PUBLICATIONS OF
THE UNIVERSITY OF EASTERN FINLAND

Dissertations in Forestry and Natural Sciences

UNIVERSITY OF
EASTERN FINLAND

LAHARI SENGUPTA

EVALUATION AND PLAYERS' PERFORMANCE OF THE LOCATION-BASED GAME O-MOPSI

EVALUATION AND PLAYERS' PERFORMANCE OF THE LOCATION-BASED GAME O-MOPSI

Lahari Sengupta

EVALUATION AND PLAYERS' PERFORMANCE OF THE LOCATION-BASED GAME O-MOPSI

Publications of the University of Eastern Finland
Dissertations in Forestry and Natural Sciences
No 358

University of Eastern Finland
Joensuu
2019

Academic dissertation
To be presented by permission of the Faculty of Science and Forestry for public examination in Louhela Auditorium in Science Park at the University of Eastern Finland, Joensuu, on December, 09, 2019, at 12 o'clock noon

Grano Oy
Jyväskylä, 2019
Editors: Pertti Pasanen, Matti Tedre,
Jukka Tuomela, Raine Kortet
Distribution: University of Eastern Finland / Sales of publications
www.uef.fi/kirjasto
ISBN: 978-952-61-3248-8 (Print)
ISBN: 978-952-61-3249-5 (PDF)
ISSNL: 1798-5668
ISSN: 1798-5668
ISSN: 1798-5676 (PDF)

Author's address: Lahari Sengupta
University of Eastern Finland
School of Computing
P.O. Box 111
80101 JOENSUU, FINLAND
email: lahari@cs.uef.fi

Supervisors: Professor Pasi Fränti, Ph.D.
University of Eastern Finland
School of Computing
P.O. Box 111
80101 JOENSUU, FINLAND
email: franti@cs.uef.fi

Dr. Radu Mariescu-Istodor, Ph.D.
University of Eastern Finland
School of Computing
P.O. Box 111
80101 JOENSUU, FINLAND
email: radum@cs.uef.fi

Reviewers: Professor (Emeritus) Jari Veijalainen, Ph.D.
University of Jyväskylä
Department of Computer Science and Information Systems
P.O. Box 35
40014 JYVÄSKYLÄ, FINLAND
email: vei5423@gmail.com

Dr. Damianos Gavalas, Ph.D.
University of the Aegean
Department of Product & Systems Design Engineering
Konstantinoupoleos 2
GR-84100, Syros, GREECE
email: dgavalas@aegean.gr

Opponent: Professor Vincent Charvillat, Ph.D.
University of Toulouse
Computer Science & Applied Mathematics Department
ENSEEIHT Engineering School.
IRIT Lab. (VORTEX research group).
TOULOUSE, FRANCE
email: Vincent.Charvillat@enseeiht.fr

Sengupta, Lahari
Evaluation and players' performance of the location-based game O-Mopsi
Joensuu: University of Eastern Finland, 2019
Publications of the University of Eastern Finland
Dissertations in Forestry and Natural Sciences 2019; 358
ISBN: 978-952-61-3248-8 (Print)
ISSNL: 1798-5668
ISSN: 1798-5668
ISBN: 978-952-61-3249-5 (PDF)
ISSN: 1798-5676 (PDF)

ABSTRACT

The number of available location-based games and applications increases every day. Providing motivations to use such an application is expected to be the key to its success. This research evaluates the challenges of a location-based game, O-Mopsi. We also study the players' performance in playing the game.

O-Mopsi is a mobile-based orienteering game where players need to visit several real-world targets in the fastest possible way. Therefore, it includes a real-world open-loop travelling salesman problem as a part of the game-playing. Players need navigational skills and path optimisation skills during play. Before beginning the play, players need to optimise their starting location as a good starting location helps to optimise the path. In this research, we study and compare different strategies for finding a better starting location for O-Mopsi. The results show that players' performance improved when they followed a strategy.

Finding an optimised path for every game instance is not equally difficult for players. We examined the characteristics of the game instances. Our results show that the number of targets and the structure of the minimum spanning tree are two crucial parameters to determine the difficulty of the game instance. We found that the difficulty increases with the number of targets. Similarly, as the number of branches in the minimum spanning tree increases, the difficulty increases.

This study provides a simple local search algorithm for path optimisation to deliver a fast and real-time reference solution to the players. This thesis also studies the quality, efficiency, and limitation of four local search operators of which two are existing (relocate and 2-opt) and two are new (3-permute and link swap). Our results show that one of the new operators, link swap outperforms the other operators.

Universal Decimal Classification: 004.382.745, 004.9, 519.83, 795

Library of Congress Subject Headings: Location-based services; Mobile apps; Games; Orienteering; Traveling salesman problem; Planning; Performance; Algorithms

Yleinen suomalainen ontologia: digitaaliset pelit; mobiilipelit; mobiilisovellukset; suunnistus; strategiat; optimointi; suoriutuminen; suorituskyky; algoritmit

ACKNOWLEDGEMENTS

This thesis is a summary of the work that has been carried out in the Machine Learning Group at the School of Computing, University of Eastern Finland, Finland, from 2015 to 2019. Along with several research results, I have discovered a new 'me' through the journey of these years. I have loved spending these years in Finland. I am very thankful to the endless forests and natural beauty of Finland that encouraged the explorer within me. I also wish to thank Finnish weather (especially Winter) for keeping me motivated to stay indoors, which indirectly increased my working hours. I want to thank Joensuu Table Tennis club (Joensuun Pöytätennis – JPT) for providing me with a place to practise my hobby. I am thankful to story & novel writers, singers & song-makers, and moviemakers around the world that they never disappointed me whenever I had sought for a new collection for refreshment and motivation. I am immensely grateful to several web-search engines (especially Google), without which, things might have been unimaginably difficult. I also want to thank my critics and non-supporters for making me strong.

I would like to express my sincere gratitude to my supervisor, Prof. Pasi Fränti, who accepted me as a PhD student in his research group. I am thankful to his continuous guidance, evaluation, and support, which helped me to stay motivated all along. Without him, this thesis would never have been accomplished. Apart from the research work, I have learnt several practical aspects of life from him. He has inspired me to experience several new things, which has helped me to gain confidence.

I wish to thank my co-supervisor, Dr. Radu Mariescu-Istodor, who has been my friend, philosopher, and guide during these days. From the very beginning, he guided me to proceed to my work, helped me to understand enormous technical concepts. His positivity always boosts my confidence. I have enjoyed his initiatives and company in long-distance cycling trips. He has even spent hours of endeavour on teaching me skiing.

I would also like to thank my colleagues at the Machine Learning Group, with whom I have had the pleasure to work with during these years. Special thanks to Nancy Fazal for commenting on my thesis.

I am especially grateful to Professor Jari Veijalainen and Dr. Damianos Gavalas, the reviewers of this thesis, for their feedback and valuable comments.

I am thankful to Dr. Tuomo Kauranne, the President of Arbonaut and Vesa Leppänen, the CTO of Arbonaut for offering me to work in their company. I am extremely grateful to all my Arbonaut colleagues for providing me with enough flexibility and support that helped in continuing my studies at the university and working in the company simultaneously.

I want to thank Dr. Nabarun Bhattacharyya, Senior Director & Centre head of C-DAC, Kolkata, who encouraged me to pursue a PhD degree.

I would like to thank my friends here in Joensuu and back in Kolkata, who have been my great support during this time. I am greatly indebted to my parents, in-laws, sister, cousins, and my husband, Subhajit. I am thankful to have my niece and Subhajit's nephew in my life. During these years, I have enjoyed thoughtful discussions with them, sometimes sitting next to them or sometimes over video calls. I often remember my grandmother, who used to encourage my interest to study new things during my childhood. I miss my aunt today, who was a role model of energy and enthusiasm to me. I have no words to thank my parents, sister, and Subhajit for being patient, listening to my frustrating gibberish, still being positive, and keeping faith with me during these years. Without their support, love, and care, I could not have accomplished anything. I will always be grateful to Subhajit who has been the main reason and motivation behind my every accomplishment in Finland. His presence, support, enthusiasm, and love has made my life much easier these days.

To my Ma.

Joensuu, 07th November 2019
Lahari Sengupta

LIST OF ABBREVIATIONS

GNSS	Global navigation satellite system
GPS	Global Positioning System
LBG	Location-based game
RFID	Radio-frequency identification
QR code	Quick response code
POI	Places of interests
TTDP	Tourist trip design problem
O-Mopsi	Mopsi orienteering
TSP	Travelling salesman problem
OSM	Open street map
MST	Minimum spanning tree

LIST OF ORIGINAL PUBLICATIONS

This thesis is based on data presented in the following articles, referred to by the Roman Numerals I-IV.

I Fränti, P., Mariescu-Istodor, R. and Sengupta, L. (2017) O-Mopsi: Mobile Orienteering Game for Sightseeing, Exercising, and Education. *ACM Trans. Multimedia Comput. Commun. Appl.* 13 (4), 56:1-25.

II Sengupta, L, Mariescu-Istodor, R., and Fränti, P. (2018), Planning your route: where to start?, *Computational Brain & Behavior*, 1(3-4), 252-265.

III Sengupta, L. and Fränti, P. (2019), Predicting the difficulty of TSP instances using MST, *IEEE Int. Conf. on Industrial Informatics* (INDIN), Helsinki, 848-852.

IV Sengupta, L, Mariescu-Istodor, R., and Fränti, P. (2019), Which Local Search Operator Works Best for the Open-Loop TSP? *Applied Sciences*. 9(19):3985.

AUTHOR'S CONTRIBUTION

I) The paper was written by Prof. Pasi Fränti. The author contributed by polishing the writing, preparing some of the illustrations, and web-developing related to the needed O-Mopsi result analysis. The author also contributed to organising SciFest workshops and collecting and analysing the survey results.

II) The idea was originated by Prof. Pasi Fränti. The author implemented and performed all the experiments, prepared most of the illustrations, and acted as principal author in writing the paper. Co-authors polished the writing.

III) The idea was developed by Prof. Pasi Fränti. The author implemented all the methods, performed experiments, and wrote the paper. Prof. Pasi Fränti polished the writing.

IV) The idea was developed by Prof. Pasi Fränti and was jointly refined by the co-authors through discussions. The method was implemented and the experiments conducted by the author and the second author. The author wrote the paper and co-authors modified the writing.

CONTENTS

1 INTRODUCTION

A global navigation satellite system (GNSS) can provide the location information of a GNSS receiver. Among several such systems, Global Positioning System (GPS) advanced significantly and gained enormous popularity in recent years. This results in the current availability of a multitude of location-based applications. The entertainment and tourism industry underwent a radical change with the introduction of these applications. While computer games are generally popular among people of all ages, the development of *location-based games* (LBG), which include location as a part of the games, has made computer games more attractive.

1.1 LOCATION-BASED GAMES

Location-based games are mobile or computer games that record the location of the player for the purposes of the game-playing. The location of the player is recorded mainly by GPS, Bluetooth, Wi-Fi, or near-field communication technologies. LBGs are kind of pervasive games, which include real-world to the virtual game environment. Magerkurth et al. (2005) reviewed pervasive games based on their genres and included location-aware games as one of the types of pervasive games. Kasapakis and Gavalas (2015) surveyed 18 pervasive games, which are mostly location-based games. They studied the challenges, design principles, and implementation guidelines for this kind of games. One of the oldest LBG is Can you see me now? [Benford et al., 2006]. Geocaching is another old game which is still popular. CityExplorer [Matyas et al., 2008], Monopoly [Li et al., 2008], Quake [Piekarski and Thomas, 2002], Human Pacman [Cheok et al., 2004], Tic-Tac-Toe [Schlieder et al., 2005], Chase and Catch [Misund et al., 2009], Snake [Chittaro and Sioni, 2012], and Invisible City: Rebels vs. Spies [Sintoris et al., 2013] are some examples of old board games or computer games that have been converted by game developers into real-life location-based games. SoundPacman [Chatzidimitris et al., 2016] is another modified version of Pacman that includes 3D sound to augment the physical world to the game environment.

In **[I]**, we reviewed location-based games based on these parameters: game-playing mode, how to use location, how to verify location, games including exercise, and games with educational goals. Most people prefer games with a simple setup that are easy to learn and play and that can be played in a short period of time. The possibility of playing a game offline also increases its popularity. In these games, location verification is mostly done through GPS. *Radio-frequency identification* (RFID), *quick response* (QR) code, Bluetooth, and other near-field communication technologies are less popular due to their instability and the cost of building extra setup. Sometimes manual verification occurs by authentication of a photograph.

Traditional computer games force people to sit idle for a long time, which can cause serious health issues in the long term. LBGs can overcome this situation and make people active. In several cases, these games aim to teach something, which adds extra benefits and motivation, in addition to physical exercise. Avouris and Yiannoutsou [2012] reviewed mobile location-based games based on their objectives. The first type of games is mainly aimed at entertainment, and learning is its secondary purpose. Learning is the main objective of the second type of game. The last type is hybrid, which means that it combines entertainment and learning. Educational LBGs can help players to learn about a zoo and animal behaviour [Sánchez et al., 2006, Facer et al., 2004, Veenhof et al., 2012], a museum [Sintoris et al., 2012], a specific historical place, or a cultural site [Vassilakis et al., 2017, La Guardia et al., 2012, Ballagas et al., 2007]. They can improve people's navigational skills by providing a treasure hunt platform [Wetzel et al., 2012, Ceipidor et al., 2009] and increase their ability to play puzzle games [Sedano et al., 2012, Moore et al., 2009] and solve mysteries [Spikol and Milrad]. Giannakas et al. [2018] reviewed educational location-based mobile games published during the period 2004–2016. With the advancement of augmented reality technology, mobile games start to combine location-based applications and augmented reality. Ingress[1], PokemonGo[2], and the very recent Harry Potter: Wizards Unite[3] are the most popular examples of such a combination.

1.2 ISSUES WITH LOCATION-BASED GAMES

Although LBGs have been available for the last 20 years and have achieved local popularity, only a few, such as PokemonGo, which was released in 2016, have gained worldwide popularity. Certain issues have prevented these games from achieving wide recognition. Early location-based games had to deal with a limited internet connection, which made it difficult for people to play online games. Furthermore, smartphone technology was not so advanced, which resulted in low battery life, low processing power, small screen sizes, limited storage, and low graphics capability, all of which made playing these games more challenging [Jacob and Coelho, 2011, Lynch, 2012]. Recently, we have overcome almost all of these issues with advanced GPS and smartphone technology. Therefore, location-based games are becoming more popular. However, safety is now the main issue with playing these games. While playing these games, people have to look at and concentrate on their smartphones, which distracts them from the real world surrounding them. However, games that use real-world targets rather than augmented reality targets draw the attention of players to the surrounding world.

[1] https://www.ingress.com/
[2] https://www.pokemongo.com/en-us/
[3] https://www.harrypotterwizardsunite.com/

1.3 TRIP PLANNING

Trip planning is another location-based application that is also very popular in recent days. The tourism industry is changing every day with the help of GPS and smartphone technology. In the past, people used to follow travel books, paper maps, and suggestions from experienced people. However, today, there are numerous travel advisory websites and mobile applications even for remote places. All these applications display the locations of *places of interest* (POI) on an online map. They help people to make customised trip plans throughout the world Similar to LBGs. Trip planning applications also need real-world information that has to be embedded. Figure 1 shows an example of a map showing the locations of museums and historic buildings around Helsinki.

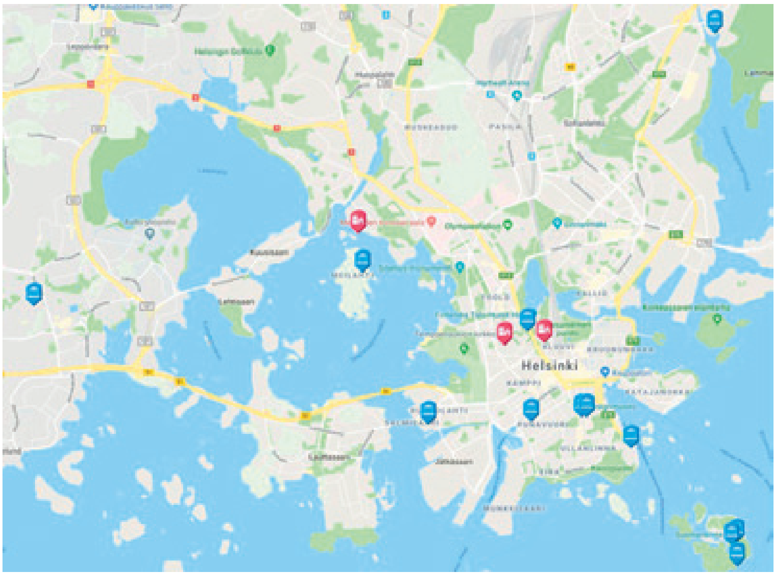

Figure 1: Museums and historic buildings in and around Helsinki
(Map source: https://www.helsinkicard.com/)

With the boom in sharing personal content on the web or mobile-based social media, research on personalised recommendation systems has increased. Surveys by Borràs et al. (2014), Gavalas et al. (2014 a), Haruna et al. (2017), and Lim et al. (2018) affirm that research and development on tourism recommendation systems have also increased. The problem of tour planning and optimisation, which forms a part of tourism recommender systems, is called the *tourist trip design problem* (TTDP). Gavalas et al.'s (2014 b) survey presents various algorithms to solve the TTDP, and De Choudhury et al. (2010), Gionis et al. (2014), Bolzoni et al. (2014), Mor and Dalyot (2018), and Agarwal et al. (2018) have proposed several approaches for planning and optimising tours. In this thesis, we present a location-based game that can also be used as a trip advisory application.

1.4 O-MOPSI

Mopsi Orienteering (O-Mopsi) is a location-based mobile game, in which a player has to find a set of targets around a city area (Figure 2). O-Mopsi targets are always clearly visible outdoor targets, which can be reached by walking. There are 8 targets in *SciFest 2018* game (example in Figure 2), which are easily noticeable objects near Joensuu Areena. Hence, it can be used as a city-tour advisory application or an educational game for students. The benefit of O-Mopsi is that it encourages physical activity as players need to walk to reach the targets. O-Mopsi includes sound navigation, a feature that means players are not forced to look at their phone screens all the time. In the next chapter, we discuss the features and playing methods in detail.

Figure 2: An O-Mopsi game layout.

The O-Mopsi application contains 158 game instances (as on 10th September 2019). Each game instance has a specific name based on mainly the area where the game is created or the event when the game is created. For example, the game shown in Figure 2 was created during a festival called SciFest in 2018. Therefore, the name of the game is SciFest 2018. Players can start and finish any game anytime. O-Mopsi keeps record per each game that enables the player to check the statistics of all players for each game. O-Mopsi also preserves statistics of each individual player and shows a ranking chart of top players. This feature also accumulates badges and points for every player. Thus players get more engaged and competitive.

