Personalization Models for Travel Recommender Systems

MOHAMMED HAMAD ALATIYYAH

Personalization Models for Travel Recommender Systems

Mohammed Hamad Alatiyyah

A Thesis presented for the degree of
Doctor of Philosophy

Durham University

Innovative Computing Group
Department of Computer Science
University of Durham
England

September and 2019

Personalization Models for Travel Recommender Systems

Mohammed Hamad Alatiyyah

Submitted for the degree of Doctor of Philosophy
September 2019

Abstract

This thesis proposes three novel personalisation models to improve travelers experiences of using constraint-based Travel Recommender Systems (TRSs). Specifically, the three models are (i) personalisation of a recommended travel plan based on user-dependent constraints; (ii) maximisation of a function to represent users satisfaction levels; and (iii) maximisation of user satisfaction levels derived from a model of conflicts that users frequently experience when travelling as a group.

The first model proposed, *Item Constraints Data Model* (ICDM) is designed to tackle the limitations of existing models, i.e. their inability to generate a recommended travel plan based on a variety of constraint types. Our proposed ICDM aims to overcome this by generating customised travel plans based on specific individual user-constraint considerations.

Moreover, our ICDM is able to handle specific constraints defined over particular time periods (e.g., a traveler who wishes to visit outdoor attractions in the afternoons). Another benefit of this approach is that it permits the use of general-purpose optimisation algorithms to generate recommended travel plans. We have conducted an ICDM validation study based on public datasets to compare our ICDMs performance against other models. Our ICDM returned good results on these public data sets when based on a general-purpose optimisation algorithm: *Ant Colony Optimisation* (ACO). It also performed satisfactorily on a dataset that we assembled from online sources.

The second model we propose, the *Happiness Model* (HM), maximises users

satisfaction levels on a tour based on their likes and dislikes. This is relevant because existing models are limited by their propensity to maximise the benefits gained by visiting particular *Points Of Interest* (POIs) while ignoring other important factors such as connections and waiting times. The main aim of our HM is to simulate a users feelings during their travel experience by maximising the main factors affecting their travel-satisfaction levels. Specifically, the HM optimises travellers journeys not only based on their particular preferences but also considering their effort spent for gaining access to their preferences, which is then reflected their overall happiness level. The validation results demonstrated that the HM model can be used to maximise user satisfaction and represents an abstract model that is able to handle any factor likely to affect user satisfaction.

Third, we have addressed the *Group Tourist Trip Design Problem* (GTTDP), which involves determining a satisfactory trip plan for a group of tourists visiting several POIs. Our proposed, *Group Tour Trip Recommender Model* (GTTRM) is designed to solve the GTTDP by maximising the group members respective satisfaction levels and reducing the potential for conflict among group members. The novelty of the GTTRM lies in the fact that it solves the GTTDP by deciding on the optimum way to divide up a particular group into a number of sub-groups for specific parts of the trip. Existing models are limited in this respect because they are only able to split up a group at the start of the trip and build recommendations for each group member separately. The results of the GTTRM show that it is effective at maximising individuals satisfaction levels within a group-travel context.

In summary, this thesis introduces the three novel models mentioned above to facilitate the building of travel recommendations based on TRSs. Specifically, we show that by considering TRSs as an optimisation problem, we are able to provide highly accurate travel recommendations and overcome the limitations of existing approaches.

March 19, 2020

Declaration

The work in this thesis is based on research carried out at the Innovative Computing Group, the Department of Computer Science, Durham University, England. No part of this thesis has been submitted elsewhere for any other degree or qualification and it is all my own work unless referenced to the contrary in the text.

Acknowledgements

- I would like to thank *Prince Sattam Bin Abdulaziz University* for the scholarship that funded my research.
- I would like to express my thanks to my supervisors Dr. Frederick Li and Dr. Ioannis Ivrissimtzis, for their guidance during my Ph.D. and helping me with helpful comments on this thesis.
- I would like to thank Dr. Barnaby Martin for his feedback on the *Happiness Model* (Chapter 5).
- I thank Dr. Noura Al Moubayed for her guidance in collecting data and the technical advice.
- My thanks to Dr. Khalid Ismail for his advice during my Ph.D. journey.
- I would like Dr. Mohammed Alhawarat for his feedback on the Item Constraints Data Model (Chapter 3).
- Additionally thanks to my examiners Dr. Boguslaw Obara and Dr. Maximilien Gadouleau, for their helpful comments on the thesis.
- Finally, I would like to thanks the conference *RecSys* (2017) participants for their valuable comments and feedback.

Contents

List of Figures

List of Tables

List of Abbreviations

RSs	Recommender Systems
TRSs	Travel Recommender Systems
GRSs	Group Recommender Systems
POIs	Points Of Interest
TTDP	Tourist Trip Design Problem
ICDM	Item Constraints Data Model
HM	Happiness Model
FTRM	Flexible Travel Recommender Model
GTTDP	Group Tour Trip Design Problem
GTTRM	Group Tour Trip Recommender Model
MOOP	Multi-Objective Orienteering Problem
GMOOP	Generalized Multi-Objective Orienteering Problem
CF	Collaborative Filtering
CB	Content-Based
KB	Knowledge-Based
DF	Demographic Filtering
HA	Hybrid Approaches
TSP	Travelling Salesman Problem
KP	Knapsack Problem
VRP	Vehicle Routing Problem
OP	Orienteering Problem
TOP	Team Orienteering Problem
OPTW	Orienteering Problem with Time Windows
TOPTW	Team Orienteering Problem with Time Windows

ACO	Ant Colony Optimization
GACO	Group Ant Colony Optimization

Glossary

In below list, number of words have been used in the thesis, and we clarify the main meaning we have used it this thesis.

Word	**Meaning**
Node, POI, or item	A place can be visited by travelers
Constraint	It is a condition
Satisfaction or happiness	Represents how much a user is a pleasure about the activity doing
Personalization	Reducing options to match the user's likes
Traveler, user, or tourism	Person is going to move from the city which he/she lives into another city
Group	A group of users
Tour trip, route, Path, or plan	A set of POIs which are placed in order (Holiday package)
Top-k Recommendation	RSs recommend a list of k items
Package Recommendation	RSs recommend a set of items
Sequence Recommendation	RSs recommend a set of items in order
Plan Recommendation	RSs recommend a set of items to visit them based on the recommended time
RSs Domain	Areas of applications of RSs

Static data	Is a type of data in ICDM where the data does not change when the time changes
Dynamic data	Is a type of data in ICDM where the data change from time to another
Dynamic Constraints	Constraints that are be customized for different users (each user can create own constraints)
Trip Constraints	It is a condition which is a located to control the trip
Connection Constraints	It is a condition which is a located to control the traveling between POIs
Item Constraints	It is a condition which is a located to choose POIs
Based Constraints	All trip must be satisfied
Extra Constraints	Based on user preferences (maybe HC or SC)
Hard Constraints or Must	must be satisfied (maybe BC or EC)
Soft Constraints or Preferring	Represent the user's satisfaction
Waiting time or Wasting time	Represents the time that the user is do nothing
Visiting time	Represents the total time to visit a POI
Connection time	Represents the total time taking by moving from a POI to another
Start Point	Represents the location which the trip start from
End Point	Represents the location which the trip end to
Start Time	The time should the traveller start
End Time	The time where the traveller should back to the End point
Trip length or T_{max}	Represents the total time must the trip not exceed

User Constraints, User requirement, or User wishes	Represents the constraints from the users (always *Extra Constraints* see Chapter 3)
POIs Data	Represents the POIs information to match it with the BC and EC
Connections Data	Represents the traveling from POI to another information to match it with Constraints
Activity	Represents the visiting
Connection	Represents the moving from a POI to another
Group constraints	Represents the aggregation of the users' constraints
Group Preferences	Represents the aggregation of the users rating
Group profile	Representing aggregation the group preferences and constraints
Splitting a Group	Divides the group into subgroup for sometimes to all
Time Windows	Representing the opening/closing time for a POI
User's Preferences	Representing the users likes and dislikes (rating for different items)

Chapter 1

Introduction

In this chapter, we present an overview of the research presented in this thesis. We present the background information first and then discuss the motivations for developing trip *personalization*[1] models. Next, we state the research objectives of this thesis and discuss the research questions. Moving on, we present a summary of the research findings and discuss their importance. Finally, we briefly outline the structure of the rest of the thesis.

1.1 Background

In this age of information overload, consumers face a multitude of choices: what to buy, what to wear, what to watch, where to go, and even whom to date. Thus, it seems evident that some may not have sufficient time or knowledge to be able to assess all the relevant information thoroughly in order to make optimal choices [48]. For example, in terms of travel, consumers face the problem of deciding which tourist destinations to visit when they visit a country or area [49], as well as dealing with the problem of planning a *trip*[1] to a destination they are visiting for the first time [22].

Tourists[1] tend to plan trips based on information gathered via digital sources e.g., travel websites, maps, and travel blogs [14, 22], or traditional sources such as books, friends, magazines. In a more formal terminology, tourists travelling to a des-

[1]See Glossary for definitions of italicized words

tination have many options for choosing which Points Of Interest (POIs)[1] to visit and in which order they will visit them. This presents a challenge because several *constraints*[1] must also be considered e.g., weather conditions, opening and closing times for certain POIs, time available for sightseeing, etc. [14, 48, 49, 63]. When all of these considerations are taken into account, planning a trip that matches a consumer's preferences can become a very complicated and time-consuming endeavour [8, 14, 20].

1.1.1 Recommender Systems

Recommender Systems (RSs) support users by guiding them to personalised choices usually based on a large number of options derived from decision-support systems [20, 42]. Travel Recommender Systems (TRSs) make the planning process more straightforward [48], and aim to match user's needs with the characteristics of their trip [20, 48]. In practice, most TRSs are designed to recommend city-based attractions [48].

When making decisions about a trip, travellers often consider factors such as time, cost, transportation, weather constraints, etc. TRSs should support (1) schedule planning; (2) the ability of users to change their input constraints; and (3) the ability to adjust the recommended results based on their own *personal requirements*[1] [36].

The main challenges in TRSs research have been identified as: (1) consumption capability [8,63], (2) trip personalisation [8,63] (3) POI's availability [115]; (4) travel time [115]; and (5) diversity of POIs [115]. In addition, the biggest universal problem facing all TRSs is the limited availability of data, such as user ratings, trip plans, trip budgets, constraints, etc.

In addition to the scarcity of data, the sparsity of data also presents a challenge in that only a few travelers may have rated certain POIs. As [111] mentions, it becomes difficult to learn from such sparse data. Similarly, [14] emphasises that a critical aspect in all TRSs is having rich enough data on users and POIs. In addition, most TRS-based systems only focus on well-known attractions [36, 63], which reduces recommendation effectiveness [36]. After all, most TRSs fail to achieve comprehensive trip plan personalisation [36] because they mostly rely solely on popular POIs, having no set trip plan, and their limitations in terms of allowing users to change their requirements [36]. A comprehensive discussion of existing systems and their limitations will follow in Chapter 2.

Group Recommender Systems (GRSs) are RSs that produce recommendations for a group of users (rather than a single user). GRS's complexities stem from (1) their intricate and interdependent processes, which include aggregating different *users preferences*[1]; (2) processing group members opinions in order to create a single, unified profile or producing recommendations for each member of the group and combining these; (3) dealing with various users roles; and (4) presenting the final recommendations.

In a basic scenario, each group is comprised of individual users, and each user has their own unique constraints and preferences. The challenge for GRSs is producing recommendations by taking into account ways to reduce conflicts among users' and maximise users' satisfaction levels.

1.1.2 Tourist Trip Design Problem

The Tourist Trip Design Problem (TTDP), which involves creating a trip plan for tourists interested in visiting multiple POIs, is defined in [107]. The TTDP is based on tourists having different preferences for POIs they wish to visit and limited time for sightseeing; each POI having a set of attributes (e.g., category, location, child-friendly, admission cost, etc.), and each trip allocating a limited time for sightseeing; therefore, a TTDP solution should maximise the total score achieved when visiting a specific POI (where each POI has different scores); the system not only chooses POIs but also selects the best route(s) between selected POIs.

The TTDP approaches TRSs as an optimisation problem where the TTDP's solutions should respect traveller's constraints and POI's characteristics [49]. The main objective of TTDP is to maximise tourist *satisfaction*[1] level by enabling them to visit high-scoring POIs while taking user's constraints into account.

1.1.3 Research Motivations

In terms of employment, the UK's fastest growing business sector since 2010 has been the tourism industry; in 2010, the tourism sector accounted for around 9.9% of the UK's total GDP [1]. In addition, tourism has both direct and indirect impacts on the UK's economy, as well as induced economic effects [68]. The direct contribution of the UK's tourism industry to the economy is in supporting service industries such as the hotel industry and encouraging improvements in accommodation facilties, the restaurant sector, transportation links, etc.). The indirect contributions of the tourism industry to the country's economy is in supporting private tourism investment spending while the induced contribution of tourism benefits economic sectors such as the leisure industry and retail [68].

The economic benefits of enhancing the tourism industry's performance has motivated computer scientists to develop new tourism-industry specific tools to provide personalised recommendations that suggest recommendations based on users' own unique constraints and preferences. For these reasons, a number of notable studies have addressed this research area from a range of different academic disciplines [24]. Other motivations for researching personalisation models include:

- Providing automated tour trip plans.
- Increasing users' trust in RSs.
- Improving the quality of recommendations for travellers.
- Reducing running costs by reducing search-related processing.

1.2 Research Objective

This thesis' wider motivation is to maximise tourists' *satisfaction*[1] with their trip by developing models to personalise users' trips. While past studies have focused on creating algorithms or predictions to maximise tourists' satisfaction with their trip, accomplishing this aim by designing a dynamic model to handle different types of constraints represents a gap in the literature. Therefore, the aim of the present study is to develop efficient RSs techniques to support users in building their trips while maximising their satisfaction levels.

- Existing models are unable to handle arbitrary users' constraints. The types of constraints existing models handle are hard-encoded into them instead of being user-customisable variables. Thus, developing a data model which can customise users' constraints related to gathering data on the time users have available is the first objective (see Chapter 3 and 4).

- Customising key trip factors (i.e. *visiting times*[1], *connection times*[1], and *waiting times*[1]) is the second objective (see Chapter 5 and 7).

- Developing an algorithm for a group of travellers which maximises the users' *satisfaction*[1] levels and minimises the conflicts among the individual group members (see Chapter 6).

Research Questions This thesis explores the following Research Questions (RQ):

- **RQ1:** How can user constraints be formally modelled in order to customise *constraints*[1] for each traveller?

- **RQ2:** How can trip modelling be based on various types of activities and how can modelling these activities affect the trip? (Note: this offers an alternative to existing POI-centred TRSs. Here, instead of optimising selections over set of places to be visited, we optimise over the time spent).

- **RQ3:** How can the different constraints within a group of travellers be modelled and how can we maximise group members' *satisfaction*[1] levels? (Note: splitting the group into sup-groups by taking a decision when building the recommendations improves the group users' individual satisfaction levels).

1.3 Summary of Contributions

This thesis makes three major contributions to the field of Recommender Systems in terms of the relevance of personalised recommendations based on user constraints and preferences. These are:

- Development of *The Item Constraints Data Model* (ICDM) which deals with different types of constraints to recommend trip plans based on *user's constraints*[1].
- The development of the *Happiness Model* (HM) to maximise user's *satisfaction*[1] levels.
- Development of *The Group Tour Trip Recommender Model* (GTTRM) which is an algorithm designed to maximise the individual satisfaction levels among a *group*[1] and reduce the conflicts among group members.

Several other minor contributions have been made by this thesis:

- Collecting a real-world dataset (*Durham Dataset*), which has unique characteristic such as price, child friendly POIs etc., for travellers (see Appendix A).
- Developing the *Flexible Travel Recommender Model* (FTRM) which is an abstract model for the Orienteering Problem (OP) (see Chapter 4).
- Investigating the main factors affecting traveller's satisfaction levels, which also affect user's happiness ratings (see Chapter 5).
- Developing the *The Constraints Data Model* (CDM), which was extended from the ICDM to deal with different *user constraints*[1] within a group of travellers (see Chapter 6).
- Developing the *Generalized Multi-Objective Orienteering Problem* (GMOOP), which is a model converting multi-objective optimisation into a single objective model (see Chapter 7).

These contributions will be further discussed in Chapters 3 to 7.

1.4 Research Importance

Perhaps the most important aspect of this thesis is improving trip personalisation based on user's constraints. The greatest impact of increasing personalisation performance is maximising users' *happiness*[1] levels. Indeed, tourism studies and happiness research have caught the attention of various researchers from different disciplines [33].

Specifically, it is important to utilise technological solutions to both maintain and improve the tourism sector. Indeed, because each traveller is subject to different types of constraints and preferences, identifying an optimal trip plan in TRSs means that the search space grows exponentially as the number of options increase. This also means that the search space is too large to carry out an exhaustive search within an acceptable timescale [24].

The key findings of this thesis are

- Using the ICDM, it is feasible to handle different types of *user constraints*[1] (see Chapter 3 and 4).
- The proposed *Happiness Model*, is an abstract model that can deal with all the factors related to user's *satisfaction*[1] on a particular trip (see Chapter 5 and 7).
- Our *Group Tour Trip Recommender Model* can be applied to reduce the conflicts among group members and maximise users' satisfaction levels by splitting the group into sup-groups.

1.5 Thesis Structure

This thesis is organised into eight chapters and contains research undertaken at the *University of Durham* between October 2016 - September 2019.

- **Chapter 2** provides an overview of the RSs and presents relevant studies. The different RSs models, types, and techniques are also discussed. Previous research into TRSs and GRSs is also presented.

- **Chapter 3** introduces the ICDM, and the data and constraint types that are used in the model. It explains in detail the ICDM's mechanism of operation, which is then employed to implement various existing models.

- **Chapter 4** presents the *Flexible Travel Recommender Model* (FTRM) which is a general model for the OP extended from the ICDM. We present comprehensive experiments on the different models (the OP and its extension). Moreover, a real-world dataset (*Durham Dataset*) is used, and various scenarios are examined to explore the ICDM's features. Finally, the general- purpose algorithm, *Ant Colony Optimization*, is developed to build recommended plans.

- **Chapter 5** introduces the HM, which is a novel model to maximise travellers' *satisfaction*[1] by building the most enjoyable trip possible. Next, an evaluation of the HM based on public datasets is provided, and the different impacts of changing particular preferences on a trip are explained.

- **Chapter 6** proposes the Group Tour Trip Design Problem (GTTDP) which defines the problem of a group of travellers who are interested in visiting multiple POIs. The GTTRM is introduced to solve the GTTDP based on real-world dataset (*Durham Dataset*), and different groups of travellers' data are described in the chapter.

- **Chapter 7** presents the *Generalized Multi-Objective Orienteering Problem* (GMOOP), which is a model to solve multi-objective optimisation by converting it into a single objective. A comparison between the results of the existing models and the GMOOP is provided.

- **Chapter 8** concludes the main context of this thesis and summarises its contributions. The limitations and future work of the research are discussed.

Chapter 2

Background and Literature Review

This chapter presents the background and a critical literature review of the relevant research on *Recommender Systems* (RSs). It reviews the problems and algorithms in the state-of-the-art works and discusses the various types of limitations and models that are employed in *Travel Recommender Systems* (TRSs) research. In detail, Section 2.1 defines RSs, and highlights the importance of RSs for users and businesses. Furthermore, it also describes the approaches to processing and producing recommendations.

Section 2.2 presents the main RSs models and reviews the existing works on these models. Additionally, it compares these models and details the drawbacks of each one in turn. Section 2.3 defines TRSs and outlines their importance. In addition, it compares TRSs with other RS applications. Section 2.4 illustrates the main types of TRS recommendations (in terms of their types of outputs) and compares TRS recommendations types. Section 2.5 analyses and summarises the different techniques that have been used in RSs in general and in TRSs specifically. Section 2.8 presents a review of *Group*[1] Recommender Systems (GRSs) and addresses the research gaps relevant to the existing proposed recommendation solutions.

[1]See Glossary for definitions of italicized words

2.1 Recommender Systems

Recommender Systems (RSs) are defined as systems that support users by guiding them to personalized items based on a huge number of options derived from decision-support systems [20,42]. Another definition by [35] defines RSs as a means to collect *user preference*[1] information for a group of items in order to provide consumers with recommendations and predictions.

RSs are becoming ever more diverse because of their ability to exploit different types of input data such as user ratings, item attributes, user specifications, and domain knowledge to produce recommendations [4]. In addition, RSs represent a web-based technology that proactively provides suggestions for items of interest to users based on their browsing behavior or explicitly-stated preferences [85]. In another definition, RSs collect data on users preferences for a set of items such as movies, songs, books, and travel destinations. This data can be captured either explicitly, for instance, via collecting users ratings, or implicitly, such as via monitoring user behavior: visiting a particular place (physical or online) or buying a particular item.

In addition, Bobadilla et al. have mentioned that an RSs can also employ user demographic data such as age, gender, or nationality. Further, social media information such as Twitter followers and those followed, Instagram pages and Facebook posts are also commonly used. On top of this, the Internet of Things (IoT) has also attracted searches by providing the use of information such as GPS locations and RFID chips [17]. Hence, Aggarwal mentions that the goals of RS are predicting problems for which relevant items are recommended and producing *top-K*[1] items for users [4].

2.1.1 The Importance of RSs

In recent years, public awareness of RSs has grown exponentially [4]. Nowadays, numerous online retail companies have applied RS-based applications [43], and such systems play important roles in the information and e-commerce ecosystem [40]. RSs have also drawn significant interest from industry. For example, Amazon has used Collaborative Filtering (a type of RSs which will be explained in the next section in more detail) for a decade to recommend products to their customers [40]. In addition, Netflix has evaluated the importance of recommendation techniques for its movie rental service worth $1M as part of the well-known Netflix Prize in 2009 [40]. Naturally, people also rely on recommendations from their friends and experts in making their decisions and exploring new material [40].

2.1.2 The Processing of RSs

In general, we can categorize RSs processing approach into three steps: (1) Data processing, (2) Algorithm processing, and (3) Output processing. Figure 2.1 presents an overview of systems processes upon which RSs are based.

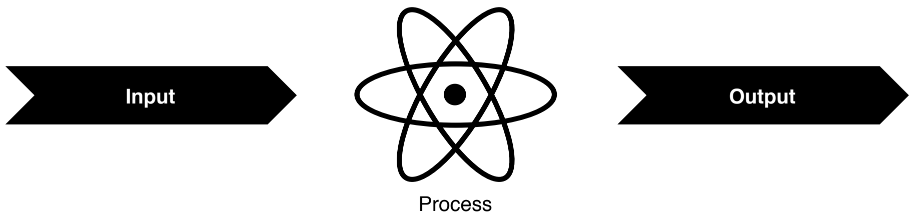


Figure 2.1: Overview of systems processes

First, *input processing* in RSs is divided into (1) explicit and (2) implicit. In the former, data is entered by the user to clarify the preferences while the latter involves RSs collecting data about the user whilst the user using the system. There is a large volume of published studies on RSs that focus on *Data Processing*.

Second, *algorithm processing* can be defined as the procedures used to produce recommendations. For instance, where a user enters their *preferences*[1], the *algorithm processing* should use these preferences to produce recommendations.

Third, *output processing* presents how RSs provide recommendations where many forms of recommendations are available, such as k-top, *package*[1], *sequence*[1], and *plan*[1]. Furthermore, explaining these recommendations is also included as a part of this processing.

2.2 Recommender Systems Models

The basic RS models are divided into five main categories: (1) Collaborative Filtering (CF), (2) Content-Based (CB), (3) Knowledge-Based (KB), (4) Hybrid-Based (HB), and (5) Demographic Filter (DF). First, CF uses the synergic power of items rated by numerous users to provide recommendations [4]. Second, Bobadilla et al. define the CB approach as providing recommendations based on a users past choices or by analyzing certain content such as text, images, and sound to identify similarities [17]. Third, in the case of knowledge-based recommender systems, these are based on user requirements explicitly rather than the previous history of the users [4]. In addition, the hybrid-based approach produces recommendations by combing more than one recommender system to take advantage of the strengths of certain methods and avoid the weaknesses of others. Finally, the DF approach collects users characteristics to produce recommendations [4]. The following sections will explore these RS models in more detail.

2.2.1 Collaborative Filtering

As CF utilises similarity in user actions, ratings, and behavior to produce recommendations, Bobadilla et al. commented that this method is based on the historical processing of human decisions [17]. However, CF makes predictions ratings for items which are not consumed by the user. Equally, Schafer et al. define CF as collecting data and using filtering techniques based on users previous opinions of items or the history of their previous purchases [89]. In addition, Ekstrand et al. acknowledge that CF is a popular RS algorithm which is based on the predictions and recommendations or behavior of other users in a particular system [40].

Furthermore, Hernando et al. confirm that RS based on CF can be classified into two types based on algorithms used to predict users tastes: (1) those based on memory, and (2) those based on models [16, 58].

Memory-based collaborative filtering involves finding similar users similar to a target user in a system to predict their ratings for items they have not yet rated [3]. Memory-based methods are also called neighborhood-based collaborative filtering algorithms [4] because they focus on neighboring users. In addition, Bobadilla et al. confirm that the most extensively-used algorithm in CF is the K-Nearest Neighbours (kNN) [17]. kNN is executed in three steps: (1) identifying the user's neighbors, (2) combining the users' ratings for items which have not yet been rated by the user, and (3) predicting the top-k recommendations [17]. Equation (2.2.1) shows the mathematical calculation used to find similarity between users u and v where r_{vi} denotes the user v rating for item i, and $\hat{r}_v$ presents the average rating for user v.

$$W_{uv} = \frac{\sum_{i=1}^{|N|} (r_{ui} - \hat{r}_u)(r_{vi} - \hat{r}_v)}{\sqrt{(r_{ui} - \hat{r}_u)^2}\sqrt{(r_{vi} - \hat{r}_v)^2}} \tag{2.2.1}$$

In systems that use vast amounts of data, scalable algorithms are needed to cope with such quantities of information. Likewise, model-based methods are based on traditional machine learning. Model-based approaches operate in two phases: a learning phase and a prediction phase [4]. Such models can predict the ratings of items for users very quickly when the learning phase has been completed [58]. Hernando et al. claim that such models provide better prediction accuracy [58]. Bobadilla et al. mention the most extensive models which have been used for this purpose, including Bayesian classifiers, neural networks, fuzzy systems, genetic algorithms, latent features, and matrix factorization [17].

2.2.2 Content-Based

CB uses items attributes to which users have linked in the past to provide recommendations [43]. In other words, CB uses the content of items rated by users to recommend them to other users [82]. For example, if a user had watched a fiction movie in the past, the CB method might recommend other recent fiction movies the user has not yet seen [17]. In addition, Bobadilla et al. declare that in CB, text, images, and sound can be analyzed to provide recommendations [17]. Felfernig et al. emphasize CBs strength is that because historical data is available, CB does not require any additional information [43].

2.2.3 Knowledge-Based

KB provides recommendations based on inference algorithms that analyze the correlations between *users preferences*[1] and items [26]. KB is based on ontologically based technology which is a powerful tool for intelligent inference [26]. In addition, Ricci et al. mention that KB provides recommendations by matching certain items with users needs [87]. However, each RSs method has its own strengths and weaknesses, and the KB approach is applicable to applications where the items needed are very rare, or new items of the same type as other items are different, such as when purchasing a car, for example. However, clearly, a 2010 Vauxhall Corsa does not share the same attributes as a 2019 Vauxhall Corsa, for instance. Another example is buying a house, which is a very rare action: each person will have different requirements.

2.2.4 Demographic Filtering

DF involves using demographic information to detect the type of users interested in certain items [83]. Table 2.1 shows information on the gender, age, education, etc. of users that rated the Superbad and The Dark Knight movies. From the information in Table 2.1, DF should be able to predict which type of user likes a particular movie. In other words, the RSs are able to produce recommendations for users based on the demographic information which is collected, and DF is used when new users signed to the RSs.

Table 2.1: An example of demographic information of users who rated movies with the rating

Users	Gender	Age	Post code	Eduction	Employed	Superbad	The Dark Knight
Sam	M	22	DH1	Undergradaute	Full Time	4	1
Mikel	M	25	DH1	Postgraduate	Part Time	5	4
Sarah	F	29	DH1	Postgraduate	Part Time	3	2
Tom	M	27	DH1	Undergradaute	Full Time	4	5

2.2.5 Hybrid Approaches

Hybrid Approaches (HA) combine different RS models or algorithms to produce better accuracy and avoid the drawbacks of individual RSs methods [25,26]. Moreover, Ekstrand et al. find that HA outperforms individual algorithms in some applications [40]. In addition, Burke mentions several combinations of methods that have been used including Weighted, Switching, Mixed, Feature-combining, Cascading, Feature-augmenting, and Meta-level [25]. Furthermore, Ekstrand et al. emphasize that HA has proven to be a powerful technique in the Netflix Prize; the winner combined 100 separate algorithms [40]. The two most-combined RS models are CF and CB, and one technique used in HA involves switching from an RS model to another based on the demands of the current situation.

In conclusion, we need to consider which RS models are the most powerful. CF and CB have been the most-studied models over the past decade [82]. CF suffers from the sparsity problem while CB suffers from a failure to recommend particular items because a user did not rate any items using a specific keyword. In addition, the kind of data that is available will determine which RS model is chosen. For example, if insufficient details about specific items were given, it would be difficult to use CB as it relies on item content.

2.3 Travel Recommender Systems

Travel Recommender Systems (TRSs) are RSs designed and built for traveling activities such as selecting accommodation, restaurants, or POIs. The main aim of TRSs is to make the planning process more straightforward [48], and match user's needs with the characteristics of their trip [20,48].

However, TRSs face four critical challenges. First, the availability of travel data is much scarcer than other traditional items such as movies. Second, TRSs handle a range of different items they should recommend to users that satisfy their requirements such as POIs and restaurants, and there is a complicated relationship between these items; for example, after having lunch in a restaurant, users may like to go to a POI where they can walk in open space, such as park. Third, *user* *preferences*[1] in TRSs are a challenge because of the extent to which ratings are available. Fourth, in TRSs, items can depreciate over time and this may occur more quickly than traditional items for a recommendation; also some items in TRSs are related to particular seasons such as skiing in winter months [71]. Moreover, in some situations where multiple criteria are taken into account (e.g. ratings, cost, time, and group satisfaction), TRSs are time-critical and budget-critical. In addition, users usually have constraints on POIs or budgets, and so TRSs should consider these constraints.

2.3.1 The Importance of TRSs

TRSs play an essential role for users, governments, and businesses. As we discuss the types of TRSs in the next section based on type, we now need to highlight the importance of TRSs.

Importance for Users

TRSs offer user the following benefits:

- Time-efficiency.
- Providing users with their preferred items with great accuracy.
- More easily allowing changes to plans based on *user requirements*[1].

Importance for Governments

TRSs offer governments the following benefits:

- Increasing the number of tourists visiting their country.
- Effective analysis of users' preferences and requirements.
- Enabling a diverse range of tourists to visit the country.

Importance for Businesses

TRSs offer businesses the following benefits:

- Increasing profit margins.
- Increasing user/customer *satisfaction*[1].
- Better understanding of users wants and preferences.

2.3.2 Comparison between TRSs with other domains

TRSs offer several unique attributes that other RS domains do not. Table 2.2 illustrates the essential characteristics of TRSs compared with several different applications. First, the cost of undertaking these activities varies: watching movies or listening to music is not usually expensive because users can benefit from a standard membership with some providers such as Netflix, and watching the news or reading the news is mostly free. However, the cost of exercise classes may be more expensive than movies and music because exercise recommendations tend to operate over several weeks, and travel activities are more costly still compared to these previous activities. Second, some RS *domains*[1] contain different types of items; for example, when booking trips, travelers will need to select accommodation, POIs, and dining options. Another example: in Exercise RSs, users might need to select specific activities in order to achieve their goals such as exercise classes, diet options, and sleep periods. However, in other domains, most of the recommendations consist of the same item types. Third, some activities are carried by more than one user, such as watching a movie with the family and listening to music in the gym. Table 2.2 shows that most of the RS domains are applicable for groups of users except exercise, which generally needs to be personalized to only a single user. Fourth, most of these activities take from 30 minutes to three hours, except the TRSs and Exercise RSs, which might take days or weeks. Fifth, most of the activities are undertaken daily or weekly except the TRSs where most travelers take one holiday each year. Sixth, while user satisfaction is an essential factor in all RS domains, TRSs are critical for producing user satisfaction because usually, users take a trip with a long working term. In other words, users are waiting for their holiday to begin so they can rest and have fun; in this case, users know they will have to wait another year for their next holiday.

Table 2.2: Comparing the TRSs and other domains

Domain	Price	Multi-items	Group	Time consuming	Repeats	User *satisfaction*[1]
TRSs	$$$$	•	•	•	Yearly	Critical
Movie RSs	$		•		Daily	Important
Music RSs	$		•		Daily	Important
News RSs			•		Daily	Important
Exercise RSs	$$$	•		•	Weekly	Important

2.4 Recommender Systems Types

As RSs are used in a diverse range of applications, RSs are designed to *personalize*[1] the specific options available; therefore, RS outputs are grouped under four categories: (1) *Top-k* recommendation, (2) *Package* recommendation, (3) *Sequence* recommendations, and (4) *Plan* recommendations.

2.4.1 Top-k Recommendation

Most RSs are built to produce top-k items, and these types of recommendations use different techniques (see Section 2.5). Top-k recommendations are suitable for some domains such as movie [69] and music [116] recommendations. This type of recommendation is mainly concerned with identifying the top-k items that a particular user may like.

Figure 2.2 shows an example of a movie top-k recommendation procedure. The top part of Figure 2.2 shows five movies with each users rating represented in stars. The user who rated these movies has not watched all of them, so the movies: *Inception* and *The Departed* are shown as *question mark* to show there is no information about if the user likes these movies or not. Next, the middle of the figure shows the main part of the RS (the RSs engine), which aims to predict the ratings of the unknown items. Then, the RS engine orders the items based on the score, and produces a top-k recommendation, which is shown at the bottom of the figure.

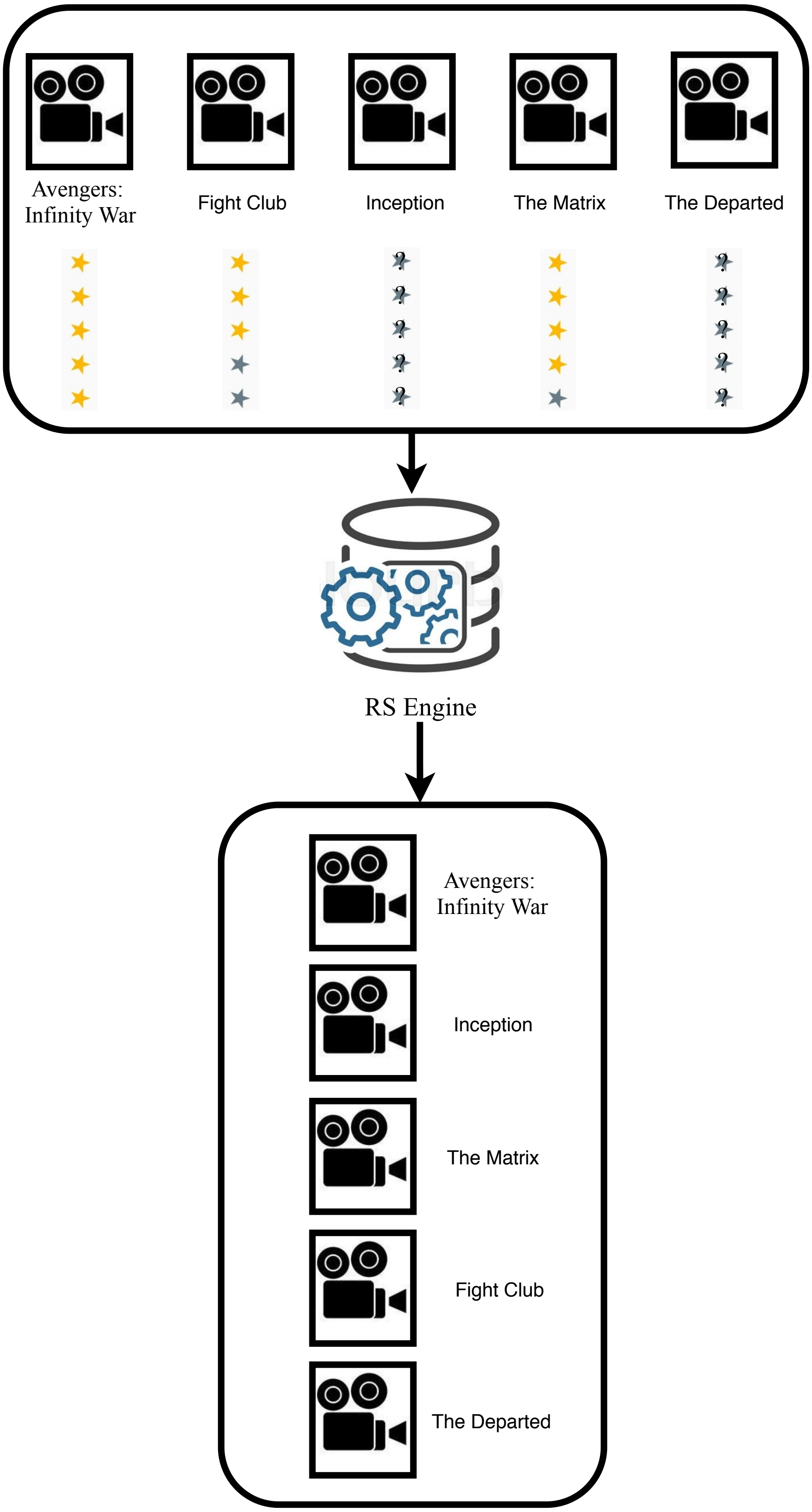


Figure 2.2: Example of a movie top-k recommendation processing to produce recommendations

2.4.2 Package Recommendation

Most RSs produce lists of single recommended items rather than composite recommendations, which are called package recommendations (as mentioned above, in many situations, package recommendations are used in applications such as healthcare planning, travel planning, and course recommendations for students). In addition, generating package recommendations is a challenge because sometimes the recommended package is comprised of different types of items. For example, in healthcare planning, the package may include medication, food items, and blood samples at different times, and in travel packages, people travel as groups or couples. Also, they do not usually visit just one POI, but are likely to visit many.

In TRSs, it is important that the system recommends a package rather than individually recommended items. In TRSs, the recommended package may consist of different types of items such as accommodation, POIs, transportation, and food or restaurant options. Not only packages consist of different items, but also each of these items is subject to certain constraints such as opening times, availability or suitability for different seasons. In addition, another challenge for package recommendations is that different types of items in a package must consider item order, and some items should not appear more than once in a particular package. For example, if we consider a package that represents one day's travel, it should not feature more than one accommodation option although it should feature more than one POI.

More recently, certain researchers have tended to support package recommendations without appearing to consider item constraints. For example, [6, 9, 23, 38, 67, 109] have produced package recommendations although these papers did not consider recommendations for groups. Also, in [38, 67, 109], these papers only focus on recommending packages featuring the same types of items.

In [71], the authors designed a model called Tourist-Area-Season Topic (TAST). This represents different packages and tourists, and the authors proposed a cocktail approach to producing recommendations. However, the shortcoming of this work is that the dataset relied solely on data from travel agents and dealt with packages only, whereas, in fact, many POIs are not usually considered by travel agents, such as parks for instance. In addition, not all packages from travel agents are personalized to suit users, so RSs should produce packages consisting of individual, personalized items for users.

In [60], Interdonato et al. presented a framework that produces recommendation packages based on individual users and ignores user constraints. Therefore, the most important aspect of travel is that many people choose to travel with others and not alone.

In [110], Xie et al. set out their design for CompRec-Trip, an RS that generates package recommendations for travel planning. The advantages of this approach are that it considers both time and financial cost when producing recommendation packages. However, the shortcomings of this research are that (1) it considers only one type of item such as POIs, although, in fact, travel packages consist of different items such as hotels, POIs, and restaurants, etc., (2) the proposed system is only suitable for individual users, (3) considering constraints such as cost is not enough because users and items are also subject to constraints such as opening times.

In [109], Xie et al. proposed a recommender system to produce package recommendations that consider cost although it does not consider groups.

2.4.3 Sequence Recommendation

While sequence recommendations are similar to package recommendations, the order of items is critical in sequence recommendations. Even though some researchers have used different names such as plan or routes, these RSs share the same characteristics of sequence recommendations, which are providing a set of items in a specific order. The most unique and important element of this approach is ordering items in a way that satisfies the user(s). In the tourist domain, based on photos taken by travelers, researchers have built sequence recommender systems for such users, such as in [63].

2.4.4 Plan Recommendation

When items should be consumed between time windows or at a specific time, we need to compare Plan Recommendations (PRs) with other RSs. PRs are considered as combinatorial optimization problems and lacking in data, varieties of taste, large numbers of options; in addition, many other limitations mean that using PRs to solve such problems is a complicated process. For example, a trip plan has a start time and end time, and between these times, the RS chooses items based on different techniques. The related work on PRs is divided into (1) Planner algorithms such as [49], and (2)Planner prototypes such as [104]. The algorithm approach focuses on improving algorithms and running times. The prototype approach deals with planner systems altogether from data collection to building the plan for users.

A few papers consider the constraints from both the user's perspective and the item-based perspective. In [51, 59, 100], they propose a TRS that considers both time and cost constraints but ignores other important constraints. [50], Ge et al. propose a cost-aware RS which combines cost preferences and user interest. This system is based on a latent factor model that connects a set of variables to a set of latent variables. However, considering finance and time constraints alone does not increase the quality of personalized recommendations. In [59], the researchers propose an algorithm based on the knapsack problem to produce travel package recommendations that consider both time and cost.

However, in [41, 53, 63, 115], the authors consider other constraints such as distance, users' interest, and seasons.

In [115], Zhang et al. formulate a TRS to provide package recommendations that maximize user satisfaction while considering various constraints. This study considers specific constraints such as (1) total traveling time, (2) each POI's opening times, (3) travel time between POIs, and (4) the different categories of POIs in a package. This study aimed to identify an optimal solution to provide package recommendations that maximize user happiness. In this work, the authors did not consider groups of users, user constraints, or POI constraints such as room size.

Moreover, Fang et al. proposed a novel framework called the Package-Attraction-based Trip Recommender (PATR) to recommend trip packages with multiple constraints [41]. This method considers time, financial cost, and user's interest.

In [63], Jiang et al. designed a TRS that recommends travel packages considering cost, time, and season. The proposed method was based on multi-source social media which used photos as the dataset. Also, Gionis et al. developed a framework that provides travel package recommendations considering budget, distance, and time constraints [53].

2.4.5 Comparing Between the Recommendations Types

Table 2.3 presents the main differences between the RS types. Firstly, each type of RS is applicable to specific *domains*[1]. For example, Top-k is applicable to movie and music RSs. Secondly, all types of RSs produce more than one item as a recommendation, whereas the Top-k produces only one item. In other words, the item consumed in Top-k is must be only a single item, wherein in other types, the consumption of items is more than a single item together. Third, as the last three types of RSs (Package, Sequence, and Plan) produce more than one item, these items may consist of different types. For example, meal-specific RSs might recommend the specific ingredients, cooking utensils, and equipment needed to cook a certain meal. Fourth, in Sequence and Plan recommendations, the order is important where other items are not required to be consumed in order. Finally, in Plan Recommendations, each item is allocated to a specific time or between specific time windows in which the item must be consumed. For instance, Travel RSs produce recommended plans where each item in the plan should be visited at an allocated time.

Table 2.3: Comparison between the recommendations types

Features	Top-k	Package	Sequence	Plan
Domain	Movie, Music	Trip, Meals	Healthcare	Travel
Items No	1	$1 <$	$1 <$	$1 <$
Items Type	1	$1 \leq$	$1 \leq$	$1 \leq$
Items' Order			•	•
Time				•

2.5 Recommender Systems Techniques

Combinatorial optimization problem [104], several data mining algorithms [73], and automatic clustering [78] are some of the techniques that are used to build RSs. Specifically, we have classified RS techniques into RSs based on previous data and RSs not based on previous data.

2.5.1 Recommendations Based on Previous Data

Most RSs produce recommendations based on user's historical data. In other words, the data on users' consumption or item properties are needed to analyze these data, and RSs calculate the probability that users might like or dislike unconsumed or unrated items. Thus, CF and CB are classified as based on user's historical data. A variety of data have been used to produce recommendations such as those based on images [63, 111], check-in data [113, 115], and tourism companies, etc. , [57]. In addition, many different techniques and frameworks based on data-mining have been proposed to build recommendations [17].

Despite using different dataset or methods, TRSs are unique and different from most RS domains because: (1) most of the datasets are very sparse, (2) most of the time, people travel as groups, and so TRSs should consider this as *Group Recommender Systems* (GRSs), (3) trip budget and time limitations, (4) a variety of user's requirements and preferences, (5) attraction constraints, (6) travel seasons, (7) travelers visit multi-attractions in a single visit, and (8) a trip composed of a set of attractions such as hotels, transportation, cities, and streets.

2.5.2 Recommendations not Based on Previous Data

The previous data does not support RSs to produce recommendations because, in some situations, old data is non-essential and not part of processing recommendations. Namely, some domains in RSs do not rely on consuming user data but rather, are based on user's requirements or specifications.

In addition, this approach is applied in applications where the items required are very rare, or new items of the same type as other items are different, such as when purchasing a car. However, as mentioned above, a 2010 Vauxhall Corsa does not share the same attributes as a 2017 Vauxhall Corsa, for instance. Another example is buying a house, which is a rarely undertaken action, and each person has different requirements.

Important to realize that KB is classified in this approach because KB does not require any historical data. As an illustration, TRSs are unique comparing to RSs because of the reasons mentioned in the previous section (see Section 2.5.1).

Table 2.4 shows the main differences between the different RS techniques. First, data is captured *Explicitly* (E) and *Implicitly* (I) in the methods based on historical data, and the techniques which are not based on historical data use explicitly captured information. Secondly, if the methods are using historical data, these data need to be analysed to produce recommendations. Third, the first type of RS techniques must be provided with enough data to produce recommendations, whereas the second type of methods request only basic information to provide recommendations. Fourth, the former RS techniques are designed to analyse the data, whereas the latter are more dynamic because they are designed to customize the data based on the user's requirements. Finally, the first type of methods suffer from several limitations, such as cold-start users, sparse data, etc.

The next section will explain other problems similar to those affecting TRSs, and explain why the solutions to these similar problems are inadequate for solving TRSs.

Table 2.4: Comparison between the RSs techniques

Features	Based on Previous Data	Not Based on Previous Data
Captured Data	E & I	E
Analysing User data	•	
Amount of Data	Enough	Basic
Dynamic		•
Drawbacks	•	

2.6 Similar Models to TRSs

This section provides an overview of the other models that are similar to TRSs.

2.6.1 Tourist Trip Design Problem

The *Tourist Trip Design Problem* (TTDP), a trip plan for tourists interested in visiting multiple POIs, has been defined by [107], and TTDP represents the TRSs. TTDP is based on tourists having different POIs to visit and limited time for sightseeing; each POI has a set of attributes (e.g., category, location, child-friendly, admission cost, etc.), and each trip has a limited time for sightseeing; therefore, the TTDP should maximize the total score generated when visiting a specific POI (where each POI has different scores).

Moreover, this system not only chooses POIs but also selects the best route(s) between selected POIs. The solutions provided by TTDP should respect traveler's constraints and POI's characteristics [49]. The main objective of TTDP is to maximize tourist *satisfaction*[1] by allowing them to visit high-scoring POIs while taking the user's constraints into account.

Most of the following models partly solve the TTDP where we have introduced an abstract model (see Chapter 3 & 4).

2.6.2 Travelling Salesman Problem

*T*he Travelling Salesman Problem (TSP) relates to finding the shortest route for a salesman to visit all the cities on a list. TSP is similar to TTDP in finding an efficient route, although it has several differences such as not all POIs need to be visited. The extension of TSP is the TSP-with-profits, which is a generalization of the TSP, and it is not necessary to visit all cities which are associated with a benefit of each path. The goal of TSP-with-profits is to maximize the profits and minimize travel costs. However, the TSP-with-profits has similarities to TTDP in terms of finding a route with benefits, although there are many differences, which make TSP-with-profits a special case of TTDP such as considering POI's opening/closing times.

2.6.3 Knapsack Problem

*T*he Knapsack problem (KP) relates to choosing items associated weight and value, in a bag subject to a size limitation; the aim of KP is to maximize the value of the selected items. In other words, KP selects from a set of items, which have specific values and weights, in a bag under a specified maximum weight. In addition. 0-1 KP is an extra restriction on KP which takes the whole item or not; 0-1 KP should consider all parts of the item and not allow an item to be divided. The 0-1 KP is an extension of KP and is very close to TTDP because POIs cannot be divided into parts (i.e. when choosing POIs). However, the differences between KP and TTDP are that TTDP does not consider the path between items while KP has a single constraint (weight).

2.6.4 Vehicle Routing Problem

*T*he Vehicle Routing Problem (VRP) relates to the distribution of goods from warehouses to final users, and it determines the optimal routes which be followed by a fleet of vehicles [102]. The VRP is similar to TSP in terms of visiting all nodes where the VRP delivers goods to all users.

2.6.5 Comparing between the TRSs and other Problems

Table 2.5 shows a comparison between TRSs and other models. First, all models are similar to the TRSs in terms of the start point and endpoint except the KP is not required to be in order or feature different locations. Second, the KP and *Orienteering Problem* (OP) [103] are similar to the TRSs in choosing some selected nodes whereas the TSP and VRP must use all nodes. Third, each node has a value (representing the rating or preference in TRSs), and the TSP and VRP do not consider any value for nodes. Fourth, the KP does not consider any values for the edges where all other problems consider values for edges (representing the distance or cost). Finally, the TRSs has a unique feature, which are the constraints required to personalize recommendations.

Because the most recent contributions to the TTDP are based on OP, we briefly explain the OP's mathematical model in the next section. As the OP represents a special case of TTDP, some extensions of OP are implemented to tackle its drawbacks.

Table 2.5: Comparison between the TRSs and other problems

Features	TRSs (TTDP)	TSP	KP	VRP	OP
Start[1]/*End Points*[1]	•	•		•	•
Selecting Nodes	•		•		•
Node Value	•		•		•
Edge Value[1]	•	•		•	•
Dynamic *Constraints*[1]	•				

2.7 Orienteering Problem Family and Models

The *Orienteering Problem* (OP) is a combination of selected nodes to determine a path between selected nodes, and the aim is to maximize the total score achieved by visiting selected nodes under the limited time budget [54]. Accordingly, the OP is based on an aggregation between two problems: KP and TSP [105].

Recent studies on the TTDP were based on the OP and its extension because the OP is a special case of the TTDP where the travelers start from a point and end at another/the same point within the limited time allocated for sightseeing.

We now list the main extensions of the OP. Several OP-extension variants have been proposed, such as Team OP (TOP), TOP with *Time Windows*[1] (TOPTW), and Time-Dependent OP (TDOP). The goal of TOP is finding P paths which are limited to T_{max}[1] and maximizing the total selected score from the paths. Each path is represented as a day trip, which means that TOP can build a plan for many days rather than only a single day as in OP. [49] classify the OP and its extension in TTDP into a single tour and multiple tour models.

2.7.1 Orienteering Problem

The OP is formulated as follows: Let $G = (V, E)$ be directed weighted graph where $V = \{1, ..., |V|\}$ be a set of nodes where each $n \in V$ representing a *Point of Interest* (POI) in a city, and E be a set of edges between these nodes (POIs). A cost of traveling between two nodes $n, u \in V$ denotes $d_E(n, u)$ and the profit of visiting a node n be S_n . Given a starting node s and terminal node t, and let the $s = 1$ and $t = |V|$. There is a time limitation T_{max}. The aim of the OP is to find a path from s to t within T_{max} and maximizing the total collected profit from visited nodes. For every path from s to t, if the path-visit node is u after n the variable x_{nu} is equal to 1 and 0 otherwise.

$$Max \sum_{n=2}^{|V|-1} \sum_{u=2}^{|V|} S_n \times X_{nu} \tag{2.7.2}$$

$$\sum_{u=2}^{|V|} X_{1u} = \sum_{n=1}^{|V|-1} X_{n|V|} = 1 \tag{2.7.3}$$

$$\sum_{n=1}^{|V|-1} X_{nr} = \sum_{u=2}^{|V|} x_{ru} \leq 1 \tag{2.7.4}$$

$$\forall r = 2, ..., |V| - 1$$

$$\sum_{n=1}^{|V|-1} \sum_{u=2}^{|V|} d_E(n, u) \times X_{nu} \leq T_{max} \tag{2.7.5}$$

$$2 \leq I_n \leq |V| \tag{2.7.6}$$

$$\forall n = 2, ..., |V|$$

$$I_n - I_u + 1 \leq (|V| - 1) \times (1 - X_{nu}) \tag{2.7.7}$$

$$\forall n, u = 2, ..., |V|$$

$$X_{nu} \in \{0, 1\} \tag{2.7.8}$$

$$\forall n, u = 1, ..., |V|$$

The objective function is provided in Equation (2.7.2), and the aim is to maximise the total collected score. Equation (2.7.3) represents a constraint to ensure the path starts from s and ends at t. Equation(2.7.4) is a constraint to ensure that the path is connected and each vertex is visited once at most. Equation (2.7.5) ensures that the total travelling time falls under the time budget T_{max}. Equation (2.7.6), I_n denotes the position of POI n in the path, and the combination of Equations (2.7.6) (2.7.7) prevents subtours.

2.7.2 Team OP

The TOP is formulated as follows: X_{nup} denotes decision variables which are $= 1$ if, in path P a visit to node n occurs after visit to node u and is $= 0$ otherwise. I_{np} denotes the position of POI n in path P. The objective function is Equation (2.7.9) and it aims to maximize the total collected score. Equation (2.7.10) is a constraint to ensure the path starts from s and ends at t. Equation(2.7.11) is a constraint to ensure that the path is connected and each vertex is visited once at most. The Equation (2.7.12) ensure that each path p is connected. Equation (2.7.13) ensures that the total travelling time is under the time budget T_{max} for each path p. The combination of Equations (2.7.14) (2.7.15) prevent subtours.

$$Max \sum_{p=1}^{|P|} \sum_{n=2}^{|V|-1} S_n \times Y_{np} \tag{2.7.9}$$

$$\sum_{p=1}^{|P|} \sum_{u=2}^{|V|} X_{1up} = \sum_{p=1}^{|P|} \sum_{n=1}^{|V|-1} X_{n|V|p} = |P| \tag{2.7.10}$$

$$\sum_{p=1}^{|P|} Y_{rp} \leq 1 \tag{2.7.11}$$

$$\forall r = 2, ..., |V| - 1$$

$$\sum_{n=1}^{|V|-1} X_{nrp} = \sum_{u=2}^{|V|} X_{rup} = Y_{rp} \tag{2.7.12}$$

$$\forall r = 2, ..., |V| - 1, \forall p = 1, ..., |P|$$

$$\sum_{n=1}^{|V|-1} \sum_{u=2}^{|V|} dE(n, u) \times X_{nup} \leq T_{max} \tag{2.7.13}$$

$$\forall p = 1, ..., |P|$$

$$2 \leq I_{np} \leq |V|, \forall n = 2, ..., |V| \tag{2.7.14}$$

$$\forall p = 1, ..., |P|$$

$$I_{np} - I_{up} + 1 \leq (|V| - 1) \times (1 - X_{nup}) \tag{2.7.15}$$

$$\forall n, u = 2, ..., |V|, \forall p = 1, ..., |P|$$

$$X_{nup}, Y_{np} \in \{0, 1\} \tag{2.7.16}$$

$$\forall n, u = 1, ..., |V|, \forall p = 1, ..., |P|$$

2.7.3 TOP with Time Windows

The next equations illustrate the formulation of TOPTW, and each node n is assigned a time window $[O_n, C_n]$. The main difference between TOP and OPTW is that time window allows a POI to be visited only during the specified time window. $X_{nu} = 1$ if visit node n then visit node u and otherwise $X_{nu} = 0$. $Y_{np} = 1$ of the node n is visited in path p; M is a large constant. The objective function in Equation (2.7.17), and the aim maximizes the total collected score. Equation (2.7.18) is a constraint to ensure the path starts from s and ends at t. Equations (2.7.19) and (2.7.20) are constraints to ensure that the path is connected and the timeline of each path p. Equation (2.7.21) ensures that every node is visited once at most, and Equation (2.7.22) ensures that the total travelling time falls under the time budget T_{max} for each path p. The combination of Equations (2.7.23) and (2.7.24) ensures that the visiting time falls within the specified time window.

$$Max \sum_{p=1}^{|P|} \sum_{n=2}^{|V|-1} S_n \times Y_{np} \tag{2.7.17}$$

$$\sum_{p=1}^{|P|} \sum_{u=2}^{|V|} X_{1up} = \sum_{p=1}^{|P|} \sum_{n=1}^{|V|-1} X_{n|V|p} = |P| \tag{2.7.18}$$

$$\sum_{n=1}^{|V|-1} X_{nrp} = \sum_{u=2}^{|V|} X_{rup} = Y_{rp} \tag{2.7.19}$$

$$\forall r = 2, ..., |V| - 1; \forall p = 1, ...|P|$$

$$s_{np} + t_{nu} = s_{up} \leq M \times (1 - X_{nup}) \tag{2.7.20}$$

$$\forall n, u = 1, ..., |V|; \forall p = 1, ..., |P|$$

$$\sum_{p=1}^{|P|} Y_{rp} \leq 1 \tag{2.7.21}$$

$$\forall r = 2, ..., |V| - 1$$

$$\sum_{n=1}^{|V|-1} \sum_{u=2}^{|V|} dE(n,u) \times X_{nup} \leq T_{max} \quad (2.7.22)$$

$$\forall p = 1, ..., |P|$$

$$O_n \leq s_{np} \quad (2.7.23)$$

$$\forall n = 1, ..., |V|; \forall p = 1, ..., |P|$$

$$s_{np} \leq C_n \quad (2.7.24)$$

$$\forall n = 1, ..., |V|; \forall p = 1, ..., |P|$$

$$X_{nup}, Y_{np} \in \{0, 1\} \quad (2.7.25)$$

$$\forall n, u = 1, ..., |V|; \forall p = 1, ..., |P|$$

2.7.4 The limitations of the OP

The limitation of the OP is that the OP does not solve the TTDP in all aspects, and the OP is a special case of TTDP. For example, in terms of choosing POIs and paths between POIs, the OP solves this aspect of the TTDP, but it fails to do so in other aspects of trip design such as considering visiting times, opening and closing times, etc. In addition, the OP is designed based on an optimization problem where the TRSs should be built to support personalization.

2.8 Group Recommender Systems

This section aims to provide an overview of Group Recommender Systems (GRSs) because most people travel as a group. In addition, GRSs deals with diverse users who may not share similar preferences [7, 52]. In some domains, users participate in certain activities together as groups [52, 61]. However, the need for GRSs has become imperative [35,61], and GRSs are required to process many varied scenarios such as producing recommendations for TV programs for friends or choosing the best restaurant for colleagues [52]. In addition, Salam et al. mention that the job of GRSs is to recommend items to a group where these items reflect the *preferences*[1] of the group as one, and these items should be reasonable and acceptable to all group members [10,40,88]. In addition, it is likely in groups of users, that individual users have a range of preferences that result in conflicting needs, and also, it might be commonplace that a group has some users who are less easy to satisfy [88].

The main limitations of the existing works are the techniques that have been used to split groups, which are based on clustering algorithms. However, we have designed an algorithm based on *Ant Colony Optimization* to take the decision on splitting the group during the recommendation-building stage (see Chapter 6).

The biggest challenge in GRSs is building a *group profile*[1] to provide recommendations suitable for all group members. Different approaches have been designed to deal with GRSs. The GRS approach consists of aggregation methods which are classified (for more details see Chapter 6)

2.9 Conclusion

In this chapter, we have presented a detailed background on and a comprehensive review of RSs. The chapter has provided an analysis of the limitations and drawbacks of existing works, which can be summarized in terms of four areas:

- **Building recommendations based on data**.

  TRSs suffer from a lack of data and sparse data; most studies are based on data where the personalization accuracy does not achieve satisfactory results.

- **Customizing tour trip based on flexible constraints**.

  RSs are based on personalization where each user can customize their constraints. Thus, because the OP is designed based on the optimization problem, flexibility in changing constraints and personalisation is not supported.

- **Maximizing and maintaining travelers' satisfaction level**.

  The existing works do not optimize traveler's journeys based on their effort spent on gaining access to their preferences, such as wasting time and traveling time between POIs.

- **Minimising the conflicts between group members and maximizing the individual satisfaction level for group members**.

  The existing study proposed spitting the group into subgroups based on clustering algorithms.

In this thesis, we have developed three main models in Recommender Systems. Table 2.6 shows the main limitation on the related works comparing the proposed models. In next chapters, we will discuss the limitation of each related works in the related chapters (ICDM in Chapter 3, HM in Chapter 5, and GTTRM in Chapter 6).

Table 2.6: The drawing an analogy between related works to the proposed models

Proposed model	Related study	Limitations of related study
Item Constraints Data Model	[21, 74, 92]	• Inflexibility on constraints • Limitation on personalization • Limitation on constraints on different times
Happiness Model	[90, 96, 106]	• Limitation on wasting time • Less considering efficiency in consuming time • Limitation on personalising connections
Group Tourist Trip Design Problem	[62, 75]	• Limitation on different users' constraints • Limitation on splitting the group

From the analysis above, we can conclude that the most appropriate RS-based model for our current work is *Knowledge-Based*. In addition, the most relevant OP-based models to our current work are: (1) the TOP is relevant to the ICDM (Chapter 3), (2) the TOPTW is relevant to the HM (Chapter 5), and (3) the MCMTOPTW is relevant to the GTTRM (Chapter 6).

Chapter 3

Item Constraints Data Model

This chapter presents the Item Constraints Data Model (ICDM); a novel approach that is able to build *personalized*[1] *tour trip*[1] based on different types of *constraints*[1]. We developed the ICDM to tackle the limitations of existing works. The novelty of our ICDM: (1) is its ability to customize and handle large number constraints, (2) its ability to set up constraints at specific times, and (3) its ability to reduce the search space dimension.

3.1 Introduction

When visiting a specific destination, travelers face the problem of deciding which particular POIs to visit [49]. Similarly, tourists have to deal with the problem of planning a trip when they travel to a new destination for the first time [22]. Tourists plan their trip based on the information available on digital sources (e.g., travel websites, maps, travel blogs, etc., [14, 22] or traditional sources such as books, friends, and magazines). In other words, tourists who travel to a new destination have many options in terms of which POIs to visit, which ones they should visit first, and in which order should they visit them. These interdependent tasks represent a challenge because several different combinations of constraints need to be considered [14, 48, 49, 63] (e.g., weather conditions, opening/closing times, time available for

[1] See Glossary for definitions of italicized words

sightseeing, etc.). An additional constraint is when tourists are interested in visiting all the attractions in large cities such as London, Paris, or Barcelona within a limited timeframe [104]. However, the limitations concerning time and budget tend to be the constraints that tourists consider the most when selecting the most attractive places to visit [104]. Finally, processing such plans in order to accurately match tourist's preferences is a very complicated and time-consuming process [8, 14, 20].

Therefore, to solve the problems described above, we have built a system that can handle user's constraints and *preferences*[1], and provides an e-tourism-service-based, whole-trip planning solution [64]. Over the past decade, advanced digital applications [64] and personalized electronic tourist guides [104] have supported travelers by building *trip recommendations*[1]. Under these circumstances, *Recommender Systems* (RSs) represent an effective solution because RSs reduce the complexity of the information that must be searched for via the Internet [64]. The main advantage of RSs is personalizing recommendations to match user's needs [64]. RSs support users by guiding users to customized items consisting of a huge number of options which are based on decision-support systems [20, 42].

Travel Recommender Systems (TRSs) make the planning processing more straightforward [48]; TRS's aim is to match specific *user's needs*[1] with the specific characteristics of a particular trip [20,48]. In particular, most TRSs focus on only well-known attractions [36, 63], which reduces recommendation effectiveness [36]. However, the most widely used technique in TRSs is based on traveler's previous data, and the most pressing problem facing TRSs is the sparsity of such data. For example, [111] mentions that it is difficult to learn effectively from sparse data. However, [14] emphasizes that a critical aspect of designing TRSs is having rich enough data on users and POIs. After all, most TRSs do not use an entirely personalized trip because only popular POIs have been considered, only a list of POIs rather than a plan is recommended, no flexibility in changing requirements is defined, and satisfying user's requirements is not completely achieved [36].

