NORTHWESTERN UNIVERSITY

Models and Approaches to Multiobjective Arc Tour Problems with an Application to Marathon Course Design

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Industrial Engineering and Management Sciences

By

Mehmet Başdere

EVANSTON, ILLINOIS

June 2018

ABSTRACT

Models and Approaches to Multiobjective Arc Tour Problems with an Application to Marathon Course Design

Mehmet Başdere

Marathons are long distance running events with many participants, often organized in heavily populated cities. A key component in marathon planning operations is the design of the marathon course to be followed during the race. In addition to the technical requirements regarding length and incline change, marathon courses must satisfy two additional requirements: (i) courses must visit certain neighborhoods in the city and (ii) they must be designed in such a way that access to the facilities that provide critical public service (e.g. hospitals) is not blocked by the tour. Motivated by marathon course design, this study develops models and approaches to solve multiobjective arc tour problems. The underlying tour finding problem, the Lock-Free Arc Tour Problem (LFATP), visits a predetermined set of edges while ensuring that the resulting tour does not block access to certain critical vertices. Apart from these restrictions, marathon course design involves a range of unique objective functions which we are classified under three categories: (i)

arc-additive, (ii) sequence-dependent and (iii) time-dependent objective functions. In arc-additive functions, the objective contribution of each arc has a constant coefficient. In sequence-dependent ones, each arc's contribution to the objective depends on the other arcs used in the route and in time-dependent ones, each arc's contribution depends on when that arc is visited.

Initial efforts focus on the LFATP with arc-additive objectives as these objectives are relatively less complex compared to sequence- and time-dependent ones. The underlying problem is formulated as a mixed integer linear program. Structurally, the LFATP suffers from excessive subtour formation, especially when the corresponding objective aims to minimize proximity to certain locations in the network, causing the standard branch-and-cut approach to perform poorly even with valid inequalities derived from locking properties of the LFATP. For this reason, we introduce path-based reformulations arising from a provably stronger disjunctive program, where disjunctions are obtained by fixing the visit orders in which must-visit edges are visited. In computational tests, the reformulations are shown to yield up to 100 times improvement in solution times. Additional tests demonstrate the benefits of using lock elimination inequalities and the value of reformulations for more general tour finding problems with visit requirements and length restrictions. Then, we extend the arc-additive model and its reformulations to handle objectives involving sequence- and time-dependent properties.

As marathon course design involves various objectives related to health, safety, performance and experience, LFATP needs to be solved in a multiobjective environment to identify *the best* course design for race organizers. For this reason, we introduce an Interactive Weight Region-Based approach (IWRA) that reaches a most preferred solution of

the decision maker (DM) in multiobjective optimization. IWRA is an iterative approach which iterates between solution generation and comparison phases: At each iteration, IWRA presents a new solution to the DM by using weight diversification, obtained by solving an integer program. The DM's preference information is used to refine the unexplored weight region. We provide finitely converging algorithms solving the weighted sum problem for multiobjective linear programs and multiobjective integer programs. Different from existing approaches, IWRA separates weight generation and solution generation efforts, and provides a systematic weight generation scheme which is capable of exploring the entire weight region. We empirically demonstrate the effectiveness of IWRA under various settings and assumptions through comparisons with related methods from the literature which also ensure identification of optimal decisions. IWRA is also extended to the weighted Tchebycheff problem setting to handle unsupported nondominated solutions.

Finally, both tour finding and multiobjective projects are combined to generate a series of course designs for the Bank of America Chicago Marathon which serves the race organizers as a catalog of options. At this stage, we select the underlying LFATP formulation based on the dominant structure of the aggregated objective. The experiments show that significant improvements can be obtained by making relatively small changes to the existing course.

Acknowledgements

I would like to express my sincere gratitude to my advisors Prof. Karen Smilowitz and Prof. Sanjay Mehrotra for their help and guidance during my graduate studies. I would like to thank the committee members Prof. Sunil Chopra, Dr. George Chiampas and Dr. Jennifer Chan for their support and suggestions, and Prof. Michel Gendreau for his suggestions on complex routing problems. I also would like to thank Carey Pinkowski, Michael Nishi, David Waskowski and the entire team at the Bank of America Chicago Marathon for their valuable input. This project is supported by National Science Foundation (CMMI-1405231).

Dedication

This work is dedicated to my beloved wife Tuğçe Gül Diken Başdere.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Introduction and Motivation

Motivated by marathon course design, this study introduces models and approaches to solve multiobjective arc tour problems. Proposed models and approaches generalize to a wide variety of tour finding problems and multiobjective optimization problems. The study is carried out in collaboration with the researchers in the Department of Industrial Engineering and Management Sciences and in the Department of Emergency Medicine at Northwestern University, and the organizers of Bank of America Chicago Marathon (BACM) and Shamrock Shuffle (BASS).

This introductory chapter describes the scope of our study while focusing on the details of the underlying application of marathon course design. The rest of the chapter is organized as follows: Section 1.1 provides a brief description of marathons and marathon planning activities and introduces marathon course design problem in detail. Section 1.2 describes the focus and the scope of this study. Section 1.3 briefly describes the BACM's response to marathon planning activities, the Chicago Model, and summarizes related studies carried out by our research team.

1.1. Marathons and Marathon Course Design

A marathon is a long-distance running event with an official distance of 26.22 miles. More than 500 marathons are organized worldwide every year; larger marathons have tens of thousands of participants. The annual running of a marathon is a test of preparedness

for mass gathering events, involving a host of significant planning and implementation challenges, particularly in heavily populated cities. Given the scale of participation and the diversity of entities involved, planning for an event often begins immediately after completion of the prior year's event. The bombings at the 2013 Boston Marathon brought a tragic new dimension to the issue of planning and managing mass participation events, and highlighted the extreme challenges.

Marathon planning involves decisions and activities regarding course design, communication and resource management (Chiampas and Jaworski, 2009). Specifically, course design focuses on the route to be followed during the race and the locations of runner resources on the course. Related decisions include adjustments to the existing course due to potential constructions or alterations, locations of diversion points, aid stations, medical tents and volunteers on the course. From communication perspective, the aim is to develop communication channels between the participating agents such as race organizers, medical doctors, volunteers, police and fire departments, and to disseminate race related information to runners and spectators in a fast and clear way. Related activities include developing communication hierarchies between the agents and designing communication tools to disseminate information effectively. Lastly, resource management focuses on estimating and securing necessary amounts of medical and other race related supplies.

Among other planning activities, course design is a key component in marathon planning since almost all decisions in planning and implementation phases are naturally affected by the route to be followed during the race. Despite its importance, most of the marathon courses in the world take their current shape as a consequence of traditional events, lacking a scientific or systematic design. Although some minor changes occur from

one year to another due to causes such as road constructions, the race organizers try to keep the course the same over the years. However, this brings up the following question: 'How good are the existing marathon courses in terms of many factors such as public safety, runner experience, and runner and spectator health?'

A marathon course is record eligible if finishing times of the runners are ratified as a world record by International Association of Athletics Federations (IAAF). Record eligibility is an important concept in marathon course design since record eligible marathons attract many talented runners across the world. In addition to distance requirement of 26.22 miles (42.195 kms), there are two technical requirements enforced by IAAF for a marathon to be considered record eligible (IAAF, 2015):

- *The start and finish points of a course, measured along a theoretical straight line between them, shall not be further apart than 50% of the race distance.*
- *The overall decrease in elevation between the start and finish shall not exceed 1:1000; i.e., 1m per km.*

The first requirement eliminates straight courses which can potentially benefit from consistent tailwind support and the second one eliminates courses with downhill slope.

As most marathons take place in heavily populated cities, they block access to some districts over a five-to-seven hour span. Thus, cities generally enforce accessibility restrictions to ensure public access to critical facilities during the race. These critical facilities commonly include hospitals since blocking access to hospitals is undesirable from public health perspective. They can also include main churches in cities since most marathons occur on Sundays. Accessibility restrictions make the resulting tour finding problem unique as such restrictions have not been studied in the literature before.