Additionally, searching for the location of targets is similar to solve a puzzle. Although targets are clearly visible, however, they might not be easily accessible. Even if a target is very easy to navigate, then also people might get confused with the presence of similar objects next to it. From the feedback of players as mentioned in **[I]**, we find that players mostly enjoy finding targets. Hence, O-Mopsi might be more motivating if the games provide a diversity in the number of targets, total length, and difficulty of finding targets.

1.5 RESEARCH CHALLENGES

O-Mopsi was mainly introduced to make people exercising, to provide a sightseeing guide, and to provide an outdoor educational puzzle game. However, we find that the application has several aspects and challenges need to be answered to make it more useful. Similar to this, there are several design parameters, which are needed to be considered for any LBG or a trip planning application.

For this thesis, we choose to study O-Mopsi game and the challenges it involves. We study different ways of evaluating game instances and human players' performance. We mainly focus on the following three aspects:

- Playing strategy
- The difficulty of game instances
- Human performance

The rest of the thesis is organised as follows. In chapter 2, we explain all the features of O-Mopsi game. We also describe the motivations and challenges of O-Mopsi. In chapter 3, we explain strategies to play O-Mopsi game, especially, strategies to select a good starting point for this game. We present the players' performance to evaluate the necessity of a strategy. In the following chapter, we determine the difficulty of a game instance for human solvers and computer algorithms. In chapter 5, we propose a simple and fast local search algorithm to optimise the visiting order of targets of an O-Mopsi instance. Additionally, we study and compare the quality of several local search operators. In chapter 6, we summarise the contributions of this study. Lastly, we conclude the findings of this study in chapter 7. The individual publications appear at the end of the thesis.

2 O-MOPSI AND ITS CHALLENGES

2.1 CLASSICAL ORIENTEERING AND O-MOPSI

In classical orienteering, organisers place specially designed flags throughout a forest or a hilly place as targets. Players need to visit those targets in a certain order with the help of a paper map and a compass (Figure 3). O-Mopsi replicates this activity by replacing the paper map with an online map and the compass with GPS on the player's phone. The first significant difference between classical orienteering and O-Mopsi is that the O-Mopsi targets are real-world objects (Figure 4), such as buildings, statues, and other pieces of architecture, which seem easier to find than the targets generally are being used in classical orienteering. O-Mopsi targets are not visible on the application screen before the player starts playing. When play starts, all targets become visible in the default zoom level and the player has to visit all targets to finish the game.

Another significant difference is that players of O-Mopsi can individually start playing at anytime and choose the order in which they visit the targets. As the order in which targets should be visited is predefined in classical orienteering, the task is relatively easy. In O-Mopsi, players have to plan the order and find the target concurrently. Therefore, it requires the use of both route planning and navigational skills on the fly.

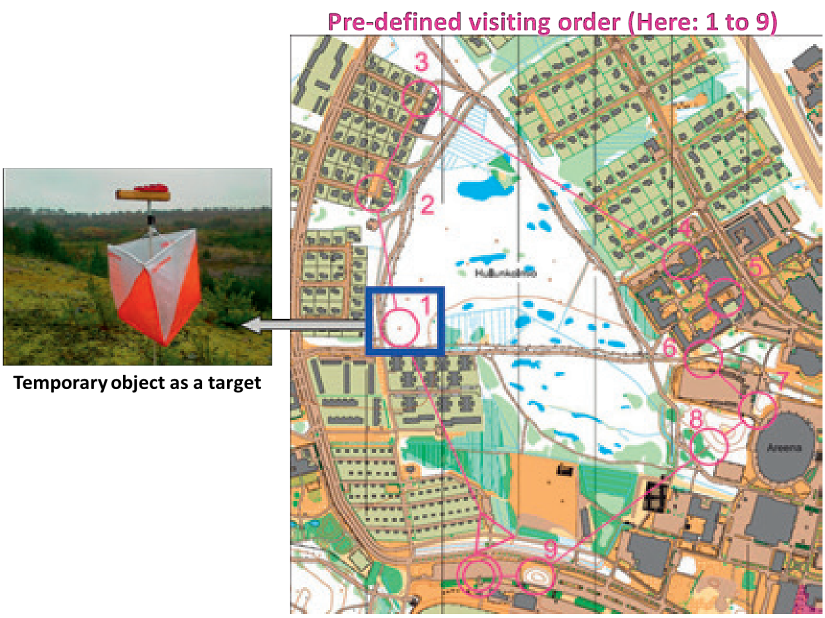

Figure 3: Setup of a classical orienteering game.

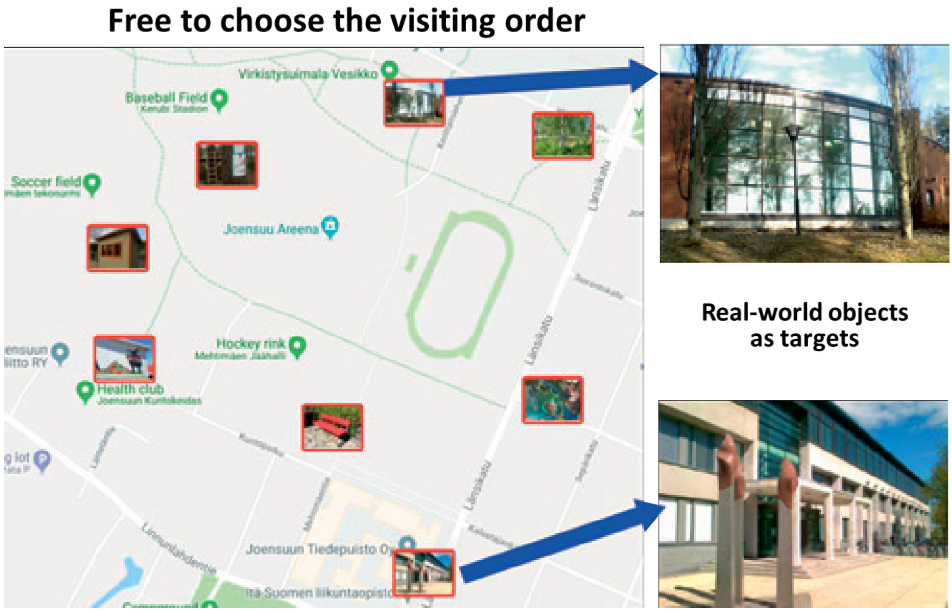

Figure 4: Real-world targets of an O-Mopsi game.

2.2 GAME PLAYING

The O-Mopsi application is available for Symbian, Windows, iPhone, and Android phones. Wan (2014) provided a detailed explanation of the game-playing rules, a description of the O-Mopsi website, and delivered game creation tips and techniques. At the time of writing (10th September 2019), the application contained 158 different game instances involving various places around the world. On average, each game contains 12 targets within a locality. Players have to walk around that locality and visit all the targets in the game. The player who visits all targets of a game in the shortest time is the winner of the game. O-Mopsi games mostly involve short distances, so players have only to walk an average of 3 to 4 km per game. O-Mopsi does not show the locations of the game's targets until the play mode is on and the timer starts (Figure 5). When the play mode is on, the application reveals the names and pictures of the targets on a map and the GPS on the phone records the location of the player.

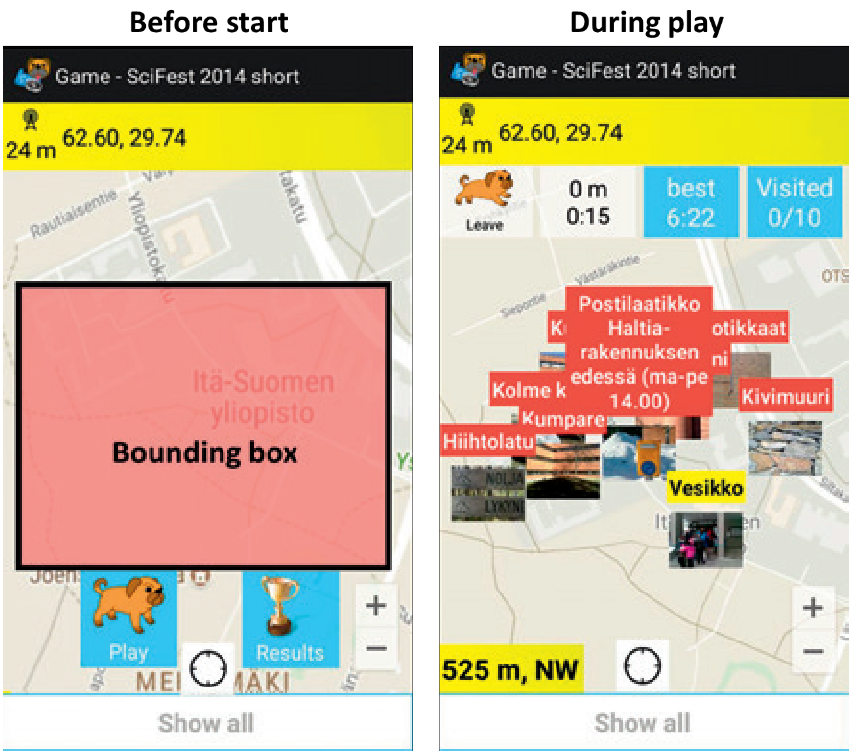

Figure 5: Screen display of an O-Mopsi game before and after play starts.

O-Mopsi also shows the distance to the nearest target and indicates when this distance is reduced to 500 metres. Additional audio guidance indicates that the player is approaching a target. The frequency and pitch of the sound increases as the player nears the target. When the distance from a target is less than 20 metres, the application automatically removes the target from the map and a *Ta-da* sound notifies the player's accomplishment.

Generally, playing time and distance are proportional in O-Mopsi game playing. Hence, if players want to visit all the targets as quickly as possible, they generally may have to shorten their travelling distance. This makes O-Mopsi a game in which a player needs to visit a set of targets using the shortest possible path.

The travelling salesman problem (TSP) is a computationally hard problem where one has to visit a set of predefined targets using the shortest possible way without visiting a target more than once. For a closed-loop TSP, one has to return to the starting target; however, for the open-loop case, no return is necessary. Thus, O-Mopsi produces an open-loop TSP (Figure 6).

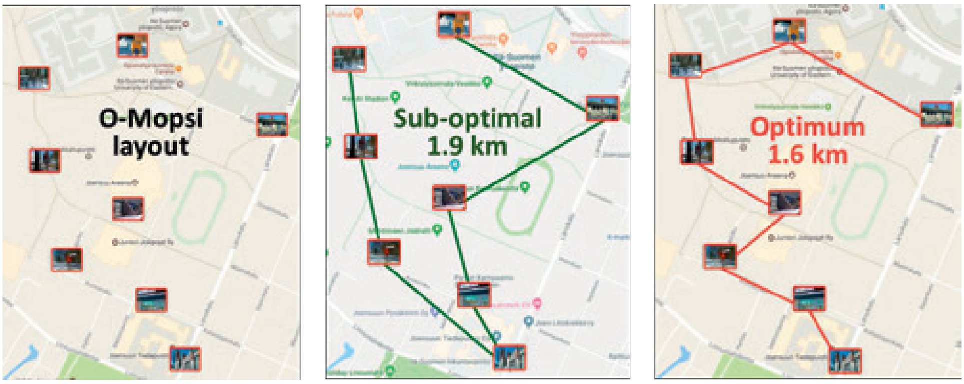

Figure 6: O-Mopsi as an open-loop TSP.

2.3 MOTIVATIONS

According to the game's rules, the player who completes the game in the shortest time is the winner. This competition is what motivates many people to play the game. We found that a player played a single game several times to improve his or her result, especially if others have completed the game in lesser time. Figure 7 shows that players are ranked by their time.

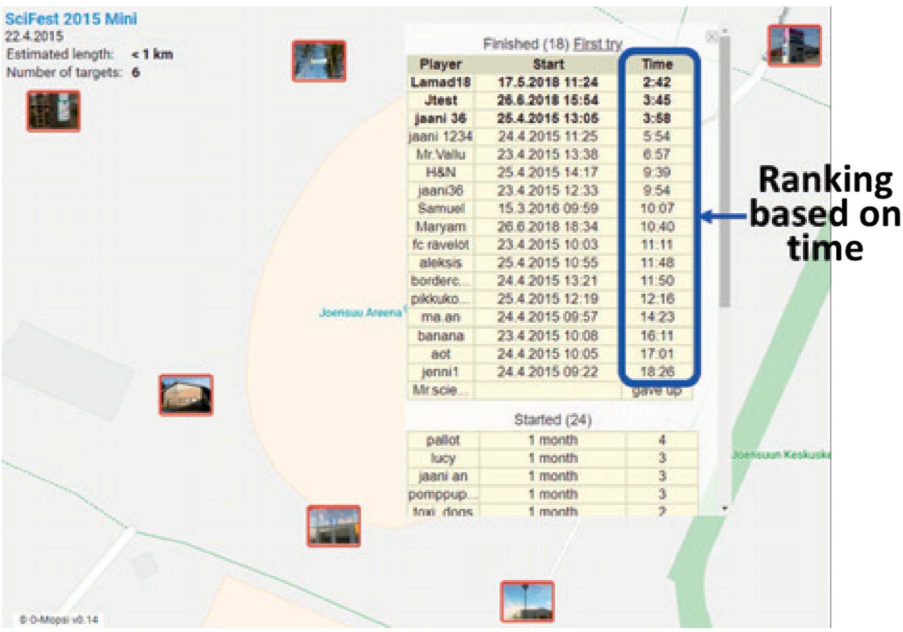

Figure 7: Completing the game in the shortest time as a motivating factor for players.

Players hardly find (one out of five players on average) the optimum solution to the TSP of O-Mopsi games. However, O-Mopsi allows for non-optimum solutions. Therefore, players can complete a game after choosing non-optimal links between targets, which we call *mismatches*. Along with the time, we also test players' skill in solving the TSP. Those players who have fewer mismatches are more skilled at solving the TSP. However, even by reducing the path length, a player might not win due to low speed. Therefore, analysis based on time is not always motivating for slower players. Competition based on TSP solving is more motivating for a certain group of people. Figure 8 shows that the rankings based on time are not the same as the rankings based on solving the TSP. Therefore, the slowest player can rank first with regard to another deciding parameter.

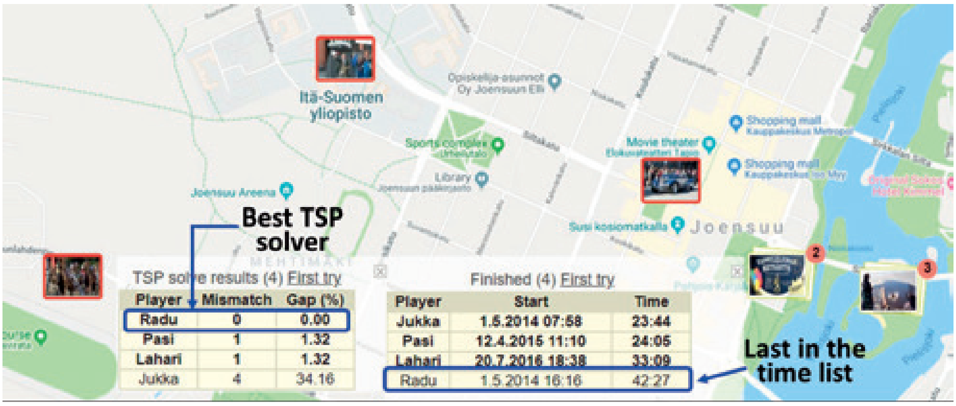

Figure 8: The fastest player might not always be the most skilful player.

Lastly, learning from the targets is also motivating. People might not always be competing. They might want to enjoy roaming around a city and getting to know places around the city. O-Mopsi targets can reflect the historical, cultural, and scientific aspects of a place. These types of targets could contribute significantly to the educational gaming industry. Figure 9 shows examples of O-Mopsi games designed as city tours.

Figure 9: O-Mopsi games as city tours.

2.4 CHALLENGES IN O-MOPSI

According to the O-Mopsi game rules, players have to visit all the targets, but they do not have to return to the starting point. Hence, the TSP behind O-Mopsi is an open-loop TSP (Figure 10). The solution to a closed-loop TSP is a tour, whereas the solution to an open-loop TSP is a path with terminal targets.

Figure 10: O-Mopsi as an open-loop TSP.

Selecting the starting point in a closed-loop TSP is irrelevant for its solution. Any random choice works as it is the order that is vital, not the starting point. Different choices do not produce different solutions in a closed-loop TSP. In contrast, the starting point of an open-loop TSP can be crucial for finding its solution as the solution changes depending on the starting point (Figure 11). In the first example, the open-loop optimum solution has a path length of 1.4 km. However, if we force the algorithm to find the optimum solution starting from a specific target, the output changes. The path length increases in this case (1.5 km in the example). In the second example, the path length increases by 200 m (0.9 km to 1.1 km) when a less adequate starting point is chosen. Hence, selecting the best starting point is important.

