RSs. In the following, we briefly describe of each these four research areas. We recommend that they are investigated in future work and use the findings of this thesis as a basis.

Explanations Many RSs suggest items without describing how the recommendations were generated and why they meet the user's requirements. This lack of transparency can diminish the user's trust in a recommendation and consequently lead to a lower satisfaction with the RS. RSs should therefore explain their recommendations to counteract these problems. Explanations do not only aim at increasing the user's confidence in the recommendations and satisfaction with the RS by providing transparency. Tintarev and Masthoff [199] identified seven goals of explanations in RSs:

- *Transparency*: Explain how the system works
- *Scrutability*: Allow users to tell the system it is wrong
- *Trust*: Increase users' confidence in the system
- *Effectiveness*: Help users make good decisions
- *Persuasiveness*: Convince users to try or buy
- *Efficiency*: Help users make decisions faster
- *Satisfaction*: Increase the ease of usability or enjoyment

Explanations in tourist trip RSs can be provided on POI or trip level. Explanations on POI level can be used to describe why a certain POI appears in the recommended trip.

For example *"This museum is recommended to you because you liked impressionist art in the past"*. This is, furthermore, an example of a content-based style explanation [199]. It is based on features that occurred in previously rated items. In this case, users liked museums with the feature *museum type: impressionism* in the past. Explanations on trip level describe the recommended sequence as a whole. For example *"This trip contains many outdoors activity because of warm temperature and sunny weather"*. This explanation also uses contextual factors (*temperature* and *weather*) that are considered by the RS to describe the recommendation. In a practical tourist trip application, both explanation types can be combined. For instance, an explanation can explain the overall purpose of the recommended trip (e.g., *"The recommendation is a cultural trip because you like to explore art in your holidays"*) and, in addition, explain selected POIs to make individual decisions more transparent (e.g., *"This restaurant is recommended for lunch because it is close to the subsequent museum"*).

Providing good explanations becomes more challenging in GRSs. Some people may feel uncomfortable when an explanation reveals that they are the reason why a certain POI was recommended or not. Therefore, explanations should "balance privacy with transparency and scrutability" [11]. In our scenario, explanations could be used to make the selected group recommendation strategy more transparent but they should avoid to expose single group members. For instance, the GRS INTRIGUE can tailor explanations to subgroups, such as children, instead of individuals (see Section 2.4.3.2). Explanations can not only help us to achieve the seven aforementioned goals in GRSs, but also make group recommendations perceived as fairer by increasing the group members' awareness of their fellow travelers' needs.

Round Trips The formulation of the TTDP that we solved in this thesis required different starting points and destinations. The idea is that a user or group has a starting point in mind (e.g., a hotel) and wants to arrive at a different destination at the end of the trip, such as a restaurant for dinner. However, in practice, travelers might be interested in round trips, that is, the trip ends at the starting point. This is particularly interesting for multi-day trips during which tourists want to receive a trip recommendation starting and ending at their hotel every day.

Most of the published research that we presented in this work solved the TTDP with different starting points and destinations. Only few works researched round trips. DAILYTRIP [58] solves the TOPTW by recommending multiple tours, each starting and ending at the same location. The published works on round trips lack many of the extensions presented in this thesis, such as context-awareness and group recommendations.

Many of our findings can be used in the round trip scenario. However, the Dijkstra-based algorithm introduced in Section 5.2 has to be adapted to enable round trips. The current implementation would return recommendations without any POIs when using the same starting point and destination as it would minimize the distance of the trip to 0. Our suggested extensions for context-aware recommendations and the integration of route attractiveness attributes can be integrated into round trip algorithms. A round trip algorithm solving the TTDP should be implemented in practical applications and evaluated in user studies with real users. For this purpose, our AnyRec framework can be used. It allows adding the new algorithm to an existing RS without having to change other components of the RS. A round trip algorithm can also be used as part of our proposed group recommendation strategies and integrated into our GRS configurations.

Integration of Different Modes of Transportation Most of the existing approaches to solving different variants of the TTDP assume fixed walking times between POIs. In our work, we generated tourist trips for walking tourists only. In practice, it might be appreciated to integrate different modes of transportation to reduce the time spent between two POIs. In the tourist trip scenario, using public transport seems to be an obvious solution. For example, tourists in Paris can take the train to visit the Palace of Versailles in the morning, and then return to Paris to continue sightseeing.

The TDOP is a variant of the OP that can be used to model different modes of transportation in a tourist trip recommendation (see Section 3.1.4). First approaches to solving the TDOP and similar variants have been developed and tested with test instances [152]. Future work could integrate public transport into our context-aware algorithm and also consider the attractiveness of routes when choosing the preferred mode of transportation. Furthermore, the choice should depend on the user's personal preferences and incorporate knowledge of others. For instance, by using CF, locals can recommend less crowded routes during peak times. In a previous project, we developed a hybrid RS for multi-modal route planning that combined CF with knowledge-based recommendations [200]. We showed that such a RS can outperform state-of-the-art route planner software in a user study. Multi-modal route recommendations could be integrated into tourist trip RSs, such as TourRec, to facilitate travel planning for tourists.

Privacy in Tourist Trip Recommender Systems RSs rely on personalized data to come up with suitable recommendations, which raises privacy questions. User may not feel comfortable with sharing personal information, such as the locations they want to visit while traveling. If they agree on sharing such data, they expect them to be protected against unauthorized access.

When solving the TTDP for individuals and groups, privacy questions can arise at different stages of the recommendation process:

- sharing personalized data with the recommendation service,
- *shoulder-surfing* when using public displays, and
- revealing personal preferences to others when using GRSs.

Future work should research solutions to increase the privacy at each of these stages to further increase the user acceptance of tourist trip RSs. We already implemented first approaches within the scope of this thesis. For instance, the TourRec application does not create a user profile on the server and does not store any personalized data in the data tier. Users and their requests are anonymized by assigning an ID to a new user. If a user deletes and re-installs the TourRec Android app, a new ID is created and the RS cannot make a connection to previous user IDs. Some of the participants of our user studies stated that they would feel uncomfortable when interacting with large displays in public displays. Furthermore, some people did not like using public displays in groups, especially when the groups are characterized by rather loose relationships. Our approaches to overcoming this problem included the integration of smartphones into the recommendation process. For instance, groups can use private devices to specify preferences and view the mutual recommendation on the large screen, if necessary. This increases privacy, however, our results showed that further effort is necessary to increase the acceptance of public displays. First approaches to improving privacy of public displays have been developed, for instance, blacking out parts of the visible content. Nevertheless, these ideas have to be adapted to GRSs and evaluated in user studies.

Bibliography

[1] World Travel and Tourism Council. Travel & Tourism Economic Impact 2019, March 2019.

[2] P. Maes. Agents that reduce work and information overload. *Commun. ACM*, 37(7):30–40, July 1994. URL: `http://doi.acm.org/10.1145/176789.176792`, `doi:10.1145/176789.176792`.

[3] D. Gavalas, C. Konstantopoulos, K. Mastakas, and G. Pantziou. Mobile recommender systems in tourism. *Journal of Network and Computer Applications*, 39:319–333, 2014.

[4] M. Xie, H. Yin, H. Wang, F. Xu, W. Chen, and S. Wang. Learning graph-based poi embedding for location-based recommendation. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, pages 15–24, New York, NY, USA, 2016. ACM. URL: `http://doi.acm.org/10.1145/2983323.2983711`, `doi:10.1145/2983323.2983711`.

[5] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi. Understanding individual human mobility patterns. *nature*, 453(7196):779, 2008.

[6] P. Vansteenwegen and D. Van Oudheusden. The mobile tourist guide: an or opportunity. *OR Insight*, 20(3):21–27, 2007.

[7] K. Sylejmani, J. Dorn, and N. Musliu. Planning the trip itinerary for tourist groups. *Information Technology & Tourism*, 17(3):275–314, Sep 2017. URL: `https://doi.org/10.1007/s40558-017-0080-9`, `doi:10.1007/s40558-017-0080-9`.

[8] G. Adomavicius and A. Tuzhilin. Context-aware recommender systems. In F. Ricci, L. Rokach, and B. Shapira, editors, *Recommender Systems Handbook*, pages 191–226. Springer US, Boston, MA, 2015. URL: `http://dx.doi.org/10.1007/978-1-4899-7637-6_6`, `doi:10.1007/978-1-4899-7637-6_6`.

[9] S. R. de M. Queiroz and F. de A.T. de Carvalho. Making collaborative group recommendations based on modal symbolic data. In A. L. C. Bazzan and S. Labidi, editors, *Advances in Artificial Intelligence – SBIA 2004: 17th Brazilian Symposium on Artificial Intelligence, Sao Luis, Maranhao, Brazil, September 29-Ocotber 1, 2004. Proceedings*, pages 307–316. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. URL: `http://dx.doi.org/10.1007/978-3-540-28645-5_31`, `doi:10.1007/978-3-540-28645-5_31`.

[10] A. Jameson and B. Smyth. Recommendation to groups. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The Adaptive Web*, pages 596–627. Springer, Berlin, Heidelberg, 2007. URL: `http://dl.acm.org/citation.cfm?id=1768197.1768221`.