Apart from technical requirements and accessibility restrictions, marathon courses are expected to visit certain locations in the city due to several reasons. First of all, many marathons have been organized for decades, generating their own traditions. Some neighborhoods or street segments are traditionally important and they must be visited by the designed courses. Secondly, some districts are visited to increase diversity along the course. Thirdly, marathons need large-space areas near the course to build medical tents and to store runner resources; therefore, the underlying courses must pass near such locations. Lastly, courses pass through some city landmarks to attract more runners and provide a more enjoyable experience. Visit requirements are less formal compared to the technical ones; however, they play a key role in satisfying the expectations of various groups such as runners, spectators and sponsors.

1.2. Research Focus

This study focuses on designing a record eligible marathon course which satisfies the technical requirements while visiting predetermined set of street segments and not blocking access to the critical facilities in the region of interest. While the presented models find 'tour-type' marathon courses with start and finish lines that are very close to each other, they can be generalized to find marathon courses with distant start and finish lines by using artificial arcs. In this setting, the problem can be modeled as a standard Arc Routing Problem (ARP) where the tour must visit a certain subset of arcs in the network as in Rural Postman Problem; however, the requirement to maintain access to critical facilities changes the problem significantly. We call this problem Lock-Free Arc Tour Problem (LFATP).

A critical component to marathon course design problem is measuring the quality of the course. In order to construct the correct objective function, it is necessary to answer the following question: 'What makes a course design better than the other?'. Quality can be measured by various performance metrics related to safety, medical access and runner experience leading to various objective function types. These types include (i) arc-additive ones for which the objective can be represented as the sum of coefficients of arcs used in the solution, (ii) sequence-dependent ones for which the contribution of each arc depends on the other arcs used in the solution, and (iii) time-dependent ones for which the coefficients of the arcs are functions of time and vary depending on the time the corresponding arcs are used.

Our research consists of three parts which are explained in detail in the following chapters: The first part models and solves the LFATP with arc-additive objective functions which are relatively more straightforward compared to sequence- and time-dependent ones. Initially, a mixed integer linear programming (MILP) formulation is proposed to solve this problem; however, the proposed MILP is found to be underperforming for some arc-additive objectives as it suffers from excessive subtour formation. Therefore, we introduce path-based reformulations of the initial formulation arising from a provably stronger disjunctive programming formulation which is obtained by exploiting visit requirements. These formulations greatly enhance our capability to solve arc-additive LFATPs, yielding up to 100 times improvements in solution times.

The second part introduces an interactive methodology to find a most preferred solution of a decision maker (DM) in a multiobjective setting. As marathon course design

involves multiple objectives, LFATP needs to be solved in a multiobjective setting. However, objective functions considered in marathon course design are generally related to intangible concepts such as health, safety and experience; therefore, it is not easy for a DM to compare or prioritize these objective functions. For this reason, we develop an interactive weight region-based approach (IWRA) that iteratively asks the DM to compare solutions rather than objective functions and indirectly translates these comparisons to preference information. Different from existing approaches, our design separates preference refinement and solution generation processes completely, scaling IWRA to optimization problems with large number of objectives. While we develop IWRA to solve multiobjective LFATP, proposed algorithms are applicable to a wide range of optimization problems.

The third part has two main contributions. First, it extends the initial LFATP models and approaches to more complex sequence- and time-dependent objectives. Extending arc additive objectives to sequence-dependent ones is relatively straightforward as it requires introducing a new set of variables and the corresponding linking inequalities. For time-dependent functions, the models can differ significantly depending on how time-dependence is interpreted. Second, it combines IWRA with LFATP to generate an interactive marathon course catalog for the race organizers. To achieve this, generalized versions of LFATP formulations are constructed and the formulation to be solved is determined based on the dominant structure of the aggregated objective.

1.3. The Chicago Model and Related Studies

The Bank of America Chicago Marathon is one of the largest marathons worldwide. Hosted in a city of 2.7 million residents, the marathon has 45,000 runners and 1.7 million spectators. See Suozzo (2002) for a detailed discussion on the evolution of Chicago Marathon and its course. The BACM has been described by organizers as a 'planned disaster': over a five-to-seven hour span, a minimum of 1,000 people seek medical treatment. Extreme heat forced the cancellation of the 2007 BACM in the middle of the race. As a result, the organizers, Chicago Event Management (CEM), developed a holistic approach to mass participation event planning and management, referred to as the Chicago Model (CM), which has become a standard for medical coverage for such events. The Chicago Model brings together all major organizations (e.g., race organizers, fire and police departments, emergency management, Red Cross) to coordinate preparation and response for the event and the surrounding areas impacted by the event. This integrated organizational structure is complemented with a comprehensive medical tracking system to monitor medical coverage in real time. This approach is being adopted by similar events in other cities, and lessons learned from the marathon experience can be applied to disaster response in general, see (McCarthy et al., 2011).

The CM creates an environment where all participating agencies contribute to preplanning and execution operations so that the agencies agree to their responsibilities in such chaotic environments and achieve a shared mental model of responsibilities (McCarthy et al., 2011; Stout et al., 1999). The CM has three key components: organizational structure, information systems, and communication. Organizational structure of the CM is based on Incident Command System (ICS) which is a top-down structure used to manage

incidents and large-scale disasters. Races are managed from a single command center (Forward Command). During the race, the CM uses multiple sources of information and collectively shares it in the command center with effective communication systems and hierarchies. See Başdere et al. (2014) for a detailed description of the CM.

Apart from marathon course design, our research team worked on several projects in collaboration with the race organizers to improve the capabilities of the CM further. These projects include field observations, patient (runner) tracking analysis and a comprehensive data visualization tool, and they lead to multiple publications in medical journals. While these studies are not directly related to marathon course design, experience from the field observations and insights from collaborations with the marathon organizers helped us in defining marathon course design problem clearly and precisely.

The first project is a case report from the field observations at 2013 BASS (Başdere et al., 2014). It explains the details of the CM and describes how the organizers used different components of the CM to respond to an acute incident caused by a man who was threatening to jump off the State Street Bridge approximately one and a half hours before the race starts. Originally State Street Bridge was on the race course; therefore, the organizers decided to reroute the course. With the help of the CM, rerouting operations were handled very efficiently, causing only 2-minute delay to the planned race start. The incident shows the dynamic nature of marathon planning and execution operations as well as the importance of the CM.

The second project examines the potential value of developing real-time patient tracking systems at marathons by using the data collected by the Chicago Medical Patient Tracking System (CMPTS) during the 2012 BACM (pilot race) and the 2013 BASS (Ross

et al., 2015). The study provides a summary of descriptive statistics for common diagnoses and admission trends at medical facilities along with a comparison between the current data collection system, which is based on paper forms, and the CMPTS. The results show that medical patient tracking systems such as the CMPTS improve data accuracy, which can potentially lead to improvements in operational decisions during planning and execution stages.

The third project is an evolving situational awareness tool based on a comprehensive data visualization system for marathons and other mass participation endurance events. This project is a collaboration of a large number of students through various student projects over the last five years. The system incorporates critical data into a user-friendly dashboard to serve as a centralized source of information during the events. It uses historical and real-time data to provide pre-event and on-site analytics via descriptive, predictive and prescriptive models. These models help race organizers and relevant stakeholders effectively manage and oversee all participants, monitor the dynamic location of race participants, and manage health and safety resources throughout the event. The system has been successfully deployed at 2014-2017 Chicago Marathon, 2014-2018 Shamrock Shuffle and 2016-2018 Houston Marathon. See Hanken et al. (2016) and Başdere et al. (2018) for detailed discussions of early and recent versions of the data visualization system, respectively.