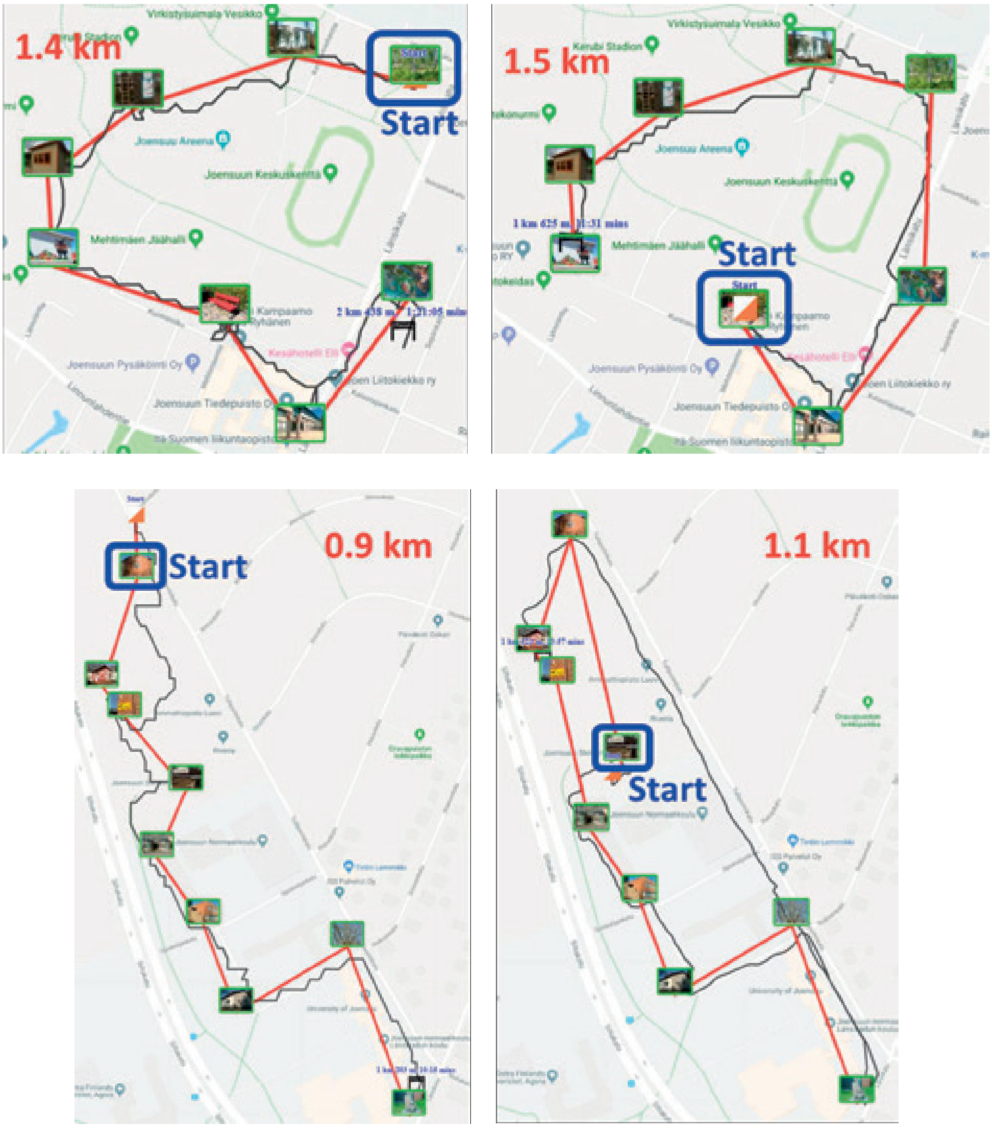

Figure 11: Different optimum paths starting from different targets

Before a game starts, the O-Mopsi application shows only the bounding box over the targets (Figure 5) instead of the exact locations of the targets. This box is a rectangle placed over the targets spanning most longitudes from left to right and most latitudes from north to south. Therefore, at least two targets locate over the edges of the box and the rest are inside the box. We call this the *game area*. When a player starts playing, targets become visible. Therefore, before starting a game, players cannot plan the order in which they will visit the targets. They can use some strategy to guess the good starting position from the game area. In most cases, players start from a random location and sometimes that random location could be the luckiest choice; hence, next to the good starting position. It could be reversed; that means players can choose to start from the most unlucky position. Hence, educated guesses might enhance performance. In **[II],** we evaluated the optimum start position for algorithms and human players.

The O-Mopsi playing record shows that a single player plays multiple games. For example, Figure 12 shows that *Jukka* played two games (*Rantakylä Tour* and *Autumn Moon*). The skill level of a particular player might be always the same. Therefore, it seems logical that a player's performance should be similar for all the games he or she plays. However, the example of Figure 12 shows that Jukka's path has seven links that differ from the optimum in the first game and no links that differ from the optimum in the second game. The obvious question is why his performance is inconsistent. The probable answer is that the two games have different levels of difficulty.

Figure 12: Difficulty levels affect human performance. Here, the player Jukka chose seven sub-optimal links in the first game but made no error in the second game.

SciFest is a festival that is held annually in Joensuu, Finland to promote discoveries in science, technology, and the environment. Usually, the workshop takes place inside a large hall. Research groups from local universities, other research organisations, local companies, and start-up ventures can set up stalls and demonstrate new inventions. Mainly students and teachers of most schools in Joensuu and near-by municipals attend to gain knowledge. Several O-Mopsi games have been created for SciFest every year. School students play these games every year. Figure 13 shows O-Mopsi stands at SciFest 2016 and 2018.

SciFest 2016 SciFest 2018

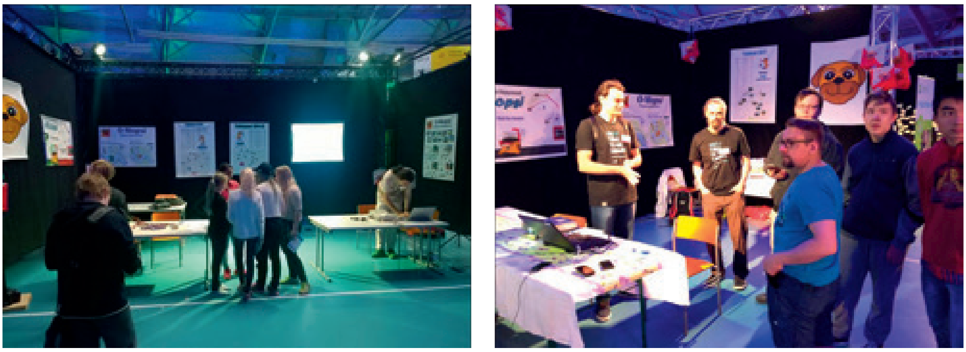

Figure 13: O-Mopsi demonstrations at SciFest 2016 and 2018. Usually, school students come and learn to play. After playing, they provide feedback about the game. Teachers and other interested people come and try to learn the scientific aspects of the application.

It is worth studying SciFest games to analyse human performance as these games have a lot of players. We found that some players attend the festival to play in consecutive years. Table 1 shows the performances of O-Mopsi players at SciFest from 2011 to 2018.

Table 1: Players' performances in SciFest games during the period 2011–18

Year	Game statistics		Player statistics				
	Targets	Length (km)	Started	Finished	Solved	Avg. mis-match	Avg. gap
2018	8	1.4	35	12 (34%)	1	0.6	6.7%
2017	14	1.4	30	18 (60%)	0	6.9	22.2%
2016	13	1.5	67	19 (28%)	0	3.3	16.2%
2015	14	1.2	45	15 (33%)	0	5.5	16.4%
2014	10	1.0	64	28 (44%)	7	3.9	16.3%
2013	16	1.1	6	2 (33%)	0	8.0	23.2%
2012	7	0.5	72	25 (35%)	7	1.4	11.2%
2011	5	0.6	33	9 (27%)	5	0.8	1.8%
Avg.	**10.9**	**1.1**	**44**	**16 (36%)**	**2.5**	**3.8**	**14.3%**

Table 1 lists the SciFest games from 2011 to 2018. We show the number of targets (Targets) and the optimum length of the games (Length) as the game features. To demonstrate players' performance, we note the number of players who started the game (Started), the number of players who finished the game (Finished), the number of players who found the optimum solution (Solved), the average number of errors in selecting optimum links (Average mismatch), and the average route length difference from the optimum (Average gap). We find that, along with the game statistics, players' performance also varies from one game to another. This raises the need to assess the difficulty of a game instance, which is another challenging aspect of O-Mopsi. Being able to estimate the difficulty level of a game instance can guide both game creators and players in choosing which game to play. In **[III]** we discuss difficulty levels and how to measure them further.

Figure 14 shows a typical example of two different players' paths for a single game. The paths are different for these two players. To be able to compare the quality of the order in which these two players visited the targets, we need a reference solution. Either the overall optimum solution or the optimum solution from a player's starting target can be used as the reference solution. A reference solution can provide players with an estimation of the length of the path of the game before playing. Solving the open-loop TSP to find the optimum path is another challenging task in O-Mopsi.

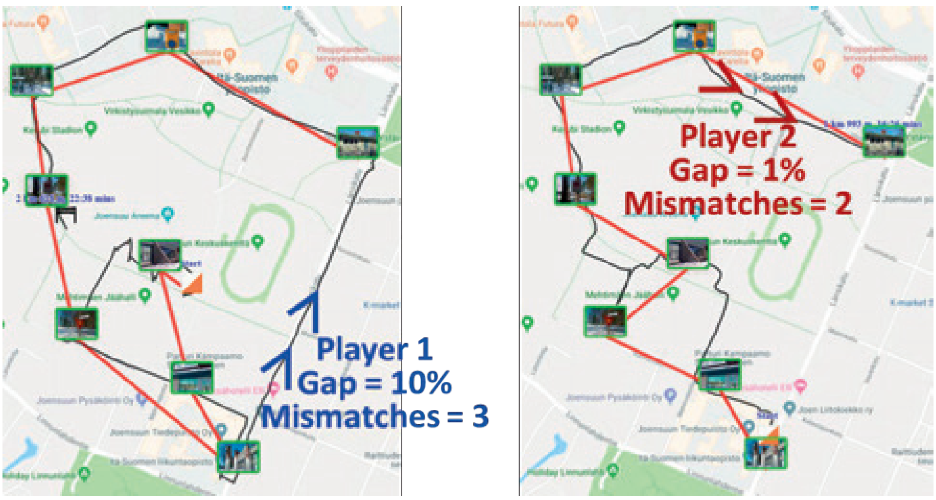

Figure 14: To compare different paths of different players we need a reference solution.

Players usually find non-optimal solutions, and the optimum tour can be used to analyse the number of mismatches of a player. A player's travel distance can also be compared with the optimum path length. The percentage difference between these two distances, which we call the *gap*, is another measure that can be used to study the player's performance. In the example shown in Figure 14, the first player had three mismatches with a 10% gap from the optimum, and the second player had two mismatches with only a 1% gap from the optimum. Based on the gap values, player 2 performed better.

The solution to a closed-loop TSP is a tour; however, the solution to an open-loop TSP is an open path. Hence, a TSP of size *N* contains *N* links in its closed-loop solution and *N-1* in its open-loop solution (Figure 15). However, the optimum solution to an open-loop TSP is not usually achievable by removing the largest link from the optimum solution of the closed-loop TSP (Figure 15).

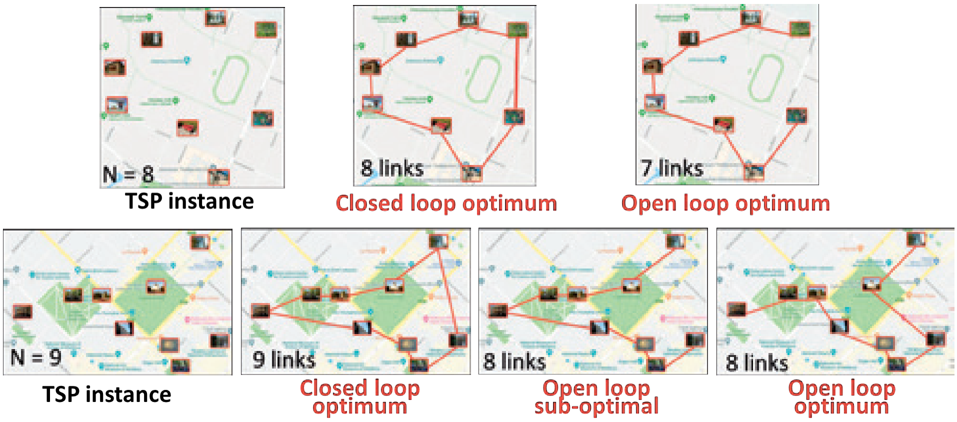

Figure 15: Difference between the open- and closed-loop TSP.

The size of O-Mopsi games varies from 4 to 27 targets. Smaller games can easily be solved with a simple branch-and-bound algorithm, although larger games are more challenging and can take hours to solve with a simple branch-and-bound method. The nearest neighbour algorithm produces solutions with an average gap of 25%. In **[IV]** we introduce a simple algorithm for solving the TSP.

Individual four studies (**[I]**, **[II]**, **[III]**, and **[IV]**) collectively form this thesis. Figure 16 briefly illustrates the objectives of individual studies. In **[I]**, we studied the research challenges present in O-Mopsi. In **[II]**, **[III]**, and **[IV]** we thoroughly discussed those research challenges sequentially.

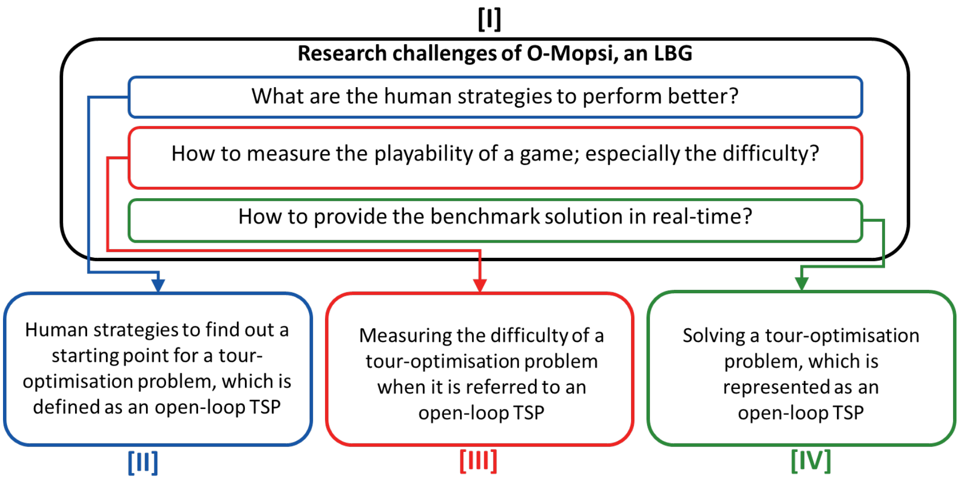

Figure 16: Interconnection between individual studies presented in this thesis.

3 PLAYING STRATEGIES FOR SOLVING THE OPEN-LOOP TSP

The solution to a closed-loop TSP is a cycle; in contrast, the solution to an open-loop TSP is a path with two terminal points (Figure 17). For a human solver, choosing the correct terminal points, especially the starting point, is very important. Unlike a closed-loop TSP, the shortest path in an open-loop TSP depends on the starting point. Either of the terminal points produces the optimum result. Hence, these are the *best starting points*. For this study, all points other than the best starting points are defined as *inferior points*.

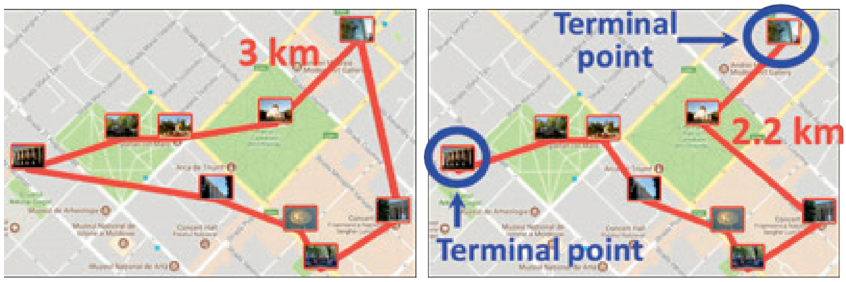

Figure 17: Optimum solutions to the closed- and open-loop TSP

Starting from an inferior point will lead to a sub-optimal solution, henceforth, costs extra path length. The *potential* of an inferior point is measured by how much extra path length it concedes. In the example shown in Figure 18, the first player started from a better position than the second player, so the first player's path is 50% shorter than the second player's path.

Selecting the best starting point is more difficult in O-Mopsi as the application hides the targets and restricts the player from planning beforehand. The player has to plan it in real-time when the game-playing mode is on. Although finding the optimum is not mandatory in O-Mopsi, players aim for the optimum path to reduce the path length.

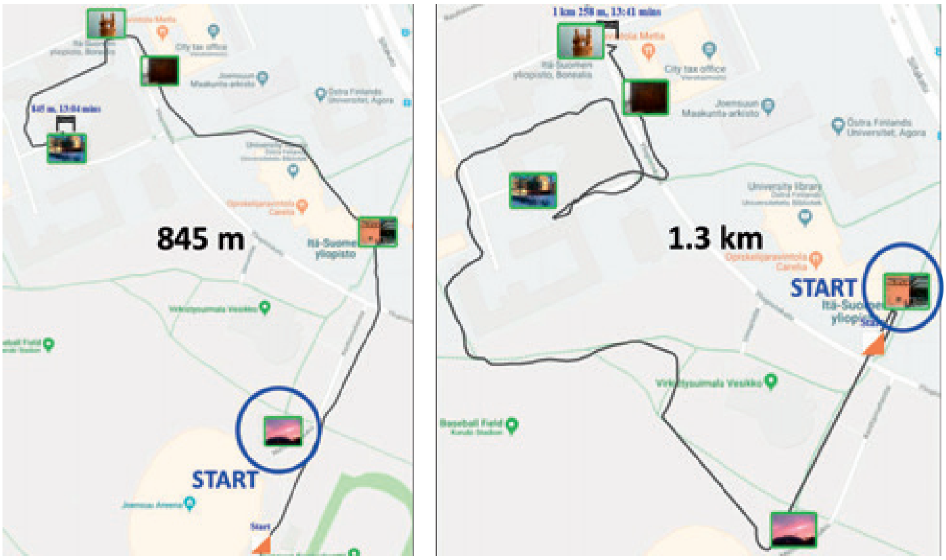

Figure 18: Starting from an inferior position can increase the path length by more than 50%.

Several studies [De Choudhury et al., 2010, Gionis et al., 2014, Bolzoni et al., 2014, Mor and Dalyot, 2018, and Agarwal et al., 2018] have discussed how to plan and optimise a tourist's tour. Mostly these studies modelled the optimisation problem for tour planning as an *orienteering problem* [Vansteenwagen et al., 2011]. For orienteering problem, people need to visit as many targets possible within a limited time. Hence, it is not mandatory to visit all the targets. Besides, the terminal targets are fixed beforehand in this case. However, our problem needs a solution to find out the best starting point. eCOMPASS [Gavalas et al., 2015], a tour planner that provides an optimised tour for tourists adds tourist's current location as the starting location in the optimised path. Still, it does not give the solution to find out the best starting point of a tourist's tour. In **[II]** we study how an exact algorithm can be used to select the starting point based on the boundary box of the instance.

Researchers have studied human performance in solving the TSP and modelled algorithms based on the approaches humans use to solve a TSP. Graham et al. (2000) showed that the time needed by a human to solve a TSP is linearly or near-linearly proportional to the problem size of the instance. They proposed a hierarchical algorithm that closely simulates a human solving approach. This was later refined by Pizlo et al. (2006).

Several researchers have claimed that humans prefers to solve a TSP using a global to local approach, such as first generating a *convex hull* with nodes and then modifying that convex hull curve to generate the solution [MacGregor and Ormerod, 1996, MacGregor et al., 1999, Macgregor et al., 2000, MacGregor et al., 2004, MacGregor et al., 2006]. Although O-Mopsi contains open-loop problems, players still start from a point on the convex hull polygon in more than 50% of cases.

Other studies have found that humans prefer a local to a global approach to solving a TSP; this correlates to *nearest neighbour technique* [Vickers et al., 2003a, Vickers et al., 2003b]. O-Mopsi suggests the nearest neighbour path between targets during playing. However, this hypothesis might not be so effective for open-loop cases as the nearest neighbour technique can produce solutions that are significantly larger than the optimum. Figure 19 shows that the nearest target (neighbour) strategy results in a path that is 25% longer than the optimum.