We have classified the *constraints*[1] involved in building TRSs into three categories: (1) Item Constraints, (2) Connection Constraints, and (3) Trip Constraints. First, Item Constraints are defined as conditions that apply to an item (or items), and this type of constraint is based either on user preferences or item requirements (e.g., fee price is an example of an Item Constraint based on user preferences, and *time window*[1] is an example of an Item Constraint based on item requirement). Second, Connections Constraints represent how users can move from one POI to another based on connection limitations such as transportation time. Third, Trip Constraints are defined as conditions applied on the whole trip such as trip budget or trip length.

3.2 Related work

The Tourist Trip Design Problem (TTDP), which is a trip plan for tourists who wish to visit multiple POIs, has been defined by [107]. The TTDP has been partially solved by several models (e.g. the Orienteering Problem, the Team Orienteering Problem, the Orienteering Problem with Time Window, etc.). Here, we mainly analyse the existing approaches that solve TTDP and compare the current works.

3.2.1 (Team) Orienteering Problem

The Orienteering Problem (OP) which is a combination of selected nodes to determine a path between selected nodes, and the OP aim to maximize the total score collected by visiting the selected nodes within a specified time budget, is considered to be a particular case of TTDP [54]. Accordingly, the OP which was introduced in [103], is based on an aggregation of two problems: the knapsack problem' and the travelling salesman problem [105]. Indeed, the OP is NP-hard while optimal solutions could be feasible with a small number of nodes [49]. However, [99] formulates the OP as an integer programming problem, which is a decision problem with a maximization or minimization objective.

The Team OP (TOP), introduced by [31], is an extension of the OP. The TOP's aim is to consider p paths, where each path represents a day trip when the TOP is limited by T_{max}[1] and maximizes the total collected scores. Moreover, the main difference between the OP and TOP is only that the TOP deals with multi-day trips whereas the OP only deals with one-day trips. Indeed, at first glance, the TOP is like the OP by repeating the algorithm p times; however, it does not always produce true outcomes. Because different node orders affect the total collected scores, the TOP is more challenging than the OP. To put it differently, to optimize p paths, it must put into account that each path should go into a set of POIs where are near to each other because of ignoring selecting a set of POIs in a direction, leads to the non-optimal traveling time between POIs.

Several studies have proposed algorithms to solve the OP and TOP [21, 74, 92]. However, the limitations of the OP and TOP are that they do not fulfill basic tour trip plan conditions such as opening/closing times. The OP is designed to cover only time budget (Trip Length) constraints, even though other constraints must also be considered, such as time windows for example. Also, the TOP is designed to take into account another important constraint: multi-day tour trips.

3.2.2 (Team) Orienteering Problem with Time Window

The OP with Time Window (OPTW) and TOP with Time Window (TOPTW) have been designed to consider the time window (opening/closing times) represented by $[O_i, C_i]$ for each node i, and nodes can visit them during the Time Window (TW). The main challenges in OP and TOP are handling (1) time budgets and (2) multi-day trips. Besides these challenges, the OPTW and TOPTW are more difficult to optimize because of the limitations in the availability of nodes at specified times.

The OPTW and TOPTW have solved the main drawback of the OP and TOP. However, designing a tour trip plan should consider more constraints to customize plans based on user's needs (e.g., weather conditions, traffic jams, or financial budgets). Any one of these aspects could represent constraints that should be taken into account for some particular users but not others.

3.2.3 Other extensions of the OP

Many researchers have introduced other models that are extensions of the OP. Every extended model adds a new constraint that previous models do not consider in their routing design.

Traditionally, the travel time between the two locations is mostly affected by the level of traffic congestion (or its absence). The Time-Dependent OP (TDOP) is designed to consider congestion levels which might affect the travel time between two POIs. Besides, the TDOP with Time Windows (TDOPTW) is introduced to combine the OP with two additional constraints: Time Dependent and Time Windows. The TDOPTW considers some restrictions that are applicable to the TTDP by maximizing the tourist's satisfaction. Another model introduced in [106] deals with multiple paths, and this model, which is called Time Dependent Team OP with Time Windows (TDTOPTW), is based on TDOPTW.

As each extension model of the OP considers a new constraint or combines two individual constraints, the need arises to design a general model that can customize a tour trip based on the specific types of constraints that are relevant to each particular user.

3.3 Item Constraints Data Model

The Item Constraints Data Model (ICDM) is designed to tackle the limitations encountered in existing studies. Specifically, the ICDM is designed to deal with data and constraints, and match user's requirements and preferences on their tour trip. Firstly, we have classified data into two types (1) *static data* and (2) *dynamic data*, to match item data with users constraints (Section 3.3.1). *Static data* represents data values that remain constant over time while *dynamic data* represents data values that do change from one time to another.

We have classified such constraints into *Hard Constraints* (HC) and *Soft Constraints* (SC) (Section 3.3.2). A mathematical model of ICDM is described in Section 3.3.3. Finally, we illustrate the features of our ICDM and provide some examples of how the model deals with data and constraints.

3.3.1 Data in ICDM

Our ICDM deals with multiple layers of data such as opening/closing times, weather conditions, entrance fees, etc. To build a general model that is able to personalize tour trips for users, we have defined *static* and *dynamic* data.

In *static data*, the values remain unchanged over time. In other words, these data have only one value that remains constant over time. For example, the British Museum's entrance fee is free at all times, so this data value does not change if travelers visit the museum on different days. Second, *dynamic data* does vary over time. For example, the opening/closing times for the British Museum are different from one day to another, so the data values vary over time. Figure 3.1 presents the two types of data in real-life examples: weather conditions, opening/closing times (Time Windows), and location. Location represents a *static data* value because it remains constant with time, whereas opening/closing times and weather conditions (heavy rain, clear, or sunny) represent *dynamic data* values as they vary over time.

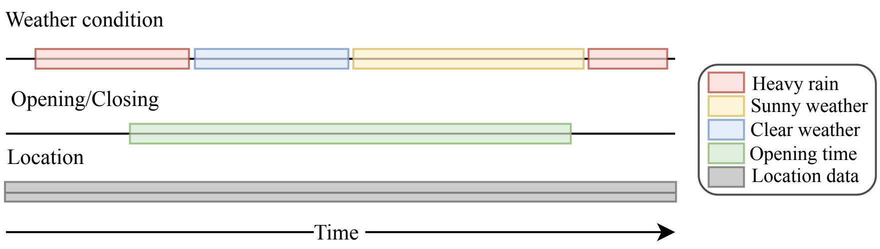


Figure 3.1: Presenting different types of data on the timeline

Dynamic data can be modelled in the ICDM based on any division of time, such as second, minute, or hour.

3.3.2 Constraints in ICDM

We have formalized constraints into HC and SC based on the user's priority level. Topaloglu et al. have divided constraints into HC and SC for solving the nurse scheduling problem, and they define them as "*The hard constraints must be satisfied at all costs, whereas the soft constraints may be violated to generate a workable solution*" [101]. We have designed our ICDM based on these types of constraints. All HCs have to be satisfied whereas each SC has a value which represents the percentage of the user's satisfaction level for each POI. Then, based on their preferences, a user can choose a constraint to be either HC or SC. For example, a user can assign weather conditions to be either HC or SC according to their preference. If the user prefers to visit some POIs under clear weather conditions, then the user will consider weather conditions as HC. On the other hand, if the user accepts to visit some POIs regardless of the weather condition, then the user will consider weather conditions as a SC.

3.3.3 Mathematical model

We define the ICDM as follows: u is denoted as a user who might have n constraints (HC and/or SC). HC is a set of hard constraints, $hc^m \in HC$, where $m = 1, 2, \ldots, |HC|$, SC is a set of soft constraints, $sc^v \in SC$, where $v = 1, 2, \ldots, |SC|$. The constrains are defined according to the following equations:

$$HC_{pti} = \prod_{m=1}^{|HC|} hc^m_{pti} \tag{3.3.1}$$

$$SC_{pti} = \text{Aggregation methods (see Table 3.1)} \tag{3.3.2}$$

$$\sum_{v=1}^{|SC|} W_v = 1 \tag{3.3.3}$$

Equation (3.3.1) shows how all the HCs are calculated, which is represented by HC_{pti}, in item i on day p at time t. All HCs must be satisfied together. If one of the HCs is not satisfied, then this will make the total value zero. In contrast, a SC indicates a specific level of satisfaction and meeting it is optional.

Equation (3.3.2) computes user's degree of satisfaction based on their SCs, which is represented SC_{pti}, in item i on day p at time t. The closer the result of Equation (3.3.2) is to one (1), the higher the satisfaction level is in relation to more constraints. W_v in Equation (3.3.3) represents the weight of the SC $\#v$, where the total equals one (1). The next equation aggregates these two equations into a single value.

$$S_{pti} = HC_{pti} \times SC_{pti} \tag{3.3.4}$$

To calculate the value that represents the degree of satisfaction for user u in each item i on day p at time t based on user's constraints.

The main idea of the model is to embed a variety of user-specific constraints into a single value which is S_{pti}, by aggregating the two Equations (3.3.1) and (3.3.2).

Table 3.1: Aggregation methods for SC

Method Name	Description	Equation
Sum	Calculate the sum of all elements in SC	$\sum_{v=1}^{\lvert SofCon \rvert} W_v \times sc_{pti}^{v}$
Least Misery	Take the minimum value of SC	$Min(sc_{pti}^{v})$
Most Pleasure	Take the maximum value of SC	$Max(sc_{pti}^{v})$
Multiplicative	Multiplies each SC value	$\prod_{v=1}^{\lvert SofCon \rvert} W_v \times sc_{pti}^{v}$

All the constraints the user would like to be considered will be absorbed into S_{pti}, reducing the dimensionality of the data by aggregating all constraints' values into a single value that represents all constraints. Also, the ICDM model will reduce searching time because the number of constraints required to match with the trip is reduced.

3.3.4 Integration of Data and Constraints into ICDM

Data in the ICDM is presented in two types and four forms. First, the *static data* isn't affected by time; they can be presented in one form (see Table 3.2), and Features are presented as the attributes of items. Secondly, the *dynamic data* can be presented in three forms: (1) items and times based on a feature, (2) items and features based on a moment (can be any part of the time, examples are seconds, minutes, and hours), and (3) features and time based on an item. Table 3.3 shows an example where each item has a value for Feature 1 at a specific time, so we have x tables, based on the number of features. The second form presents the items and features based on a specific time; for each time we might have different values for each feature, and Table 3.4 shows an example for items and features based on T_1, and we need to have p tables for each time. The third form is presented such that each item has some features and has different values at different times. Table 3.5 shows an example for features and time for $Item_1$ where we need n tables for each item.

To explain the ICDM using an example, a user is planning to visit a city that has three POIs ($I = \{i_1, i_2, i_3\}$), and each POI has four features: (1) entrance fee (f_1),

Features	$Item_1$	$Item_2$	$\dots$	$Item_n$
F_1	v_{11}	v_{12}	$\dots$	v_{1n}
F_2	v_{21}	v_{22}	$\dots$	v_{2n}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
F_x	v_{x1}	v_{x2}	$\dots$	v_{xn}

Table 3.2: Data representation in form 1

Time	$Item_1$	$Item_2$	$\dots$	$Item_n$
T_1	$V_{f_1t_1i_1}$	$V_{f_1t_1i_2}$	$\dots$	$V_{f_1t_1i_n}$
T_2	$V_{f_1t_2i_1}$	$V_{f_1t_2i_2}$	$\dots$	$V_{f_1t_2i_n}$
$\vdots$	$\vdots$	$\vdots$	$\dots$	$\vdots$
T_p	$V_{f_1t_pi_1}$	$V_{f_1t_pi_2}$	$\dots$	$V_{f_1t_pi_n}$

Table 3.3: Data representation for F_1

Features	$Item_1$	$Item_2$	$\dots$	$Item_n$
F_1	$V_{f_1t_1i_1}$	$V_{f_1t_1i_2}$	$\dots$	$V_{f_1t_1i_n}$
F_2	$V_{f_2t_1i_1}$	$V_{f_2t_1i_2}$	$\dots$	$V_{f_2t_1i_n}$
$\vdots$	$\vdots$	$\vdots$	$\dots$	$\vdots$
F_x	$V_{f_xt_1i_1}$	$V_{f_xt_1i_2}$	$\dots$	$V_{f_xt_1i_n}$

Table 3.4: Data representation for T_1

Time	F_1	F_2	$\dots$	F_x
T_1	$V_{f_1t_1i_1}$	$V_{f_2t_1i_1}$	$\dots$	$V_{f_xt_1i_1}$
T_2	$V_{f_1t_2i_1}$	$V_{f_2t_2i_1}$	$\dots$	$V_{f_xt_2i_1}$
$\vdots$	$\vdots$	$\vdots$	$\dots$	$\vdots$
T_p	$V_{f_1t_pi_1}$	$V_{f_2t_pi_1}$	$\dots$	$V_{f_xt_pi_1}$

Table 3.5: Data representation for $Item_1$

(2) opening/closing time (f_2), (3) weather condition (f_3), and (4) rating for each POI (f_4), and the user has one day ($P\{p_1\}$) (six hours) for sighting ($T = \{t_1, t_2, \dots, t_{|T|}\}$ the six hours could be divided into minutes based on the user's preferences). Figure 3.2 shows the ICDM data in the example where $V_{f_1t_1i_1}$ represents the value of f_1 in i_1 at time t_1.

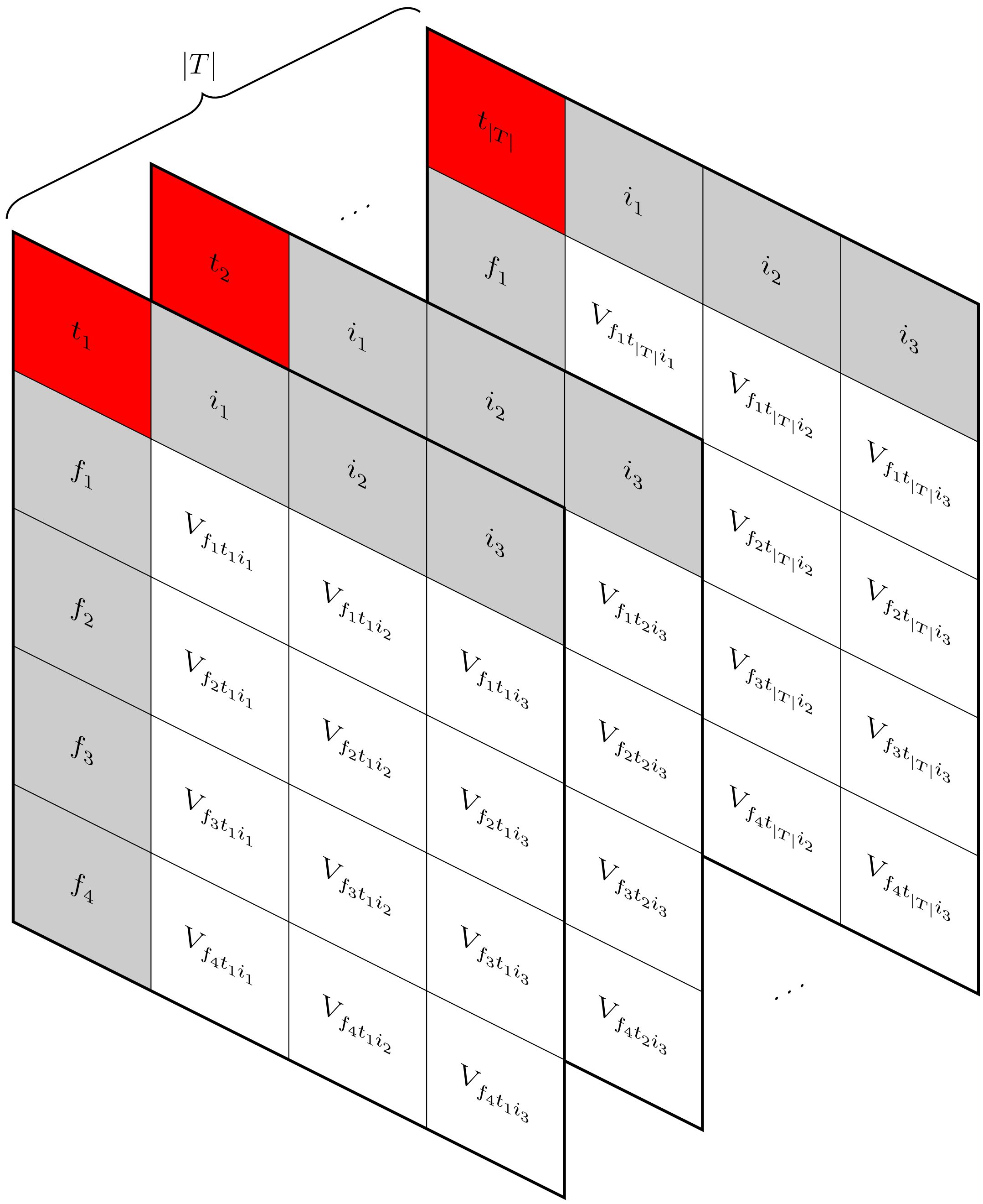


Figure 3.2: An Example of Data in ICDM

In addition, the user has two constraints: (1) a SC on the entrance fee that is visiting any POI with free entrance, and (2) a HC on weather conditions which is visiting an outdoors POI when it is not raining. Traditionally, opening/closing time is a base constraint that must be considered (we will illustrate the *Base Constraints* in Section 3.4), and rating (preferences) for each POI is considered. In order to apply these constraints into the ICDM, we use the Equations 3.3.1 and 3.3.2. In our example, $HC = \{hc^1, hc^2\}$ and $SC = \{sc^1, sc^2\}$ where the HC has two constraints and the SC has two constraints. Figures 3.3 and 3.4 present the HC and SC.

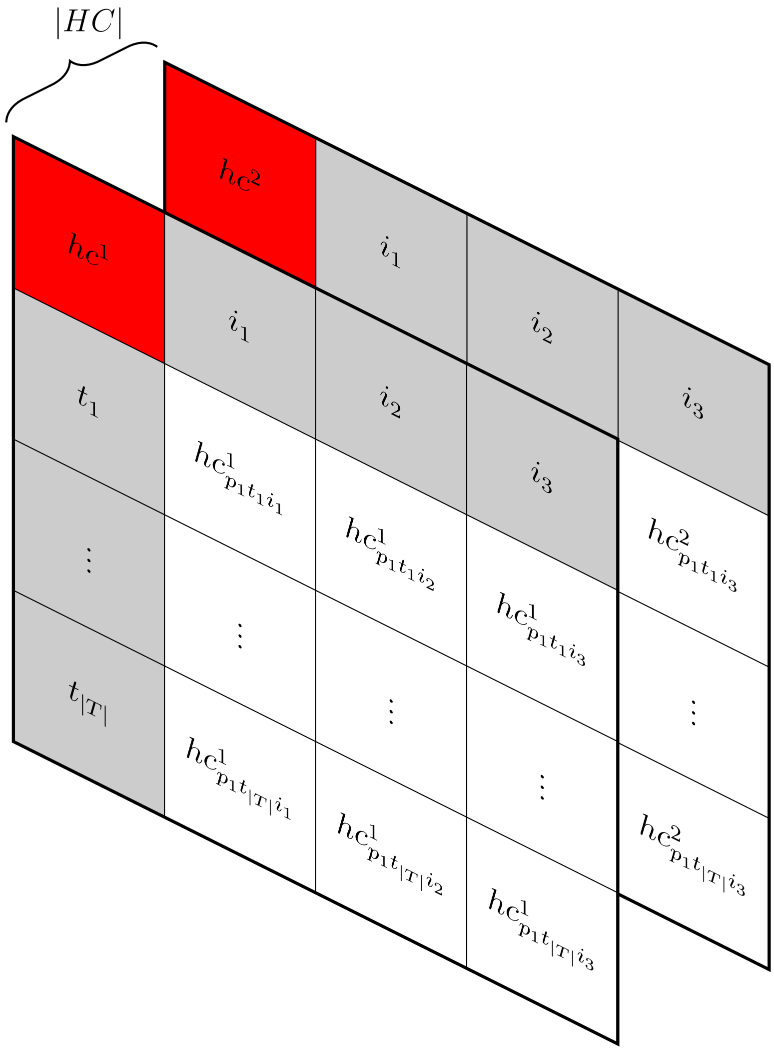


Figure 3.3: An Example of HC in ICDM

Figures 3.5 and 3.6 highlight the cells which will be consider in the Equations (3.3.1) and (3.3.2). Figure 3.7 illustrates the results of the aggregation of the HC's elements. Figure 3.8 shows the results of the aggregation of SC's elements. Finally, Figure 3.9 shows the final data presentation in the ICDM as will be explained in the next section.

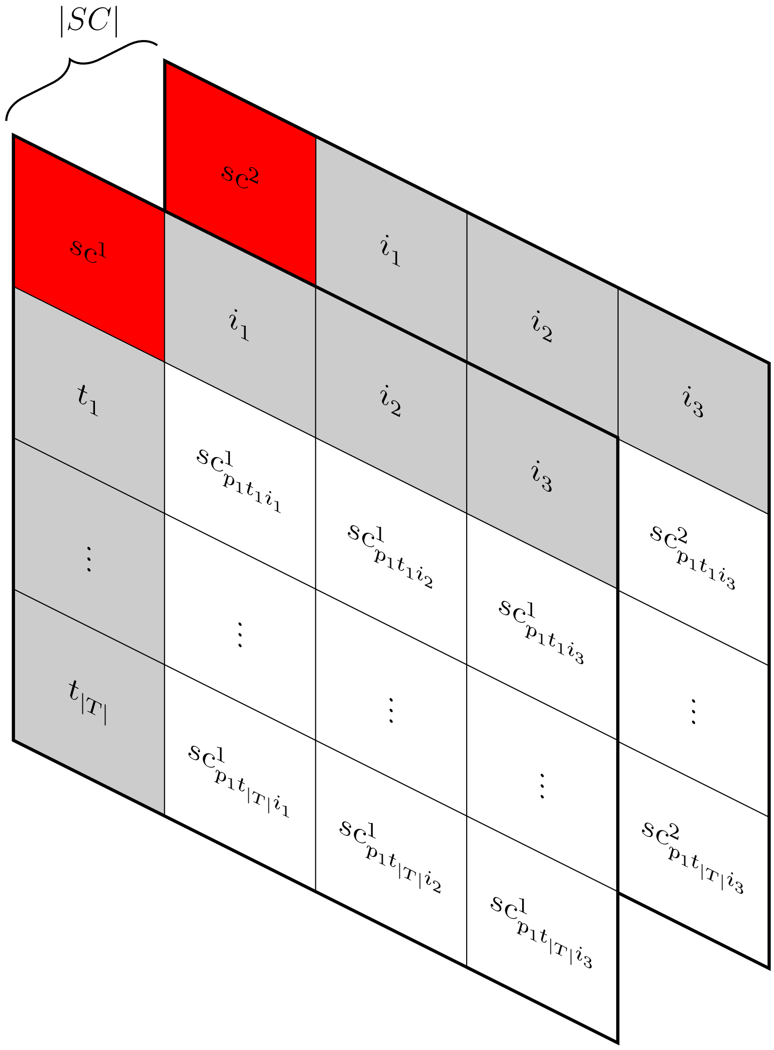


Figure 3.4: An Example of SC in ICDM

h

3.3.5 The OP Model constraints with ICDM

The ICDM has some powerful features. Firstly, the ICDM reduces the search space and helps optimization algorithms to become faster. Secondly, the ICDM can help optimization algorithms to choose the most appropriate items for users.

Reduction of the search space

The main feature of the ICDM is that it reduces search space by aggregating different constraints into a single value. The aggregation in Equation (3.3.4) shows how the

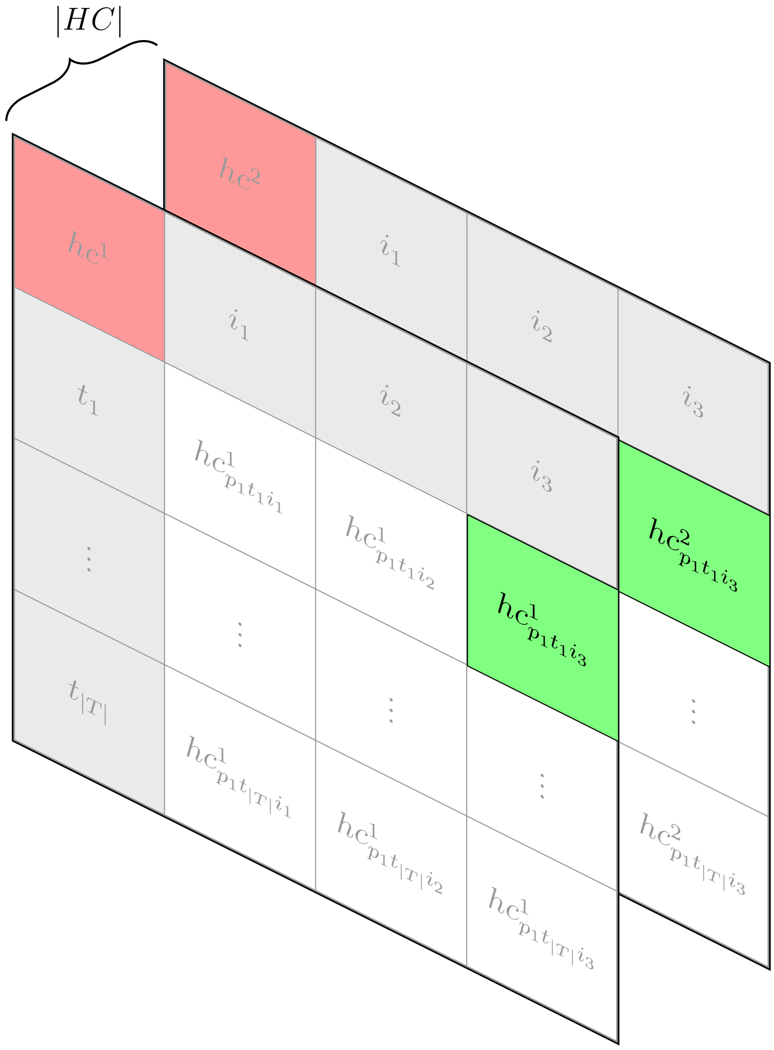


Figure 3.5: An Example of the Equation (3.3.1) in ICDM

HC and SC are combined into a single value, which is represented by S_{pti}. Figures 3.5 and 3.6 show how the ICDM aggregates constraints and data into a single value as shown in Figure 3.9.

Figure 3.9 shows an example of the final data representation after applying all the HCs and SCs. As shown in Figure 3.9, we have reduced the searching space from $|HC|+|SC|$ tables into one table, and we have retained the information about all the values in the final data representation.

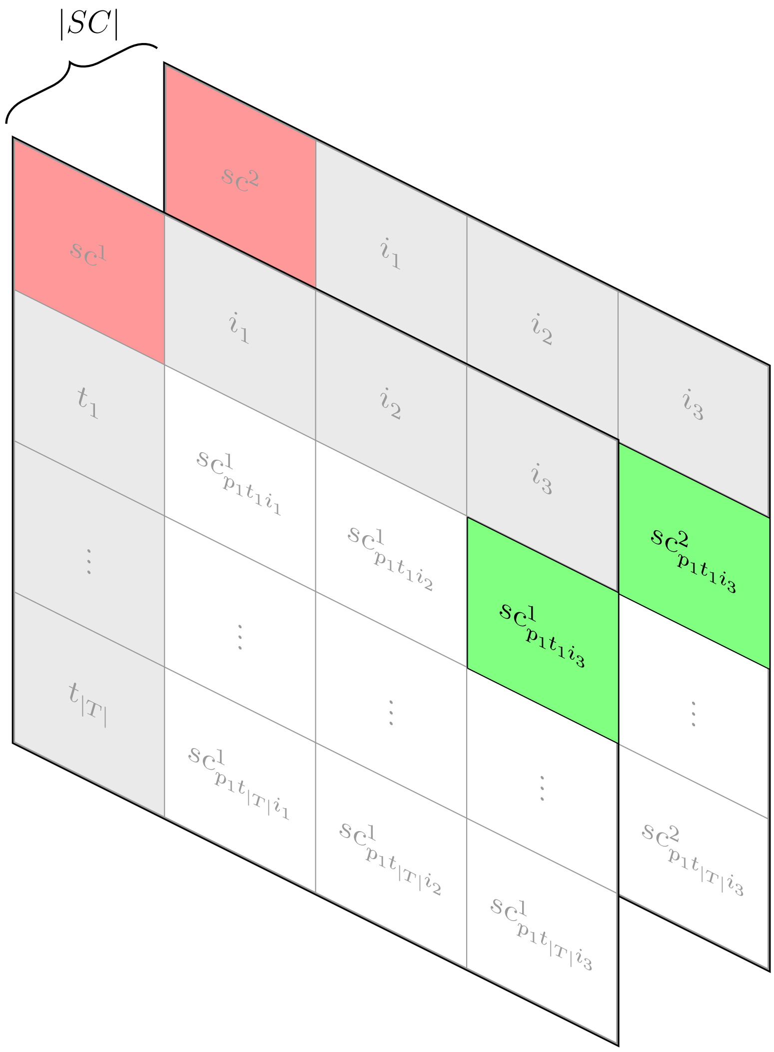


Figure 3.6: An Example of the Equation (3.3.2) in ICDM

Supporting optimization algorithms in taking decision

Not only does the ICDM reduce the search space but it also supports the algorithms to choose items. The ICDM represents the data in a form that gives indicators to optimization algorithms as to the most appropriate items for users. In other words, each value of S_{pti} provides the optimization algorithms with an indicator on which POIs are highly recommended.

HC	i_1	i_2	i_3
t_1	$HC_{p_1 t_1 i_1}$	$HC_{p_1 t_1 i_2}$	$HC_{p_1 t_1 i_3}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$t_{\lvert T \rvert}$	$HC_{p_1 t_{\lvert T \rvert} i_1}$	$HC_{p_1 t_{\lvert T \rvert} i_2}$	$HC_{p_1 t_{\lvert T \rvert} i_3}$

Figure 3.7: An Example of the finalization of HC in ICDM

SC	i_1	i_2	i_3
t_1	$SC_{p_1 t_1 i_1}$	$SC_{p_1 t_1 i_2}$	$SC_{p_1 t_1 i_3}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$t_{\lvert T \rvert}$	$SC_{p_1 t_{\lvert T \rvert} i_1}$	$SC_{p_1 t_{\lvert T \rvert} i_2}$	$SC_{p_1 t_{\lvert T \rvert} i_3}$

Figure 3.8: An Example of the finalization of SC in ICDM

3.4 Using ICDM to describe other models

As the ICDM is a general model that can solve TTDP, we show how the ICDM is able to handle the OP, TOP, OPTW, and TOPTW. However, the ICDM is not limited to solving these models only, it can go beyond that.

Before explaining how the ICDM handles the models, we classify the constraints

ICDM	i_1	i_2	i_3
t_1	$S_{p_1 t_1 i_1}$	$S_{p_1 t_1 i_2}$	$S_{p_1 t_1 i_3}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$t_{\|T\|}$	$S_{p_1 t_{\|T\|} i_1}$	$S_{p_1 t_{\|T\|} i_2}$	$S_{p_1 t_{\|T\|} i_3}$

Figure 3.9: An example of final data representation from ICDM

for a particular trip into three categories: (1) Trip Constraints, (2) Connection Constraints, and (3) Item Constraints. First, Trip Constraints are defined as all conditions that could be applied generally to the whole trip such as budget. These could be all conditions that are implemented in a specific category rather than a whole trip. An example of a Trip Constraint in a category is accepting to visit any POI under the parks category, only if it is free'. Connection constraints are defined over conditions when the user is moving from one POI to another. They are based on connection limitations such as transportation time or budget. Finally, Item Constraints are defined over conditions applied to a POI, such as distance from the start point.

In addition, all these three constraints exist in two types: *Base Constraints* (BC) and *Extra Constraints* (EC). BCs are defined when all conditions must be applied to a trip plan to make the trip plan applicable (valid). For example, any trip plan must have at least one POI to visit. In contrast, ECs are defined as any constraint that does not affect a trip plan's applicability if it is removed (e.g. children-friendly POI, fee less than £10).

The importance of the ECs is that they enable the ability to personalize constraints for different travelers. Table 3.6 shows a comparison between the ICDM with other models. It is clear that none of the models in Table 3.6 support (this symbol − represents not supporting) any EC in all three types of constraints except the ICDM, which supports the EC in Item Constraints; further, none of these models support ECs in Trip Constraints and Connection Constraints.

Instead, the ICDM can support algorithms to customize trips for travelers based on their constraints. For example, users, who travel to a coastal city would like to consider weather conditions when visiting outdoor POIs. In addition, the OP and TOP models do not support the BC in Item Constraints because one of the default constraints is not applied (opening/closing time). However, the ICDM can deal with Multi-Time Window (MTW), that is multiple opening/closing intervals over a single day. For example, a POI is open from the morning until the afternoon, then open from 4:00 PM until 9:00 PM. Furthermore, the TOP, TOPTW, and ICDM support multi-day trips' shown with on (M) in Table 3.6 in column BC in Trip Constraints.

Table 3.6: Comparing between tourist trip planning models based on different constraints

Model abbreviation	Trip Constraints		Connection Constraints		Item Constraints	
	BC	EC	BC	EC	BC	EC
OP	•	-	•	-	-	-
OPTW	•	-	•	-	•	-
TOP	M	-	•	-	-	-
TOPTW	M	-	•	-	•	-
ICDM	M	-	•	•	MTW	•

3.4.1 Comparison Between Constraints

Table 3.7 shows the main differences between the various types of constraints of the ICDM. Firstly, the HCs and BCs must be satisfied, and the BC also must be applied to all plans which means the BC usually corresponds to the most important constraints to be satisfied, such as opening/closing times. Secondly, the SC represents the satisfaction level. Unlike the BC, the HC, SC, and EC are created based on user's choices. To sum up, the BC and EC represent the two categories of the constraints; the former represents natural limitations and the latter user's preferences, respectively. In addition, the BC accepts only HCs, whereas the EC can accept either HCs or SCs.

Table 3.7: Comparison between HC, SC, BC, and EC

Feature	HC	SC	BC	EC
Must satisfied	•		•	
Must apply in all plans			•	
Satisfaction level		•		
Based on user preferences	•	•		•

3.4.2 Implement Popular Models with ICDM

The ICDM can personalize and produce trip tours based on user's constraints. In addition, the main feature of the ICDM is its capacity to be generalized to implement existing models and solve their problems in a uniform manner.

To understand how the ICDM implements other models, in Table 3.6, we identify the constraints in these models. All models in Table 3.6 have common constraints which are: (1) start/end point for each trip and (2) each day in the trip has certain times for sightseeing (T_{max}). Table 3.8 shows different types constraints that are applied in the ICDM. First, all models have one SC which is Score (S) while the Equation (3.3.2) is applied, with $W_v = 1$ for Score. Moreover, the OPTW and TOPTW have one HC which is Time Window (TW), which is described in Equation (3.3.1). However, the OP and TOP do not have any HCs (i.e. no constraints such as opening/closing times are considered), and we put the $HC(i, p, t)$ values into Equation (3.3.4) all equal to 1 (for all i, p, and t) because there are no HCs (in these models) to implement.

Table 3.8: Apply other models in ICDM

Model abbreviation	HC	SC
OP	-	S
OPTW	TW	S
TOP	-	S
TOPTW	TW	S

3.5 Conclusion

We have designed and developed a novel model Item Constraints Data Model (ICDM) that can manage n item constraints. The ICDM has been designed to reduce multi-data dimensionality into a 2-D data structure. Another critical point is that the ICDM has been designed to support and solve the Tourist Trip Design Problem (TTDP), which is a trip plan for tourists who are attracted to visiting multiple POIs, as defined in [107]. In the next chapter, we discuss the experiments performed on the ICDM and their results.

The ICDM's contribution lies in: (1) its ability to personalise constraints that are applicable to each POI, (2) its ability to place constraints into two categories based on their importance level, (3) its ability to place various constraints at any specific time interval, and (4) its ability to deal with constraints whose values are time-dependent.

Chapter 4

Flexible Travel Recommender Model

4.1 Introduction

Following the introduction of the ICDM (in Chapter 3), we then introduce the Flexible Travel Recommender Model (FTRM) by generalizing the OP. Furthermore, we have developed an algorithm based on the *Ant Colony Optimization* (ACO) that achieves comparable outcomes to the state-of-the-art algorithms. As a result, this chapter illustrates a novel approach that can deal with different type of *constraints*[1] and customize user's trips according to their specific *wishes*[1].

In detail, the FTRM is built on top of the ICDM whereas the ICDM deals with different types of constraints. Figure 4.1 shows the interactions between the FTRM and ICDM. First, the ICDM formalises the data and constraints into an understandable formula suitable for general-purpose algorithms before the FTRM builds a tour trip based on the ICDMs output (ICDM formula).

[1]See Glossary for definitions of italicized words

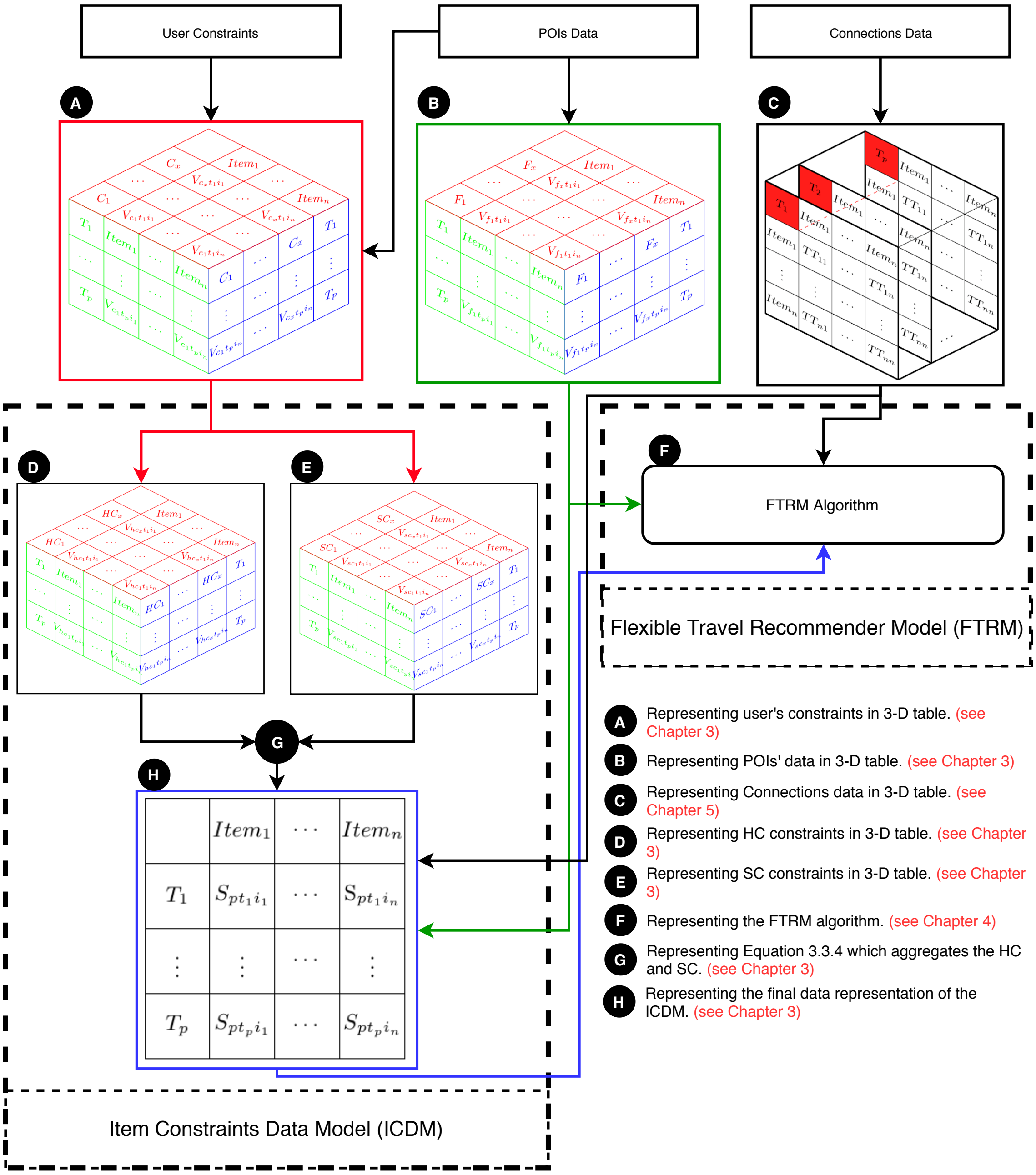


Figure 4.1: An overview of the interaction between the FTRM and ICDM

4.2 Flexible Travel Recommender Model

The Flexible Travel Recommender Model is a generalization model for the OP where travelers can personalize their tour trip based on their constraints and *preferences*[1]. The constraints can be defined as a condition on the trip, whereas preferences represent how much the user likes these POIs. The main difference between the FTRM and the OP and their extensions, such as OP, TOP, OPTW, TOPTW, etc., is to customise constraints based on the user's preferences.

4.2.1 Problem description and formulation

The FTRM can be defined as follows. Let $G = (V, E)$ be a undirected weighted graph where $i \in V$ and $i = 1, ..., |V|$ be a set of nodes represent a *Point of Interests* (POIs) in a city, and E be a set of edges between these nodes. A cost of travelling between two nodes $i, j \in V$ denotes D_{ij} and the profit of visiting a node i be S_i. For each trip, M denotes a set of trip days where $d \in M$ and $d = 1, ..., |M|$, and each day of the trip has some periods of time denotes $t \in d$ where $t = 1, ..., |d|$. The starting node is s and the terminal node is e where $s = 1$ and $e = |V|$. In addition, ST_i denotes the staying time in the node i where the time limitation is T_{max}.

$$Max \sum_{d=1}^{|M|} \sum_{i=1}^{|V|} \sum_{j=2}^{|V|} \left(\left(\frac{\sum_{t=1}^{|d|} X_{dtij}}{ST} \right) \times S_{dti} \right) \tag{4.2.1}$$

$$\sum_{j=2}^{|V|} X_{d11j} = 1; \forall d = 1, \ldots, |M| \tag{4.2.2}$$

$$\sum_{i=1}^{|V|-1} \left(\prod_{t=2}^{|d|} X_{dti|V|} \right) = 1; \forall d = 1, \ldots, |M| \tag{4.2.3}$$

$$\sum_{d=1}^{|M|} \sum_{i=1}^{|V|} \left(\prod_{t=1}^{|d|} X_{dtir} \right) \leq 1; \forall r = 1, \ldots, |V| \tag{4.2.4}$$

Equation (4.2.1) presents the objective function of FTRM where X_{dtij} denotes the decision variable on day d at time t from node i to node j, and S_{dti} presents the result of Equation (3.3.4) in Chapter 3. Equation (4.2.2) ensures the start node is the first node in V on each day of the trip, and Equation (4.2.3) ensures the last node of each day of the trip is the last node in V. Equation (4.2.4) ensures that each node is visited only once.

$$\sum_{d=1}^{|M|} \left(\prod_{t=n}^{t_x = n + D_{rh} + ST_h} X_{dtrh} \right) = \sum_{d=1}^{|M|} \left(\prod_{t=t_x+1}^{t_x + D_{hm}} X_{dthm} \right) \leq 1, \tag{4.2.5}$$

$$\forall h = 2, \ldots, |V| - 1; \forall d = 1, \ldots, |M|; \forall n \in 1, \ldots, |d - 3|; \forall r, m = 1, \ldots, |V|$$

$$\sum_{i=1}^{|V|-1} \sum_{j=2}^{|V|} \left(\left(\frac{\sum_{t=1}^{|d|} X_{dtij}}{\sum_{t=1}^{|d|} X_{dtij}} \right) \times (D_{ij} + ST_i) \right) \leq T_{max} \tag{4.2.6}$$

$$\forall d = 1, \ldots, |M|$$

$$X_{dtij} \in \{0, 1\} \tag{4.2.7}$$

$$\forall i, j = 1, \ldots, |V|; \forall d = 1, \ldots, |M|; t = 1, \ldots, |d|$$

Equation (4.2.5) is a constraint that ensures that the path is connected. For example, Figure 4.2 shows how Equation (4.2.5) works, where $d = 1$, $n = 1$, $r = 1$, $h = 5$, $m = |V|$, $D_{15} = 2$, $D_{5|V|} = 3$, and $ST_5 = 4$. Also, $X_{1115} = 1, X_{1215} = 1, X_{1315} = 1, X_{1415} = 1, X_{1515} = 1$, and $X_{1615} = 1$ represent the transfer time from the start point (node #1) to node #5 and the visiting time for node #5. Moreover, $X_{175|V|} = 1$, $X_{185|V|} = 1$, and $X_{195|V|} = 1$ represent the transferring time from node #5 to the end point (node #$|V|$). Equation (4.2.6) prevents the total time in each trip day exceeding T_{max}.

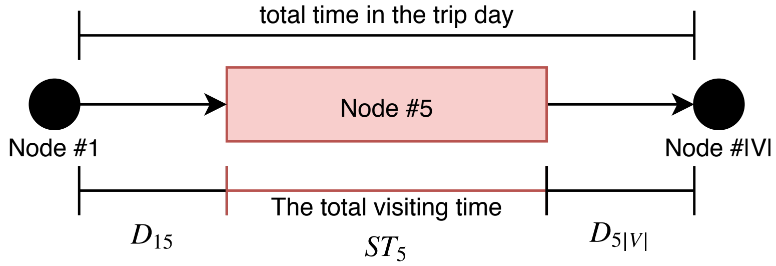


Figure 4.2: Overview of Equation (4.2.5) how it works

4.2.2 Solution approaches

To solve the FTRM, the ICDM has been designed (see Section 3.3 in Chapter 3). Also, an algorithm has been developed, which is the Ant Colony Optimization (see Section 4.3).

4.2.3 Benchmark instances

We have selected the most popular public datasets that are available for the OP, TOP, OPTW, and TOPTW models. Table 4.1 shows all datasets that have been employed in the experiments.

Firstly, The OP datasets have about 89 different scenarios (instances), and these instances are grouped into two groups based on the authors. The first group is based on [103], and the second group is based on [30].

Secondly, The TOP datasets have around 330 different scenarios, and the datasets ($Dataset_1$ to $Dataset_3$, $Dataset_5$, and $Dataset_6$) are based on the OP's datasets with adding more than one day trip.

Thirdly, The OPTW and TOPTW datasets about 105 different scenarios. The first three datasets are based on one of the vehicle routing problems with time windows datasets.

Table 4.1: List of each problem and all datasets

Problem	Dataset Name	Reference	Number of instances	Number of items $\|V\|$
OP	$Dataset_1$ (Tsiligirides 1)	[103]	18	32
	$Dataset_2$ (Tsiligirides 2)		11	21
	$Dataset_3$ (Tsiligirides 3)		20	33
	$Dataset_4$ (Chao 1993)	[30]	26	66
	$Dataset_5$ (Chao 1996)		14	64
TOP	$Dataset_1$ (Chao 32)	[31]	3 × 18	32
	$Dataset_2$ (Chao 21)		3 × 11	21
	$Dataset_3$ (Chao 33)		3 × 20	33
	$Dataset_4$ (Chao 100)		3 × 20	100
	$Dataset_5$ (Chao 66)		3 × 26	66
	$Dataset_6$ (Chao 64)		3 × 14	64
	$Dataset_7$ (Chao 102)		3 × 20	102
OPTW & TOPTW	$Dataset_1$ (c*10,r*10, and cr*10)	[106]	29	100
	$Dataset_2$ (c10,r10, and cr10)		29	50
	$Dataset_3$ (c20, r20, and cr20)		27	100
	$Dataset_4$ (pr01 - pr10)		10	48 to 288
	$Dataset_5$ (pr11 - pr20)		10	48 to 288

4.3 Ant Colony Optimization

Many algorithms have been proposed (e.g. the Heuristic Algorithm, Greedy Algorithm, Genetic Algorithm, Local Search, Branch-and-cut algorithm, Particle Swarm Optimization, Simulated Annealing, Branch-and-price algorithm) to solve the models (see Table 4.1). This is because the OP has been admitted an approximation scheme [8] because the running time for the OP overgrows as the underlying graph grows.

Swarm Intelligence algorithms are successful approaches for complex problems [91]. Swarm intelligence (SI) is an artificial intelligence (AI) method which is designed based on intelligent multi-agent systems by observing the behaviour of social insects [15]. Examples of the multi-agent systems in social insects are ants, termites, bees, and wasps. Swarm Intelligence algorithms include Genetic Algorithms (GA), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Differential Evolution (DE), Artificial Bee Colony (ABC), and Cuckoo Search Algorithm (CSA) [2]. The most feature of SI algorithms is self-organized where is no need for a central control [15].

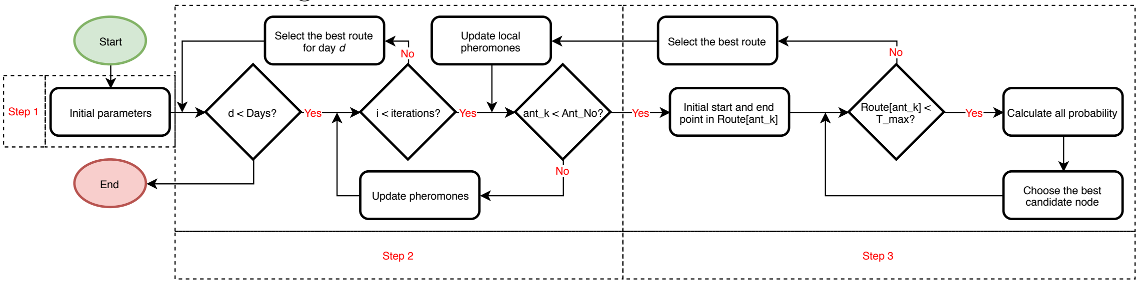
The most successful SI algorithms are ACO and PSO [15, 91], and [28] emphasis that the number of publications in ACO is at least five times more than the other SI algorithms, which indicates that the ACO has been attracted the researchers' attention in different disciplines.

The Ant Colony Optimization (ACO), introduced by Dorigo et al. [39], is an algorithm inspired by the life of ants. The main concept of the ACO is based on multiple agents that represent real-life ants where these agents communicate with each other via the pheromones they produce. To explain, ants produce pheromones to lead other ants to a food source where a higher concentration of the pheromone indicates a significant food source.

We have chosen the ACO algorithm because the ACO is unaffected by problem instance, problem size, and degree of constraint [70]. Moreover, the advantage of ACO over other techniques is that robustness and flexibility [15]. Also, the ACO technique has been tested and provides better running-time results compared to other algorithms [95]. Moreover, ACO approaches have been widely implemented to solve a range of different problems such as the Traveling Salesman Problem [72], Scheduling [32], Digital Image Processing [94], Clustering [108], Routing Algorithm [86], and so on.

We have adjusted the ACO to solve the FTRM; Figure 4.3 shows a flowchart of the ACO's processes. We divide the ACO algorithm into three steps. First, the ACO's Initial parameters are listed based on Table 4.2. Second, the ACO is controlled by adjudicating the loops based on the initial step. The third and most important step is where the ACO releases the ants to find the best route.

Figure 4.3: Overview of the ACO flowchart



We have conducted experiments to determine the iteration number for the ACO. Important to realize that running time is a critical factor especially when the problem is NP-hard. Figure 4.4 shows the average running time for different iteration numbers where is the minimum running time is 18 milliseconds, and the maximum is 41286 milliseconds. We have chosen ten times for the iteration parameter.

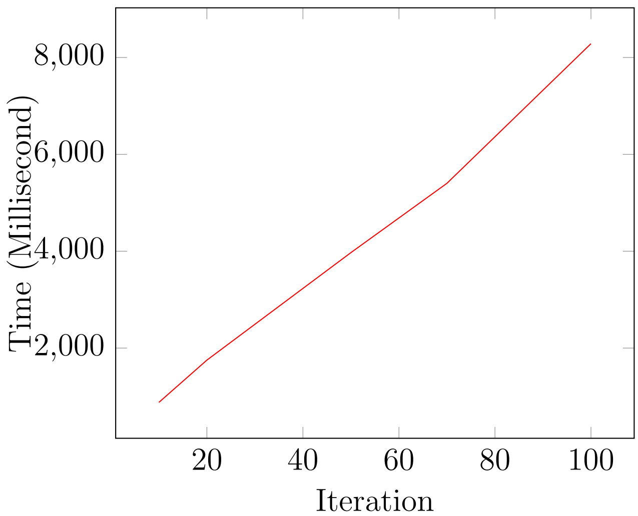


Figure 4.4: The average running time for different iteration numbers

Here, *Alpha* and *Beta* values represent the importance of the score and rate of score to distance (see Equation (4.3.9)). In addition, choosing the optimal value for the *Alpha* and *Beta* values are critical because the performance of the algorithm is based on these values. Therefore, we have used the tuning method that derives the parameters from zero to until reaching the ultimate gain. We have conducted several different experiments (each scenario in each dataset has been processed over 200 times for different values for the *Alpha* and *Beta*) to determine the ACO's best performance across all the datasets. Figures 4.5 and 4.6 present the total score (the score has been normalized where 1 presents the highest total scores for all datasets) of all datasets in all scenarios for different values of *Alpha* and *Beta*. We have chosen the optimal value for *Alpha* and *Beta* to perform better in all datasets where it is clearly the performance of the algorithm over all datasets is the best: $Beta = 5$ and *Alpha* (all values are shown in dark red in Figure 4.5 and 4.6). In other words, any value of *Alpha* and *Beta* in the dark red section will achieve the same results over all the datasets.

Figure 4.5: Overview of the ACO's performance based on different values of *Alpha* and *Beta*

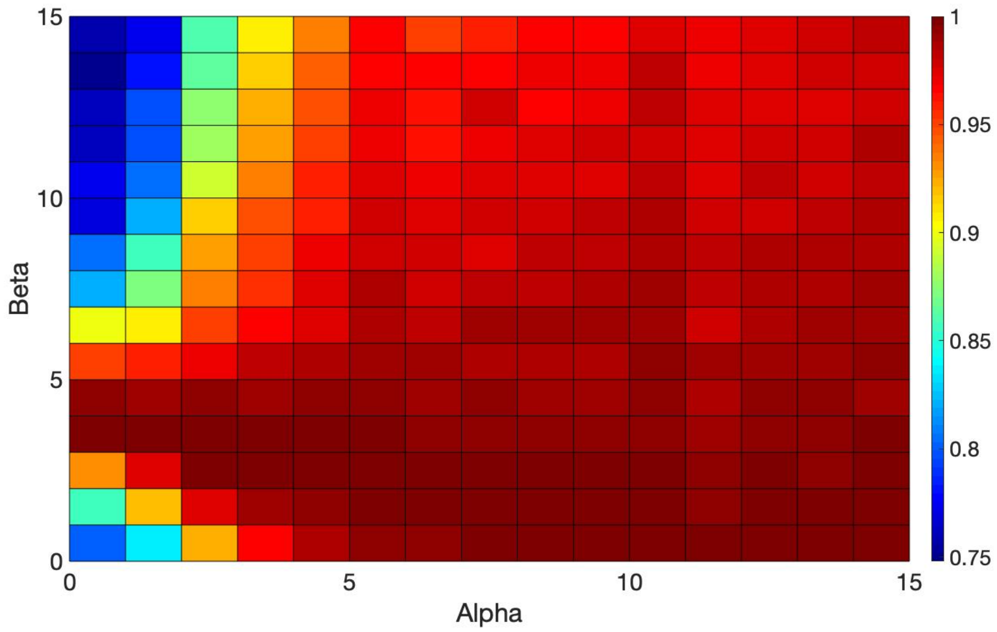


Figure 4.6: Overview of the ACO's performance based on different values of *Alpha* and *Beta*

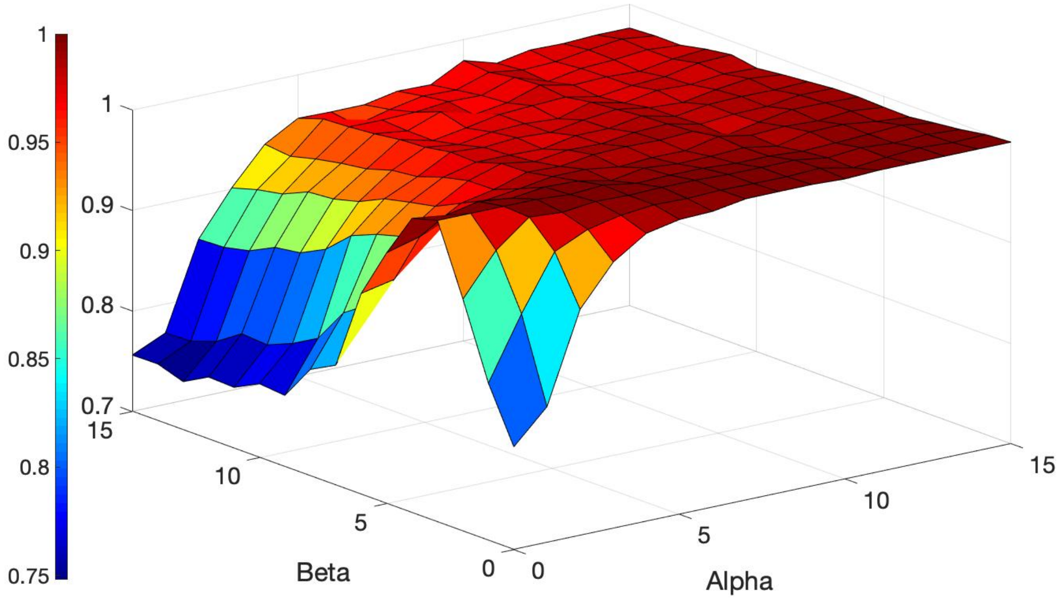


Table 4.2: Initial parameters for the ACO in the first step

Parameter	Initial Value	Description
α	4	The value of *Alpha* presents the importance of *Tau*
β	2	The value of *Beta* presents the importance of *Eta*
ρ	0.1	The value of pheromone evaporation
Ant_No	200	Number of ants
Iterations	10	Number of iteration
NodeSize	Number of nodes	
T_{max}	Total sighting time	
$\eta_{i,j}$	Based on Equation (4.3.8)	The *Eta* presents the rate *score(j)* to *distance(i,j)*
$\tau_{i,j}$	Allocate 1000 value	The *Tau* presents the *Pheromones* level from i to j
$\delta_{i,j}$	Allocate 0 value	The *Delta* presents the maximum total path scores use i to j

The fitness function of ACO is the Equation 4.2.1, and the ACO features two steps to update the pheromone trail because the two update steps is helping *Ants* to find better route in faster time. The first update is called *update local pheromones* in step two (see Figure 4.3); after releasing an ant, it checks that the ant finds a better score for the path found, so the Equation (4.3.10) shows the update for delta for all nodes that been allocated into the better path. After that, the *Tau* is updated based on Equation (4.3.11). The second update is after all ants have been released, and the update is based on Equation (4.3.12).

$$\eta_{i,j} = \frac{Score(j)}{Distance(i,j)} \tag{4.3.8}$$

$$P_{i,j} = \frac{(\tau_{i,j})^{\alpha}\,(\eta_{i,j})^{\beta}}{\Sigma\left((\tau_{i,j})^{\alpha}\,(\eta_{i,j})^{\beta}\right)} \tag{4.3.9}$$

$$\delta_{i,j} = Max(\delta_{i,j}, Ant_k(i,j)) \tag{4.3.10}$$

$$\tau_{i,j} = (1-\rho) \times \tau_{i,j} + \delta_{i,j} \tag{4.3.11}$$

$$\tau_{i,j} = \rho \times \tau_{i,j} + (1-\rho) \times \delta_{i,j} \tag{4.3.12}$$

4.4 Experimental results

We have conducted two experiments (on existing models and features of the ICDM), and the primary purposes of the investigations are to (1) illustrate the ICDMs ability to solve other models, and (2) provide a wide range of situations for the ICDM to handle.

We performed all our experiments on a laptop computer equipped with an Intel Core i5 (1.6GHz) processor with 8 GB RAM, running on macOS (Version 10.14.3).

4.4.1 Applying ICDM to solve existing models

We present computational results illustrating how the ICDM solved the OP, OPTW, TOP, and TOPTW datasets (see the list of all datasets we tested in Table 4.1). We used the *Ant Colony Optimization* that we developed (see Section 4.3).

OP benchmark instances

We have applied the OP to our model to test the ICDMs ability to provide solutions. Figures 4.7 and 4.8 show the results of the $Dataset_1$ and $Dataset_2$ where we present the gap between our results and the state-of-the-art using the red-colored scale. In addition, Figures 4.10 to 4.13 show the results of $Dataset_3$ compared to $Dataset_5$.

Figures 4.7 to 4.13 present our results compared to the state of the art. Each figure represents a dataset where each dataset features different scenarios (a scenario represents a user plan to travel to a city, which represents different POIs, with their own constraints and preferences where each scenario represents a different user and possible different cities), which are labeled above the box. The values inside the boxes represent the total score, which is generated by the user visiting different POIs, where the boxes are assigned a different colour depending on the gap between the result and the state of the art (each figure shows in the right side of the figure a scale which represents the different colours based on the gap between our results and the existing state-of-the-art works).

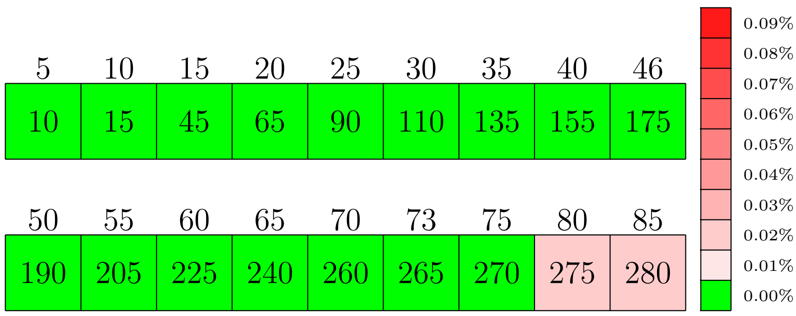


Figure 4.7: The results of $Dataset_1$ for the OP

Figure 4.7 shows the results of $Dataset_1$ where in most of the scenarios, we achieved results comparable with the state-of-the-art (green boxes). The results of these two scenarios that failed to reach a comparable level with the state-of-the-art only failed by 0.02% (red boxes).

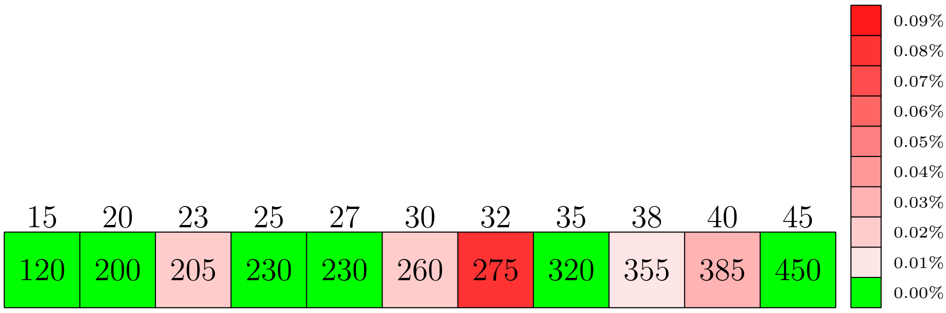


Figure 4.8: The results of $Dataset_2$ for the OP

Figure 4.8 shows the varied performance for $Dataset_2$ where the worst-case result was 0.08% and 54% of the scenarios achieved an optimal solution (green boxes). However, each dataset featured a different number of POIs and different scenarios, and some of these datasets featured POIs closer together. The results in $Dataset_2$ are less satisfactory than the results of $Dataset_1$ because: (1) there are 34% fewer POIs, and (2) the Start/End point location is different.