CHAPTER 2

Lock-Free Arc Tour Problem with Arc-additive Objectives and Path-based Reformulations

This chapter introduces the Lock-Free Arc Tour Problem (LFATP), a novel tour finding problem which ensures that the resulting tour does not block access to certain critical vertices. This problem is motivated by our work in marathon course design. The quality of a course can be measured by various performance metrics related to safety, health and race experience, leading to different objective function forms. This chapter focuses on arc-additive objective functions where the objective function is an additive function of the arcs in the tour, based on our motivating metric of average distance to the nearest medical facility. The proximity to medical care (i.e. hospitals) throughout the race is a critical consideration given the number of injuries and the risk of exercise related collapse (Chiampas and Jaworski, 2009). Equally important, the course itself should not *lock* hospitals and other critical facilities within the course, thereby reducing access for the general public. In this chapter, we introduce a new arc routing problem to design marathon courses with both factors in mind. While the discussion throughout the chapter focuses on arc-additive objectives, Chapter 4 discusses extensions to sequence-dependent metrics for which the objective contributions of arcs are dependent on adjacent arcs in the tour (e.g., incline change or number of sharp turns) and time-dependent metrics for

which the objective contributions of arcs change over time (e.g., shade intensity along the course).

Marathon courses often are expected to visit certain neighborhoods for the benefit of runners and the host city. As such, the course design problem resembles arc routing problems (ARPs) where the route must visit a predefined subset of edges in the network (Corberán and Laporte, 2015). ARPs find a minimum cost (or length) route which traverses a specified arc subset M, which we call must-visit arcs, of the corresponding network. In our marathon setting, the course can be considered as a tour as one of the arcs in M contains the start and the finish lines of the race. For record-eligible marathons, the distance between the finish and the start lines, measured along a theoretical straight line between them, cannot be more than 50% of the race length (IAAF, 2015). Further, organizers often prefer a short distance between the start and the finish lines to consolidate resources and personnel. Generally, ARP-related studies differ based on the underlying network structure, vehicle restrictions and time dependence. In this study, we introduce a new ARP variant, the Lock-Free Arc Tour Problem (LFATP), which considers the vertices locked by the resulting tour. Unlike many ARPs, the LFATP does not allow multiple traversals of the arcs and vertices of the network; i.e., the LFATP finds a simple tour. The LFATP is defined on a directed graph $G(V, A)$ with border vertex subset $B \subset V$, defining the region. We define border vertices on boundaries of the network (see Figure 2.1), yet they can be anywhere depending on the corresponding problem. A critical vertex $q \in V \setminus B$ is said to be *locked* by a tour T if no $q - b$ path exists for any vertex $b \in B$ without passing through at least one vertex in T. In other words, q is locked if there is no connectivity between vertex q and set B once the vertices in T and associated outgoing

and incoming arcs are removed from the network. The LFATP finds a minimum cost simple tour of specific length that visits arcs in a subset $M \subset A$ while not locking critical vertices in the subset $Q \subset (V \setminus B)$.

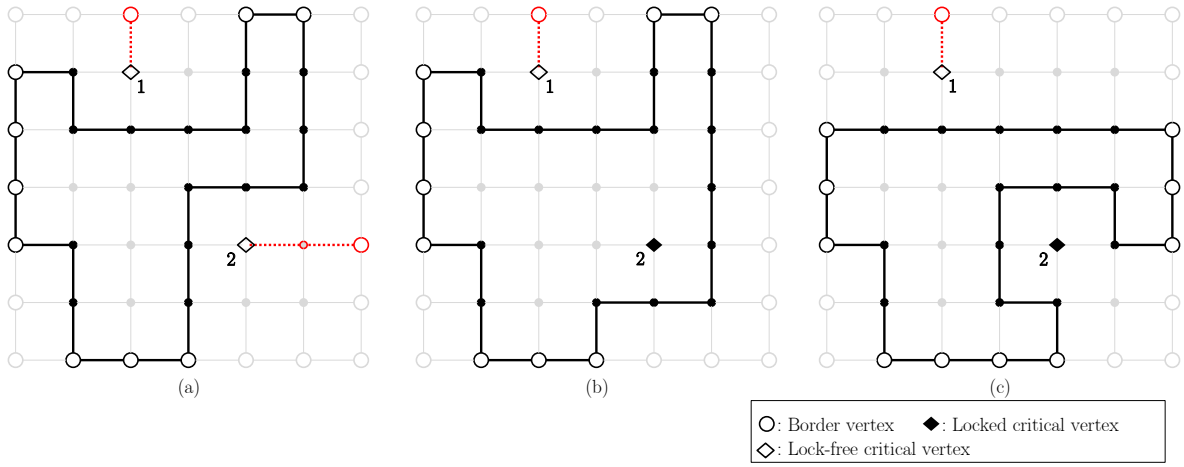


Figure 2.1. Sample LFATP solutions: (a) Feasible LFATP tour; (b) Infeasible LFATP tour due to internal locking; (c) Infeasible LFATP tour due to blocked access.

Figure 2.1 provides examples of LFATP solutions. Vertices in B are represented with white circles; the two critical vertices are represented with diamonds. In the feasible tour in Figure 2.1a, both critical vertices can be connected to a border vertex without intersecting the vertices of the tour. In the infeasible tour in Figure 2.1b, there is no feasible path to connect critical vertex 2 to any vertex in B without intersecting the vertices on the tour. In the infeasible tour in Figure 2.1c, both critical vertices are outside the inner region of the tour, yet all paths for critical vertex 2 are blocked by the tour.

The contributions of this study are twofold. First, a new ARP variant is introduced. Although the LFATP is similar to the prize-collecting arc routing problem, it is a novel tour finding problem with constraints which impose restrictions on the elements blocked

by the tour. To the best of our knowledge, such restrictions have not been studied. Valid inequalities are derived from the locking properties in subgraphs. A heuristic separation algorithm is developed to identify these valid inequalities since the proposed separation is shown to be NP-Hard. Second, in exploring ways to solve this challenging problem, we develop path-based reformulations which arise from a provably stronger disjunctive programming (DP) formulation that exploits the tour length bounds to efficiently solve the LFATP by enforcing visit orders of must-visit arcs. We demonstrate our ability to generalize this approach for similar tour finding problems which share some characteristics of the LFATP through a series of computational experiments.

The remainder of the chapter is organized as follows: In Section 2.1, related ARP studies are discussed. In Section 2.2, the mixed integer linear programming (MILP) formulation of the LFATP is presented. In Section 2.3, the path-based formulation of the LFATP is proposed along with a discussion of reformulations. In Section 2.4, valid inequalities are derived for the LFATP and their separation is described. In Section 2.5, numerical experiments from test instances and the Bank of America Chicago Marathon (BACM) are provided and the study is concluded with a discussion on future research directions in Section 2.6.

2.1. Literature Review

The LFATP falls into the broader class of ARPs; see Dror (2000), Corberán and Prins (2010) and Corberán and Laporte (2015) for comprehensive reviews of ARP literature. To the best of our knowledge, there is no study which considers vertex connectivity issues such as the vertex access restrictions of the LFATP. Furthermore, the LFATP differs

from classical ARPs since its tour length is bounded from below and above, and the tour must be a simple cycle. Ignoring these restrictions, the LFATP can be classified as a variant of prize-collecting (or profitable) arc routing problems (PARP) (Aráoz et al., 2006). These problems maximize the profit obtained by traversing arcs, subject to various operational constraints. Unlike classical ARPs, prize-collecting versions do not impose visit requirements.