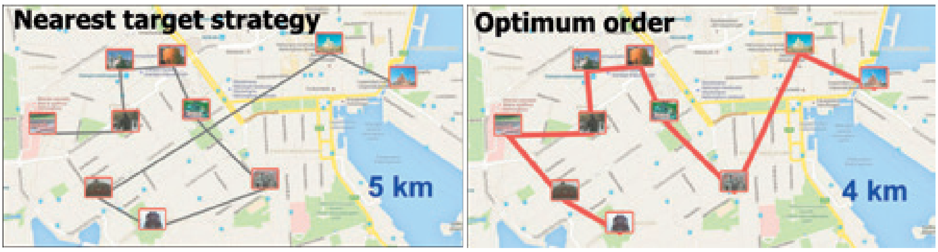

Figure 19: Following the nearest target strategy can add 25% extra length to the path.

Van Rooij et al. (2003) proposed the *crossing avoidance* hypothesis as the nearest match to human performance. This approach is not easy to follow in O-Mopsi. During play, especially in the outdoors, keeping track of whether the path crosses itself becomes difficult as O-Mopsi does not show the already travelled route on the map.

Although these studies address how humans solve a TSP, we do not know how humans select a starting point for an open-loop TSP. In **[II],** we collected a set of data of selected starting points in open-loop TSPs by a group of computer science students.

Targets of O-Mopsi games are not visible before the game begins. For our calculations in **[II]**, we considered the game area, which is the game bounding box containing all the targets. MacGregor (2012) classified the nodes of problem instances into two classes: boundary nodes and interior nodes. He also claimed that human prefers to start from a boundary node. To obtain a more detailed classification, we divided the bounding box, which we referred to as the game area, into a 5 x 5 regular square grid and categorised the grid cells into four regions (*corner, long edge, short edge,* and *middle*) as shown in Figure 20.

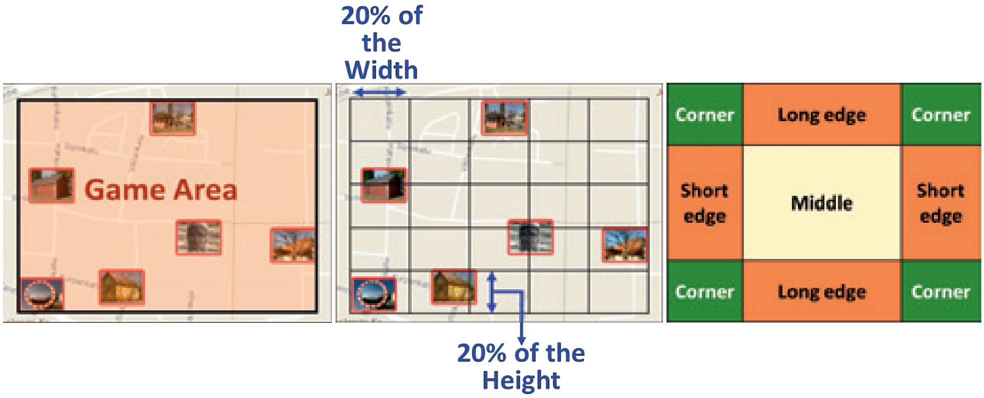

Figure 20: A grid of 25 cells making up the corner, long edge, short edge, and middle segments of the game area

3.1 HOW AN ALGORITHM STARTS

An exact solver always gives the optimum path. To find the optimum path, we used Concorde solver [Applegate et al., 2011]. However, Concorde is only for closed-loop cases. Following Papadimitriou (1977), in **[II]**, we add a phantom node to the open-loop instance and solve the closed-loop problem using Concorde. Thereafter, we remove the phantom node from the solution to recover the open-loop solution. We use the game instances of the O-Mopsi dataset,[4] which consists of 147 open-loop TSP instances with sizes varying from 4 to 27. Thus, for each game, we calculated the optimum path. We then determine the specific regions where the terminal points of each optimum path belong to. We found that corners are the most likely regions for terminal points (Table 2).

In Table 2, we compare our observation with the *a priori* probabilities of the grid cell regions. The a priori probabilities of a region are computed by calculating the ratio of the number of grid cells in that category to the total number of grid cells. Table 2 also shows that although the middle region has the most number of cells, optimum paths do not usually start or end there. Therefore, we can conclude that the larger the area of the region does not help to fit more terminal points. The results also show that optimum paths also start from boundary points, which are the points from which humans prefer to start [MacGregor's (2012)].

[4] http://cs.uef.fi/o-mopsi/datasets/o-mopsi/

Table 2: A priori and observed probabilities of a given cell containing a terminal point

	Cells	Probability	
		A priori	Observed
Any corner	4	16%	46%
Any short edge	6	24%	30%
Middle	9	36%	7%

O-Mopsi players can start a game individually at any time. They can only see the game area and they can move freely any places of the game area before starting a game. Therefore, they might start from any random location in the game area instead of a target. Therefore, an extra distance from the starting position to the first visited target adds to the total path length (Figure 21).

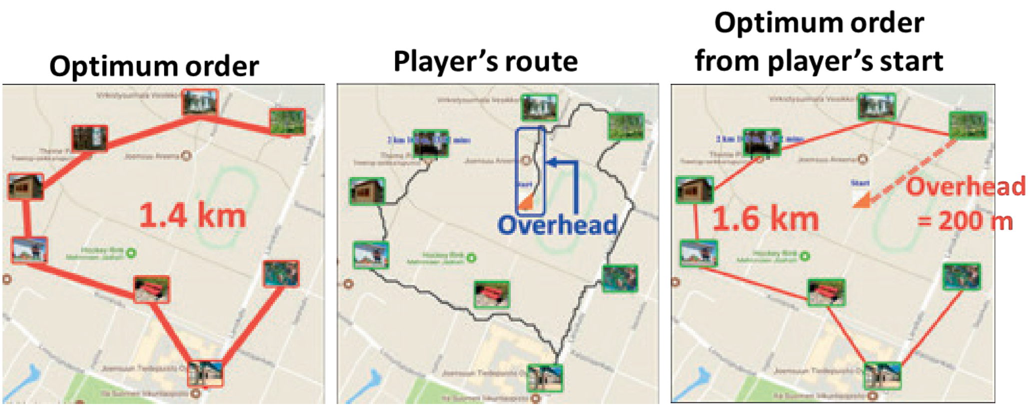

Figure 21: Players can start from any location. However, the distance from the starting point to the first target he or she visits must be added to the total path length. We add this extra distance to the optimum length from the player's starting point.

We analysed the probability of each region of the grid being the best location to start. We fixed the centre point of each category as the start point and find the optimum solution. Here again, we modified the Concorde solver to find the fixed-start optimum solution. As shown in Figure 22, in this case, we first identify the fixed starting point and add a large constant length to that target. This increase its distance from all other points. We then add the phantom node to make it a closed instance. We solve the instance using Concorde, remove the phantom node, and, finally, restore the start point to its original position. This enables us to find the open-loop optimum solution with a fixed-start using Concorde in all cases.

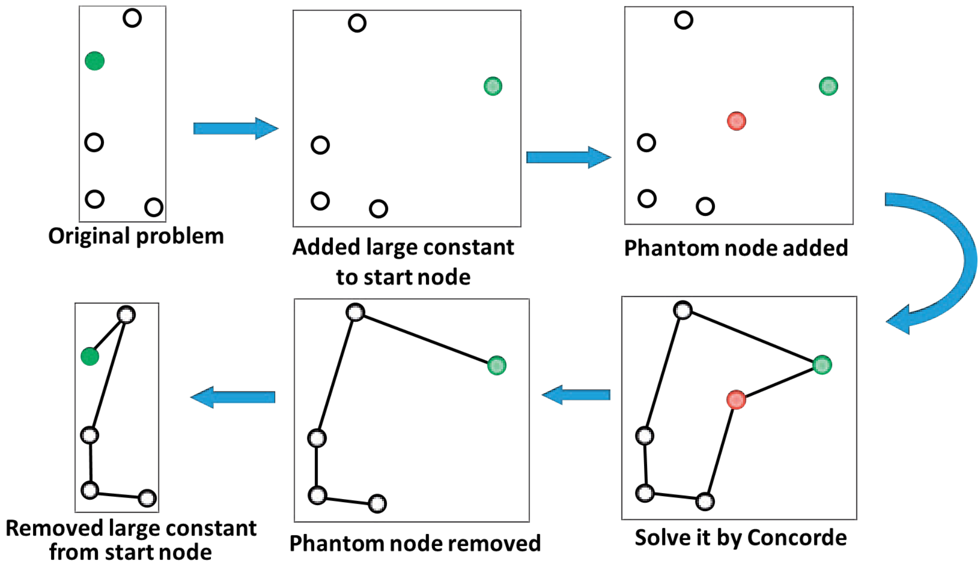

Figure 22: Finding the optimum order from a fixed starting point using Concorde (the green dot is the start location selected by the player, and the red dot is the phantom node that connects the two terminal nodes). [II]

The shortest of all fixed-start solutions for each of the four grid cell regions is defined as the best, and the longest is defined as the worst. In this case, our results again show that the corners are the most efficient cells to start with as starting in these cells offers the highest probability (32%) of finding the optimum (Table 3). However, corners are also risky choices as they are highly probable (45%) to produce the worst fixed-start solution. In a particular game, one corner can provide the best fixed-start solution, while another corner might produce the worst fixed-start solution. Therefore, corners are risky choices. Among all four categories, the least risky region to start in is the short edge, with a 23% probability of finding the best solution and only a 6% probability of finding the worst solution.

We also evaluated the potential of a random point in the game area. Table 3 shows that the probability of finding the optimum solution from a random point (3%) was even worse than the probability of finding the optimum by starting in the middle region (9%). Therefore, following a strategy is always better than selecting a random point to start.

Table 3: Comparison of different starting point selection strategies

Region	The probability of finding the best solution from the region
Random	3%
Middle	9%
Any corner	32%
Any short edge	23%

The Concorde solver produces the visiting order; however, to provide a path, the algorithm needs a routing technique. The provided path should simulate a human path as much as possible. We studied the bird's path using Haversine distance and the distance of the walking-road path using *Open Street Map* (OSM). The bird's distance would be much different from the actual walking distance when the game is designed in an urban area. In contrast, when a game is designed in a park or campus area, players can follow the shortcut path that does not consist in the road path. Therefore, neither of these is a perfect approach (Figure 23), our method uses the bird's path as it correlates slightly higher (bird's path: 0.95, walking path: 0.93) than the walking path of a human.

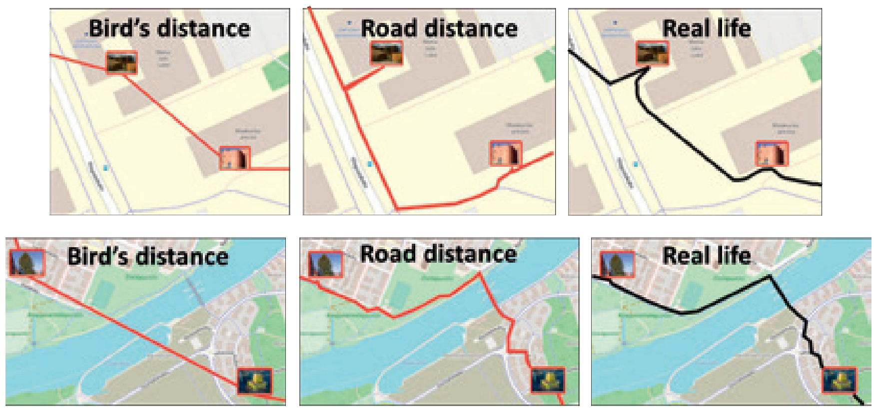

Figure 23: Cases when the bird's path gives a more realistic estimation (above) and when the road network gives more realistic estimation (below) of the distance travelled in real life.

3.2 HOW A HUMAN STARTS

A set of student volunteers helped us to collect results to study human performance. We collected human results for two types of setups for a set of TSP instances. In the first setup, all targets are visible to the volunteers, whereas in the second one, the targets are hidden and only the bounding box is visible to the volunteers during the selection of the starting point (Figure 24).

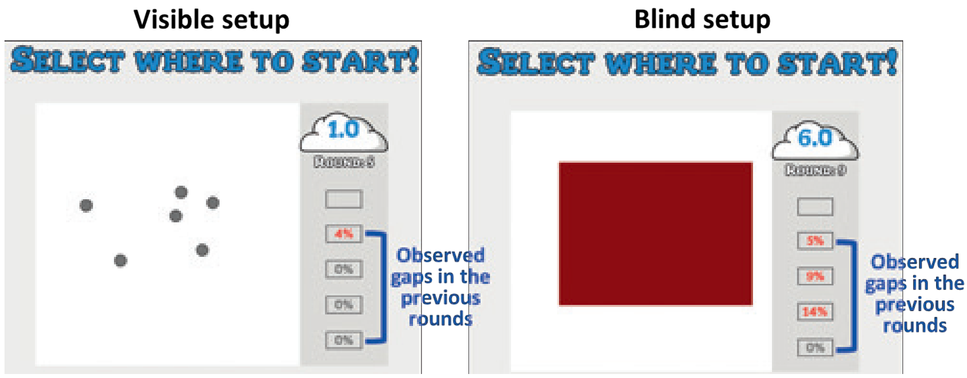

Figure 24: Screenshots of visible and blind setups

For the visible setup,[5] volunteers anticipated and selected the most appropriate starting target. The optimum path from that target was then generated and drew by the computer using the Concorde solver. The gap (%) between the overall optimum length and the optimum length starting from the volunteer-selected target was recorded, as shown in Figure 24, and shown to the volunteer.

We call the other setup, in which the bounding box is given but the targets are invisible, blind.[6] This setup replicates the exact condition of an O-Mopsi game. Here, volunteers anticipated and selected the most appropriate point inside the bounding box to start. Then the computer found and drew the optimum path using the Concorde solver from the nearest target to the selected point and calculated the gap (%) from the overall optimum.

Most volunteers achieved a gap of less than 1% in the visible setup. The average gap for the blind setup was 3.3%, which is worse than the visible setup (Figure 25). Therefore, we can conclude the blind setup requires different skills than the visible setup. We found that some volunteers performed abnormally, i.e. as if they did not understand the objective. They are categorised as the bottom group in Figure 25.

[5] http://cs.uef.fi/~radum/StartPoint/
[6] http://cs.uef.fi/~radum/Blind/

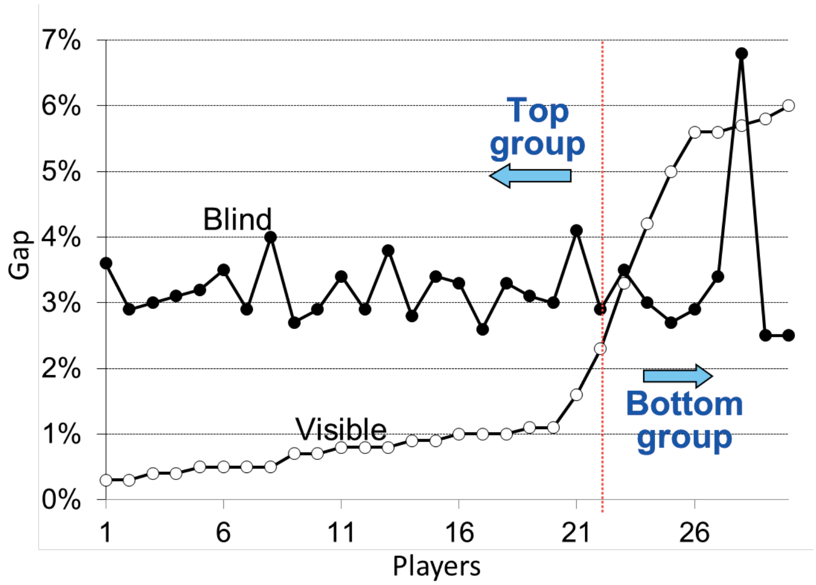

Figure 25: Human performance (gap) in the selection of a start point.

We further investigated *volunteers' strategies* for choosing the starting location. By volunteers' strategies, we imply to indicate that volunteers choose to start from a specific location instead of starting from a random location. For example, from Figure 26 we notice that in the visible setup, volunteers frequently preferred to start from a point on the convex hull and the furthest point from the centre. For a typical problem instance, if a volunteer chooses such a starting point, which is also a terminal point of the optimum path, then the gap value becomes zero. It indicates that the volunteer is able to find out the optimum starting point for that instance. We represent this as the volunteer has solved the problem. Here, we evaluate the performance of a volunteer by the number (%) of instances solved in total. In Figure 26, we refer to this measure as the *amount solved (%)*. We provided 90 instances each for the visible and blind setup to volunteers. From the statistics of each volunteer, we calculate the number (%) of instances where that volunteer chose the furthest point from the centre and a point on the convex hull as the starting point for the visible setup. From Figure 26, we found a distinct descent in performance from the case when a strategy was chosen to the case when a strategy was not chosen. Furthermore, for the blind setup, volunteers chose to start mostly from a corner (Figure 26). However, for the blind case, the performance difference between following a strategy and not following a strategy was not as significant as the visible setup. Still, Figure 26 shows that the volunteers achieved better results when they followed a strategy in the blind setup.

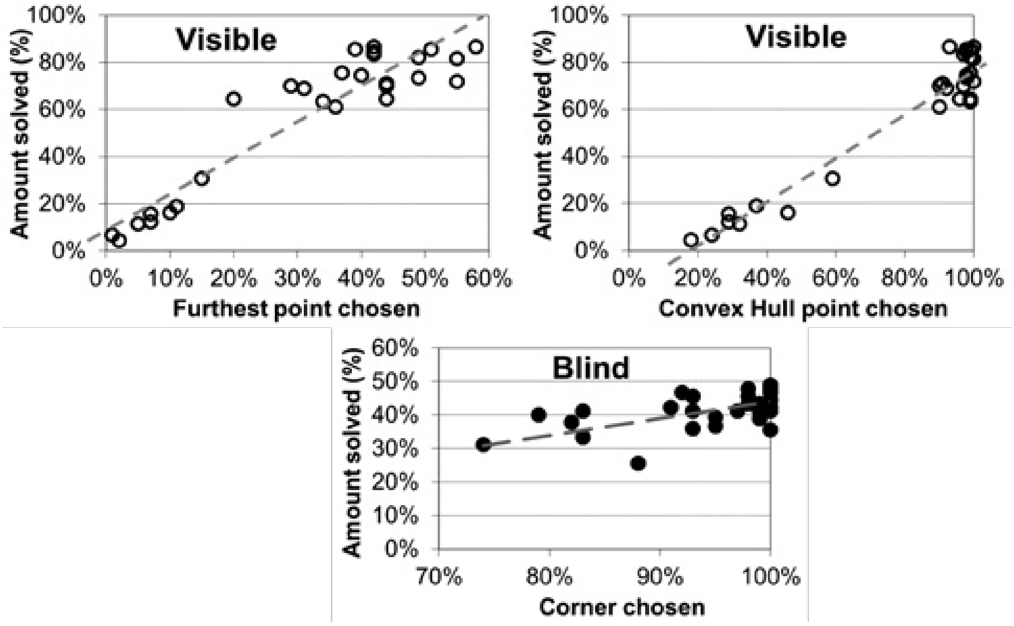

Figure 26: Human performance and strategies for visible and blind setups.