First, $Dataset_1$ provides more options for the algorithm to choose ($Dataset_1$ provides more POIs than $Dataset_2$). Second, Figure 4.9 shows the locations of POIs for both $Dataset_1$ and $Dataset_2$ where the red points and the black points represent the start/end points for $Dataset_1$ and $Dataset_2$. The main difference between the datasets is the location of the start/end points: in $Dataset_1$ these are at the centre of the map while in $Dataset_2$, these are the bottom of the map. For these reasons, the $Dataset_2$ performs more poorly than $Dataset_1$. The main effect of the start/end location is selecting nodes (POIs) where the algorithm might not choose the best POIs in the city because the start point might be far from these nodes which have high scores.

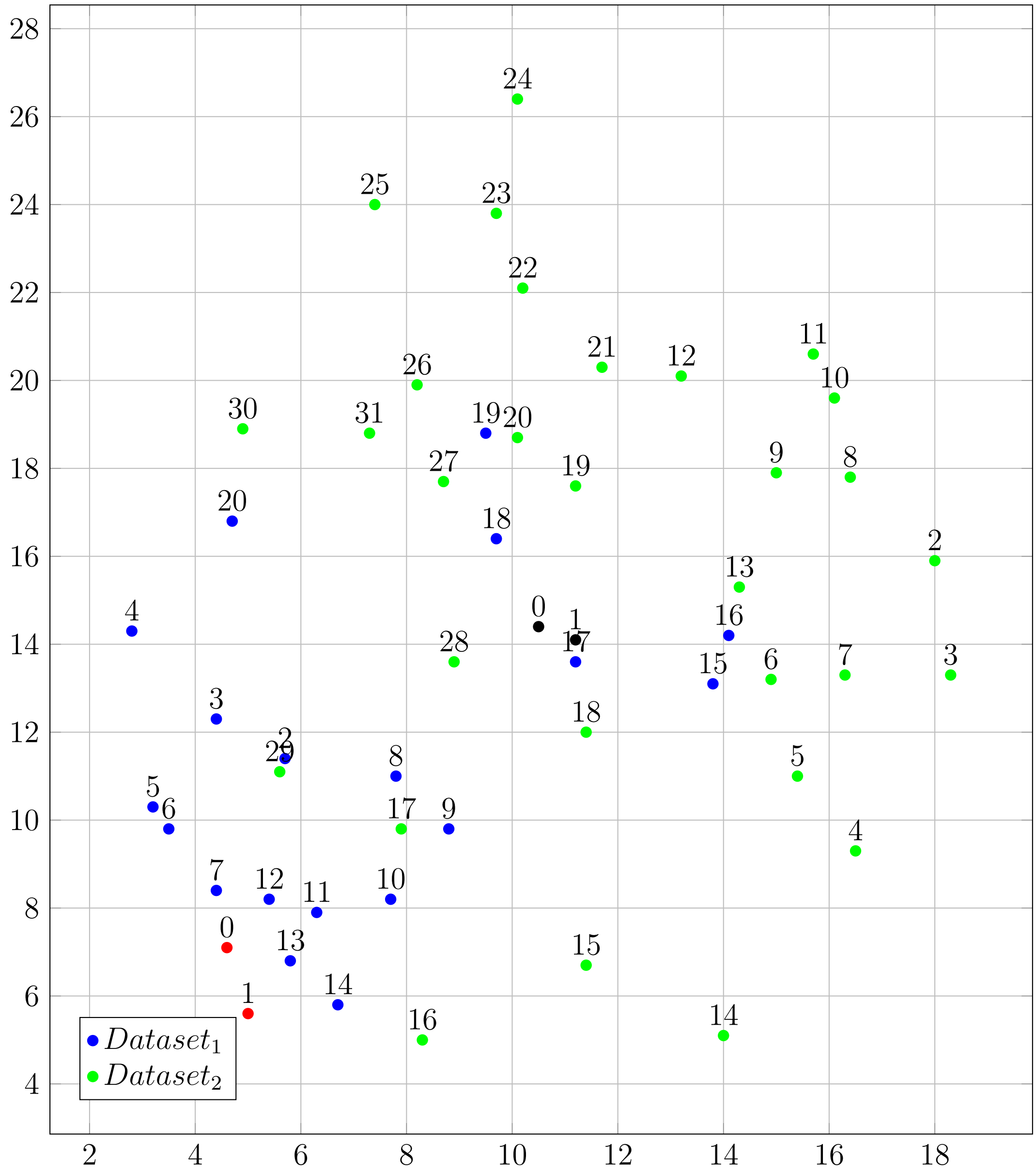


Figure 4.9: Comparing the OP's $Datasets_1$ and $Dataset_2$

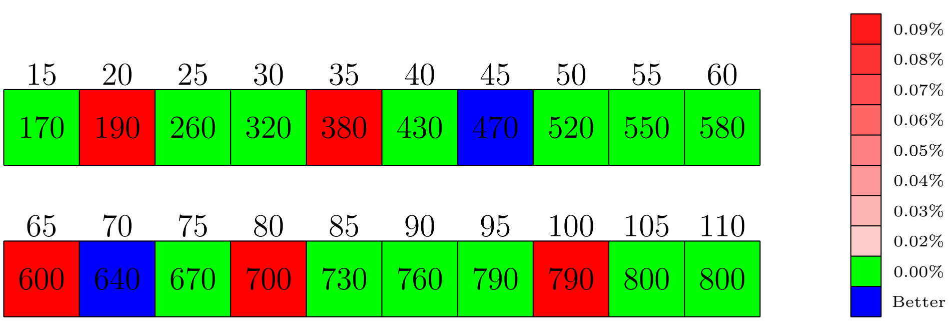


Figure 4.10: The results of $Dataset_3$ for the OP

Figure 4.10 shows that about 65% of the scenarios perform at the same level as the state-of-the-art works for $Dataset_3$. In addition, our algorithm performed better (collecting more scores than the existing works e.g. compared to [74]) than the existing works in the two scenarios (in blue). Finally, $Dataset_3$ performs better than $Dataset_1$ and $Dataset_2$ because (1) the POIs are closer to each other, and (2) the trip length is longer.

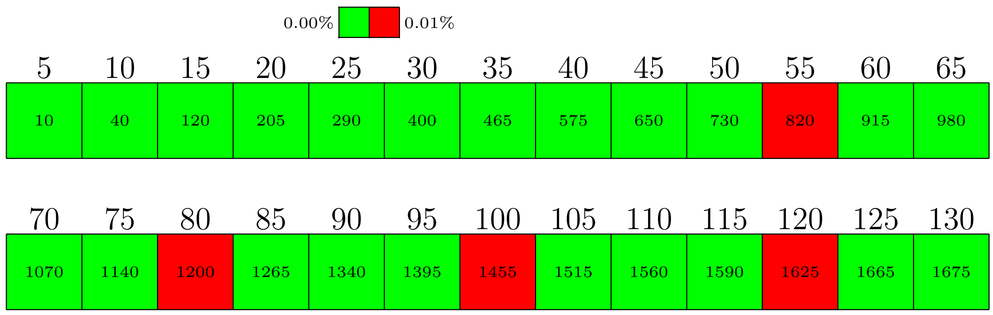


Figure 4.11: The results of $Dataset_4$ for the OP

Figure 4.11 shows that about 84% of the scenarios perform at the same level as the state of the art works for $Dataset_4$. However, 16% of our scenarios perform to within a 0.01% gap between those of state-of-the-art works. The performance of $Dataset_4$ is better than the other the OPs datasets because (1) the POI locations are organised in lines (as seen in Figure 4.12), and (2) the start/end points are located in the centre of the map.

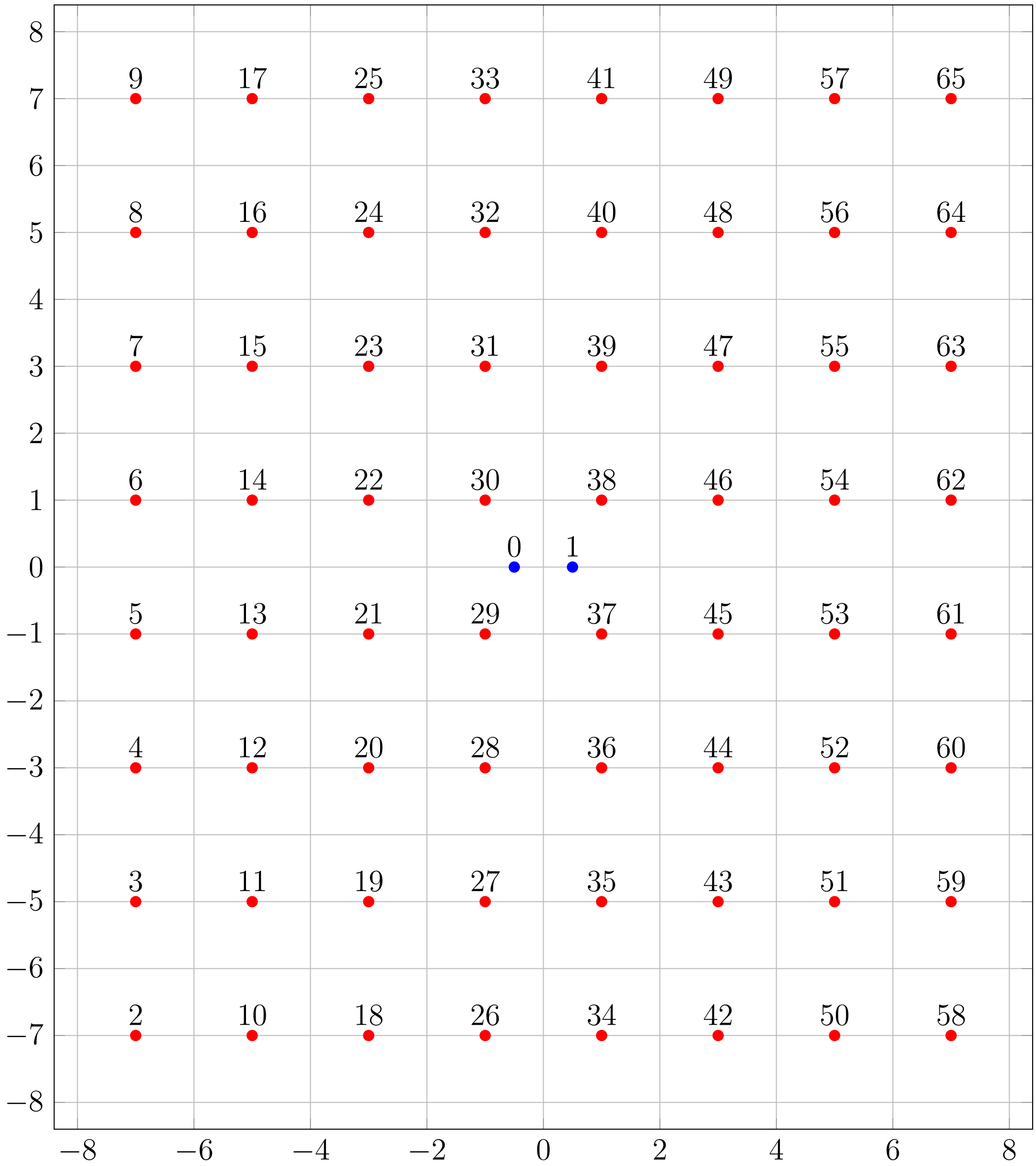


Figure 4.12: Presenting the POIs' location of OP's $Dataset_4$

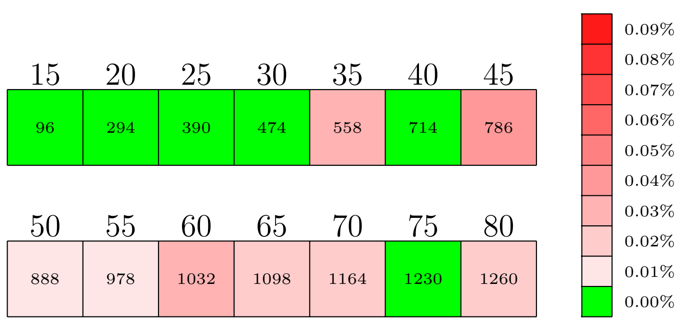


Figure 4.13: The results of $Dataset_5$ for the OP

Figure 4.13 shows that about 42% of scenarios perform to the same level as the state of the art for $Dataset_5$. In addition, other scenarios vary in performance behind the state of the art from 0.01% to 0.05%. However, the performance of the $Dataset_5$ is poorer than other datasets performance because of the location of the start/end points and the locations of POIs. Figure 4.14 shows the location of the POIs for $Dataset_5$ where the start point is located at the top and the end point is located at the bottom.

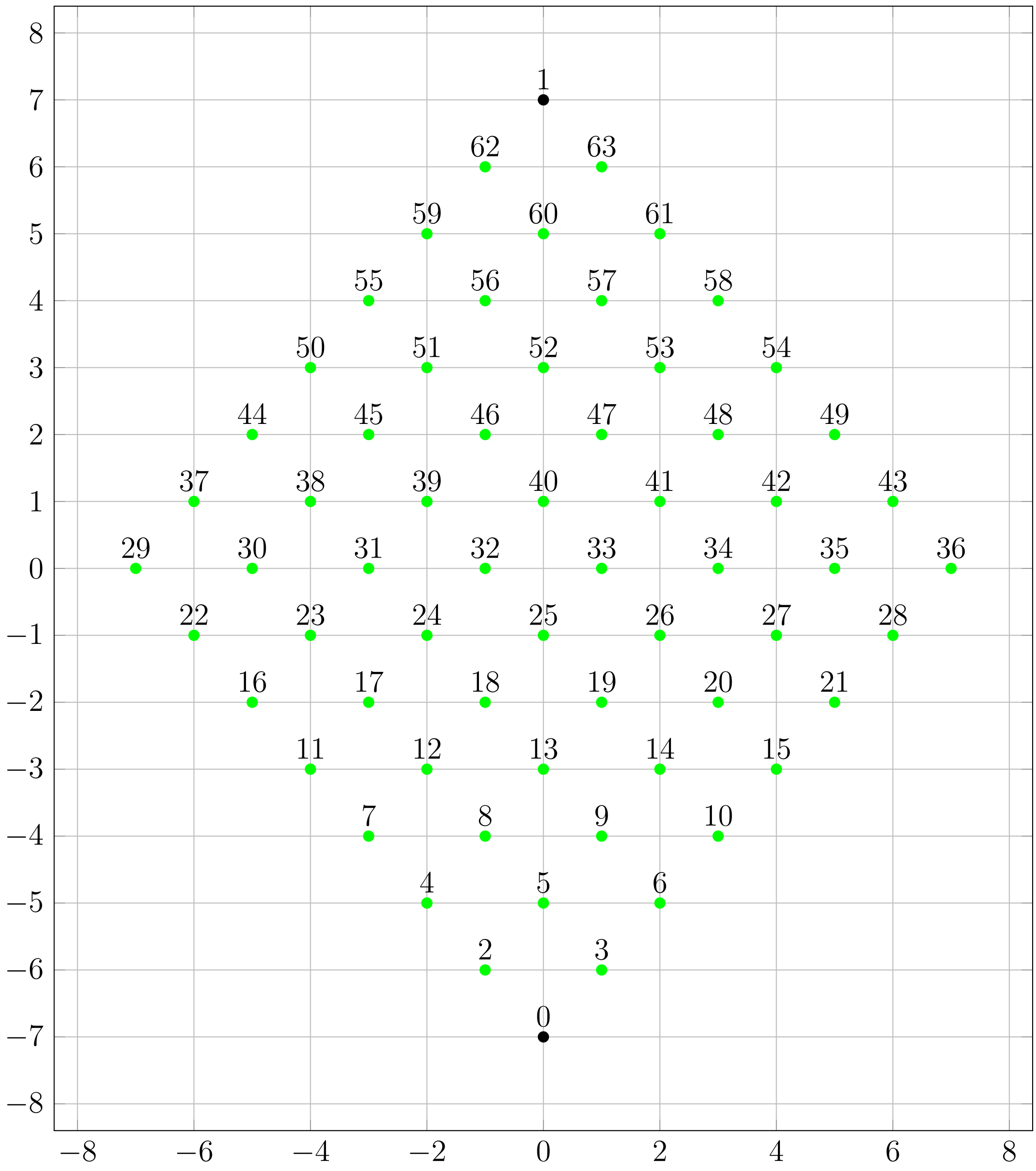


Figure 4.14: Presenting the POIs' location of OP's $Dataset_5$

The results of the OP showed that the performance of the ACO based on the

ICDM is very competitive; pleasingly, we even achieved some better results than the state of the art (see Figure 4.10). The main challenge in $Dataset_1$, $Dataset_2$, and $Dataset_3$ is that some nodes with high scores are in the opposite direction of the optimal route. The model is limited to looking forward (looking for all nodes together) for the best direction. For example, the model might choose to go south from the start point where is the optimal route is north of the start point. In addition, the challenges in $Dataset_4$ and $Dataset_5$ is that the distance between every two nodes (where are next to each other) is fixed. In other words, the distance between any node with its neighbours is the same distance. Moreover, the $Dataset_5$ has an extra challenge comparing with the $Dataset_4$ where are the start/end points located in a different location (start point is located in the north of the city, and the end point is located in the south of the city) see Figure 4.14.

TOP benchmark instances

Also, we have applied the TOP to our model to present the novelty of our ICDM in solving a number of models using a single algorithm. Figures 4.15 to 4.18 show the results for the $Dataset_1$ to $Dataset_3$ (see the results of $Dataset_4$ to $Dataset_7$ in Appendix B).

Each figure below represents a dataset where each dataset has different scenarios labeled next to the box while the values inside the boxes represent the total score, which is collected by visiting different POIs, where the boxes are coloured depending on the gap between the result and the state of the art (each figure provides a scale on the right-hand side that represents the different colours based on the gap between our results and the existing works).

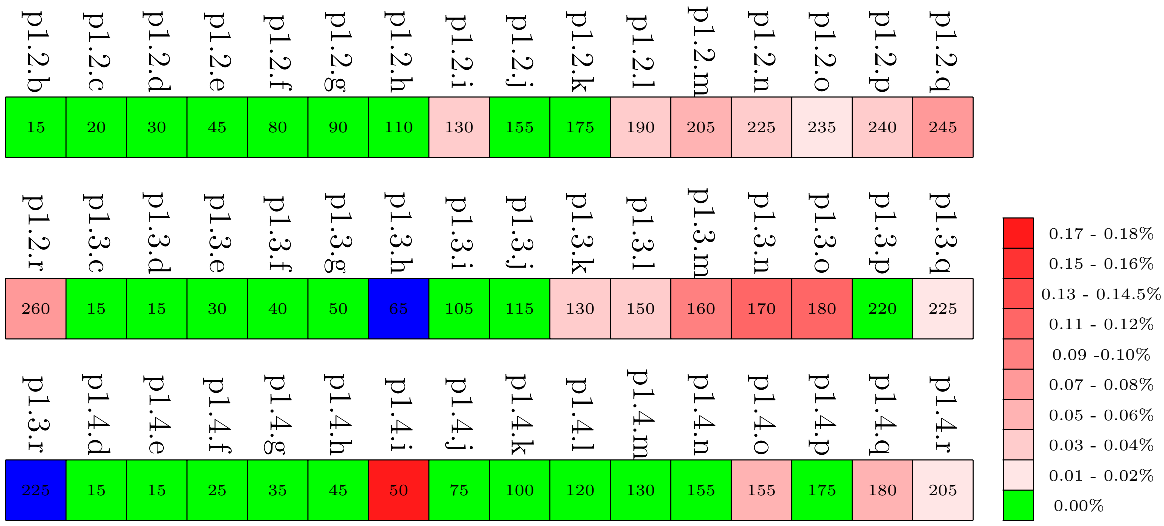


Figure 4.15: The results of $Dataset_1$ for the TOP

Figure 4.15 shows 58% of our scenarios perform at the same level as the existing work while 4% of our scenarios achieve better results (in blue) (building a tour which collects more scores than the existing works). Also, some of the scenarios perform less well than the existing works (the red section of the scale).

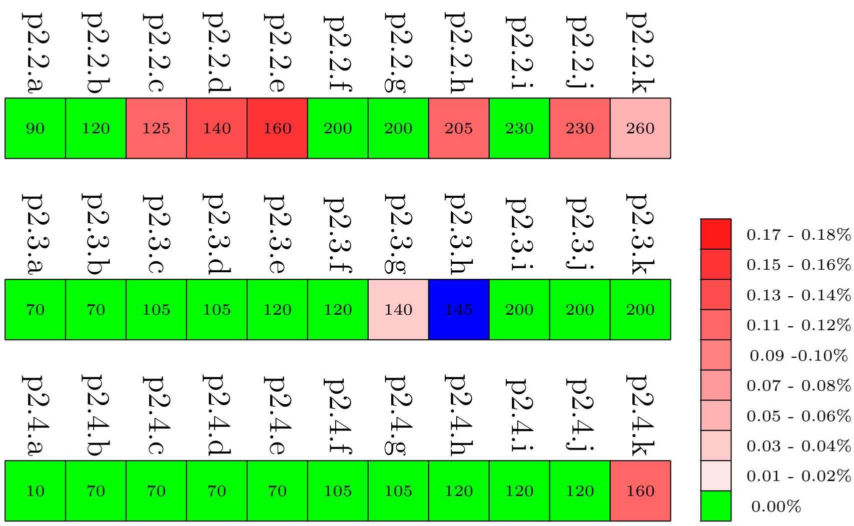


Figure 4.16: The results of $Dataset_2$ for the TOP

Figure 4.16 shows 72% of our scenarios perform at the same level as the existing work while 3% of our scenarios achieve better results (in blue) (building a tour which collects more scores than the existing works). However, some of our scenarios perform less well than the existing works (the red section of the scale).

Figure 4.17 shows how the POI locations are distributed differently in $Dataset_1$ and $Dataset_2$. In addition, the location of start/end points mainly affects the results as the start/end points in $Dataset_1$ are located in the middle of the map while the start/end points in $Dataset_2$ are located at the bottom of the map. Also, the number of POIs in each dataset affected the performance of the algorithm ($Dataset_1$ has 32 POIs and $Dataset_2$ has 21 POIs). The performance of $Dataset_2$ is slightly better than $Dataset_1$ because $Dataset_2$ features fewer options (fewer POIs to visit).

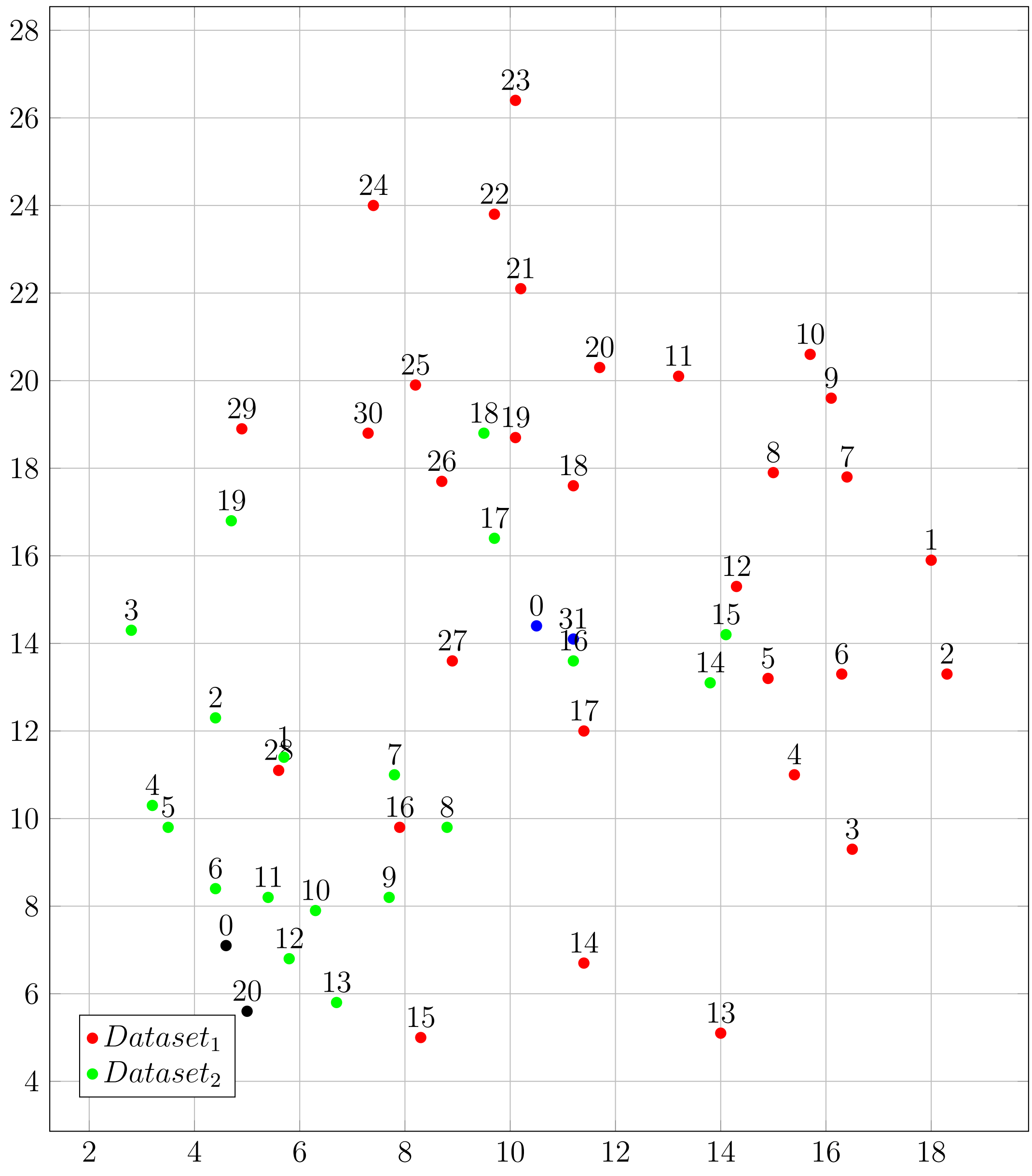


Figure 4.17: Comparing the TOP's $Datasets_1$ and $Dataset_2$

Figure 4.18: The results of $Dataset_3$ for the TOP

Figure 4.18 shows that 46% of our scenarios perform comparably with the existing work. However, some of our scenarios perform more poorly than the existing work (the red parts of the scale). $Dataset_3$ performs most poorly because this dataset features a higher number of options than the other datasets.

The results show that the ACO, which is based on the ICDMs performance, provides good results, and it even performs better than the state of the art in $Dataset_1$ and $Dataset_2$. The main challenge in TOP datasets is that choosing a set of POIs for each day with maximizing the total collected scores. The ICDM with ACO performs acceptably where the model is limited to have the future of looking for selecting a set of POIs together for each day.

Analysis of the OP and TOP results

The results (shown in the OP and TOP section) were comparable with those of the state-of-the-art results in most instances except for a small number of instances in the TOP. In addition, we show that the results in blue (that refer to our results) outperform the existing state-of-the-art models results.

OPTW and TOPTW benchmark instances

Furthermore, we have tested the OPTW and TOPTW on our model and the results are shown in Figures 4.19 to 4.23.

Each figure below represents a dataset where each dataset features different scenarios that are labelled above the boxes, and each scenario has four different trip lengths labelled on the left side of the boxes (d= number of days). In addition, the values inside the boxes represent the total scores, which are generated by users visiting different POIs. Here, the boxes are coloured according to the gap between the result and the state of the art (each figure shows in the right side of the figure a scale which represents the different colours based on the gap between our results and the existing state-of-the-art works).

	c101	c102	c103	c104	c105	c106	c107	c108	c109	r101
$d = 1$	0	190	240	240	200	50	250	250	260	25
$d = 2$	0	210	390	440	330	80	440	460	480	30
$d = 3$	0	210	420	560	380	80	590	600	640	30
$d = 4$	0	210	420	640	390	80	710	710	760	30

	r102	r103	r104	r105	r106	r107	r108	r109	r110	r111
$d = 1$	169	197	218	115	186	202	216	175	192	209
$d = 2$	253	309	375	227	341	352	388	310	372	370
$d = 3$	281	415	489	316	457	481	504	421	488	477
$d = 4$	286	449	561	316	541	589	612	511	559	567

	r112	rc101	rc102	rc103	rc104	rc105	rc106	rc107	rc108
$d = 1$	213	150	200	200	220	190	190	200	210
$d = 2$	405	300	370	400	420	350	380	410	420
$d = 3$	537	430	520	600	630	500	530	560	610
$d = 4$	632	520	630	730	810	620	680	710	780

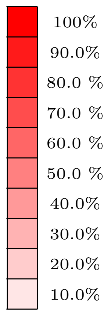


Figure 4.19: The results of $Dataset_1$ for the OPTW and TOPTW

Figure 4.19 shows that the singularity of the $Dataset_1$ affects the performance of our model. Our model achieves good results in some scenarios while in other scenarios it has difficulties in identifying an optimal tour.

	c101	c102	c103	c104	c105	c106	c107	c108	c109	r101
$d = 1$	0	220	300	330	280	260	260	280	280	46
$d = 2$	0	360	530	600	480	480	530	540	520	76
$d = 3$	0	400	700	810	660	670	770	760	760	101
$d = 4$	0	400	810	980	810	820	970	980	970	118

	r102	r103	r104	r105	r106	r107	r108	r109	r110	r111
$d = 1$	264	276	283	221	275	277	286	262	251	278
$d = 2$	430	458	512	369	488	486	519	476	478	495
$d = 3$	546	624	713	507	656	688	712	641	672	687
$d = 4$	624	750	872	629	809	849	889	776	839	839

	r112	rc101	rc102	rc103	rc104	rc105	rc106	rc107	rc108
$d = 1$	265	187	232	249	245	211	236	267	274
$d = 2$	496	363	435	476	502	413	442	499	512
$d = 3$	707	507	634	690	746	576	643	716	736
$d = 4$	877	660	797	883	941	730	808	918	948

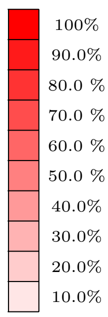


Figure 4.20: The results of $Dataset_2$ for the OPTW and TOPTW

Figure 4.20 shows that our model performs well in $Dataset_2$ where the gap between our model and the state of the art is better than $Dataset_1$.

	c201	c202	c203	c204	c205	c206	c207	c208	r201	r202
$d = 1$	660	690	810	820	730	740	760	770	717	830
$d = 2$	1210	1280	1300	1350	1290	1350	1290	1340	1155	1192
$d = 3$	1720	1420	1380	1630	1770	1740	1730	1730	1370	1395
$d = 4$	1720	1420	1390	1640	1780	1740	1750	1760	1442	1402

	r203	r204	r205	r206	r207	r208	r209	r210	r211	rc201
$d = 1$	924	1013	841	925	987	1035	834	887	982	755
$d = 2$	1257	1384	1271	1316	1368	1411	1275	1279	1390	1288
$d = 3$	1406	1396	1436	1453	1456	1450	1454	1426	1458	1574
$d = 4$	1406	1396	1436	1458	1456	1450	1454	1426	1458	1580

	rc202	rc203	rc204	rc205	rc206	rc207	rc208
$d = 1$	875	933	1032	807	831	891	954
$d = 2$	1421	1484	1602	1324	1411	1473	1578
$d = 3$	1649	1681	1721	1588	1683	1692	1724
$d = 4$	1655	1681	1721	1588	1695	1692	1724

0.28 - 0.30%
0.25 - 0.27%
0.22 - 0.24%
0.19 - 0.21%
0.16 - 0.18 %
0.13 - 0.15 %
0.010 - 0.12 %
0.07 - 0.09 %
0.04 - 0.06%
0.01 - 0.03%
00.0%
Better

Figure 4.21: The results of $Dataset_3$ for the OPTW and TOPTW

Figure 4.21 shows our models performance in five scenarios compared to the existing works (in green). In scenario $c205$ (a three-day trip), the results in blue show where we achieved better results than the existing works. In general, $Dataset_3$ has less singularity, which allows our algorithm to produce good results.

	pr01	pr02	pr03	pr04	pr05	pr06	pr07	pr08	pr09	pr10
$p = 1$	0	352	343	387	443	447	0	387	384	472
$p = 2$	0	352	343	718	846	853	0	687	739	843
$p = 3$	0	352	343	1050	1157	1189	0	687	739	1125
$p = 4$	0	352	343	1050	1157	1189	0	687	739	1463

Figure 4.22: The results of $Dataset_4$ for the OPTW and TOPTW

Figure 4.22 shows that our algorithm has not been able to consider waiting time, which led to two scenarios that could not be solved ($Pr01$ and $Pr07$) because no POI that is open is available at the beginning of the trip time (other models have modelled waiting procedures). In other scenarios in the same dataset, our model is able to produce a tour trip.

	pr11	pr12	pr13	pr14	pr15	pr16	pr17	pr18	pr19	pr20
$p = 1$	315	117	416	480	581	478	339	449	504	564
$p = 2$	494	240	776	849	1067	939	568	821	942	1013
$p = 3$	494	321	1058	1177	1067	1341	706	1115	1323	1414
$p = 4$	494	321	1279	1466	1067	1676	816	1115	1643	1802

Figure 4.23: The results of $Dataset_5$ for the OPTW and TOPTW

Figure 4.23 shows that our algorithm outperforms the existing works in six scenarios (in blue). Furthermore, our model produces good results in most of the other scenarios in the dataset.

Analysis of the OPTW and TOPTW results

It is important to note that the results of the OPTW and TOPTW differ in performance compared to the state-of-the-art models. The main two reasons for this are: (1) our model has been tested based on a general-purpose algorithm, and (2) the dataset is made in such a way where some algorithms produce misleading near-optimal results.

In $Dataset_1$, scenarios c101, c106, and r101 perform the least well. First, our model could not identify a solution for scenario c101 because all the POIs were closed (by moving from the starting point at the starting time to any POI). Second, scenarios c106 and r101 only feature a few POIs that are available at times when other POIs will need to be visited later (the model must calculate waiting times between POIs to able to visit other POIs). In addition, the performance of $Dataset_2$ compared to $Dataset_1$ is worse because the number of POIs in $Dataset_2$ is double the number of POIs in $Dataset_1$. Also, $Dataset_2$ still suffers from the same problem as $Dataset_1$ (in scenarios (c101 and r101)).

The performance of $Dataset_3$ is better than that of $Dataset_1$ and $Dataset_2$ because the opening/closing times for POIs are more homogenous and similar whereas in $Dataset_1$ and $Dataset_2$, opening times are less homogenous and dissimilar. In addition, we produce a better result than the existing works and five results which are similar to the state of the art.

The performance of $Dataset_5$ is better than $Dataset_4$ where scenarios Pr01 and Pr07 could not be solved because all the POIs are closed (by moving from a starting point at a starting time to any POI). In addition, $Dataset_5$ provides six results that are better than the existing works. The main reason for this performance of $Dataset_5$ is that opening/closing times are similar and there is no need to model waiting times between POIs.

In the final analysis, the challenge in $Dataset_1$ to $Dataset_5$ is that the optimal route is hidden where it is very difficult for general algorithms to find it. For example, to find the optimal route in $Dataset_1$, the algorithm should wait for some time until the first POI of the optimal route is open, so it seems that it is wasting time. However, it is only a way to find the optimal route. Our model is limited to find such these challenges in the dataset because in the actual life most POIs are open at similar times rather than opening POIs after each other in the timeline.

4.4.2 Applications of ICDM's features

Now we have demonstrated that our ICDM can solve many models, we present the ICDM's other characteristics. Certainly, detailed traveler data such as starting/ending times, POIs with particular attributes, etc., are lacking in availability. We have generated several scenarios (instances) based on real-world data (see Appendix A) that represent particular travelers' situations. Table 4.3 shows these instances along with these instance's conditions. The main aim of presenting these different instances is to cover the ICDM's features where other models are limited to solve it.

We have simulated five different scenarios to express the overall ability of our ICDM to customize each trip based on each instance. First, we have simulated that a family (consisting of two parents and three young children 5, 3, and 1.5 years old) plan to travel to *Durham*, the UK in the Easter holiday (*April*). In addition, the family consider one HC: that POIs must be suitable for children, and one SC: high priority for POIs that provide a baby-care room.

Second, we simulated some colleagues who are visiting *Durham*, the UK for business purposes in *March*; they would like to take a tour of the city in their free time. In addition, this group only stipulates one HC: weather conditions must not include rain.

Third, a young couple who wish to marry in the summer holiday and take their honeymoon in *Durham*, UK. This couple has two HCs: (1) weather conditions must not be heavy rain, and (2) they wish to only visit POIs with free entry because of their limited budget.

Fourth, a retired couple that plan to spend the Christmas holiday with their son who lives in *Durham*, UK. This couple wishes to tour the city when the weather conditions do not feature heavy rain (the HC), and their ability to pay max 10£ for each POI.

Finally, five students who plan to visit their friend studying as a postgraduate at *Durham University* in the half-term holiday (*October*). This group has one HC: all recommended POIs must have wheelchair access (e.g. parks and open-door POIs), and the four SCs shown in Table 4.3.

Therefore, these five scenarios cover the ICDM's main features such as customising constraints based on each traveler, the ability to customize constraints based on time (in the fifth scenario), and the ability to combine HCs and SCs. Broadly speaking, existing models would not be able to generate a valid tour based on the above scenarios (see Section 4.4.2) due to the fixed constraints and the fixed algorithms they employ. In other words, the existing studies have built their model based on fixed constraints which will be applied to all travellers where the personalization is not apply based on different constraints. For example, travelling in different seasons or weather condition should be considered where the user can explicitly apply his/her constraints.

In addition, we show different scenarios based on the assumption that all travellers have the same preferences. The main idea of setting up experiments based on travellers having the same preferences is to show how much impact the different constraints have.

Table 4.3: The scenarios description

No	Travellers category	Constraints
1	Family	• Must all recommended POIs suitable for children • Preferring to visit places which provide a baby-care room
2	Colleagues	• Must all be outdoor POIs in good weather conditions (not raining)
3	Young couple	• Must all be outdoor POIs in acceptable weather conditions (not heavy rain) • Must all be POIs with free entry.
4	Retired couple	• Must all be outdoor POIs in acceptable weather conditions (not heavy rain) • Must all be POIs that cost less than 11£
5	Students	• Preferring all outdoor POIs in acceptable weather conditions (not heavy rain) • Must all POIs have wheel access • Preferring all recommended POIs are free • Preferring to visit open door POIs afternoon • Preferring all POIs have free parking

Table 4.4: The results of the real-world dataset

Experiment No	Tour trip
1	10 76 47 70 14 7 44 54 85 55 77 10
2	10 81 70 44 77 47 14 7 85 2 23 76 74 72 10
3	10 2 47 77 72 85 74 11 70 7 55 81 54 10
4	10 47 74 81 5 85 70 7 11 2 55 77 72 10
5	10 79 70 14 7 76 54 55 44 40 2 72 10

Table 4.4 provides experimental data on the ICDMs features where the recommended tour is presented for each scenario. Table 4.4 shows the results of different scenarios where the recommended tour is presented by the number of POIs. Also, the start/end point is represented by POI #10, which shows the first POI and the last POIs in all recommended tours. The analysis examining the impact of the

constraints on a tour in the case that travelers have the same preferences has a significant impact because almost every scenario requires a different tour.

Table 4.5: Analysing the result of the first experiment

Constraint Name	Constraint Type	76	47	70	14	7	44	54	85	55	77
Children friendly	HC	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Baby-Care room	SC	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 4.5 shows that all recommended POIs have satisfied all constraints.

Table 4.6: Analysing the result of the second experiment

Constraint Name	Constraint Type	81	70	44	77	47	14	7	85	2	23	76	74	72
No raining	HC	•	•	•	•	•	•	•	•	✓	✓	•	•	•

Table 4.6 shows that all recommended POIs are indoor (labelled as •) and two POIs (#2 and #23, which are outdoor) have satisfied the weather constraints

Table 4.7: Analysing the result of the third experiment

Constraint Name	Constraint Type	2	47	77	72	85	74	11	70	7	55	81	54
No heavy raining	HC	✓	•	•	•	•	•	•	•	•	•	•	•
Fee free	HC	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 4.7 shows that all POIs that are recommended, are indoor (labelled as •), and one POI (which is #2, outdoor) have satisfied the weather constraints at these two POIs visiting times. In addition, all POIs have satisfied the fee constraint.

Table 4.8: Analysing the result of the fourth experiment

Constraint Name	Constraint Type	47	74	81	5	85	70	7	11	2	55	77	72
No heavy raining	HC	•	•	•	•	•	•	•	•	✓	•	•	•
Fee less than 11£	HC	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 4.8 shows that all POIs that are recommended are indoor (labelled as •), and one POI (which is #2, outdoor) have satisfied the weather constraints at these two POIs visiting times. In addition, all POIs have satisfied the fee constraint.

Table 4.9: Analysing the result of the fifth experiment

Constraint Name	Constraint Type	79	70	14	7	76	54	55	44	40	2	72
No heavy raining	SC	•	•	•	•	•	•	•	•	✓	✓	•
Wheel access	HC	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Fee free	SC		✓		✓	✓	✓	✓			✓	✓
Open door afternoon	SC	•	•	•	•	•	•	•	•	✓	✓	•
Free parking	SC	✓				✓			✓		✓	

Table 4.9 shows that all the SCs have been satisfied by the different POIs. In addition, all POIs have satisfied the HC (wheelchair access).

The results show that our model can efficiently handle all the various constraints presented the scenarios above. Also, another benefit of our model is that it can successfully build plan recommendations for different seasons and different months of the year

4.5 Conclusion

The present study is designed to determine the effect of the limitations of the OP and its extensions in customizing tour trips based on different constraints. In general, the FTRM has been designed to represent a generalized model of the OP, and the

FTRM has been built on the ICDM. The FTRM can solve many of the OP extensions such as the OP, TOP, OPTW, and TOPTW. One of the more significant findings is that while user preferences are essential for building a tour trip, considering different constraints is also important to enable the personalization of a tour trip. The results of this chapter indicate that only various constraints with the same preferences have made the recommended tour trips are vast.

The chapter makes a significant contribution to the field of Recommender Systems; specifically, because the FTRM can deal with many models related to the OP extensions and provide comparable results to the state-of-the-art models.

Chapter 5

Happiness Model

In this chapter, we introduce the *Happiness Model* (HM), which is a novel approach that can build *personalised*[1] tour trips[1] based on a measure of travelers' satisfaction, defined as a function of time and maximised over the duration of the trip. The HM addresses the limitations of existing approaches by considering different types of activities travellers may wish to do during a trip (such as visiting POIs, travelling between POIs, and reducing wasted time) to recommend the most appropriate tour routing for them. The HMs results based on tests run on publicly available datasets are reasonable in terms of the number and the values of visited POIs, and, moreover, reflect individual preferences on how time should be spent. Most characteristically, shorter tours are recommended to travellers who wish to spend less time travelling between POIs.

The main difference between the HM and existing models is that the HM is time-centric rather than POI-centric and builds recommended tours by maximising user satisfaction over the set of moments throughout a given trip. Here, we implemented the HM in conjunction with the ICDM model, introduced in Chapter 3 as a data model that reduces data dimensionality and the size of the search space.

[1]See Glossary for definitions of italicized words

5.1 Introduction

In recent years, governments, industry, and tourism studies in academia have all increasingly turned towards happiness research in search of solutions to the problems they face [33]. Travellers take holidays to increase their happiness[1] level, and indeed, a recent study emphasis that people benefit from vacation in terms of happiness [80]. From the perspective of Recommender Systems (RSs), we introduce the *Happiness Model* (HM), which represents a time function designed to handle user's preference to build the most suitable tour trip to match their preferences. The HM handles different trip decision factors based on user's preferences.

The HM is designed to fill the gaps other models fail to consider. Mainly, the current works optimize tour trips based on POIs and their constraints[1]. However, choosing a connection (Path) between POIs and considering the time available (between Start[1]/End[1] times) for the trip are also essential factors that must be taken into account when building tour trips. We define $Happiness(t)$ as a function of time that represents the state of pleasure for the user at a specific moment (t). Uniquely, we have classified all aspects of a trip into (1) Activity, (2) Connection, and (3) Waiting. The first represents the users visiting a POI; the second represents the connection (i.e. the action of moving from a POI to another POI); the third represents the entire durations of time that travellers wait for the next activity or the difference in the duration between the actual plan (Recommended Plan) and the preferred trip length (T_{max}).

The importance of HM is concentrated in maximizing travelers' *happiness* by choosing their preferred POIs, selecting satisfying connections, and managing waiting times. To illustrate the problem that the HM solves, we provide an example of two groups of travelers (Group A and B) who are visiting the city shown on the map in Figure 5.1. Group A prefers natural, open places such as zoos, parks, mountains, etc. In contrast, group B prefers modern places such as shopping malls, street markets, and museums. Indeed, both groups have various preferences[1], and thus, they will each require different trip plans to suit these preferences.

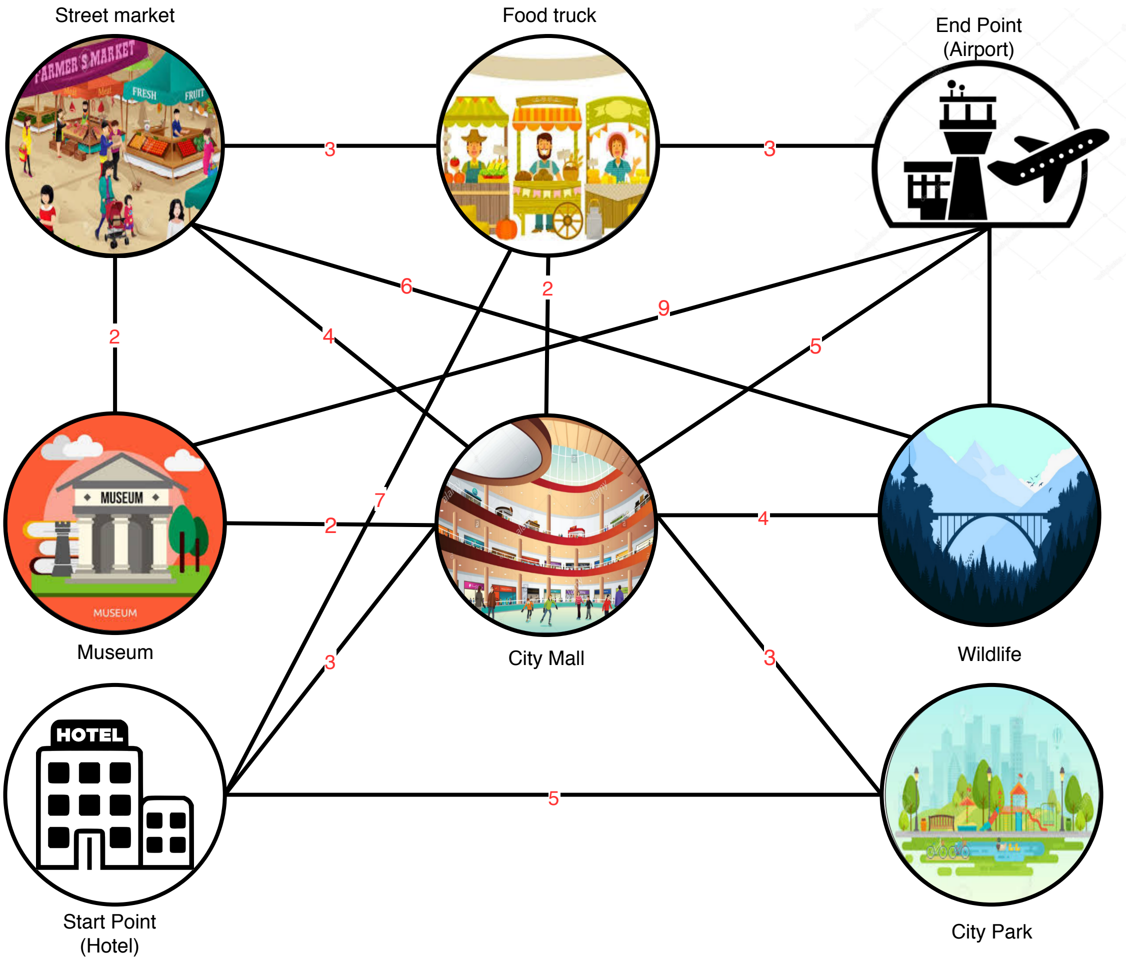


Figure 5.1: The city map with several POIs and specific start/end points

In addition, not only does each group prefer different types of POIs, but also each group has different travel preferences in moving from one POI to another. For example, group B prefers to walk (Connection Type) between POIs through the city so they can see more of the traditional places and street markets. However, group A prefers to travel by car so they can visit more natural POIs such as wildlife parks and rivers. Figure 5.1 shows a map of the city the two groups will visit along with several POIs and the costs of the connections between the various POIs.

5.1.1 Overview of HM

This section presents an overview of the HM and shows how it has been designed to tackle some of the limitations of existing works. First, as we classify the various actions involved in trips (e.g. activity, connection, and waiting), time is the main component of the HM because the HM can maximise the total collected scores from the three trip actions. To explain, in each moment, the HM considers if the users are currently engaged in an activity, in transfer between them, or waiting. Next, the HM calculates the total scores from different actions.

Figure 5.2 shows an example of how the HM handles the different actions involved in a trip. In the example, the POI #13 and #25 has multiple scores, the selected path from *Start Point*[1] to POI #13, from POI #13 to POI #25, and from POI #25 to *End Point*[1] have multiple scores, and the waiting time is represented as a yellow diamond where the users have more time for activities, although the RSs recommend a plan with less than the total time available (T_{max}). In addition, the value of each action is multiplied by the time it takes to complete (e.g. transferring from POI #13 to POI #25 takes 19 minutes (19 minutes $\times$ $Score_{13,25}$)).

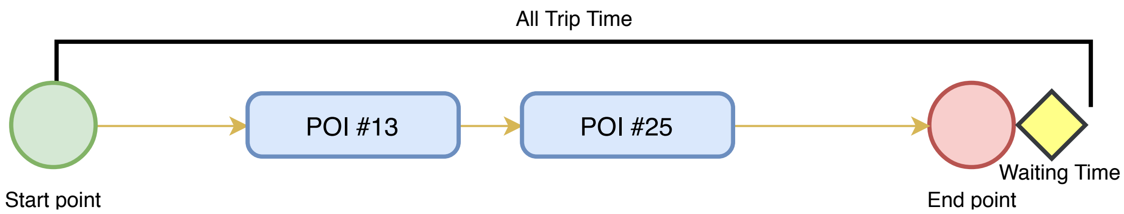


Figure 5.2: An example of HM calculation

In summary, the *Happiness Model* represents a new approach to modelling traveler's satisfaction level. In addition, our novel model solves the limitations of the existing models, as the existing models do not consider connection's values and waiting time. In particular, the HM optimizes tour trips based on the three categories of actions mentioned above. However, while the existing models optimize tour trips based on collecting scores from different POIs, the OP is NP-hard problem, the HM is designed to optimise different types of trip activities, which leads to the complicated nature of the HM.

5.2 Review of Previous Work

As travelers have different preferences and requirements for their trips, personalizing and recommending a tour trip is essential because fulfilling traveler's preferences increases their happiness level. Within the context of optimizing the traveler's options, designing and optimizing the most appropriate objective functions is highly applicable to solving the problems travellers face.

In the literature, many methods and approaches have been proposed for Travel Recommender Systems (TRSs) (for example, the *Tour Trip Design Problem* (TTDP)). However, previous studies have proposed solving the TTDP based on selecting POIs although they fail to address other important factors. For example, the previous studies do not consider traveler's preference in terms of traveling from one POI to another, and consuming all available trip time. In other words, all existing models build routes based only on the collected scores from visited nodes. In the final analysis, because all existing works are focused on only POI values, the existing models have significant limitations to modelling satisfaction, whereas the HM has been designed to evaluate each moment of the trip by considering different trip actions. In addition, the HM is designed based on time, using an algorithm to optimize each moment's activity rather than an aggregated collection of items.

As most RSs are based on data mining techniques, they are not able to provide a better-personalized tour than recommending tours based optimization techniques alone. However, while all the previous models have dealt with the TTDP, their solutions are subject to several important limitations in regards to fulfilling traveler's requirements[1] such handling waiting time, budget constraints, and/or easy routes. A special case of the TTDP is the OP, which is classified as an NP-hard problem. In addition, changing particular constraints within a model will change the performance of its algorithms and lead to the model being unable to solve the OP. Thus, we have introduced the *Happiness Model* to overcome the limitations of the existing models.

5.2.1 Happiness Model

To the best of our knowledge, all previous models focus only visited POI's total scores, and this represents their most serious drawback. Table 5.1 shows the main differences between the HM and other models. First, the HM and the MOOP provide multi-values (MV) for each node. However, the HM is designed to have multi-values for the different components of any trip, such as POIs, travel options, and time waste. In addition, the MCTOPMTW considers multi-attributes to have multi-constraints (MC) where the trip does not exceed each number of these attributes. Secondly, the MOOP deals with discrete categories as each category offers different benefits and the aim is to maximize the total benefits of the different categories. In addition, the MCTOPMTW supports tags where each POI has multiple tags (attributes).

On the contrary, first, the HM can deal with both categories and attributes as it takes advantage of both types of models. Second, our novel approach considers multi-values (MV) between each node while other models consider only distance. In other words, transferring between POIs involves various values such as time, length of journey, and price, etc. Third, the HM aggregates these values into a single value that represents traveler's satisfaction level. Fourth, travelers, with reservations (e.g., flights, hotels, or trains) will prefer to consume all the time allocated for a particular trip. Thus, the HM has been designed to consider the time waste as an essential factor likely to affect traveler's level of satisfaction.

Table 5.1: Comparison between Happiness model and previous works

Features	HM	MOOP	MCTOPMTW
Multi-score	•	•	MC
Categorising POIs	⋆	•	•
Personalising Connections	MV	∗	∗
Considering Wasting Time	•		

5.2.2 Multi-objective Orienteering Problem (MOOP) and Happiness Model

When making a travel-related decision, travelers take into account multiple objectives and constraints. [24] shows that higher-quality personalization results lead to travellers being more convinced to take a particular trip. In other words, when travelers feel the recommended tour trip closely matches their preferences, their trust level in the recommender system is increased. As the time factor becomes ever more important in traveler's busy lives (i.e.travelers often book hotels, flights, tours, and party tickets which should be coordinated at the time of booking), there is a need for a model that can handle all of these factors to build a robust trip plan recommendation.

While the main limitation of the MOOP is in its lack of support for personalisation (i.e. it does not consider waiting time), the HM is designed to consider time as a critical factor when building a tour trip. Besides, the HM considers other necessary decision-making parameters (i.e. multi-value for POIs and connections) for a tour trip whereas the MOOP considers only multi-value POIs. In essence, the HM is developed to cover the majority of tourists' decision-making parameters classifying the parameters into (1) Activities, (2) Connections, and (3) Waiting Time. The HM encompasses the existing models by considering all three of these key constraints.

5.3 Happiness Model (HM)

We have designed the *Happiness Model* (HM) to handle traveler's preferences in order to maximise user's satisfaction[1] level. First, we present the mathematical model, and next, we show a combination of the HM and the ICDM (see Chapter 3). Finally, the discussion section summarises the importance of the HM to its current applications within the tourism industry.

5.3.1 The Mathematical Model

The HM can be defined as follows. Let $G = (V, E)$ be a directed weighted graph where $i \in I$ and $i = 1, ..., |I|$ are a set of nodes representing a *Point of Interest* (POI) in a city, and each node has multiple values for different attributes. Also, $k \in K$; $k = 1, ..., |K|$ is a set of attributes for each node. Let E be a set of edges between these nodes (POIs), and each edge has multiple values (e.g. price, travel time, comfort level). Let $r \in R$; $r = 1, ..., |R|$ be a set of multiple values for each edge. A travel time between two nodes $i, j \in I$ denotes TT_{ij}, and ST_i denotes the time spent at the i node. Given a starting node s and terminal node t, and let the $s = 1$ and $t = |I|$. The trip length may be a day or longer, so $p \in P$; $p = \{1, ..., |P|\}$ is a set of trip days. In addition, each trip has a start time and end time where $t \in p$; let $t = 1, ..., |p|$ be a set of moments in the trip p day. The time limitation for each day of the trip is represented by T_{max}, and S_{max} is the maximum value for all actions (we have a user one scale for all activities from 0 to 1).

Also, let X_{pti} be an activity decision variable, which is equal 1 when the user stays at the i node on p day at t time, otherwise it is equal to 0. Let Y_{ptij} be a connection decision variable equal to 1 when the user moves from the i to j nodes on p day at t time, otherwise it is equal to 0. Let Z_{pt} be a waiting decision variable that is equal 1 when the user is waiting on p day at t time, otherwise it is equal to 0. In addition, let S_{pti} be the score value of activities at i node on p day at t time. Also, let C_{ptijr} be the value of a connection from i to j nodes based on the r attribute on p day at t time. In addition, let W_{pt} be the value of the waiting time on p day at t time.

The equations below describe the HM in relation to the constraints. Equation (5.3.1) is the *objective function*, which maximizes the total points from the three different actions: (1) activity, (2) connection, and (3) waiting; the equation has three main functions which are $f_1(a)$, $f_2(c)$, and $f_3(w)$, and each of these functions represent one of the main actions (activity, connection, and waiting). The HM has a value of between 0 and 1 where 1 represents the highest level of user satisfaction in the tour trip.

$$Max\left(\frac{f_1(a)+f_2(c)+f_3(w)}{T_{max}\times S_{max}\times |P|}\right) \tag{5.3.1}$$

The three functions $f_1(a)$, $f_2(c)$, and $f_3(w)$ are shown in Equations (5.3.2), (5.3.3), and (5.3.4). The $f_1(a)$ illustrates the happiness level for an activity. It adopts a value of between 0 and 1 by dividing the value by $(T_{max} \times S_{max} \times |P|)$.

$$f_1(a) = \sum_{p=1}^{|P|}\sum_{t=1}^{|p|}\sum_{i=1}^{|I|} X_{pti} \times S_{pti} \tag{5.3.2}$$

$f_2(c)$ represents the happiness level for the connection activity. It adopts values between 0 and 1 by dividing the value by $(T_{max} \times S_{max} \times |P|)$.

$$f_2(c) = \sum_{p=1}^{|P|}\sum_{t=1}^{|p|}\sum_{i=1}^{|I|}\sum_{j=1}^{|I|}\left(Y_{ptij} \times \sum_{r=1}^{|R|} C_{ptijr}\right) \tag{5.3.3}$$

The $f_3(w)$ represents the happiness level for waiting time, again dividing the value by $(T_{max} \times S_{max} \times |P|)$.

$$f_3(w) = \sum_{p=1}^{|P|}\sum_{t=1}^{|p|}(Z_{pt} \times W_{pt}) \tag{5.3.4}$$

Equation (5.3.5) is a constraint that allows only one action at the same time. In addition, Equation (5.3.6) ensures that the trip on each p day starts from s, which is the start point. Also, Equation (5.3.7) ensures that on each p day, the trip ends at the e POI that is the end point.

$$\sum_{i=1}^{|I|} X_{pti} + \left(\sum_{i=1}^{|I|}\sum_{j=1}^{|I|} Y_{ptij}\right) + Z_{pt} = 1; \forall t = 1,\ldots,|p|; \forall p = 1,\ldots,|P| \tag{5.3.5}$$

$$\sum_{j=1}^{|I|} Y_{p11j} = 1; \forall p = 1,\ldots,|P| \tag{5.3.6}$$

$$\sum_{i=1}^{|I|-1}\left(\frac{\sum_{t=1}^{|p|} Y_{pti|I|}}{TT_{i|I|}}\right) = 1; \forall p = 1,\ldots,|P| \tag{5.3.7}$$

Equation (5.3.8) is a constraint to ensure the tour trip is connected, and it ensures the connection time (travelling time) and visiting times are equal to data which represent by $TT_{i,j}$ and ST_i. Also, Equation (5.3.8) is equal to the Equations (2.7.4) and (2.7.5) in the OP model.

$$\frac{\sum_{t=n}^{t_1=n+TT_{s,r}} Y^t_{p,s,r} \times \sum_{t=t_1+1}^{t_2=t1+ST_r} X^t_{p,r}}{TT_{s,r}+ST_r} = \frac{\sum_{t=t_2+1}^{t_3=t_2+TT_{r,m}} Y^t_{p,r,m}}{TT_{r,m}} \leq 1 \tag{5.3.8}$$

$\forall n \in \{1,\ldots,|p-3|\}; \forall p = 1,\ldots,|P|; m \in I; \forall s,m = 1,\ldots,|I|; \forall r = 2,\ldots,|I-1|$

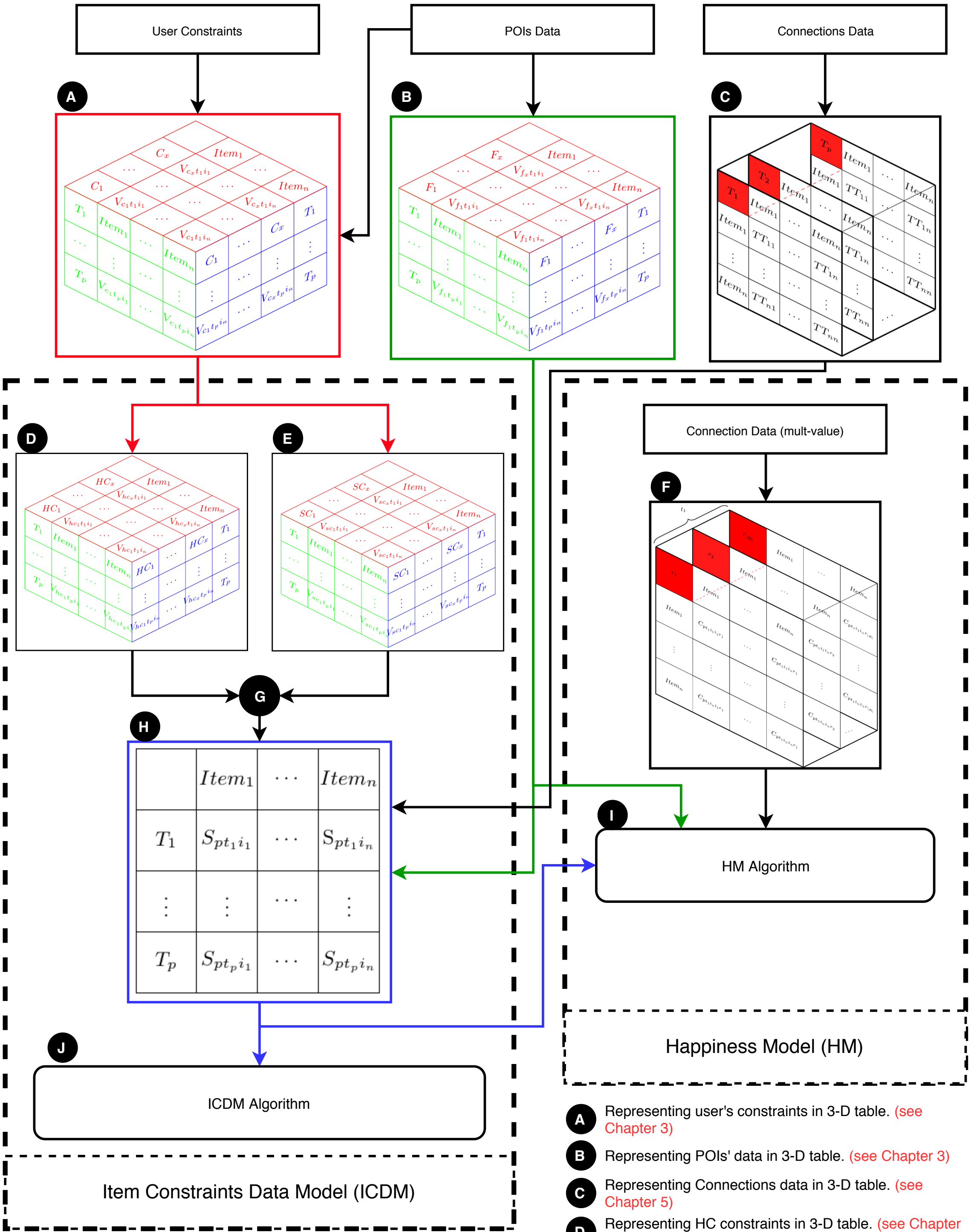
5.3.2 Overview of HM with ICDM

This section presents how we coupled the ICDM (see Chapter 3) with the HM. Briefly, the ICDM is designed to deal with multi-item constraints. However, the HM is designed to consider a multi-value for POIs and multi-values for connections. Figure 5.3 shows an overview of the HM, and how the HM and ICDM work together. Figure 5.3 can be classified into three parts: (1) data, (2) the ICDM, and (3) the HM.