[11] J. Masthoff. Group recommender systems: Aggregation, satisfaction and group attributes. In F. Ricci, L. Rokach, and B. Shapira, editors, *Recommender Systems Handbook*, pages 743–776. Springer US, Boston, MA, 2015. URL: `http://dx.doi.org/10.1007/978-1-4899-7637-6_22`, `doi:10.1007/978-1-4899-7637-6_22`.

[12] A. R. Hevner, S. T. March, J. Park, and S. Ram. Design science in information systems research. *MIS Q.*, 28(1):75–105, March 2004. URL: `http://dl.acm.org/citation.cfm?id=2017212.2017217`.

[13] T. Asikis and G. Lekakos. Operations research and recommender systems. In S. Yamamoto, editor, *Human Interface and the Management of Information. Information and Knowledge in Applications and Services*, pages 579–589, Cham, 2014. Springer International Publishing.

[14] D. Gavalas, C. Konstantopoulos, K. Mastakas, and G. Pantziou. A survey on algorithmic approaches for solving tourist trip design problems. *Journal of Heuristics*, 20(3):291–328, June 2014. URL: `http://dx.doi.org/10.1007/s10732-014-9242-5`, `doi:10.1007/s10732-014-9242-5`.

[15] S. M. McNee, J. Riedl, and J. A. Konstan. Being accurate is not enough: How accuracy metrics have hurt recommender systems. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '06, pages 1097–1101, New York, NY, USA, 2006. ACM. URL: `http://doi.acm.org/10.1145/1125451.1125659`, `doi:10.1145/1125451.1125659`.

[16] A. Anagnostopoulos, R. Atassi, L. Becchetti, A. Fazzone, and F. Silvestri. Tour recommendation for groups. *Data Mining and Knowledge Discovery*, 31(5):1157–1188, Sep 2017. URL: `https://doi.org/10.1007/s10618-016-0477-7`, `doi:10.1007/s10618-016-0477-7`.

[17] F. Ricci, L. Rokach, and B. Shapira. Recommender systems: Introduction and challenges. In F. Ricci, L. Rokach, and B. Shapira, editors, *Recommender Systems Handbook*, pages 1–34. Springer US, Boston, MA, 2015. URL: `http://dx.doi.org/10.1007/978-1-4899-7637-6_1`, `doi:10.1007/978-1-4899-7637-6_1`.

[18] J. B. Schafer, J. Konstan, and J. Riedl. Recommender systems in e-commerce. In *Proceedings of the 1st ACM Conference on Electronic Commerce*, EC '99, pages 158–166, New York, NY, USA, 1999. ACM. URL: `http://doi.acm.org/10.1145/336992.337035`, `doi:10.1145/336992.337035`.

[19] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. Grouplens: Applying collaborative filtering to usenet news. *Commun. ACM*,

40(3):77–87, March 1997. URL: http://doi.acm.org/10.1145/245108.245126, doi:10.1145/245108.245126.

[20] J. Borràs, A. Moreno, and A. Valls. Intelligent tourism recommender systems: A survey. *Expert Systems with Applications*, 41(16):7370 – 7389, 2014. URL: http://www.sciencedirect.com/science/article/pii/S0957417414003431, doi:http://dx.doi.org/10.1016/j.eswa.2014.06.007.

[21] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. Collaborative Filtering Recommender Systems. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The Adaptive Web: Methods and Strategies of Web Personalization*, chapter Collaborative Filtering Recommender Systems, pages 291–324. Springer-Verlag, Berlin, Heidelberg, 2007. URL: http://dl.acm.org/citation.cfm?id=1768197.1768208.

[22] E. I. Sparling and S. Sen. Rating: How difficult is it? In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 149–156, New York, NY, USA, 2011. ACM. URL: http://doi.acm.org/10.1145/2043932.2043961, doi:10.1145/2043932.2043961.

[23] R. Burke. Hybrid Web Recommender Systems. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The Adaptive Web: Methods and Strategies of Web Personalization*, pages 377–408. Springer-Verlag, Berlin, Heidelberg, 2007. URL: http://dl.acm.org/citation.cfm?id=1768197.1768211.

[24] M. Balabanović and Y. Shoham. Fab: Content-based, collaborative recommendation. *Commun. ACM*, 40(3):66–72, March 1997. URL: http://doi.acm.org/10.1145/245108.245124, doi:10.1145/245108.245124.

[25] A. Felfernig, L. Boratto, M. Stettinger, and M. Tkalčič. *Algorithms for Group Recommendation*, pages 27–58. Springer International Publishing, Cham, 2018. URL: https://doi.org/10.1007/978-3-319-75067-5_2, doi:10.1007/978-3-319-75067-5_2.

[26] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749, June 2005. URL: https://doi.org/10.1109/TKDE.2005.99, doi:10.1109/TKDE.2005.99.

[27] M. de Gemmis, P. Lops, C. Musto, F. Narducci, and G. Semeraro. Semantics-aware content-based recommender systems. In F. Ricci, L. Rokach, and B. Shapira, editors, *Recommender Systems Handbook*, pages 119–159. Springer US, Boston, MA, 2015. URL: https://doi.org/10.1007/978-1-4899-7637-6_4, doi:10.1007/978-1-4899-7637-6_4.

[28] B. Smyth. Case-based recommendation. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The Adaptive Web*, pages 342–376. Springer, Berlin, Heidelberg, 2007.

[29] J. Cohen. *Statistical power analysis for the behavioral sciences.* Hillsdale, NJ: Erlbaum., New York, 2nd edition, 1988.

[30] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 285–295, New York, NY, USA, 2001. ACM. URL: `http://doi.acm.org.eaccess.ub.tum.de/10.1145/371920.372071`, `doi:10.1145/371920.372071`.

[31] A. Felfernig, G. Friedrich, D. Jannach, and M. Zanker. Constraint-based recommender systems. In F. Ricci, L. Rokach, and B. Shapira, editors, *Recommender Systems Handbook*, pages 161–190. Springer US, Boston, MA, 2015. URL: `https://doi.org/10.1007/978-1-4899-7637-6_5`, `doi:10.1007/978-1-4899-7637-6_5`.

[32] R. D. Burke, K. J. Hammond, and B. C. Young. Knowledge-based navigation of complex information spaces. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 1*, AAAI'96, pages 462–468. AAAI Press, 1996. URL: `http://dl.acm.org/citation.cfm?id=1892875.1892944`.

[33] R. Burke. Knowledge-Based Recommender Systems. In *Encyclopedia of Library and Information Systems*. Marcel Dekker, 2000.

[34] P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *Eighteenth National Conference on Artificial Intelligence*, pages 187–192, Menlo Park, CA, USA, 2002. American Association for Artificial Intelligence. URL: `http://dl.acm.org/citation.cfm?id=777092.777124`.

[35] L. Baltrunas, B. Ludwig, S. Peer, and F. Ricci. Context relevance assessment and exploitation in mobile recommender systems. *Personal Ubiquitous Comput.*, 16(5):507–526, June 2012. URL: `http://dx.doi.org/10.1007/s00779-011-0417-x`, `doi:10.1007/s00779-011-0417-x`.

[36] A. K. Dey. Understanding and using context. *Personal Ubiquitous Comput.*, 5(1):4–7, January 2001. URL: `http://dx.doi.org.eaccess.ub.tum.de/10.1007/s007790170019`, `doi:10.1007/s007790170019`.

[37] W. Woerndl, J. Huebner, R. Bader, and D. Gallego-Vico. A model for proactivity in mobile, context-aware recommender systems. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 273–276, New York, NY, USA, 2011. ACM. URL: `http://doi.acm.org/10.1145/2043932.2043981`, `doi:10.1145/2043932.2043981`.

[38] B. Lamche. *Improving the User Experience in Mobile Recommender Systems.* Dissertation, Technische Universität München, München, 2016.

[39] M. van Setten, S. Pokraev, and J. Koolwaaij. Context-aware recommendations in the mobile tourist application compass. In P. M. E. De Bra and W. Nejdl, editors, *Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 235–244. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[40] K. McCarthy, L. McGinty, B. Smyth, and M. Salamó. The needs of the many: A case-based group recommender system. In *Proceedings of the 8th European Conference on Advances in Case-Based Reasoning*, ECCBR'06, pages 196–210, Berlin, Heidelberg, 2006. Springer-Verlag. URL: `http://dx.doi.org.eaccess.ub.tum.de/10.1007/11805816_16`, `doi:10.1007/11805816_16`.

[41] W. Wörndl, A. Hefele, and D. Herzog. Recommending a sequence of interesting places for tourist trips. *Information Technology & Tourism*, 17(1):31–54, 2017. URL: `http://dx.doi.org/10.1007/s40558-017-0076-5`, `doi:10.1007/s40558-017-0076-5`.

[42] D. Herzog, L. W. Dietz, and W. Wörndl. Tourist trip recommendations – foundations, state of the art, and challenges. In *Personalized Human-Computer Interaction*, pages 159–182. De Gruyter, sep 2019. `doi:10.1515/9783110552485-006`.

[43] K. Cheverst, N. Davies, K. Mitchell, A. Friday, and C. Efstratiou. Developing a context-aware electronic tourist guide: Some issues and experiences. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '00, pages 17–24, New York, NY, USA, 2000. ACM. URL: `http://doi.acm.org/10.1145/332040.332047`, `doi:10.1145/332040.332047`.