4 PREDICTING THE DIFFICULTY OF A TSP

The execution time for solving the TSP using a straightforward exact solver, such as a brute force algorithm, increases exponentially as the number of nodes increases. However, solving the TSP is also difficult for a human. Humans need visual-spatial abilities to solve a two-dimensional TSP instance [Vicker et al., 2004, Dry et al., 2012]. The O-Mopsi application contains 158 different games as on 10th September 2019. Several players start to play each game and some of them finish. The number of playing instances of a game is the number of finished game-playings by players. We calculate the total number of playing instances for all games and found only 18% of them achieved the optimum solution during play **[III]**. The TSP that O-Mopsi represents is a real-world problem. Hence, it is not only players' TSP solving skills but also their navigational skills that affect their ability to find the solution.

We have another dataset of computer-generated open-loop TSPs named Dots games.[7] Dots games are two-dimensional computer-based puzzles, which do not require navigational skills. Even so, only 15% of all playing instances found the optimum solution at their first trial. Therefore, what makes an open-loop TSP difficult is an open question. For exact computer solvers, increasing size of the instances is particularly problematic. However, not all instances of the same size present the same difficulties. For instance, the three instances shown in Figure 27 do not seem equally difficult. The first one ('Otsola') is straightforward, while the second one ('Hukanhauta 3km') is comparatively complex, and the third one ('Christmas Star') is the most complicated. These three instances show that problems of similar sizes are not equally difficult for humans. Hence, for human solvers, size is not the only factor.

[7] http://cs.uef.fi/o-mopsi/datasets/dots/

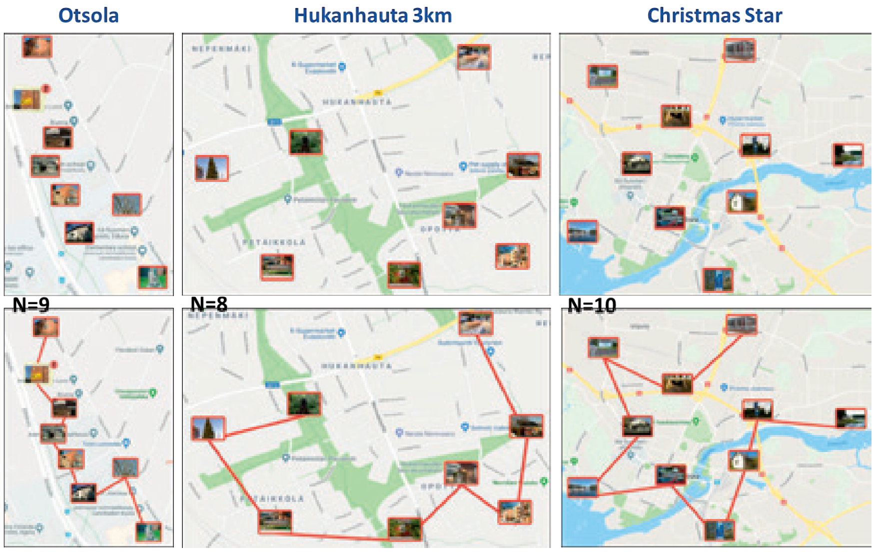

Figure 27: Problem size is not the only factor affecting the difficulty for human solvers of the TSP.

Many researchers have studied human performance in solving the TSP. Graham et al. (2000) and Dry et al. (2006) found that the time required by a human to solve the TSP increases linearly or almost linearly as the problem size increases. A significant amount of research has found that a greater number of points on the convex hull makes instances easier for human solvers [Macgregor and Ormerod, 1996, Macgregor et al., 1999, Macgregor et al., 2000, Graham et al., 2000, Macgregor et al., 2004]. However, contradictory results have also been reported [Vickers et al., 2003 a, Dry & Fontaine, 2014].

4.1 KNOTS IN A MINIMUM SPANNING TREE (MST)

A minimum spanning tree (MST) is another graph problem, the aim of which is to connect all the points of the problem instance into a tree structure with the shortest possible length (Figure 28). It is a relatively easy problem for computers, and polynomial-time algorithms developed by Prim (1957) and Kruskal (1956) can solve it.

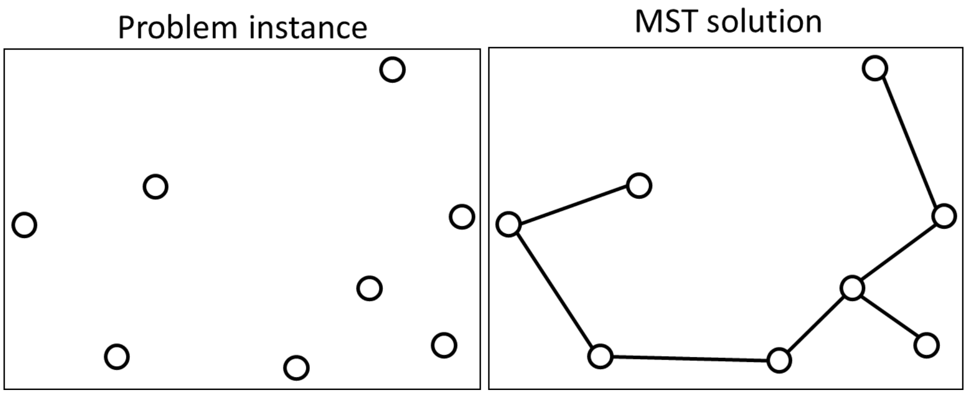

Figure 28: A problem instance and its MST solution

The open-loop TSP and the MST for a given problem instance share some similarities. First, they have the same number of links. The open-loop TSP is a *spanning tree* as it is a connected graph and does not have any cycle. However, it is not the MST until both have the same structure. Therefore, the second similarity is that the Euclidean-length of the MST is always equal to or less than the Euclidean-length of the open-loop TSP. Therefore, for a particular problem instance, the MST solution and open-loop TSP solution might be identical (Figure 29).

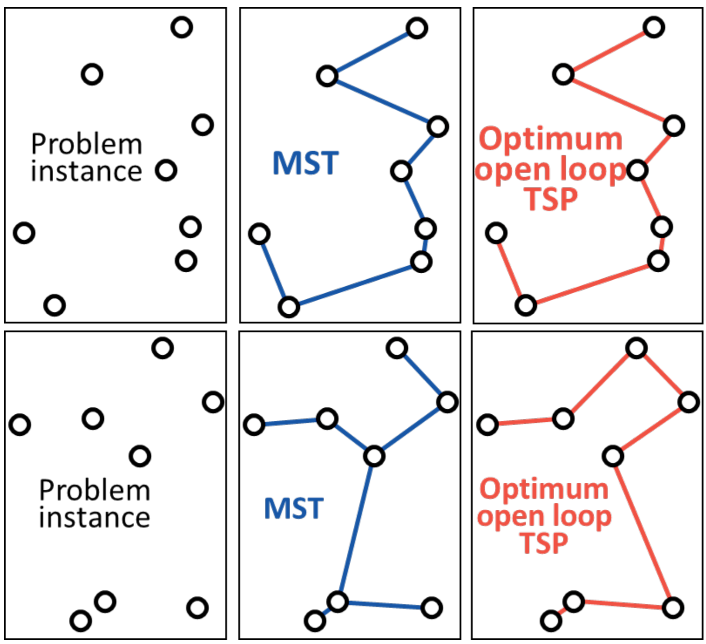

Figure 29: Two instances with MST and open-loop TSP solutions.

The MST is a tree structure so it can have nodes with more than two links associated. We call these nodes the MST knots (Figure 30). There are 0 to $N/2$-1 possible knots in an MST of size N.

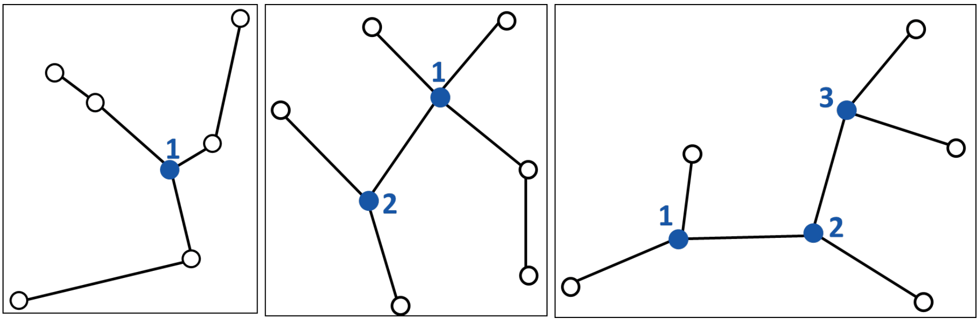

Figure 30: Examples of one, two, and three MST knots in problem sets

Vickers et al. (2004) showed that human performance in solving the TSP and the MST correlates well (0.66). In **[III]** we discovered that a greater number of knots in the MST solution makes the TSP more difficult for a human. We confirm this in Figure 31 by finding MST knots for the three instances shown in Figure 27 earlier. Figure 31 demonstrates that an increased number of MST knots correlates with an increased level of difficulty. We then studied relationships between the number of MST knots and human performance in solving the TSP.

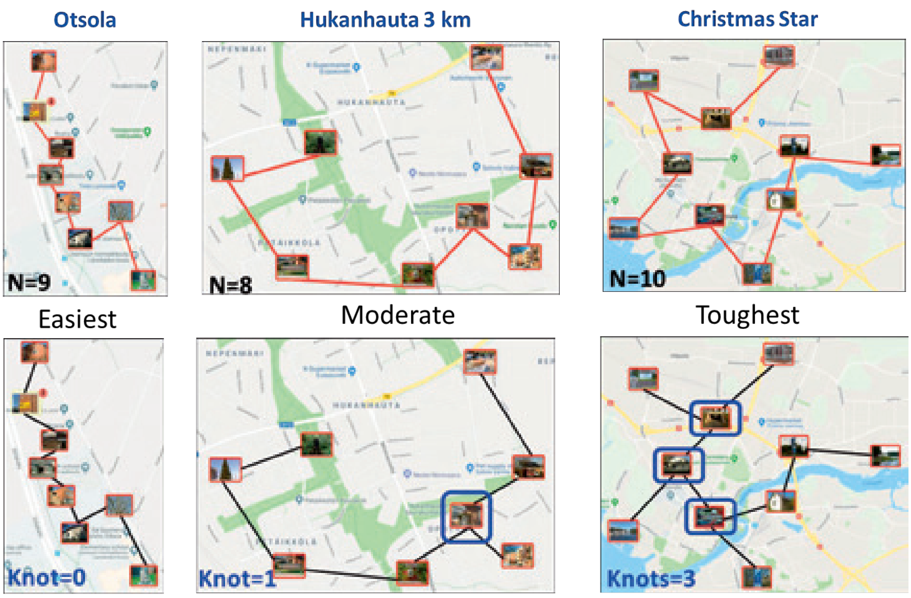

Figure 31: The number of MST knots predicts the difficulty of TSP instances.

We used Prim's (1957) algorithm to compute the MST solution in our algorithm of **[III]**. During the creation of the MST solution, our algorithm keeps track of the number of links being added to each node. If this number exceeds two, we mark it as a knot.

We used the number of MST knots and the normalised number of MST knots in **[III]**. The normalised number of MST knots is the number of MST knots normalised with respect to the problem size to reduce the effect of the problem size (Figure 32).

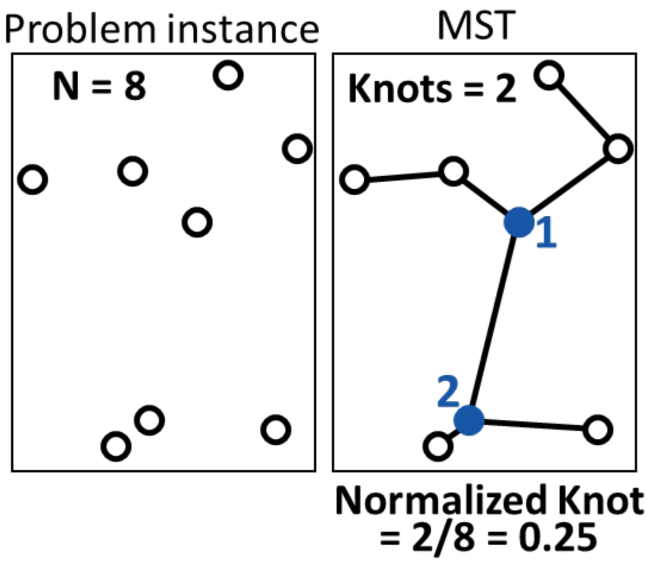

Figure 32: The number of MST knots and the normalised knot

4.2 HUMAN MISTAKES

To evaluate human performance, in **[III],** we considered *human mistakes* and *human playing time*. O-Mopsi allows for non-optimal solutions. Therefore, players can have errors with respect to both order and length. Increased difficulty results in an increased number of errors. Hence, we considered playing error to be a measure of human performance. To avoid GPS errors, instead of working with the players' original route, we noted the order in which players visited targets and computed the Euclidean or bird's paths from these orders. To measure playing error with respect to order, we considered the number of mismatches, which is the number of link differences between the optimum and the visiting order of the player (Figure 33).

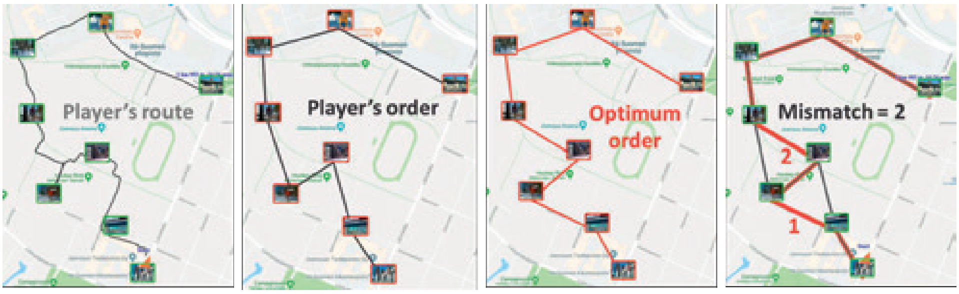

Figure 33: A playing instance in which the player's visiting order contains two mismatches.

However, this measure counts a single fault twice. In the example shown in Figure 33, after making the first mismatch, the player is compelled to visit the rest of the targets in a particular order, which automatically counts as another mismatch. Therefore, we introduced another measure, the *mistake*, which dynamically computes the optimum order along with the unvisited targets after every fault occurs (Figure 34). This measure is more logical for assessing human performance.

Figure 34: Player's visiting order contains one mistake in the same playing instance of Figure 33.

The human *gap* is a measure of human performance in terms of the difference in path length. It is calculated as the percentage difference between the length of the player's visiting order and the optimum order. We consider this measure because the number of mistakes sometimes conveys incomplete or wrong information. Figure 35 shows two examples of game playing with almost the same number of mistakes. This might imply that both games are almost equally difficult. However, the gap shows that the effect of mistakes is significantly different in these two examples.

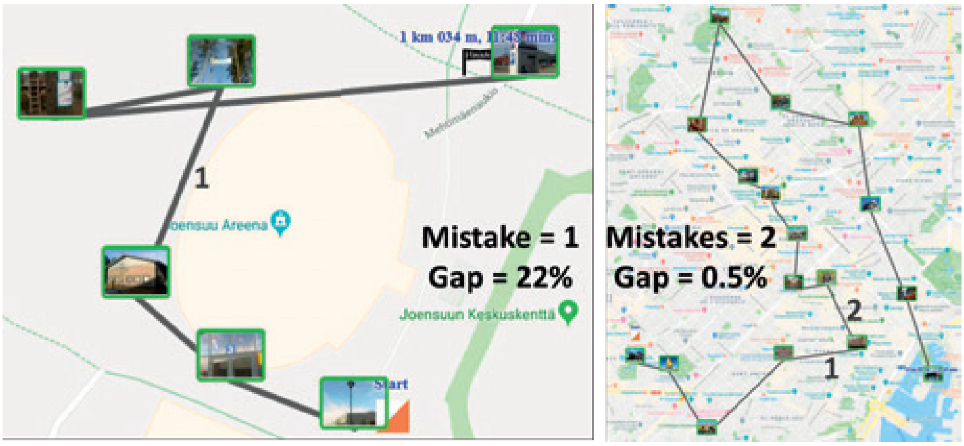

Figure 35: The number of mistakes does not always reveal the significance of the faults. The first example contains only one mistake, but the solution is rather poor (22% gap). The second example contains two mistakes, but the solution is still very close to the optimum (0.5% gap).

4.3 HUMAN PLAYING TIME

Dots games do not allow sub-optimal solutions; therefore, players need to solve these games in one or more trials. More difficult games require more trials; consequently, they require more time to solve. Hence, we considered the playing time of dots games as another measure of human performance. For our experiments, we considered only instances with a playing time of less than 5 minutes.

4.4 EVALUATION OF COMPUTER PERFORMANCE

We added a phantom node to turn our open-loop cases into closed-loop cases and used the Concorde algorithm [Appelgate et al., 2011] to solve it as explained in **[II]**. We measured the execution time for all O-Mopsi and dots games.

4.5 RESULTS

We tested human and computer performance and considered both problem size and the number of MST knots. Table 4 shows that the Pearson correlation between both human and computer performances and problem size is substantial for O-Mopsi (252 playing instances) and dots cases (12124 playing instances). Furthermore, the number of MST knots is also significantly correlated with both human and computer performances.

Table 4: Correlation between problem size and MST knots with human and computer performance.

	O-Mopsi[8]			Dots[9]	
	Human mistake	Human gap	Concorde time	Human time	Concorde time
Problem size (N)	0.46	0.06	0.75	0.38	0.61
MST Knots	0.54	0.16	0.68	0.35	0.56
Normalised Knots	0.44	0.11	0.48	0.13	0.22

Table 5 compares the game and player statistics for SciFest games. We list the number of targets, optimum length, and the number of MST knots as game statistics. For the players' performance, we report average mismatch, average mistake, and average gap. We observe that the number of targets and the number of MST knots explain human performance satisfactorily.