Firstly, the ICDM deals with *User Constraints*, *POIs Data*, and *Connections Data*. The user constraints are defined as conditions on some or all POIs based on user constraints such as location, entrance fee, or weather conditions. These user constraints are illustrated as a 3D table (labeled A in Figure 5.3). The main three dimensions of the user constraints are (1) each constraint provides a value on an item at a specific moment; (2) at each moment, each constraint provides a value on some items; and (3) each item provides some constraints on every moment. The POI's data represents the information about the POIs such as opening/closing times, location or category. The POI data is shown in a 3D table (labeled B in Figure 5.3). Besides, connections data represents the travelers moving from one POI to another (labeled C in Figure 5.3). As the ICDM is designed to consider the congestion level, which is represented as layers of p tables, where each table represents the time consumed by moving from one POI to another POI at T_p time.

Second, the main part of the ICDM is the constraints (classified into HC and SC; see Chapter 3): (1) HC (labelled D in the Figure 5.3) and (2) SC (labelled E in Figure 5.3). Equation (3.3.4) (labeled G in Figure 5.3) is the aggregate of all constraints values in a single value (S_{pti}). The main output of the ICDM is a matrix that reduces the search space (labeled H in Figure 5.3).

Third, the HM is designed to handle user constraints by using the ICDM and considering multi-values for connections (labeled F in Figure 5.3). The graph below shows the complexity of the connection data when multi-values are used. The graph below (Figure 5.4) shows $|R|$ tables based on the number of multi-values in the connections, and all these tables represent only one moment, such as t_1 where it needs to be a p number of aggregated tables to represent the whole multi-value problem. Finally, we have developed an algorithm to solve the HM problem (labeled I in Figure 5.3).



A Representing user's constraints in 3-D table. (see Chapter 3)
B Representing POIs' data in 3-D table. (see Chapter 3)
C Representing Connections data in 3-D table. (see Chapter 5)
D Representing HC constraints in 3-D table. (see Chapter 3)
E Representing SC constraints in 3-D table. (see Chapter 3)
F Representing the Connections data with multi-value. (see Chapter 5)
G Representing Equation 3.3.4 which aggregates the HC and SC. (see Chapter 3)
H Representing the final data representation of the ICDM. (see Chapter 3)
I Representing the algorithm that will be used in the HM. (see Chapter 5)
J Representing the algorithm that will be used in the ICDM. (see Chapter 3)

Figure 5.3: An overview of building the HM with ICDM

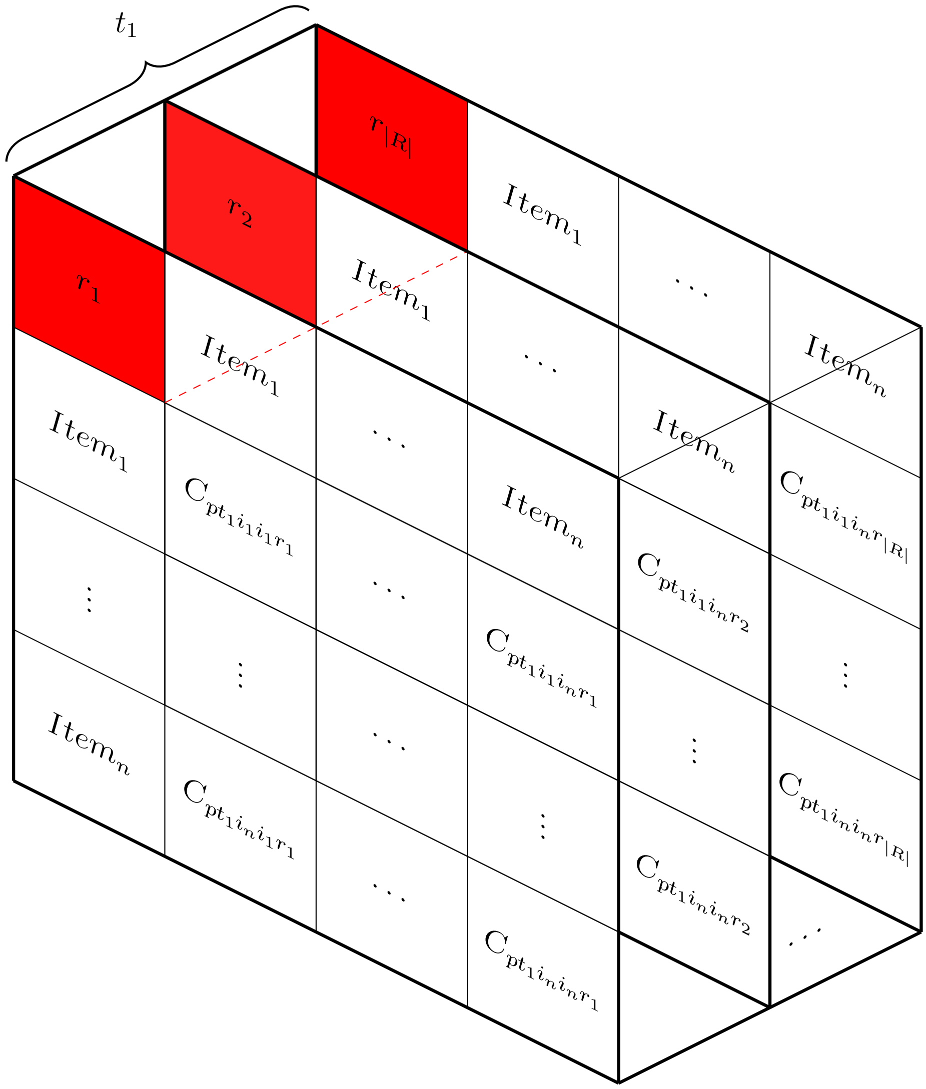


Figure 5.4: An overview of multi-value connections

5.4 Experiments

We have conducted a number of experiments to determine the capabilities of the HM. First, we used the HM on some of the existing datasets (see Table 5.2) to measure the impact of the use of the HM modelling on building a tour trip. Furthermore, we developed an Ant Colony Optimization (ACO) algorithm to produce comparable results with the existing models.

We selected these datasets based on the availability of public data. Specifically, we selected the OPTW & TOPTW datasets to conduct the HM experiment because these datasets provide several relevant features (e.g., visiting times, time windows, etc.).

Table 5.2: List of all datasets have been used in the experiments

Problem	Dataset Name	Reference	Number of Scenarios	Number of items $\|I\|$
OPTW & TOPTW	$Dataset_1$	[106]	10	48 to 288
	$Dataset_2$		10	48 to 288

5.4.1 An Ant Colony Optimization

As we have disused Ant Colony Optimization (ACO) in Chapter 4 Section 4.3. Briefly, we have adjusted to solve the HM, and Figure 5.5 shows the flowchart of the ACO. Based on the setting in Section 4.3.

We have conducted several experiments to determine the best value for the ACO parameters. ACO's initial parameters are listed based on Table 5.3, and Figures 5.8 and 5.7 show the different values based on different parameters values

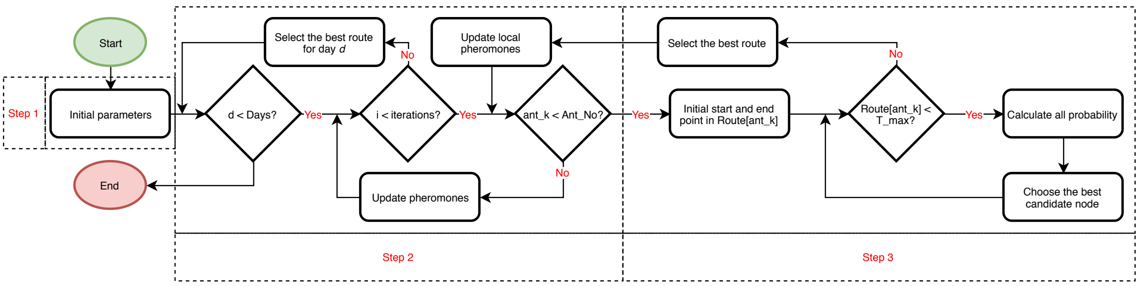


Figure 5.5: Overview of the ACO flowchart

We have conducted experiments to determine the iteration number for the ACO. Important to realize that running time is a critical factor especially when the problem is NP-hard. Figure 5.6 shows the average running time for different iteration numbers where is the minimum running time is 1541 milliseconds, and the maximum is 59875 milliseconds. We have chosen ten times for the iteration parameter.

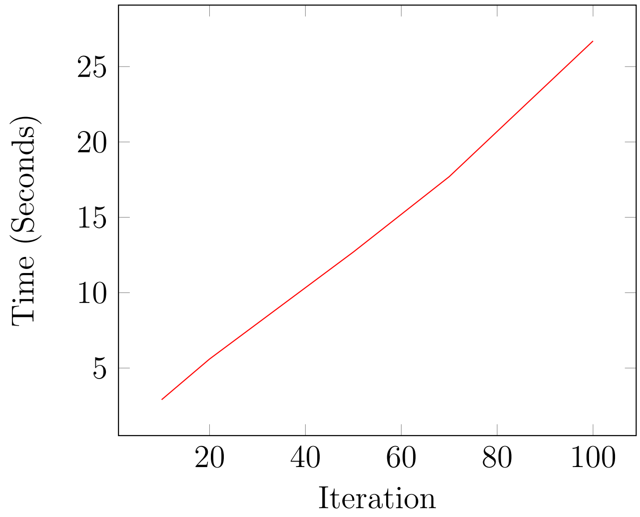


Figure 5.6: The average running time for different iteration numbers

Alpha and *Beta* represent the importance of score and rate of score to distance (see Equation (5.4.10)). In addition, choosing the optimal value for the *Alpha* and *Beta* is critical because the performance of the algorithm is based on these values. Consequently, we have used the tuning method, which calculates the parameters from zero to until it reaches the ultimate gain. We have conducted several different experiments (each scenario in each dataset has been conducted over 200 times for different values for the *Alpha* and *Beta*) to find the ACO's best performance across all the datasets. Figures 5.7 and 5.8 present the total scores (which have been normalised in which 1 represents the highest total scores for all the datasets) of all the datasets in all scenarios for different values of *Alpha* and *Beta*.

As shown in Figures 5.7 and 5.8, only some of the results have achieved the top scores whereas other results vary; there is not a point here which shows the *Alpha* and *Beta* make the algorithm perform at the same level (each new value for *Alpha* and *Beta* give different results). We have chosen the optimal value for *Alpha* and *Beta* which appears many times (e.g. $Alpha = 1$ and $Beta = 13$, $Alpha = 15$ and $Beta = 8$, etc.) to perform better in all datasets (all values are shown in dark red in 5.7 and 5.8 that have achieved the top results). In other words, any value of *Alpha* and *Beta* in dark red will provide the same results over all the datasets.

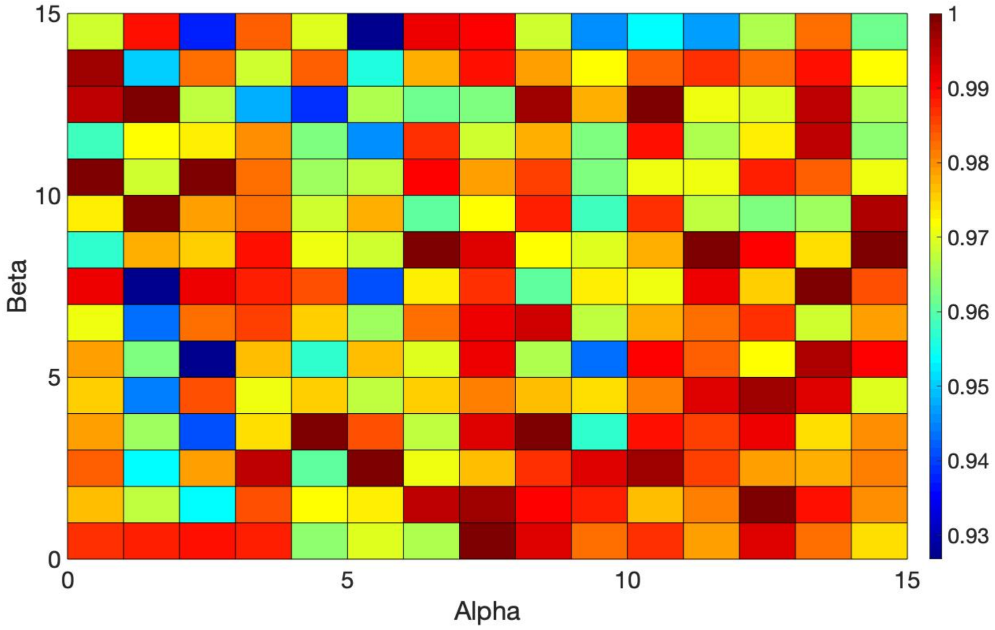


Figure 5.7: Overview of the ACOs performance based on different values of *Alpha* and *Beta*

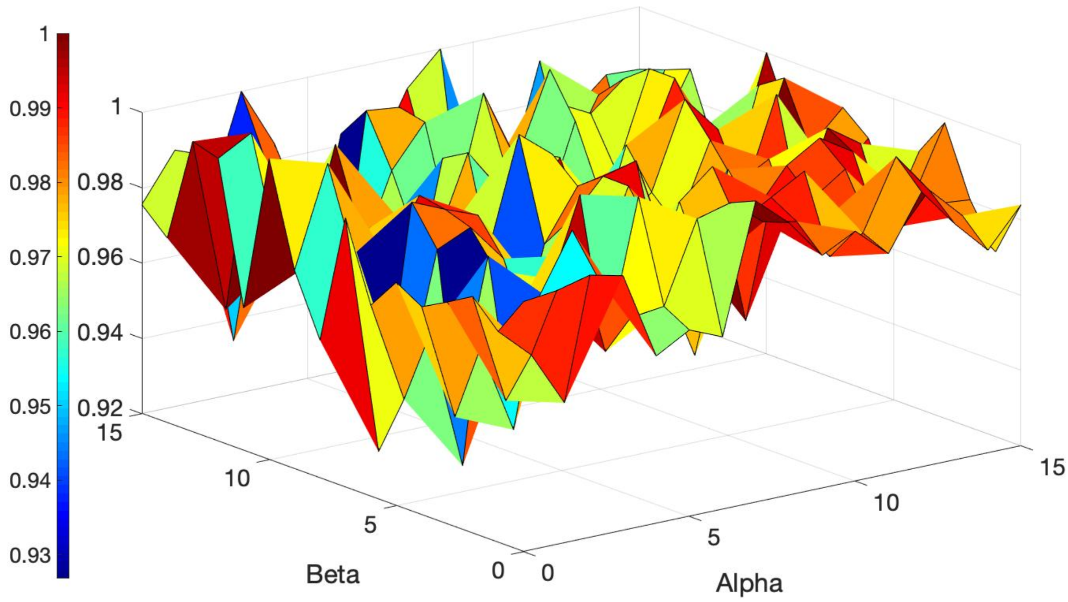


Figure 5.8: Overview of the ACOs performance based on different values of *Alpha* and *Beta*

Table 5.3: Initial parameters for ACO in the first step

Parameter	Initial Value	Description
α	15	*Alpha* represents the importance of Tau
β	8	*Beta* represents the importance of Eta
ρ	0.1	The value of pheromone evaporation
Ant_No	200	Number of ants
Iterations	10	Number of iteration
NodeSize		Number of nodes
η_{ij}	Equation (5.4.9)	*Eta* represents the rate of score to distance
$\tau_{i,j}$	Allocate 1000 value	representing the *Pheromones* level from i to j
$\delta_{i,j}$	Allocate 0 value	representing the maximum total path use i to j

In the ACO, two steps are used to update the pheromones to improve the effectiveness of the algorithm. The first update is called *update local pheromones* and this happens in step two (see Figure 5.5); after releasing an ant, it checks if the ant can find a better score for the path found (see Equation (5.4.11)). The second update is after all ants have been released (see Equation (5.4.13)).

Equation (5.4.9) calculates the rate of Activity value (S_i) to distance (TT_{ij}) and Connection value (C_{ij}) to distance (TT_{ij}). The value of *Eta* represents how the POI is preferred to the user. In addition, Equation (5.4.10) represents the probability calculation for the POI i to j. Equation (5.4.11) shows that function that find the max total score collected by Ant_x, and Equation (5.4.12) represents the local update (the first update). Also, Equation (5.4.13) represents the global update (the second update).

$$\eta_{ij} = \frac{S_i}{TT_{ij}} + \frac{C_{ij}}{TT_{ij}} \tag{5.4.9}$$

$$P_{i,j} = \frac{(\tau_{i,j})^{\alpha} (\eta_{ij})^{\beta}}{\Sigma \left((\tau_{i,j})^{\alpha} (\eta_{ij})^{\beta} \right)} \tag{5.4.10}$$

$$\delta_{i,j} = Max(\delta_{i,j}, Ant_x(i,j)) \tag{5.4.11}$$

$$\tau_{i,j} = (1-\rho) \times \tau_{i,j} + \delta_{i,j} \tag{5.4.12}$$

$$\tau_{i,j} = \rho \times \tau_{i,j} + (1 - \rho) \times \delta_{i,j} \tag{5.4.13}$$

5.5 Results

We provide the results achieved by the HM based on the experiments. In addition, the discussion section explains the challenges and the peculiarities of the existing datasets.

5.5.1 The HM benchmark instances

This section compares the results of the HM with the TOPTW results on $Dataset_1$ and $Dataset_2$ (more information is provided in Table 5.2). In general, the results clarify the effect of the HM on building a tour trip by customizing the tour based on users preferences. In other words, the outputs of the HM show that the HM can indeed achieve a greater level of personalization in tour trips because the HM considers and optimizes other factors of tour trips such as wasted time and connections. In addition, the values of the scores (activities), connections, and waiting times have been normalized between 0 to 1 where 1 represents the highest value.

Table 5.4 shows the eight different experiments we have conducted on the HM. First, we have used the values for the activities based on the public datasets values, and then generated different values for the connections and waiting times in these experiments. These experiments cover different scenarios to highlight the benefits the HM provides.

Experiments E_1 to E_4 have been conducted on $Dataset_1$, and Tables 5.5 to 5.8 show the results of each experiment. In each experiment, the scenario code and the total number of POIs in the recommended tour are shown. In addition, the main three actions of the trip are shown, and we present the total amount of time for each action and the total scores for each specific action.

Table 5.4: Comparing between different experiments

Experiment	Connection Score	Waiting Score	Dataset
E_1	1	0	$Dataset_1$
E_2	1	0.5	$Dataset_1$
E_3	$Random$	0	$Dataset_1$
E_4	$\frac{Random}{2}$	0	$Dataset_1$
E_5	1	0	$Dataset_2$
E_6	1	0.5	$Dataset_2$
E_7	$Random$	0	$Dataset_2$
E_8	$\frac{Random}{2}$	0	$Dataset_2$

Table 5.5: The result of the E_1 on the HM

Scenario	Activity		Connection		Waiting		Number
Code	Time (Minutes)	Total Scores	Time (Minutes)	Total Scores	Time (Minutes)	Total Scores	of POIs
Pr02	227	125	384	384	389	0	19
Pr03	197	93	368	368	435	0	17
Pr04	223	120	382	382	395	0	19
Pr05	389	245	351	351	260	0	24
Pr06	304	145	371	371	325	0	24
Pr08	252	120	417	417	331	0	18
Pr09	222	120	464	464	314	0	20
Pr10	276	160	403	403	321	0	25

Table 5.5 shows the results of E_1 based on the settings in Table 5.4. Each row in the table represents a different scenario where the total scores of each action differ from one scenario to another (except the total waiting-time scores where the value of each waiting moment is equal to zero (based on Table 5.4)). In addition, we normalise all scores for all types of actions between 0 to 1 (the total score is the summation of each moment where the total score of any action will not exceed the total of moments in the activity).

Table 5.6: The result of the E_2 on the HM

Scenario	Activity		Connection		Waiting		Number
Code	Time (Minutes)	Total Scores	Time (Minutes)	Total Scores	Time (Minutes)	Total Scores	of POIs
Pr02	227	125	384	384	389	194	19
Pr03	197	93	368	368	435	217	17
Pr04	223	120	382	382	395	197	19
Pr05	389	245	351	351	260	130	24
Pr06	304	145	371	371	325	162	24
Pr08	252	120	417	417	331	166	18
Pr09	222	120	464	464	314	157	20
Pr10	276	160	403	403	321	160	25

Table 5.6 shows the result of E_2. Here, the main difference between E_1 and E_2 is that the waiting-time score produces different values. However, the results of E_1 and E_2 are the same except for the total waiting-time score. The main reason why these results are similar is that we have modelled the HM without including waiting time in between POIs. In other words, the HM has only two options (choosing a POI or moving from one POI to another) where no option is considered for waiting at any time between POIs.

Table 5.7: The result of the E_3 on the HM

Scenario	Activity		Connection		Waiting		Number
Code	Time (Minutes)	Total Scores	Time (Minutes)	Total Scores	Time (Minutes)	Total Scores	of POIs
Pr02	227	125	413	214	360	0	19
Pr03	222	110	436	255	342	0	18
Pr04	220	117	396	292	384	0	18
Pr05	376	234	365	281	259	0	23
Pr06	298	141	468	294	234	0	23
Pr08	283	139	454	246	263	0	20
Pr09	268	150	415	295	317	0	23
Pr10	254	145	374	252	372	0	23

Table 5.7 shows the results of E_3. Here, the total time of actions is different from E_1 and E_2 (see Table C.3 in Appendix C which shows the tour in E_3 differs from E_1 and E_2).

Table 5.8: The result of the E_4 on the HM

Scenario	Activity		Connection		Waiting		Number
Code	Time (Minutes)	Total Scores	Time (Minutes)	Total Scores	Time (Minutes)	Total Scores	of POIs
Pr02	249	146	414	110	337	0	21
Pr03	197	93	443	110	360	0	17
Pr04	220	117	366	145	414	0	18
Pr05	376	234	367	108	257	0	23
Pr06	270	128	393	135	337	0	21
Pr08	252	120	446	133	302	0	18
Pr09	248	130	420	123	332	0	22
Pr10	254	145	457	143	289	0	23

Table 5.8 shows the results of E_4 which is very similar to E_3 except that the connections scores are divided in half. The results of E_4 show that the travellers

preferences affect the building of the tour. However, Experiments E_5 to E_8 have been conducted on $Dataset_2$, and Tables 5.9 to 5.12 show the results of each experiment.

Table 5.9: The result of the E_5 on the HM

Scenario Code	Activity Time (Minutes)	Activity Total Scores	Connection Time (Minutes)	Connection Total Scores	Waiting Time (Minutes)	Waiting Total Scores	Number of POIs
Pr11	205	87	394	394	401	0	19
Pr12	82	41	201	201	717	0	7
Pr13	273	130	347	347	380	0	22
Pr14	309	173	259	259	432	0	24
Pr15	429	273	327	327	244	0	30
Pr16	324	162	345	345	331	0	26
Pr17	231	102	379	379	390	0	18
Pr18	287	141	350	350	363	0	21
Pr19	288	151	351	351	361	0	25
Pr20	324	184	313	313	363	0	29

Table 5.9 shows the results of E_5 based on the settings in Table 5.4. Each row in the table represents a different scenario where the total scores differ from one scenario to another. In addition, we normalised all the scores for all types of actions between 0 to 1.

Table 5.10: The result of the E_6 on the HM

Scenario Code	Activity Time (Minutes)	Activity Total Scores	Connection Time (Minutes)	Connection Total Scores	Waiting Time (Minutes)	Waiting Total Scores	Number of POIs
Pr11	205	87	394	394	401	200	19
Pr12	82	41	201	201	717	359	7
Pr13	273	130	347	347	380	190	22
Pr14	309	173	259	259	432	216	24
Pr15	429	273	327	327	244	122	30
Pr16	324	162	345	345	331	166	26
Pr17	231	102	379	379	390	195	18
Pr18	287	141	350	350	363	182	21
Pr19	288	151	351	351	361	180	25
Pr20	324	184	313	313	363	182	29

Table 5.10 shows the results of E_6, and, as we explained in E_2, the main difference between E_1 and E_2 is that the waiting-time score has different values. However, the results of E_5 and E_6 are the same except for the total waiting-time score.

Table 5.11: The result of the E_7 on the HM

Scenario Code	Activity Time (Minutes)	Activity Total Scores	Connection Time (Minutes)	Connection Total Scores	Waiting Time (Minutes)	Waiting Total Scores	Number of POIs
Pr11	182	79	395	200	423	0	17
Pr12	51	26	268	169	681	0	5
Pr13	266	127	341	209	393	0	21
Pr14	292	160	340	241	368	0	23
Pr15	406	259	315	183	279	0	28
Pr16	298	141	377	227	325	0	23
Pr17	237	103	364	236	399	0	19
Pr18	290	144	330	216	380	0	22
Pr19	306	164	343	209	351	0	26
Pr20	324	184	351	205	325	0	29

Table 5.11 shows the results of E_7 where the connections scores are random between 0 to 1. The main observation in this result is that the number of POIs is less than the number of POIs in E_5 and E_6. We can characterise this as a more comfortable trip style in which fewer POIs are recommended for travellers who wish to have a more relaxed trip.

Table 5.12: The result of the E_8 on the HM

Scenario Code	Activity Time (Minutes)	Activity Total Scores	Connection Time (Minutes)	Connection Total Scores	Waiting Time (Minutes)	Waiting Total Scores	Number of POIs
Pr11	195	81	355	94	450	0	18
Pr12	51	26	221	70	728	0	5
Pr13	266	127	371	113	363	0	21
Pr14	309	173	312	78	379	0	24
Pr15	397	251	331	112	272	0	27
Pr16	308	148	382	110	310	0	25
Pr17	231	102	401	111	368	0	18
Pr18	283	139	293	81	424	0	20
Pr19	306	164	363	96	331	0	26
Pr20	328	184	329	81	343	0	30

Table 5.12 shows the results of E_8, which is very similar to E_7 except that the connections scores are divided in half, and the results are different where the total of waiting time in E_8 is more than in E_7.

As shown above, the HM has a significant impact on building tour trips based on user preferences because the HM considers all components of the tour trip (i.e. activities, connections, and waiting times). The next section will discuss the impact of the HM on building a tour trip.

5.5.2 Discussion

This section discusses the results of the HM and states the findings. In general, as we do not provide a waiting option for travellers between POIs, so changing the weight of waiting-time might not change the results.

In experiments E_2 and E_6, we have assigned a value of 0.5 to each waiting moment. However, because the algorithm does not consider taking an option to wait, the results are dependant on the activity and connection values. The paths of E_2 and E_6 are the same as in E_1 and E_5 because the main differences is the waiting-time value.

Tables 5.13 and 5.14 show a comparison between different preference values for waiting time: in E_1 and E_5 the waiting time is equal to 0 while in E_2 and E_6 the waiting time is equal to 0.5. The main finding of these comparisons is the different waiting-time values do not affect the recommended tour. The main reason for this is that we do not assume the travellers would like to have the option of waiting between activities. In other words, we do not factor in waiting times for travellers between POIs. However, the total waiting-time values differ when the waiting-time preferences vary. To sum up, we assume that travellers do not prefer to wait between POIs and the results of different waiting-time preferences do not affect the tour trip.

Table 5.13: Comparing the results of all scenarios in E_1 and E_2

Experiment	Scenario Code	Activity		Connection		Waiting		Number of POIs
		Time (M)	Score	Time (M)	Score	Time (M)	Score	
E_1	Pr02	227	125	384	384	389	0	19
E_2		227	125	384	384	389	194	19
E_1	Pr03	197	93	368	368	435	0	17
E_2		197	93	368	368	435	217	17
E_1	Pr04	223	120	382	382	395	0	19
E_2		223	120	382	382	395	197	19
E_1	Pr05	389	245	351	351	260	0	24
E_2		389	245	351	351	260	130	24
E_1	Pr06	304	145	371	371	325	0	24
E_2		304	145	371	371	325	162	24
E_1	Pr08	252	120	417	417	331	0	18
E_2		252	120	417	417	331	166	18
E_1	Pr09	222	120	464	464	314	0	20
E_2		222	120	464	464	314	157	20
E_1	Pr10	276	160	403	403	321	0	25
E_2		276	160	403	403	321	160	25

Table 5.14: Comparing the results of all scenarios in E_5 and E_6

Experiment	Scenario Code	Activity		Connection		Waiting		Number of POIs
		Time (M)	Score	Time (M)	Score	Time (M)	Score	
E_5	Pr11	205	87	394	394	401	0	19
E_6		205	87	394	394	401	200	19
E_5	Pr12	82	41	201	201	717	0	7
E_6		82	41	201	201	717	359	7
E_5	Pr13	273	130	347	347	380	0	22
E_6		273	130	347	347	380	190	22
E_5	Pr14	309	173	259	259	432	0	24
E_6		309	173	259	259	432	216	24
E_5	Pr15	429	273	327	327	244	0	30
E_6		429	273	327	327	244	122	30
E_5	Pr16	324	162	345	345	331	0	26
E_6		324	162	345	345	331	166	26
E_5	Pr17	231	102	379	379	390	0	18
E_6		231	102	379	379	390	195	18
E_5	Pr18	287	141	350	350	363	0	21
E_6		287	141	350	350	363	182	21
E_5	Pr19	288	151	351	351	361	0	25
E_6		288	151	351	351	361	180	25
E_5	Pr20	324	184	313	313	363	0	29
E_6		324	184	313	313	363	182	29

Table 5.15 compares the results of different experiments. First, E_1 features at least 50% more POIs in the recommended tour results because all the connections values (the edges between POIs) are equal to 1 (representing the highest satisfaction level). The second finding is that E_3 and E_4 perform better than E_1 in terms of reducing waiting time. The main reason that E_1 performs less effectively than E_3 and E_4 is that it involves choosing the longest path between POIs because users

prefer a long journey (moving between POIs). In addition, E_4 produces a tour that features a longer total connection time by at least 50% because a short connection period is selected that allows the model able to select more POIs.

Another point is that the HM aims to personalize tour trips based on user preferences and the results show that the HM successfully achieves this aim. Figure 5.9 shows the difference between the HM and the TOPTW based on the same dataset ($Datasets_2$) and the same algorithm.

Figure 5.9 shows four tours based on different experiments (E_5, E_7, E_8, and TOPTW). The main difference between these experiments is the different preferences for the connections scores and waiting-time scores. First of all, there are no inherently good or bad results here (E_5, E_7, E_8, and TOPTW) because all of these tours are based on specific user preferences (where one user may like a particular tour whereas another user may dislike the same tour). Broadly, we refer to the main purpose of RSs (i.e. personalisation) in which the HM helps travellers to personalise their tours based on their specific preferences.

Table 5.15: Comparing the results of all experiments

Scenario Code	Experiment	Activity		Connection		Waiting		Number of POIs
		Time (M)	Score	Time (M)	Score	Time (M)	Score	
Pr02	E_1	227	125	384	384	389	0	19
	E_3	227	125	413	214	360	0	19
	E_4	249	146	414	110	337	0	21
Pr03	E_1	197	93	368	368	435	0	17
	E_3	222	110	436	255	342	0	18
	E_4	197	93	443	110	360	0	17
Pr04	E_1	223	120	382	382	395	0	19
	E_3	220	117	396	292	384	0	18
	E_4	220	117	366	145	414	0	18
Pr05	E_1	389	245	351	351	260	0	24
	E_3	376	234	365	281	259	0	23
	E_4	376	234	367	108	257	0	23
Pr06	E_1	304	145	371	371	325	0	24
	E_3	298	141	468	294	234	0	23
	E_4	270	128	393	135	337	0	21
Pr08	E_1	252	120	417	417	331	0	18
	E_3	283	139	454	246	263	0	20
	E_4	252	120	446	133	302	0	18
Pr09	E_1	222	120	464	464	314	0	20
	E_3	268	150	415	295	317	0	23
	E_4	248	130	420	123	332	0	22
Pr10	E_1	276	160	403	403	321	0	25
	E_3	254	145	374	252	372	0	23
	E_4	254	145	457	143	289	0	23

Table 5.16: Comparing the results of scenario Pr12 in E_5, E_7, and E_8

Experiment	Scenario	Activity		Connection		Waiting		Number
	Code	Time (M)	Score	Time (M)	Score	Time (M)	Score	of POIs
E_5	Pr12	82	41	201	201	717	0	7
E_7	Pr12	51	26	268	169	681	0	5
E_8	Pr12	51	26	221	70	728	0	5
TOPTW	Pr12	66	39	320	0	614	0	6

Table 5.16 shows a comparison between the results of experiments E_5, E_7, E_8, and TOPTW. The main observation here is the number of POIs in each tour: E_5 generated seven POIs whereas E_7 and E_8 generated five POIs. In addition, the total connection time in E_5 is less than all the other results by at least 10%.

Table 5.17: Comparing the results of scenario Pr12 in E_5, E_7, and E_8

Experiment	Scenario Code	Path
E_5	Pr12	0 27 19 17 79 14 32 65 0
E_7	Pr12	0 6 52 4 35 87 0
E_8	Pr12	0 6 52 21 32 33 0
TOPTW	Pr12	0 6 52 21 32 41 95 0

Table 5.17 shows the recommended tour for different experiments. The most noteworthy finding here is that the results of E_5 differ from those of the other experiments on almost all of the recommended POIs. However, E_7, E_8, and TOPTW feature some common POIs where E_8, and TOPTW are similar to each other (almost 66% similarity).

Figure 5.9 shows that E_5 recommended the shortest total distance between nodes comparing to other experiments. In conclusion, our HM can more successfully personalize tour trips based on user's preferences compared to the existing models in a wider variety of ways, some of which can be considered to be highly intuitive.

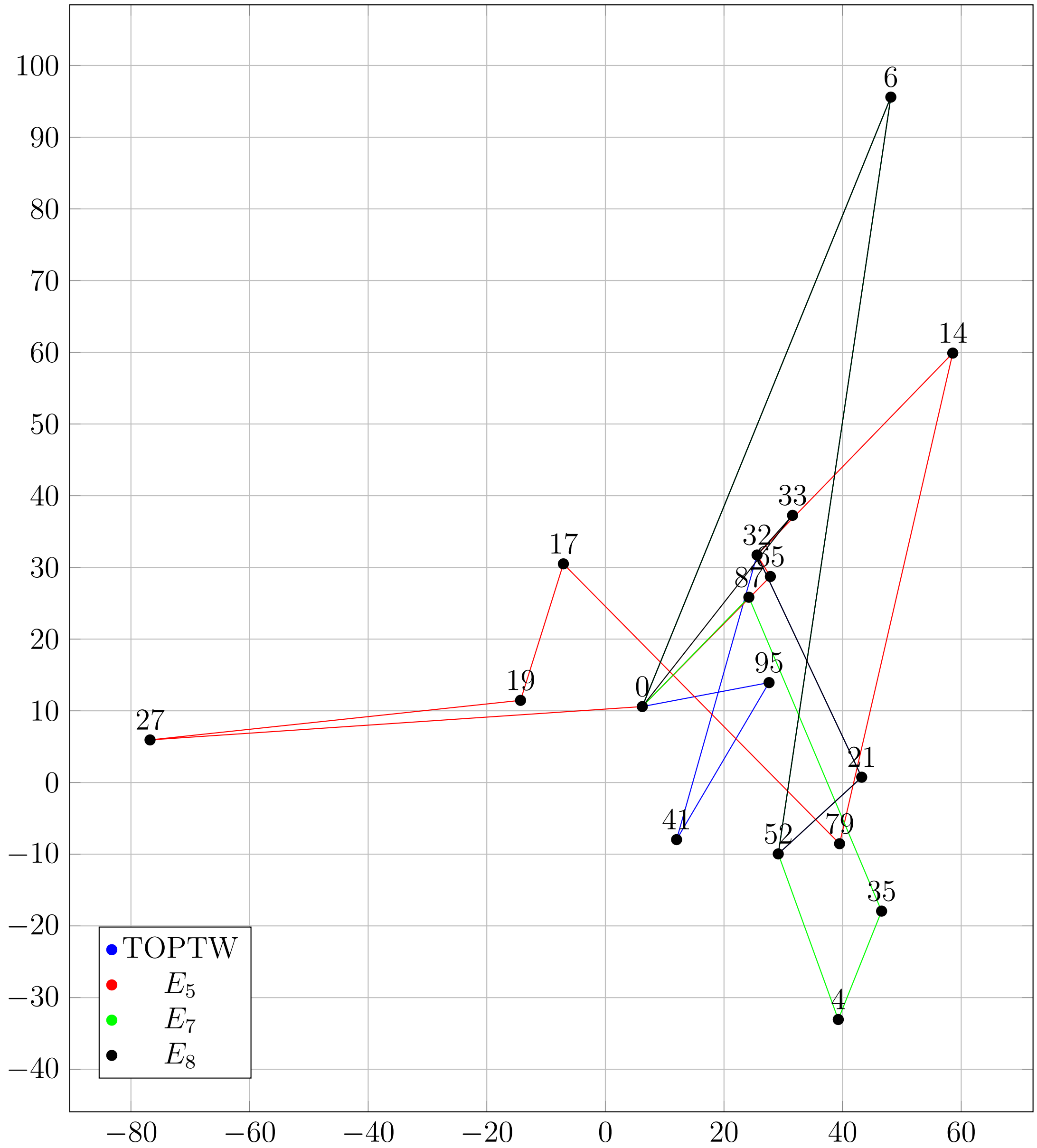


Figure 5.9: Comparing the HM path with TOPTW path based on $Pr12$ on $Datasets_2$

5.6 Conclusion

This chapter has presented our HM that is designed to personalise recommended tours based on user's preferences. Furthermore, our results show that the HM places a greater emphasis on personalizing tour trips much more effectively and intuitively than the existing models.

To the best of our knowledge, the HM is the first dynamic model to successfully customize tour trips based on tour trip's specific components:(*Activity*, *Connection*, and *Waiting Time*).

Chapter 6

Group Tourist Trip Design Problem

This chapter investigates the problems related to *Group Recommendation Systems* (GRSs). We have introduced the Group Tourist Trip Design Problem (GTTDP), which deals with the problem of a group of travellers going to the same destination although they might have different preferences and constraints. Also, we propose the Group Tour Trip Recommender Model (GTTRM), which has been designed to maximize the *satisfaction*[1] level among a group of travellers.

In addition, we have also developed an algorithm based on *Ant Colony Optimization*, for GTTRM. The proposed algorithm is used to decide how to split the group into subgroups (see Section 6.4).

6.1 Introduction

While the previous chapters have dealt with tour trips for individuals, this chapter deals with tour trips based on a group of travelers. Generally, people travel in groups as couples, families or friends, so when recommendations for two or more people are needed, GRSs are applicable and useful. Nowadays, due to the increased access to cheap and efficient transport systems, the number of people traveling internation-

[1]See Glossary for definitions of italicized words

Table 6.1: List of features have been used in GTTDP

Abbreviation	Full Form	Meaning
M	Multi-day	Building tour trip for more than a day
SRF	Social Relationship Factor	Extra values for a user of the group when two users visit the same place at the same time
SV	Score Value	Each POI has a value
CV	Connection Value	Moving from a POI to another has a value
WV	Waiting time Value	When the time of tour trip is less than T_{max}
SG	Splitting Group	Divide the group of tour into subgroups
IC	Item Constraints	Dynamic constraints on items (POIs)
TW	Time Windows	Opening/closing time
MC	Multi-Constraint	Limited number of type of category not exceed
UC	User Constraints	Considering difference constraints from member of the group

ally is ever increasing. However, this presents certain difficulties for inexperienced travelers planning international trips[1] [112]. Borras et al. mention that a wealth of information is available on the internet for tourists concerning travel destinations, points of interest (POI), activities, and events, and these tend to be unpersonalized and overwhelming when booking a group trip tour [20].

The need for GRS has become imperative [35, 61], and GRS are required to process many varied scenarios such as recommendations for suitable TV programs for friends to watch together or choosing the most suitable restaurant for colleagues [52]. By comparison, personalized recommendation techniques provide customized information based on the user's specific desires[1] or limitations[1] [20]. Aside from this, Xie et al. posit that travelers are interested in recommended POIs customized by time limitations, budget limitations [110], and distance limitations . Yoon et al. explained that travelers face a tricky decision when faced with many possible places to visit and also that those who wish to visit places, which are of interest, have limited time available (T_{max}) [112]. In addition, Xie et al. point out that to choose a trip tour independently and set budget, a package of POIs should not only be compatible, but should also apply user's wishes (e.g. visiting POIs within 10 km of their hotel and including no more than three museums and not more than two parks) [110]. Finally, Salam et al. mention that the job of GRSs is to recommend items to a group that reflect the preferences of the group as one, and these items should be reasonable and acceptable to all group members [10, 40, 88].

6.1.1 GRSs Classifications

A universal classification of GRS has not yet been agreed. For instance, Bok et al. posit that recent GRSs schemes should be divided into collaborative filtering-based recommendation schemes and social-based recommendation schemes [18] while Guo et al. propose that GRS methods should be classified into recommendation aggregation and preference aggregation [55]. In addition, Ghazarian and Nematbakhsh add that the majority of GRS strategies are divided into individual rating aggregation and the aggregation of individual recommendations [52]. In the present study, the GRSs approaches are divided into (1) aggregation approaches and (2) optimization approaches. The former improves the quality of recommendations for a group of users by enhancing or modifying algorithms to fit into GRSs such as optimization algorithms (see section 6.2).

6.1.2 Problems of GRSs

As some trip activities are to be carried out by users together as groups [52, 61], new issues have arisen in GRSs. For example, specific differences such as aggregate users' profile[1], users' roles, and limitations have become influential. Furthermore, GRSs can recommend tour trips to a group based on matching all group member's constraints and preferences. In addition, GRSs deal with diverse users who may not share similar preferences [7, 52]. Equally important, Ghazarian and Nematbakhsh emphasize that the challenges facing GRS fall into four areas: collecting information about users' preferences, generating recommendations, explaining recommendations, and helping the group to reach a final decision [40, 52]. For instance Salam et al. mention that it is unreasonable to provide recommendations to a group where all members have consistent choices or similar preferences because it operates as a single-user scenario. On the contrary, groups of users are likely to have a range of preferences that result in conflicting needs, and also, particular groups are likely to consist of users who are less easy to satisfy than others [27, 88].

6.1.3 Group Tourist Trip Design Problem

Uniquely, Traveller Recommender Systems (TRSs) recommend sequences of different items such as POIs, meals, or accommodation. The role of GRS for travelers is dealing with different tourists in a group to maximize their satisfaction levels, and reduce the member's conflicts constraints and preferences. In addition, the challenges that generating group travel recommendations represent for GRSs as "group decisions are more complex" [35, 84]. In addition, because GRS issues affect this process, it is considerably more complicated.

The Tourist Trip Design Problem (TTDP), a trip plan for a tourist interested in visiting multiple POIs, has been defined by [107]. The TTDP is based on a tourist wishing to visit different POIs with limited time (Time Windows) for sightseeing; each POI has a set of attributes (e.g., category, location, child-friendly, admission cost, etc.), and each trip provides a limited time (T_{max}) for sightseeing. Therefore, the TTDP should maximize the total score gained when visiting a specific POI (with each POI having different scores); the system not only chooses POIs but also selects the best route(s) between selected POIs. The TTDP's solutions should respect traveler's constraints and POI's characteristics [49]. The main objective of the TTDP is to maximize tourist's satisfaction level by allowing them to visit high-scoring POIs while taking the user's constraints into account.

The Group Tourist Trip Design Problem (GTTDP) is an extension of the TTDP; the main difference between them is that the GTTDP deals with a group of travellers, and the main challenge of GTTDP is how to deal with conflict constraints or preferences.

6.1.4 Group Tour Trip Recommender Model

Group Tour Trip Recommender Model (GTTRM) is designed to solve conflicts among travelers using several different strategies and algorithms (see Section 6.3). The primary approach has been implemented is splitting the group into a subgroup for part of the trip time to maximize the satisfaction level in the case that a complex conflict appears.

The GTTRM is a novel model that can decide if it is possible to divide the group into subgroups to maximize satisfaction levels based on measuring the similarities among the group members. In addition, the GTTRM calculates where and when should the group be divided.

6.2 Related work

The biggest challenge for GRSs is building a recommender system for a group that is suitable for all group members. A range of different approaches have been adopted to deal with the GRSs as detailed next.

The GRSs approaches consist of aggregation methods classified into three categories: (1) individual preferences aggregation; (2) individual rating aggregation; and (3) individual recommendations aggregation [62]. First, individual preferences aggregation combines all group members' preferences into a group preference; this is sometimes called the construction of group preference models [62]. Crucially, one of the main problems the GRSs must solve is how to adapt users' preferences to the group's preferences as a whole [75]. In this method, the users' preferences are aggregated into a group preference (the group model G). Then, for each candidate item, the G model is used to predict the ratings. Second, individual rating aggregation combines the ratings of each item from each group member. Third, by applying individual RSs for each member of the group, the recommendations for the group are then aggregated.

However, before discussing the aggregation methods themselves, it is necessary to examine the different techniques they adopt. For instance, Masthoff mentions 11 aggregation strategies, which have been widely used in GRSs, and these are applied to different types of aggregations [75]. Table 6.2 shows the aggregation strategies with explanations on how each one works. Most of the previous works use one of the aggregation strategies shown in Table 6.2, however, sometimes, there are small variations from this [75].

Table 6.2: Common aggregation techniques are used in GRSs

Strategy	How it works
Plurality Voting	The item with a higher number of votes is chosen
Average	Averages of individual ratings
Multiplicative	Multiplies of individual ratings
Borda Count	Ordering each user's preferences and assigning a value for each item. The lowest rated item is awarded 0, and the next one is 1, and so on and so forth
Copeland Rule	Count how many times an item beats other items, and minus for when it loses
Approval Voting	Count how many times it has been rated
Least Misery	Take users' minimum rating
Most Pleasure	Take users' maximum rating
Average without Misery	Average of users' ratings with the exception any rate below a certain threshold
Fairness	Top items from all users are selected
Most respected person	Using the most-respected user's rating

6.2.1 Recommendation aggregation

Individual recommendations aggregation combines individual recommendation lists into group recommendations using various different methods. In [10,37], Christensen and Schiaffino implemented six different aggregation techniques for the GRSs. One of these techniques involves merging individual recommendations [37]. This technique is based on the generation of recommendations for individuals; it is easy to implement because it is an extension of existing RSs. Also, Baltrunas et al. discuss four different rank aggregation methods; (i) Spearman footrule which is a method to minimise the average distance between individual ranking items, (ii) Borda count, (iii) Average, and (iv) Least Misery [10]. Next, in terms of the most performed technique, Masthoff conducted experiments to discover which strategy is best [75]. The researcher found that users cared about increasing fairness and decreasing misery. However, Masthoff clarified that the multiplicative strategy created the highest satisfaction levels among users.

6.2.2 Preference aggregation

In [5, 11, 19, 37, 44–46, 66, 76, 88, 93, 114], aggregation strategies are used to build a common profile based on the aggregation of individual preferences. In other words, each user has preferences (or likes) such as parks, museums, Indian food, e.g., so this method aggregates these preferences to create a common group preference. Users can choose one or more preferences, and it is possible that some users may choose the same preferences. Then, the aggregation function generates a group preference that represents the whole group. [114], Yu et al. propose a strategy to generate group preference by measuring users' preference for TV viewing. The proposed strategy is based on total distance minimization to calculate the distance between users' preferences. Each user is able to like and unlike particular preferences, so when a user likes a particular preference, this is assigned a value of 1 and when it is unliked, it is assigned a value of -1. In addition, when the state of a user's preference is unknown, the preference is assigned a value of 0. In [66], Kim et al. proposed a GRS that increases the effectiveness of group recommendations and increases group member's satisfaction. This system applies a CF model and builds a group preference that aggregates group members' preferences to produce a candidate recommendation set. If a user is a member of the group and has read a particular book, it is assigned 1. Next, the group rating for each item is formed. The following process finds the similarity between group profiles using the *nearest-neighbour algorithm*.

6.2.3 Rating aggregation

In [7, 13, 37, 47, 65, 79, 81, 84, 97], Average and Least Misery are the most popular aggregation functions that have been proposed in recommendations or rating aggregations [7]. However, in [7], a consensus function is proposed that consists of relevance and disagreement functions. The relevance function aggregates all group members' ratings using the Average and Least Misery strategies, whereas the disagreement functions are calculated based on how much the whole group has liked or unliked a particular item. In other words, the consensus function calculates if an item is worthy of recommendation to a group by calculating its relevance and the disagreement among a group's users. In [13], Berkovsky and Freyne propose a food recommender model that aims to uncover which data aggregation strategy is most appropriate for a group of family members. Four strategies are applied; two are static, and two are based on the user's interactions with the content. The first is a uniform model where each user has the same weight. The second, a heuristic model, is role-based, using an applicant's weight = 0.5, their partner's weight = 0.3, and a child's weight = 0.1. The third is a role-based model that, for each user, calculates a specific weight based on their activities which represents how many ratings have been observed from this user. In [47], Gartrell et al. proposed a framework for a group recommender system that analyses different group characteristics. This framework is designed to predict group preferences by implementing a group-consensus function. Their association rules have been designed to discover interesting relationships or patterns between items or users in a dataset.

6.2.4 Optimization of GRSs

To the best of the author's knowledge, only these papers [8, 98] have solved the GRSs based on the optimizations approach. All of these papers developed their model based on the OP. Because the challenges in solving the problem for individual tourists remain partially unsolved, a few researchers have proposed models for this problem for a group of travelers.

Solving the GRSs as optimization problem is an effective approach because (1) the GRSs for travellers includes the problem of the TRSs, which is data problem (for more information see Chapter 2), and (2) personalisation is required for various users' tastes. The OP is the closest model to solving TTDP, and the OP is a special case of TTDP because the OP is based on base constraints.

The authors in [8] formulate the OP very similarly to the MOOP (in Chapter

5), and each k score from POIs represents the users(U_k) scores. In addition, they have applied three aggregation techniques (Sum, Least Misery, Average). Moreover, the authors in [98] have designed the MCMTOPTW based on the Multi-Constraint TOPTW (MCTOPTW). The MCMTOPTW considers the social relationship factor, which, when some users visit a POI at the same time together, an extra value will be added based on this factor.

The existing models neglect some important factors to build group tour trip: (1) constraints from different users, (2) multi-values for POIs, (3) connection values, (4) waiting time values and (5) aggregations for constraints and preferences. The GTTRM has been designed to tackle these issues. Section 6.3 illustrates the GTTRM in more detail.

6.3 Group Tour Trip Recommender Model

This section illustrates the Group Tour Trip Recommender Model (GTTRM), which is a model that can solve the problem of the GTTDP. First, it makes a comparison between the GTTRM and existing models. Second, the overview of the GTTRM shows how it is built on other models. Third, a constraint model has been developed based on the ICDM (see Chapter 3). Fourth, the mathematical model is designed for the GTTRM.

6.3.1 Introduction

Existing models are subject to several limitations such as dealing with different constraints from the group members. The GTTRM is introduced to tackle these limitations. Table 6.1 shows all the features considered in the GTTDP. In addition, Table 6.3 shows the main differences between the GTTRM and the MCMTOPTW. First, the GTTRM and the MCMTOPTW provide features such as M, SRF, and SG.

Second, these models feature slight differences in SV, IC, and TW. The SV in the GTTRM considers multi-score values (MSV) whereas the MCMTOPTW consider only a single score value for each POI form each user in the group. The

MCMTOPTW considers only entrance fees (F) whereas the GTTRM builds on the top of ICDM, which deals with dynamic constraints on items. Also, in the TW, the GTTRM deals with Multi-Time Windows (MTW) for each POI.

Third, the GTTRM overcomes the MCMTOPTW by considering the features of the HM (see Chapter 5) which are SV, CV, and WV, and also aggregates user constraints (UC) where is each user of the group might have different constraints from another member of the group.

Table 6.3: Comparison between the GTTRM with the MCMTOPTW

Problem	M	SRF	SV	CV	WV	SG	IC	TW	MC	UC
GTTRM	•	•	MSV	•	•	•	•	MTW		•
MCMTOPTW	•	•	•			•	F	•	•	

6.3.2 Overview of the GTTRM

This section reviews the GERM, and presents the components of the model. Based on the ICDM (see Chapter 3), we have implemented the ICDM into the activity and connection constraints. Mainly, the ICDM has been developed, which is called the Constraints Data Model (CDM), to handle constraints and references from a group of users. In particular, the CDM is composed of (1) the Activity Constraints Data Model (ACDM) and (2) the Connection Constraints Data Model (CCDM). Section 6.3.3 illustrates the CDM in detail.

The GTTRM provides two aggregations methods: (1) Group Aggregation (GA) and (2) User Aggregation (UA). First, the GA is a method for aggregating all user constraints and preferences into a group profile. Second, the UA is a method where the algorithm takes part in building recommended tours. Mainly, the GTTRM Algorithm decides to aggregate some users together to maximize the users' satisfaction and reduce the conflict among the group members. Figure 6.1 provides an overview of the GA and Figure 6.2 provides an overview of the UA.

Table 6.4 illustrates the main differences between the aggregation methods in the GTTRM. First, the GA does not support making sub-routes for the group because

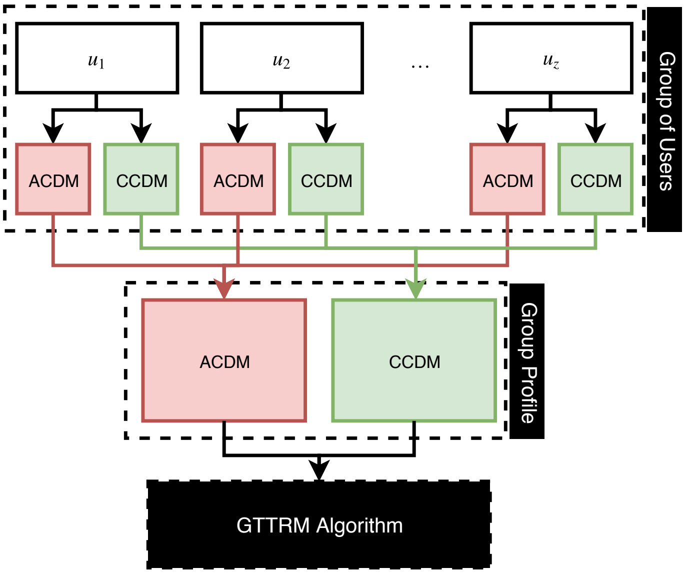


Figure 6.1: Overview of the group aggregation for the GTTRM

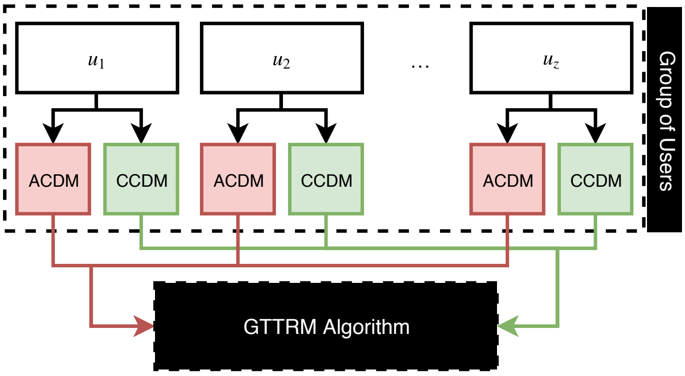


Figure 6.2: Overview of the user aggregation for the GTTRM

the aim of the method is aggregating all group members into one file. However, the UA aims to maximize the satisfaction level for each user in the group by considering other options, such as making sub-routes. Second, the GA produces one file for the group based on one of the aggregation's techniques (see Table 6.2) whereas the UA deals with users in the group where every user is treated as an individual by considering other users in the group in calculating the probability. Third, the GA reduces the search space by merging all users' preferences and constraints into one profile. Finally, the UA is able to produce better results by maximizing user satisfaction because it recommends options based on each user's preferences.

Table 6.4: Comparison between the aggregation methods

Features	Group Aggregation	User Aggregation
Splitting the group		•
Ignoring different tasting	•	
Reducing the search space	•	
Maximising satisfaction level		•

6.3.3 Constraints Data Model

The ICDM is designed for the different constraints proposed by one user. Therefore, the CDM is developed based on the ICDM to build a constraints model for a group of users. Importantly, the CDM is divided into (1) the Activity Constraints Data Model (ACDM) and (2) the Connection Constraints Data Model (CCDM). The term ACDM is a relatively new name for the ICDM: the main difference is that the ACDM considers a group of users. Comparatively, the CCDM is a model that manipulates *connection* constraints and preferences for a group of users.

In general, the main differences between the ACDM and the CCDM is based on the data and constraints will apply. The ACDM is designed to match activity data with the user's constraints, whereas the CCDM is designed to confirm the connection data with the user's constraints.

The CDM is defines as follows: $u_z \in G$ is denoted as a user in the group G where $z = 1, 2, \ldots, |G|$, and each user u_z might have n constraints (HC and/or SC). $ActHC_{u_z}$ and $ConHC_{u_z}$ represent a set of hard constraints from the user u_z for *Activity* and *Connection*, $hc(Act)^{u_z}_{ptim} \in ActHC_{u_z}$ and $hc(Con)^{u_z}_{ptim} \in ConHC_{u_z}; \forall u_z \in U; \forall p \in P; \forall t \in p; \forall i \in I$, where $m = 1, 2, \ldots, (|ActHC_{u_z}|$ or $|ContHC_{u_z}|)$.

Also, $ActSC_{u_z}$ and $ConSC_{u_z}$ are set of soft constraints from the user u_z for *Activity* and *Connection*, $sc(Act)^{u_z}_{ptim} \in ActSC_{u_z}$ and $sc(Con)^{u_z}_{ptim} \in ConSC_{u_z}$, where $v = 1, 2, \ldots, (|ActSC_{u_z}|$ and $|ConSC_{u_z}|)$.

$$ActHC^{u_z}_{pti} = \prod_{m=1}^{|ActHC_{u_z}|} hc(Act)^{u_z}_{ptim} \qquad (6.3.1)$$

$$ActHC_{pti}^{G} = \prod_{z=1}^{|G|} ActHC_{pti}^{u_z} \tag{6.3.2}$$

Equation (6.3.1) combines all the HCs for the user u_z, and Equation (6.3.2) combines all the HCs for all users in the group. Equation (6.3.2) represents the *Group Aggregation* (GA).

$$ConHC_{ptij}^{u_z} = \prod_{m=1}^{|ConHC_{u_z}|} hc(Con)_{ptijm}^{u_z} \tag{6.3.3}$$

$$ConHC_{ptij}^{G} = \prod_{z=1}^{|G|} ConHC_{ptij}^{u_z} \tag{6.3.4}$$

Equations (6.3.3) and (6.3.4) calculate the HC for user and group.

$$ActSC_{pti}^{u_z} \text{ and } ConSC_{ptij}^{u_z} = \text{Aggregation methods for the user } (u_z) \tag{6.3.5}$$

(see Table 6.5)

$$ActSC_{pti}^{G} \text{ and } ConSC_{ptij}^{G} = \text{Aggregation methods for the group } (G) \tag{6.3.6}$$

(see Table 6.5)

The equations 6.3.5 and 6.3.6 represent the aggregations of all SCs for the user u_z and the group G.

$$\sum_{v=1}^{|ActSC_{u_z}|} W_v = 1 \text{ and } \sum_{v=1}^{|ConSC_{u_z}|} W_v = 1 \tag{6.3.7}$$

$$\sum_{u_z=1}^{|G|} W_{u_z}^{G} = 1 \text{ and } \sum_{u_z=1}^{|G|} W_{u_z}^{G} = 1 \tag{6.3.8}$$

Equation 6.3.7 shows that each SC might have a different weight, which represents the importance of one SC compared with another. Equation 6.3.8 represents the ability to adjust the different expertise of group members where a user with a high level of expertise is assigned a higher weighting.

$$A_{pti}^{u_z} = ActHC_{pti}^{u_z} \times ActSC_{pti}^{u_z} \tag{6.3.9}$$

$$A_{pti} = ActSC_{pti}^{G} \times ActSC_{pti}^{G} \tag{6.3.10}$$

$$C_{ptij}^{u_z} = ConHC_{ptij}^{u_z} \times ConSC_{ptij}^{u_z} \tag{6.3.11}$$

$$C_{ptij} = ConHC_{ptij}^{G} \times ConSC_{ptij}^{G} \tag{6.3.12}$$

Equation (6.3.9) shows the aggregation between equations (6.3.1) and (6.3.5) to calculate the weight that represents the degree of satisfaction for user u_z in each item i on day p at time t based on the user's constraints. Equation (6.3.10) gathers the group's SCs and HCs.

$$ActHC_{pti}^{u_z}, ActHC_{pti}^{G}, ConHC_{pti}^{G}, ConHC_{pti}^{u_z} \in \{0,1\} \tag{6.3.13}$$

$$ActSC_{pti}^{u_z}, ActSC_{pti}^{G}, ConSC_{pti}^{G}, ConSC_{pti}^{u_z} \in \mathbb{Q}; \tag{6.3.14}$$

$$\text{Where is } 0 \leq ActSC_{pti}^{u_z}, ActSC_{pti}^{G} \leq 1;$$

$$\text{Where is } 0 \leq ConSC_{pti}^{G}, ConSC_{pti}^{u_z} \leq 1$$

Table 6.5: Aggregation methods for the SC

Method Name	Description	Equation
Sum	Calculate the sum of all elements in SC	$\sum_{v=1}^{\lvert ActSC_{u_z}\rvert} W_v \times sc(Act)_{ptim}^{u_z}$
Least Misery	Take the minimum value of SC	$Min(ActSC_{u_z})$
Most Pleasure	Take the maximum value of SC	$Max(ActSC_{u_z})$
Multiplicative	Multiplies each SC value	$\prod_{v=1}^{\lvert ActSC_{u_z}\rvert} W_v \times sc(Act)_{ptim}^{u_z}$

The main function of splitting the group into subgroups is provided by our algorithm. Section 6.4 illustrates how the algorithm works.

6.3.4 Mathematical model

The GTTRM can be defined as follows. Let $G = (I, TT)$ be a directed weighted graph where $i \in I$ and $i = 1, ..., |I|$ are a set of nodes representing a *Point of Interest* (POI) in a city. A travel time between two nodes $i, j \in I$ denotes TT_{ij}, and ST_i denotes the time spent at the i node. Given a starting node s and terminal node t, let the $s = 1$ and $t = |I|$. The trip length may be a day or longer, so let $p \in P$;

$p = \{1, ..., |P|\}$ be a set of trip days. In addition, each trip has start time and end time where $t \in p$; $t = 1, ..., |p|$ are a set of moments in the trip p day. The time limitation for each day of the trip is represented by T_{max}.