Aráoz et al. (2006) introduces the PARP as an arc routing counterpart of the prize-collecting traveling salesman problem (e.g. Golden et al. (1987); Laporte and Martello (1990)) in the context of the Privatized Rural Postman Problem (PRPP). The authors show that the PRPP is NP-Hard and establish relationships among arc routing and other related problems. Their ILP formulation is solved in Aráoz et al. (2009b) with both branch-and-cut (B&C) and heuristics. In a similar study, Aráoz et al. (2009a) consider a clustered PARP, visiting all or none of the edges in a cluster. This problem, again, is solved with B&C using newly identified cuts along with the cuts derived in Aráoz et al. (2006). Corberán et al. (2011) analyze the polyhedral structure of the PARP with direction-dependent costs and provide valid inequalities along with heuristic and exact separation algorithms. They solve the problem by using cut-and-branch iteratively where cuts are identified either at the root node or at the end of branch-and-bound (B&B) solution. Aráoz et al. (2013) propose greedy randomized adaptive search procedure (GRASP) and path relinking heuristics for the clustered PARP.

Tour length restrictions are often considered in capacitated ARPs (CARP) in multiple vehicle settings as in capacitated VRPs (Corberán and Prins, 2010; Bartolini et al., 2013b). Feillet et al. (2005) focus on a PARP that finds a set of cycles to maximize profit with a

restriction on the maximum cycle length. The authors propose an exact solution procedure which uses branch-and-price (B&P) with a novel branching method that branches on short sequences of arcs. Archetti et al. (2010) use a similar B&P algorithm and provide heuristics in the context of undirected CARP with profits. Bartolini et al. (2013a) study a version of CARP with capacity consumption caused by deadheading and provide an exact algorithm which uses cut and column generation. Black et al. (2013) consider PARP with a time-dependent objective function and propose metaheuristics; tour length is restricted implicitly through a restriction on the interval the vehicles can travel.

The node counterpart of the LFATP is closely related to the Orienteering Problem (OP). In the OP, each vertex has a specific score and the aim is to find a tour which maximizes the total score collected by visiting those vertices without exceeding the time limit (Golden et al., 1987). See Vansteenwegen et al. (2011) for a recent review. Many ARP studies utilize node transformations of ARPs: After solving the node counterpart problem, the solutions are transformed back to solutions for the original problem; see Laporte (1997). The studies of Deitch and Ladany (2000), Clossey et al. (2001), Corberán et al. (2002), and Irnich (2008) consider operational restrictions in ARPs using node transformations. Corberán et al. (2002) report that node transformation is outperformed by a direct heuristic approach on large instances. Feillet et al. (2005) also advise against node transformations, given the size of the resulting graph in their profitable ARP setting.

Souffriau et al. (2011) define the OP in the arc routing setting, the Arc OP (AOP). In AOP, each arc has an associated cost and score and, similar to the OP, the aim is to find the most profitable simple path (or cycle) without exceeding the cost limit. Archetti et al. (2013) introduce the Team Orienteering ARP (TOARP) extends AOP to multi-vehicle

setting. Archetti et al. (2013) and Archetti et al. (2015) propose exact and heuristic solution methods for TOARP while Archetti et al. (2014) investigate split deliveries in the node counterpart of TOARP. Cost limit restrictions of these problems and simple cycle restriction of AOP are similar to those of the LFATP; however, the LFATP has additional visit and length requirements and, most importantly, locking considerations.

The LFATP has a unique structure among ARPs because of its locking property; therefore, its classification is not trivial. We exploit the unique locking properties of the LFATP to obtain valid inequalities, and the visit and length requirements common in many ARPs to develop path-based reformulations.

2.2. Standard MILP Formulation

In this section, we provide an MILP formulation of the LFATP for a general arc-additive objective function. The constraints of the LFATP can be classified with two sets: Tour feasibility and lock prevention. Tour feasibility constraints ensure that the model finds a single subtour which satisfies the length and visit requirements. Lock prevention constraints ensure that each critical vertex $q \in Q$ is connected to one of the border vertices $b \in B$ with a feasible $q - b$ path. In Model 1, M is not an arc subset since traversal direction does not matter here; therefore, M is a set of edges (arc pairs) and e_{ij} is the edge representing the arc pair between nodes i and j. Let $A(S)$ and $V(S)$ denote the arcs and vertices within a given set S, respectively. $\delta(S)$ denotes the arcs with one vertex in S and one outside S.

Let c_{ij} and d_{ij} be the objective contribution and length of arc $(i, j) \in A$, respectively. Let x_{ij} be a binary variable which takes value 1 if arc $(i, j) \in A$ exists in the tour. We

introduce two sets of auxiliary variables. The first set is used to represent the subtour elimination inequalities in the LFATP setting. A feasible LFATP solution is a single subtour in the network; for a solution which uses k arcs, a subtour of less than k arcs is infeasible. Let h_k variables take value 1 if more than k arcs are used to construct the tour, $k \in \{2, 3, 4, ..., |V| - 1\}$. The second variable set is used to construct $q - b$ paths. Let y_{ij} be a nonnegative continuous variable representing the amount of flow, called lock feasibility flow, sent through arc $(i, j) \in A$ on $q-b$ paths. An arc $(i, j) \in A$ can be used in multiple connecting $q - b$ paths and flow on the arc can be fractional, indicating multiple paths from a critical vertex to border vertices. The LFATP is formulated as follows:

$$\text{(2.1a)} \quad \min \sum_{(i,j)\in A} c_{ij}x_{ij}$$

s.t.

$$\text{(2.1b)} \quad \sum_{j:(j,i)\in A} x_{ji} = \sum_{j:(i,j)\in A} x_{ij} \qquad i \in V,$$

$$\text{(2.1c)} \quad L \leq \sum_{(i,j)\in A} d_{ij}x_{ij} \leq L + \epsilon,$$

$$\text{(2.1d)} \quad x_{ij} + x_{ji} = 1 \qquad e_{ij} \in M,$$

$$\text{(2.1e)} \quad \sum_{(i,j)\in A} x_{ij} \leq |V|h_k + k \qquad k \in \{2, 3, 4, ..., |V| - 1\},$$

$$\text{(2.1f)} \quad \sum_{\substack{i\in S,\, j\in S \\ (i,j)\in A}} x_{ij} \leq |S| - h_{|S|} \qquad S \subset V,\ |S| \geq 2,$$

$$\text{(2.1g)} \quad \sum_{j:(q,j)\in A} y_{qj} - \sum_{j:(j,q)\in A} y_{jq} = 1 \qquad q \in Q,$$

$$\sum_{b\in B}\Big(\sum_{i:(i,b)\in A} y_{ib} - \sum_{i:(b,i)\in A} y_{bi}\Big) = |Q|, \tag{2.1h}$$

$$\sum_{j:(j,i)\in A} y_{ji} = \sum_{j:(i,j)\in A} y_{ij} \qquad i \in V \setminus (Q \cup B), \tag{2.1i}$$

$$\sum_{j:(i,j)\in A} y_{ij} + \sum_{j:(j,i)\in A} y_{ji} \leq 2|Q|\Big(1 - \sum_{l:(l,i)\in A} x_{li}\Big) \qquad i \in V, \tag{2.1j}$$

$$x_{ij} \in \{0,1\} \qquad (i,j) \in A, \tag{2.1k}$$

$$h_k \in \{0,1\} \qquad k \in \{2,3,...,|V|-1\}, \tag{2.1l}$$

$$y_{ij} \geq 0 \qquad (i,j) \in A. \tag{2.1m}$$

We refer to Model 2.1 as *LFATP-S* to distinguish this [S]tandard formulation from reformulations in Section 2.3. The objective function (2.1a) is a minimization over a general arc additive measure.