[44] M. Braunhofer, M. Elahi, M. Ge, and F. Ricci. Sts: Design of weather-aware mobile recommender systems in tourism. In *In Proceedings of the 1st Workshop on AI*HCI: Intelligent User Interfaces (AI*HCI 2013)*, 2013.

[45] F. Ricci and Q. N. Nguyen. Mobyrek: A conversational recommender system for on-the-move travelers. In D. R. Fesenmaier, K. W. Wöber, and H. Werthner, editors, *Destination recommendation systems: behavioural foundations and applications*, pages 281–294. CABI, 2006. `doi:10.1079/9780851990231.0281`.

[46] M. Xie, L. V. S. Lakshmanan, and P. T. Wood. Comprec-trip: A composite recommendation system for travel planning. In *2011 IEEE 27th International Conference on Data Engineering*, pages 1352–1355, April 2011. `doi:10.1109/ICDE.2011.5767954`.

[47] I. Benouaret and D. Lenne. A package recommendation framework for trip planning activities. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, pages 203–206, New York, NY, USA, 2016. ACM. URL: `http://doi.acm.org/10.1145/2959100.2959183`, `doi:10.1145/2959100.2959183`.

[48] R. Baral and T. Li. Maps: A multi aspect personalized poi recommender system. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys

'16, pages 281–284, New York, NY, USA, 2016. ACM. URL: http://doi.acm.org/10.1145/2959100.2959187, doi:10.1145/2959100.2959187.

[49] M. Lenz. Cabata: Case-based reasoning for holiday planning. In *Proceedings of the International Conference on Information and Communications Technologies in Tourism*, pages 126–132, Secaucus, NJ, USA, 1994. Springer-Verlag New York, Inc. URL: http://dl.acm.org/citation.cfm?id=184620.184769.

[50] M. Lenz. Imtas: Intelligent multimedia travel agent system. In S. Klein, B. Schmid, A. M. Tjoa, and H. Werthner, editors, *Information and Communication Technologies in Tourism*, pages 11–17, Vienna, 1996. Springer Vienna.

[51] F. Ricci and H. Werthner. Case base querying for travel planning recommendation. *Information Technology & Tourism*, 4:215–226, 01 2001.

[52] F. Ricci, D. R. Fesenmaier, N. Mirzadeh, H. Rumetshofer, E. Schaumlechner, A. Venturini, K. W. Wöber, and A. H. Zins. Dietorecs: a case-based travel advisory system. In D. R. Fesenmaier, K. W. Wöber, and H. Werthner, editors, *Destination recommendation systems: behavioural foundations and applications*, pages 227–239. CABI, 2006.

[53] A. Venturini and F. Ricci. Applying trip@dvice recommendation technology to www.visiteurope.com. In *Proceedings of the 2006 Conference on ECAI 2006: 17th European Conference on Artificial Intelligence August 29 – September 1, 2006, Riva Del Garda, Italy*, pages 607–611, Amsterdam, The Netherlands, The Netherlands, 2006. IOS Press. URL: http://dl.acm.org/citation.cfm?id=1567016.1567148.

[54] F. Ricci. Travel recommender systems. *IEEE Intelligent Systems*, pages 55–57, 2002.

[55] T. Mahmood, F. Ricci, and A. Venturini. Improving recommendation effectiveness: Adapting a dialogue strategy in online travel planning. *Information Technology & Tourism*, 11(4):285–302, 2009.

[56] L. Sebastia, I. Garcia, E. Onaindia, and C. Guzman. e-tourism: A tourist recommendation and planning application. In *Proceedings of the 2008 20th IEEE International Conference on Tools with Artificial Intelligence - Volume 02*, ICTAI '08, pages 89–96, Washington, DC, USA, 2008. IEEE Computer Society. URL: https://doi.org/10.1109/ICTAI.2008.18, doi:10.1109/ICTAI.2008.18.

[57] P. Vansteenwegen, W. Souffriau, G. V. Berghe, and D. V. Oudheusden. The city trip planner. *Expert Syst. Appl.*, 38(6):6540–6546, June 2011. URL: http://dx.doi.org/10.1016/j.eswa.2010.11.085, doi:10.1016/j.eswa.2010.11.085.

[58] D. Gavalas, M. Kenteris, C. Konstantopoulos, and G. Pantziou. Web application for recommending personalised mobile tourist routes. *Software, IET*, 6(4):313–322, 2012.

[59] B. Rodríguez, J. Molina, F. Pérez, and R. Caballero. Interactive design of personalised tourism routes. *Tourism Management*, 33(4):926 – 940, 2012. URL: `http://www.sciencedirect.com/science/article/pii/S0261517711001993`, `doi:http://dx.doi.org/10.1016/j.tourman.2011.09.014`.

[60] Y. Tanahashi and K.-L. Ma. Onmyway: A task-oriented visualization and interface design for planning road trip itinerary. In *Proceedings of the 2013 International Conference on Cyberworlds*, CW '13, pages 199–205, Washington, DC, USA, 2013. IEEE Computer Society. URL: `http://dx.doi.org/10.1109/CW.2013.16`, `doi:10.1109/CW.2013.16`.

[61] I. Refanidis, C. Emmanouilidis, I. Sakellariou, A. Alexiadis, R.-A. Koutsiamanis, K. Agnantis, A. Tasidou, F. Kokkoras, and P. S. Efraimidis. myvisitplannergr: Personalized itinerary planning system for tourism. In A. Likas, K. Blekas, and D. Kalles, editors, *Artificial Intelligence: Methods and Applications*, pages 615–629, Cham, 2014. Springer International Publishing.

[62] D. Gavalas, V. Kasapakis, C. Konstantopoulos, G. Pantziou, N. Vathis, and C. Zaroliagis. The ecompass multimodal tourist tour planner. *Expert Syst. Appl.*, 42(21):7303–7316, November 2015. URL: `http://dx.doi.org/10.1016/j.eswa.2015.05.046`, `doi:10.1016/j.eswa.2015.05.046`.

[63] D. Gavalas, V. Kasapakis, C. Konstantopoulos, G. Pantziou, and N. Vathis. Scenic route planning for tourists. *Personal and Ubiquitous Computing*, pages 1–19, 2016. URL: `http://dx.doi.org/10.1007/s00779-016-0971-3`, `doi:10.1007/s00779-016-0971-3`.

[64] T. Arentze, A. Kemperman, and P. Aksenov. Estimating a latent-class user model for travel recommender systems. *Information Technology & Tourism*, Feb 2018. URL: `https://doi.org/10.1007/s40558-018-0105-z`, `doi:10.1007/s40558-018-0105-z`.

[65] M. De Choudhury, M. Feldman, S. Amer-Yahia, N. Golbandi, R. Lempel, and C. Yu. Automatic construction of travel itineraries using social breadcrumbs. In *Proceedings of the 21st ACM Conference on Hypertext and Hypermedia*, HT '10, pages 35–44, New York, NY, USA, 2010. ACM. URL: `http://doi.acm.org/10.1145/1810617.1810626`, `doi:10.1145/1810617.1810626`.

[66] X. Lu, C. Wang, J.-M. Yang, Y. Pang, and L. Zhang. Photo2trip: Generating travel routes from geo-tagged photos for trip planning. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pages 143–152, New York, NY, USA, 2010. ACM. URL: `http://doi.acm.org.eaccess.ub.tum.de/10.1145/1873951.1873972`, `doi:10.1145/1873951.1873972`.

[67] I. Brilhante, J. A. Macedo, F. M. Nardini, R. Perego, and C. Renso. Where shall we go today?: Planning touristic tours with tripbuilder. In *Proceedings of the*

22Nd ACM International Conference on Information & Knowledge Management, CIKM '13, pages 757–762, New York, NY, USA, 2013. ACM. URL: http://doi.acm.org/10.1145/2505515.2505643, doi:10.1145/2505515.2505643.

[68] K. H. Lim, J. Chan, C. Leckie, and S. Karunasekera. Personalized trip recommendation for tourists based on user interests, points of interest visit durations and visit recency. *Knowledge and Information Systems*, 54(2):375–406, Feb 2018. URL: https://doi.org/10.1007/s10115-017-1056-y, doi:10.1007/s10115-017-1056-y.

[69] D. Quercia, R. Schifanella, and L. M. Aiello. The shortest path to happiness: Recommending beautiful, quiet, and happy routes in the city. In *Proceedings of the 25th ACM Conference on Hypertext and Social Media*, HT '14, pages 116–125, New York, NY, USA, 2014. ACM. URL: http://doi.acm.org/10.1145/2631775.2631799, doi:10.1145/2631775.2631799.

[70] Z. Yu, H. Xu, Z. Yang, and B. Guo. Personalized travel package with multi-point-of-interest recommendation based on crowdsourced user footprints. *IEEE Transactions on Human-Machine Systems*, 46(1):151–158, Feb 2016. doi:10.1109/THMS.2015.2446953.

[71] E. Haris and K. H. Gan. Mining graphs from travel blogs: a review in the context of tour planning. *Information Technology & Tourism*, 17(4):429–453, Dec 2017. URL: https://doi.org/10.1007/s40558-017-0095-2, doi:10.1007/s40558-017-0095-2.