Table 6.6: List of notations in GTTRM

Notation	Meaning
$X^{u_z}_{pti}$	Decision variable equal 1 if user u_z visit POI i on day p at time t, otherwise equal 0
$A^{u_z}_{pti}$	The satisfaction level for user u_z to visit POI i on day p at time t (see Equation (6.3.9))
$ActG^{u_z u_o}_{pti}$	Decision variable equal 1 if user u_z and user u_o visit POI i on day p at time t, otherwise equal 0
$V^{u_z}_{u_o}$	The satisfaction level for user u_z if user u_o visit a POI with user u_z
$Y^{u_z}_{ptij}$	Decision variable equal 1 if user u_z travel from POI i to POI j on day p at time t, otherwise equal 0
$C^{u_z}_{ptij}$	The satisfaction level for user u_z to travelling from POI i to POI j on day p at time t (see Equation (6.3.11))
$ConG^{u_z u_o}_{ptij}$	Decision variable equal 1 if user u_z and user u_o travel together from POI i to POI i on day p at time t, otherwise equal 0
$T^{u_z}_{u_o}$	The satisfaction level for user u_z if user u_o travel from a POI to another with user u_z
$Z^{u_z}_{pt}$	Decision variable equal 1 if user u_z does not do anything (Waiting time) on day p at time t, otherwise equal 0
$W^{u_z}_{pt}$	The satisfaction level for user u_z when does not do anything (Waiting time) on day p at time t

The equations below represent the GTTRM with the constraints. In Equation 6.3.15 is the *objective function* which maximizes the total points from three different actions: (1) activity, (2) connection, and (3) waiting for a group of users; the equation has three main functions which are $f_1(a)$, $f_2(c)$, and $f_3(w)$, and each of these functions represents one of the main actions (activity, connection, and waiting).

$$Max\,(f_1(a) + f_2(c) + f_3(w)) \tag{6.3.15}$$

The three functions $f_1(a)$, $f_2(c)$, and $f_3(w)$ are shown in equations 6.3.16, 6.3.18, and 6.3.20.

$$f_1(a) = \sum_{p=1}^{|P|} \sum_{t=1}^{|p|} \sum_{i=1}^{|I|} \sum_{z=1}^{|G|} X^{u_z}_{pti} \times A^{u_z}_{pti} + \sum_{o=1}^{|G|} ActG^{u_z u_o}_{pti} \times V^{u_z}_{u_o} \tag{6.3.16}$$

Equation (6.3.16) represents the satisfaction level for the group G by visiting some POIs. In addition, if user u_z visits some POIs with a member of the group, their satisfaction level may increase.

$$ActG_{pti}^{u_z u_o} = X_{pti}^{u_z} = X_{pti}^{u_o} \tag{6.3.17}$$

$$\forall t = 1, \dots, |p|; \forall p = 1, \dots, |P|; \forall z = 1, \dots, |G|; \forall o = 1, \dots, |G|; \forall i = 1, \dots, |I|$$

Equation (6.3.17) ensures that if user u_z visits a POI at least one of the group member will also visit the POI.

$$f_2(c) = \sum_{p=1}^{|P|} \sum_{t=1}^{|p|} \sum_{i=1}^{|I|} \sum_{j=1}^{|I|} \sum_{z=1}^{|G|} Y_{ptij}^{u_z} \times C_{ptijr}^{u_z} + \sum_{o=1}^{|G|} ConG_{ptij}^{u_z u_o} \times T_{u_o}^{u_z} \tag{6.3.18}$$

Equation (6.3.18) represents the total satisfaction level for the group G by choosing the most preferred connection between i and j. In addition, if user u_z travels from one POI to another with a member of the group, their satisfaction level might increase.

$$ConG_{ptij}^{u_z u_o} = Y_{ptij}^{u_z} = Y_{ptij}^{u_o} \tag{6.3.19}$$

$$\forall t = 1, \dots, |p|; \forall p = 1, \dots, |P|; \forall z = 1, \dots, |G|;$$

$$\forall o = 1, \dots, |G|; \forall i = 1, \dots, |I|; \forall j = 1, \dots, |I|$$

Equation (6.3.19) ensures that if user u_z travels to a POI, at least one of the group member will also travel to the same POI.

$$f_3(w) = \sum_{p=1}^{|P|} \sum_{t=1}^{|p|} \sum_{z=1}^{|G|} Z_{pt}^{u_z} \times W_{pt}^{u_z} \tag{6.3.20}$$

Equation (6.3.20) calculates the total waiting time for all group members to measure the satisfaction level of the group members.

$$\sum_{i=1}^{|I|} X_{pti}^{u_z} + \left(\sum_{i=1}^{|I|} \sum_{j=1}^{|I|} Y_{ptij}^{u_z} \right) + Z_{pt}^{u_z} = 1 \tag{6.3.21}$$

$$\forall t = 1, \dots, |p|; \forall p = 1, \dots, |P|; \forall z = 1, \dots, |G|$$

$$\sum_{j=1}^{|I|} Y_{p11j}^{u_z} = 1 \tag{6.3.22}$$

$$\forall p = 1, \ldots, |P|; \forall z = 1, \ldots, |G|$$

$$\sum_{i=1}^{|I|-1} \left(\frac{\sum_{t=1}^{|p|} Y_{pti|I|}^{u_z}}{TT_{i|I|}} \right) = 1 \tag{6.3.23}$$

$$\forall p = 1, \ldots, |P|; \forall z = 1, \ldots, |G|$$

The Equation 6.3.21 is a constraint that allows one action at the same time. In addition, Equation 6.3.22 ensures that the trip on each p day starts from s which is the start point. Also, Equation 6.3.23 ensures that on each p day the trip ends at e the POI which represents the end point.

$$\frac{\sum_{t=n}^{t_1=n+TT_{sr}} Y_{ptsr}^{u_z} + \sum_{t=t_1+1}^{t_2=t1+ST_r} X_{ptr}^{u_z}}{TT_{sr} + ST_r} = \frac{\sum_{t=t_2+1}^{t_3=t_2+TT_{rm}} Y_{ptrm}^{u_z}}{TT_{rm}} \leq 1 \tag{6.3.24}$$

$$\forall n \in \{1, \ldots, |p-3|\}; \forall p = 1, \ldots, |P|; \forall s, m = 1, \ldots, |I|; \forall r = 2, \ldots, |I-1|;$$

$$\forall z = 1, \ldots, |G|$$

Equation 6.3.24 is a constraint that ensures the tour trip is connected, and ensures the connection time and visiting time are equal to $TT_{i,j}$ and ST_i.

6.4 Algorithm

The novelty of the GTTRM is based on the decision to split the group into subgroups based on user's preferences and constraints. An algorithm has been developed based on *Ant Colony Optimization* to make the splitting decision for a group of travelers where the travelers might have a conflict preferences and constraints. The proposed algorithm solves the amount of conflict among travelers in the group by making sub-routes for some of the group members. The main aim of the algorithm is to maximize the satisfaction level for each group member.

6.4.1 Group Ant Colony Optimization Overview

Group Ant Colony Optimization (GACO) is an algorithm developed to solve the GTTDP. Note that the main parameters that have been used in the GACO are shown in Table 6.7. Moreover, Table 6.7 show the initial values for the GACO parameters after we have conducted several experiments to determine the best parameter values under acceptable running time.

Table 6.7: List of parameters and initial values for GACO

Parameter	Initial Value	Description
α	0.5	The Alpha represents the importance of τ_{iju_z}
β	2	The Beta represents the importance of η_{iju_z}
γ	2	The Gamma represents the importance of λ_{iju_z}
η_{iju_z}		The Eta represents the rate of score to distance
τ_{iju_z}		The Tau represents the *Pheromones* level from i to j
$\lambda_{u_zu_o}$		The Lambda represents the *Social Relationship* between the users u and o
ρ	0.5	The Rho represents the value of pheromone evaporation
δ_{iju_z}		The Delta represents the maximum of total path scores i to j
Ant_No	20	Number of ants
Iterations	10	Number of iteration

Second, the key equations used in the GACO are shown below. The Eta is based on the rate-of-activity score for distance and the connection score for distance (see Equation (5.4.9)).

$$\eta_{iju_z} = \left(\frac{A_i^{u_z}}{TT_{ij}}\right) + \left(\frac{C_{ij}^{u_z}}{TT_{ij}}\right) \tag{6.4.25}$$

Equation (6.4.26) calculates the probability of each node for each user in the group.

$$P_{iju_z} = \frac{(\tau_{iju_z})^{\alpha}\,(\eta_{iju_z})^{\beta}\left(\sum_{v=1}^{|G|}\lambda_{iju_zu_o}\right)^{\gamma}}{\left(\sum_{i,j=1}^{|I|}\tau_{iju_z}\right)^{\alpha}\left(\sum_{i,j=1}^{|I|}\eta_{iju_z}\right)^{\beta}\left(\sum_{i,j=1}^{|I|}\sum_{o=1}^{|G|}\lambda_{iju_zu_o}\right)^{\gamma}} \tag{6.4.26}$$

Equation (6.4.27) and (6.4.28) show the local update pheromones.

$$\delta_{iju_z} = Max(\delta_{iju_z}, Ant_k(iju_z)) \tag{6.4.27}$$

$$\tau_{iju_z} = (1 - \rho) \times \tau_{iju_z} + \delta_{iju_z} \tag{6.4.28}$$

Equation (6.4.29) shows the global update pheromones.

$$\tau_{iju_z} = \rho \times \tau_{iju_z} + (1 - \rho) \times \delta_{iju_z} \tag{6.4.29}$$

Table 6.8 shows a list of functions and the main tasks of each that have been used in the GACO algorithm. The GACO algorithm is represented in Algorithms 1 to 4.

Table 6.8: List of functions in GACO and the main task for each

Function name	The main task
$initialization()$	Initializing the initialize values
$FindLastNodeTour(list)$	Find the last node in a tour
$FindCandidateNodes(list)$	Find a list of node which able to visit it under the constraints
$SelectedNode(list)$	Select a node from a list
$FindUsers(node)$	Find all users which are able to visit the node
$FOUVSN(node)$	Find other users visiting the same node
$FOUTSNTAN(node)$	Find other users traveling from same node to another node
$FindAllNodeAvailable(time)$	Find all nodes which are available at the time

Algorithm 1 represents the pseudo code of the main algorithm. The algorithm starts with the initialization function where the initial values for all parameters are allocated. Afterwards, the loops of $TripLength$, $Iterations$ and $AntNo$ are embedded into the main function, and the $Route(AntK)$ function is illustrated in Algorithm 2. In addition, the $ScoreCalculate()$ function is shown in Algorithm 3.

Algorithm 1: An overview of GACO

```
initialization();
while day < TripLength do
    while i < Iterations do
        while AntK < AntNo do
            MultiDaysRoute= Route(AntK);
            while user < GroupSize do
                CurrentScore(user) = ScoreCalculate(MultiDaysRoute(user));
                if CurrentScore(user) > BestScore(user) then
                    BestScore(user) = CurrentScore(user);
                end
            end
            LocalUpdatePheromones();
        end
        GlobalUpdatePheromones();
    end
end
```

Algorithm 2: An overview of Route function

```
while true do
    LastNodeRoutes = FindLastNodeTour();
    while user < GroupSize do
        FinishGroupTour * = FinisthTour(user);
    end
    if FinishGroupTour then
        break;
    end
    while user < GroupSize do
        Probability(user) = CalculateProbability(user);
        CandidateNodesList(user) = FindCandidateNodes(user);
    end
    SelectNodeUser[GroupSize] = 0;
    while user < GroupSize do
        SelectNodeUser[user] = SelectedNode(CandidateNodesList(user));
        if CandidateNodesList.Users(SelectNodeUser[user]) > 1 then
            AllUsersInCandidateNodesList =
             FindUsers(CandidateNodesList(SelectNodeUser[user]));
            SelectNodeUser[AllUsersInCandidateNodesList] = 1;
        end
    end
    while user < GroupSize do
        if sum(Probability(user)) == 0 then
            Routes(user) = [Routes(user),EndNode];
            FinisthTour(user) = 1;
        else
            Routes(user) = [Routes(user),SelectNodeUser(user)];
        end
    end
end
```

Algorithm 3: An overview of ScoreCalculate function

```
while user < GroupSize do
    Routes(user) = MultiDaysRoute(user);
    while index < length(Routes(user)) -1 do
        ActivityScores(user) += Score(Routes(user)(index)) *
         VisitingTime(Routes(user)(index));
        ActivityScores(user) += FOUVSN(Routes(user)(index));
        ConnectionsScores(user) +=
         Connections(Routes(user)(index),Routes(user)(index + 1)) *
         Distance(Routes(user)(index),Routes(user)(index + 1));
        ConnectionsScores(user) += FOUTSNTAN(Routes(user)(index));
    end
    WaitingScores(user) += TripTotalTime - T_max * WaitingTimeWeight;
end
```

Algorithm 4: An overview of Calculate Probability function

```
while user < GroupSize do
    LastNodeRoutes = FindLastNodeTour(user);
    AccessibleNodes(user) = FindAllNodeAvailable(time);
    while i < length(AccessibleNodes(user)) do
        Probability.Node(user,i) = AccessibleNodes(user)(i);
        Probability.User(user,i) = user;
        Probability.Percentage(user,i) =
         power(Eta(LastNodeRoutes,i,user),Alpha) *
         power(Tau(LastNodeRoutes,i,user),Beta) *
         power(SocialRelationship(user,user),Gamma);
    end
end
```

6.5 Experiments

We have conducted two experiments based on aggregation methods (see Section 6.3.2). The first experiment is based on aggregating all group members into one profile using the *Average* aggregation method. The second experiment is based on the GACO algorithm (see Section 6.4).

6.5.1 Benchmark instances

Because no dataset is available for the GRSs in travel applications, a real-world dataset has been collected. The dataset *Durham, UK* shows the geographical location where the data were collected (see Appendix A for more details).

Table 6.9 describes the different scenarios and group sizes. In addition, Tables 6.10 to 6.14 show the social relationship values among the group members. The social relationship values have been chosen randomly from 1 to 2, where 1 represents the weakest relationship and 2 represents the strongest relationship.

Table 6.9: The groups' members description

No	Group members		Gender No		Relationship
	Adult No	Children No	Male	Female	
1	2	3	2	3	Family
2	6	0	4	2	Colleagues
3	2	0	1	1	Young couple
4	4	0	2	2	Retired friends
5	15	0	6	9	Students

6.6 Results

This section compares the results of the two aggregations methods and found the GACO algorithm outperforms the other methods by maximising group members satisfaction levels.

Table 6.10: The relationship value in the first group

	U_1	U_2	U_3	U_4	U_5
U_1	1	1.5	2	2	2
U_2	1.5	1	2	2	2
U_3	2	2	1	2	2
U_4	2	2	2	1	2
U_5	2	2	2	2	1

Table 6.11: The relationship value in the second group

	U_1	U_2	U_3	U_4	U_5	U_6
U_1	1.0	1.2	1.2	1.8	1.1	1.2
U_2	2.0	1.0	1.6	1.5	1.5	1.4
U_3	1.7	1.5	1.0	1.4	1.8	1.6
U_4	1.9	1.6	1.2	1.0	1.6	1.2
U_5	1.6	1.9	2.0	1.9	1.0	1.4
U_6	1.6	1.3	1.5	1.8	1.6	1.0

Table 6.12: The relationship value in the third group

	U_1	U_2
U_1	1.0	2.0
U_2	2.0	1.0

Table 6.13: The relationship value in the fourth group

	U_1	U_2	U_3	U_4
U_1	1.0	2.0	2.0	1.0
U_2	1.8	1.0	1.4	1.9
U_3	1.4	1.0	1.0	1.3
U_4	1.6	1.4	1.5	1.0

6.6.1 Group Aggregation

This section shows the results of the experiments on *Group Aggregation* where group members are aggregated into one profile. We have presented the *Happiness Function*

Table 6.14: The relationship value in the fifth group

	U_1	U_2	U_3	U_4	U_5	U_6	U_7	U_8	U_9	U_{10}	U_{11}	U_{12}	U_{13}	U_{14}	U_{15}
U_1	1.0	1.4	1.1	1.3	1.5	2.0	1.1	1.1	1.8	1.7	1.5	1.1	1.8	2.0	1.1
U_2	1.3	1.0	1.9	1.7	1.3	1.9	1.9	2.0	1.3	1.4	1.4	1.2	1.3	1.2	1.8
U_3	1.6	1.7	1.0	1.1	1.5	2.0	1.4	1.9	1.8	1.5	1.0	1.2	1.1	1.8	1.3
U_4	1.9	1.7	1.0	1.0	1.8	1.2	2.0	1.9	1.6	1.1	1.0	1.8	1.3	1.8	1.6
U_5	1.9	1.3	2.0	1.7	1.0	1.3	1.5	1.2	1.2	1.5	1.3	1.7	1.3	1.2	1.6
U_6	1.3	1.9	1.0	1.6	1.3	1.0	1.6	1.7	1.4	1.7	1.0	1.8	1.7	1.2	1.5
U_7	1.1	1.1	1.8	1.0	1.5	1.7	1.0	1.9	1.6	1.8	1.2	2.0	1.9	1.8	1.7
U_8	1.7	1.0	1.4	1.7	1.6	1.4	1.4	1.0	1.4	1.7	1.6	1.9	1.3	1.6	1.1
U_9	2.0	1.0	1.8	1.5	1.8	1.5	1.1	1.1	1.0	1.4	1.1	1.8	1.6	2.0	1.5
U_{10}	1.6	1.7	1.9	1.3	1.4	1.1	1.4	1.2	1.4	1.0	1.4	1.5	1.2	1.3	1.5
U_{11}	1.9	1.3	1.0	1.9	2.0	1.5	1.6	1.8	1.5	1.7	1.0	1.5	1.8	1.4	1.5
U_{12}	1.4	1.4	1.9	1.1	1.8	1.7	1.9	1.1	1.2	1.2	1.2	1.0	1.5	1.2	1.9
U_{13}	1.6	1.6	1.8	1.9	1.4	1.8	1.1	1.5	1.3	1.2	1.2	1.3	1.0	1.9	1.7
U_{14}	1.7	1.9	2.0	2.0	1.8	1.7	1.8	1.2	1.6	1.1	1.5	1.9	1.1	1.0	1.8
U_{15}	1.9	1.9	1.2	1.3	1.9	1.4	1.9	1.4	1.7	1.7	1.3	1.4	1.3	1.9	1.0

for each user in the groups.

Figures 6.3 to 6.7 show the satisfaction level for each user from the beginning of the trip until the end; it is clear that the group members have varying satisfaction levels.

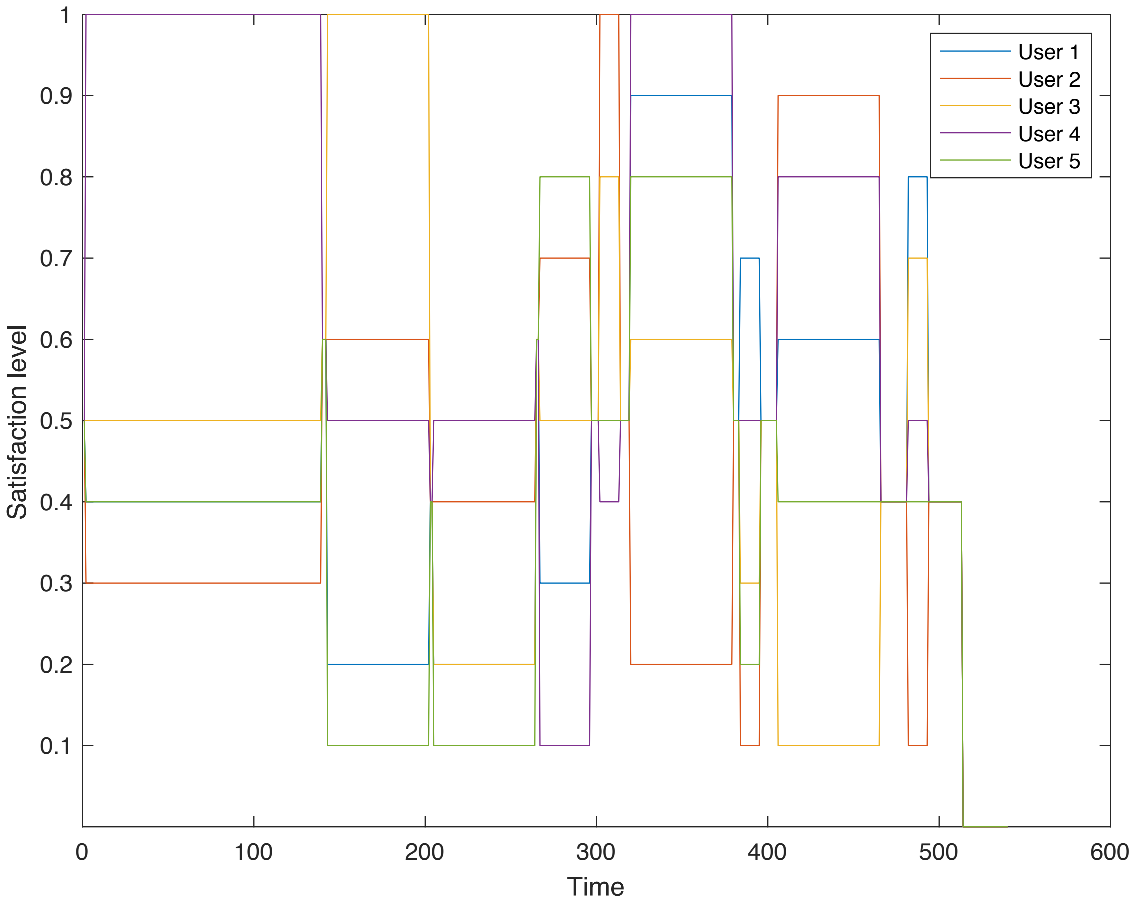


Figure 6.3: Overview of the first group for all members based on group aggregation

Figure 6.3 shows the timeline for all users in the first group. Each line represents each users satisfaction level in the group where the highest level of satisfaction is 1. Users satisfaction levels in the first group differ based on their different preferences.

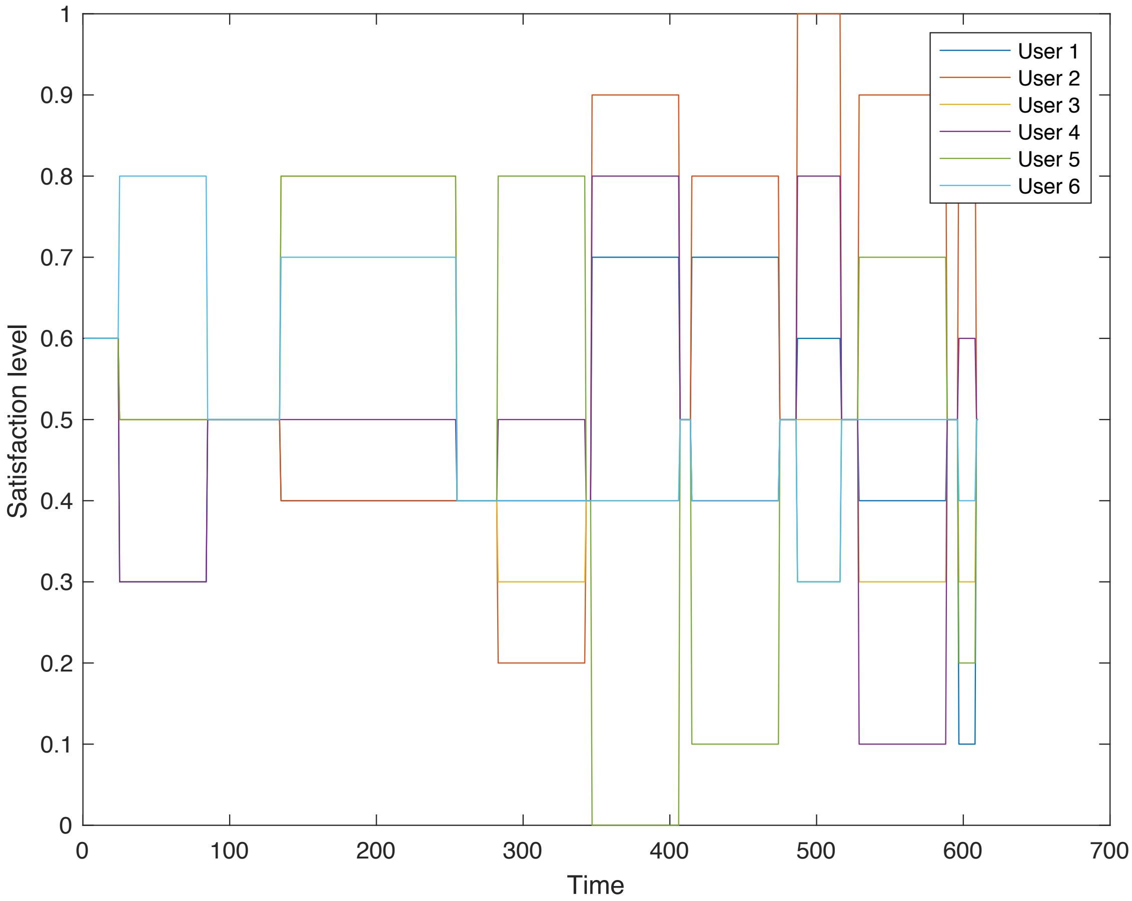


Figure 6.4: Overview of the second group for all members based on group aggregation

Figure 6.4 shows the total satisfaction level for all members of the second group. Most of the users do not achieve the highest level of satisfaction because the aggregation method aggregates all users values into a value that may cause other users to be unsatisfied.

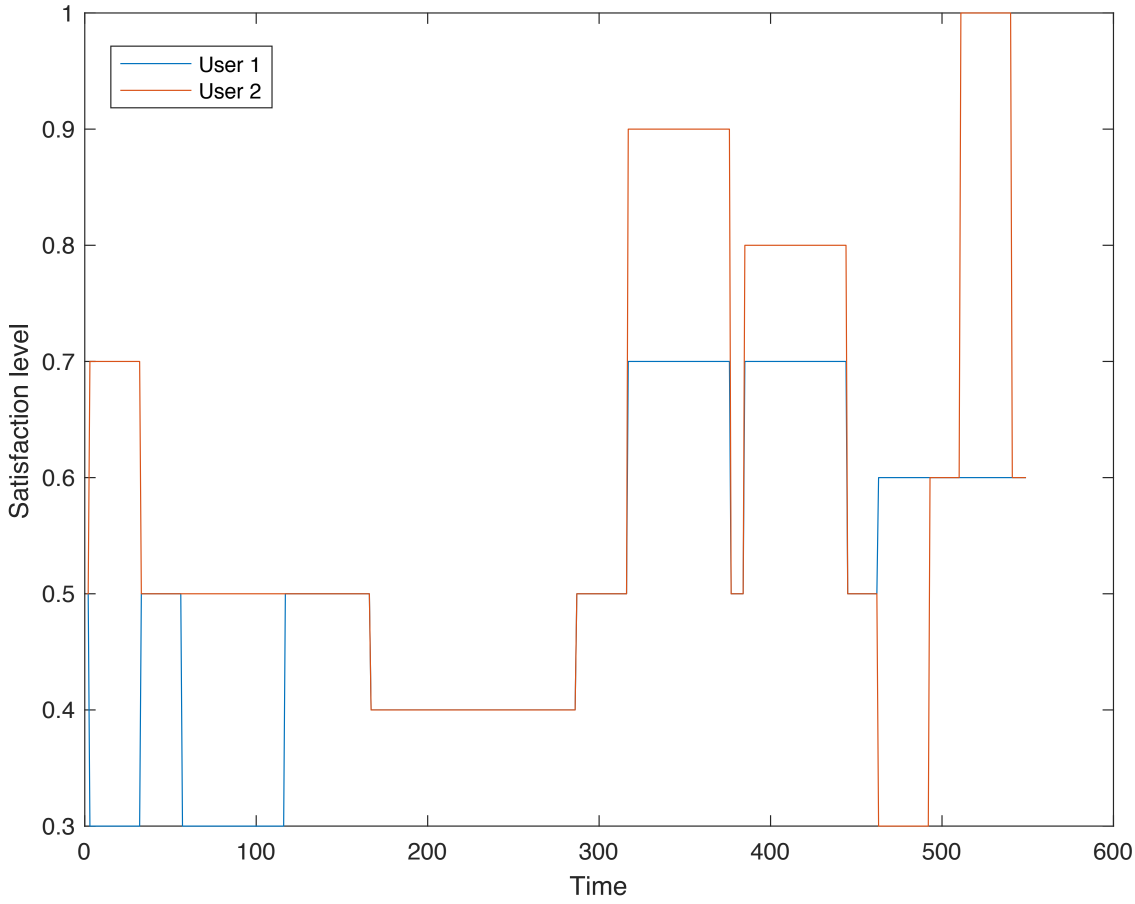


Figure 6.5: Overview of the third group for all members based on group aggregation

Figure 6.5 shows the satisfaction level for user 1 and user 2 in the third group where the satisfaction level between the users varies from the beginning of the trip until 150 minutes into the trip (the trip length is divided into minutes). The satisfaction level for both of the users in the last two-thirds of the trip time is much better than in the first third.

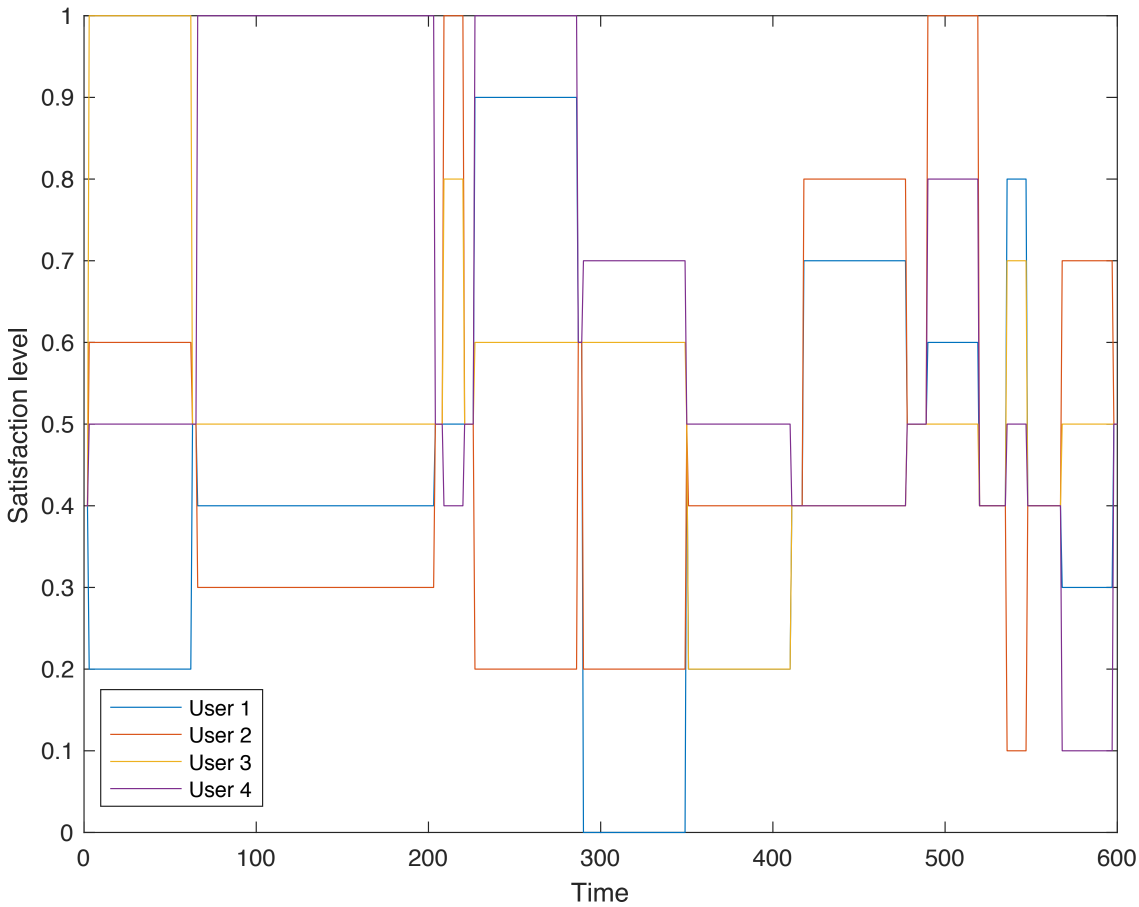


Figure 6.6: Overview of the fourth group for all members based on group aggregation

Figure 6.6 shows the variant preferences of different users in the fourth group where the group aggregation method provides a recommended tour that fails to satisfy all of the users.

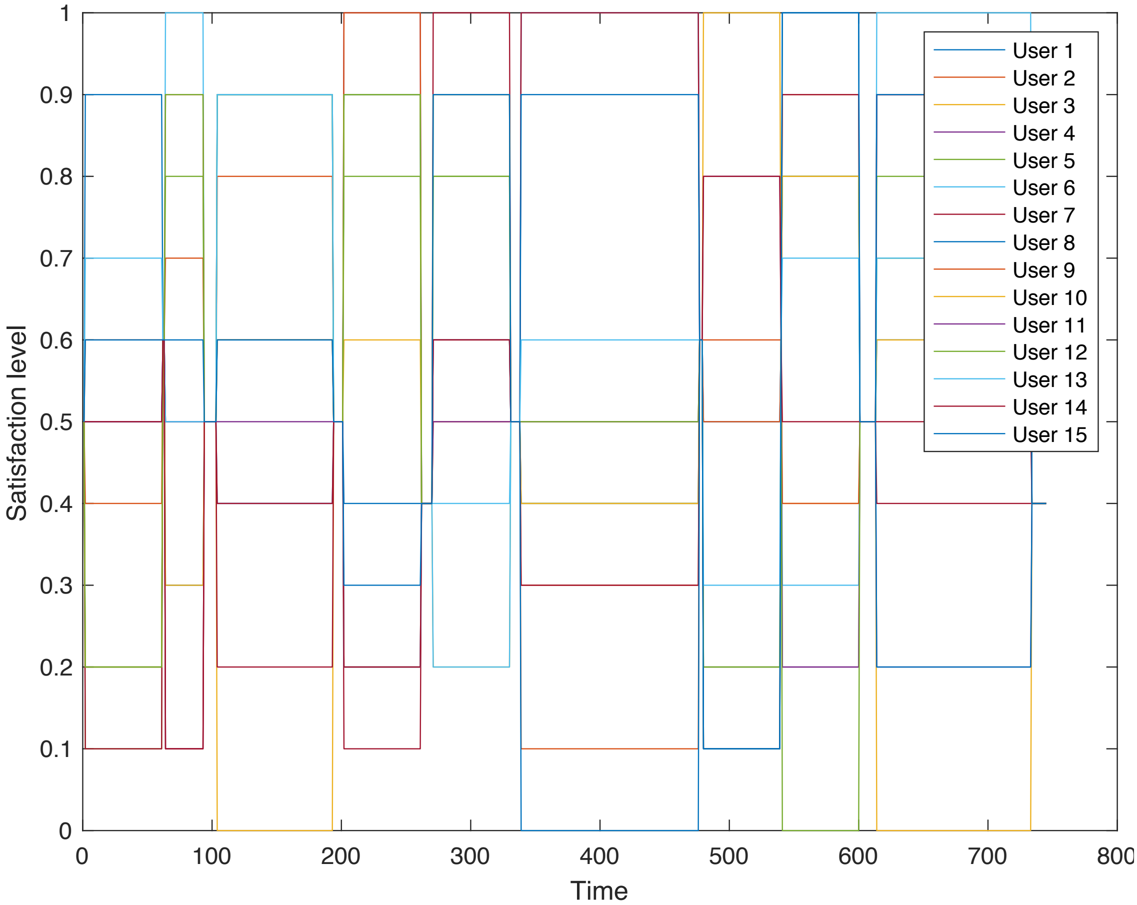


Figure 6.7: Overview of the fifth group for all members based on group aggregation

Figure 6.7 shows the huge gap between users' preferences because they remain together at all times on the trip as there is no consideration of their different preferences.

6.6.2 User Aggregation

This section presents the results of applying the GACO algorithm into the GTTRM.

First Group

We compare the satisfaction levels for each user related to the user aggregation and the group aggregation; we found that, in general, the user aggregation based on the GACO algorithm provides better results compared to the group aggregation. Figures 6.8 to 6.12 show the satisfaction level for the first groups members.

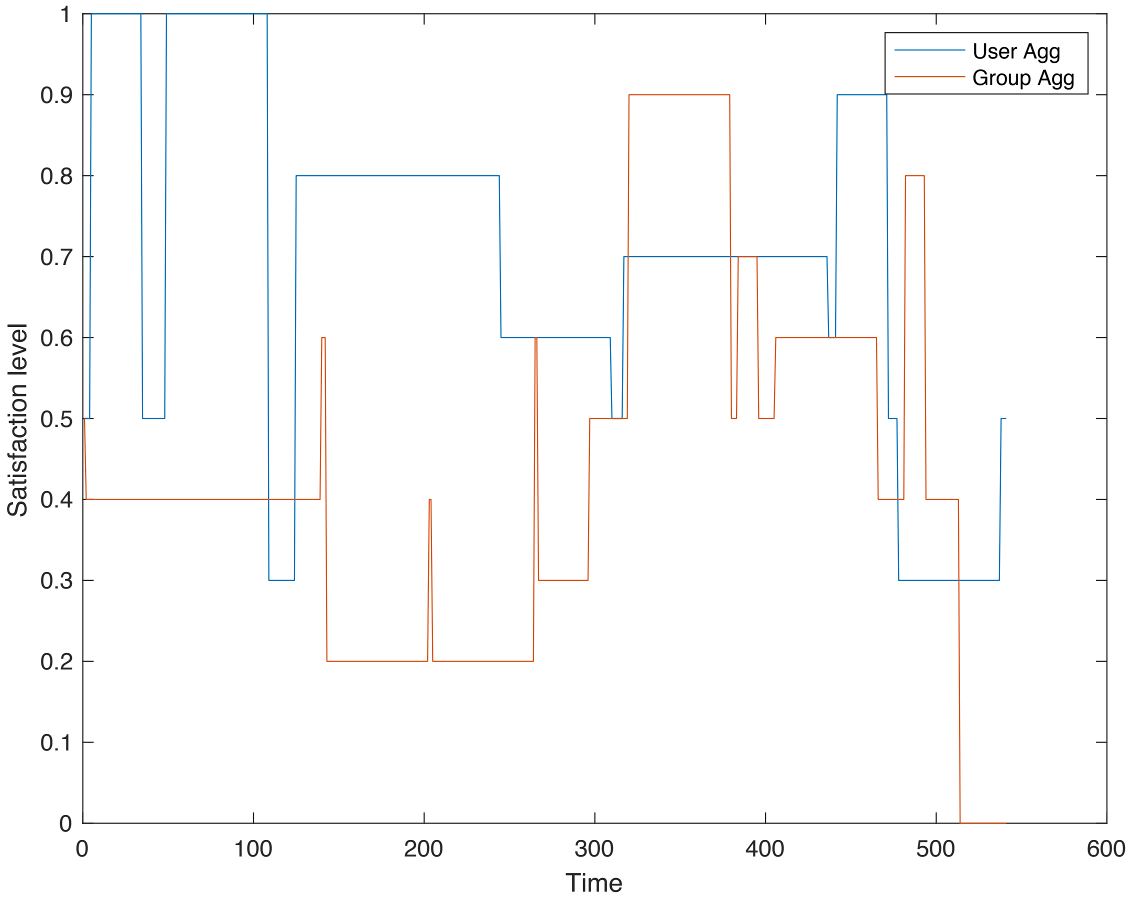


Figure 6.8: Comparison of the happiness function for user 1 between the group aggregation and user aggregation

Figure 6.8 shows user 1's satisfaction level in the first group in terms of different methods where the user aggregation method provides a higher satisfaction level compared to the other method. In addition, the figure shows the models work differently at the same time because the first method (group aggregation) takes the decision based on group profile where the second method (user aggregation) is based on individual user for the decision.

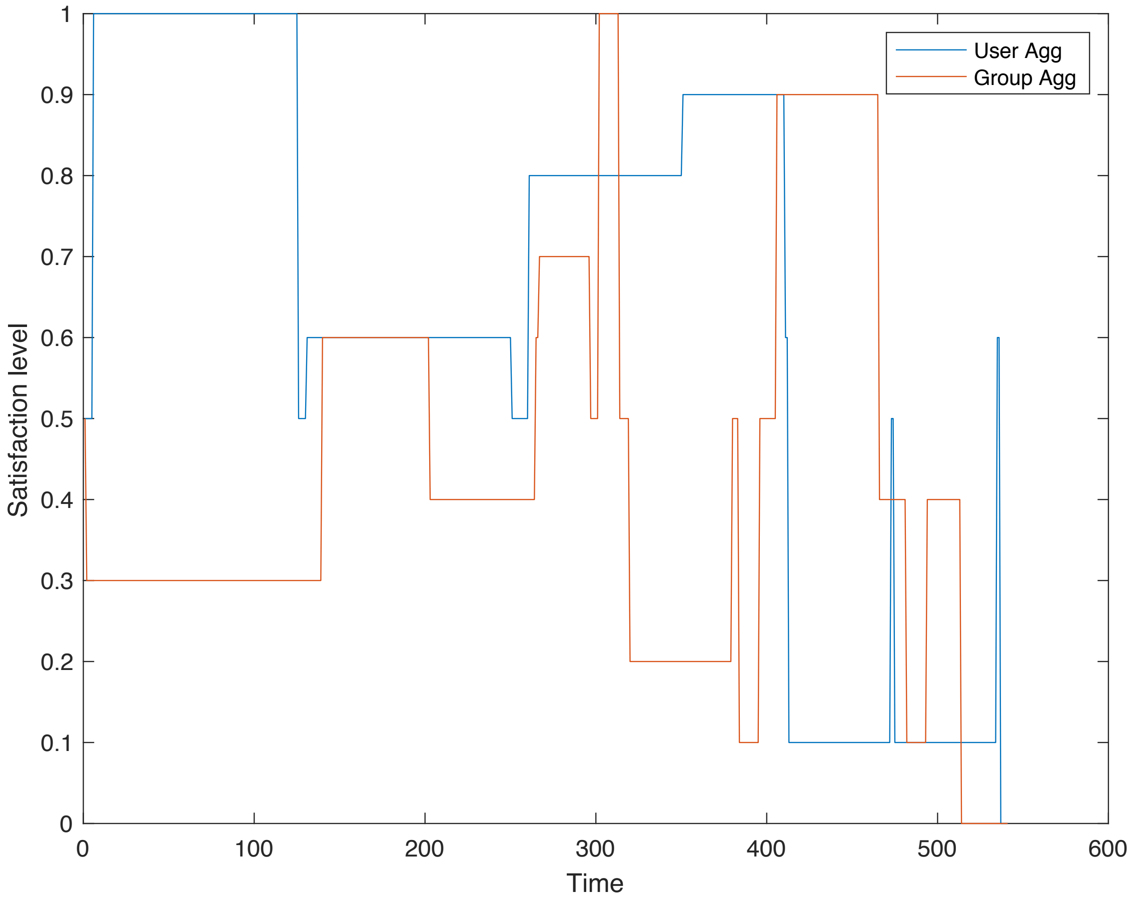


Figure 6.9: Comparison of the happiness function for user 2 between the group aggregation and user aggregation

Figure 6.9 shows user 2's satisfaction level in the first group where the satisfaction level for the user is also higher than the group aggregation method. The performance of user aggregation is better than the other method, however, at a specific time, the group aggregation (for short time) produce a better result. The main reason the group aggregation sometimes has good results is that the GACO is designed to choose only one POI after each other where no consideration whole route together.

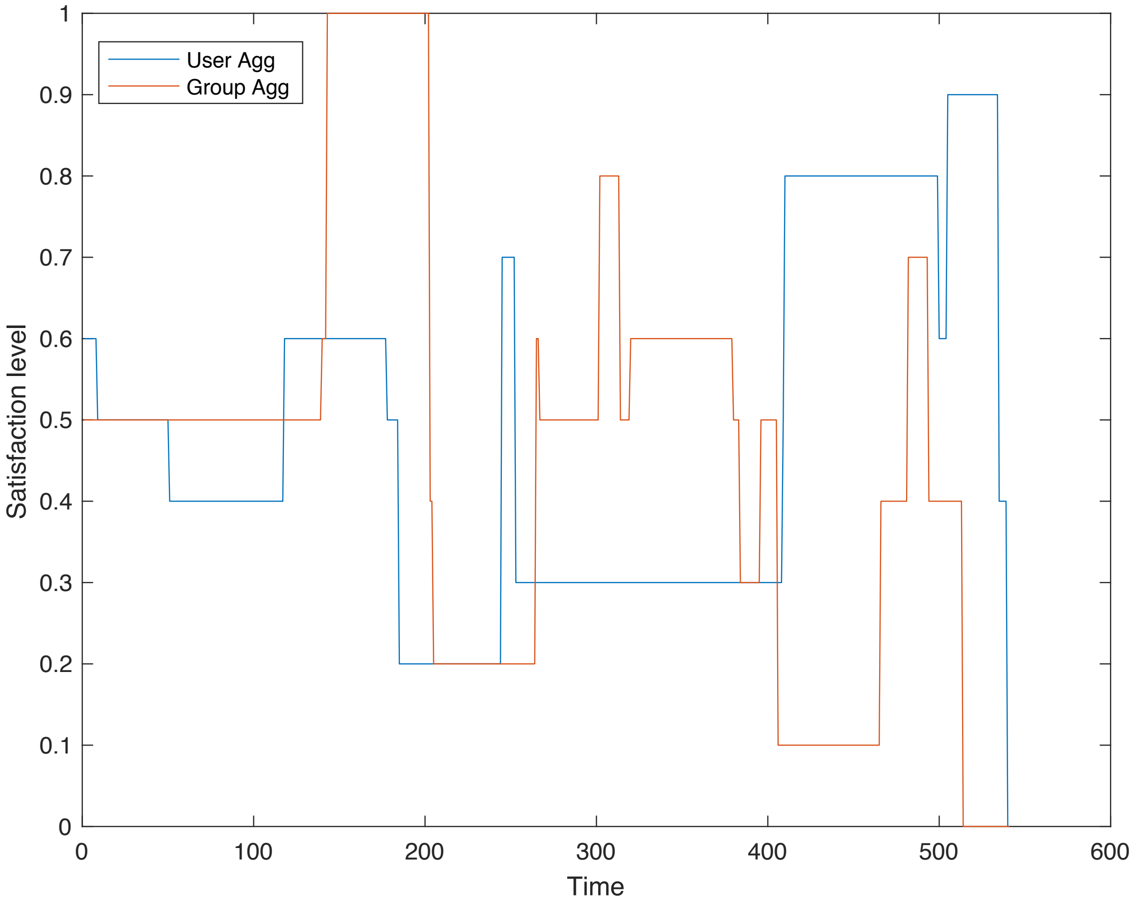


Figure 6.10: Comparison of the happiness function for user 3 between the group aggregation and user aggregation

Figure 6.10 shows user 3's satisfaction level in the first group where both methods produce similar results. Also, the figure shows at specific times the group aggregation perform better than user aggregation because sometimes the user aggregation's algorithms choose a POI after that the algorithms could not find another good POI to reach it, so the result is the low satisfaction POI is chosen.

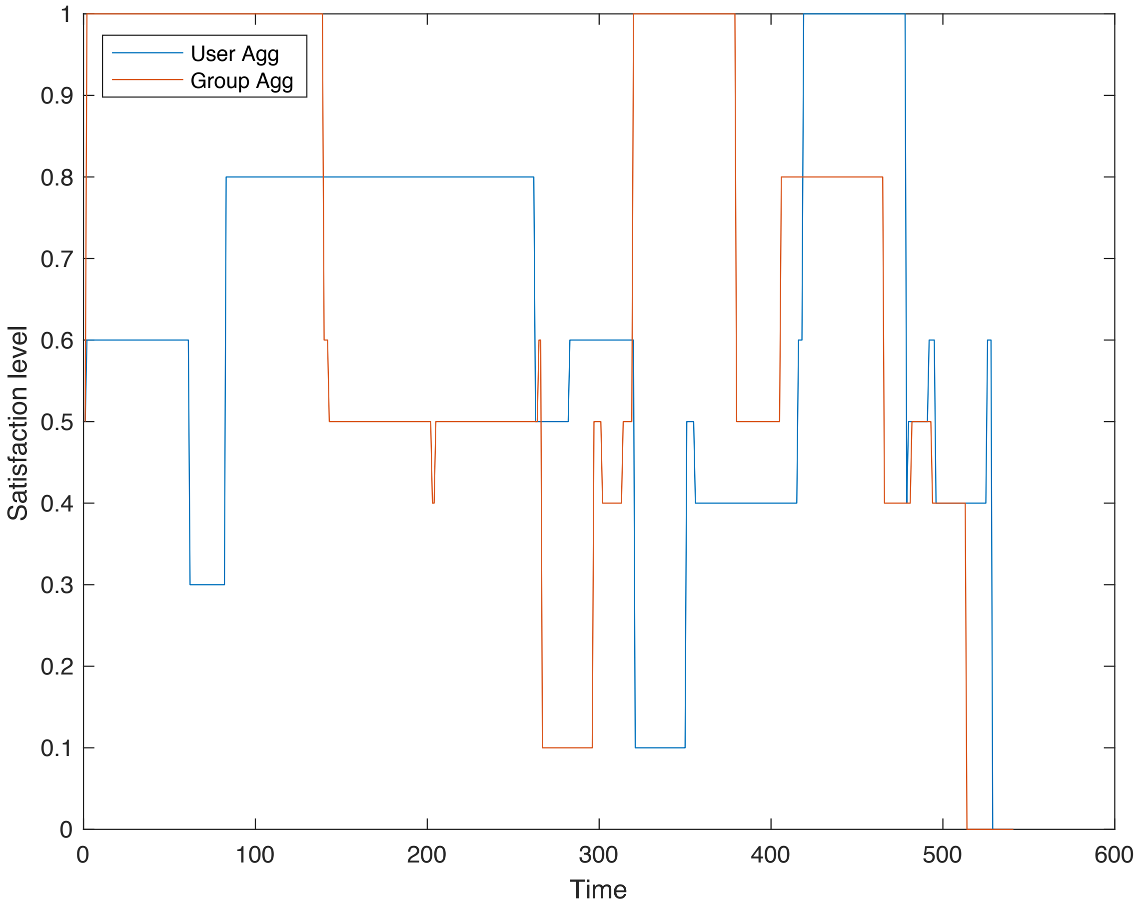


Figure 6.11: Comparison of the happiness function for user 4 between the group aggregation and user aggregation

Figure 6.11 provides a comparison between the two methods for user 4's satisfaction level where both methods produce similar results. In the beginning, the group method performs better than the user method, and the main reason because the User 4 has a good relationship with one of the group member where the GACO has chosen the User 4 to go with.

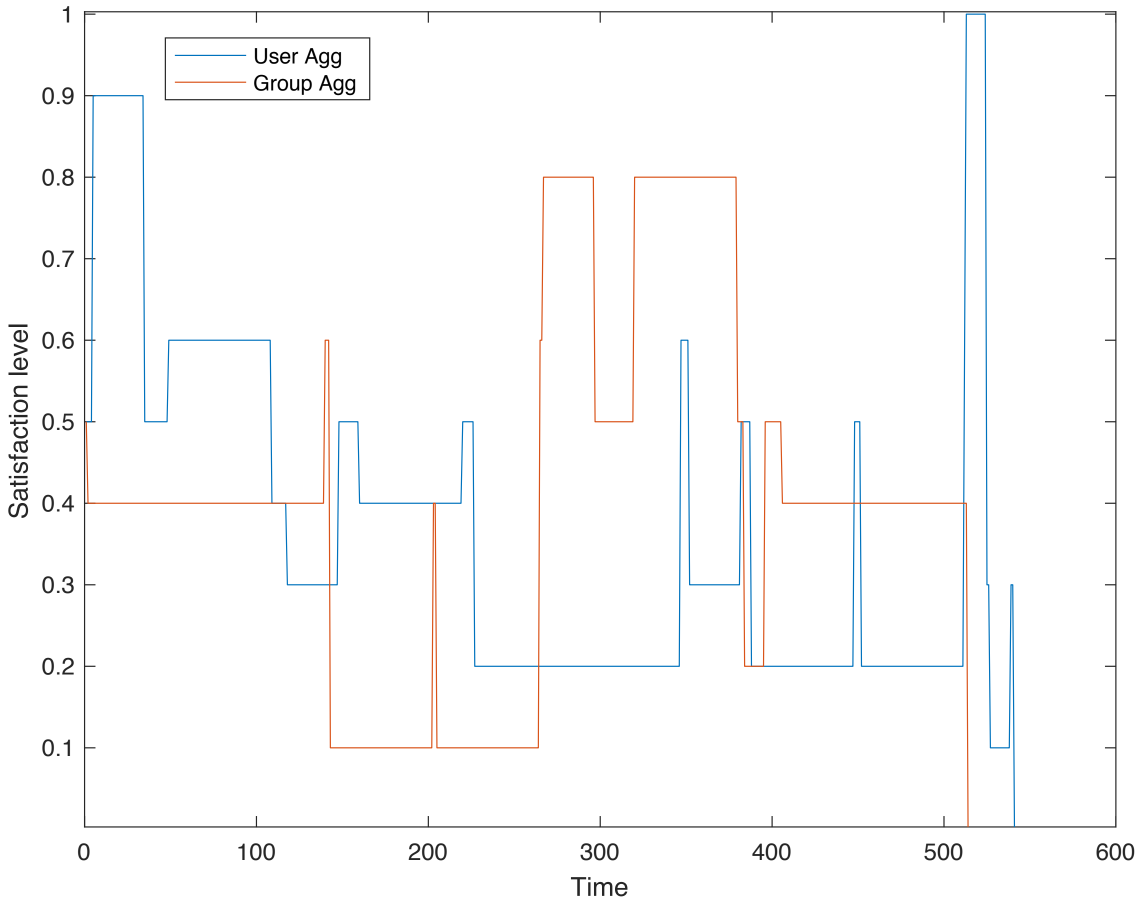


Figure 6.12: Comparison of the happiness function for user 5 between group aggregation and user aggregation

Figure 6.12 shows the satisfaction level for user 5 in the first group where sometimes the user aggregation method produces better results and sometimes the group aggregation method produces better results. As we mentioned in User 3 result, the GACO may choose a POI wherein a side of the city all the POIs around have low satisfaction.

Second Group

Secondly, we compared the satisfaction level of the second group's members with the group aggregation. Figures 6.13 to 6.18 show a comparison between the user aggregation and the group aggregation based on the satisfaction level.

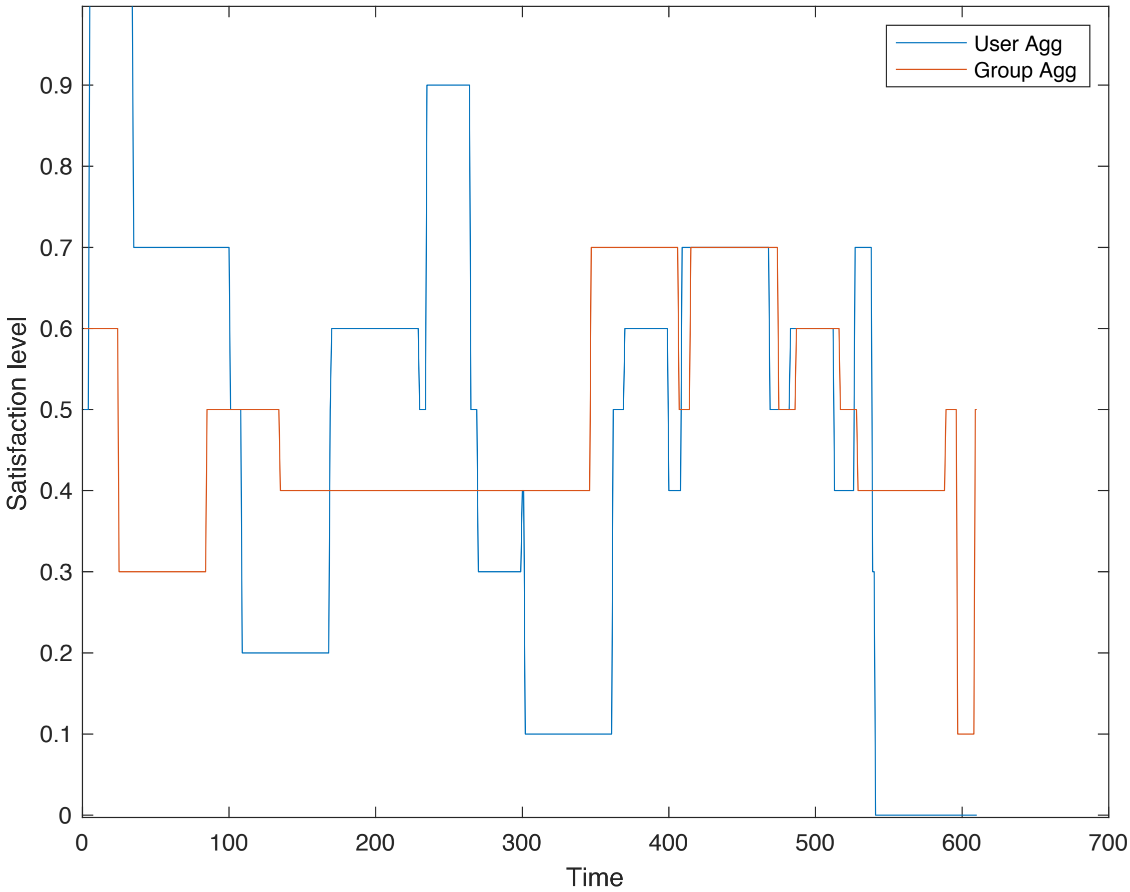


Figure 6.13: Comparison of the happiness function for user 1 between group aggregation and user aggregation

Figure 6.13 shows the differences between group aggregation and user aggregation where user aggregation produces better results while group aggregation produces a long tour trip. It is shown that the user aggregation end the trip before the group aggregation, but in general, the user aggregation provide higher satisfaction trip in total.

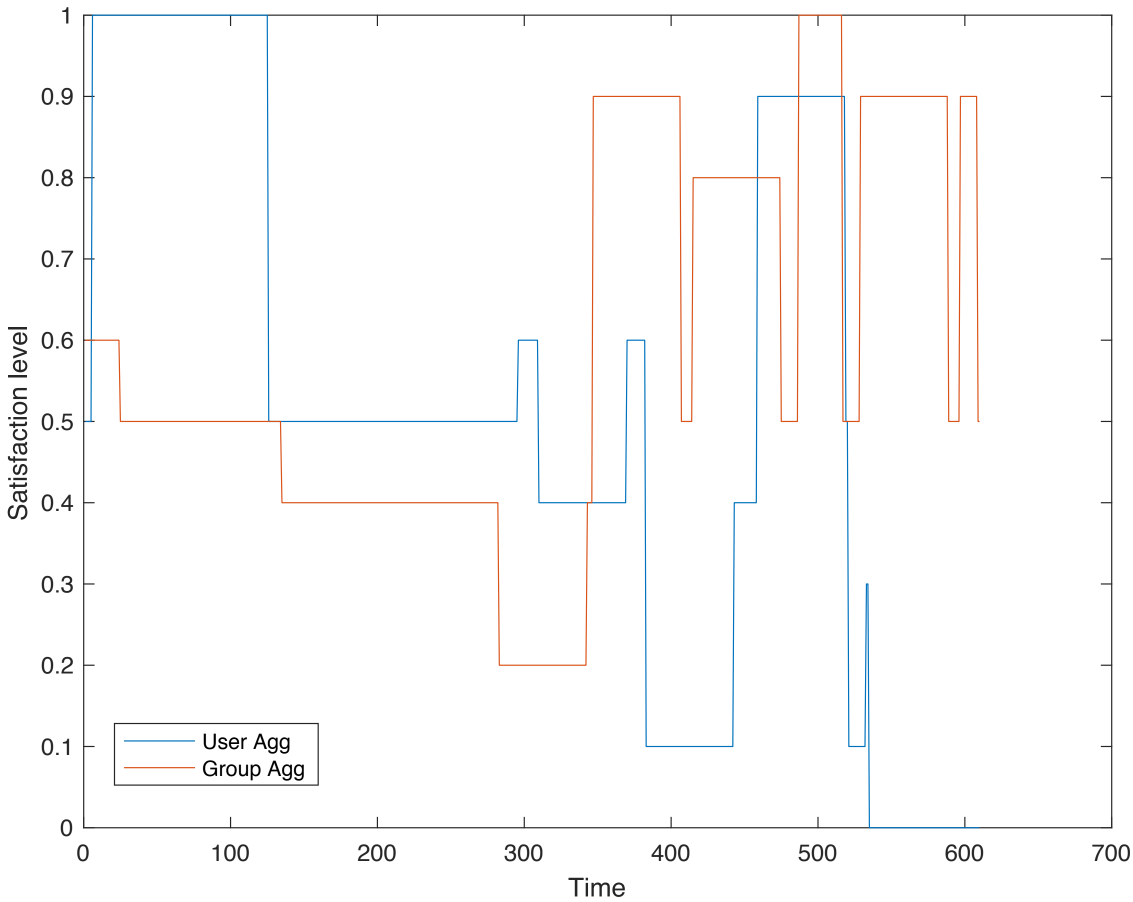


Figure 6.14: Comparison of the happiness function for user 2 between group aggregation and user aggregation

Figure 6.14 shows a comparison between the results of the aggregation methods. The user aggregation method produces a better overall result while the group aggregation produces a long tour trip. The GACO is limited to choose series (together at the same time) of POIs which will provide stable satisfaction level, and the main reason for the result is that the GACO choose POI as an individual.

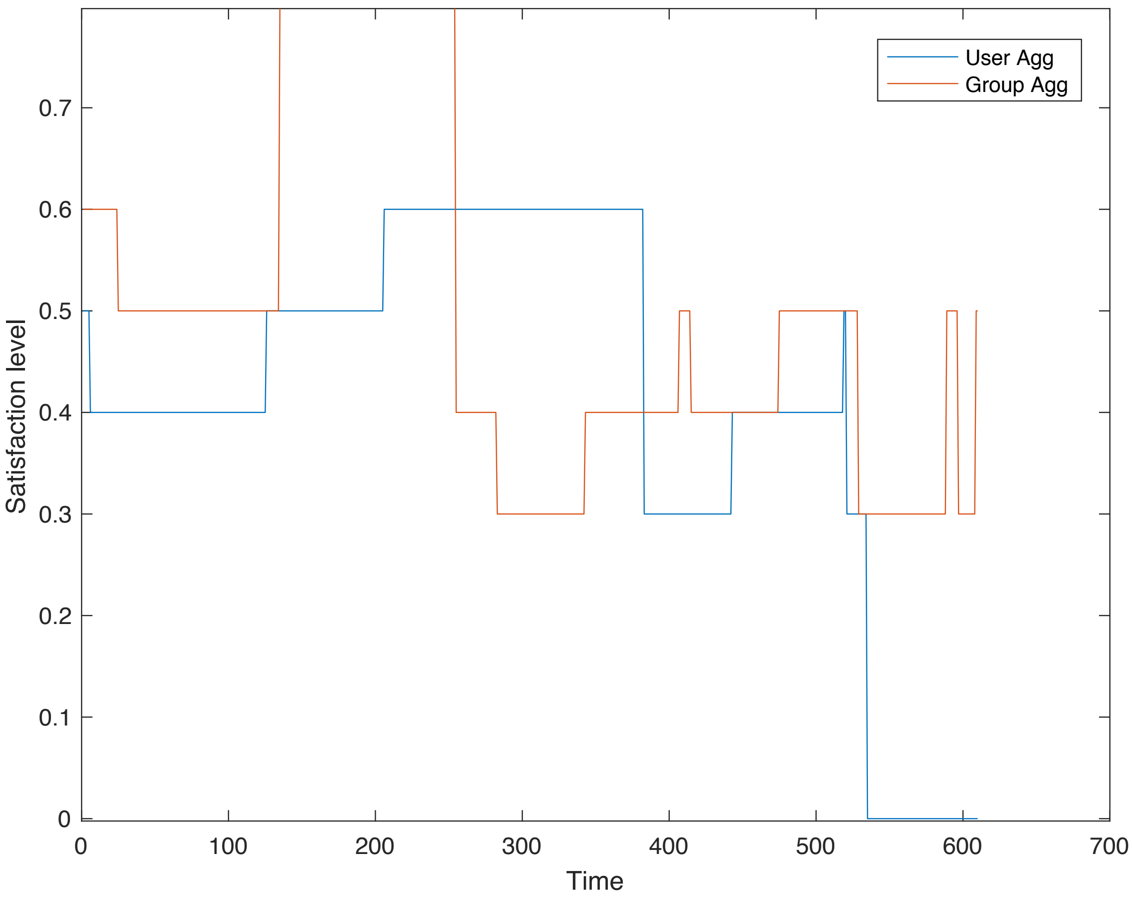


Figure 6.15: Comparison of the happiness function for user 3 between group aggregation and user aggregation

Figure 6.15 shows that the user aggregation method produces a recommended tour that is shorter than the tour based on group aggregation. As it is shown in the picture, group aggregation provides a better result for the user 3, and the main reason is that the GACO has chosen the User 3 to go with another user (the relationship value is high between them) without considering User 3 preference.