Constraints (2.1b)-(2.1f) are tour feasibility constraints. Constraints (2.1b) guarantee incoming and outgoing arc balance at each vertex. Constraints (2.1c) and (2.1d) ensure that tour length and visit requirements are satisfied, respectively. Setting the tour length equal to a single target value results in a highly restricted feasible region with few (if any) solutions. Therefore, we set an acceptable interval $[L, L+\epsilon]$ where L is the target length and ϵ is the maximum allowed positive deviation from this target length. In the motivating marathon setting, ϵ represents the allowable distance between start and finish lines which is assumed to be smaller than the length of the edge in M which contains these lines. Constraint (2.1c) can be generalized to maximum tour length or maximum duration constraints in OP by setting $L = 0$ and ϵ to the limit value, and constraints (2.1d) can be generalized to single direction requirements. Constraints (2.1e) and (2.1f) together ensure

that the solution is a single subtour. The left-hand side of constraints (2.1e) represents the number of arcs in the solution. For a solution of k' arcs, $\sum_{(i,j)\in A} x_{ij} = k'$ and constraints (2.1e) set $h_k = 1$ for all $k < k'$. Constraints (2.1f) form classical subtour elimination constraints for any subset S with $|S| < k'$.

Constraints (2.1g)-(2.1j) are lock prevention constraints. Constraints (2.1g) guarantee that one unit of net lock feasibility flow is sent from each $q \in Q$ and constraint (2.1h) ensures that all lock feasibility flow paths arrive to a border vertex in B. Constraints (2.1i) are the flow balance constraints for lock feasibility flow paths creating connected paths from each $q \in Q$ to border vertices. Constraints (2.1j) are linking constraints which link tour and lock feasibility flow path variables. If vertex i is in the tour, then there exists an incoming arc to vertex i, say $(\hat{l}, i)$, with $x_{\hat{l}i} = 1$. Then, the right-hand side of (2.1j) is 0 and no lock feasibility flow can enter or leave i. If vertex i is not on the tour, the right-hand side takes value of $2|Q|$ and corresponding y_{ij} variables can be used in $q-b$ paths.

Proposition 2.1. *The LFATP is NP-Hard.*

Proof of Proposition 2.1. Consider the NP-Hard quota traveling salesman problem (quota TSP) with arc traversal cost c'_{ij}, $(i, j) \in A$, and vertex prize s'_i, $i \in V$, with no penalty for unvisited vertices (Awerbuch et al., 1998). The quota TSP finds the minimum cost simple tour such that total prize of the vertices in the tour exceeds a predetermined level of P'. This problem can be transformed to the LFATP in polynomial time, letting $Q = \emptyset$, $M = \emptyset$, $L = P'$, $\epsilon = \sum_{i\in V} s'_i$, $c_{ij} = c'_{ij}$ and $d_{ij} = \frac{s'_i + s'_j}{2}$. Constraints (2.1d), (2.1m) and (2.1g)-(2.1j) are trivially satisfied since $Q = \emptyset$ and $M = \emptyset$. The right-hand side of

constraint (2.1c) is trivially satisfied as ϵ is set to $\sum_{i \in V} s'_i$ and a simple tour cannot visit a vertex more than once. Letting $A(T)$ and $V(T)$ be the arcs and vertices of LFATP solution T, $\sum_{(i,j) \in A(T)} d_{ij} = \sum_{i \in V(T)} s'_i$ since each vertex in the solution has exactly one incoming and one emanating active arc. The optimal solution of the LFATP is the minimum cost simple subtour whose total vertex prize is greater than P'; i.e., the optimal solution of quota TSP. Thus, the LFATP is NP-Hard. □

2.3. Disjunctive Programming Formulation and Path-based Reformulations

The LFATP can be solved with a standard B&C approach on the standard model, *LFATP-S*, utilizing valid inequalities designed for the locking constraints (see Section 2.4.2). Even with these valid inequalities, excessively many subtour elimination inequalities can be required. We exploit the fundamental idea that the edges in M can be connected with paths to form optimal tours and we develop provably stronger path-based reformulations to solve LFATP. These reformulations are shown to have promise for many similar arc tour problems.

Our initial path-based reformulation is constructed as a DP where each disjunction imposes a visit order on the must-visit edges and connects these edges with paths satisfying the order. Importantly, these paths consume a large portion of the *length budget* specified by constraint (2.1c). Let O_M be the set of visit orders. An element $o \in O_M$ is an ordered set of edges in M, $o_{(1)}, ..., o_{(|M|+1)}$, where $o_{|M|+1} = o_{(1)}$. Letting L_o be the length of the shortest tour that connects edges in M in visit order o, the remaining length budget for arcs that are not used to connect edges in M reduces to $L + \epsilon - L_o$, significantly restricting the potential for subtours in the solution.

Let $u_{o(m)}$ and $v_{o(m)}$ be the endpoints of edge $o_{(m)}$ for visit order $o \in O_M$ (for easier representation, we abuse notation slightly and use u_m and v_m instead of $u_{o(m)}$ and $v_{o(m)}$). Let $\bar{A}$ be the set of arcs constructed by removing arcs $A(M)$ from A. Similarly, let $\bar{V}$ be the set of vertices constructed by removing $V(M)$ from V. Let z_{ij} be a binary variable that takes value 1 if edge $e_{ij} \in M$ is traversed in the direction of i to j and 0 otherwise. Let $\bar{x}_{ijm}$ be a binary variable that takes value 1 if arc $(i,j) \in \bar{A}$ is on the path between $o_{(m)}$ and $o_{(m+1)}$.

The path-based model, *LFATP-P*, is formulated as a DP where the selection of visit order o specifies the disjunction. In the model below, constraint (2.2k) is the disjunctive constraint which ensures that a visit order o from the set of all visit orders is selected and the model respects selected order when constructing the paths:

$$\text{(2.2a)} \quad \min \sum_{e_{ij} \in M} (c_{ij} z_{ij} + c_{ji} z_{ji}) + \sum_{(i,j) \in \bar{A}} \sum_{m=1}^{|M|} c_{ij} \bar{x}_{ijm}$$

s.t.

$$\text{(2.2b)} \quad \sum_{j:(j,i) \in \bar{A}} \bar{x}_{jim} = \sum_{l:(i,l) \in \bar{A}} \bar{x}_{ilm} \qquad i \in \bar{V},\ m \in \{1, ..., |M|\},$$

$$\text{(2.2c)} \quad \sum_{j:(j,i) \in A(M)} z_{ji} + \sum_{j:(j,i) \in \bar{A}} \sum_{m=1}^{|M|} \bar{x}_{jim} = \sum_{l:(i,l) \in A(M)} z_{il} + \sum_{l:(i,l) \in \bar{A}} \sum_{m=1}^{|M|} \bar{x}_{ilm} \qquad i \in V \setminus \bar{V},$$

$$\text{(2.2d)} \quad \sum_{j:(v_m,j) \in \bar{A}} \bar{x}_{v_m jm} + \sum_{j:(u_m,j) \in \bar{A}} \bar{x}_{u_m jm} = 1 \qquad m \in \{1, ..., |M|\},$$

$$\text{(2.2e)} \quad \sum_{i:(i,u_{(m+1)}) \in \bar{A}} \bar{x}_{i u_{(m+1)} m} + \sum_{i:(i,v_{(m+1)}) \in \bar{A}} \bar{x}_{i v_{(m+1)} m} = 1 \qquad m \in \{1, ..., |M|\},$$

$$z_{ij} + z_{ji} = 1 \qquad e_{ij} \in M, \tag{2.2f}$$

$$L \leq \sum_{(i,j)\in\bar{A}} \sum_{m=1}^{|M|} d_{ij}\bar{x}_{ijm} + \sum_{(i,j)\in A(M)} d_{ij} z_{ij} \leq L + \epsilon, \tag{2.2g}$$

$$\sum_{e_{ij}\in M} (z_{ij} + z_{ji}) + \sum_{(i,j)\in\bar{A}} \sum_{m=1}^{|M|} \bar{x}_{ijm} \leq |V| h_k + k \qquad k \in \{2, 3, ..., |V| - 1\}, \tag{2.2h}$$

$$\sum_{\substack{i\in S,\, j\in S \\ e_{ij}\in M}} (z_{ij} + z_{ji}) + \sum_{\substack{i\in S,\, j\in S \\ (i,j)\in\bar{A}}} \sum_{m=1}^{|M|} \bar{x}_{ijm} \leq |S| - h_{|S|} \qquad S \subset V,\ |S| \geq 2, \tag{2.2i}$$