[72] D. R. Forsyth. *Group dynamics*. Cengage Learning, 2018.

[73] M. Shaw. *Group dynamics, the psychology of small group behavior*. McGraw-Hill series in psychology. McGraw-Hill, 1971.

[74] B. Mullen and G. Goethals. *Theories of group behavior*. Springer series in social psychology. Springer-Verlag, 1987. URL: https://books.google.de/books?id=sxlHAAAAMAAJ.

[75] H. Bonner. *Group dynamics: principles and applications*. A psychology series. Ronald Press, 1959. URL: https://books.google.de/books?id=RAFXAAAAYAAJ.

[76] J. McGrath. *Groups: Interaction and Performance*. Prentice-Hall, 1984. URL: https://books.google.de/books?id=4pzZAAAAMAAJ.

[77] I. Steiner. *Group Processes and Productivity*. Cell Biology. Academic Press, 1972. URL: https://books.google.de/books?id=20S3AAAAIAAJ.

[78] J. Masthoff. Group modeling: Selecting a sequence of television items to suit a group of viewers. *User Modeling and User-Adapted Interaction*, 14(1):37–85, February 2004. URL: http://dx.doi.org/10.1023/B:USER.0000010138.79319.fd, doi:10.1023/B:USER.0000010138.79319.fd.

[79] L. M. de Campos, J. M. Fernández-Luna, J. F. Huete, and M. A. Rueda-Morales. Managing uncertainty in group recommending processes. *User Modeling and User-Adapted Interaction*, 19(3):207–242, Aug 2009. URL: `https://doi.org/10.1007/s11257-008-9061-1`, `doi:10.1007/s11257-008-9061-1`.

[80] A. Jameson. More than the sum of its members: Challenges for group recommender systems. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '04, pages 48–54, New York, NY, USA, 2004. ACM. URL: `http://doi.acm.org.eaccess.ub.tum.de/10.1145/989863.989869`, `doi:10.1145/989863.989869`.

[81] C. Senot, D. Kostadinov, M. Bouzid, J. Picault, A. Aghasaryan, and C. Bernier. Analysis of strategies for building group profiles. In *Proceedings of the 18th International Conference on User Modeling, Adaptation, and Personalization*, UMAP'10, pages 40–51, Berlin, Heidelberg, 2010. Springer-Verlag. URL: `http://dx.doi.org/10.1007/978-3-642-13470-8_6`, `doi:10.1007/978-3-642-13470-8_6`.

[82] J. F. McCarthy and T. D. Anagnost. Musicfx: An arbiter of group preferences for computer supported collaborative workouts. In *Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work*, CSCW '98, pages 363–372, New York, NY, USA, 1998. ACM. URL: `http://doi.acm.org.eaccess.ub.tum.de/10.1145/289444.289511`, `doi:10.1145/289444.289511`.

[83] H. Lieberman, N. W. Van Dyke, and A. S. Vivacqua. Let's browse: A collaborative web browsing agent. In *Proceedings of the 4th International Conference on Intelligent User Interfaces*, IUI '99, pages 65–68, New York, NY, USA, 1999. ACM. URL: `http://doi.acm.org/10.1145/291080.291092`, `doi:10.1145/291080.291092`.

[84] M. O'Connor, D. Cosley, J. A. Konstan, and J. Riedl. Polylens: A recommender system for groups of users. In *Proceedings of the Seventh Conference on European Conference on Computer Supported Cooperative Work*, ECSCW'01, pages 199–218, Norwell, MA, USA, 2001. Kluwer Academic Publishers. URL: `http://dl.acm.org.eaccess.ub.tum.de/citation.cfm?id=1241867.1241878`.

[85] L. Quijano-Sanchez, J. A. Recio-Garcia, B. Diaz-Agudo, and G. Jimenez-Diaz. Social factors in group recommender systems. *ACM Trans. Intell. Syst. Technol.*, 4(1):8:1–8:30, February 2013. URL: `http://doi.acm.org/10.1145/2414425.2414433`, `doi:10.1145/2414425.2414433`.

[86] G. Popescu. Group recommender systems as a voting problem. In *Proceedings of the 5th International Conference on Online Communities and Social Computing*, OCSC'13, pages 412–421, Berlin, Heidelberg, 2013. Springer-Verlag. URL: `http://dx.doi.org/10.1007/978-3-642-39371-6_46`, `doi:10.1007/978-3-642-39371-6_46`.

[87] L. Ardissono, A. Goy, G. Petrone, M. Segnan, and P. Torasso. Tailoring the recommendation of tourist information to heterogeneous user groups. In *Revised*

Papers from the Nternational Workshops OHS-7, SC-3, and AH-3 on Hypermedia: Openness, Structural Awareness, and Adaptivity, pages 280–295, London, UK, UK, 2002. Springer-Verlag. URL: `http://dl.acm.org/citation.cfm?id=648105.746696`.

[88] J. F. McCarthy. Pocket restaurantfinder: A situated recommender system for groups. In *Proceeding of the Workshop on Mobile Ad-Hoc Communication at the 2002 ACM Conference on Human Factors in Computer Systems*, 2002.

[89] F. Guzzi, F. Ricci, and R. Burke. Interactive multi-party critiquing for group recommendation. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 265–268, New York, NY, USA, 2011. ACM. URL: `http://doi.acm.org/10.1145/2043932.2043980`, `doi:10.1145/2043932.2043980`.

[90] T. N. Nguyen and F. Ricci. A chat-based group recommender system for tourism. In R. Schegg and B. Stangl, editors, *Information and Communication Technologies in Tourism 2017*, pages 17–30, Cham, 2017. Springer International Publishing.

[91] I. Benouaret and D. Lenne. A package-to-group recommendation framework. In A. Hameurlain, R. Wagner, D. Benslimane, E. Damiani, and W. I. Grosky, editors, *Transactions on Large-Scale Data- and Knowledge-Centered Systems XXXIX: Special Issue on Database- and Expert-Systems Applications*, pages 43–66. Springer Berlin Heidelberg, Berlin, Heidelberg, 2018. URL: `https://doi.org/10.1007/978-3-662-58415-6_2`, `doi:10.1007/978-3-662-58415-6_2`.

[92] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan. Collaborative filtering recommender systems. *Found. Trends Hum.-Comput. Interact.*, 4(2):81–173, February 2011. URL: `http://dx.doi.org/10.1561/1100000009`, `doi:10.1561/1100000009`.

[93] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, CSCW '94, pages 175–186, New York, NY, USA, 1994. ACM. URL: `http://doi.acm.org.eaccess.ub.tum.de/10.1145/192844.192905`, `doi:10.1145/192844.192905`.

[94] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2):133–151, Jul 2001. URL: `https://doi.org/10.1023/A:1011419012209`, `doi:10.1023/A:1011419012209`.

[95] F. Ricci. Mobile recommender systems. *Information Technology & Tourism*, (3):205–231, 2010. URL: `https://www.ingentaconnect.com/content/cog/itt/2010/00000012/00000003/art00002`, `doi:doi:10.3727/109830511X12978702284390`.

[96] M. Kenteris, D. Gavalas, and D. Economou. Electronic mobile guides: A survey. *Personal Ubiquitous Comput.*, 15(1):97–111, January 2011. URL: http://dx.doi.org/10.1007/s00779-010-0295-7, doi:10.1007/s00779-010-0295-7.

[97] C. Parker, M. Tomitsch, and J. Kay. Does the public still look at public displays?: A field observation of public displays in the wild. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(2):73:1–73:24, July 2018. URL: http://doi.acm.org.eaccess.ub.tum.de/10.1145/3214276, doi:10.1145/3214276.

[98] P. Peltonen, E. Kurvinen, A. Salovaara, G. Jacucci, T. Ilmonen, J. Evans, A. Oulasvirta, and P. Saarikko. It's mine, don't touch!: Interactions at a large multi-touch display in a city centre. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 1285–1294, New York, NY, USA, 2008. ACM. URL: http://doi.acm.org/10.1145/1357054.1357255, doi:10.1145/1357054.1357255.

[99] J. Müller, F. Alt, D. Michelis, and A. Schmidt. Requirements and design space for interactive public displays. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pages 1285–1294, New York, NY, USA, 2010. ACM. URL: http://doi.acm.org.eaccess.ub.tum.de/10.1145/1873951.1874203, doi:10.1145/1873951.1874203.

[100] H. Brignull and Y. Rogers. Enticing people to interact with large public displays in public spaces. In *Human-Computer Interaction INTERACT '03: IFIP TC13 International Conference on Human-Computer Interaction, 1st-5th September 2003, Zurich, Switzerland*, 2003.

[101] D. Michelis and J. Müller. The audience funnel: Observations of gesture based interaction with multiple large displays in a city center. *International Journal of Human–Computer Interaction*, 27(6):562–579, 2011. URL: https://doi.org/10.1080/10447318.2011.555299, arXiv:https://doi.org/10.1080/10447318.2011.555299, doi:10.1080/10447318.2011.555299.

[102] F. Alt, T. Kubitza, D. Bial, F. Zaidan, M. Ortel, B. Zurmaar, T. Lewen, A. S. Shirazi, and A. Schmidt. Digifieds: Insights into deploying digital public notice areas in the wild. In *Proceedings of the 10th International Conference on Mobile and Ubiquitous Multimedia*, MUM '11, pages 165–174, New York, NY, USA, 2011. ACM. URL: http://doi.acm.org.eaccess.ub.tum.de/10.1145/2107596.2107618, doi:10.1145/2107596.2107618.