Table 5: Comparison of players' performances with the number of MST knots for SciFest games in the period 2011–2018

Year	Game statistics			Player statistics		
	Targets	Length (km)	MST knots	Avg. mismatch	Avg. mistake	Avg. gap
2018	8	1.4	1	0.6	1.1	6.7%
2017	14	1.4	3	6.9	6.0	22.2%
2016	13	1.5	2	3.3	3.2	16.2%
2015	14	1.2	2	5.5	3.8	16.4%
2014	10	1.0	1	3.9	3.1	16.3%
2013	16	1.1	5	8.0	5.5	23.2%
2012	7	0.5	0	1.4	1.1	11.2%
2011	5	0.6	1	0.8	0.6	1.8%
Avg.	**10.9**	**1.1**	**1.9**	**3.8**	**3.1**	**14.3%**

Human mistakes increase linearly as the problem size increases, as shown in Figure 36. Therefore, our results support the findings of Graham et al. (2000) and Dry et al.,

[8] http://cs.uef.fi/o-mopsi/datasets/o-mopsi/
[9] http://cs.uef.fi/o-mopsi/datasets/dots/

(2006). Human mistakes also increase linearly as the number of MST knots increases. Hence, a greater number of MST knots forces players to make more mistakes. Therefore, the number of MST knots can predict the difficulty of a TSP instance for a human player. However, neither the problem size nor the MST knots can predict the difficulty of the problem instances with respect to the human gap. Computer running time also increases linearly as the problem size and the number of MST knots increase (Figure 36). Therefore, the number of MST knots can estimate the difficulty of a TSP instance for a computer algorithm.

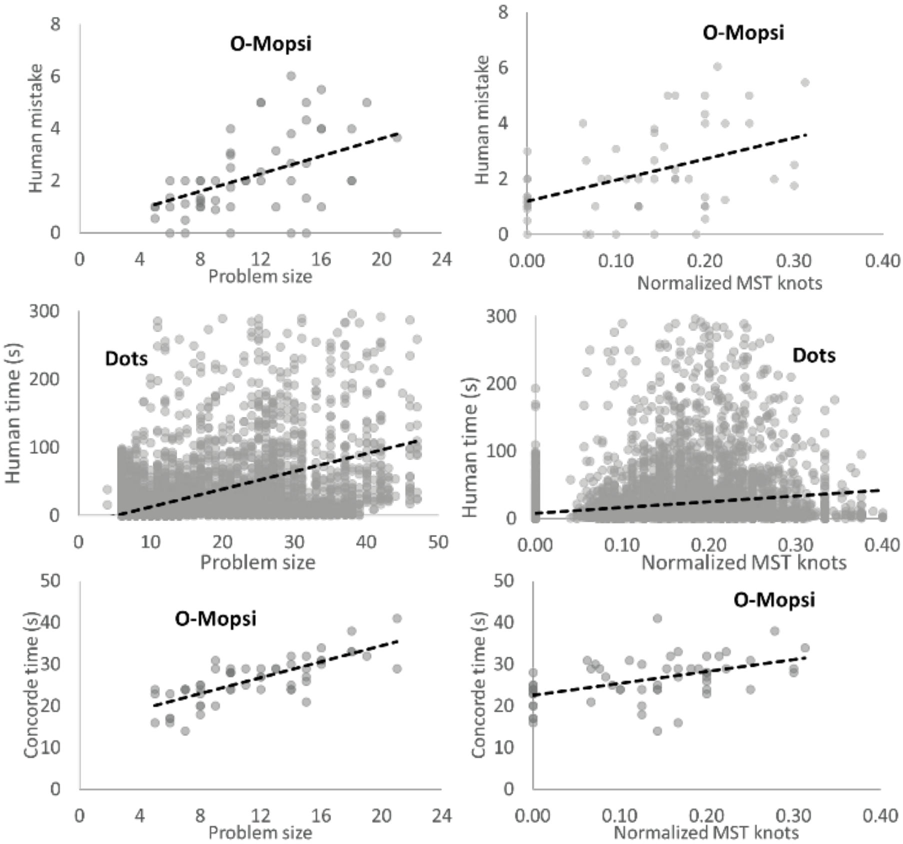

Figure 36: Human and algorithm performance for games of varying problem sizes and the number of MST knots.

5 SOLVING THE OPEN-LOOP TSP

Papadimitriou (1977) showed that both closed and open-loop variants are NP-hard problems. Therefore, the time required to find exact solutions increases exponentially as the number of targets increases [Cormen et al., 2009]. Despite knowing this fact, researchers have been working on exact solvers for the TSP since 1950. As a result, we have several exact algorithms [Dantzig, Fulkerson and Johnson, 1954, Held and Karp, 1972, Padberg & Rinaldi, 1991, Grötschel and Holland, 1991, Laporte, 1992a, Applegate et. al., 1998, Laporte, 2010, Gutin and Punnen, 2006, Applegate et al., 2011] mostly for closed-loop TSPs.

Other researchers have focused on heuristics, which can provide near-optimum solutions in a relatively short time. Local search is one of the simplest yet one of the most powerful heuristic approaches to optimisation. Exact solvers check all possible combinations and finally either produce the best solution or prune the search space efficiently using various criteria. In contrast, local search first constructs a far-from-optimum initial solution and then improves the initial solution iteratively using certain local optimisation techniques. The methods for constructing and improving have been extensively studied resulting in several algorithms for solving the TSP, the vehicle routing problem, and other related combinatorial problems [Laporte, 1992a, Laporte, 1992b, Laporte 2010, Jünger et al.,1995, Johnson and McGoech, 1997, Johnson and McGoech, 2007, Ahuja et. al., 2002, Rego et. al., 2011, Vidal et al., 2013]. The Lin-Kernighan heuristic [Lin and Kernighan, 1973] is the most popular and effective local search algorithm for large TSPs [Rego & Glover, 2007]. Lawler et al. (1985) explored the performance guarantees of local search algorithms, Aarts et al. (2003) discussed the advantages and disadvantages of local search, Johnson and McGoech, (2007) conducted an experimental analysis of symmetric TSPs, Johnson et al. (1988) discussed the characteristics of local search problems, Okano et al. (1999) and Englert et al. (2007) analysed the performance of the 2-opt algorithm, and Funke et al. (2005) examined local search for vehicle routing problems. However, which operators are effective for open-loop cases is still not known with certainty. Furthermore, a detailed study of the performances of various local search operators is lacking. Therefore, in **[IV]**, we studied two existing methods, the *relocate* [Gendreau et al., 1992] and the *2-opt* [Croes, 1958] local search operators, and two new methods, the *3-permute* and *link swap* local search operators, for solving open-loop cases. Moreover, we proposed a *random mixed local search algorithm* using these operators.

5.1 RANDOM MIXED LOCAL SEARCH

5.1.1 Local search operators

Local search operators define the way to improve the current solution. Such improvements create a set of new solutions called neighbourhood solutions. The search strategy of the algorithm finds the improved solution from the neighbourhood solutions. Figure 37 shows an example of how the relocate local search operator works. As the name indicates, the relocate operator changes the location of a target or node in the solution, which changes the order. Here, we call the node as the *target* that changes its location. We also call the new location for the target as its *destination*. The relocation of each node can give an improved solution. For a problem with N nodes (here, we assume $N>2$), we can relocate each node to N-2 locations. Therefore, in a single relocation, we could get a particular solution out of $N(N-2)$ possible solutions. Hence, the neighbourhood search size is $O(N^2)$.

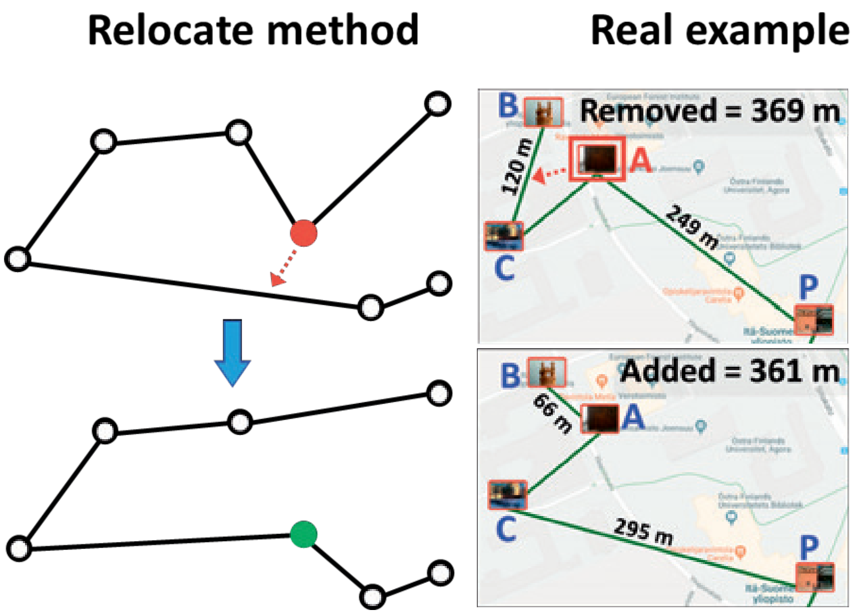

Figure 37: The relocate method and an example of it shortens the tour in an O-Mopsi game by 8 metres.

The 2-opt method developed by Croes [1958] is a popular link exchange method. Figure 38 shows how changing the connections between two subsequent pairs of nodes produces an improved solution. Here, the size of the neighbourhood is also $O(N^2)$.

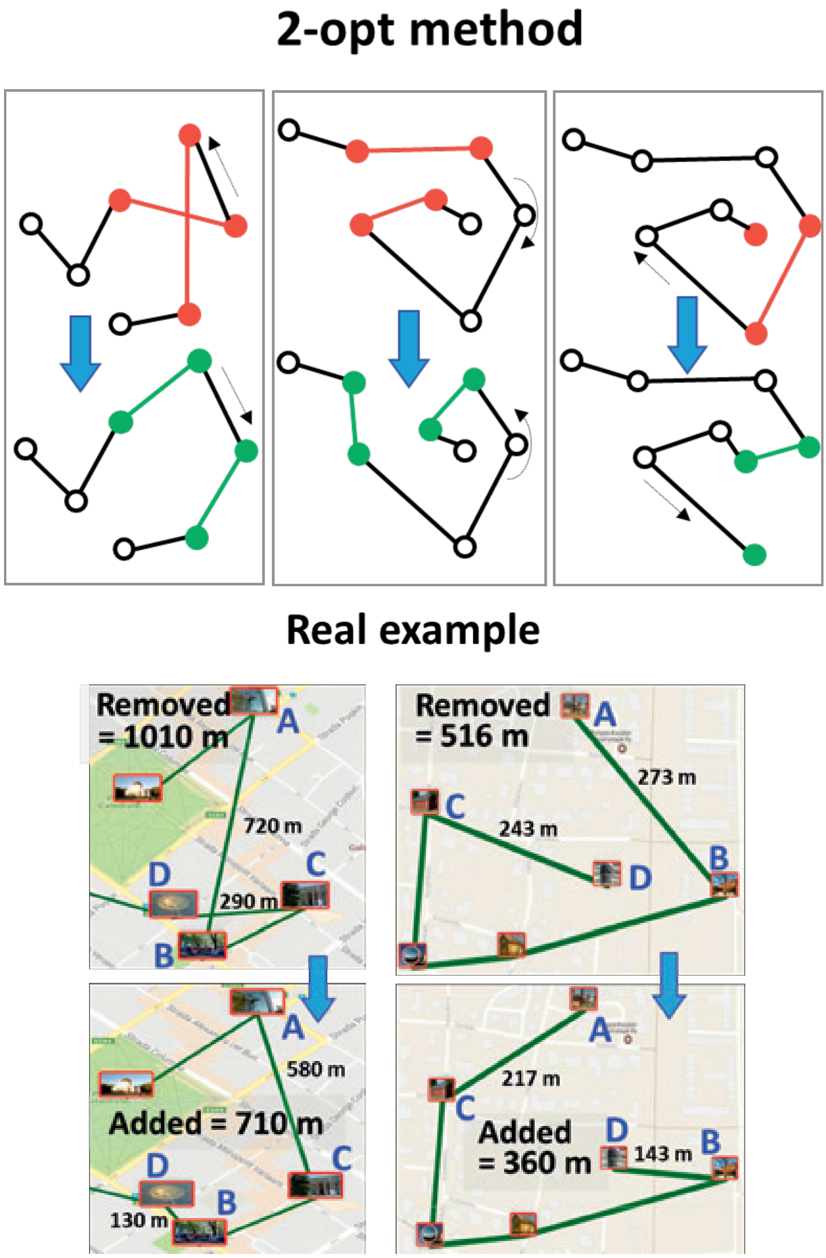

Figure 38: Examples of the 2-opt method.

In **[IV],** we studied two new methods: the 3-permute and the link swap methods. The 3-permute method creates neighbourhood solutions by considering all permutations of three consecutive nodes (Figure 39). Link swap exchanges the location of a link, as shown in Figure 40. The neighbourhood size for both 3-permute and link swap is O(N).

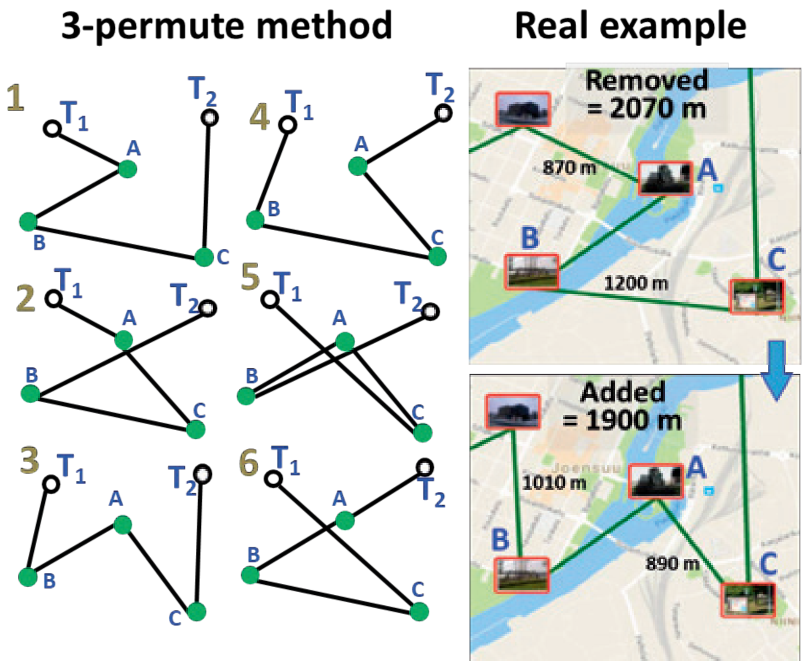

Figure 39: All six combinations of the 3-permute and an O-Mopsi game example that is improved by 170 m using this method.

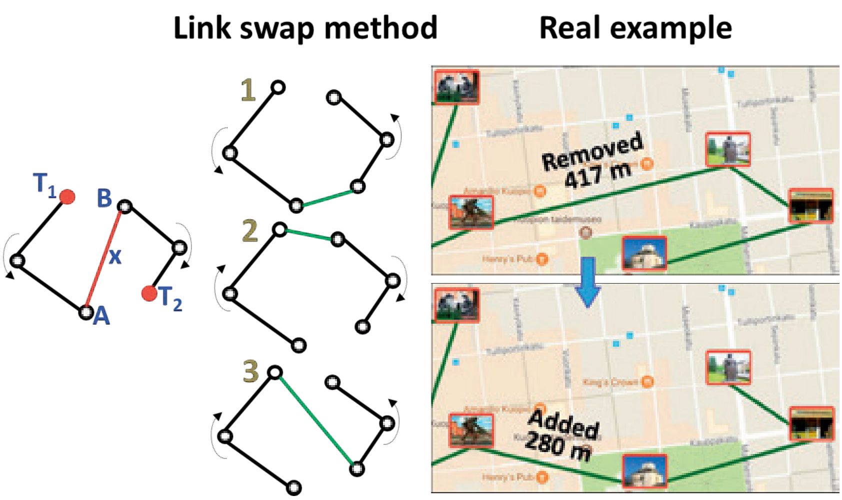

Figure 40: Original tour and three alternatives created by link swap (left) (in each case, the link x is replaced by a new link connected to one or both terminal nodes). An O-Mopsi example (right), in which the tour is improved by 137m using this method.

From the definition of relocate, we can observe that 2-opt and link swap can achieve the same solution if the destination of the target is next to it. Even if the destination is within three nodes of the target, then 3-permute can achieve the same solutions. However, when the destination is more than three nodes away from the target, only relocate can provide such solutions. These are unique solutions for relocate. Figure 41 shows relocate has such unique solutions in its neighbourhood.

From the arrows of Figure 38, we can notice that the 2-opt method reverses the order of a sub-set of nodes. A single operation of 2-opt can reverse only one sub-set of nodes of an instance. When this sub-set is more than three-nodes long and does not contain any terminal node, then the 2-opt operation gives a unique solution. If this sub-set contains only two nodes, relocate can achieve that solution. 3-permute can provide the same solution if the sub-set contains only three nodes. Again if this sub-set contains any terminal node, link swap can also achieve that solution. Figure 41 shows the unique solutions of 2-opt.

Figure 40 shows link swap can only reverse two sub-sets of nodes of an instance. However, both sub-sets must contain terminal nodes. Hence, an instance with close terminal nodes can only be improved by link swap. Figure 41 shows such an example.

3-permute does not provide any such unique solution as all six combinations of three consecutive nodes are always achievable by anyone of relocate, 2-opt, and link swap. If we consider permuting the locations of A, B, C of the example of Figure 39, Table 6 shows that other operators are able to achieve all five permutations from A, B, C.

Table 6: Solutions provided by 3-permute can be achieved by other operators.