$$\sum_{j:(i,j)\in A} y_{ij} + \sum_{j:(j,i)\in A} y_{ji}$$

$$\leq 2|Q|\Big(1 - \sum_{m=1}^{|M|} \sum_{l:(l,i)\in\bar{A}} \bar{x}_{lim} - \sum_{l:(l,i)\in A(M)} z_{li}\Big) \qquad i \in V, \tag{2.2j}$$

$$(2.1g) - (2.1i), (2.1m), (2.1l),$$

$$o \in O_M, \tag{2.2k}$$

$$\bar{x}_{ijm} \in \{0, 1\} \qquad (i,j)\in\bar{A}, m\in\{1, ..., |M|\}, \tag{2.2l}$$

$$z_{ij}, z_{ji} \geq 0 \qquad e_{ij} \in M. \tag{2.2m}$$

Objective (2.2a) minimizes arc-additive costs of edges in M (either direction) and paths connecting these edges. For the vertices in $\bar{V}$, the balance constraints (2.2b) are similar to constraints (2.1b), further requiring that the incoming and emanating arcs must belong to the same path. The balance constraints (2.2c) ensure arc balance for the endpoints of the edges in M relating $\bar{x}$ and z variables. Constraints (2.2d) guarantee that the m^{th} path starts from an endpoint of m^{th} must-visit edge in visit order o and

constraints (2.2e) guarantee that the m^{th} path ends at an endpoint of $(m+1)^{st}$ must-visit edge in visit order o. With balance constraints (2.2c), one endpoint of $(m+1)^{st}$ must-visit edge becomes the terminal point for m^{th} path whereas the other endpoint becomes the initial point for $(m+1)^{st}$ path. Similar to constraints (2.1d), constraints (2.2f) ensure that the edges in M are traversed in one direction. Constraint (2.2g) is the tour length constraint. Similar to constraints (2.1e) and (2.1f), constraints (2.2h) and (2.2i) eliminate subtour formation. Lock prevention constraints (2.1g)-(2.1i) are added directly and x_{li} variables in the linking constraints (2.1j) are replaced with z_{li} and $\sum_{i=1}^{|M|} \bar{x}_{lim}$ for the edges in set M and arcs $\bar{A}$, respectively.

Theorem 2.2. *LFATP-P is a stronger formulation than LFATP-S.*

Proof of Theorem 2.2. Let P_S and P_P be the polytopes representing the linear relaxations of *LFATP-S* and *LFATP-P*, in affine spaces of (x, y, h) and $(\bar{x}, y, z, h)$ variables, respectively. Note that the solution spaces of P_S and P_P are not the same; we define P_P^+ as follows:

$$P_P^+ = \Big\{(x, \bar{x}, y, z, h)\Big|(\bar{x}, y, z, h) \in P_P; \quad x_{ij} = \sum_{m=1}^{|M|} \bar{x}_{ijm},\ (i,j) \in \bar{A};\ \ x_{ij} = z_{ij},\ x_{ji} = z_{ji},\ e_{ij} \in M\Big\}.$$

P_P^+ extends polytope P_P into the affine space of $(x, \bar{x}, y, z, h)$ variables relating x variables with $\bar{x}$ and z variables with affine transformations. Let $Proj_{x,y,h}(P_P^+)$ be the projection of P_P^+ onto the affine space of (x, y, h) variables. Constructed this way, comparing P_S with P_P reduces to comparing P_S with $Proj_{x,y,h}(P_P^+)$ (Balas, 2005).

To prove that *LFATP-P* is a stronger formulation than *LFATP-S*, we need to show that $Proj_{x,y,h}(P_P^+) \subseteq P_S$ and there exists an instance for which the inclusion is proper. First, we show that $Proj_{x,y,h}(P_P^+) \subseteq P_S$. Consider $(\bar{x}^s, y^s, z^s, h^s) \in P_P$ for an arbitrary $o \in O_M$ and let $(x^s, y^s, h^s) \in Proj_{x,y,h}(P_P^+)$ be the projected solution corresponding to $(\bar{x}^s, y^s, z^s, h^s)$. We now show that $(x^s, y^s, h^s) \in P_S$. Since $(\bar{x}^s, y^s, z^s, h^s) \in P_P$, it satisfies all constraints defining P_P. Replacing $\sum_{m=1}^{|M|} \bar{x}_{ijm}$ and z_{ij} variables with x_{ij} variables in constraints (2.2g), (2.2f), (2.2h), (2.2i) and (2.2j), these constraints reduce to (2.1c), (2.1d), (2.1e), (2.1f) and (2.1j) of P_S, respectively, ensuring that (x^s, y^s, h^s) satisfies (2.1c), (2.1d), (2.1e), (2.1f) and (2.1j). Summing up constraints (2.2b) over $m = 1, ..., |M|$ and applying the same variable replacement to the resulting expression together with constraints (2.2c) ensure that (x^s, y^s, h^s) satisfies constraints (2.1b) of P_S. (x^s, y^s, h^s) also satisfies constraints (2.1g)-(2.1i) since these constraints are already considered when defining P_P. In terms of variable restrictions, it is clear that $y^s \geq 0$ and $0 \leq h^s \leq 1$ since $(\bar{x}^s, y^s, z^s, h^s) \in P_P$. Furthermore, we know that $x^s \geq 0$ since $\bar{x}^s \geq 0$ and $x^s \leq 1$ as constraints (2.2i) ensure that the sum of active variables for subsets of size 2 cannot be greater than 1. (x^s, y^s, h^s) satisfies all constraints defining P_S, i.e., $(x^s, y^s, h^s) \in P_S$ indicating that $Proj_{x,y,h}(P_P^+) \subseteq P_S$.

Since $Proj_{x,y,h}(P_P^+) \subseteq P_S$, constructing a fractional solution $(x^f, y^f, h^f) \in P_S$ such that $(x^f, y^f, h^f) \notin Proj_{x,y,h}(P_P^+)$ is sufficient to prove that inclusion is proper. Consider $(\bar{x}^s, y^s, z^s, h^s) \in P_P$ for an arbitrary visit order $o \in O_M$ and let $e_{kl} \in M$ be the $(m')^{th}$ must-visit edge in o. Considering constraint (2.2d) for m', the following equality can be

written for e_{kl} (note that $u(m')$ and $v(m')$ are indices k and l respectively):

$$\sum_{j:(l,j)\in\bar{A}} \bar{x}^s_{ljm'} + \sum_{j:(k,j)\in\bar{A}} \bar{x}^s_{kjm'} = 1. \tag{2.3}$$

For any visit order o, there exists an $m' \in \{1, ..., |M|\}$ such that equality (2.3) holds for each $e_{kl} \in M$. Therefore, the following must hold for any $e_{kl} \in M$ for any visit order $o \in O_M$:

$$\sum_{m=1}^{|M|} \sum_{j:(l,j)\in\bar{A}} \bar{x}^s_{ljm} + \sum_{m=1}^{|M|} \sum_{j:(k,j)\in\bar{A}} \bar{x}^s_{kjm} \geq 1, \qquad e_{kl} \in M. \tag{2.4}$$

Considering the affine transformation for $(i,j) \in \bar{A}$, $\sum_{m=1}^{|M|} \bar{x}_{ijm} = x_{ij}$, any solution $(x^s, y^s, h^s) \in Proj_{x,y,h}(P_P^+)$ must satisfy the following inequalities:

$$\sum_{j:(l,j)\in\bar{A}} x^s_{lj} + \sum_{j:(k,j)\in\bar{A}} x^s_{kj} \geq 1, \qquad e_{kl} \in M. \tag{2.5}$$