[103] F. Brudy, D. Ledo, S. Greenberg, and A. Butz. Is anyone looking? mitigating shoulder surfing on public displays through awareness and protection. In *Proceedings of The International Symposium on Pervasive Displays*, PerDis '14, pages 1:1–1:6, New York, NY, USA, 2014. ACM. URL: http://doi.acm.org.eaccess.ub.tum.de/10.1145/2611009.2611028, doi:10.1145/2611009.2611028.

[104] F. Alt, A. S. Shirazi, T. Kubitza, and A. Schmidt. Interaction techniques for creating and exchanging content with public displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 1709–1718, New York, NY, USA, 2013. ACM. URL: `http://doi.acm.org.eaccess.ub.tum.de/10.1145/2470654.2466226`, `doi:10.1145/2470654.2466226`.

[105] E. M. Huang, A. Koster, and J. Borchers. Overcoming assumptions and uncovering practices: When does the public really look at public displays? In *Proceedings of the 6th International Conference on Pervasive Computing*, Pervasive '08, pages 228–243, Berlin, Heidelberg, 2008. Springer-Verlag. URL: `http://dx.doi.org/10.1007/978-3-540-79576-6_14`, `doi:10.1007/978-3-540-79576-6_14`.

[106] F. Alt, S. Schneegaß, A. Schmidt, J. Müller, and N. Memarovic. How to evaluate public displays. In *Proceedings of the 2012 International Symposium on Pervasive Displays*, PerDis '12, pages 17:1–17:6, New York, NY, USA, 2012. ACM. URL: `http://doi.acm.org/10.1145/2307798.2307815`, `doi:10.1145/2307798.2307815`.

[107] N. Elmqvist. *Distributed User Interfaces: State of the Art*, pages 1–12. Springer London, London, 2011. URL: `https://doi.org/10.1007/978-1-4471-2271-5_1`, `doi:10.1007/978-1-4471-2271-5_1`.

[108] J. Vanderdonckt. Distributed user interfaces: How to distribute user interfaceelements across users, platforms, and environments. In *Proc. of XIth Congreso Internacional de Interacción Persona-Ordenador Interacción'2010*, pages 3–14, Valencia, September 2010.

[109] D. Grolaux, P. Van Roy, and J. Vanderdonckt. Migratable user interfaces: beyond migratory interfaces. In *The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004.*, pages 422–430, Aug 2004. `doi:10.1109/MOBIQ.2004.1331749`.

[110] M. Kenteris, D. Gavalas, and D. Economou. An innovative mobile electronic tourist guide application. *Personal Ubiquitous Comput.*, 13(2):103–118, February 2009. URL: `http://dx.doi.org/10.1007/s00779-007-0191-y`, `doi:10.1007/s00779-007-0191-y`.

[111] D. Schmidt, J. Seifert, E. Rukzio, and H. Gellersen. A cross-device interaction style for mobiles and surfaces. In *Proceedings of the Designing Interactive Systems Conference*, DIS '12, pages 318–327, New York, NY, USA, 2012. ACM. URL: `http://doi.acm.org/10.1145/2317956.2318005`, `doi:10.1145/2317956.2318005`.

[112] J. Vepsäläinen, A. Di Rienzo, M. Nelimarkka, J. A. Ojala, P. Savolainen, K. Kuikkaniemi, S. Tarkoma, and G. Jacucci. Personal device as a controller for interactive surfaces: Usability and utility of different connection methods. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces*, ITS '15, pages 201–204, New York, NY,

USA, 2015. ACM. URL: http://doi.acm.org.eaccess.ub.tum.de/10.1145/2817721.2817745, doi:10.1145/2817721.2817745.

[113] N. Davies, A. Friday, P. Newman, S. Rutlidge, and O. Storz. Using bluetooth device names to support interaction in smart environments. In *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services*, MobiSys '09, pages 151–164, New York, NY, USA, 2009. ACM. URL: http://doi.acm.org.eaccess.ub.tum.de/10.1145/1555816.1555832, doi:10.1145/1555816.1555832.

[114] S. Boring, D. Baur, A. Butz, S. Gustafson, and P. Baudisch. Touch projector: Mobile interaction through video. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 2287–2296, New York, NY, USA, 2010. ACM. URL: http://doi.acm.org/10.1145/1753326.1753671, doi:10.1145/1753326.1753671.

[115] W. Abdrabo and W. Wörndl. Direc: A distributed user interface video recommender. In *Proceedings of the Joint Workshop on Interfaces and Human Decision Making for Recommender Systems co-located with ACM Conference on Recommender Systems (RecSys 2016), Boston, MA, USA, September 16, 2016.*, pages 44–47, 2016.

[116] A. Gunawardana and G. Shani. Evaluating recommender systems. In F. Ricci, L. Rokach, and B. Shapira, editors, *Recommender Systems Handbook*, pages 265–308. Springer US, Boston, MA, 2015. URL: https://doi.org/10.1007/978-1-4899-7637-6_8, doi:10.1007/978-1-4899-7637-6_8.

[117] B. P. Knijnenburg and M. C. Willemsen. Evaluating recommender systems with user experiments. In F. Ricci, L. Rokach, and B. Shapira, editors, *Recommender Systems Handbook*, pages 309–352. Springer US, Boston, MA, 2015. URL: http://dx.doi.org/10.1007/978-1-4899-7637-6_9, doi:10.1007/978-1-4899-7637-6_9.

[118] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, January 2004. URL: http://doi.acm.org/10.1145/963770.963772, doi:10.1145/963770.963772.

[119] X. Amatriain and J. M. Pujol. Data mining methods for recommender systems. In F. Ricci, L. Rokach, and B. Shapira, editors, *Recommender Systems Handbook*, pages 227–262. Springer US, Boston, MA, 2015. URL: https://doi.org/10.1007/978-1-4899-7637-6_7, doi:10.1007/978-1-4899-7637-6_7.

[120] P. Pu, L. Chen, and R. Hu. A user-centric evaluation framework for recommender systems. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, RecSys '11, pages 157–164, New York, NY, USA, 2011. ACM. URL: http://doi.acm.org/10.1145/2043932.2043962, doi:10.1145/2043932.2043962.

[121] J. Brooke. SUS-A quick and dirty usability scale. *Usability evaluation in industry*, pages 189–194, 1996.

[122] J. Sauro. *A Practical Guide to the System Usability Scale: Background, Benchmarks & Best Practices*. CreateSpace Independent Publishing Platform, 2011. URL: https://books.google.de/books?id=BLOkKQEACAAJ.

[123] B. Laugwitz, T. Held, and M. Schrepp. Construction and evaluation of a user experience questionnaire. In A. Holzinger, editor, *HCI and Usability for Education and Work*, pages 63–76, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[124] B. P. Knijnenburg, M. C. Willemsen, Z. Gantner, H. Soncu, and C. Newell. Explaining the user experience of recommender systems. *User Modeling and User-Adapted Interaction*, 22(4-5):441–504, October 2012. URL: http://dx.doi.org/10.1007/s11257-011-9118-4, doi:10.1007/s11257-011-9118-4.

[125] P. Vansteenwegen, W. Souffriau, and D. Van Oudheusden. The orienteering problem: A survey. *European Journal of Operational Research*, 209(1):1–10, 2011.

[126] A. Gunawan, H. C. Lau, and P. Vansteenwegen. Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, 2016. URL: http://www.sciencedirect.com/science/article/pii/S037722171630296X, doi:http://dx.doi.org/10.1016/j.ejor.2016.04.059.

[127] T. Tsiligirides. Heuristic methods applied to orienteering. *Journal of the Operational Research Society*, pages 797–809, 1984.

[128] B. L. Golden, L. Levy, and R. Vohra. The orienteering problem. *Naval Research Logistics (NRL)*, 34(3):307–318, 1987. URL: http://dx.doi.org/10.1002/1520-6750(198706)34:3<307::AID-NAV3220340302>3.0.CO;2-D, doi:10.1002/1520-6750(198706)34:3<307::AID-NAV3220340302>3.0.CO;2-D.

[129] A. Wren and A. Holliday. Computer scheduling of vehicles from one or more depots to a number of delivery points. *Journal of the Operational Research Society*, 23(3):333–344, Sep 1972. URL: https://doi.org/10.1057/jors.1972.53, doi:10.1057/jors.1972.53.

[130] I.-M. Chao, B. L. Golden, and E. A. Wasil. A fast and effective heuristic for the orienteering problem. *European Journal of Operational Research*, 88(3):475 – 489, 1996. URL: http://www.sciencedirect.com/science/article/pii/0377221795000356, doi:https://doi.org/10.1016/0377-2217(95)00035-6.

[131] M. Gendreau, G. Laporte, and F. Semet. A tabu search heuristic for the undirected selective travelling salesman problem. *European Journal of Operational Research*, 106(2):539 – 545, 1998. URL: http://www.sciencedirect.com/science/article/pii/S0377221797002890, doi:https://doi.org/10.1016/S0377-2217(97)00289-0.