1.	T_1, A, B, C, T_2	Starting order
2.	T_1, A, B, C, T_2 -> T_1, A, C, B, T_2	At least relocate can find when the target is C and it moves between A and B
3.	T_1, A, B, C, T_2 -> T_1, B, A, C, T_2	At least relocate can find when the target is B and it moves between T_1 and A
4.	T_1, A, B, C, T_2 -> T_1, B, C, A, T_2	At least relocate can find when the target is A and it moves between C and T_2
5.	T_1, A, B, C, T_2 -> T_1, C, A, B, T_2	At least relocate can find when the target is C and it moves between T_1 and A
6.	T_1, A, B, C, T_2 -> T_1, C, B, A, T_2	2-opt can find as the A, B, C sub-set can be reversed to C, B, A

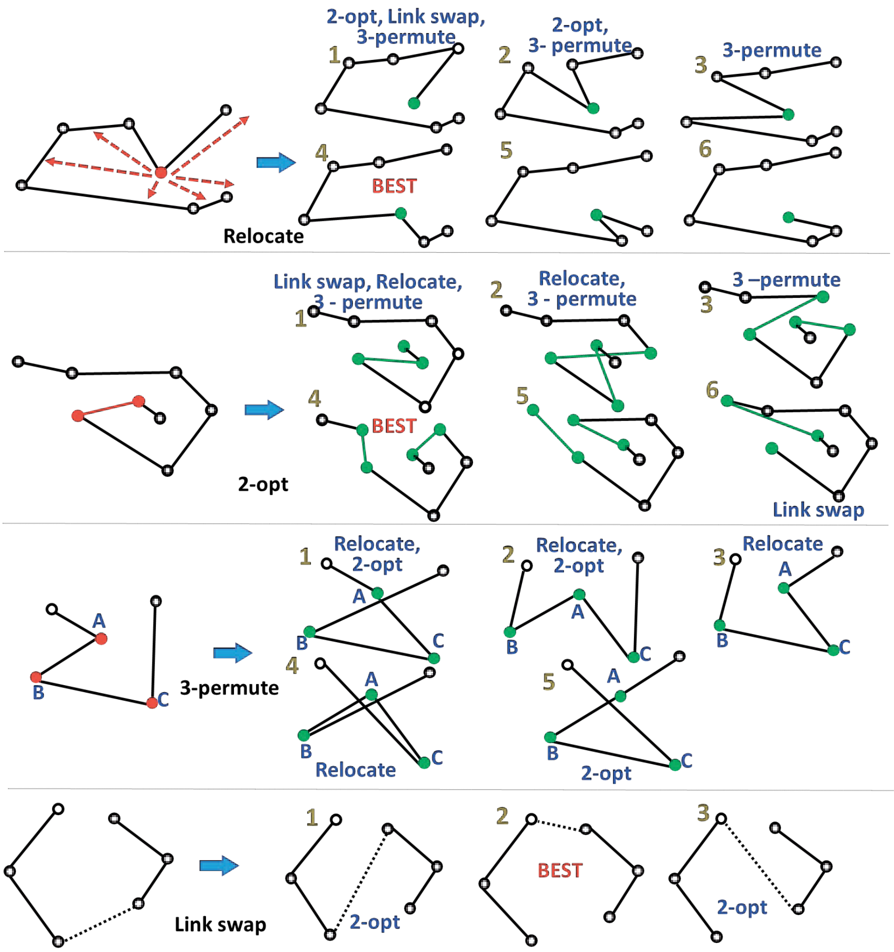

Figure 41: Relocate generates three unique solutions, 2-opt generates two unique solutions, and link swap generates one unique solution; 3-permute does not generate any unique solution.

Therefore, all operators except 3-permute generate complementary neighbourhoods (Figure 42). Consequently, relocate, 2-opt, and link swap are not redundant when used in a combination, but the neighbourhood of 3-permute does not generate a unique solution. Hence, in **[IV]**, we studied all possible ordered combinations of relocate, 2-opt, and link swap. Although, our proposed random mixed local search algorithm uses random mixing of these operators instead of any ordered combination as randomisation might benefit more.

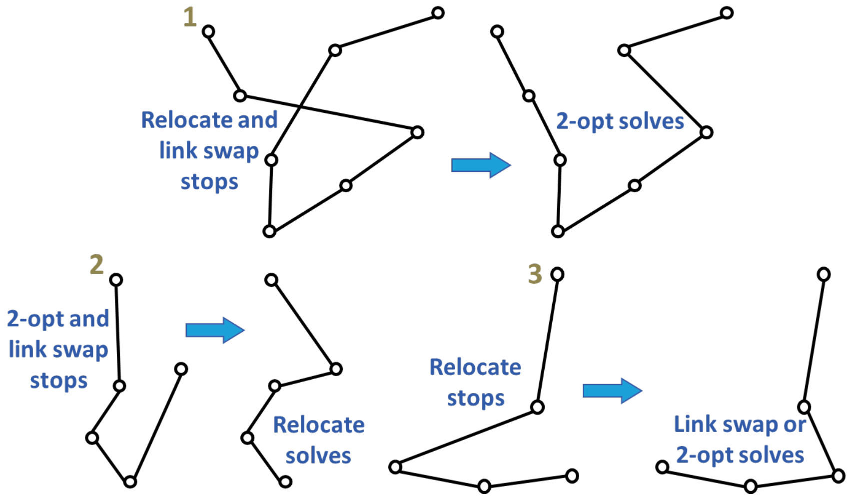

Figure 42: Operators balance each other's limitations.

When a single operator stops improving the solution, another operator may continue to improve it; therefore, a combination of operators improves the result further. It is also possible that an operator that has stopped working can start to work again when other operators have made further improvements. Hence, the works of other operators can reactivate the first operator. Figure 43 shows that three such examples where the first operator becomes trapped within a local optimum. Other operators then recover the process from the local optimum. The solution continues improving and the first operator starts working again later in the process.

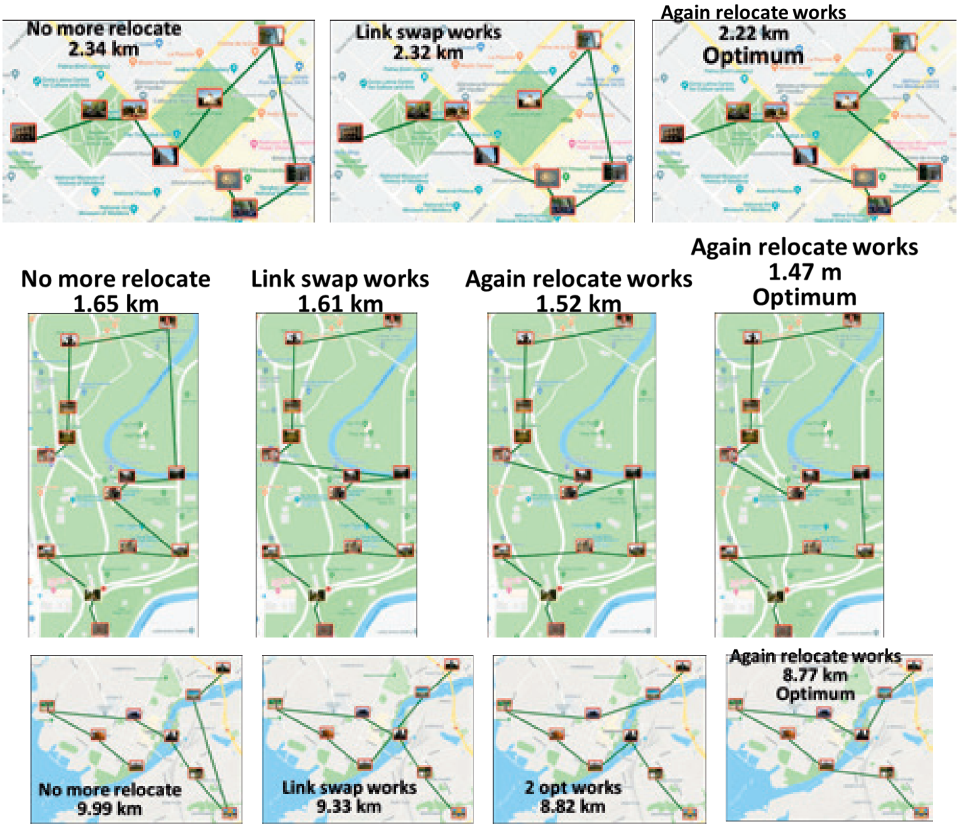

Figure 43: A single operator can start working after becoming stuck when other operators have made improvements.

Although all three operators are complementary, the question of whether the improvement rates of all three operators remain equal. To address this question, we counted the number of improvements found by each operator during the execution of the random mixed local search algorithm. We tested our algorithm on two open-loop datasets, O-Mopsi[10] and Dots[11]. We also tested our algorithm on 12 closed-loop instances from the TSPLIB [Reinelt, 1991] dataset. As a closed-loop instance does not have an open link to swap its position, the link swap operator is designed for open-loop cases only. Therefore, for TSPLIB instances, we used the relocate and 2-opt operators. Figure 44 shows that 2-opt and relocate were almost equally productive in all cases. However, the situation of the closed-loop cases differed from that of the open-loop cases. While 2-opt and relocate both accounted for about 50% of the total improvement in length in the closed-loop cases (TSPLIB), they were weaker than the new link swap operator in the open-loop cases (O-Mopsi and Dots). The link swap

[10] http://cs.uef.fi/o-mopsi/datasets/o-mopsi/
[11] http://cs.uef.fi/o-mopsi/datasets/dots/

operator dominated in both open-loop cases by accounting for almost 50% of the total improvement in length while 2-opt and relocate each only accounted for approximately 25% of the total improvement in length. Relocate was also slightly more productive than 2-opt in the open-loop cases.

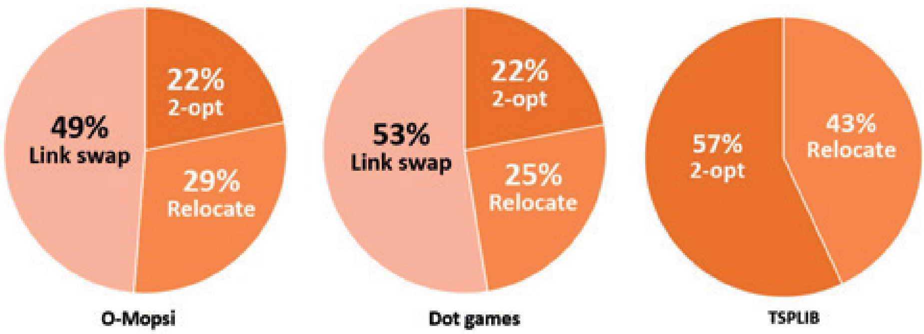

Figure 44: Share of improvements achieved by each of the three methods.

5.1.2 Initialization

Apart from the operators, there are two other aspects of local search to consider. One is the construction of the initial solution and the other is the search among the neighbourhood solutions. Our random mixed local search algorithm computes a randomly selected path through all targets, hence, it uses a random initialization strategy.

5.1.3 Search strategy

There could be several improved solutions in the neighbourhood. However, the algorithm chooses only one of them using the search strategy. The algorithm can select the best of the improved solutions, the first available improved solution, or a random improved solution. We studied all three strategies in **[IV]**. However, for the random mixed local search, we used the random improvement method.

Local search algorithms can easily become trapped within local optimum points (Figure 45). To avoid local optimum points, the random mixed local search uses the multi-start mechanism. Thus, the algorithm starts from several random initial solutions and repeats the whole process for each. Finally, it provides the best result out of all repeats.

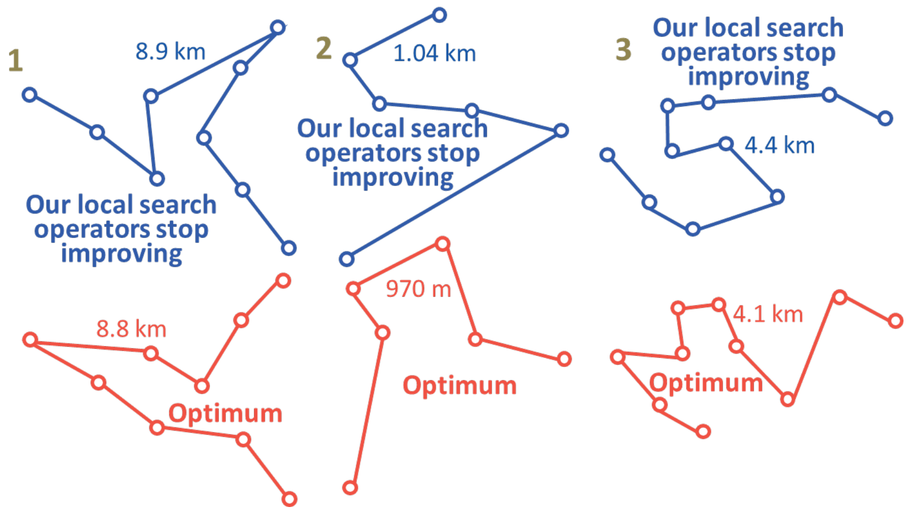

Figure 45: Examples of local optima that are overcome by repeats.

5.2 EVALUATION

The time complexity of an algorithm is an essential factor to consider for its evaluation. As the search strategy is random, the time complexity of our random mixed local search does not depend on the size of the neighbourhood of each local operator. Instead, its time complexity depends on the number of iterations required for the tour improvement process and the number of repetitions necessary for the multi-start. Each iteration is a trial; it can be successful, in which case improvement is found, or it can be unsuccessful, in which case no improvement is found. A successful iteration contributes to the path update process, which is dependent on the number of targets. Sometimes, the effect of improvement in a successful iteration is significant enough to limit the number of these iterations relatively small (0.002) concerning the total number of iterations. Even being dependent on the number of targets, henceforth the update process has very little effect on the total time. Therefore, the number of iterations and the number of repetitions are the only crucial parameters here.

In **[IV]**, we empirically calculated the time complexity of an exact solver, Concorde that is exponential (1.5e^(0.004N)). This algorithm is expected to be fast for instances with a few thousands of targets. However, for larger instances, the running time increases significantly.

As instances in the O-Mopsi and Dots datasets are small in size (N < 50), just 10,000 iterations and 25 repeats per each instance were enough to find optimum solutions for almost all instances. Since TSPLIB [Reinelt, 1991] instances are significantly larger (N = 52 to 3,795), we used 10^8 iterations and 25 repeats.

Random mixed local search did not perform better if metaheuristics, such as tabu search or simulated annealing, were added to it. We executed tabu search and simulated annealing on more than 4000 open-loop TSP instances of O-Mopsi and Dots datasets. Table 7 shows the comparative results of random mixed local search, tabu search, and simulated annealing. The results of the tabu search and simulated annealing are not significantly better. Even the productivity of the operators was similar.

Table 7: Summary of overall results (mean). The execution time is calculated on a 2.7 GHz processor.

Results	Random mixed local search		Tabu		SA	
	Gap	Execution time	Gap	Execution time	Gap	Execution time
O-Mopsi	0%	16 ms	0%	1.3 s	0%	25 ms
Dots	0.001%	16 ms	0.0003%	1.3 s	0.007%	20 ms

6 SUMMARY OF CONTRIBUTIONS

This chapter summarises the contributions of our four publications. In publication **[I],** we study the features of the O-Mopsi game and the research challenges it poses. Publication **[II]** studies the appropriate start positions for both computer algorithms and humans when solving an open-loop TSP. In **[III],** we measure the difficulty of open-loop TSPs using MST structures. Publication **[IV]** studies several local search operators and presented the random mixed local search algorithm for open-loop TSPs of small sizes.

In **[I]**, we present a review of location-based games. We explain the characteristics of O-Mopsi, a location-based mobile game. We analyse the main motivations to play this game. We study the average performance of players of a particular game and compare it to the features of that game. Thus, we reveal the challenges of the game and the type of skills required by a player.

In **[II]**, we investigate various strategies for selecting start positions for both computers and humans to solve the open-loop TSP. We construct a grid over the area of the problem. We find that corners of such a grid are the most probable regions to have a starting point for an optimised path computed by an exact algorithm. By contrast, humans prefer to start from a point on the convex hull of the problem or the target that is furthest away. The conclusions from this study can be applied in any tour-planning or tour-optimisation application where the starting point is not determined beforehand.

In **[III]**, we show that the MST structure can estimate the difficulty of an open-loop TSP. We call the nodes with more than two links MST knots. We find that human performance and execution time for the computer are linearly dependent on the number of MST knots. Hence, this number measures the difficulty of the problem. This measure is useful as it might help people to select a suitable game to play.

In **[IV]**, we study ways of solving open-loop TSPs, which is one of the challenges of O-Mopsi. We consider the local search technique for solving the TSP and study two existing local search operators (relocate and 2-opt) and introduce two new operators (3-permute and link swap). The link swap operator is found to be the most productive operator for improving the path. We also find it to be rather influential as, without this operator, the productivity of the relocate operator diminishes. We develop random mixed local search algorithm for solving open-loop TSPs using relocate, 2-opt, and link swap. This algorithm randomly mixes these three operators and iteratively tries to optimise a path. The whole process repeatedly starts from scratch to avoid local optima. Similar to this algorithm, these operators can be used for computer-based TSP games or in a tour planning application.

7 CONCLUSIONS

Although O-Mopsi is an orienteering game, it offers players to choose their ways to visit targets. Similarly, most tour planning applications provide POIs but no fixed path. Hence, these types of applications represent relatively small open-loop TSPs. We show that choosing an appropriate starting position is crucial. We also show that the difficulty level of a problem is a significant factor that affects a human's ability to solve the problem.

The particularly challenging task of finding the appropriate starting point applies solely in the open-loop case. A significant amount of research has been devoted to analysing human performance in solving TSPs. Several algorithms have been developed based on human strategies for solving a TSP. However, studies of the optimum start location for open-loop cases are lacking. Our study shows that corners are the most probable regions to have a starting point for the optimised path. We also show that humans prefer to choose a target on the convex hull or the target that is located furthest away from the centre as a starting point.

Researchers have discovered a correlation between the problem size and human performance in solving TSPs. The convex hulls of the problems have also been found to be related to human performance. We show that, for the open-loop TSP, apart from the problem size, the MST structure is also an important factor for both human and computer solvers. Our proposed method of measuring the difficulty of an open-loop TSP using the MST structure is a simple, polynomial-time algorithm, which does not require the optimum solution to determine the difficulty.

Many studies have used local search algorithms to solve TSPs. However, very few have studied local search operators in detail. Furthermore, most existing studies focus on closed-loop cases, while open-loop cases remain relatively unexplored. We introduce a new local search operator, link swap, which is specially designed for open-loop cases. Our experiments show that link swap is the most productive operator in a local search. Our proposed random mix local search algorithm works efficiently for small-sized problems. The speed of the algorithm is almost independent of the problem size.

In the future, it would be useful to study the playing strategy of the total game instead of only the starting point. The frequency of choosing the nearest neighbour as the next target by the players would be also interesting. This would enable us to determine the effect of using the nearest neighbour strategy on performance. In the future, we can as well analyse comparative performance of several players based on their routes. This might act as another performance determiner. We can conclude more on the difficulty by studying players' performance more analytically. Apart from the TSP difficulty, finding other playability issues, accessibility issues to the targets would make the application more practical.

Additionally, content creation for any LBG is another topic that needs to be studied thoroughly. There is a large amount of geotagged multimedia content available. However, they need to be structured and filtered according to the needs of the location-based application. Besides, several multimedia contents lack the location information within them for several reasons. Therefore, we can explore all possible sources thoroughly for comprehensive content or can develop a location-estimator system that can provide the location information to the non-geotagged content.

BIBLIOGRAPHY

Aarts, E., Aarts, E. H., & Lenstra, J. K. (Eds.). (2003). *Local search in combinatorial optimization*. Princeton University Press.