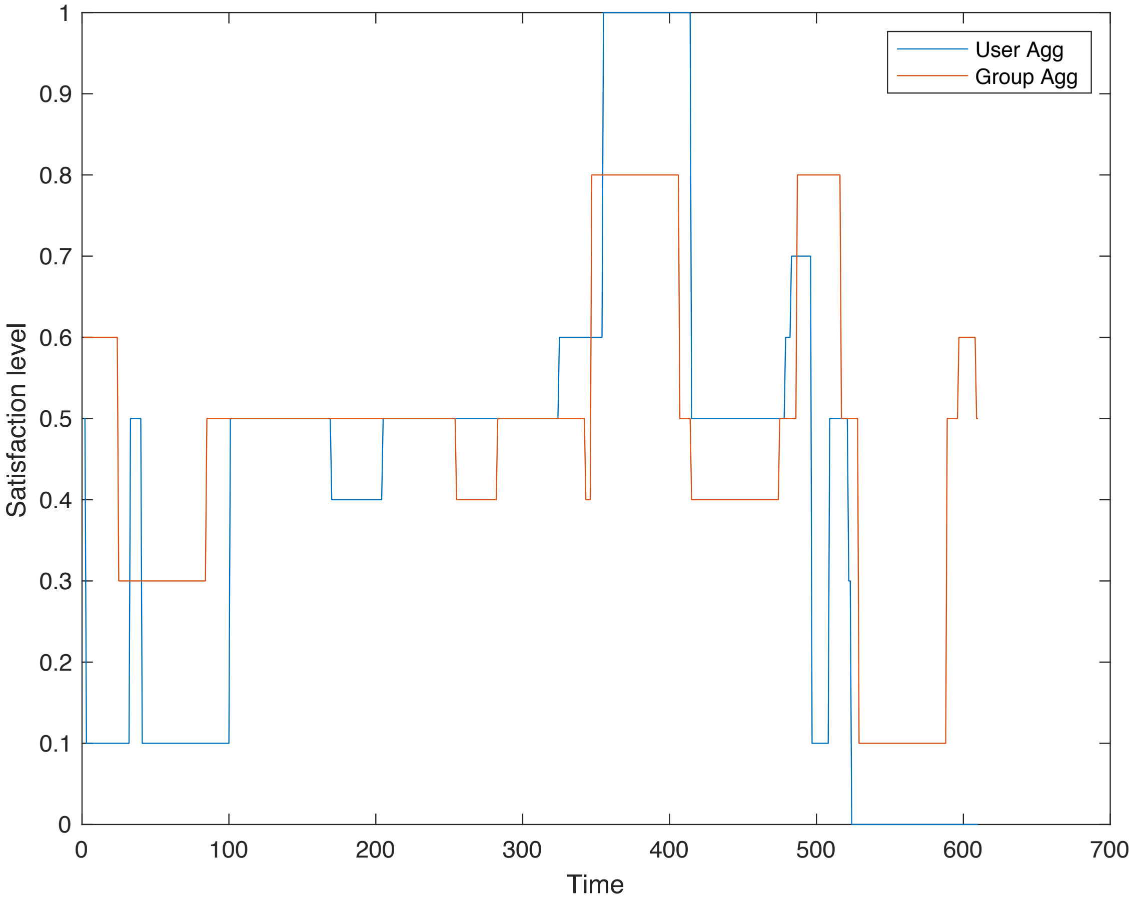


Figure 6.16: Comparison of the happiness function for user 4 between group aggregation and user aggregation

Figure 6.16 shows that the group aggregation and user aggregation methods produce similar recommended tours in terms of the level of satisfaction. Because the GACO has chosen the User 4 to go with another user in the group (for a part of the trip), the satisfaction level was lower than the group aggregation in total.

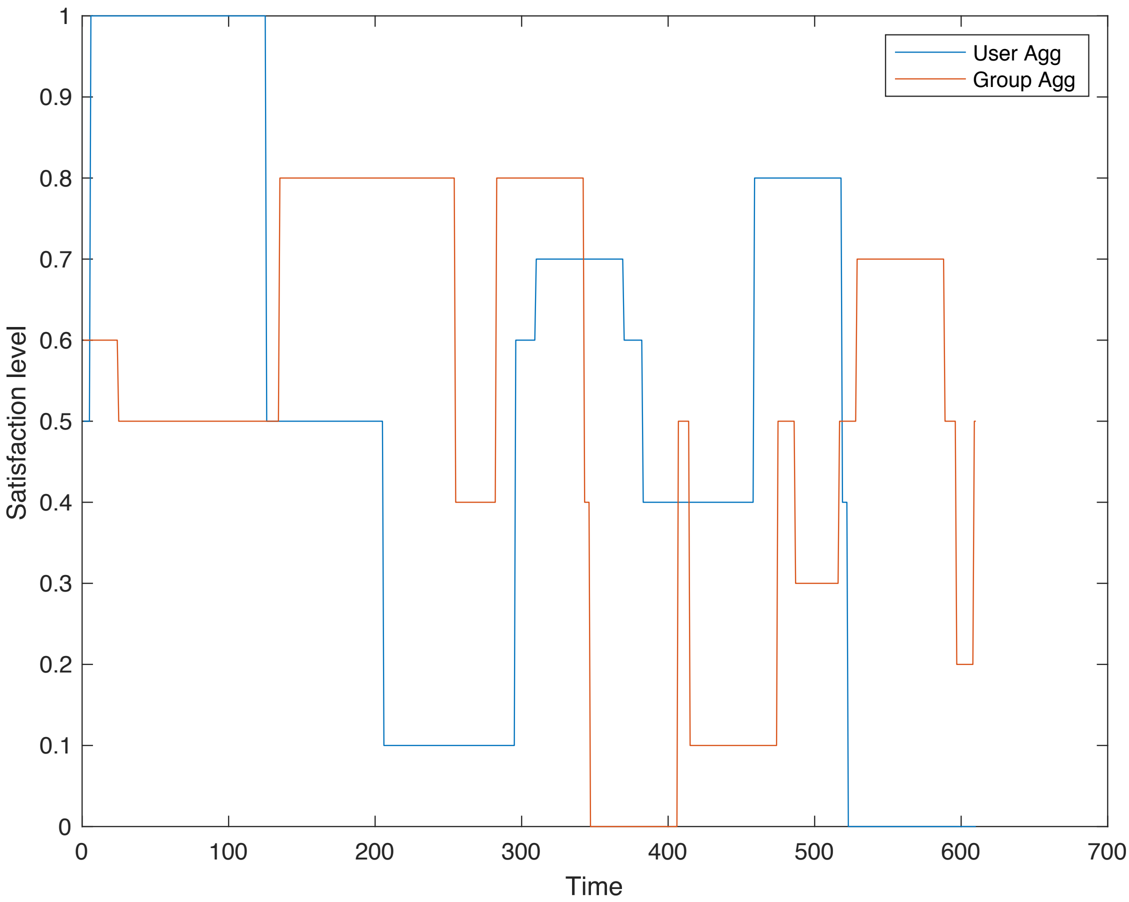


Figure 6.17: Comparison of the happiness function for user 5 between group aggregation and user aggregation

In addition, Figure 6.17 highlights that the results of group aggregation and user aggregation generate a high level of satisfaction for the user 5 in the second group. When GACO chooses a POI in a direction, the next POI might be not satisfied with the user because the number of option available is a few in this direction. We can see the first part of the trip in user aggregation is very high, then GACO could not find another good POI to keep the satisfaction level at the same.

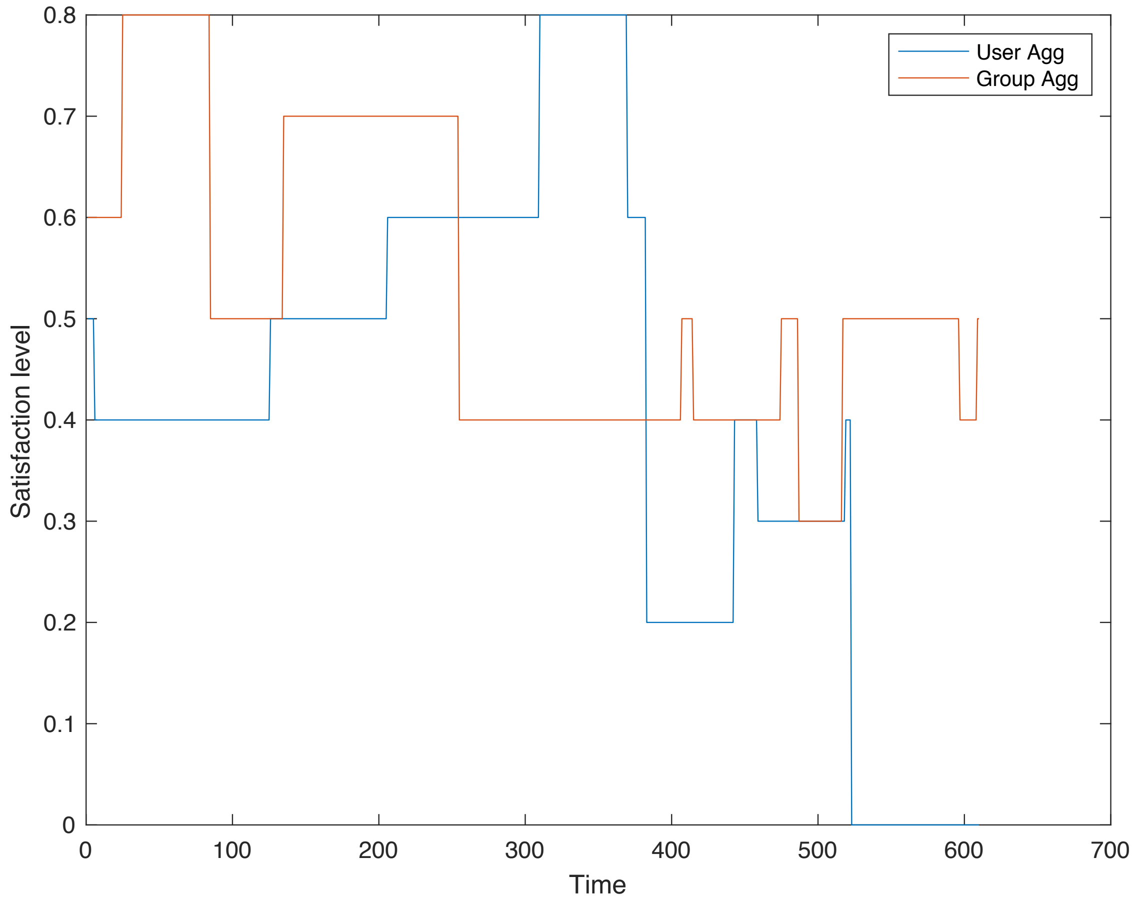


Figure 6.18: Comparison of the happiness function for user 6 between group aggregation and user aggregation

Figure 6.18 shows the group aggregation method produces better results at the beginning of the trip, while the user aggregation method provides better results at the midpoint of the trip. As User 6 have been chosen to go with another user in the same group the satisfaction level in user aggregation has no stable.

6.6.3 Discussion

We have conducted several experiments on the two aggregation methods. The user aggregations results outperform the group aggregations results. For example, Figure 6.8 compares the performance of the two aggregation methods; the user aggregation method outperforms the group aggregation method by maximising the user's happiness function. In addition, the first and second users of the groups always had higher satisfaction levels compared with other users in the group.

In summary, we have demonstrated that our GACO algorithm can produce better results (i.e. a higher satisfaction level for travellers) compared to the group aggregation method for the first and second users in the group. The main reason our algorithm always produces better results for the first and second users in the group is because of the social relationship values. These values drive the algorithm to make decisions to satisfy the first user who is travelling with a second user who may not be fully satisfied by visiting the POIs that the first user prefers.

6.7 Conclusion

In summary, this chapter has addressed the problem of *Group Travelers* in building a personalized tour trip. We have introduced the GTTDP in the context of a group of travelers planning to travel together. In addition, the GACO algorithm has been developed to maximize the satisfaction level of each group member by recommending sub-routes for a subgroup of the main group to reduce potential conflicts. The results of the GACO show that, in general, the GACO is able to improve the satisfaction level of the individual group members.

In this chapter, we have proposed a novel algorithm that has been designed and developed to make a significant contribution to the field of Recommender Systems. First, we introduced the GTTDP in the context of solving the problem of building a tour trip for a group of travellers, and second, we have described the design of the GTTRM, which is a mathematical model to solve the problem of the GTTDP. Finally, we have developed the GACO algorithm to maximize the level of satisfaction for each user in a group.

Chapter 7

Generalized Multi-Objective Orienteering Problem

This chapter describes the Generalized Multi-Objective Orienteering Problem (GMOOP), which a generalized model using a linear, single-valued objective function with user-defined weights to produce solutions for the Multi-Objective OP (MOOP). That is, we use the GMOOP to solve the multi-objective problem by substituting the multi-valued objective function with a weighted average of its components, and allowing travellers to assign weights to each component.

Our tests show that despite the proposed GMOOP approachs simplicity, on some long journeys it yields results that are Pareto superior to the current state-of-the-art, demonstrating the challenges still facing the more complex MOOP approach.

7.1 Introduction

Many real-world problem optimizations suffer from multiple conflicting objectives, and, because of the lack of suitable solutions, this transforms multi-objective optimization into a single-objective problem [24]. Specifically, multi-objective problems produce many Pareto-optimal solutions where any two solutions represent a trade-off between the specified objectives. However, the main objective of *Recommender Systems* (RSs) is to personalize the vast number of options where the multi-objective optimization provides more than a single solution. In other words, the multi-objective approach usually produces more than one Pareto-optimal solution, and deciding which solution is the most recommended option is difficult.

Patently, the matter of building a tour trip represents a problem based on multi-objectives each with their own respective conflicting values such as price, time, and budget, etc. Therefore, because some travellers find deciding how to plan their trip challenging because of group members conflicting preferences, the GMOOP is designed to convert multi-objective tour trips into a single objective, thus aiming to maximize individual user's satisfaction levels.

After we had developed the ICDM, HM, and GTTRM to provide high accurate recommendations, we developed GMOOP that solves the multi-objective problem where individual objectives are usually in conflict with each other, and the GMOOP is based on the single-objective problem. Figure 7.1 highlights the main problems of TRSs and how, in this thesis, we have developed models to solve these problems. First, the data, constraints, and preferences problems have been solved by the ICDM and the FTRM (see Chapter 3 and 4). Second, the problem of utilising a trip based on user's preferences has been solved by the HM (see Chapter 5), and the group-of-travellers problem is solved by the GTTRM (see Chapter 6). Finally, this chapter proposes a method of solving the multi-objective problem. In conclusion, developing models to address the problems affecting TRSs affect the accuracy of the recommendations produced.

As we have discussed in previous chapters about the datasets for TRSs, a few of generated datasets have been implemented in MOOP, and we have used these datasets (see Table 7.3). In addition, These datasets (have been used in GMMOP) have some challenges (we have discussed in Section 7.6).

7.2 Related work

Travellers have different preferences and requirements for their trip, and each tour trip involves multi-values for selecting a POI. GMOOP aims to personalize tour trip based on multi-values where the current models deal with these multi-values as a multi-objective problem.

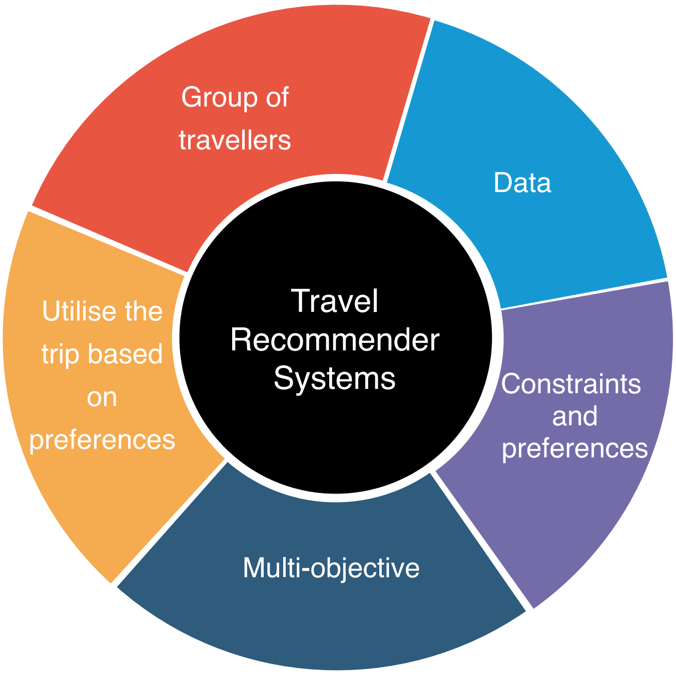


Figure 7.1: Overview TRSs problems

The literature has proposed many methods and approaches for Travel Recommender Systems (TRSs); however, the *Multi-Objective Orienteering Problem* (MOOP) has been specifically introduced to solve the problem where each POI features multiple scores [90]. In addition, the Multi-Constraint Team Orienteering Problem with Multiple Time Windows (MCTOPMTW) has been designed to allow travellers to consider a limited number of attributes for each node [96]. To explain, each node has multiple attributes, and users are able to restrict the total number of attributes over the whole trip.

7.2.1 MOOP

The Multi-Objective Orienteering Problem has been introduced to solve the problem of which POI provides maximum benefits (e.g., educational, cultural, shopping), and the aim of the MOOP is maximizing these total collected benefits based on users preferences.

The MOOP can be defined as follows. Let $G = (V, E)$ be directed weighted graph where $i \in V$ and $i = 1, ..., |V|$ be a set of nodes representing *Points of Interest* (POIs) in a city, and the score for visiting a node i be S_{ik} for k where $k \in K$ which represents K benefits for each item. We are given a starting node s and terminal node t, and let $s = 1$ and $t = |V|$. Also, let Y_i be a decision variable equal to 1 when visiting the i item, otherwise equal to 0, and $X_{i,j}$ be a decision variable equal to 1 when visiting j item after i item, otherwise equal to 0. The time limitation on each day of the trip is represented by T_{max}.

The MOOP has been formulated as follows. The Equation (7.2.1) shows the objective function of MOOP, which maximizes the total of the POIs multiple scores.

$$Max(f_k); \forall k = (1, 2, \ldots, |K|) \tag{7.2.1}$$

Equation (7.2.2) defines f_k which is the sum of the scores for benefits k.

$$f_k = \sum_{i=1}^{|V|} S_{ik} \times Y_i; \forall k = (1, 2, \ldots, K) \tag{7.2.2}$$

Equations 7.2.3 and 7.2.4 ensure that as long as Y_i is equal to 1 the $X_{i,j}$ must also be 1. In other words, as long as node is visited i, one must leave node i to item j.

$$\sum_{i=1}^{|I|-1} X_{i,j} = Y_i \tag{7.2.3}$$

$$\sum_{i=2}^{|V|} X_{i,j} = Y_j \tag{7.2.4}$$

Equation (7.2.5) ensures that there no sub-routes.

$$\sum_{i=1}^{|V|} \sum_{j=1}^{|V|} X_{i,j} \leq |S| - 1; \forall S \subseteq I \land S \neq \emptyset \tag{7.2.5}$$

Equation (7.2.6) is a constraint that ensures the starting point begins at the first node and ends at the last node.

$$Y_0 = Y_{|V|} = 1 \tag{7.2.6}$$

Equation (7.2.7) represents the constraint that the total travel time between nodes is less than or equal to the T_{max}.

$$\sum_{i=1}^{|V|} \sum_{j=1}^{|V|} TT_{(i,j)} \times X_{i,j} \leq T_{max} \tag{7.2.7}$$

Multi-Objective (Team) Orienteering Problem with Time Windows

Multi-Objective (Team) Orienteering Problem with Time Windows (MOTOPTW) is an extension of the MOOP where multi-day (Team) and time windows are considered.

7.2.2 MCTOPMTW

The MCTOPMTW has been introduced by [96], and the main goal of the MCTOPMTW is to maximize the total collected scores by considering maximum numbers in terms of the trips various attributes. Mainly, the MCTOPMTW is an extension of the Team Orienteering Problem with Time Windows (TOPTW). The MCTOPMTW has been formulated as follows. Equation (7.2.8) shows the objective function that maximises the total collected scores under the maximum of each benefit.

$$Max \left(\sum_{p=1}^{|P|} \sum_{t=1}^{|p|} \sum_{i=2}^{|I|-1} S_i \times Y_{pti} \right) \tag{7.2.8}$$

Equation (7.2.9) ensures that the trip starts at node #1 and ends at node #$|P|$.

$$\sum_{p=1}^{|P|} \sum_{j=2}^{|V|} X_{p1j} = \sum_{p=1}^{|P|} \sum_{i=1}^{|I|-1} X_{pi|V|} = |P| \tag{7.2.9}$$

Equation (7.2.10) ensures connectivity within each tour (e.g. tours should be connected).

$$\sum_{i=1}^{|I|-1} X_{pik} = \sum_{j=2}^{|V|} X_{pkj} = \sum_{t=1}^{|p|} Y_{ptk}; \forall k = 2, \ldots, |V|; \forall p = 1, \ldots, |P|. \tag{7.2.10}$$

Equation (7.2.11) ensures that the arrival time in each item for each tour is less is than or equal to the decision variable multiplied by a large number (a large number is used to make sure the right side of the equation is bigger than left side when X_{pkj} is equal 0). The S_{pi} denotes the arrival time at node i in tour p, and the t_{ij} denotes the required time to move from i to j, while L represents a large constant.

$$S_{pi} + t_{ij} - S_{pj} \leq L(1 - X_{pij}) \tag{7.2.11}$$

Equation (7.2.12) guarantees that each node is visited only once.

$$\sum_{p=1}^{|P|}\sum_{t=1}^{|p|} Y_{pti} \leq 1; \forall i = 1, \ldots, |V| \tag{7.2.12}$$

Equation (7.2.13) ensures that the total attributes from each node are less than or equal to E_z (e.g. the i node has a value for z attribute in p tour can be denoted e_{piz}), and E_z denotes the maximum number of attributes of z in the trip.

$$\sum_{p=1}^{|P|}\sum_{t=1}^{|p|} e_{piz} \times Y_{pti} \leq E_z; \forall z = 1, \ldots, Z \tag{7.2.13}$$

Equation (7.2.14) ensures that the visiting time will be displayed in the opening window, while O_{pti} denotes the opening time for i node at t time in tour p, and C_{pti} denotes the closing time for i node at t time in p tour.

$$O_{pti} \leq S_{pi} \leq C_{pti}; \forall i = 1, \ldots, |V|; \forall p = 1, \ldots, |P| \tag{7.2.14}$$

To sum up, because the aim of Recommender Systems (RSs) is to reduce the significant number of options, the proposed model, the GMOOP, supports the RSs aim by aggregating all the multi-values into a single value to optimize them.

7.3 Generalized Multi-Objective Orienteering Problem

We have designed the Generalized Multi-Objective Orienteering Problem (GMOOP) to solve multi-objective problems based on the OP. The GMOOP aims to solve many models using a single algorithm.

We have generalized our model based on the models shown in the following table 7.1.

Table 7.1: List of abbreviation for different problems

Abbreviation	Problems
MOOP	Multi-Objective Orienteering Problem [90]
MOOPTW	MOOP with Time-Windows [34]
MOTOP	Multi-objective Team Orienteering Problem [12]
MOTOPTW	MOTOP with Time-Windows [56]
MOTDOP	Multi-Objective Time-Dependent Orienteering Problem [77]

Table 7.2 shows the main differences between the GMOOP and other models. First, all models are based on the Multi-Objective (MO) approach, where each node has more than one score. Second, Time-Windows (TW) represents the opening/closing times for each node where it is possible to visit the node between these times. The MOOPTW and MOTOPTW consider TW in their problem to solve the problem, where the GMOOP considers not only solves TW but can also solve Multi-Time-Windows (MTW) in which each item might have more than one TW per day. Third, the Multi-Day (MD) trip is critical for building tour routing for travelers. The MOTOP and MOTOPTW include MD considerations when building the tour. However, the GMOOP is also able to solve MD constraints. Fourth, to address the real problem of traffic jams affecting the travel time between nodes, the Time-Dependent (TD) element is considered in the MOTDOP and in our model, the GMOOP. Finally, while on a trip, people spent time at different POIs, and so it is essential to model Staying-Time (ST) at each POI. The GMOOP handles ST whereas other models are unable to model it. In conclusion, the GMOOP has been designed to personalize multi-objectives in the OP, which is represented by the *Travel Recommender Systems* (TRSs), and also, the MOOP represents an optimisation problem that is not specifically designed for TRSs.

Table 7.2: Comparison between GMOOP with other models

Problem	MO	TM	MD	TD	ST
MOOP	•				
MOOPTW	•	•			
MOTOP	•	•			
MOTOPTW	•	•	•		
MOTDOP	•		•		
GMOOP	•	MTW	•	•	•

7.3.1 Mathematical model of GMOOP

The GMOOP can be defined as follows. Let $G = (V, E)$ be directed weighted graph where $i \in V$ and $i = 1, ..., |V|$ be a set of nodes representing a *Point of Interest* (POI) in a city, and E be a set of edges between these nodes (POIs). A cost of traveling between two nodes $i, j \in V$ is denoted by TT_{ij} and the profit of visiting a node i is S_i. Given a starting node s and a terminal node t, let $s = 1$ and $t = |V|$. In addition, ST_i denotes the staying time in the i node. S_{ik} denotes the score k for the item i where $k \in K_i$ and $k = 1, ..., |K_i|$. Also, let X_{ptij} be a decision variable on p day at t time moving from i to j node. The time limitation on each day of the trip is denoted by T_{max}.

Equation (7.3.15) presents the objective function of the GMOOP. However, the main difference between Equation (7.3.15) and Equation 7.2.1 is that the GMOOP aggregates all the different values into a single value whereas the MOOP provides different solutions.

$$Max \left(\sum_{p=1}^{|P|} \sum_{i=1}^{|V|-1} \sum_{j=1}^{|V|} \left(\frac{\sum_{t=1}^{|p|} X_{ptij}}{ST_i + TT_{ij}} \right) \times \sum_{k=1}^{|K|} S_{ik} \right) \tag{7.3.15}$$

Equation (7.3.16) includes constraints to ensure that the total of the decision variable X_{ptij} is equal to $ST_i + TT_{ij}$ (the ST_i and TT_{ij} represent the moving time and visiting time, respectively).

$$\sum_{j=1}^{|V|}\left(\frac{\sum_{t=1}^{|p|}X_{ptij}}{ST_i+TT_{ij}}\right)\leq 1; \tag{7.3.16}$$

$$\forall p=1,\ldots,|P|;\forall i=1,\ldots,|V|$$

Equation (7.3.17) ensures that each tour starts from the 1 node and terminates at the $|V|$ node.

$$\sum_{j=2}^{|V|}\left(\frac{\sum_{t=1}^{|p|}X_{pt1j}}{ST_1+TT_{1j}}\right)=\sum_{p=1}^{|P|}\sum_{i=1}^{|V|-1}\left(\frac{\sum_{t=1}^{|p|}X_{pti|V|}}{ST_i+TT_{i|V|}}\right)=1; \tag{7.3.17}$$

$$\forall p=1,\ldots,|P|$$

Equation (7.3.18) ensures that the total time of each day trip is less than or equal to T_{max}.

$$\left(\sum_{p=1}^{|P|}\sum_{i=1}^{|V|-1}\sum_{j=1}^{|V|}\left(\frac{\sum_{t=1}^{|p|}X_{ptij}}{ST_i+TT_{ij}}\right)\times(ST_i+TT_{ij})\right)\leq T_{max} \tag{7.3.18}$$

Equation (7.3.19) ensures the validity of the continuously connected timeline.

$$\sum_{j=2}^{|V|}\left(\frac{\sum_{t=1}^{|p|}X_{ptuj}}{ST_u+TT_{uj}}\right)=\sum_{p=1}^{|P|}\sum_{i=1}^{|V|-1}\left(\frac{\sum_{t=1}^{|p|}X_{ptiu}}{ST_i+TT_{iu}}\right)\leq 1; \tag{7.3.19}$$

$$\forall p=1,\ldots,|P|;\forall u=1,\ldots,|V|$$

Finally, The fitness function of ACO is Equation 7.3.15 and the GMOOP is able to aggregate the multiple-scores for each node into a single value. Then, we are able to personalize and produce recommendations.

7.4 Experiments

A number of experiments have been conducted to determine the GMOOPs performance based on the existing datasets (see Table 7.3).

These datasets have been selected based on the availability of public data. In other words, the GMOOP is a general model for the MOOP and its extensions, and the experiments have been conducted based on the MOOP's datasets.

Table 7.3: List of all datasets have been used in the experiments

Problem	Dataset Name	Reference	Number of instances	Number of items $\|V\|$
	$Dataset_1$		18	32
	$Dataset_2$		11	21
	$Dataset_3$		20	33
	$Dataset_4$		26	66
GMOOP	$Dataset_5$	[90]	14	64
	$Dataset_6$		20	97
	$Dataset_7$		20	273
	$Dataset_8$		29	559
	$Dataset_9$		29	2143

7.4.1 An Ant Colony Optimization

As the OP represents an approximation scheme [8] because the running time for the OP grows rapidly as the underlying graph grows. The ACO has been adjusted to solve the GMOOP, and Figure 7.2 shows the flowchart of the ACO. The ACO algorithm has been divided into three steps. First, the ACOs Initial parameters are listed based on Table 7.4. Next, the second step is controlling the ACO by adjusting the loops based on the initial step. Third, the most important function of the ACO is in this step where the ants are released to find the best route.

As previously described in Chapter 5, the ACO performs two steps to update the pheromones. The first update is called *update local pheromones* where in step two (see Figure 7.2); after releasing an ant, it checks that the ant has found a better score for the path it has found, so the equation (7.4.22) shows the update for delta for all nodes that have been allocated into the better path. After that, the *Tau* is updated based on Equation (7.4.23). The second update occurs after all ants have been released, and the update is based on equation (7.4.24).

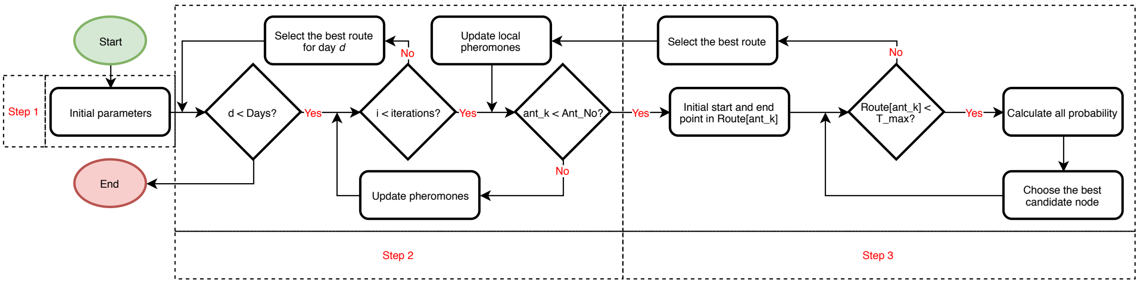


Figure 7.2: Overview of the ACO flowchart

Table 7.4: Initial parameters for ACO in the first step

Parameter	Initial Value	Description
α	0	The value of *Alpha* presents the importance of *Tau*
$\forall\beta_k$	4	The value of *Beta* presents the importance of Eta^k
ρ	0.1	The value of pheromone evaporation
Ant_No	200	Number of ants
Iterations	10	Number of iteration
NodeSize	Number of nodes	
η_{ijk}	Equation (7.4.20)	The Eta^k presents the rate of score to distance
$\tau_{i,j}$	Allocate 1000 value	representing the *Pheromones* level from i to j
$\delta_{i,j}$	Allocate 0 value	representing the maximum total path use i to j

$$\eta_{ijk} = \frac{S_{ik}}{TT_{ij}} \tag{7.4.20}$$

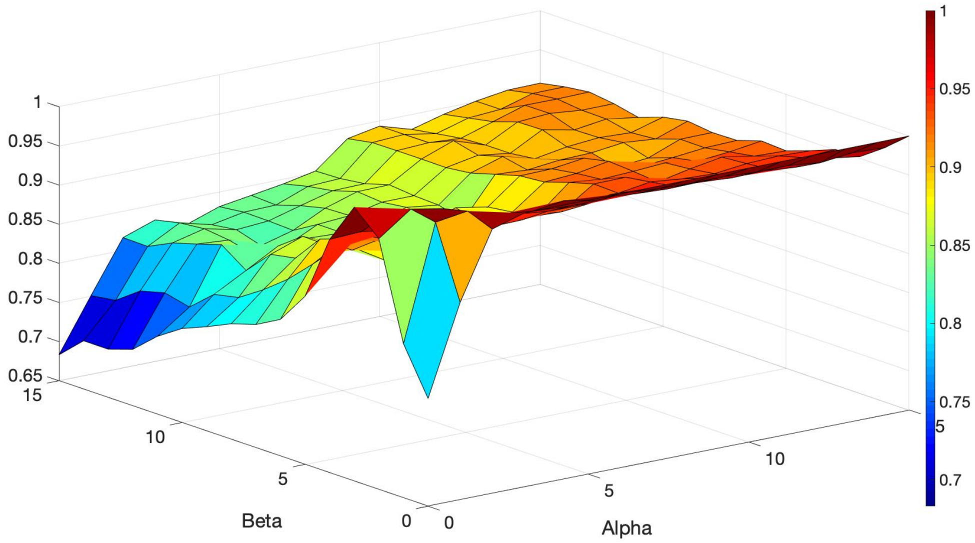
$$P_{i,j} = \frac{(\tau_{i,j})^{\alpha}\left(\prod_{k=1}^{|K|}\eta_{ijk}\right)^{\beta_k}}{\Sigma\left((\tau_{i,j})^{\alpha}\left(\prod_{k=1}^{|K|}\eta_{ijk}\right)^{\beta_k}\right)} \tag{7.4.21}$$

$$\delta_{i,j} = Max(\delta_{i,j}, Ant_x(i,j)) \tag{7.4.22}$$

$$\tau_{i,j} = (1-\rho)\times\tau_{i,j} + \delta_{i,j} \tag{7.4.23}$$

$$\tau_{i,j} = \rho\times\tau_{i,j} + (1-\rho)\times\delta_{i,j} \tag{7.4.24}$$

Figure 7.3: Overview of the ACOs performance based on different values of *Alpha* and $\forall Beta_k$



7.5 Results

Here we present the results for the GMOOP based on the experiments. In addition, the discussion section outlines the challenges and the defects of the existing datasets.

7.5.1 GMOOP benchmark instances

The GMOOP has been examined to compare the results with existing models results. Table 7.5 to 7.13 shows the results of our model where the total scores (total score 1 and total score 2) are shown alongside the total scores achieved by the existing models. In addition, we illustrate the gaps (in percentages) between the results of our model and the existing models. Specifically, the results in green show where our model produced better results than the state of the art while the results in red show where our model performs less effectively than the existing models.

Figure 7.4: Overview of the ACOs performance based on different values of *Alpha* and $\forall Beta_k$

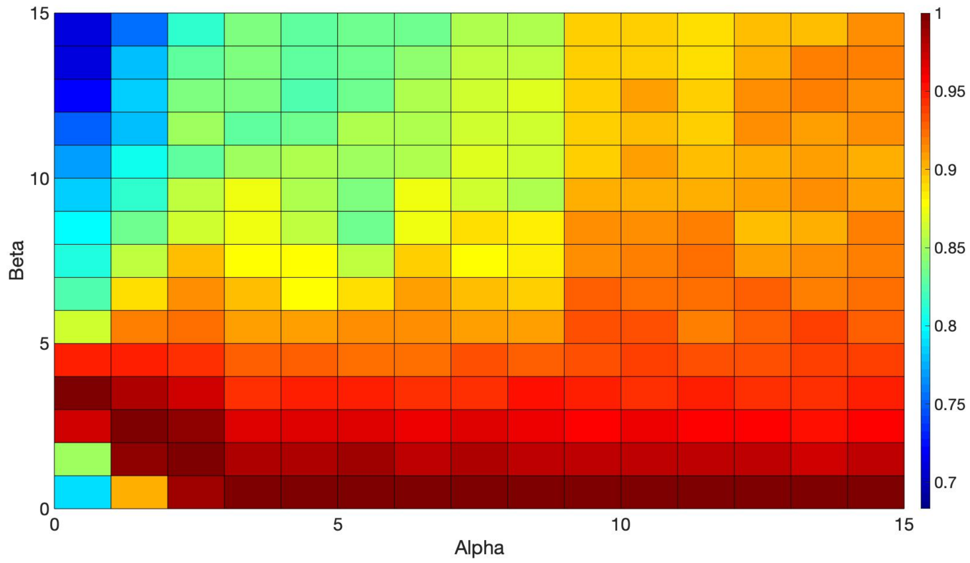


Table 7.5: The $Dataset_1$ results for GMOOP

T_{max}	The GMOOP		The MOOP [90]		The gap between the models	
	Score 1	Score 2	Score 1	Score 2	Score 1	Score 2
5	10	11	10	11	0%	0%
10	15	27	15	27	0%	0%
15	45	55	45	55	0%	0%
20	**65**	79	55	**92**	18%	-14%
25	85	106	**90**	**119**	-6%	-11%
30	105	143	**110**	**158**	-5%	-9%
35	**135**	156	120	**192**	13%	-19%
40	**145**	185	135	**217**	7%	-15%
45	**160**	210	155	**238**	3%	-12%
50	**190**	221	180	**249**	6%	-11%
55	**200**	230	190	**263**	5%	-13%
60	**220**	233	215	**273**	2%	-15%
65	**240**	248	225	**288**	7%	-14%
70	235	295	**245**	**304**	-4%	-3%
73	235	295	**255**	**299**	-8%	-1%
75	255	293	**265**	**306**	-4%	-4%
80	260	309	**275**	**317**	-5%	-3%
85	275	315	**285**	**324**	-4%	-3%

Table 7.5 illustrates the results for the GMOOP and MOOP models in $Dataset_1$. The GMOOP model produces good results, which are within 5% of the results produced by the state-of-the-art models and 44% of Score 1's results in the GMOOP outperform the MOOP (the next section provides a discussion of the results).

Table 7.6: The $Dataset_2$ results for GMOOP

T_{max}	The GMOOP		The MOOP [90]		The gap between the models	
	Score 1	Score 2	Score 1	Score 2	Score 1	Score 2
15	100	290	**120**	**310**	-17%	-6%
20	190	360	**200**	**400**	-5%	-10%
23	200	400	200	400	0%	0%
25	230	410	230	410	0%	0%
27	230	410	230	410	0%	0%
30	240	410	**265**	**450**	-9%	-9%
32	275	420	**290**	**500**	-5%	-16%
35	300	470	**310**	**510**	-3%	-8%
38	330	500	**340**	**530**	-3%	-6%
40	**375**	490	370	**570**	1%	-14%
45	430	590	**450**	**610**	-4%	-3%

Table 7.6 illustrates the GMOOPs performance where 27% of the results are equal to the performance of existing studies. In addition, our algorithm performed better than the state of the art by 9%.

Table 7.7: The $Dataset_3$ results for GMOOP

T_{max}	The GMOOP		The MOOP [90]		The gap between the models	
	Score 1	Score 2	Score 1	Score 2	Score 1	Score 2
15	120	220	**160**	**233**	-25%	-6%
20	140	297	**190**	297	-26%	0%
25	190	**338**	**240**	336	-21%	1%
30	240	396	**280**	**411**	-14%	-4%
35	270	453	**320**	**471**	-16%	-4%
40	310	**518**	**380**	505	-18%	3%
45	380	457	**430**	**554**	-12%	-18%
50	410	516	**440**	**605**	-7%	-15%
55	440	547	**550**	**581**	-20%	-6%
60	470	604	**560**	**632**	-16%	-4%
65	500	627	**580**	**655**	-14%	-4%
70	540	598	**590**	**701**	-8%	-15%
75	560	676	**650**	**696**	-14%	-3%
80	560	685	**670**	**719**	-16%	-5%
85	570	719	**710**	**730**	-20%	-2%
90	650	742	**750**	742	-13%	0%
95	650	742	**750**	742	-13%	0%
100	650	742	**800**	742	-19%	0%
105	650	742	**800**	742	-19%	0%
110	650	742	**800**	742	-19 %	0%

Table 7.7 illustrates the GMOOPs performance compared to the MOOP; our model produced better results by 1% of the results (Score 2). Also, 3% of the results achieve the same performance as the existing studies (Score 2).

Table 7.8: The $Dataset_4$ results for GMOOP

T_{max}	The GMOOP		The MOOP [90]		The gap between the models	
	Score 1	Score 2	Score 1	Score 2	Score 1	Score 2
5	10	97	10	97	0%	0%
10	**40**	156	30	**177**	33%	-12%
15	**95**	227	75	**255**	27%	-11%
20	**185**	211	115	**311**	61%	-32%
25	**280**	242	210	**336**	33%	-28%
30	**330**	271	260	**379**	27%	-28%
35	**455**	264	365	**389**	25%	-32%
40	**525**	283	425	**425**	24%	-33%
45	**650**	282	510	**460**	27%	-39%
50	**690**	320	520	**542**	33%	-41%
55	**825**	299	625	**552**	32%	-46%
60	**885**	325	685	**582**	29%	-44%
65	**900**	414	830	**554**	8%	-25%
70	**1070**	331	880	**591**	22%	-44%
75	**1110**	378	925	**662**	20%	-43%
80	**1195**	417	1045	**627**	14%	-33%
85	**1180**	547	1150	**640**	3%	-15%
90	**1290**	514	1230	**642**	5%	-20%
95	**1395**	536	1295	**681**	8%	-21%
100	**1445**	581	1405	**656**	3%	-11%
105	**1490**	638	1470	**709**	1%	-10%
110	**1550**	674	1500	**760**	3%	-11%
115	**1595**	753	1585	**784**	1%	-4%
120	**1625**	778	1605	**839**	1%	-7%
125	1645	851	1650	**889**	0%	-4%
130	1680	916	1680	916	0%	0%

Table 7.8 shows that 88% of the GMOOPs results perform better than the

MOOPs results (Score 1). In addition, our algorithm achieves the same performance as the MOOP in around 1% of the results.

Table 7.9: The $Dataset_5$ results for GMOOP

T_{max}	The GMOOP		The MOOP [90]		The gap between the models	
	Score 1	Score 2	Score 1	Score 2	Score 1	Score 2
15	96	246	96	**253**	0%	-3%
20	**276**	369	234	**427**	18%	-14%
25	**384**	411	288	**529**	33%	-22%
30	**456**	487	408	**563**	12%	-13%
35	**528**	551	462	**693**	14%	-20%
40	**642**	683	636	**713**	1%	-4%
45	**768**	710	738	**763**	4%	-7%
50	**828**	782	798	**854**	4%	-8%
55	**876**	906	858	**980**	2%	-8%
60	**1032**	911	948	**1071**	9%	-15%
65	**1032**	1049	972	**1188**	6%	-12%
70	**1104**	1136	1080	**1266**	2%	-10%
75	**1188**	1198	1176	**1307**	1%	-8%
80	**1260**	1280	1236	**1414**	2%	-9%

Table 7.9 shows that the GMOOP produced a better result by 92% (Score 1). In addition, our algorithm performs the same performance of the MOOP in approximately 7% of the results.

Table 7.10: The $Dataset_6$ results for GMOOP

T_{max}	The GMOOP		The MOOP [90]		The gap between the models	
	Score 1	Score 2	Score 1	Score 2	Score 1	Score 2
1	80	47	80	47	0%	0%
2	80	47	80	47	0%	0%
3	121	72	121	72	0%	0%
4	**121**	72	86	**145**	41%	-50%
5	**121**	72	86	**145**	41%	-50%
6	127	170	127	170	0%	0%
7	**172**	140	127	**170**	35%	-18%
8	**213**	165	178	**238**	20%	-31%
9	219	263	219	263	0%	0%
10	219	263	219	263	0%	0%
11	**300**	204	243	**310**	23%	-34%
12	**306**	302	284	**335**	8%	-10%
13	331	341	331	341	0%	0%
14	372	366	372	366	0%	0%
15	372	366	371	**374**	0%	-2%
16	418	380	418	380	0%	0%
17	459	405	459	405	0%	0%
18	459	405	459	405	0%	0%
19	**459**	405	339	**412**	35%	-2%
20	**459**	405	346	**436**	33%	-7%

Table 7.10 shows the performance of the GMOOP is the same performance of the MOOP by 60% of the results. In addition, the GMOOP outperforms the other models in approximately 40% of the results (Score 1).

Table 7.11: The $Dataset_7$ results for GMOOP

T_{max}	The GMOOP		The MOOP [90]		The gap between the models	
	Score 1	Score 2	Score 1	Score 2	Score 1	Score 2
1	**92**	50	13	**58**	608%	-14%
2	152	60	**180**	**109**	-16%	-45%
3	**240**	119	202	**192**	19%	-38%
4	332	176	**366**	**273**	-9%	-36%
5	409	235	**453**	**339**	-10%	-31%
6	**406**	369	401	**457**	1%	-19%
7	554	384	**635**	**426**	-13%	-10%
8	644	431	**686**	**524**	-6%	-18%
9	706	563	703	**647**	0%	-13%
10	830	**697**	**947**	648	-12%	8%
11	1007	678	**1074**	**722**	-6%	-6%
12	**1143**	692	1109	**909**	3%	-24%
13	**1247**	769	1206	**1004**	3%	-23%
14	**1320**	837	1306	**1015**	1%	-18%
15	1344	950	**1475**	**1015**	-9%	-6%
16	**1322**	1087	1252	**1401**	6%	-22%
17	1469	1100	**1677**	**1164**	-12%	-5%
18	1470	1258	**1575**	**1445**	-7%	-13%
19	1689	1163	**1888**	**1212**	-11%	-4%
20	1837	1202	**1954**	**1348**	-6%	-11%

Table 7.11 shows our model overcomes the MOOP by 608% when the T_{max} equals 1 (Score 1). In general, by 40% of the results, we produced better results where our model produces the same performance of the MOOP by 5% of the results.

T_{max}	The GMOOP Score 1	The GMOOP Score 2	The MOOP [90] Score 1	The MOOP [90] Score 2	The gap between the models Score 1	The gap between the models Score 2
10	0	0	0	0	0%	0%
15	0	0	0	0	0%	0%
20	0	0	**86**	**98**	-100%	-100%
25	0	0	**152**	**164**	-100%	-100%
30	0	0	**152**	**164**	-100%	-100%
35	0	0	**177**	**245**	-100%	-100%
40	0	0	**285**	**377**	-100%	-100%
45	0	0	**444**	**470**	-100%	-100%
50	0	0	**453**	**534**	-100%	-100%
55	101	90	**530**	**596**	-81%	-85%
60	222	410	**601**	**671**	-63%	-39%
65	311	575	**712**	**666**	-56%	-14%
70	415	723	**785**	**820**	-47%	-12%
75	433	727	**846**	**897**	-49%	-19%
80	622	729	**935**	**1003**	-33%	-27%
85	774	885	**958**	**1158**	-19%	-24%
90	791	687	**1064**	**1174**	-26%	-41%
95	**1183**	1185	1089	**1324**	9%	-10%
100	**1202**	1344	1151	**1456**	4%	-8%
105	**1674**	1454	1206	**1518**	39%	-4%
110	**1875**	1571	1292	**1620**	45%	-3%
115	**2185**	**1820**	1512	1618	45%	12%
120	**2196**	**1966**	1689	1769	30%	11%
125	**2461**	**2151**	1822	1971	35%	9%
130	**2875**	**2775**	2017	2003	43%	39%
135	**3109**	**2710**	2059	2174	51%	25%
140	**3389**	**3175**	2213	2320	53%	37%
145	**3353**	**2934**	2311	2424	45%	21%
150	**3842**	**3615**	2388	2482	61%	46%

Table 7.12: The $Dataset_8$ results for GMOOP

Table 7.12 shows the results of $Dataset_8$ where the GMOOP performs better than the MOOP in about 41% of the results (the next section provides a discussion of the results' performance).

T_{max}	The GMOOP		The MOOP [90]		The gap between the models	
	Score 1	Score 2	Score 1	Score 2	Score 1	Score 2
10	0	0	**482**	**692**	-100%	-100%
15	0	0	**898**	**973**	-100%	-100%
20	0	0	**1431**	**1239**	-100%	-100%
25	41	72	**2074**	**1843**	-98%	-96%
30	416	315	**2074**	**1843**	-80%	-83%
35	483	663	**2211**	**2467**	-78%	-73%
40	879	723	**2551**	**2794**	-66%	-74%
45	938	686	**2811**	**3047**	-67%	-77%
50	1108	827	**3178**	**3395**	-65%	-76%
55	1448	1062	**3592**	**3641**	-60%	-71%
60	1577	1656	**4000**	**3755**	-61%	-56%
65	1890	1831	**4280**	**4046**	-56%	-55%
70	2061	1995	**4711**	**4322**	-56%	-54%
75	2976	2438	**4932**	**4726**	-40%	-48%
80	2599	2362	**5164**	**4986**	-50%	-53%
85	3201	2623	**5558**	**5295**	-42%	-50%
90	3225	2518	**5838**	**5480**	-45%	-54%
95	3569	2787	**6037**	**5676**	-41%	-51%
100	3738	2957	**6222**	**5957**	-40%	-50%
105	4199	3085	**6528**	**6123**	-36%	-50%
110	4344	3285	**6715**	**6413**	-35%	-49%
115	4743	3281	**7078**	**6552**	-33%	-50%
120	5215	3875	**7373**	**6804**	-29%	-43%
125	5290	3877	**7468**	**7024**	-29%	-45%
130	5697	4581	**7710**	**7221**	-26%	-37%
135	6466	4793	**7896**	**7468**	-18%	-36%
140	6739	5346	**8226**	**7532**	-18%	-29%
145	6797	5530	**8517**	**7749**	-20%	-29%
150	7229	5943	**8676**	**8094**	-17%	-27%

Table 7.13: The $Dataset_9$ results for GMOOP

Table 7.13 shows the results of $Dataset_9$ where the GMOOP performs more poorly than the MOOP in terms of results because of the singularities in the dataset (the next section will explain these singularities).

7.6 Discussion

This section discusses the results of the GMOOP and shows the main findings based on the results. We have divided this section into three subsections: (1) discussion on datasets challenges, (2) discussion GMOOP results, and (3) discussion on the Limitation and the challenges of the ACO in GMOOP.

7.6.1 The GMOOP's results Discussion

Table 7.14 shows the performance of our algorithm compared to [90]. The first remark about the performance is that our algorithm provides better results for a long journey in most of the datasets. For example, in $Datasets_8$, our model performed badly when the T_{max} equals 20, but when T_{max} equals 95 or more, our algorithm performed better than the state-of-the-art models.

Table 7.14: The comparison between datasets' results

Dataset	Percentage of the results			
	Same Performance		Better Performance	
	Score 1	Score 2	Score 1	Score 2
$Datasets_1$	16%	16%	44%	-
$Datasets_2$	27%	27%	9%	-
$Datasets_3$	-	30%	-	10%
$Datasets_4$	10%	7%	88%	-
$Datasets_5$	7%	7%	92%	-
$Datasets_6$	60%	55%	40%	-
$Datasets_7$	5%	-	35%	5%
$Datasets_8$	6%	6%	41%	27%
$Datasets_9$	-	-	-	-

$Dataset_1$ to $Dataset_5$ are based on the OP datasets where $Dataset_1$ to $Dataset_5$ feature extra scores for each POI (two scores for each POI) rather than a single score for each POI in the OP. In addition, each one of these datasets is different from the other in terms of the number of POIs, POI locations, start/endpoint locations, and T_{max} value.

Figure 7.5 shows the POI locations and total score for each POI in $Dataset_1$ where the location index is provided above the POIs location and the total scores are in brackets. In addition, the start/end points are shown in blue, and the other POIs are shown in red. Here it is noteworthy that, in terms of performance, the group of POIs located north of the start/end points have high scores. Further, our model is misleading because it chose a high-score, short-distance POI (our model chose POIs located south of the start/end points).

In other words, our model has chosen POI#27 as the first POI when the model was building the tour trip. Thus, choosing a POI located on the opposite side of a group of high-score POIs affects the performance of the model.

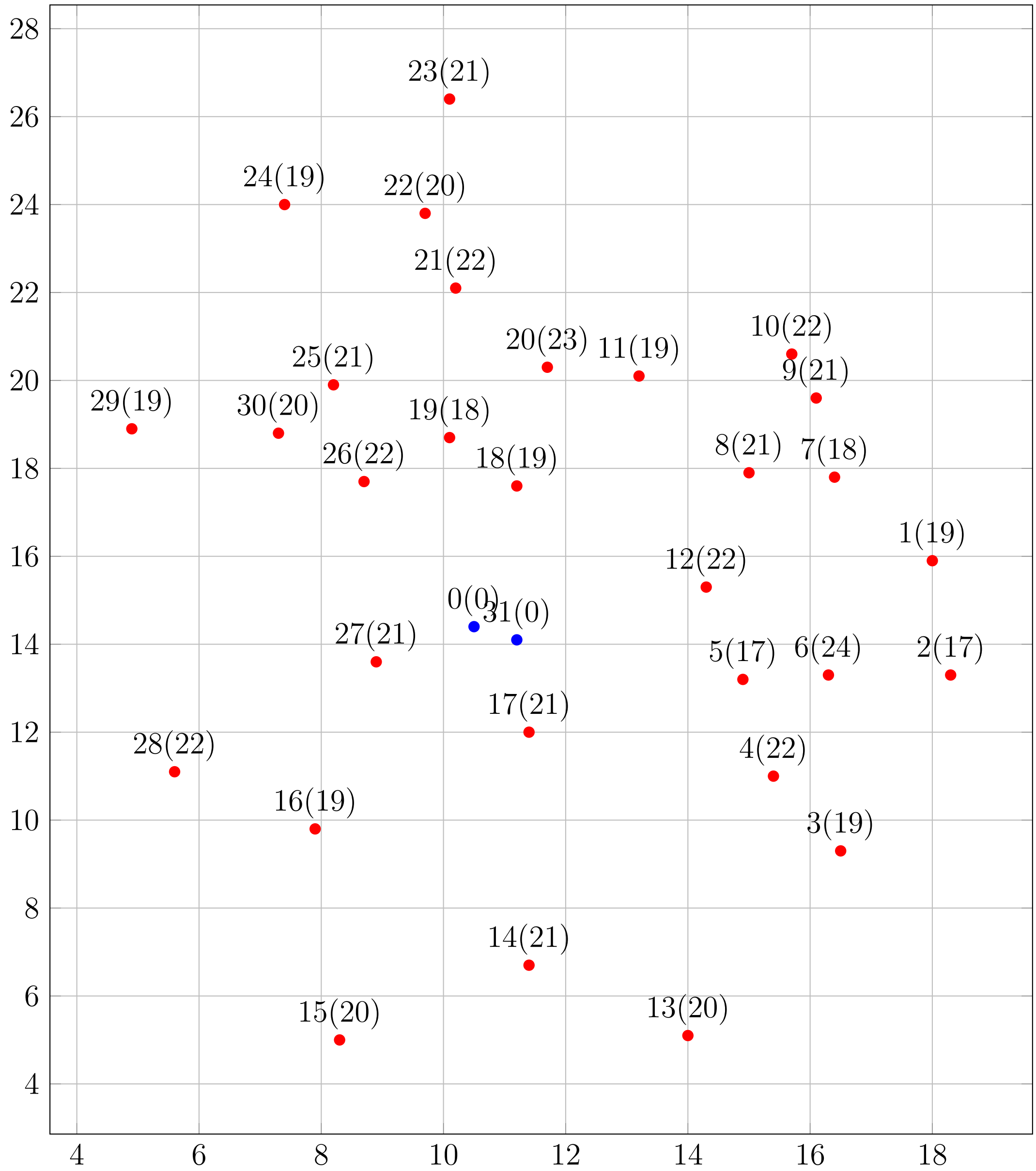


Figure 7.5: Presenting the POIs' location of MOOP's $Dataset_1$

Figure 7.6 shows the POIs' location in $Dataset_2$. In addition, Figure 7.6 shows the index for each POI in the map (in black). Here it is noteworthy to consider the location of the start/end points that are located at the bottom of the map; the location of the start/end points affects the performance of the results because of limitation (T_{max}) in reaching the POIs that are located to the north of the map such as, POI #17, POI #18, and POI #19.

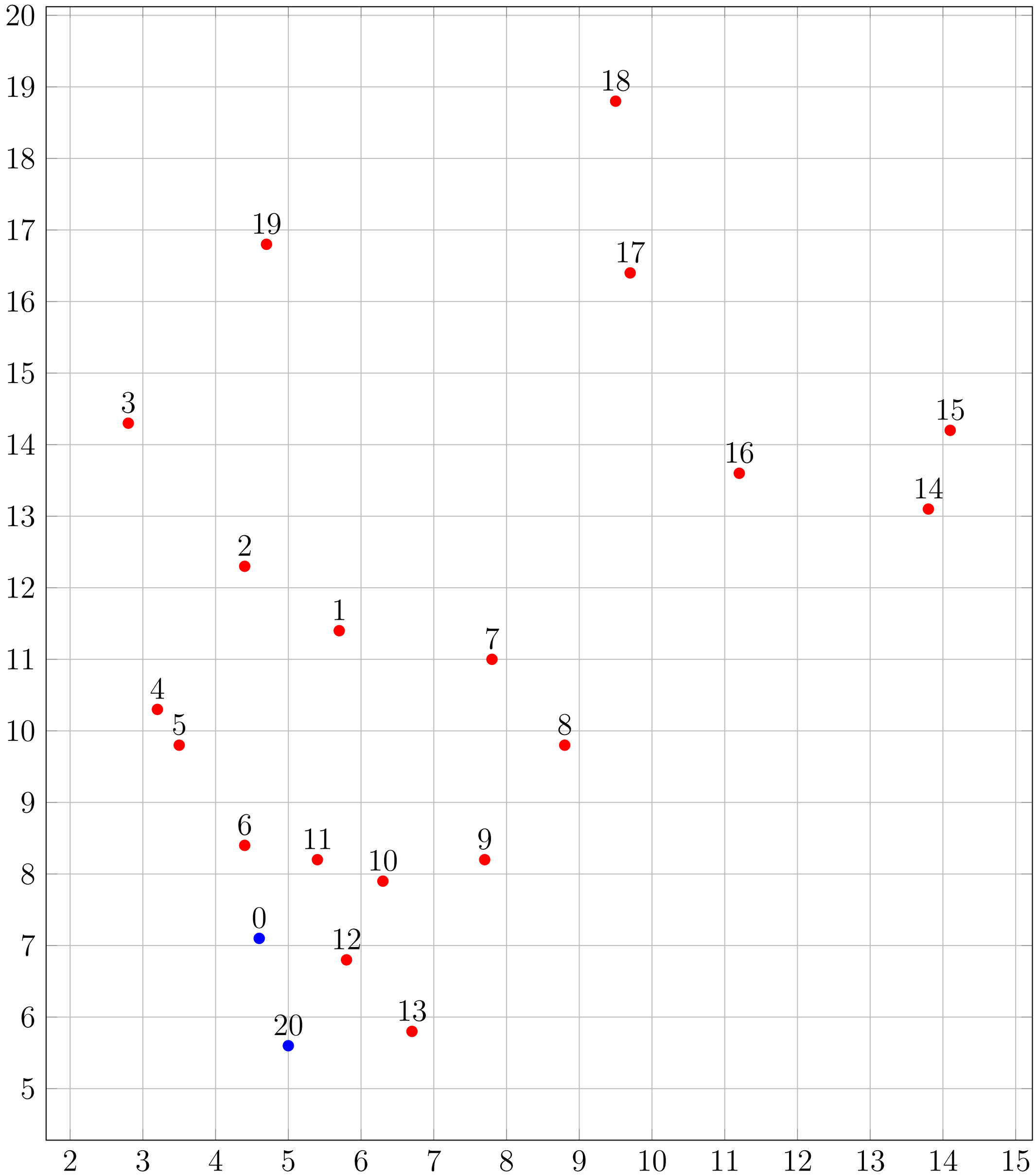


Figure 7.6: Presenting the POIs' location of MOOP's $Dataset_2$

Figure 7.7 shows the POIs' locations and total scores in $Dataset_3$, and the POI's index and total scores are shown above each POI in the map. Our model achieved 30% performance similarity with Score 2's results and 10% of the results were better than the state-of-the-art models for Score 2. However, intelligent guidelines for the model are highly recommended to improve the quality of the results. For example, our model has chosen a sequence of POIs (POI#0, POI#23, POI#6, ..., POI#32); therefore, choosing POI#23 and then POI#6 is not an intelligent choice where POI#21 is located between POI#23 and POI#6. The main reasons that our algorithm could not choose a POI located between two POIs are: (1) our algorithm chooses POIs on an individual basis without taking into account the next choice, (2) the *Alpha* and *Beta* values are chosen based on the best performance over all the datasets.

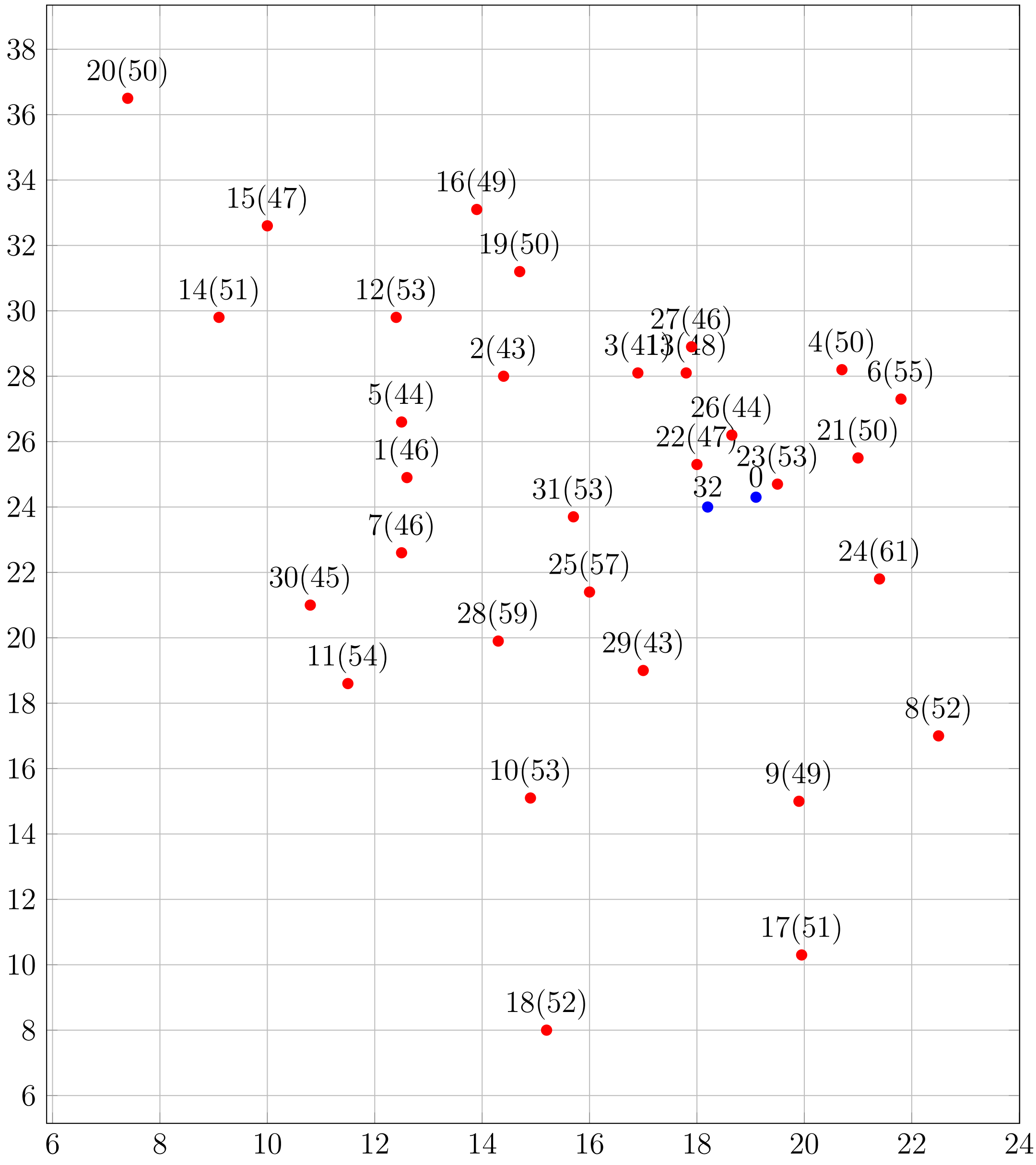


Figure 7.7: Presenting the POIs' location of MOOP's $Dataset_3$

Figures 7.8 and 7.9 show the locations of the POIs in $Dataset_4$ and $Dataset_5$, and the total scores for each POI are provided in parentheses. Our algorithm has produced good results (better results than the-state-of-the-art models) because the fixed distances between POIs and the total scores are close to each other.