[132] G. Kobeaga, M. Merino, and J. A. Lozano. An efficient evolutionary algorithm for the orienteering problem. *Computers & Operations Research*, 90:42 – 59, 2018. URL: `http://www.sciencedirect.com/science/article/pii/S0305054817302241`, `doi:https://doi.org/10.1016/j.cor.2017.09.003`.

[133] V. Campos, R. Martí, J. Sánchez-Oro, and A. Duarte. Grasp with path relinking for the orienteering problem. *Journal of the Operational Research Society*, 65(12):1800–1813, Dec 2014. URL: `https://doi.org/10.1057/jors.2013.156`, `doi:10.1057/jors.2013.156`.

[134] Y. Marinakis, M. Politis, M. Marinaki, and N. Matsatsinis. A memetic-grasp algorithm for the solution of the orienteering problem. In H. A. Le Thi, T. Pham Dinh, and N. T. Nguyen, editors, *Modelling, Computation and Optimization in Information Systems and Management Sciences*, pages 105–116. Springer International Publishing, Cham, 2015.

[135] K. Taylor, K. H. Lim, and J. Chan. Travel itinerary recommendations with must-see points-of-interest. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, pages 1198–1205, Republic and Canton of Geneva, Switzerland, 2018. International World Wide Web Conferences Steering Committee. URL: `https://doi.org/10.1145/3184558.3191558`, `doi:10.1145/3184558.3191558`.

[136] I.-M. Chao, B. L. Golden, and E. A. Wasil. The team orienteering problem. *European journal of operational research*, 88(3):464–474, 1996.

[137] S. E. Butt and D. M. Ryan. An optimal solution procedure for the multiple tour maximum collection problem using column generation. *Computers & Operations Research*, 26(4):427–441, 1999.

[138] S. Boussier, D. Feillet, and M. Gendreau. An exact algorithm for team orienteering problems. *4OR*, 5(3):211–230, Sep 2007. URL: `https://doi.org/10.1007/s10288-006-0009-1`, `doi:10.1007/s10288-006-0009-1`.

[139] M. Keshtkaran, K. Ziarati, A. Bettinelli, and D. Vigo. Enhanced exact solution methods for the team orienteering problem. *International Journal of Production Research*, 54(2):591–601, 2016. URL: `https://doi.org/10.1080/00207543.2015.1058982`, `arXiv:https://doi.org/10.1080/00207543.2015.1058982`, `doi:10.1080/00207543.2015.1058982`.

[140] S. E. Butt and T. M. Cavalier. A heuristic for the multiple tour maximum collection problem. *Comput. Oper. Res.*, 21(1):101–111, January 1994. URL: `http://dx.doi.org/10.1016/0305-0548(94)90065-5`, `doi:10.1016/0305-0548(94)90065-5`.

[141] C. Archetti, A. Hertz, and M. G. Speranza. Metaheuristics for the team orienteering problem. *Journal of Heuristics*, 13(1):49–76, 2007. URL: `http://dx.doi.org/10.1007/s10732-006-9004-0`, `doi:10.1007/s10732-006-9004-0`.

[142] W. Souffriau, P. Vansteenwegen, G. V. Berghe, and D. Oudheusden. A greedy randomised adaptive search procedure for the team orienteering problem. In *EU-/MEeting*, pages 23–24, 2008.

[143] T. A. Feo and M. G. Resende. A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, 8(2):67 – 71, 1989. URL: `http://www.sciencedirect.com/science/article/pii/0167637789900023`, `doi:http://dx.doi.org/10.1016/0167-6377(89)90002-3`.

[144] W. Souffriau, P. Vansteenwegen, G. Vanden Berghe, and D. Van Oudheusden. A path relinking approach for the team orienteering problem. *Computers & Operations Research*, 37(11):1853–1859, November 2010. URL: `http://dx.doi.org/10.1016/j.cor.2009.05.002`, `doi:10.1016/j.cor.2009.05.002`.

[145] Z. Friggstad, S. Gollapudi, K. Kollias, T. Sarlos, C. Swamy, and A. Tomkins. Orienteering algorithms for generating travel itineraries. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, pages 180–188, New York, NY, USA, 2018. ACM. URL: `http://doi.acm.org/10.1145/3159652.3159697`, `doi:10.1145/3159652.3159697`.

[146] M. G. Kantor and M. B. Rosenwein. The orienteering problem with time windows. *Journal of the Operational Research Society*, 43(6):629–635, 1992. URL: `http://dx.doi.org/10.1057/jors.1992.88`, `doi:10.1057/jors.1992.88`.

[147] P. Vansteenwegen, W. Souffriau, G. Vanden Berghe, and D. Van Oudheusden. Iterated local search for the team orienteering problem with time windows. *Comput. Oper. Res.*, 36(12):3281–3290, December 2009. URL: `http://dx.doi.org/10.1016/j.cor.2009.03.008`, `doi:10.1016/j.cor.2009.03.008`.

[148] D. Gavalas, C. Konstantopoulos, K. Mastakas, G. Pantziou, and Y. Tasoulas. Cluster-based heuristics for the team orienteering problem with time windows. In V. Bonifaci, C. Demetrescu, and A. Marchetti-Spaccamela, editors, *Experimental Algorithms*, pages 390–401. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[149] Q. Hu and A. Lim. An iterative three-component heuristic for the team orienteering problem with time windows. *European Journal of Operational Research*, 232(2):276 – 286, 2014. URL: `http://www.sciencedirect.com/science/article/pii/S0377221713004803`, `doi:http://dx.doi.org/10.1016/j.ejor.2013.06.011`.

[150] F. V. Fomin and A. Lingas. Approximation algorithms for time-dependent orienteering. *Information Processing Letters*, 83(2):57–62, 2002.

[151] A. Garcia, O. Arbelaitz, M. T. Linaza, P. Vansteenwegen, and W. Souffriau. Personalized tourist route generation. In *Proceedings of the 10th International Conference on Current Trends in Web Engineering*, ICWE'10, pages 486–497, Berlin, Heidelberg, 2010. Springer-Verlag. URL: `http://dl.acm.org/citation.cfm?id=1927229.1927285`.

[152] A. Garcia, P. Vansteenwegen, O. Arbelaitz, W. Souffriau, and M. T. Linaza. Integrating public transportation in personalised electronic tourist guides. *Computers & Operations Research*, 40(3):758 – 774, 2013. Transport Scheduling. URL: `http://www.sciencedirect.com/science/article/pii/S0305054811000967`, `doi:https://doi.org/10.1016/j.cor.2011.03.020`.

[153] D. Gavalas, C. Konstantopoulos, K. Mastakas, G. Pantziou, and N. Vathis. Efficient heuristics for the time dependent team orienteering problem with time windows. In P. Gupta and C. Zaroliagis, editors, *Applied Algorithms*, pages 152–163. Springer International Publishing, Cham, 2014.

[154] A. Garcia, P. Vansteenwegen, W. Souffriau, O. Arbelaitz, and M. Linaza. Solving multi constrained team orienteering problems to generate tourist routes. *Tech. rep., Centre for Industrial Management / Traffic & Infrastructure, Katholieke Universiteit Leuven, Leuven, Belgium*, 2009.

[155] W. Souffriau, P. Vansteenwegen, G. Vanden Berghe, and D. Van Oudheusden. The multiconstraint team orienteering problem with multiple time windows. *Transportation Science*, 47(1):53–63, February 2013. URL: `http://dx.doi.org/10.1287/trsc.1110.0377`, `doi:10.1287/trsc.1110.0377`.

[156] K. Sylejmani, J. Dorn, and N. Musliu. A tabu search approach for multi constrained team orienteering problem and its application in touristic trip planning. In *Hybrid Intelligent Systems (HIS), 2012 12th International Conference on*, pages 300–305, Dec 2012. `doi:10.1109/HIS.2012.6421351`.

[157] P. Bolzoni, S. Helmer, K. Wellenzohn, J. Gamper, and P. Andritsos. Efficient itinerary planning with category constraints. In *Proceedings of the 22Nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '14, pages 203–212, New York, NY, USA, 2014. ACM. URL: `http://doi.acm.org/10.1145/2666310.2666411`, `doi:10.1145/2666310.2666411`.

[158] A. Gionis, T. Lappas, K. Pelechrinis, and E. Terzi. Customized tour recommendations in urban areas. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 313–322, New York, NY, USA, 2014. ACM. URL: `http://doi.acm.org.eaccess.ub.tum.de/10.1145/2556195.2559893`, `doi:10.1145/2556195.2559893`.

[159] W. Souffriau, P. Vansteenwegen, G. V. Berghe, and D. V. Oudheusden. The planning of cycle trips in the province of east flanders. *Omega*, 39(2):209 – 213, 2011. URL: `http://www.sciencedirect.com/science/article/pii/S0305048310000496`, `doi:http://dx.doi.org/10.1016/j.omega.2010.05.001`.

[160] C. Verbeeck, P. Vansteenwegen, and E.-H. Aghezzaf. An extension of the arc orienteering problem and its application to cycle trip planning. *Transportation*

Research Part E: Logistics and Transportation Review, 68:74–78, 2014. URL: http://dx.doi.org/10.1016/j.tre.2014.05.006.