Agarwal, P., Sourabh, M., Sachdeva, R., Sharma, S., & Mehta, S. (2018, August). Recommending Optimal Tour for Groups Using Ant Colony Optimization. In *2018 Eleventh International Conference on Contemporary Computing (IC3)*(pp. 1-6). IEEE.

Ahuja, R. K., Ergun, Ö., Orlin, J. B., & Punnen, A. P. (2002). A survey of very large-scale neighborhood search techniques. *Discrete Applied Mathematics*, *123*(1-3), 75-102.

Applegate, D. L., Bixby, R. E., Chvatal, V., & Cook, W. J. (2011). *The traveling salesman problem: a computational study*. Princeton university press.

Applegate, D., Bixby, R., Cook, W., & Chvátal, V. (1998). On the solution of traveling salesman problems. *Documenta Mathematica Journal der Deutschen Mathematiker-Vereinigung, International Congress of Mathematicians,* 645-656.

Avouris, N. M., & Yiannoutsou, N. (2012). A review of mobile location-based games for learning across physical and virtual spaces. *J. UCS*, *18*(15), 2120-2142.

Ballagas, R. A., Kratz, S. G., Borchers, J., Yu, E., Walz, S. P., Fuhr, C. O.,... & Tann, M. (2007, April). REXplorer: a mobile, pervasive spell-casting game for tourists. In *CHI'07 extended abstracts on Human factors in computing systems* (pp. 1929-1934). ACM.

Benford, S., Crabtree, A., Flintham, M., Drozd, A., Anastasi, R., Paxton, M.,... & Row-Farr, J. (2006). Can you see me now?. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *13*(1), 100-133.

Bolzoni, P., Helmer, S., Wellenzohn, K., Gamper, J., & Andritsos, P. (2014). Efficient itinerary planning with category constraints. *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - SIGSPATIAL 14*.

Borràs, J., Moreno, A. & Valls, A. (2014). Intelligent tourism recommender systems: A survey. *Expert Syst. Appl.*, *41*, 7370-7389.

Ceipidor, U. B., Medaglia, C. M., Perrone, A., De Marsico, M., & Di Romano, G. (2009, June). A museum mobile game for children using QR-codes. In *Proceedings of the 8th international conference on interaction design and children* (pp. 282-283). ACM.

Chatzidimitris, T., Gavalas, D., & Michael, D. (2016). SoundPacman: Audio augmented reality in location-based games. In 2016 18th Mediterranean Electrotechnical Conference (MELECON) (pp. 1-6). IEEE.

Cheok, A. D., Goh, K. H., Liu, W., Farbiz, F., Fong, S. W., Teo, S. L., & Yang, X. (2004). Human Pacman: a mobile, wide-area entertainment system based on physical, social, and ubiquitous computing. *Personal and ubiquitous computing*, *8*(2), 71-81.

Chittaro, L., & Sioni, R. (2012, June). Turning the classic snake mobile game into a location–based exergame that encourages walking. In *International Conference on Persuasive Technology* (pp. 43-54). Springer, Berlin, Heidelberg.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms*. MIT press.

Croes, G. A. (1958). A method for solving traveling-salesman problems. *Operations research*, *6*(6), 791-812.

Dantzig, G., Fulkerson, R., & Johnson, S. (1954). Solution of a large-scale traveling-salesman problem. *Journal of the operations research society of America*, *2*(4), 393-410.

De Choudhury, M., Feldman, M., Amer-Yahia, M., Golbandi, N., Lempel, R., & Yu., C. (2010). Automatic construction of travel itineraries using social breadcrumbs. *In Proceedings of the 21st ACM conference on Hypertext and Hypermedia* (HT '10), ACM, New York, NY, USA, 35-44.

Dry, M. J., Lee, M. D., Vickers, D., & Hughes, P. (2006). Human performance on visually presented traveling salesperson problems with varying numbers of nodes. *Journal of Problem Solving, 1,* 20-32.

Dry, M. J., Preiss, K., & and Wagemans, J. (2012). Clustering, Randomness, and Regularity: Spatial Distributions and Human Performance on the Traveling Salesperson Problem and Minimum Spanning Tree Problem. *The Journal of Problem Solving*, Vol. 4: Iss. 1, Article 2.

Dry, M., Lee, M. D., Vickers, D., & Hughes, P. (2006). Human performance on visually presented traveling salesperson problems with varying numbers of nodes. *The Journal of Problem Solving*, *1*(1), 4.

Dry, M., Preiss, K., & Wagemans, J. (2012). Clustering, randomness, and regularity: Spatial distributions and human performance on the traveling salesperson problem and minimum spanning tree problem. *The Journal of Problem Solving*, 4(1), 2.

Dry, Matthew J. and Fontaine, Elizabeth L. (2014). Fast and Efficient Discrimination of Traveling Salesperson Problem Stimulus Difficulty. *The Journal of Problem Solving*: Vol. 7: Iss. 1, Article 9.

Englert, M., Röglin, H., & Vöcking, B. (2007, January). Worst case and probabilistic analysis of the 2-Opt algorithm for the TSP. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms* (pp. 1295-1304). Society for Industrial and Applied Mathematics.

Facer, K., Joiner, R., Stanton, D., Reid, J., Hull, R., & Kirk, D. (2004). Savannah: mobile gaming and learning?. *Journal of Computer assisted learning*, *20*(6), 399-409.

Funke, B., Grünert, T., & Irnich, S. (2005). Local search for vehicle routing and scheduling problems: Review and conceptual integration. *Journal of heuristics*, *11*(4), 267-306.

Gavalas, D., Kasapakis, V., Konstantopoulos, C., Pantziou, G., Vathis, N., & Zaroliagis, C. (2015). The eCOMPASS multimodal tourist tour planner. Expert systems with Applications, 42(21), 7303-7316.

Gavalas, D., Konstantopoulos, C., Mastakas, K., & Pantziou, G. (2014) b. A survey on algorithmic approaches for solving tourist trip design problems. *Journal of Heuristics*, *20*(3), 291-328.

Gavalas, D., Konstantopoulos, C., Mastakas, K., & Pantziou, G.E. (2014) a. Mobile recommender systems in tourism. *J. Network and Computer Applications, 39*, 319-333.

Gendreau, M., Hertz, A., & Laporte, G. (1992). New insertion and postoptimization procedures for the traveling salesman problem. *Operations Research*, *40*(6), 1086-1094.

Giannakas, F., Kambourakis, G., Papasalouros, A., & Gritzalis, S. (2018). A critical review of 13 years of mobile game-based learning. *Educational Technology Research and Development*, *66*(2), 341-384.

Gionis, A., Lappas, T., Pelechrinis, K., & Terzi, E. (2014). Customized tour recommendations in urban areas. *Proceedings of the 7th ACM International Conference on Web Search and Data Mining - WSDM 14*.

Graham, S. M., Joshi, A., & Pizlo, Z. (2000). The traveling salesman problem: A hierarchical model. *Memory & cognition*, *28*(7), 1191-1204.

Graham, S. M., Joshi, A., & Pizlo, Z. (2000). The traveling salesman problem: A hierarchical model. *Memory & Cognition*, 28(7), 1191-1204.

Grötschel, M., & Holland, O. (1991). Solution of large-scale symmetric travelling salesman problems. *Mathematical Programming*, *51*(1-3), 141-202.

Gutin, G., & Punnen, A. P. (Eds.). (2006). *The traveling salesman problem and its variations* (Vol. 12). Springer Science & Business Media.

Haruna, K., Akmar Ismail, M., Suhendroyono, S., Damiasih, D., Pierewan, A., Chiroma, H., & Herawan, T. (2017). Context-aware recommender system: A review of recent developmental process and future research direction. *Applied Sciences*, *7*(12), 1211.

Held, M., & Karp, R. M. (1971). The traveling-salesman problem and minimum spanning trees: Part II. *Mathematical programming*, *1*(1), 6-25.

Jacob, J. T. P. N., & Coelho, A. F. (2011). Issues in the development of location-based games. *International Journal of Computer Games Technology*, *2011*.

Johnson, D. S., & McGeoch, L. A. (1997). The traveling salesman problem: A case study in local optimization. *Local search in combinatorial optimization*, *1*(1), 215-310.

Johnson, D. S., & McGeoch, L. A. (2007). Experimental analysis of heuristics for the STSP. In *The traveling salesman problem and its variations* (pp. 369-443). Springer, Boston, MA.

Johnson, D. S., Papadimitriou, C. H., & Yannakakis, M. (1988). How easy is local search?. *Journal of computer and system sciences*, *37*(1), 79-100.

Jünger, M., Reinelt, G., & Rinaldi, G. (1995). The traveling salesman problem. *Handbooks in operations research and management science*, *7*, 225-330.

Kasapakis, V., & Gavalas, D. (2015). Pervasive gaming: Status, trends and design principles. Journal of Network and Computer Applications, 55, 213-236.

Kruskal, J. B. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. Proceedings of the American Mathematical society, 7(1), 48-50.

La Guardia, D., Arrigo, M., & Di Giuseppe, O. (2012). A location-based serious game to learn about the culture. *Proc. FOE*.

Laporte, G. (1992) a. The traveling salesman problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, *59*(2), 231-247.

Laporte, G. (1992) b. The vehicle routing problem: An overview of exact and approximate algorithms. *European journal of operational research*, *59*(3), 345-358.

Laporte, G. (2010). A concise guide to the traveling salesman problem. *Journal of the Operational Research Society*, *61*(1), 35-40.

Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H., & Shmoys, D. B. (1985). The traveling salesman problem; a guided tour of combinatorial optimization.

Li, M., O'Grady, M. J., & O'Hare, G. M. (2008). Geo-Gaming: The Mobile Monopoly Experience. In *WEBIST (1)*(pp. 220-223).

Lim, K. H., Chan, J., Karunasekera, S., & Leckie, C. (2018). Tour recommendation and trip planning using location-based social media: a survey. *Knowledge and Information Systems*, 1-29.

Lin, S., & Kernighan, B. W. (1973). An effective heuristic algorithm for the traveling-salesman problem. *Operations research*, *21*(2), 498-516.

Lynch, M. (2012). Location, location, location—A review of location based gaming. *Journal of International Computer Science*, *18*, 15.

Macgregor, J. N. (2012). Indentations and Starting Points in Traveling Sales Tour Problems: Implications for Theory. *The Journal of Problem Solving*,*5*(1).

MacGregor, J. N., Chronicle, E. P., & Ormerod, T. C. (2004). Convex hull or crossing avoidance? Solution heuristics in the traveling salesperson problem. *Memory & cognition*, *32*(2), 260-270.

Macgregor, J. N., Chronicle, E. P., & Ormerod, T. C. (2006). A Comparison of Heuristic and Human Performance on Open Versions of the Traveling Salesperson Problem. *The Journal of Problem Solving,1*(1).

MacGregor, J. N., Ormerod, T. C., & Chronicle, E. P. (1999). Spatial and contextual factors in human performance on the travelling salesperson problem. *Perception*, *28*(11), 1417-1427.

MacGregor, J. N., Ormerod, T. C., & Chronicle, E. P. (2000). A model of human performance on the traveling salesperson problem. *Memory & Cognition*, *28*(7), 1183-1190.

Macgregor, J.N. & Ormerod, T. (1996). Human performance on the traveling salesman problem. *Perception & Psychophysics* 58: 527.

Magerkurth, C., Cheok, A. D., Mandryk, R. L., & Nilsen, T. (2005). Pervasive games: bringing computer entertainment back to the real world. Computers in Entertainment (CIE), 3(3), 4-4.

Matyas, S., Matyas, C., Schlieder, C., Kiefer, P., Mitarai, H., & Kamata, M. (2008, December). Designing location-based mobile games with a purpose: collecting geospatial data with CityExplorer. In *Proceedings of the 2008 international conference on advances in computer entertainment technology* (pp. 244-247). ACM.

Misund, G., Holone, H., Karlsen, J., & Tolsby, H. (2009). Chase and Catch-simple as that?: old-fashioned fun of traditional playground games revitalized with location-aware mobile phones. In *Proceedings of the International Conference on Advances in Computer Enterntainment Technology* (pp. 73-80). ACM.

Moore, A., Goulding, J., Brown, E., & Swan, J. (2009). AnswerTree–a hyperplace-based game for collaborative mobile learning. In *Proceedings of the World Conference on Mobile and Contextual Learning (mLearn'09)*.

Mor, M. & Dalyot, S. (2018). Computing Touristic Walking Routes using Geotagged Photographs from Flickr. *In Proceedings of the 14th International Conference on Location Based Services.*, 63-68.

Okano, H., Misono, S., & Iwano, K. (1999). New TSP construction heuristics and their relationships to the 2-opt. *Journal of Heuristics*, *5*(1), 71-88.

Padberg, M., & Rinaldi, G. (1991). A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM review*, *33*(1), 60-100.

Papadimitriou, C. H. (1977). The Euclidean travelling salesman problem is NP-complete. *Theoretical computer science*, *4*(3), 237-244.

Piekarski, W., & Thomas, B. (2002). ARQuake: the outdoor augmented reality gaming system. *Communications of the ACM*, *45*(1), 36-38.

Pizlo, Z., Stefanov, E., Saalweachter, J., Li, Z., Haxhimusa, Y., and Kropatsch, W. G. (2006). Traveling salesman problem: a foveating pyramid model. *The Journal of Problem Solving*, 1(1).

Prim, R. C. (1957). Shortest connection networks and some generalizations. The Bell System Technical Journal, 36(6), 1389-1401.

Rego, C., & Glover, F. (2007). Local search and metaheuristics. In *The traveling salesman problem and its variations* (pp. 309-368). Springer, Boston, MA.

Rego, C., Gamboa, D., Glover, F., & Osterman, C. (2011). Traveling salesman problem heuristics: Leading methods, implementations and latest advances. *European Journal of Operational Research*, *211*(3), 427-441.

Reinelt, G. (1991). TSPLIB—A traveling salesman problem library. *ORSA journal on computing*, *3*(4), 376-384.

Sánchez, J., Salinas, A., & Sáenz, M. (2006, November). Mobile game-based science learning. In *Proceedings of the Distance Learning and Internet Conference* (pp. 18-30).

Schlieder, C., Kiefer, P., & Matyas, S. (2005, November). Geogames: A conceptual framework and tool for the design of location-based games from classic board games. In *International Conference on Intelligent Technologies for Interactive Entertainment* (pp. 164-173). Springer, Berlin, Heidelberg.

Sedano, C. I., Sutinen, E., Vinni, M., & Laine, T. H. (2012). Designing hypercontextualized games: a case study with LieksaMyst. *Journal of Educational Technology & Society*, *15*(2), 257-270.

Sintoris, C., Stoica, A., Papadimitriou, I., Yiannoutsou, N., Komis, V., & Avouris, N. (2012). MuseumScrabble: Design of a mobile game for children’s interaction with a digitally augmented cultural space. In *Social and organizational impacts of emerging mobile devices: Evaluating use* (pp. 124-142). IGI Global.

Sintoris, C., Yiannoutsou, N., Demetriou, S., & Avouris, N. M. (2013). Discovering the invisible city: Location-based games for learning in smart cities. *IxD&A*, *16*, 47-64.

Spikol, D., & Milrad, M. (2008, March). Combining physical activities and mobile games to promote novel learning practices. In *Fifth IEEE International Conference on Wireless, Mobile, and Ubiquitous Technology in Education (wmute 2008)* (pp. 31-38). IEEE.

Spikol, D., & Milrad, M. (2008, March). Combining physical activities and mobile games to promote novel learning practices. In *Fifth IEEE International Conference on Wireless, Mobile, and Ubiquitous Technology in Education (wmute 2008)* (pp. 31-38). IEEE.

Van Rooij, I., Stege, U., & Schactman, A. (2003). Convex hull and tour crossings in the Euclidean traveling salesperson problem: Implications for human performance studies. *Memory & cognition*, *31*(2), 215-220.

Vansteenwegen, P., Souffriau, W., & Van Oudheusden, D. (2011). The orienteering problem: A survey. European Journal of Operational Research, 209(1), 1-10.

Vassilakis, K., Charalampakos, O., Glykokokalos, G., Kontokalou, P., Kalogiannakis, M., & Vidakis, N. (2017). Learning history through location-based games: The fortification gates of the venetian walls of the city of Heraklion. In *Interactivity, Game Creation, Design, Learning, and Innovation* (pp. 510-519). Springer, Cham.

Veenhof, G., Sandberg, J., & Maris, M. (2012). ZooQuest: A mobile game-based learning application for fifth graders. In *International Conference on Intelligent Tutoring Systems* (pp. 687-688). Springer, Berlin, Heidelberg.

Vickers, D., Bovet, P., Lee, M. D., & Hughes, P. (2003) b. The perception of minimal structures: Performance on open and closed versions of visually presented Euclidean travelling salesperson problems. *Perception*, 32(7), 871-886.

Vickers, D., Lee, M. D., Dry, M., & Hughes, P. (2003) a. The roles of the convex hull and the number of potential intersections in performance on visually presented traveling salesperson problems. *Memory & Cognition*, 31(7), 1094-1104.

Vickers, D., Mayo, T., Heitmann, M., Lee, M. D., & Hughes, P. (2004). Intelligence and individual differences in performance on three types of visually presented optimisation problems. *Personality and Individual Differences*, *36*(5), 1059-1071.

Vickers, D., Mayo, T., Heitmann, M., Lee, M.D., & Hughes, P. (2004). Intelligence and individual differences on three types of visually presented optimisation problems. *Personality and Individual Differences*, 36, 1059-1071.

Vidal, T., Crainic, T. G., Gendreau, M., & Prins, C. (2013). Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. *European Journal of Operational Research*, *231*(1), 1-21.

Wan, Z. (2014). *O-Mopsi: Location-based Orienteering Mobile Game* (M. Sc. Thesis. School of Computing, University of Eastern Finland).

Wetzel, R., Blum, L., & Oppermann, L. (2012). Tidy city: a location-based game supported by in-situ and web-based authoring tools to enable user-created content. In *Proceedings of the international conference on the foundations of digital games* (pp. 238-241). ACM.

LAHARI SENGUPTA

Location-based games and trip planning applications are gaining popularity worldwide. In several cases, they include the path optimisation problem. People need to know the characteristics of such a problem and planning strategies to solve them. O-Mopsi is one of such location-based games that contains small scale path optimisation problems. In this thesis, we study the problems contained in O-Mopsi and how human perform in solving those. We present methods to estimate their difficulty and algorithms to solve them.

UNIVERSITY OF
EASTERN FINLAND

uef.fi

**PUBLICATIONS OF
THE UNIVERSITY OF EASTERN FINLAND**
Dissertations in Forestry and Natural Sciences

ISBN 978-952-61-3248-8
ISSN 1798-5668