Using the same arguments for constraints (2.2e), the following must also hold for $(x^s, y^s, h^s) \in Proj_{x,y,h}(P_P^+)$:

$$\sum_{i:(i,l)\in\bar{A}} x^s_{il} + \sum_{i:(i,k)\in\bar{A}} x^s_{ik} \geq 1, \qquad e_{kl} \in M. \tag{2.6}$$

Therefore, constructing a fractional solution $(x^f, y^f, h^f) \in P_S$ such that at least one of the inequalities (2.5) and (2.6) is violated is sufficient to show proper inclusion. Such solutions can be constructed by isolating an arbitrary must-visit edge $e_{kl} \in M$ as follows: Set $z_{kl} = z_{lk} = 0.5$ and find a fractional LFATP solution which does not use the arcs

in $\delta(\{k\} \cup \{l\})$. Constructed this way $\sum_{j:(l,j)\in\bar{A}} x^f_{lj} + \sum_{j:(k,j)\in\bar{A}} x^f_{kj} = \sum_{i:(i,l)\in\bar{A}} x^f_{il} + \sum_{i:(i,k)\in\bar{A}} x^f_{ik} = 0$ violating both (2.5) and (2.6) for e_{kl}. (See Figure 2.2). □

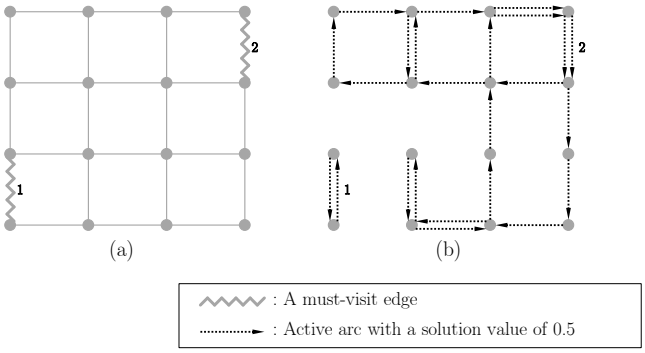


Figure 2.2. Example for proper inclusion: (a) The underlying network with unit arc length and two must-visit edges, (b) A fractional solution with disconnected must-visit edges.

Figure 2.2 provides an example for proper inclusion. Consider the network in Figure 2.2a with unit arc length and two must-visit edges. For simplicity, the network does not include any critical vertices. Assuming that an LFATP solution of 12 units is feasible with respect to length requirements, the fractional solution provided in Figure 2.2b is feasible for *LFATP-S* as it satisfies constraints (2.1b)-(2.1f). However, the same fractional solution is infeasible for *LFATP-P* as inequalities (2.5) and (2.6) are not satisfied for must-visit edge 1.

Theorem 2.2 indicates that connecting edges in M with paths potentially eliminates a subset of fractional solutions from *LFATP-S* polytope, P_S. *LFATP-P* uses a large fraction of the length budget when connecting the edges in M, tightening both left and right hand-sides of constraint (2.1c). Therefore, *LFATP-P* results in a stronger formulation for the LFATP as constraints (2.2b)-(2.2e) guarantee that the edges in M are connected to each other even at fractional solutions. This indicates that the DP-based idea show promise for

a broader range of arc routing problems with length budget restrictions, including those without locking constraints as the locking property is not a key feature of path-based reformulations. Additionally, tour length minimization problems without explicit bounds on tour length (i.e., without Constraints (2.1c)) have an implicit length budget in the decision problem which broadens the applicability further.

2.3.1. Solving LFATP-P

2.3.1.1. Reformulation Basics. As a disjunctive program, *LFATP-P* can be partitioned into subproblems with fixed visit orders. Once the visit order o is fixed, the resulting subproblem, *LFATP-P(o)*, is Model 2.2 without constraint (2.2k) which is a standard MILP. A straightforward approach to solve *LFATP-P* would be to solve these subproblems sequentially while benefiting from previously found bounds on the objective function. Each (feasible) subproblem returns an optimal LFATP solution over the corresponding partition of the original feasible region, yielding an upper bound for the LFATP. When solving the subproblems, we can add a constraint which ensures that the objective function value of the subproblem to be solved is bounded by the best upper bound obtained from the previous subproblems. This way, the current subproblem is solved to optimality only if it improves the best upper bound (not necessarily strictly); otherwise, it becomes infeasible and is pruned faster as the added constraint is not satisfied. Once all subproblems are solved, the LFATP solution yielding the best upper bound is an optimal solution for the LFATP. An alternative approach to avoid enumerating all subproblems would be to solve the MILP representation of *LFATP-P* resulting from a (big-M, Beaumont or convex hull) relaxation of the disjunctive set O in constraints (2.2k) (Vecchietti et al., 2003).

Both approaches can be impractical due to potentially large number of disjunctions as $|O_M|$ increases faster than exponential rate as $|O_M| = \frac{1}{2}(|M|-1)!$ for $|M| > 2$ for symmetric cost structure. For instance, when $|M| = 6$, $|M| = 12$ and $|M| = 24$, the number of disjunctions are $|O_M| = 60$, $|O_M| \approx 2.00 \times 10^7$ and $|O_M| \approx 1.29 \times 10^{22}$, respectively.

To eliminate the dependence on the number of must-visit edges, *LFATP-P* can be reformulated by using a g-edge subset of must-visit edges $M_g \subseteq M$, $1 \leq g \leq |M|$ when $|M|$ is high. In this case, visit orders are replaced by *partial* visit orders where the latter contain only the edges from subset M_g. When using partial visit orders, the following constraints must be satisfied for edges in $M \setminus M_g$:

$$\sum_{m=1}^{g} (\bar{x}_{ijm} + \bar{x}_{jim}) = 1, \qquad e_{ij} \in M \setminus M_g, \tag{2.7}$$

which ensures that each edge in $M \setminus M_g$ is visited by one path that connects the edges in M_g. Constructed this way, the resulting reformulation is a hybrid between *LFATP-P* and *LFATP-S* which connects edges in M_g with paths as in *LFATP-P* and ensures that the remaining edges in $M \setminus M_g$ are visited by the resulting tour as in *LFATP-S*. Using a relatively small M_g instead of M reduces the number of disjunctions to be considered as complete visit orders are replaced with partial ones. Letting $o' \in O_{M_g}$ be a partial visit order formed by edges in M_g where O_{M_g} is the set of all partial orders constructed from edges in M_g, solving subproblem *LFATP-P(o')* with constraints (2.7) finds the best solution among the set of complete visit orders where o' is respected.

With partial visit orders, the total number of subproblems to be solved is reduced from $|O_M|$ to $|O_{M_g}|$, which is significant for large $|M|$. Apart from solving fewer subproblems, the size of each subproblem is smaller since $\bar{x}$ variables are used to form $|M_g| = g$ paths

instead of $|M|$. However, using subsets of M decreases length budget consumption which creates a trade-off between the size of the disjunctive set and the strength of the resulting reformulations: Large subsets of M yield stronger reformulations with large disjunctive sets while smaller subsets yield weaker reformulations with smaller disjunctive sets.

Since using partial visit orders potentially decreases the strength of the resulting formulation, we can increase the strength of small subsets by strategically choosing (i) the edges in M_g and (ii) the order in which to solve the resulting subproblems. Selecting farthest g edges is a promising option since connecting such edges consumes a relatively large portion of the length budget. This strategy is based on results for TSPs in lower dimensional spaces in which the optimal tour lengths of the worst case problems increase at a relatively slow rate in the number of edges (vertices) to be visited (Few, 1955; Karloff, 1989). Unless the length budget is notably larger than the minimum tour length of the underlying problem, significant reductions in subtour formation can be achieved even with few edges from M. In terms of solving order, choosing partial orders which potentially yield quality solutions with tight upper bounds can decrease solution times significantly, pruning the subsequent problems quickly. With these insights, we first develop a generalized DP-based B&C approach which takes the desired cardinality of M_g as an input and uses a high-level fixed-depth B&C tree that branches on must-visit edges to include in the tour, imposing visit orders with these edges at the nodes of the tree.