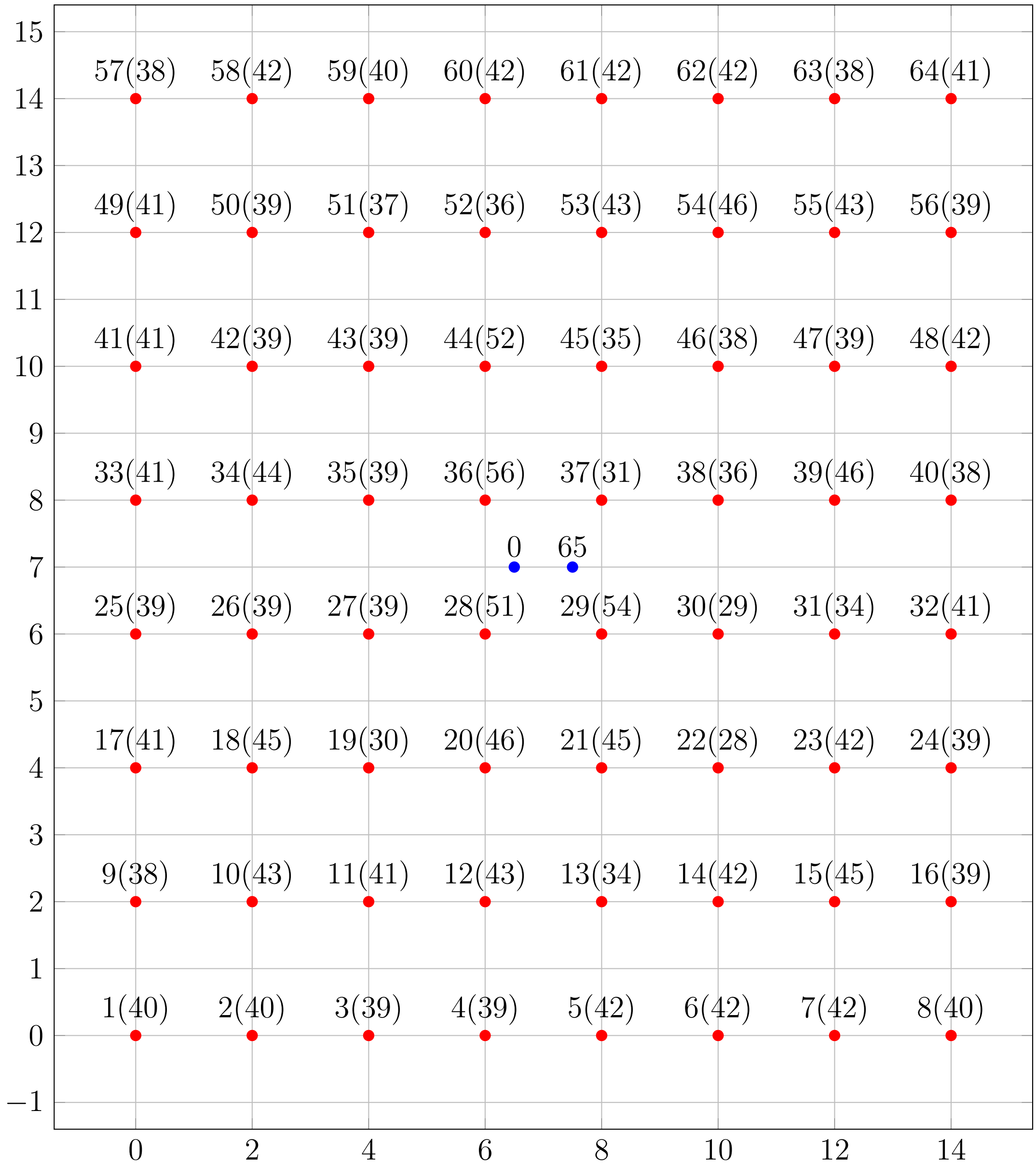


Figure 7.8: Presenting the POIs' location of MOOP's $Dataset_4$

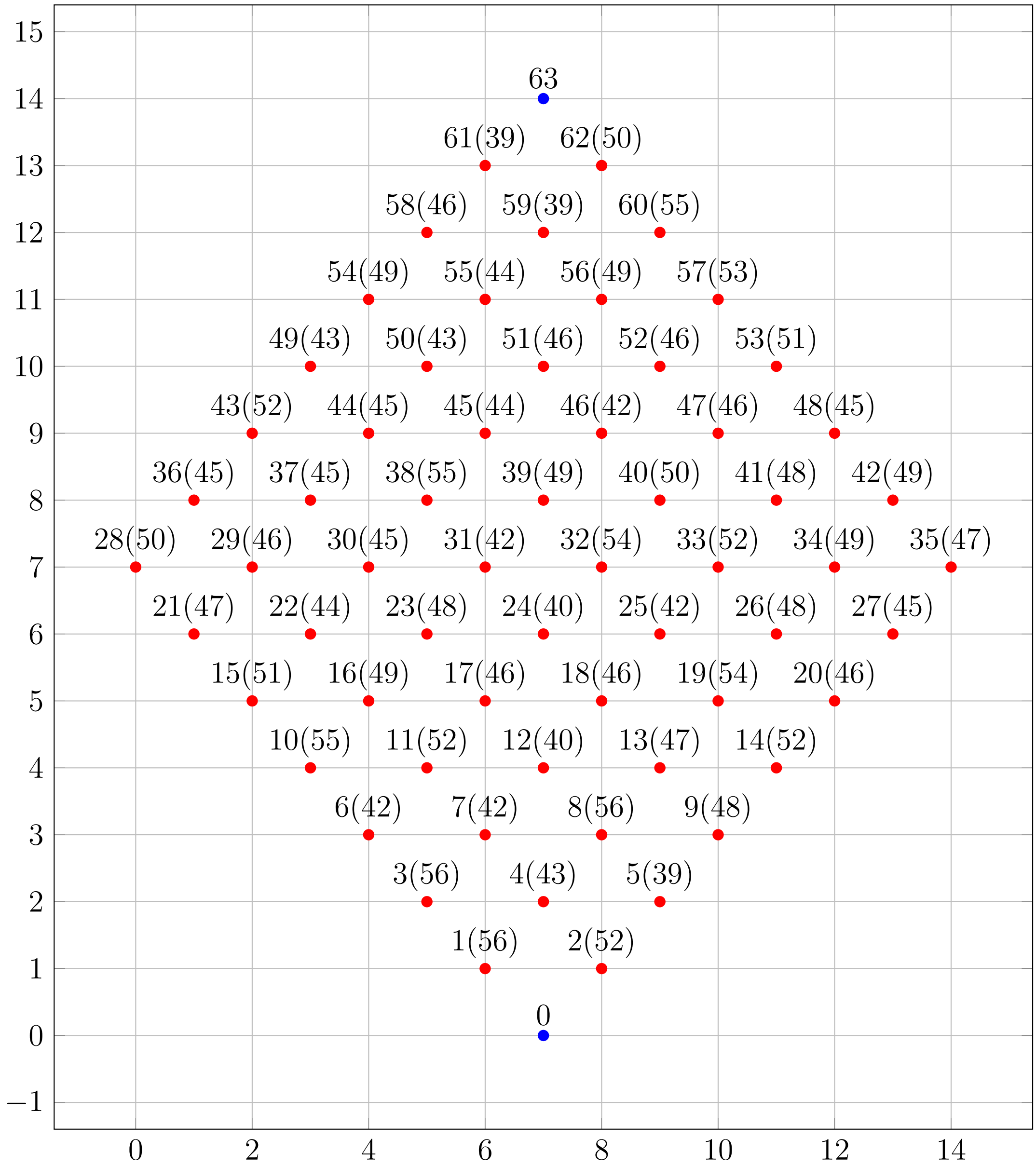


Figure 7.9: Presenting the POIs' location of MOOP's $Dataset_5$

7.6.2 Datasets Discussion

Although some of the GMOOPs results ($Dataset_6$ to $Dataset_9$) fall below those of the state-of-the-art results, there are several reasons for this. It is important to realize that the datasets feature a singularity in terms of the distances listed. For example, because almost all distances from nodes to the end node are large, it is impossible to choose a node to fit with a specific tour trip. Figure 7.10 provides an example of this problem with the datasets ($Dataset_6$ to $Dataset_9$).

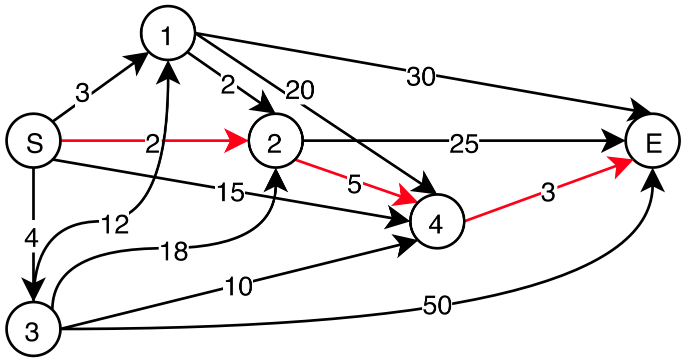


Figure 7.10: Example of the datasets defects

First, the start point is labelled S, and the end point is labelled E. The primary singularity of this dataset is that the cost of moving to and from the same node is not equal to zero (where logically this must be the distance to and from the same node).

In addition, the greatest challenge of these datasets is that almost all nodes feature accessible distances from start nodes except a node, and, at the same time, all these nodes have distance values to the end node that exceed the available time budget (T_{max}). As shown in Figure 7.10, all nodes (except node #4) have accessible distance values, and, at the same time, all of these nodes feature a long-distance value to the end node, which is, in fact, higher than the T_{max}. The red arrow in Figure 7.10 shows the only possible path from the start node to the end node under the T_{max}.

Figures 7.11 to 7.14 show the distribution of the POIs in each dataset where all of these datasets suffer from the distance problem (mentioned above). To clarify, the performance of each dataset is based on how much the dataset suffers from the distance problem. For example, $Dataset_6$, $Dataset_7$, and $Dataset_8$ perform better than $Dataset_9$ because $Dataset_9$ has huge impact on the datasets.

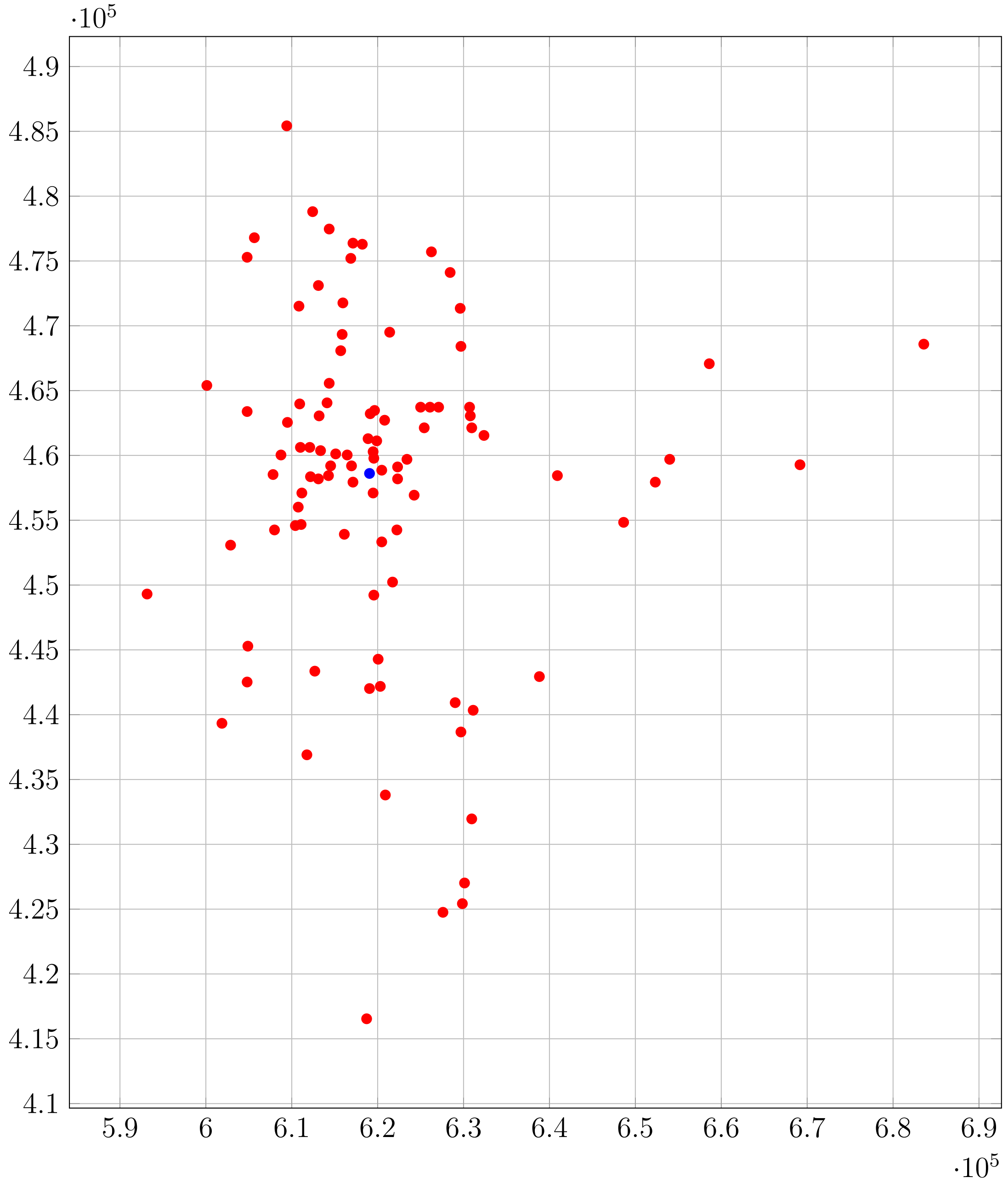


Figure 7.11: Presenting the POIs' location of MOOP's $Dataset_6$

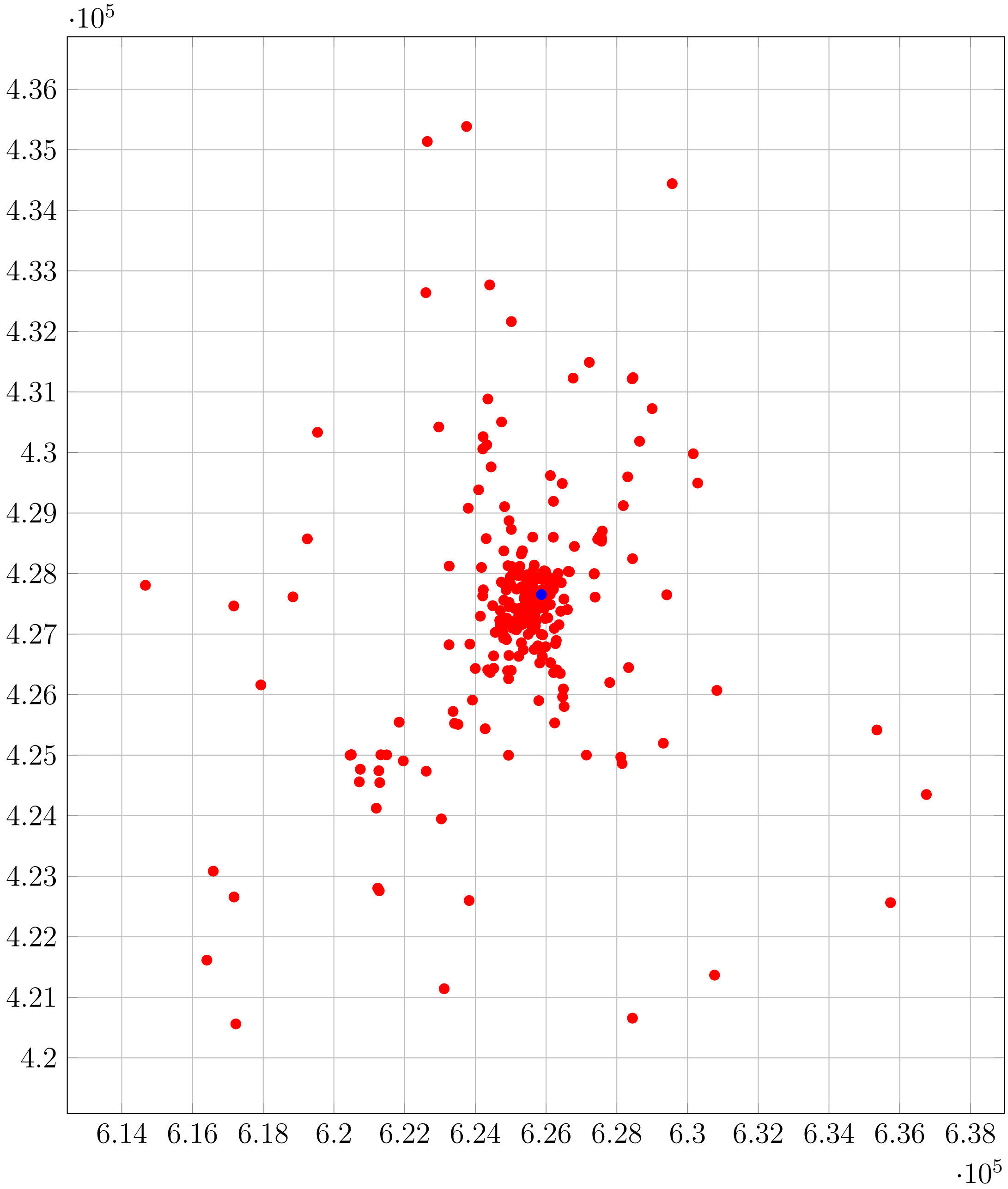


Figure 7.12: Presenting the POIs' location of MOOP's $Dataset_7$

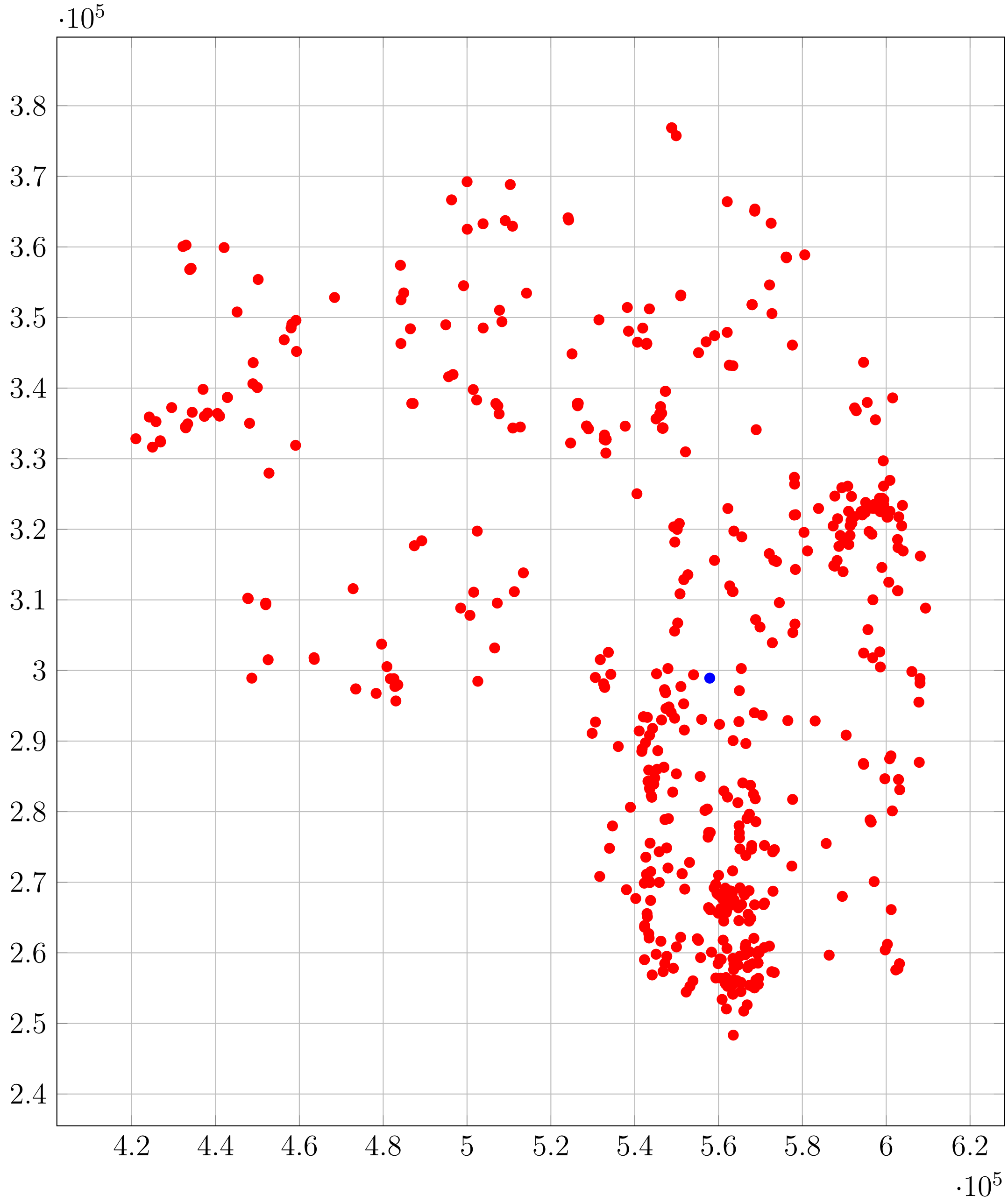


Figure 7.13: Presenting the POIs' location of MOOP's $Dataset_8$

7.6.3 Limitation and challenging of the ACO in GMOOP

As the problem of the MOOP is a subproblem of the OP, and the OP is classified as NP-hard problem. The exact algorithms maybe are applied to the problem under a limited number of POIs. However, the MOOP's datasets ($Dataset_6$ to $Dataset_9$) have a huge number of POIs where the exact algorithms will take a very long time

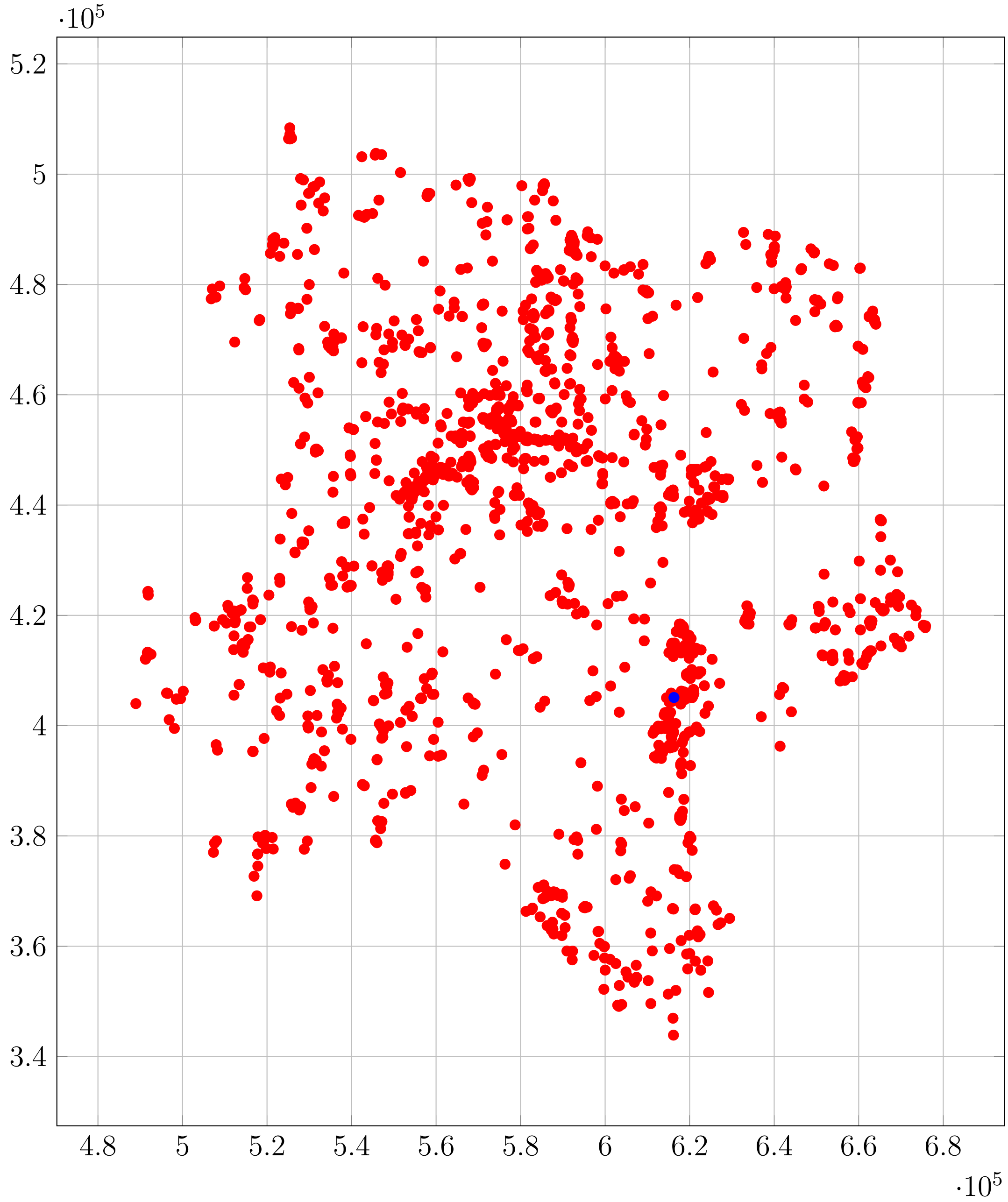


Figure 7.14: Presenting the POIs' location of MOOP's $Dataset_9$

to produce results.

As we have explained in the previous section, the MOOP's datasets have some singularities where affect the ACO. The main proposed to solve the performance of ACO is dynamic settings of ACO. Though, the dynamic setting will not change the ACO performance because the datasets have been designed to be solved differently which sometimes is not logically. The main problem of the datasets is that the algorithms (which will solve the MOOP's datasets) must select two or more POIs each time. In general, general-purpose algorithms are designed to select a candidate node (POI) in each step where it will not perform properly because it will a few or no candidate nodes available to select.

7.7 Conclusion

This chapter has presented the GMOOP, which is a generalized model based on the MOOP. The results of the GMOOP show that while it is possible to produce a tour based on different models, the datasets present several challenges which make it difficult to achieve optimal results with the ACO.

7.7.1 The contributions of this chapter

This chapter has made several significant contributions to the field of Recommender Systems: (1) designing a mathematical model for the GMOOP, (2) implementing the ACO to produce comparable results with a more natural dataset compared with the results of other models.

Chapter 8

Conclusion

This thesis is organized into three main models designed to increase the quality of *personalization*[1] in *tour trips*[1]. These models include aspects of personalization based on *constraints*[1], maximizing the user's *happiness*[1] based on utilizing their preferences[1], and reducing the conflicts among a group of travelers by maximizing the group members' individual satisfaction[1].

8.1 Contributions

This thesis proposes three main models designed to make several significant contributions to the body of research on *Travel Recommender Systems*.

8.1.1 Item Constraints Data Model

The first major contribution of this thesis is the Item Constraints Data Model (ICDM), a novel model that can personalize a tour-trip plan based on the user's constraints[1]. Prior to the research being conducted for this thesis, I designed a number of models featuring fixed constraints where users are not able to customize their constraints based on their requirements.

[1]See Glossary for definitions of italicized words

8.1.2 Happiness Model

The second major contribution this thesis makes is the *Happiness Model* (HM), a novel approach that can maximize a user's happiness based on the user's preferences. The results of the HM show the model can personalize each moment of the trip based on the users preferences as much as possible.

8.1.3 Group Tour Trip Recommender Model

The third major contribution this thesis makes is the Group Tour Trip Recommender Model (GTTRM), which is designed to handle different users' constraints[1]. We developed the *Group Ant Colony Optimization* (GACO) algorithm to solve the Group Tourist Trip Design Problem (GTTDP) by recommending sub-routes for the group to reduce the potential conflicts among them.

8.2 Limitations

The main limitations of the three models proposed in this thesis will be discussed as follows:

- Although the first model handles item constraints, it only handles fixed trip constraints (trip constraints are conditions which are located to control the trip), such as trip length. Most of the existing models are designed based on base constraints (are those conditions which are must be satisfied to make a feasible tour solution).
- The second model is limited because its optimization is very complicated as the optimizations are based on three separate trip components.
- The third model is limited because the social relationship between users in the group is based on an explicit value, while social influence is not considered.

8.3 Future Work

In recent years, the smart wearable systems are increased [29], and in future, more people will use wearable devices which have GPS, heart monitor, and accelerometer. In the near future, the increases in using wearing devices such as smartwatch might lead to the availability of travellers data. Not only availability of data is will be accessible but also, understanding the travellers' reactions and happiness level by collecting the vital data (such as heart rate) for each traveller.

The most impact of the work in this thesis is providing a recommendation for travellers. As voice commands systems have been increased (such as Google Assistant where is able to book an appointment for the user), in future, such as these systems might be connected to the RSs for support travellers where the travellers just ask the system to recommend a trip next weekend or next holiday.

Several possible applications of this research are (1)mainly in Travel Recommender Systems, (2)Delivery Systems, (3)Cycling Routing Problem, and (4)Electric Vehicle Routing Problem.

We consider several possible directions for future work on *Travel Recommender Systems.* The suggested future directions are as follows:

- Developing the ICDM to build a more dynamic and comprehensive model that considers all types of constraints (trip, connection, and item).
- Developing an algorithm that can optimize all trip components to improve performance.
- Investigating the social influence factors and how they can be modelled in the GTTRM.

Bibliography

[1] Britain's visitor economy facts — VisitBritain. [Online] Available from: https://www.visitbritain.org/visitor-economy-facts [Accessed 13 June 2019].

[2] M. N. Ab Wahab, S. Nefti-Meziani, and A. Atyabi. A comprehensive review of swarm optimization algorithms. *PloS one*, 10(5), 2015.

[3] P. H. Abreu, D. C. Silva, F. Almeida, and J. Mendes-Moreira. Improving a simulated soccer team's performance through a Memory-Based Collaborative Filtering approach. *Applied Soft Computing Journal*, 2014.

[4] C. C. Aggarwal. Knowledge-Based Recommender Systems. *Recommender Systems*, pages 167–197, 2016.

[5] I. Alina Christensen and S. Schiaffino. Social influence in group recommender systems. *Online Information Review*, 38(4):524–542, jun 2014.

[6] K. Alodhaibi, A. Brodsky, and G. A. Mihaila. COD: Iterative utility elicitation for diversified composite recommendations. In *Proceedings of the Annual Hawaii International Conference on System Sciences*, pages 1–10, 2010.

[7] S. Amer-Yahia, S. B. Roy, A. Chawlat, G. Das, and C. Yu. Group recommendation: Semantics and efficiency. *Proceedings of the VLDB Endowment*, 2(1):754–765, 2009.

[8] A. Anagnostopoulos, R. Atassi, L. Becchetti, A. Fazzone, and F. Silvestri. Tour recommendation for groups. *Data Mining and Knowledge Discovery*, 31(5):1157–1188, 2017.

[9] A. Angel, S. Chaudhuri, G. Das, and N. Koudas. Ranking objects based on relationships and fixed associations. In *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, pages 910–921, 2009.

[10] L. Baltrunas, T. Makcinskas, and F. Ricci. Group recommendations with rank aggregation and collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 119–126, 2010.

[11] J. P. Baskin and S. Krishnamurthi. Preference aggregation in group recommender systems for committee decision-making. In *Proceedings of the third ACM conference on Recommender systems*, pages 337–340. ACM, 2009.

[12] H. Bederina and M. Hifi. A hybrid multi-objective evolutionary algorithm for the team orienteering problem. In *2017 4th International Conference on Control, Decision and Information Technologies (CoDIT)*, pages 0898–0903. IEEE, apr 2017.

[13] S. Berkovsky and J. Freyne. Group-based recipe recommendations: analysis of data aggregation strategies. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 111–118, 2010.

[14] C. Binucci, F. De Luca, E. Di Giacomo, G. Liotta, and F. Montecchiani. Designing the Content Analyzer of a Travel Recommender System. *Expert Systems with Applications*, 87:199–208, 2017.

[15] C. Blum and X. Li. *Swarm intelligence in optimization*. Springer, 2008.

[16] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal. Generalization of recommender systems: Collaborative filtering extended to groups of users and restricted to groups of items. *Expert Systems with Applications*, 39(1):172–186, 2012.

[17] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez. Recommender systems survey. *Knowledge-Based Systems*, 46:109–132, 2013.

[18] K. Bok, J. Lim, H. Yang, and J. Yoo. Social group recommendation based on dynamic profiles and collaborative filtering. *Neurocomputing*, 209:3–13, 2016.

[19] L. Boratto and S. Carta. ART: group recommendation approaches for automatically detected groups. *International Journal of Machine Learning and Cybernetics*, 6(6):953–980, 2015.

[20] J. Borràs, A. Moreno, and A. Valls. Intelligent tourism recommender systems: A survey. *Expert Systems with Applications*, 41(16):7370–7389, 2014.

[21] H. Bouly, D. C. Dang, and A. Moukrim. A memetic algorithm for the team orienteering problem. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 4974 LNCS, pages 649–658, 2008.

[22] I. R. Brilhante, J. A. Macedo, F. M. Nardini, R. Perego, and C. Renso. On planning sightseeing tours with TripBuilder. *Information Processing and Management*, 51(2):1–15, 2015.

[23] A. Brodsky, S. M. Henshaw, and J. Whittle. CARD : A Decision-Guidance Framework and Application for Recommending Composite Alternatives. In *[RecSys2008]Proceedings of the 2008 ACM conference on Recommender systems*, 2008.

[24] E. K. Burke and G. Kendall. *Search methodologies*. Springer, 2014.

[25] R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and UserAdapted Interaction*, 12(4):331–370, 2002.

[26] W. Carrer-Neto, M. L. Hernández-Alcaraz, R. Valencia-García, and F. García-Sánchez. Social knowledge-based recommender system. Application to the movies domain. *Expert Systems with Applications*, 39(12):10990–11000, 2012.

[27] J. Castro, R. Yera, and L. Martínez. An empirical study of natural noise management in group recommendation systems. *Decision Support Systems*, 94:1–11, 2017.

[28] A. Chakraborty and A. K. Kar. Swarm intelligence: A review of algorithms. In *Nature-Inspired Computing and Optimization*, pages 475–494. Springer, 2017.

[29] M. Chan, D. EstèVe, J.-Y. Fourniols, C. Escriba, and E. Campo. Smart wearable systems: Current status and future challenges. *Artificial intelligence in medicine*, 56(3):137–156, 2012.

[30] 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.

[31] I. M. Chao, B. L. Golden, and E. A. Wasil. The team orienteering problem. *European Journal of Operational Research*, 88(3):464–474, 1996.

[32] W. Chen, Y. jun Shi, H. fei Teng, X. ping Lan, and L. chen Hu. An efficient hybrid algorithm for resource-constrained project scheduling. *Information Sciences*, 180(6):1031–1039, 2010.

[33] Y. Chen and X. R. Li. Does a happy destination bring you happiness? Evidence from Swiss inbound tourism. *Tourism Management*, 65:256–266, 2018.

[34] Y.-H. Chen, W.-J. Sun, and T.-C. Chiang. Multiobjective orienteering problem with time windows: An ant colony optimization algorithm. In *2015 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, pages 128–135. IEEE, nov 2015.

[35] Y.-L. Chen, L.-C. Cheng, and C.-N. Chuang. A group recommendation system with consideration of interactions among group members. *Expert Systems with Applications*, 34(3):2082–2090, apr 2008.

[36] H. S. Chiang and T. C. Huang. User-adapted travel planning system for personalized schedule recommendation. *Information Fusion*, 21(1):3–17, 2015.

[37] I. A. Christensen and S. Schiaffino. Entertainment recommender systems for group of users. *Expert Systems with Applications*, 38(11):14127–14135, 2011.

[38] M. De Choudhury, R. Lempel, C. Yu, N. Golbandi, M. Feldman, and S. Amer-Yahia. Automatic construction of travel itineraries using social breadcrumbs. In *Proceedings of the 21st ACM conference on Hypertext and hypermedia*, pages 35–44, 2010.

[39] M. Dorigo, V. Maniezzo, and A. Colorni. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, 26(1):29–41, 1996.

[40] M. D. Ekstrand, J. T. Riedl, J. A. Konstan, et al. Collaborative Filtering Recommender Systems. *Foundations and Trends in HumanComputer Interaction*, 4(2):81–173, 2011.

[41] S. H. Fang, E. H. C. Lu, and V. S. Tseng. Trip recommendation with multiple user constraints by integrating point-of-interests and travel packages. In *Proceedings - IEEE International Conference on Mobile Data Management*, volume 1, pages 33–42, 2014.

[42] A. Felfernig, L. Boratto, M. Stettinger, and M. Tkalčič. *Group Recommender Systems: An Introduction*. 2018.

[43] A. Felfernig, G. Friedrich, and L. Schmidt-Thieme. Guest Editors' Introduction: Recommender Systems. *IEEE Intelligent Systems*, 22(3):18–21, may 2007.

[44] A. Garcia, M. Linaza, and O. Arbelaitz. Evaluation of Intelligent Routes for Personalised Electronic Tourist Guides. *Information and Communication Technologies in Tourism 2012 SE - 25*, 3:284–295, 2012.

[45] I. Garcia, L. Sebastia, and E. Onaindia. On the design of individual and group recommender systems for tourism. *Expert Systems with Applications*, 38(6):7683–7692, 2011.

[46] I. Garcia, L. Sebastia, E. Onaindia, and C. Guzman. A group recommender system for tourist activities. In *Lecture Notes in Computer Science (including*

subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pages 26–37, 2009.

[47] M. Gartrell, R. Han, S. Mishra, Q. Lv, A. Beach, X. Xing, and K. Seada. Enhancing group recommendation by incorporating social relationship interactions. In *Proceedings of the 16th ACM international conference on Supporting group work*, pages 97–106, 2010.

[48] 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, 2014.

[49] D. Gavalas, C. Konstantopoulos, K. Mastakas, and G. Pantziou. Mobile recommender systems in tourism. *Journal of Network and Computer Applications*, 39(1):319–333, 2014.

[50] Y. Ge, Q. Liu, H. Xiong, A. Tuzhilin, and J. Chen. Cost-aware travel tour recommendation. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '11*, page 983, New York, New York, USA, 2011. ACM Press.

[51] Y. Ge, H. Xiong, A. Tuzhilin, and Q. Liu. Cost-Aware Collaborative Filtering for Travel Tour Recommendations. *ACM Transactions on Information Systems*, 32(1):1–31, 2014.

[52] S. Ghazarian and M. A. Nematbakhsh. Enhancing memory-based collaborative filtering for group recommender systems. *Expert Systems with Applications*, 42(7):3801–3812, may 2015.

[53] A. Gionis, T. Lappas, K. Pelechrinis, and E. Terzi. Customized Tour Recommendations in Urban Areas. *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, pages 313–322, 2014.

[54] A. Gunawan, H. C. Lau, and P. Vansteenwegen. Orienteering Problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, 255(2):315–332, 2016.

[55] J. Guo, Y. Zhu, A. Li, Q. Wang, and W. Han. A Social Influence Approach for Group User Modeling in Group Recommendation Systems. *IEEE Intelligent Systems*, 31(5):40–48, sep 2016.

[56] I. Hapsari, I. Surjandari, and K. Komarudin. Solving multi-objective team orienteering problem with time windows using adjustment iterated local search. *Journal of Industrial Engineering International*, pages 1–15, jun 2019.

[57] J. He, H. Liu, and H. Xiong. SocoTraveler: Travel-package recommendations leveraging social influence of different relationship types. *Information & Management*, 53:934–950, 2016.

[58] A. Hernando, J. Bobadilla, and F. Ortega. A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model. *Knowledge-Based Systems*, 97:188–202, 2016.

[59] D. Herzog and W. Wörndl. A travel recommender system for combining multiple travel regions to a composite trip. 2014.

[60] R. Interdonato, S. Romeo, A. Tagarelli, and G. Karypis. A versatile graph-based approach to package recommendation. In *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, pages 857–864, 2013.

[61] A. Jameson, S. Baldes, and T. Kleinbauer. Enhancing mutual awareness in group recommender systems. In *Proceedings of the IJCAI*, 2003.

[62] A. Jameson and B. Smyth. Recommendation to groups. In *The adaptive web*, pages 596–627. Springer, 2007.

[63] S. Jiang, X. Qian, T. Mei, and Y. Fu. Personalized Travel Sequence Recommendation on Multi-Source Big Social Media. *IEEE Transactions on Big Data*, 2(1):43–56, 2016.

[64] K. Kabassi. Personalizing recommendations for tourists. *Telematics and Informatics*, 27(1):51–66, feb 2010.

[65] H. N. Kim and A. E. Saddik. A stochastic approach to group recommendations in social media systems. *Information Systems*, 50:76–93, 2015.

[66] J. K. Kim, H. K. Kim, H. Y. Oh, and Y. U. Ryu. A group recommendation system for online communities. *International Journal of Information Management*, 30(3):212–219, 2010.

[67] G. Koutrika, B. Bercovitz, and H. Garcia-Molina. FlexRecs: expressing and combining flexible recommendations. In *Proceedings of the 35th SIGMOD international conference on Management of data*, pages 745–758, 2009.

[68] A. F. Lemma. Professional guides and applied knowledge services. Technical report, Economic and private sector, 2014.

[69] J. Li, W. Xu, W. Wan, and J. Sun. Movie recommendation based on bridging movie feature and user interest. *Journal of Computational Science*, 26:128–134, may 2018.

[70] Y. C. Liang and A. E. Smith. An ant colony approach to the orienteering problem. *Journal of the Chinese Institute of Industrial Engineers*, 23(5):403–414, 2006.

[71] Q. Liu, E. Chen, H. Xiong, Y. Ge, Z. Li, and X. Wu. A cocktail approach for travel package recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 26(2):278–293, 2014.

[72] M. López-Ibáñez and C. Blum. Beam-ACO for the travelling salesman problem with time windows. *Computers and Operations Research*, 2010.

[73] J. P. Lucas, N. Luz, M. N. Moreno, R. Anacleto, A. Almeida Figueiredo, and C. Martins. A hybrid recommendation approach for a tourism system. *Expert Systems with Applications*, 40(9):3532–3550, jul 2013.

[74] Y. Marinakis, M. Politis, M. Marinaki, and N. Matsatsinis. A memetic-GRASP algorithm for the solution of the orienteering problem. In *Advances in Intelligent Systems and Computing*, 2015.

[75] J. Masthoff. Group Recommender Systems: Combining Individual Models. In *Recommender Systems Handbook*, pages 677–702. Springer US, Boston, MA, 2011.

[76] 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*, pages 363–372. ACM, 1998.

[77] Y. Mei, F. D. Salim, and X. Li. Efficient meta-heuristics for the Multi-Objective Time-Dependent Orienteering Problem. *European Journal of Operational Research*, 254(2):443–457, oct 2016.

[78] A. Moreno, A. Valls, D. Isern, L. Marin, and J. Borràs. SigTur/E-Destination: Ontology-based personalized recommendation of Tourism and Leisure Activities. *Engineering Applications of Artificial Intelligence*, 26(1):633–651, jan 2013.

[79] L. Naamani-Dery, M. Kalech, L. Rokach, and B. Shapira. Iterative voting under uncertainty for group recommender systems. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 265–268, 2010.

[80] J. Nawijn. Determinants of Daily Happiness on Vacation. *Journal of Travel Research*, 50(5):559–566, sep 2011.

[81] K. O'Hara, M. Lipson, M. Jansen, A. Unger, H. Jeffries, and P. Macer. Jukola: democratic music choice in a public space. In *Proceedings of the 5th conference on Designing interactive systems: processes, practices, methods, and techniques*, 2004.

[82] D. H. Park, H. K. Kim, I. Y. Choi, and J. K. Kim. A literature review and classification of recommender systems research. *Expert Systems with Applications*, 2012.

[83] M. Pazzani. A framework for collaborative, content-based and demographic filtering. *Springer*, 1999.

[84] G. Popescu. Group recommender systems as a voting problem. In *International Conference on Online Communities and Social Computing*, pages 412–421. Springer, 2013.

[85] P. Pu, L. Chen, and R. Hu. Evaluating recommender systems from the user's perspective: Survey of the state of the art. *User Modeling and User-Adapted Interaction*, 2012.

[86] A. Puris, R. Bello, and F. Herrera. Analysis of the efficacy of a Two-Stage methodology for ant colony optimization: Case of study with TSP and QAP. *Expert Systems with Applications*, 2010.

[87] F. Ricci, L. Rokach, and B. Shapira. Introduction to Recommender Systems Handbook. In *Recommender Systems Handbook*, pages 1–35. Springer US, Boston, MA, 2011.

[88] M. Salamó, K. McCarthy, and B. Smyth. Generating recommendations for consensus negotiation in group personalization services. In *Personal and Ubiquitous Computing*, 2012.

[89] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. Collaborative Filtering Recommender Systems. In *The Adaptive Web*, pages 291–324. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[90] M. Schilde, K. F. Doerner, R. F. Hartl, and G. Kiechle. Metaheuristics for the bi-objective orienteering problem. *Swarm Intelligence*, 3(3):179–201, 2009.

[91] V. Selvi and R. Umarani. Comparative analysis of ant colony and particle swarm optimization techniques. *International Journal of Computer Applications*, 5(4):1–6, 2010.

[92] Z. Sevkli and E. Sevilgen. Discrete Particle Swarm Optimization for the Orienteering Problem. *IEEE World Congress on Computational Intelligence*, pages 18–23, 2010.

[93] S. Shang, Y. Hui, P. Hui, P. Cuff, S. K. P. of the 29th, and U. 2014. Beyond personalization and anonymity: Towards a group-based recommender system. *dl.acm.org*, 2014.

[94] K. Siang Tan and N. A. Mat Isa. Color image segmentation using histogram thresholding Fuzzy C-means hybrid approach. *Pattern Recognition*, 2011.

[95] K. M. Sim and W. H. Sun. Ant colony optimization for routing and load-balancing: Survey and new directions. *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans.*, 2003.

[96] 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, 2013.

[97] D. Sprague, F. Wu, and M. Tory. Music selection using the PartyVote democratic jukebox. In *Proceedings of the working conference on Advanced visual interfaces*, pages 433–436. ACM, 2008.

[98] K. Sylejmani, J. Dorn, and N. Musliu. Planning the trip itinerary for tourist groups. *Information Technology and Tourism*, 17(3):275–314, 2017.

[99] H. A. Taha. Integer Programming: Theory, Applications, and Computations. 1975.

[100] C. Tan, Q. Liu, E. Chen, H. Xiong, and X. Wu. Object-Oriented Travel Package Recommendation. *ACM Transactions on Intelligent Systems and Technology*, 5(3):1–26, 2014.

[101] S. Topaloglu and H. Selim. Nurse scheduling using fuzzy modeling approach. *Fuzzy Sets and Systems*, 161(11):1543–1563, 2010.

[102] P. Toth and D. Vigo. *The vehicle routing problem.* SIAM, 2002.

[103] T. Tsiligirides. Heuristic methods applied to orienteering. *Journal of the Operational Research Society*, 35(9):797–809, 1984.

[104] P. Vansteenwegen, W. Souffriau, G. V. Berghe, and D. V. Oudheusden. The city trip planner: An expert system for tourists. *Expert Systems with Applications*, 2011.

[105] P. Vansteenwegen, W. Souffriau, and D. V. Oudheusden. The orienteering problem: A survey. *European Journal of Operational Research*, 209(1):1–10, 2011.

[106] P. Vansteenwegen, W. Souffriau, G. Vanden Berghe, and D. Van Oudheusden. Iterated local search for the team orienteering problem with time windows. *Computers & Operations Research*, 36(12):3281–3290, 2009.

[107] P. Vansteenwegen and D. Van Oudheusden. The Mobile Tourist Guide: An OR Opportunity. *OR Insight*, 20(3):21–27, 2007.

[108] W. Verbeke, D. Martens, C. Mues, and B. Baesens. Building comprehensible customer churn prediction models with advanced rule induction techniques. *Expert Systems with Applications*, 2011.

[109] M. Xie, L. V. Lakshmanan, and P. T. Wood. Breaking out of the box of recommendations: from items to packages. In *Proceedings of the fourth ACM conference on Recommender systems - RecSys '10*, 2010.

[110] M. Xie, L. V. Lakshmanan, and P. T. Wood. CompRec-Trip: A composite recommendation system for travel planning. In *Proceedings - International Conference on Data Engineering*, 2011.

[111] Z. Xu, L. Chen, Y. Dai, and G. Chen. A Dynamic Topic Model and Matrix Factorization-Based Travel Recommendation Method Exploiting Ubiquitous Data. *IEEE Transactions on Multimedia*, 19(8):1933–1945, 2017.

[112] H. Yoon, Y. Zheng, X. Xie, and W. Woo. Social itinerary recommendation from user-generated digital trails. In *Personal and Ubiquitous Computing*, volume 16, pages 469–484, 2012.

[113] 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.

[114] Z. Yu, X. Zhou, Y. Hao, and J. Gu. TV program recommendation for multiple viewers based on user profile merging. *User modeling and user-adapted interaction*, 16(1):63–82, 2006.

[115] C. Zhang, H. Liang, and K. Wang. Trip Recommendation Meets Real-World Constraints. *ACM Transactions on Information Systems*, 35(1):1–28, 2016.

[116] E. Zheng, G. Y. Kondo, S. Zilora, and Q. Yu. Tag-aware dynamic music recommendation. *Expert Systems with Applications*, 106:244–251, sep 2018.

Appendix A

Real World Dataset (Durham, UK)

A.1 Durham Location Information

Table A.1: Durham dataset overview (Name and duration)

ID	Name	Duration
0	Durham Cathedral	120
1	Hall Hill Farm	120
2	Durham Riverside Walk	30
3	Ushaw College	120
4	Palace Green	60
5	Durham Castle	60
6	Finchale Priory	60
7	Marquess of Londonderry's Statue	60
8	St Nicholas Church	60
9	The Durham Light Infantry Memorial	60
10	Neptune Statue	60
11	Durham Town Hall	60

Continued on next page

Table A.1 – continued from previous page

ID	Name	Duration
12	St Laurence Church	90
13	St. John the Evangelist Church	60
14	Escape Rooms Durham	60
15	Durham Climbing Centre	120
16	Infinite Air	120
17	Durham County Cricket Club	180
18	Dragonflies Durham	60
19	Topgear Karting	30
20	Icarus Simulation Limited	30
21	Apollo Bingo	120
22	The Geordie Games	60
23	Durham University Botanic Garden	30
24	Wharton Park	60
25	Low Burnhall Woods	60
26	Old Durham Gardens	60
27	Adventure Valley	180
28	Durham Museum	60
29	Mini Moos Fun Park	60
30	Seaham Beach	120
31	Riverside Park	60
32	Crook Hall and Gardens	120
33	East Durham Coast	60
34	Prince Bishop River Cruiser	60
35	Browns Rowing Boats	60
36	Aquanorth Diving Centre	120
37	Supreme Adventure Sports	60
38	Durham parkrun	60
39	Durham Fencing Centre	60

Continued on next page

Table A.1 – continued from previous page

ID	Name	Duration
40	Wear Canoes	30
41	CrossFit DHM Durham	60
42	Peterlee Parachute Centre	180
43	The Durham Fly Fishing Company	180
44	Poplar Tree Garden Centre and Coffee Shop	30
45	Old Cinema Launderette	120
46	The Durham Brewery Ltd	90
47	Viva Cuba	60
48	Prince Bishops' Golf Course	160
49	Hetton-le-Hill Community Golf Club	160
50	North of England trike tours	120
51	Durham on Foot	60
52	Durham City Walking Tours	60
53	Tin of Sardines	60
54	Old Tom's Gin Bar	60
55	The Dun Cow	60
56	Shakespeare Tavern	60
57	Half Moon Inn	120
58	The Station House	60
59	The Woodman Inn	180
60	The Waiting Room	60
61	Klute	60
62	The Big Jug	60
63	Ustinov College Bar	60
64	Loveshack Nightclub	60
65	Fishtank	120
66	Wiff Waff	60
67	Oriental Museum	120

Continued on next page

Table A.1 – continued from previous page

ID	Name	Duration
68	Lindisfarne Gospels	90
69	Museum of Archaeology	90
70	Durham Market Hall	60
71	Crushed Chilli Gallery	10
72	The Mugwump	10
73	Rpmdiscs - Vinyl Record Store	10
74	The Georgian Window	10
75	People's Bookshop Durham	10
76	The Riverwalk	30
77	The Sweet Shop	10
78	Bannatyne Health Club And Spa - Durham	90
79	Spa at Ramside	60
80	Guru Holistics	30
81	Bonappefeet	30
82	PACE Health Club	60
83	Ritual Beauty	30
84	Gala Theatre Durham	140
85	Durham World Heritage Site Visitor Centre	60
86	Palace Green Library	90
87	Rainton Meadows Nature Reserve and Visitor Centre	120
88	The Surtees Arms And Yard of Ale Brewery	60
89	The Cocktail Project	60
90	Northern Wilderness Bushcraft School - Day Courses	120

Table A.2: Durham dataset overview (Category, Indoor, Parking, and Price)

ID	Category	Indoor	Parking	Price
0	0	1	-1	0
1	0	0	0	8.95
2	0	0	0	0
3	0	0	0	5
4	0	1	-1	0
5	0	1	-1	5
6	0	0	3	5
7	0	1	-1	0
8	0	1	-1	0
9	0	1	-1	0
10	0	0	-1	0
11	0	1	-1	0
12	0	1	-1	0
13	0	1	-1	0
14	1	1	-1	50
15	1	0	0	7.5
16	1	1	0	10
17	1	0	0	12
18	1	1	-1	1
19	1	1	-1	20
20	1	1	-1	0
21	1	1	0	0
22	1	0	-1	20
23	2	0	0	3
24	2	0	1	0
25	2	0	-1	0

Continued on next page

Table A.2 – continued from previous page

ID	Category	Indoor	Parking	Price
26	2	0	0	0
27	0, 1	0	0	9.95
28	3, 0	1	0	0
29	0, 1	0	0	8.5
30	0, 4, 2	0	0	0
31	1, 2	0	0	0
32	2, 0	0	1.4	7.5
33	0, 2, 4	0	0	0
34	4, 5, 6	0	2	5
35	4, 6, 5	0	-1	7
36	4, 5, 6	0	0	15
37	4, 7	0	0	0
38	4, 5	0	0	0
39	4, 7	0	0	10
40	4, 5, 6	0	-1	15
41	4, 8, 7	0	0	0
42	4, 5	0	0	0
43	4, 5, 6	0	0	55
44	2, 9, 10	1	0	15
45	9, 11	1	0	15.40
46	9, 12	1	0	35.45
47	9, 12	1	0	0
48	4	0	0	89
49	4	0	0	15
50	5	0	0	35
51	5	0	0	0
52	5	0	-1	7.5
53	9	1	-1	28

Continued on next page

Table A.2 – continued from previous page

ID	Category	Indoor	Parking	Price
54	9	1	-1	0
55	9	1	-1	0
56	9	1	-1	0
57	9	1	-1	0
58	9	1	-1	0
59	9	1	-1	0
60	9	1	-1	0
61	9	1	-1	0
62	9	1	-1	0
63	9	1	-1	0
64	9	1	-1	0
65	9	1	-1	0
66	9	1	-1	0
67	3	1	0	1.5
68	3	0	0	7.5
69	3	1	0	25
70	10	1	-1	0
71	10	1	-1	0
72	10	1	-1	0
73	10	1	-1	0
74	10	1	-1	0
75	10	1	-1	0
76	10	1	0	0
77	10	1	0	0
78	8	1	0	15
79	8	1	0	119
80	8	1	-1	25
81	8	1	-1	0

Continued on next page

Table A.2 – continued from previous page

ID	Category	Indoor	Parking	Price
82	8	1	-1	10
83	8	1	0	17
84	11	1	0	7.5
85	13	1	-1	0
86	13	1	-1	7.5
87	13	0	0	0
88	12	1	-1	0
89	7	1	-1	5
90	7	0	0	0

Table A.3: Durham dataset overview (Suitable for Children, Baby Care Room, and Wheel Access)

ID	Suitable for Children	Baby Care Room	Wheel Access
0	1	1	1
1	1	1	1
2	0	0	1
3	0	0	1
4	1	1	1
5	1	1	1
6	1	1	0
7	1	1	1
8	1	1	0
9	0	0	1
10	1	1	0
11	0	0	0

Continued on next page

Table A.3 – continued from previous page

ID	Suitable for Children	Baby Care Room	Wheel Access
12	0	0	0
13	0	0	0
14	1	1	1
15	1	1	0
16	1	1	1
17	0	0	0
18	1	1	1
19	1	1	1
20	1	1	1
21	1	1	0
22	1	1	1
23	1	1	1
24	1	1	1
25	1	1	1
26	1	1	1
27	1	1	1
28	1	1	1
29	1	1	1
30	1	1	1
31	1	1	1
32	1	1	1
33	1	1	1
34	1	1	1
35	1	1	1
36	1	1	1
37	1	1	1
38	0	0	1
39	1	1	1

Continued on next page

Table A.3 – continued from previous page

ID	Suitable for Children	Baby Care Room	Wheel Access
40	0	0	1
41	1	1	0
42	1	1	1
43	0	0	1
44	1	1	1
45	0	0	1
46	1	1	0
47	1	1	0
48	0	0	1
49	0	0	1
50	0	0	1
51	1	1	1
52	1	1	1
53	1	1	0
54	1	1	1
55	1	1	1
56	1	1	1
57	1	1	1
58	0	0	0
59	1	1	1
60	1	1	1
61	1	1	0
62	0	0	1
63	1	1	1
64	0	0	1
65	1	1	1
66	0	0	0
67	1	1	1

Continued on next page

Table A.3 – continued from previous page

ID	Suitable for Children	Baby Care Room	Wheel Access
68	1	1	1
69	1	1	1
70	1	1	1
71	1	1	1
72	0	0	1
73	0	0	0
74	0	0	0
75	0	0	0
76	1	1	1
77	1	1	0
78	0	0	1
79	0	0	1
80	0	0	1
81	0	0	0
82	0	0	0
83	0	0	0
84	1	1	1
85	1	1	1
86	1	1	0
87	1	1	1
88	0	0	0
89	1	1	0
90	0	0	1

A.2 Durham Weather Condition

Table A.4: List of the weather conditions codes

Weather Condition	Weather Code
Sunny intervals	1
Sunny	2
Cloudy	3
Windy	4
Heavy rain	5
Light rains	6
Light snow	7
Heavy snow	8
Clear	9

Table A.5: Weather condition for three days in each month (from January to June)

Time	January			February			March			April			May			June		
Day	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
00:00	3	6	8	9	6	8	9	3	5	1	3	5	9	9	3	9	9	3
01:00	3	6	8	9	6	8	9	3	5	1	3	5	9	9	3	9	9	3
02:00	3	6	8	9	6	8	9	3	5	1	3	5	9	9	3	9	9	3
03:00	3	6	8	9	6	8	9	3	5	1	3	5	9	9	3	9	9	3
04:00	3	6	8	9	6	8	9	3	5	1	3	5	9	9	3	9	9	3
05:00	3	6	8	9	6	8	9	3	5	1	3	5	9	9	3	9	9	3
06:00	3	6	8	9	6	8	9	3	6	1	3	6	9	9	3	9	9	3
07:00	9	3	8	2	3	7	2	1	6	1	1	6	1	1	1	2	1	1
08:00	9	3	7	2	3	7	2	1	6	1	1	6	1	1	1	2	1	1
09:00	9	3	7	2	3	7	2	1	6	1	1	6	2	1	1	2	1	1
10:00	9	3	7	2	3	7	2	1	3	1	1	3	2	1	1	2	1	1
11:00	9	3	7	2	3	7	2	1	3	2	1	3	2	1	1	2	1	1
12:00	9	3	7	2	3	7	2	1	3	2	1	3	2	1	1	2	1	1
13:00	9	3	7	2	3	7	2	1	3	2	1	3	2	1	1	2	1	1
14:00	9	3	7	2	3	7	2	1	6	2	1	6	2	1	1	2	1	1
15:00	9	3	7	2	3	7	2	1	6	2	1	6	2	1	1	2	1	1
16:00	9	3	7	2	3	7	2	1	6	2	1	6	2	1	1	2	1	1
17:00	9	3	7	2	3	7	2	1	6	2	1	6	9	1	1	9	1	1
18:00	9	6	8	9	6	8	9	3	5	9	1	5	9	9	3	9	9	3
19:00	9	6	8	9	6	8	9	3	5	9	1	5	9	9	3	9	9	3
20:00	3	6	8	9	6	8	9	3	5	9	3	5	9	9	3	9	9	3
21:00	3	6	8	9	6	8	9	3	5	9	3	5	9	9	3	9	9	3
22:00	3	6	8	9	6	8	9	3	5	9	3	5	9	9	3	9	9	3
23:00	3	6	8	9	6	8	9	3	5	9	3	5	9	9	3	9	9	3

Table A.6: Weather condition for three days in each month (from July to December)

Time	July			August			September			October			November			December		
Day	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
00:00	9	9	3	9	3	6	9	3	6	9	3	6	9	6	6	9	6	6
01:00	9	9	3	9	3	6	9	3	6	9	3	6	9	6	6	9	6	6
02:00	9	9	3	9	3	6	9	3	6	9	3	6	9	6	6	9	6	6
03:00	9	9	3	9	3	6	9	3	6	9	3	6	9	6	6	9	6	6
04:00	9	9	3	9	3	6	9	3	6	9	3	6	9	6	6	9	6	6
05:00	9	9	3	9	3	6	9	3	6	9	3	6	9	6	6	9	6	6
06:00	9	9	3	9	3	6	9	3	6	9	3	6	9	6	6	9	6	6
07:00	2	1	3	2	1	3	2	1	3	2	3	3	1	3	5	1	3	5
08:00	2	1	3	2	1	3	2	1	3	2	3	3	1	3	5	1	3	5
09:00	2	1	3	2	1	3	2	1	3	2	3	3	1	3	5	1	3	5
10:00	2	1	3	2	1	3	2	1	3	2	3	3	1	3	5	1	3	5
11:00	2	1	3	2	1	3	2	1	3	2	3	3	1	3	5	1	3	5
12:00	2	1	3	2	1	3	2	1	3	2	3	3	1	3	5	1	3	5
13:00	2	1	3	2	1	3	2	1	3	2	3	3	1	3	5	1	3	5
14:00	2	1	3	2	1	3	2	1	3	2	3	3	1	3	5	1	3	5
15:00	2	1	3	2	1	3	2	1	3	2	3	3	1	3	5	1	3	5
16:00	2	1	3	2	1	3	2	1	3	2	3	3	1	3	5	1	3	5
17:00	9	1	3	9	1	3	9	1	3	9	3	3	9	6	6	9	6	6
18:00	9	9	3	9	3	6	9	3	6	9	3	6	9	6	6	9	6	6
19:00	9	9	3	9	3	6	9	3	6	9	3	6	9	6	6	9	6	6
20:00	9	9	3	9	3	6	9	3	6	9	3	6	9	6	6	9	6	6
21:00	9	9	3	9	3	6	9	3	6	9	3	6	9	6	6	9	6	6
22:00	9	9	3	9	3	6	9	3	6	9	3	6	9	6	6	9	6	6
23:00	9	9	3	9	3	6	9	3	6	9	3	6	9	6	6	9	6	6

A.3 Durham Time Windows

Table A.7: List of the Time Windows in Durham datasets

ID	Monday		Tuesday		Wednesday		Thursday		Friday		Saturday		Sunday	
	O	C	O	C	O	C	O	C	O	C	O	C	O	C
0	10	18	10	18	10	18	10	18	10	18	10	18	10	17
1	10	17	10	17	10	17	10	17	10	17	10	17	10	17
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	11	14	11	14	11	17	11	17	11	17	-	-	-	-
4	9	17	9	17	9	17	9	17	9	17	-	-	-	-
5	10	16	10	16	10	16	10	16	10	16	10	16	10	16
6	10	17	10	17	10	17	10	17	10	17	10	17	10	17
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	7	19	7	19	7	19	7	19	7	19	7	19	-	-
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	9	17	9	17	9	17	9	17	9	17	10	15	-	-
12	-	-	-	-	-	-	-	-	-	-	14	16	-	-
13	-	-	-	-	-	-	-	-	-	-	10	18	-	-
14	11	22	11	22	11	22	11	22	11	22	11	22	11	22
15	10	22	10	22	10	22	10	22	10	22	9	20	9	20
16	10	19	4	19	10	19	10	19	10	19	9	18	9	18
17	9	17	9	17	9	17	9	17	9	17	-	-	-	-
18	9	18	9	18	9	18	9	18	9	18	10	18	10	18
19	12	20	12	20	12	20	12	20	12	20	9	21	10	18
20	-	-	-	-	-	-	3	18	1	19	10	18	-	-
21	12	22	12	22	12	22	12	22	12	22	11	22	12	22
22	10	16	10	16	10	16	10	16	10	16	10	16	10	16
23	10	17	10	17	10	17	10	17	10	17	10	17	10	17
24	10	17	10	17	10	17	10	17	10	17	10	17	10	17
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Continued on next page