[161] Z. W. Geem, C.-L. Tseng, and Y. Park. Harmony search for generalized orienteering problem: best touring in china. In L. Wang, K. Chen, and Y. S. Ong, editors, *Advances in natural computation*, pages 741–750. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[162] D. Herzog and W. Wörndl. Exploiting item dependencies to improve tourist trip recommendations. In *Proceedings of the Workshop on Recommenders in Tourism co-located with 10th ACM Conference on Recommender Systems (RecSys 2016), Boston, MA, USA, September 15, 2016*, pages 55–58, 2016. URL: http://ceur-ws.org/Vol-1685/paper10.pdf.

[163] G. Erdogan and G. Laporte. The orienteering problem with variable profits. *Networks*, 61(2):104–116, March 2013. URL: http://opus.bath.ac.uk/41196/, doi:10.1002/net.21496.

[164] H. M. Afsar and N. Labadie. Team orienteering problem with decreasing profits. *Electronic Notes in Discrete Mathematics*, 41:285 – 293, 2013. URL: http://www.sciencedirect.com/science/article/pii/S1571065313001078, doi:http://dx.doi.org/10.1016/j.endm.2013.05.104.

[165] E. Angelelli, C. Archetti, and M. Vindigni. The clustered orienteering problem. *European Journal of Operational Research*, 238(2):404 – 414, 2014. URL: http://www.sciencedirect.com/science/article/pii/S0377221714003154, doi:http://dx.doi.org/10.1016/j.ejor.2014.04.006.

[166] T. Ilhan, S. M. R. Iravani, and M. S. Daskin. The orienteering problem with stochastic profits. *IIE Transactions*, 40(4):406–421, 2008. URL: http://dx.doi.org/10.1080/07408170701592481, doi:10.1080/07408170701592481.

[167] A. M. Campbell, M. Gendreau, and B. W. Thomas. The orienteering problem with stochastic travel and service times. *Annals of Operations Research*, 186(1):61–81, 2011. URL: http://dx.doi.org/10.1007/s10479-011-0895-2, doi:10.1007/s10479-011-0895-2.

[168] K. H. Lim, J. Chan, C. Leckie, and S. Karunasekera. Towards next generation touring: Personalized group tours. In *Proceedings of the Twenty-Sixth International Conference on International Conference on Automated Planning and Scheduling*, ICAPS'16, pages 412–420. AAAI Press, 2016. URL: http://dl.acm.org/citation.cfm?id=3038594.3038646.

[169] M. D. Ekstrand. The lkpy package for recommender systems experiments. Computer Science Faculty Publications and Presentations 147, Boise State University, Aug 2018. URL: https://md.ekstrandom.net/pubs/lkpy, doi:10.18122/cs_facpubs/147/boisestate.

[170] G. Guo, J. Zhang, Z. Sun, and N. Yorke-Smith. Librec: A java library for recommender systems. In *Posters, Demos, Late-breaking Results and Workshop Proceedings of the 23rd Conference on User Modelling, Adaptation and Personalization (UMAP)*, 2015.

[171] S. Vargas. *Novelty and diversity enhancement and evaluation in Recommender Systems.* PhD thesis, Universidad Autónoma de Madrid, Spain, February 2015.

[172] M. Mihelčić, N. Antulov-Fantulin, M. Bošnjak, T. Šmuc, and R. Bošković. Extending RapidMiner with recommender systems algorithms. In *Proceedings of the 3rd RapidMiner Community Meeting and Conference (RCOMM 2012)*, 2012.

[173] C. Laß, W. Wörndl, and D. Herzog. A multi-tier web service and mobile client for city trip recommendations. In *The 8th EAI International Conference on Mobile Computing, Applications and Services (MobiCASE)*. ACM, 2016. `doi:10.4108/eai.30-11-2016.2267194`.

[174] H. Iltifat. Generation of paths through discovered places based on a recommender system. Master's thesis, Technical University of Munich, 2014.

[175] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, Dec 1959. URL: `https://doi.org/10.1007/BF01386390`, `doi:10.1007/BF01386390`.

[176] W. Wörndl and A. Hefele. Generating paths through discovered places-of-interests for city trip planning. In A. Inversini and R. Schegg, editors, *Information and Communication Technologies in Tourism 2016: Proceedings of the International Conference in Bilbao, Spain, February 2-5, 2016*, pages 441–453. Springer International Publishing, Cham, 2016. URL: `http://dx.doi.org/10.1007/978-3-319-28231-2_32`, `doi:10.1007/978-3-319-28231-2_32`.

[177] J. Melià-Seguí, R. Zhang, E. Bart, B. Price, and O. Brdiczka. Activity duration analysis for context-aware services using foursquare check-ins. In *Proceedings of the 2012 International Workshop on Self-aware Internet of Things*, Self-IoT '12, pages 13–18, New York, NY, USA, 2012. ACM. URL: `http://doi.acm.org/10.1145/2378023.2378027`, `doi:10.1145/2378023.2378027`.

[178] C. Laß, D. Herzog, and W. Wörndl. Context-aware tourist trip recommendations. In *Proceedings of the 2nd Workshop on Recommenders in Tourism co-located with 11th ACM Conference on Recommender Systems (RecSys 2017), Como, Italy, August 27, 2017.*, pages 18–25, 2017. URL: `http://ceur-ws.org/Vol-1906/paper3.pdf`.

[179] D. Herzog, S. Sikander, and W. Wörndl. Integrating route attractiveness attributes into tourist trip recommendations. In *Companion Proceedings of The 2019 World Wide Web Conference*, WWW '19, pages 96–101, New York, NY, USA, 2019. ACM. URL: `http://doi.acm.org/10.1145/3308560.3317052`, `doi:10.1145/3308560.3317052`.

[180] W. Hawthorn. Why Ontarians Walk, Why Ontarians Don't Walk More: A Study Into the Walking Habits of Ontarians., 1989. Energy Probe, Toronto,.

[181] T. Sugiyama, M. Neuhaus, R. Cole, B. Giles-Corti, and N. Owen. Destination and route attributes associated with adults' walking: a review. *Medicine and science in sports and exercise*, 44(7):1275–1286, 2012.

[182] N. Owen, N. Humpel, E. Leslie, A. Bauman, and J. F. Sallis. Understanding environmental influences on walking: Review and research agenda. *American Journal of Preventive Medicine*, 27(1):67 – 76, 2004. URL: http://www.sciencedirect.com/science/article/pii/S0749379704000509, doi:https://doi.org/10.1016/j.amepre.2004.03.006.

[183] T. Pikora, B. Giles-Corti, F. Bull, K. Jamrozik, and R. Donovan. Developing a framework for assessment of the environmental determinants of walking and cycling. *Social Science & Medicine*, 56(8):1693 – 1703, 2003. URL: http://www.sciencedirect.com/science/article/pii/S0277953602001636, doi:https://doi.org/10.1016/S0277-9536(02)00163-6.

[184] H. C. Borst, S. I. de Vries, J. M. Graham, J. E. van Dongen, I. Bakker, and H. M. Miedema. Influence of environmental street characteristics on walking route choice of elderly people. *Journal of Environmental Psychology*, 29(4):477 – 484, 2009. URL: http://www.sciencedirect.com/science/article/pii/S0272494409000498, doi:https://doi.org/10.1016/j.jenvp.2009.08.002.

[185] B. E. Saelens, J. F. Sallis, and L. D. Frank. Environmental correlates of walking and cycling: Findings from the transportation, urban design, and planning literatures. *Annals of Behavioral Medicine*, 25(2):80–91, Apr 2003. URL: https://doi.org/10.1207/S15324796ABM2502_03, doi:10.1207/S15324796ABM2502_03.

[186] B. Giles-Corti and R. J. Donovan. Relative influences of individual, social environmental, and physical environmental correlates of walking. *American Journal of Public Health*, 93(9):1583–1589, 2003. PMID: 12948984. URL: https://doi.org/10.2105/AJPH.93.9.1583, arXiv:https://doi.org/10.2105/AJPH.93.9.1583, doi:10.2105/AJPH.93.9.1583.

[187] H. Pretzsch, P. Biber, E. Uhl, J. Dahlhausen, T. Rötzer, J. Caldentey, T. Koike, T. van Con, A. Chavanne, T. Seifert, B. du Toit, C. Farnden, and S. Pauleit. Crown size and growing space requirement of common tree species in urban centres, parks, and forests. *Urban Forestry & Urban Greening*, 14(3):466 – 479, 2015. URL: http://www.sciencedirect.com/science/article/pii/S1618866715000473, doi:https://doi.org/10.1016/j.ufug.2015.04.006.

[188] S. Elshout, K. Leger, H. Heich, N. Hodges, and F. Nussio. Www.airqualitynow.eu, a common website and air quality indices to compare cities across europe. pages 157–168, 04 2007. doi:10.2495/AIR070161.

[189] D. Herzog. Recommending a sequence of points of interest to a group of users in a mobile context. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, RecSys '17, pages 402–406, New York, NY, USA, 2017. ACM. URL: http://doi.acm.org/10.1145/3109859.3109860, doi:10.1145/3109859.3109860.

[190] O. J. Dunn. Multiple comparisons using rank sums. *Technometrics*, 6(3):241–252, 1964. URL: http://www.jstor.org/stable/1266041.

[191] D. Herzog, C. Laß, and W. Wörndl. TourRec: A Tourist Trip Recommender System for Individuals and Groups. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, pages 496–497, New York, NY, USA, 2018. ACM. URL: http://doi.acm.org/10.1145/3240323.3241612, doi:10.1145/3240323.3241612.

[192] D. Herzog, N. Promponas-Kefalas, and W. Wörndl. Integrating Public Displays into Tourist Trip Recommender Systems. In *Proceedings of the 3rd Workshop on Recommenders in Tourism co-located with 12th ACM Conference on Recommender Systems (RecSys '18)*, pages 18–22, 2018. URL: http://ceur-ws.org/Vol-2222/paper4.pdf.

[193] N. Promponas-Kefalas. Integration of Public Displays into a Mobile Tourist Trip Recommender System. Master's thesis, Technical University of Munich, 2018.

[194] D. Herzog and W. Wörndl. User-centered evaluation of strategies for recommending sequences of points of interest to groups. In *Proceedings of the 13th ACM Conference on Recommender Systems*, RecSys '19, pages 96–100, New York, NY, USA, 2019. ACM. URL: http://doi.acm.org/10.1145/3298689.3346988, doi:10.1145/3298689.3346988.

[195] W. Conover. *Practical nonparametric statistics.* Wiley series in probability and statistics. Wiley, New York, NY [u.a.], 3. ed edition, 1999. URL: http://gso.gbv.de/DB=2.1/CMD?ACT=SRCHA&SRT=YOP&IKT=1016&TRM=ppn+24551600X&sourceid=fbw_bibsonomy.

[196] D. Herzog and W. Wörndl. A user study on groups interacting with tourist trip recommender systems in public spaces. In *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization*, UMAP '19, pages 130–138, New York, NY, USA, 2019. ACM. URL: http://doi.acm.org/10.1145/3320435.3320449, doi:10.1145/3320435.3320449.

[197] M. Stettinger, A. Felfernig, G. Leitner, and S. Reiterer. Counteracting anchoring effects in group decision making. In F. Ricci, K. Bontcheva, O. Conlan, and S. Lawless, editors, *User Modeling, Adaptation and Personalization*, pages 118–130, Cham, 2015. Springer International Publishing.

[198] D. M. Corey, W. P. Dunlap, and M. J. Burke. Averaging correlations: Expected values and bias in combined pearson rs and fisher's z transformations. *The Journal of General Psychology*, 125(3):245–261, 1998. URL: https://doi.org/10.1080/00221309809595548, arXiv:https://doi.org/10.1080/00221309809595548, doi:10.1080/00221309809595548.

[199] N. Tintarev and J. Masthoff. Explaining recommendations: Design and evaluation. In F. Ricci, L. Rokach, and B. Shapira, editors, *Recommender Systems Handbook*, pages 353–382. Springer US, Boston, MA, 2015. URL: https://doi.org/10.1007/978-1-4899-7637-6_10, doi:10.1007/978-1-4899-7637-6_10.

[200] D. Herzog, H. Massoud, and W. Wörndl. Routeme: A mobile recommender system for personalized, multi-modal route planning. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*, UMAP '17, pages 67–75, New York, NY, USA, 2017. ACM. URL: http://doi.acm.org/10.1145/3079628.3079680, doi:10.1145/3079628.3079680.

A Excerpts from own Publications

The results of this thesis project have been published in peer-reviewed journals, conference and workshop papers, and book chapters. This thesis contains excerpts from our own publications which we summarize in the following.

Chapter 2 Parts of the overview in Section 2.3 have been published in [41, 42]. The literature research was done by Daniel Andreas Herzog.

Chapter 3 Parts of the overview in Section 3.1 have been published in [41, 42]. The literature research was done by Daniel Andreas Herzog.

Chapter 4 A brief description of the architecture presented in Section 4.2 has been published in [173]. The initial idea of the architecture has been proposed by Christopher Laß within the scope of his bachelor thesis and extended by all authors since then. The implementation of the AnyRec framework was done by Christopher Laß within the scope of his student assistant work at the Chair of Connected Mobility and under supervision of Daniel Andreas Herzog. The concept was derived in several iterations and discussions between Daniel Andreas Herzog and Christopher Laß.

Chapter 5 The description of the Dijkstra-based tourist trip algorithm and the results from the user study evaluating the algorithm have been published in [41], which is an extended and updated version of a previously published conference paper [176].

The content of Section 5.3 has been published in [178] with some revisions. The implementation of the context-aware algorithm and the evaluation in a user study was done by Christopher Laß within the scope of his guided research project and under supervision of Daniel Andreas Herzog. The concept was derived in several iterations and discussions between Daniel Andreas Herzog and Christopher Laß.

The content of Section 5.4 has been published in [179] with some revisions. The implementation of the algorithm and the evaluation in a user study was done by Sherjeel Sikander within the scope of his master thesis and under supervision of Daniel Andreas Herzog and two researchers from the Chair of Traffic Engineering and Control. The concept was derived in several iterations and discussions between all supervisors and Sherjeel Sikander.

Chapter 6 The content of Section 6.1 has been published in [178] with some revisions. The implementation of the web application and the evaluation in a user study was done by Christopher Laß within the scope of his guided research project and under supervision of Daniel Andreas Herzog. The concept was derived in several iterations and discussions between Daniel Andreas Herzog and Christopher Laß.

The content of Section 6.2 has been published in [173, 191] with some revisions. The implementation of the initial Android application and the evaluation in a user study was done by Christopher Laß within the scope of his bachelor thesis. The development of the final Android application, as presented in Section 6.2, was done by Daniel Andreas Herzog.

The content of Sections 6.3, 6.4, and 6.5 has been published in [192] with some revisions. The implementation of the public display application and the DUI approach and the evaluation in a user study was done by Nikolaos Promponas-Kefalas within the scope of his master thesis and under supervision of Daniel Andreas Herzog. The concept was derived in several iterations and discussions between Daniel Andreas Herzog and Nikolaos Promponas-Kefalas.

Chapter 7 The content of Section 7 has been published in [194] with some revisions. The implementation of the algorithms and the evaluation in a user study was done by Daniel Andreas Herzog.

Chapter 8 The content of Section 8 has been published in [196] with some revisions. The implementation of the prototypes and the entire user study was done by Daniel Andreas Herzog.

B TourRec Categories

Table B.1 lists all categories that can be rated in the publicly available TourRec Android app (× in the last column indicates categories that were used for the experiments in Chapters 7 and 8).

Table B.1: TourRec POI Categories.

Main Category	Subcategory	User Studies
Arts & Entertainment	Amphitheater	
	Aquarium	
	Art Gallery	×
	Circus	×
	Comedy Club	×
	Exhibit	
	Historic Site	×
	Memorial Site	×
	Art Museum	×
	History Museum	×
	Planetarium	×
	Science Museum	×
	Opera House	×
	Theater	×
	Public Art	×
	Stadium	
	Zoo	
Nightlife	Bar	×
	Brewery	×
	Lounge	×
	Nightclub	×
Food	Afghan Restaurant	×
	African Restaurant	

Table B.1 (continued)

Main Category	Subcategory	User Studies
	Asian Restaurant	×
	Café	×
	Eastern European Restaurant	
	Fast Food Restaurant	×
	French Restaurant	×
	German Restaurant	×
	Greek Restaurant	×
	Indian Restaurant	×
	Irish Pub	×
	Italian Restaurant	×
	Latin American Restaurant	
	Mexican Restaurant	×
	Middle Eastern Restaurant	×
	Turkish Restaurant	×
	Vegetarian / Vegan Restaurant	
Outdoors & Recreation	Bathing Area	
	Beach	×
	Botanical Garden	×
	Bridge	×
	Castle	
	Fountain	×
	Garden	×
	Hot Spring	
	Lake	×
	Mountain	
	National Park	
	Nature Preserve	
	Palace	×
	Park	×
	Pedestrian Plaza	
	Plaza	×

Table B.1 (continued)

Main Category	Subcategory	User Studies
	Scenic Lookout	×
	Sculpture Garden	
	Volcano	
	Waterfall	
Shopping	Clothing Store	×
	Department Store	×
	Flea Market	
	Market	
	Shopping Mall	×
	Shopping Plaza	

C Demographic and Group-Related Questionnaire for the User Studies in Chapters 7 and 8

The demographic and group-related questionnaire was presented once to the participants after finishing both user studies.

1. What is your age? (<18 / 18-24 / 25-34 / 35-44 / 45-54 / >54)
2. What is your gender? (Female / Male / Other / Prefer not to comment)
3. What is your highest qualification? (Master's degree or higher / Bachelor's degree / High school diploma or equivalent degree / Less than high school diploma)
4. How frequently do you use tourism applications on smartphones or tablets? (such as Yelp, or Foursquare)? (Daily / Weekly / Monthly / Less than once a month / Never)
5. How frequently do you interact with public and interactive displays or information kiosks? (in tourist areas, shopping malls, at train stations, etc.)? (Daily / Weekly / Monthly / Less than once a month / Never)
6. How often do you go on holiday per year? (weekend trips, travelling abroad etc.) (<1 / 1 / 2-3 / 4-5 / >5)
7. How would you categorize your group in this study? (Family / Close friends / Student fellows / Coworker / Other (specify) / I don't know this group)