Table A.7 – continued from previous page

ID	Monday		Tuesday		Wednesday		Thursday		Friday		Saturday		Sunday	
	O	C	O	C	O	C	O	C	O	C	O	C	O	C
26	-	-	-	-	-	-	14	16	-	-	-	-	14	16
27	10	16	10	16	10	16	10	16	10	18	10	17	10	17
28	11	16	11	16	11	16	11	16	11	16	11	16	11	16
29	9	20	9	20	9	20	9	20	9	20	9	20	9	20
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	10	18	10	18	10	18	10	18	10	18	10	18	10	18
32	11	17	11	17	10	17	-	-	-	-	-	-	10	17
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	9	16	9	16	9	16	9	16	9	16	9	16	9	16
35	10	18	10	18	10	18	10	18	10	18	10	18	10	18
36	10	17	10	17	10	17	10	17	10	17	10	17	10	17
37	10	17	10	17	10	17	10	17	10	17	10	17	10	17
38	-	-	-	-	-	-	-	-	-	-	9	16	9	16
39	-	-	-	-	-	-	-	-	19	21	-	-	-	-
40	11	17	11	17	11	17	11	17	11	17	10	17	10	17
41	10	20	11	20	10	20	11	20	11	8	10	12	10	11
42	9	17	9	17	9	17	9	17	9	17	8	19	8	19
43	7	16	7	16	7	16	7	16	7	16	7	16	7	16
44	9	17	9	17	9	17	9	17	9	17	9	17	10	16
45	9	17	9	17	9	17	9	17	9	17	9	17	-	-
46	9	16	9	16	9	16	9	16	9	16	10	16	-	-
47	5	11	5	23	5	23	5	23	5	23	2	23	5	11
48	-	-	-	-	9	17	9	17	9	17	9	17	9	17
49	8	18	8	18	8	18	8	18	8	18	8	18	8	18
50	9	18	9	18	9	18	9	18	9	18	9	18	9	18
51	8	12	8	12	8	12	8	12	8	12	8	12	8	12
52	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Continued on next page

Table A.7 – continued from previous page

ID	Monday		Tuesday		Wednesday		Thursday		Friday		Saturday		Sunday	
	O	C	O	C	O	C	O	C	O	C	O	C	O	C
53	12	23	12	23	12	23	12	23	12	23	12	23	12	23
54	12	23	12	23	12	23	12	23	12	0	12	0	12	23
55	11	23	11	23	11	23	11	23	11	23	11	23	12	23
56	11	23	11	23	11	23	11	23	11	23	11	23	11	23
57	11	23	11	23	11	23	11	23	11	0	11	0	12	23
58	16	22	16	22	16	22	16	23	12	23	12	23	14	22
59	-	-	17	23	17	23	17	23	12	0	12	0	12	22
60	11	22	11	22	11	22	11	22	10	22	10	22	10	22
61	22	2	22	2	-	-	22	2	22	2	22	2	22	2
62	15	21	15	21	15	21	15	21	15	21	12	23	15	21
63	18	23	18	23	18	23	18	23	18	23	18	23	18	23
64	8	23	8	23	8	23	8	1	8	1	8	1	8	23
65	-	-	-	-	-	-	-	-	-	-	-	-	-	-
66	-	-	-	-	-	-	17	0	17	1	15	2	-	-
67	10	17	10	17	10	17	10	17	10	17	12	17	13	18
68	9	16	9	16	9	16	9	16	9	16	9	16	9	16
69	12	16	10	16	10	16	10	16	10	16	10	16	10	16
70	9	16	9	16	9	16	9	17	9	17	9	16	-	-
71	-	-	-	-	-	-	10	16	10	16	10	16	-	-
72	10	17	10	17	10	17	10	17	10	17	10	17	10	17
73	11	18	11	18	11	18	11	18	11	16	11	16	11	16
74	10	17	10	17	10	17	10	17	10	17	10	17	12	16
75	-	-	12	17	12	17	12	17	12	17	10	17	-	-
76	9	17	9	17	9	17	9	17	9	17	9	17	11	17
77	9	17	9	17	9	15	9	17	9	17	9	15	9	15
78	6	22	6	22	6	22	6	22	6	22	8	22	8	22
79	6	20	6	20	6	20	6	20	6	20	6	20	6	20

Continued on next page

Table A.7 – continued from previous page

ID	Monday		Tuesday		Wednesday		Thursday		Friday		Saturday		Sunday	
	O	C	O	C	O	C	O	C	O	C	O	C	O	C
80	9	17	9	17	9	20	9	16	9	18	-	-	-	-
81	8	16	8	16	8	16	8	16	8	16	8	16	8	16
82	6	23	6	23	6	23	6	23	6	22	6	22	6	22
83	10	17	9	18	9	18	9	18	9	18	9	15	-	-
84	10	20	10	20	10	20	10	20	10	20	10	20	14	20
85	9	17	9	17	9	17	9	17	9	17	9	17	9	17
86	9	17	9	17	9	17	9	17	9	17	-	-	-	-
87	10	16	10	16	10	16	10	16	10	16	10	16	10	16
88	-	-	16	23	16	23	16	23	16	23	12	0	12	0
89	12	22	12	22	12	22	12	22	12	22	12	22	12	22
90	-	-	9	18	9	18	9	18	9	18	9	18	9	18

A.4 Durham Distance Information

Table A.8: Durham dataset distance between POIs

ID	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	0	20	4	8	6	6	12	3	3	3	270	5	16	13	2	13	10	17	11	13
1	20	0	20	15	23	23	24	21	20	20	21	21	30	22	22	22	25	25	25	27
2	4	20	0	11	4	4	12	2	2	2	2	1	11	9	3	9	6	13	6	8
3	8	15	11	0	13	13	16	11	11	11	11	11	21	9	12	9	15	17	16	17
4	6	23	4	13	0	1	15	3	3	3	3	4	16	12	2	12	10	17	10	12
5	6	23	4	13	1	0	16	3	3	3	3	4	16	13	2	13	10	17	10	12
6	12	24	12	16	15	16	0	12	12	12	12	12	19	16	13	16	16	14	17	18
7	3	21	2	11	3	3	12	0	1	1	0	1	12	8	1	9	7	13	7	8

Continued on next page

Table A.8 – continued from previous page

ID	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
8	3	20	2	11	3	3	12	1	0	1	1	1	12	8	1	9	7	13	7	8
9	3	20	2	11	3	3	12	1	1	0	1	1	12	8	1	8	7	13	7	9
10	270	21	2	11	3	3	12	0	1	1	0	1	12	8	1	9	7	14	7	9
11	5	21	1	11	4	4	12	1	1	1	1	0	14	12	6	12	9	16	9	10
12	16	30	11	21	16	16	19	12	12	12	12	14	0	17	13	17	9	16	10	10
13	13	22	9	9	12	13	16	8	8	8	8	12	17	0	9	1	12	15	13	14
14	2	22	3	12	2	2	13	1	1	1	1	6	13	9	0	10	8	14	8	9
15	13	22	9	9	12	13	16	9	9	8	9	12	17	1	10	0	13	18	14	16
16	10	25	6	15	10	10	16	7	7	7	7	9	9	12	8	13	0	16	4	6
17	17	25	13	17	17	17	14	13	13	13	14	16	16	15	14	18	16	0	16	18
18	11	25	6	16	10	10	17	7	7	7	7	9	10	13	8	14	4	16	0	6
19	13	27	8	17	12	12	18	8	8	9	9	10	10	14	9	16	6	18	6	0
20	14	15	11	6	14	14	15	10	10	10	11	15	20	12	12	14	16	16	16	18
21	8	23	5	14	8	8	15	5	5	5	5	7	7	11	6	12	7	18	6	5
22	17	15	13	12	17	17	17	13	13	13	13	17	23	15	14	16	18	16	20	20
23	11	25	5	12	11	10	17	8	8	8	8	12	13	8	8	10	12	20	12	14
24	6	21	3	12	6	6	12	3	3	3	3	7	13	8	4	9	8	14	8	10
25	11	22	5	9	11	10	14	7	7	7	7	11	13	5	8	6	12	16	12	14
26	14	29	10	22	15	14	21	11	10	11	11	13	11	17	12	18	12	22	12	10
27	15	23	12	15	15	15	5	11	11	11	12	15	18	14	12	16	16	14	18	18
28	2	23	4	13	2	2	14	2	2	2	2	7	14	10	1	12	9	18	9	12
29	17	15	13	14	17	16	15	13	13	13	13	17	22	15	13	18	18	12	18	20
30	26	41	22	32	26	26	29	22	22	22	22	25	20	28	24	30	22	24	22	22
31	16	24	21	30	16	15	13	22	21	22	22	24	20	27	23	30	22	20	22	16
32	7	21	4	12	7	7	13	4	4	4	4	8	13	9	5	10	9	16	9	10
33	28	43	24	34	28	28	31	24	23	24	24	26	19	29	25	35	24	26	26	26
34	4	21	1	11	4	4	12	2	2	2	2	6	11	8	2	9	6	14	6	7
35	3	22	4	13	3	3	14	1	2	2	1	6	13	9	1	12	7	16	8	8

Continued on next page

Table A.8 – continued from previous page

ID	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
36	13	17	9	10	12	12	12	9	9	9	9	13	18	11	9	14	14	10	14	12
37	11	25	7	16	11	11	17	7	7	8	8	10	10	13	9	16	4	16	4	7
38	9	24	3	12	8	8	15	6	6	6	6	9	10	8	6	8	9	18	9	10
39	9	44	3	12	8	8	15	6	6	6	6	9	10	8	6	9	10	18	9	9
40	8	24	4	14	8	8	14	5	5	5	5	9	13	10	6	12	9	18	8	10
41	12	27	7	17	12	12	18	8	8	8	8	10	9	14	9	16	6	16	6	1
42	23	37	18	28	23	23	28	18	18	19	19	21	13	23	20	26	20	26	20	18
43	25	40	18	28	25	25	30	21	21	21	21	23	17	23	22	24	20	28	20	18
44	11	26	5	15	10	10	17	8	8	8	8	11	11	10	8	12	12	20	12	10
45	8	23	4	14	8	8	15	5	5	5	5	7	9	11	6	12	5	16	5	4
46	13	29	7	17	13	12	20	10	10	10	10	14	12	13	10	14	12	18	14	14
47	2	22	3	12	2	2	13	1	1	1	1	5	12	9	1	10	7	16	7	9
48	11	26	7	16	11	10	16	7	7	7	7	9	7	13	8	14	6	14	7	7
49	17	31	12	22	17	16	20	13	13	13	13	15	6	19	14	20	12	18	12	12
50	32	15	29	21	32	32	34	28	28	28	28	32	36	19	29	20	35	40	35	35
51	6	21	2	11	6	5	12	3	3	3	3	7	11	8	4	9	6	14	6	8
52	4	20	2	10	4	4	12	1	1	1	1	4	12	8	2	9	6	16	7	8
53	3	22	3	12	3	3	13	1	1	1	1	6	13	9	1	10	7	16	8	8
54	6	20	3	11	6	6	12	3	3	3	3	7	12	8	4	9	8	16	8	12
55	28	43	2	12	29	28	13	4	3	4	4	7	12	9	4	10	7	16	7	8
56	2	22	3	12	2	2	13	1	1	1	1	5	13	9	1	10	8	16	8	9
57	5	21	1	11	5	5	12	3	3	3	3	6	11	8	3	9	6	14	6	7
58	6	19	3	9	6	5	11	3	3	3	3	6	12	6	4	7	7	14	8	8
59	5	21	2	12	6	5	12	2	2	2	2	4	10	9	4	10	5	16	6	6
60	7	21	4	11	8	7	12	4	4	4	4	8	14	8	5	9	9	16	9	10
61	3	22	3	12	3	3	13	1	1	1	1	6	13	9	1	10	7	16	8	8
62	4	20	3	11	4	4	12	1	1	1	1	4	12	8	2	9	6	14	7	7
63	1	22	4	13	1	1	14	2	2	2	2	6	13	10	1	12	8	16	8	10

Continued on next page

Table A.8 – continued from previous page

ID	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
64	35	22	4	13	3	3	14	2	2	2	2	6	13	10	1	10	8	16	8	8
65	7	21	4	11	7	6	13	4	4	4	4	7	13	7	5	9	9	16	9	12
66	5	22	2	12	5	5	13	1	1	2	1	1	12	9	3	10	7	16	7	10
67	10	23	3	10	10	10	14	7	7	7	7	11	13	6	8	7	12	16	12	14
68	7	23	4	13	0	1	14	2	2	2	2	7	14	10	1	12	10	16	9	12
69	1	23	5	14	1	1	15	2	3	2	2	7	14	10	1	12	9	16	9	12
70	3	21	2	11	3	3	12	1	1	1	1	1	12	10	1	9	7	16	7	8
71	10	24	5	11	10	9	15	6	6	6	6	10	14	8	7	9	10	16	10	12
72	2	22	3	12	2	2	13	1	1	1	1	5	13	9	1	10	7	16	8	9
73	22	38	17	26	22	22	29	19	19	19	19	22	16	21	19	24	20	30	20	18
74	1	22	4	13	1	1	14	1	2	1	1	6	13	9	1	10	8	16	9	9
75	2	22	3	12	2	2	13	1	1	1	1	5	12	9	1	10	7	16	8	9
76	5	20	3	10	6	5	11	2	2	2	2	6	12	7	1	9	7	14	8	9
77	3	21	2	11	3	3	12	1	1	1	1	5	12	8	1	9	7	14	7	8
78	10	25	6	15	10	10	16	6	7	7	7	9	8	13	8	14	4	14	5	4
79	10	26	7	16	11	11	16	7	7	8	8	9	7	13	8	14	7	14	7	7
80	7	22	3	12	6	6	13	4	4	4	4	6	9	10	5	10	6	14	6	5
81	6	20	3	10	6	5	11	2	2	2	2	6	12	7	3	9	7	14	8	9
82	6	21	3	11	6	6	12	3	3	3	3	7	13	8	4	9	8	14	8	9
83	17	27	13	18	17	17	13	13	13	14	14	15	11	17	14	18	14	5	14	14
84	4	20	2	10	4	3	12	1	1	1	1	4	12	8	2	9	7	14	7	9
85	1	23	4	13	1	1	14	2	2	2	2	6	13	10	1	12	8	18	9	10
86	1	23	5	14	0	1	15	2	2	2	2	7	14	10	1	12	10	16	10	12
87	16	31	12	22	17	17	18	13	13	13	13	15	11	19	14	20	14	12	14	14
88	16	35	17	22	22	22	26	18	18	19	18	20	18	14	19	16	18	24	20	20
89	28	35	24	12	5	5	13	1	1	2	1	1	12	9	3	14	9	16	9	10
90	16	24	13	16	16	15	1	12	12	12	12	16	18	14	13	16	18	16	18	20

Table A.9: Durham dataset distance between POIs

ID	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
0	14	8	17	11	6	11	14	15	2	17	26	16	7	28	4	3	13	11	9	9
1	15	23	15	25	21	22	29	23	23	15	41	24	21	43	21	22	17	25	24	44
2	11	5	13	5	3	5	10	12	4	13	22	21	4	24	1	4	9	7	3	3
3	6	14	12	12	12	9	22	15	13	14	32	30	12	34	11	13	10	16	12	12
4	14	8	17	11	6	11	15	15	2	17	26	16	7	28	4	3	12	11	8	8
5	14	8	17	10	6	10	14	15	2	16	26	15	7	28	4	3	12	11	8	8
6	15	15	17	17	12	14	21	5	14	15	29	13	13	31	12	14	12	17	15	15
7	10	5	13	8	3	7	11	11	2	13	22	22	4	24	2	1	9	7	6	6
8	10	5	13	8	3	7	10	11	2	13	22	21	4	23	2	2	9	7	6	6
9	10	5	13	8	3	7	11	11	2	13	22	22	4	24	2	2	9	8	6	6
10	11	5	13	8	3	7	11	12	2	13	22	22	4	24	2	1	9	8	6	6
11	15	7	17	12	7	11	13	15	7	17	25	24	8	26	6	6	13	10	9	9
12	20	7	23	13	13	13	11	18	14	22	20	20	13	19	11	13	18	10	10	10
13	12	11	15	8	8	5	17	14	10	15	28	27	9	29	8	9	11	13	8	8
14	12	6	14	8	4	8	12	12	1	13	24	23	5	25	2	1	9	9	6	6
15	14	12	16	10	9	6	18	16	12	18	30	30	10	35	9	12	14	16	9	9
16	16	7	18	12	8	12	12	16	9	18	22	22	9	24	6	7	14	4	10	10
17	16	18	16	20	14	16	22	14	18	12	24	20	16	26	14	16	10	16	18	18
18	16	6	20	12	8	12	12	18	9	18	22	22	9	26	6	8	14	4	9	9
19	18	5	20	14	10	14	10	18	12	20	22	16	10	26	7	8	16	7	10	9
20	0	14	9	16	10	14	22	16	14	12	35	16	12	35	12	12	8	18	16	16
21	14	0	16	10	7	9	7	14	7	16	22	16	7	24	4	5	12	7	7	7
22	9	16	0	18	14	16	22	16	16	9	35	35	14	40	14	14	8	18	18	18
23	16	10	20	0	9	4	16	16	10	18	28	18	9	28	6	7	14	12	6	6
24	10	7	12	9	0	8	14	12	6	14	24	14	4	26	3	4	9	9	7	7
25	14	9	16	4	8	0	16	14	10	16	28	14	8	28	6	7	12	12	6	6
26	22	7	24	16	14	16	0	20	14	22	28	22	12	28	9	10	18	12	10	10

Continued on next page

Table A.9 – continued from previous page

ID	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
27	16	14	18	16	12	14	20	0	14	16	28	12	14	30	12	12	12	18	16	16
28	14	7	18	10	6	10	14	14	0	16	24	16	6	28	4	5	12	10	8	8
29	12	16	9	18	14	16	22	16	16	0	30	10	14	35	12	12	12	20	16	16
30	35	22	35	28	24	28	28	28	24	30	0	22	24	14	20	24	28	20	26	30
31	16	16	14	18	14	14	22	12	16	10	22	0	12	26	12	14	8	14	14	14
32	12	7	16	9	4	8	12	14	6	14	24	12	0	26	4	6	10	9	10	10
33	35	24	40	28	26	28	28	30	28	35	14	26	26	0	22	26	30	24	16	16
34	12	4	14	6	3	6	9	12	4	12	20	12	4	22	0	3	8	6	14	14
35	12	5	14	7	4	7	10	12	5	12	24	14	6	26	3	0	9	7	8	8
36	8	12	8	14	9	12	18	12	12	12	28	8	10	30	8	9	0	14	14	35
37	18	7	20	12	9	12	12	18	10	20	20	14	9	24	6	7	14	0	40	40
38	16	7	18	6	7	6	10	16	8	16	26	14	10	16	14	8	14	40	0	14
39	16	7	18	6	7	6	10	16	8	16	30	14	10	16	14	8	35	40	14	0
40	14	7	18	7	6	7	12	14	7	14	22	14	6	24	3	4	12	9	14	14
41	18	5	20	14	10	14	9	18	12	20	22	14	10	22	7	8	14	8	16	16
42	28	16	35	22	20	22	18	28	22	28	14	24	20	12	18	18	26	20	14	14
43	30	16	35	20	22	20	20	30	22	35	22	23	27	26	25	20	19	14	14	14
44	18	7	20	7	9	7	12	18	10	9	24	18	9	22	5	7	14	12	12	14
45	14	2	18	9	6	9	7	16	7	16	20	14	6	20	5	4	12	7	14	12
46	20	10	22	12	12	12	14	22	14	12	24	16	12	24	8	9	16	14	12	14
47	12	5	16	8	4	8	12	14	2	14	22	12	6	24	3	4	10	9	16	12
48	18	7	20	12	8	12	12	16	9	20	16	10	9	20	5	6	14	8	14	16
49	24	10	26	18	16	18	14	20	16	26	14	16	14	16	12	12	20	14	12	14
50	24	30	26	28	28	24	40	35	35	35	50	35	30	50	28	28	30	35	16	12
51	12	5	14	6	4	6	10	12	5	12	20	12	4	24	1	3	9	7	14	16
52	12	5	14	8	4	8	10	12	3	12	22	12	4	24	2	3	8	8	16	14
53	12	5	14	7	4	8	10	12	5	12	22	14	6	26	3	4	10	9	16	16
54	14	8	16	8	5	6	14	14	7	16	22	12	3	24	3	4	9	8	16	16

Continued on next page

Table A.9 – continued from previous page

ID	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
55	12	5	16	6	4	6	10	14	5	14	24	14	5	26	26	4	10	8	90	16
56	12	6	16	8	4	9	12	14	2	14	22	22	5	24	2	2	10	8	16	90
57	12	4	14	5	3	6	10	12	4	12	25	20	4	22	1	4	9	6	14	16
58	10	5	12	7	3	6	10	10	5	10	0	0	0	0	0	0	0	0	16	14
59	12	3	16	7	4	8	9	14	5	14	22	14	5	24	2	3	10	6	14	16
60	12	7	14	9	5	8	14	12	6	12	30	40	6	26	4	35	12	10	16	14
61	12	5	14	7	4	8	10	12	5	12	22	14	6	26	3	4	10	9	14	16
62	12	4	16	8	4	8	10	14	4	14	22	12	5	24	3	4	9	8	14	14
63	14	7	16	9	5	10	12	14	1	16	24	14	7	26	4	5	10	9	7	14
64	12	5	14	7	4	8	10	12	5	12	24	22	5	26	1	10	10	7	7	7
65	14	8	16	7	5	6	14	14	7	14	26	14	6	28	5	6	12	10	90	7
66	16	7	18	12	7	12	14	16	7	16	22	12	5	24	2	3	10	7	6	90
67	16	9	16	4	7	3	14	16	8	16	26	16	9	24	5	6	14	12	5	6
68	14	7	18	10	6	10	14	16	2	16	85	75	85	90	80	85	85	85	90	5
69	14	8	18	10	6	10	14	16	3	16	24	14	7	26	4	5	12	10	8	90
70	12	5	14	8	4	8	10	12	3	12	20	12	5	24	2	3	9	8	6	8
71	14	8	16	8	5	7	14	14	7	14	26	16	8	26	6	7	12	12	6	6
72	12	6	16	8	4	9	12	14	2	14	22	14	6	24	3	4	10	9	7	6
73	30	16	35	20	20	20	18	30	22	30	16	26	20	14	16	18	26	20	16	7
74	14	6	16	9	5	9	12	14	1	14	22	14	6	26	3	4	10	9	7	16
75	12	5	16	8	4	8	12	14	2	14	22	12	6	24	3	4	10	8	6	7
76	12	5	14	8	3	7	12	12	5	12	22	12	3	24	2	3	8	8	6	6
77	12	5	14	8	4	8	10	12	3	14	16	12	5	24	2	3	9	8	6	6
78	16	4	20	12	8	12	8	18	9	18	20	12	8	22	5	6	12	6	9	6
79	18	7	20	12	9	12	12	16	9	16	16	10	9	20	6	6	14	8	12	9
80	14	2	16	8	5	8	7	14	5	14	20	12	5	22	3	4	10	6	7	12
81	12	5	14	8	3	7	12	12	5	14	22	12	3	24	2	3	8	8	6	7
82	12	6	14	9	4	8	12	12	5	14	22	12	1	24	3	4	9	8	7	6

Continued on next page

Table A.9 – continued from previous page

ID	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
83	18	16	18	20	16	18	20	12	18	12	20	5	14	24	12	12	14	14	18	7
84	12	5	14	8	4	8	10	12	3	14	22	12	4	24	2	3	8	8	6	18
85	14	7	16	9	5	10	14	14	2	16	24	14	7	26	4	5	10	9	7	6
86	14	7	18	10	6	10	14	16	2	16	24	14	7	26	4	5	12	10	8	7
87	24	14	24	20	16	20	20	18	16	20	16	12	7	20	4	5	10	14	7	8
88	28	18	28	18	22	14	24	28	22	30	30	22	20	35	18	18	24	20	18	18
89	16	7	18	12	7	12	12	16	6	7	24	16	28	26	22	26	22	24	6	6
90	16	16	18	18	12	16	22	6	16	6	30	12	14	35	12	12	12	18	18	18

Table A.10: Durham dataset distance between POIs

ID	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
0	8	12	23	25	11	8	13	2	11	17	32	6	4	3	6	28	2	5	6	5
1	24	27	37	40	26	23	29	22	26	31	15	21	20	22	20	43	22	21	19	21
2	4	7	18	18	5	4	7	3	7	12	29	2	2	3	3	2	3	1	3	2
3	14	17	28	28	15	14	17	12	16	22	21	11	10	12	11	12	12	11	9	12
4	8	12	23	25	10	8	13	2	11	17	32	6	4	3	6	29	2	5	6	6
5	8	12	23	25	10	8	12	2	10	16	32	5	4	3	6	28	2	5	5	5
6	14	18	28	30	17	15	20	13	16	20	34	12	12	13	12	13	13	12	11	12
7	5	8	18	21	8	5	10	1	7	13	28	3	1	1	3	4	1	3	3	2
8	5	8	18	21	8	5	10	1	7	13	28	3	1	1	3	3	1	3	3	2
9	5	8	19	21	8	5	10	1	7	13	28	3	1	1	3	4	1	3	3	2
10	5	8	19	21	8	5	10	1	7	13	28	3	1	1	3	4	1	3	3	2
11	9	10	21	23	11	7	14	5	9	15	32	7	4	6	7	7	5	6	6	4
12	13	9	13	17	11	9	12	12	7	6	36	11	12	13	12	12	13	11	12	10
13	10	14	23	23	10	11	13	9	13	19	19	8	8	9	8	9	9	8	6	9
14	6	9	20	22	8	6	10	1	8	14	29	4	2	1	4	4	1	3	4	4

Continued on next page

Table A.10 – continued from previous page

ID	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
15	12	16	26	24	12	12	14	10	14	20	20	9	9	10	9	10	10	9	7	10
16	9	6	20	20	12	5	12	7	6	12	35	6	6	7	8	7	8	6	7	5
17	18	16	26	28	20	16	18	16	14	18	40	14	16	16	16	16	16	14	14	16
18	8	6	20	20	12	5	14	7	7	12	35	6	7	8	8	7	8	6	8	6
19	10	1	18	18	10	4	14	9	7	12	35	8	8	8	12	8	9	7	8	6
20	14	18	28	30	18	14	20	12	18	24	24	12	12	12	14	12	12	12	10	12
21	7	5	16	16	7	2	10	5	7	10	30	5	5	5	8	5	6	4	5	3
22	16	20	30	35	20	16	22	14	18	24	26	14	14	14	14	14	14	14	12	14
23	7	14	22	20	7	9	12	8	12	18	28	6	8	7	8	6	8	5	7	7
24	6	10	20	22	9	6	12	4	8	16	28	4	4	4	5	4	4	3	3	4
25	7	14	22	20	7	9	12	8	12	18	24	6	8	8	6	6	9	6	6	8
26	12	9	18	20	12	7	14	12	12	14	40	10	10	10	14	10	12	10	10	9
27	14	18	28	30	18	16	22	14	16	20	35	12	12	12	14	14	14	12	10	14
28	7	12	22	22	10	7	14	2	9	16	35	5	3	5	7	5	2	4	5	5
29	14	20	28	35	9	16	12	14	20	26	35	12	12	12	16	14	14	12	10	14
30	22	22	14	22	24	20	24	22	16	14	50	20	22	22	22	24	22	25	0	22
31	14	14	24	23	18	14	16	12	10	16	35	12	12	14	12	14	22	20	0	14
32	6	10	20	27	9	6	12	6	9	14	30	4	4	6	3	5	5	4	0	5
33	24	22	12	26	22	20	24	24	20	16	50	24	24	26	24	26	24	22	0	24
34	3	7	18	25	5	5	8	3	5	12	28	1	2	3	3	26	2	1	0	2
35	4	8	18	20	7	4	9	4	6	12	28	3	3	4	4	2	2	4	0	3
36	12	14	26	19	14	12	16	10	14	20	30	9	8	10	9	4	10	9	0	10
37	9	8	20	14	12	7	14	9	8	14	35	7	8	9	8	10	8	6	0	6
38	14	16	14	14	12	14	12	16	14	12	16	14	16	16	16	90	16	14	16	14
39	14	16	14	14	12	14	12	16	14	12	16	14	16	16	16	90	16	14	16	14
40	0	10	20	30	7	6	10	6	9	14	30	3	5	7	5	2	7	5	0	5
41	10	0	16	32	12	5	12	9	7	12	35	8	9	10	9	8	9	7	0	7
42	20	16	0	22	18	16	20	20	18	12	45	18	20	20	20	20	20	18	0	18

Continued on next page

Table A.10 – continued from previous page

ID	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
43	30	32	22	0	24	25	28	20	20	16	20	16	23	20	26	27	28	23	21	25
44	7	12	18	24	0	8	7	8	12	14	30	7	7	10	8	6	9	5	0	5
45	6	5	16	25	8	0	12	5	6	10	30	5	5	5	8	5	5	4	5	3
46	10	12	20	28	7	12	0	10	12	16	35	8	10	10	12	8	12	8	10	12
47	6	9	20	20	8	5	10	0	8	14	28	3	2	3	6	4	1	3	3	3
48	9	7	18	20	12	6	12	8	0	8	35	7	7	7	10	7	8	7	7	5
49	14	12	12	16	14	10	16	14	8	0	40	14	14	14	16	14	14	12	14	12
50	30	35	45	20	30	30	35	28	35	40	0	30	30	30	28	30	30	30	28	30
51	3	8	18	16	7	5	8	3	7	14	30	0	4	3	6	1	4	2	3	3
52	5	9	20	23	7	5	10	2	7	14	30	4	0	3	5	3	2	2	2	2
53	7	10	20	20	10	5	10	3	7	14	30	3	3	0	6	4	1	3	3	3
54	5	9	20	26	8	8	12	6	10	16	28	6	5	6	0	4	4	3	3	4
55	2	8	20	27	6	5	8	4	7	14	30	1	3	4	4	0	4	2	4	3
56	7	9	20	28	9	5	12	1	8	14	30	4	2	1	4	4	0	4	4	4
57	5	7	18	23	5	4	8	3	7	12	30	2	2	3	3	2	4	0	3	3
58	0	0	0	21	0	5	10	3	7	14	28	3	2	3	3	4	4	3	0	4
59	5	7	18	25	5	3	12	3	5	12	30	3	2	3	4	3	4	3	4	0
60	6	10	26	12	9	7	35	5	9	16	30	5	4	5	5	5	5	5	2	5
61	7	10	20	13	10	5	10	85	7	14	30	3	3	4	4	4	1	3	4	4
62	6	8	20	19	7	4	12	3	6	12	30	4	1	3	4	4	3	4	3	2
63	7	10	22	25	10	7	12	2	9	16	35	5	3	3	5	5	1	5	4	4
64	5	8	20	30	7	5	10	3	7	14	30	3	3	4	4	4	1	3	4	4
65	8	12	22	28	9	8	12	6	10	16	28	5	5	6	5	5	4	5	3	6
66	5	8	18	15	7	7	14	5	90	16	35	7	5	6	7	3	3	6	4	3
67	6	12	20	10	6	9	9	7	12	16	28	6	7	8	7	5	8	5	5	7
68	85	85	90	18	90	80	90	90	80	90	110	80	85	90	85	80	85	80	85	80
69	8	12	22	16	12	8	14	3	10	16	35	5	4	3	6	5	2	5	5	5
70	5	8	20	20	7	5	10	1	7	14	35	3	1	2	3	3	1	3	3	3

Continued on next page

Table A.10 – continued from previous page

ID	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
71	7	12	22	22	7	8	12	6	10	16	28	6	5	6	5	6	7	6	5	7
72	6	10	20	25	8	6	12	1	8	14	30	4	2	1	4	4	1	3	4	3
73	18	16	10	15	16	16	18	20	20	16	45	18	20	20	20	20	20	18	20	18
74	7	10	20	16	10	6	12	1	8	14	30	4	2	2	5	4	1	4	4	4
75	6	9	20	30	8	5	10	1	8	14	30	3	1	1	4	4	1	3	3	3
76	5	9	20	15	7	5	10	4	8	14	28	3	3	4	1	4	4	3	2	4
77	5	9	9	22	7	5	14	1	7	14	24	3	1	2	4	3	1	3	3	3
78	8	4	18	26	10	3	12	8	6	9	35	7	7	8	8	7	7	6	7	5
79	9	7	18	24	12	6	12	8	1	10	35	7	8	8	8	7	8	7	7	5
80	5	6	16	27	8	2	10	4	6	12	30	3	4	5	5	4	4	3	4	2
81	5	9	20	25	7	5	10	4	8	14	28	3	3	4	1	4	4	3	2	4
82	5	9	20	20	8	6	12	4	8	14	30	3	3	5	2	4	4	4	3	4
83	14	16	22	18	18	14	18	16	9	14	40	14	16	16	16	14	14	14	14	12
84	5	9	20	19	7	5	10	2	7	14	30	3	1	2	4	3	2	3	2	2
85	7	10	20	20	10	7	12	2	9	16	35	5	3	3	5	5	1	5	4	4
86	8	10	22	13	12	8	14	3	10	16	35	5	4	3	6	5	2	5	5	5
87	7	10	18	19	10	40	18	16	9	12	40	14	14	16	16	14	14	14	4	12
88	18	20	26	20	16	24	14	20	18	22	28	20	20	20	20	18	20	20	20	18
89	26	24	35	25	30	24	26	24	20	28	45	24	24	24	24	24	26	24	24	22
90	16	20	30	23	18	16	20	14	16	20	35	12	12	14	12	14	14	12	12	14

Table A.11: Durham dataset distance between POIs

ID	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
0	7	3	4	1	35	7	5	10	7	1	3	10	2	22	1	2	5	3	10	10
1	21	22	20	22	22	21	22	23	23	23	21	24	22	38	22	22	20	21	25	26
2	4	3	3	4	4	4	2	3	4	5	2	5	3	17	4	3	3	2	6	7

Continued on next page

Table A.11 – continued from previous page

ID	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
3	11	12	11	13	13	11	12	10	13	14	11	11	12	26	13	12	10	11	15	16
4	8	3	4	1	3	7	5	10	0	1	3	10	2	22	1	2	6	3	10	11
5	7	3	4	1	3	6	5	10	1	1	3	9	2	22	1	2	5	3	10	11
6	12	13	12	14	14	13	13	14	14	15	12	15	13	29	14	13	11	12	16	16
7	4	1	1	2	2	4	1	7	2	2	1	6	1	19	1	1	2	1	6	7
8	4	1	1	2	2	4	1	7	2	3	1	6	1	19	2	1	2	1	7	7
9	4	1	1	2	2	4	2	7	2	2	1	6	1	19	1	1	2	1	7	8
10	4	1	1	2	2	4	1	7	2	2	1	6	1	19	1	1	2	1	7	8
11	8	6	4	6	6	7	1	11	7	7	1	10	5	22	6	5	6	5	9	9
12	14	13	12	13	13	13	12	13	14	14	12	14	13	16	13	12	12	12	8	7
13	8	9	8	10	10	7	9	6	10	10	10	8	9	21	9	9	7	8	13	13
14	5	1	2	1	1	5	3	8	1	1	1	7	1	19	1	1	1	1	8	8
15	9	10	9	12	10	9	10	7	12	12	9	9	10	24	10	10	9	9	14	14
16	9	7	6	8	8	9	7	12	10	9	7	10	7	20	8	7	7	7	4	7
17	16	16	14	16	16	16	16	16	16	16	16	16	16	30	16	16	14	14	14	14
18	9	8	7	8	8	9	7	12	9	9	7	10	8	20	9	8	8	7	5	7
19	10	8	7	10	8	12	10	14	12	12	8	12	9	18	9	9	9	8	4	7
20	12	12	12	14	12	14	16	16	14	14	12	14	12	30	14	12	12	12	16	18
21	7	5	4	7	5	8	7	9	7	8	5	8	6	16	6	5	5	5	4	7
22	14	14	14	16	16	14	14	16	16	16	14	16	14	35	14	14	14	14	18	18
23	9	7	8	9	7	7	12	4	10	10	8	8	8	20	9	8	8	8	12	12
24	5	4	4	5	4	5	7	7	6	6	4	5	4	20	5	4	3	4	8	9
25	8	8	8	10	8	6	12	3	10	10	8	7	9	20	9	8	7	8	12	12
26	14	10	10	12	10	14	14	14	14	14	10	14	12	18	12	12	12	10	8	12
27	12	12	14	14	12	14	16	16	16	16	12	14	14	30	14	14	12	12	18	16
28	6	5	4	1	5	7	7	8	2	3	3	7	2	22	1	2	5	3	9	9
29	12	12	14	16	12	14	16	16	16	16	12	14	14	30	14	14	12	14	18	16
30	30	22	22	24	24	26	22	26	85	24	20	26	22	16	22	22	22	16	20	16

Continued on next page

Table A.11 – continued from previous page

ID	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
31	40	14	12	14	22	14	12	16	75	14	12	16	14	26	14	12	12	12	12	10
32	6	6	5	7	5	6	5	9	85	7	5	8	6	20	6	6	3	5	8	9
33	26	26	24	26	26	28	24	24	90	26	24	26	24	14	26	24	24	24	22	20
34	4	3	3	4	1	5	2	5	80	4	2	6	3	16	3	3	2	2	5	6
35	35	4	4	5	10	6	3	6	85	5	3	7	4	18	4	4	3	3	6	6
36	12	10	9	10	10	12	10	14	85	12	9	12	10	26	10	10	8	9	12	14
37	10	9	8	9	7	10	7	12	85	10	8	12	9	20	9	8	8	8	6	8
38	16	14	14	7	7	90	6	5	90	8	6	6	7	16	7	6	6	6	9	12
39	16	14	14	7	7	90	6	5	90	8	6	6	7	16	7	6	6	6	9	12
40	6	7	6	7	5	8	5	6	85	8	5	7	6	18	7	6	5	5	8	9
41	10	10	8	10	8	12	8	12	85	12	8	12	10	16	10	9	9	9	4	7
42	26	20	20	22	20	22	18	20	90	22	20	22	20	10	20	20	20	9	18	18
43	12	13	19	25	30	28	15	10	18	16	20	22	25	15	16	30	15	22	26	24
44	9	10	7	10	7	9	7	6	90	12	7	7	8	16	10	8	7	7	10	12
45	7	5	4	7	5	8	7	9	80	8	5	8	6	16	6	5	5	5	3	6
46	35	10	12	12	10	12	14	9	90	14	10	12	12	18	12	10	10	14	12	12
47	5	85	3	2	3	6	5	7	90	3	1	6	1	20	1	1	4	1	8	8
48	9	7	6	9	7	10	90	12	80	10	7	10	8	20	8	8	8	7	6	1
49	16	14	12	16	14	16	16	16	90	16	14	16	14	16	14	14	14	14	9	10
50	30	30	30	35	30	28	35	28	110	35	35	28	30	45	30	30	28	24	35	35
51	5	3	4	5	3	5	7	6	80	5	3	6	4	18	4	3	3	3	7	7
52	4	3	1	3	3	5	5	7	85	4	1	5	2	20	2	1	3	1	7	8
53	5	4	3	3	4	6	6	8	90	3	2	6	1	20	2	1	4	2	8	8
54	5	4	4	5	4	5	7	7	85	6	3	5	4	20	5	4	1	4	8	8
55	5	4	4	5	4	5	3	5	80	5	3	6	4	20	4	4	4	3	7	7
56	5	1	3	1	1	4	3	8	85	2	1	7	1	1	1	1	4	1	7	8
57	5	3	4	5	3	5	6	5	80	5	3	6	3	18	4	3	3	3	6	7
58	2	4	3	4	4	3	4	5	85	5	3	5	4	20	4	3	2	3	7	7

Continued on next page

Table A.11 – continued from previous page

ID	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
59	5	4	2	4	4	5	3	7	80	5	3	7	3	18	4	3	4	3	5	5
60	0	5	5	6	5	4	5	7	90	7	5	6	5	22	6	5	4	5	9	9
61	5	0	3	4	280	6	3	7	85	5	3	7	4	20	4	3	3	3	7	7
62	5	3	0	3	3	5	3	7	85	4	1	7	2	20	3	2	3	1	6	6
63	6	4	3	0	4	7	4	9	85	2	2	8	2	22	1	2	5	2	8	9
64	5	280	3	4	0	5	3	7	80	5	3	7	4	20	4	3	3	3	7	7
65	5	6	5	7	5	0	6	6	90	7	5	5	6	22	6	6	5	5	10	10
66	5	3	3	4	3	6	0	12	85	7	5	10	6	22	6	5	7	5	9	9
67	7	7	7	9	7	6	12	0	90	8	6	5	7	20	7	7	6	6	10	12
68	90	85	85	85	80	90	85	90	0	90	85	90	90	100	90	90	90	100	85	85
69	7	5	4	2	5	7	7	8	90	0	5	9	3	22	2	3	5	3	9	10
70	5	3	1	2	3	5	5	6	85	5	0	10	5	22	6	5	6	5	9	9
71	6	7	7	8	7	5	10	5	90	9	10	0	6	22	7	6	5	5	10	10
72	5	4	2	2	4	6	6	7	90	3	5	6	0	20	1	1	4	1	7	8
73	22	20	20	22	20	22	22	20	100	22	22	22	20	0	22	20	20	7	18	20
74	6	4	3	1	4	6	6	7	90	2	6	7	1	22	0	1	4	2	8	8
75	5	3	2	2	3	6	5	7	90	3	5	6	1	20	1	0	3	1	7	7
76	4	3	3	5	3	5	7	6	90	5	6	5	4	20	4	3	0	16	7	8
77	5	3	1	2	3	5	5	6	100	3	5	5	1	7	2	1	16	0	18	18
78	9	7	6	8	7	10	9	10	85	9	9	10	7	18	8	7	7	18	0	6
79	9	7	6	9	7	10	9	12	85	10	9	10	8	20	8	7	8	18	6	0
80	6	4	3	5	4	6	6	7	85	6	6	7	5	16	5	4	4	16	4	6
81	4	3	3	5	3	5	7	6	90	5	6	5	4	20	4	3	1	16	7	8
82	5	4	4	5	4	5	7	7	90	6	7	5	4	22	5	4	2	16	8	9
83	16	14	14	16	14	18	16	18	85	16	16	18	14	24	16	14	14	24	12	10
84	4	3	1	3	3	5	5	6	90	4	4	5	2	20	2	1	3	16	7	8
85	6	4	3	1	4	7	6	8	90	1	6	7	1	22	1	1	5	18	9	9
86	6	5	4	2	5	7	7	8	90	1	7	7	2	22	1	2	5	18	9	10

Continued on next page

Table A.11 – continued from previous page

ID	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
87	16	4	3	1	14	18	6	8	85	1	6	7	1	22	1	1	5	22	9	9
88	26	20	18	22	20	22	22	18	100	22	22	20	20	22	20	20	20	10	18	16
89	28	26	24	26	26	28	24	30	70	28	24	30	24	35	26	24	24	20	22	20
90	14	14	14	14	14	14	18	16	90	16	16	14	14	30	14	14	12	14	18	18

Table A.12: Durham dataset distance between POIs

ID	80	81	82	83	84	85	86	87	88	89	90
0	7	6	6	17	4	1	1	16	16	28	16
1	22	20	21	27	20	23	23	31	35	35	24
2	3	3	3	13	2	4	5	12	17	24	13
3	12	10	11	18	10	13	14	22	22	35	16
4	6	6	6	17	4	1	0	17	22	28	16
5	6	5	6	17	3	1	1	17	22	28	15
6	13	11	12	13	12	14	15	18	26	26	1
7	4	2	3	13	1	2	2	13	18	24	12
8	4	2	3	13	1	2	2	13	18	2	12
9	4	2	3	14	1	2	2	13	19	24	12
10	4	2	3	14	1	2	2	13	18	6	12
11	6	6	7	15	4	6	7	15	20	24	16
12	9	12	13	11	12	13	14	11	18	12	18
13	10	7	8	17	8	10	10	19	14	9	14
14	5	3	4	14	2	1	1	14	19	3	13
15	10	9	9	18	9	12	12	20	16	14	16
16	6	7	8	14	7	8	10	14	18	9	18
17	14	14	14	5	14	18	16	12	24	16	16
18	6	8	8	14	7	9	10	14	20	9	18

Continued on next page

Table A.12 – continued from previous page

ID	80	81	82	83	84	85	86	87	88	89	90
19	5	9	9	14	9	10	12	14	20	10	20
20	14	12	12	18	12	14	14	24	28	16	16
21	2	5	6	16	5	7	7	14	18	7	16
22	14	14	14	18	14	16	16	24	28	14	18
23	8	8	9	20	8	9	10	20	18	12	18
24	5	3	4	16	4	5	6	16	22	7	12
25	8	7	8	18	8	10	10	20	14	12	16
26	7	12	12	20	10	14	14	20	24	12	22
27	14	12	12	12	12	14	16	18	28	16	6
28	5	5	5	18	3	2	2	16	22	6	16
29	14	14	14	12	14	16	16	20	30	7	6
30	20	22	22	20	22	24	24	16	30	24	30
31	12	12	12	5	12	14	14	12	22	16	12
32	5	3	1	14	4	7	7	7	20	28	14
33	22	24	24	24	24	26	26	20	35	26	35
34	3	2	3	12	2	4	4	4	18	22	12
35	4	3	4	12	3	5	5	5	18	26	12
36	10	8	9	14	8	10	12	10	24	22	12
37	6	8	8	14	8	9	10	14	20	24	18
38	7	6	7	18	6	7	8	16	18	6	18
39	7	6	7	18	6	7	8	16	18	6	18
40	5	5	5	14	5	7	8	7	18	26	16
41	6	9	9	16	9	10	10	10	20	24	20
42	16	20	20	22	20	20	22	18	26	35	30
43	27	25	20	18	19	20	13	19	20	25	23
44	8	7	8	18	7	10	12	10	16	30	18
45	2	5	6	14	5	7	8	40	24	24	16
46	10	10	12	18	10	12	14	18	14	26	20

Continued on next page

Table A.12 – continued from previous page

ID	80	81	82	83	84	85	86	87	88	89	90
47	4	4	4	16	2	2	3	16	20	24	14
48	6	8	8	9	7	9	10	9	18	20	16
49	12	14	14	14	14	16	16	12	22	28	20
50	30	28	30	40	30	35	35	40	28	45	35
51	3	3	3	14	3	5	5	14	20	24	12
52	4	3	3	16	1	3	4	14	20	24	12
53	5	4	5	16	2	3	3	16	20	24	14
54	5	1	2	16	4	5	6	16	20	24	12
55	4	4	4	14	3	28	28	18	35	24	35
56	4	4	4	14	2	1	2	14	20	26	14
57	3	3	4	14	3	5	5	14	20	24	12
58	4	2	3	14	2	4	5	4	20	24	12
59	2	4	4	12	2	4	5	12	18	22	14
60	6	4	5	16	4	6	6	16	26	28	14
61	4	3	4	14	3	4	5	4	20	26	14
62	3	3	4	14	1	3	4	3	18	24	14
63	5	5	5	16	3	1	2	1	22	26	14
64	4	3	4	14	3	4	5	14	20	26	14
65	6	5	5	18	5	7	7	18	22	28	14
66	6	7	7	16	5	6	7	6	22	24	18
67	7	6	7	18	6	8	8	8	18	30	16
68	85	90	90	85	90	90	90	85	100	70	90
69	6	5	6	16	4	1	1	1	22	28	16
70	6	6	7	16	4	6	7	6	22	24	16
71	7	5	5	18	5	7	7	7	20	30	14
72	5	4	4	14	2	1	2	1	20	24	14
73	16	20	22	24	20	22	22	22	22	35	30
74	5	4	5	16	2	1	1	1	20	26	14

Continued on next page

Table A.12 – continued from previous page

ID	80	81	82	83	84	85	86	87	88	89	90
75	4	3	4	14	1	1	2	1	20	24	14
76	4	1	2	14	3	5	5	5	20	24	12
77	16	16	16	24	16	18	18	22	10	20	24
78	4	7	8	12	7	9	9	9	18	22	18
79	6	8	9	10	8	9	10	9	16	20	18
80	0	4	5	14	4	5	5	5	18	22	14
81	4	0	2	14	3	5	5	5	20	22	12
82	5	2	0	16	3	5	5	5	20	24	14
83	14	14	16	0	12	16	18	9	24	18	14
84	4	3	3	12	0	3	3	3	20	24	14
85	5	5	5	16	3	0	1	16	22	26	16
86	5	5	5	18	3	1	0	35	22	28	16
87	5	5	5	9	3	16	35	0	35	24	35
88	18	20	20	24	20	22	22	35	0	30	28
89	22	22	24	18	24	26	28	24	30	0	26
90	14	12	14	14	14	16	16	35	28	26	0

Appendix B

The Flexible Travel Recommender Model Results

p4.2.a	p4.2.b	p4.2.c	p4.2.d	p4.2.e	p4.2.f	p4.2.g	p4.2.h	p4.2.i	p4.2.j	p4.2.k	p4.2.l	p4.2.m	p4.2.n	p4.2.o
160	315	401	513	542	622	678	737	804	851	925	972	980	1052	1080

p4.2.p	p4.2.q	p4.2.r	p4.2.s	p4.2.t	p4.3.a	p4.3.b	p4.3.c	p4.3.d	p4.3.e	p4.3.f	p4.3.g	p4.3.h	p4.3.i	p4.3.j
1113	1161	1168	1213	1223	0	38	171	283	425	501	628	674	684	773

p4.3.k	p4.3.l	p4.3.m	p4.3.n	p4.3.o	p4.3.p	p4.3.q	p4.3.r	p4.3.s	p4.3.t	p4.4.a	p4.4.b	p4.4.c	p4.4.d	p4.4.e
812	861	910	976	1061	1092	1083	1159	1187	1212	0	0	0	38	181

p4.4.f	p4.4.g	p4.4.h	p4.4.i	p4.4.j	p4.4.k	p4.4.l	p4.4.m	p4.4.n	p4.4.o	p4.4.p	p4.4.q	p4.4.r	p4.4.s	p4.4.t
269	443	526	573	649	718	771	813	849	900	945	1012	1062	1144	1139

Figure B.1: The results of $Dataset_4$ for the TOP

p5.2.a	p5.2.b	p5.2.c	p5.2.d	p5.2.e	p5.2.f	p5.2.g	p5.2.h	p5.2.i	p5.2.j	p5.2.k	p5.2.l	p5.2.m	p5.2.n	p5.2.o
-	20	50	80	180	240	315	410	480	580	670	800	860	925	1020

p5.2.p	p5.2.q	p5.2.r	p5.2.s	p5.2.t	p5.2.u	p5.2.v	p5.2.w	p5.2.x	p5.2.y	p5.2.z	p5.3.a	p5.3.b	p5.3.c	p5.3.d
1150	1185	1240	1300	1355	1410	1475	1525	1585	1600	1635	-	15	20	60

p5.3.e	p5.3.f	p5.3.g	p5.3.h	p5.3.i	p5.3.j	p5.3.k	p5.3.l	p5.3.m	p5.3.n	p5.3.o	p5.3.p	p5.3.q	p5.3.r	p5.3.s
95	110	185	260	335	470	495	585	650	755	870	990	1055	1110	1170

p5.3.t	p5.3.u	p5.3.v	p5.3.w	p5.3.x	p5.3.y	p5.3.z	p5.4.a	p5.4.b	p5.4.c	p5.4.d	p5.4.e	p5.4.f	p5.4.g	p5.4.h
1205	1290	1380	1440	1510	1555	1585	0	0	20	20	20	80	140	140

p5.4.i	p5.4.j	p5.4.k	p5.4.l	p5.4.m	p5.4.n	p5.4.o	p5.4.p	p5.4.q	p5.4.r	p5.4.s	p5.4.t	p5.4.u	p5.4.v	p5.4.w
240	340	340	425	555	620	690	760	860	960	1020	1160	1300	1320	1360

p5.4.x	p5.4.y	p5.4.z
1420	1435	1535

Figure B.2: The results of $Dataset_5$ for the TOP

p6.2.d	p6.2.e	p6.2.f	p6.2.g	p6.2.h	p6.2.i	p6.2.j	p6.2.k	p6.2.l	p6.2.m	p6.2.n	p6.3.g
192	360	588	660	780	888	948	1008	1104	1140	1200	276

p6.3.h	p6.3.i	p6.3.j	p6.3.k	p6.3.l	p6.3.m	p6.3.n	p6.4.j	p6.4.k	p6.4.l	p6.4.m	p6.4.n
438	630	828	894	1002	1068	1140	360	498	672	912	1068

Figure B.3: The results of $Dataset_6$ for the TOP

p7.2.a	p7.2.b	p7.2.c	p7.2.d	p7.2.e	p7.2.f	p7.2.g	p7.2.h	p7.2.i	p7.2.j	p7.2.k	p7.2.l	p7.2.m	p7.2.n	p7.2.o
30	64	100	190	279	375	440	511	560	619	680	724	772	814	836

p7.2.p	p7.2.q	p7.2.r	p7.2.s	p7.2.t	p7.3.a	p7.3.b	p7.3.c	p7.3.d	p7.3.e	p7.3.f	p7.3.g	p7.3.h	p7.3.i	p7.3.j
924	942	935	966	1000	-	46	79	117	158	228	307	404	473	544

p7.3.k	p7.3.l	p7.3.m	p7.3.n	p7.3.o	p7.3.p	p7.3.q	p7.3.r	p7.3.s	p7.3.t	p7.4.a	p7.4.b	p7.4.c	p7.4.d	p7.4.e
607	662	730	771	831	891	917	955	996	1030	-	30	46	79	123

p7.4.f	p7.4.g	p7.4.h	p7.4.i	p7.4.j	p7.4.k	p7.4.l	p7.4.m	p7.4.n	p7.4.o	p7.4.p	p7.4.q	p7.4.r	p7.4.s	p7.4.t
164	208	264	331	426	496	566	619	677	746	804	870	919	973	1006

Figure B.4: The results of $Dataset_7$ for the TOP

Appendix C

The Happiness Model Results

Tables C.1 to C.8 show the full path (Recommended tour trip) for Experiment #1 to #8.

Table C.1: The full path of the Experiment #1 results on $Dataset_1$

Scenario Code	Path
Pr02	0 6 22 32 95 57 29 58 52 61 45 26 21 91 46 54 11 25 69 81 0
Pr03	0 69 62 139 5 104 29 82 59 93 33 128 123 18 22 25 43 63 0
Pr04	0 147 31 4 23 44 37 175 8 124 83 136 54 61 121 104 152 117 65 5 0
Pr05	0 44 80 29 81 211 4 154 75 157 193 11 91 203 142 235 187 20 55 35 197 181 28 9 32 0
Pr06	0 80 149 278 116 74 279 95 210 130 219 123 241 196 246 129 166 228 167 215 33 182 121 197 286 0
Pr08	0 117 87 108 68 125 77 132 106 133 127 109 102 140 10 42 23 28 19 0
Pr09	0 141 96 50 49 160 195 102 76 156 21 118 16 73 108 127 207 197 170 5 175 0
Pr10	0 19 103 206 246 263 108 251 22 277 92 201 192 100 179 233 17 177 173 144 129 274 157 118 14 54 0

Table C.2: The full path of the Experiment #2 results on $Dataset_1$

Scenario Code	Path
Pr02	0 6 22 32 95 57 29 58 52 61 45 26 21 91 46 54 11 25 69 81 0
Pr03	0 69 62 139 5 104 29 82 59 93 33 128 123 18 22 25 43 63 0
Pr04	0 147 31 4 23 44 37 175 8 124 83 136 54 61 121 104 152 117 65 5 0
Pr05	0 44 80 29 81 211 4 154 75 157 193 11 91 203 142 235 187 20 55 35 197 181 28 9 32 0
Pr06	0 80 149 278 116 74 279 95 210 130 219 123 241 196 246 129 166 228 167 215 33 182 121 197 286 0
Pr08	0 117 87 108 68 125 77 132 106 133 127 109 102 140 10 42 23 28 19 0
Pr09	0 141 96 50 49 160 195 102 76 156 21 118 16 73 108 127 207 197 170 5 175 0
Pr10	0 19 103 206 246 263 108 251 22 277 92 201 192 100 179 233 17 177 173 144 129 274 157 118 14 54 0

Table C.3: The full path of the Experiment #3 results on $Dataset_1$

Scenario Code	Path
Pr02	0 6 32 60 72 95 57 29 41 61 52 21 91 46 54 11 25 87 33 81 0
Pr03	0 69 39 62 17 5 104 29 82 59 93 33 128 123 18 22 25 4 12 0
Pr04	0 147 31 4 23 51 185 175 8 37 100 169 96 61 191 26 144 67 152 0
Pr05	0 44 8 24 116 29 102 188 183 105 165 18 132 79 85 182 30 198 64 225 121 28 9 32 0
Pr06	0 258 44 265 39 180 118 179 111 184 11 172 261 150 273 78 228 33 215 182 121 167 252 286 0
Pr08	0 117 87 125 77 139 108 134 132 65 24 92 1 76 112 124 21 138 41 144 111 0
Pr09	0 171 180 59 105 42 138 113 204 179 65 108 16 121 127 73 207 5 170 51 116 181 74 175 0
Pr10	0 19 188 143 166 92 32 25 114 97 267 232 205 34 151 134 141 49 86 126 51 54 14 83 0

Table C.4: The full path of the Experiment #4 results on $Dataset_1$

Scenario Code	Path
Pr02	0 6 18 67 80 24 32 65 57 26 35 31 45 52 21 91 54 88 33 78 87 81 0
Pr03	0 69 39 87 17 139 52 88 80 26 71 107 43 22 123 18 25 4 0
Pr04	0 147 31 4 23 175 8 37 44 185 189 9 177 121 67 161 117 65 152 0
Pr05	0 44 80 29 81 231 115 59 102 188 183 105 132 85 182 212 30 64 198 136 108 9 32 103 0
Pr06	0 80 267 195 219 16 130 138 229 95 188 127 261 87 78 14 197 287 143 228 167 201 0
Pr08	0 117 87 108 68 77 125 56 135 54 123 24 92 81 144 71 111 28 101 0
Pr09	0 171 166 192 90 138 89 134 183 114 51 5 108 73 127 121 16 118 156 21 31 74 170 0
Pr10	0 19 263 108 181 115 253 158 243 27 2 258 72 155 242 71 59 101 285 156 208 6 152 128 0

Table C.5: The full path of the Experiment #5 results on $Dataset_2$

Scenario Code	Path
Pr11	0 29 34 21 3 8 25 41 16 10 1 27 35 2 28 40 44 38 9 12 0
Pr12	0 27 19 17 79 14 32 65 0
Pr13	0 66 40 29 64 78 9 122 44 117 18 105 123 23 99 31 115 10 63 129 79 104 141 0
Pr14	0 114 96 60 35 179 174 26 88 76 178 180 9 168 120 139 84 91 95 138 184 25 85 147 183 0
Pr15	0 169 103 69 165 80 107 4 11 150 119 44 237 2 145 203 71 48 136 153 85 77 125 89 202 67 25 209 204 21 36 0
Pr16	0 124 149 230 128 121 30 33 215 228 106 139 81 167 182 122 143 197 287 34 159 116 78 137 261 216 69 0
Pr17	0 51 50 29 54 15 30 12 56 6 5 58 65 37 21 45 9 10 69 0
Pr18	0 118 75 70 9 133 7 89 63 41 138 11 36 88 116 115 8 14 93 48 50 82 0
Pr19	0 50 214 53 123 30 137 88 74 213 108 113 75 106 9 72 195 206 118 166 200 87 135 14 105 5 0
Pr20	0 58 246 205 27 101 114 105 277 220 144 173 20 8 208 196 279 2 256 159 124 89 85 138 179 228 56 12 241 181 0

Table C.6: The full path of the Experiment #6 results on $Dataset_2$

Scenario Code	Path
Pr11	0 29 34 21 3 8 25 41 16 10 1 27 35 2 28 40 44 38 9 12 0
Pr12	0 27 19 17 79 14 32 65 0
Pr13	0 66 40 29 64 78 9 122 44 117 18 105 123 23 99 31 115 10 63 129 79 104 141 0
Pr14	0 114 96 60 35 179 174 26 88 76 178 180 9 168 120 139 84 91 95 138 184 25 85 147 183 0
Pr15	0 169 103 69 165 80 107 4 11 150 119 44 237 2 145 203 71 48 136 153 85 77 125 89 202 67 25 209 204 21 36 0
Pr16	0 124 149 230 128 121 30 33 215 228 106 139 81 167 182 122 143 197 287 34 159 116 78 137 261 216 69 0
Pr17	0 51 50 29 54 15 30 12 56 6 5 58 65 37 21 45 9 10 69 0
Pr18	0 118 75 70 9 133 7 89 63 41 138 11 36 88 116 115 8 14 93 48 50 82 0
Pr19	0 50 214 53 123 30 137 88 74 213 108 113 75 106 9 72 195 206 118 166 200 87 135 14 105 5 0
Pr20	0 58 246 205 27 101 114 105 277 220 144 173 20 8 208 196 279 2 256 159 124 89 85 138 179 228 56 12 241 181 0

Table C.7: The full path of the Experiment #7 results on $Dataset_2$

Scenario Code	Path
Pr11	0 20 8 21 3 42 45 24 23 46 26 10 1 13 7 38 22 28 0
Pr12	0 6 52 4 35 87 0
Pr13	0 82 118 53 29 64 33 49 105 125 44 117 18 4 2 95 81 15 48 16 70 74 0
Pr14	0 181 60 35 179 174 26 3 134 124 20 85 108 103 157 155 91 95 180 33 66 176 16 162 0
Pr15	0 216 53 117 230 135 218 11 80 123 78 192 48 120 150 119 24 49 125 40 193 176 2 25 209 204 154 174 179 0
Pr16	0 124 149 230 128 30 121 12 33 215 228 139 167 81 106 182 143 197 287 34 116 78 86 3 0
Pr17	0 43 51 50 29 23 54 22 32 30 36 35 37 15 60 61 45 9 10 69 0
Pr18	0 124 128 113 71 106 34 75 70 87 30 18 100 24 48 83 2 46 89 77 62 107 82 0
Pr19	0 50 214 53 123 27 63 192 166 80 83 74 213 108 13 148 203 201 186 49 175 8 9 72 149 14 23 0
Pr20	0 125 249 189 242 156 142 5 285 225 243 163 84 176 205 97 129 63 275 276 263 251 26 120 132 168 221 138 143 207 0

Table C.8: The full path of the Experiment #8 results on $Dataset_2$

Scenario Code	Path
Pr11	0 20 8 21 3 42 27 35 2 29 43 37 7 30 9 26 33 22 28 0
Pr12	0 6 52 21 32 33 0
Pr13	0 134 29 24 111 107 26 13 82 4 2 95 44 117 18 7 130 61 86 58 70 74 0
Pr14	0 114 65 127 123 152 124 12 168 120 81 179 174 26 188 157 39 91 95 126 151 13 104 61 183 0
Pr15	0 216 211 150 215 120 184 80 117 231 11 182 163 165 62 74 107 49 88 75 119 77 36 79 142 61 149 193 0
Pr16	0 124 149 230 128 121 30 33 215 228 106 167 81 182 122 143 287 197 34 116 78 137 261 168 253 3 0
Pr17	0 5 47 44 43 63 35 1 22 32 30 62 15 60 61 45 9 10 69 0
Pr18	0 125 17 54 18 131 32 1 84 61 115 8 30 116 34 70 87 39 105 92 99 0
Pr19	0 131 107 88 117 184 76 80 35 50 214 116 96 195 206 137 49 8 9 72 165 105 5 68 58 21 93 0
Pr20	0 7 208 14 161 101 141 134 267 72 83 63 275 173 4 184 219 65 119 140 191 263 251 23 71 233 11 228 110 143 207 0