2.3.1.2. DP-based B&C Approach. The DP-based B&C approach is a high-level fixed depth B&C tree which branches on must-visit edges to include in the tour, imposing visit orders to these edges at the nodes. Figure 2.3 presents an example. The root node begins with the maximum number of edges from M that result in a single tour permutation

of visit orders for these edges. In the example, the root node of B&C tree contains three edges since they yield a single order when cost structure is symmetric, similar to following the vertices of a triangle from any start point to any direction. Let M_g denote the set of must-visit edges that must be in the tour where g is the number of edges in set M_g. For the root node, we have $M_3 = \{e_1, e_2, e_3\}$. Furthermore, one edge can be fixed as the starting and ending edge since we are finding circular permutations. W.l.o.g. we assume that e^1 is the fixed edge for the rest of the discussion.

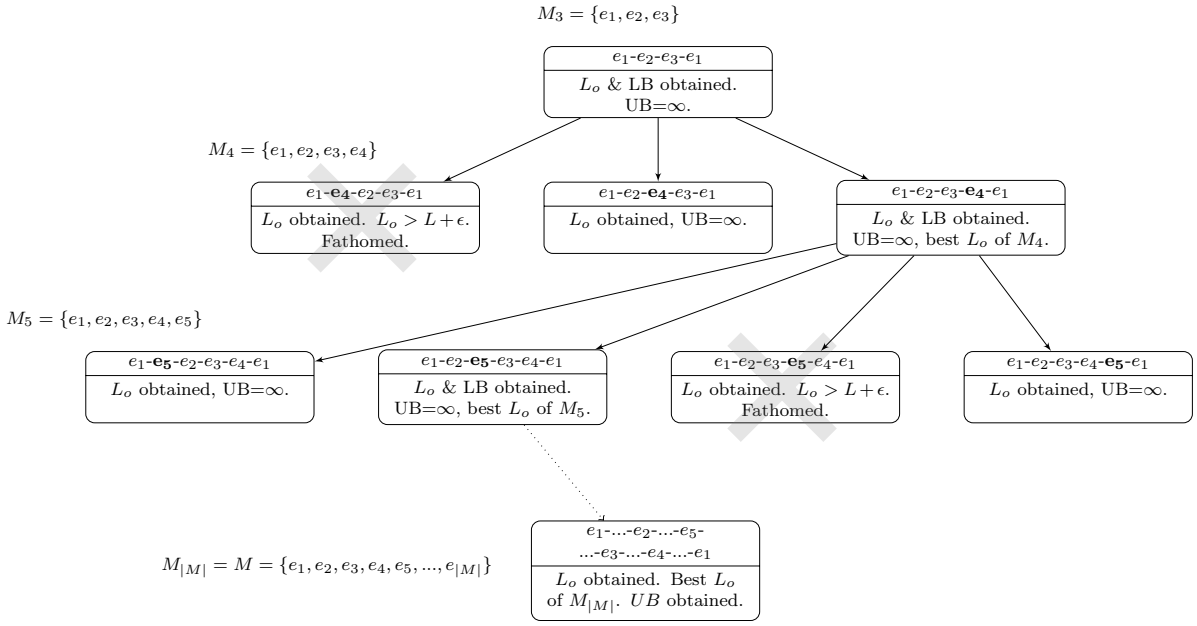


Figure 2.3. Sample DP-based B&C tree.

The DP-based B&C approach branches on must-visit edges; a new edge from $M \setminus M_g$ is selected during branching to join the set M_{g+1}. The selected edge e^{new} is used to generate all visit orders which contain edge e^{new} and M_g, and satisfy the visit order of the parent node. For example, branching on e^4 from the root node in Figure 2.3 generates three child nodes with visit orders e^1-e^4-e^2-e^3-e^1, e^1-e^2-e^4-e^3-e^1 and e^1-e^2-e^3-e^4-e^1 which satisfy the visit order of the root node, e^1-e^2-e^3-e^1. The branching edge is selected by finding the

edge from $M \setminus M_g$ which maximizes the average distance to the edges in M_g. As with the choice of edges at the root node, these edges become farther from each other and the paths connecting them become longer using a larger portion of the length budget and yielding a stronger subproblem. At the root node, the most distant two edges are selected first and then the edge which is farthest away from them is added to construct the initial visit order.

The branching continues until $|M_g| = g = \varphi$ where φ is a predetermined fixed depth parameter. When $|M_g| < \varphi$, the subproblem at the corresponding node is *LFATP-P(o′)* without constraints (2.7) where $o' \in O_{M_g}$. In this case, the edges in $M \setminus M_g$ are not required and *LFATP-P(o′)* yields a lower bound. If this lower bound exceeds the global upper bound, the node can be fathomed. To speed up fathoming due to length infeasibility, before solving the subproblem, we first solve a tour length feasibility problem (Model 2.8) which checks whether current (partial) visit order yields a feasible length tour:

$$L_o^* = \min \sum_{(i,j)\in\bar{A}} \sum_{m=1}^{|M_o|} d_{ij}\bar{x}_{ijm} + \sum_{e_{ij}\in M_g} (d_{ij}z_{ij} + d_{ji}z_{ji}) \tag{2.8a}$$

$$\text{s.t.} \sum_{(i,j)\in\bar{A}} \sum_{m=1}^{|M_o|} d_{ij}\bar{x}_{ijm} + \sum_{e_{ij}\in M_g} (d_{ij}z_{ij} + d_{ji}z_{ji}) \leq L + \epsilon, \tag{2.8b}$$

$$(2.2b) - (2.2f), (2.2j), (2.2l), (2.2m).$$

Tour length feasibility model is a simplified version of Model 2.2 without subtour elimination constraints (2.2h) and (2.2i), finding the minimum length tour connecting the edges in M_g while ignoring locking considerations. If L_o^* is greater than $L + \epsilon$, that node

can be fathomed as the corresponding visit order cannot yield a feasible solution (see Figure 2.3).

When $|M_g| = \varphi$, we solve *LFATP-P(o′)* with constraints (2.7) to get an upper bound unless the subproblem is infeasible. In this case, the subproblem finds the best solution in the subtree emanating from the node with order o'. To obtain an early upper bound, the branching strategy is set to depth-first that favors promising partial visit orders: From a parent node, we select the child node with minimum $L^*_{o'}$: low $L^*_{o'}$ values have more length capacity to accommodate the additional must-visit edges ($M \setminus M_g$) further down the tree. Branching from such nodes increases the chance of finding feasible solutions sooner. Once a feasible solution (an upper bound) is obtained, the search strategy switches to breadth-first to fathom unpromising nodes at higher levels.

Subproblems with constraints (2.7) enable us to solve the DP formulation at various fixed depth levels of B&C, including the root node. As we go down in the tree of DP-based B&C approach, there is a trade-off between the size of formulation and the structure added to the subproblem. At deeper levels, subproblems are larger in size since the number of constraints and variables increases by g. On the other hand, the problems become easier to solve as there is more structure in terms of order.

In the experiments with DP-based B&C approach (see Appendix A.1), fixed depth levels of $\varphi = 2$ and $\varphi = 3$ are found to be the more effective compared to larger fixed depth levels. Note that the path-based reformulations constructed when $\varphi = 2$ and $\varphi = 3$, *LFATP-R2* and *LFATP-R3*, are special reformulations as M_2 and M_3 subsets yield a single partial visit order ($|O_{M_2}| = |O_{M_3}| = 1$). This eliminates the need for disjunctive visit order constraint (2.2k), resulting in a single MILP formulation to be solved without

any disjunctions. In other words, reformulations *LFATP-R2* and *LFATP-R3* are MILP formulations that benefit from length budget consumption at the cost of increased number of constraints and variables. We consider these two special reformulations for the rest of the chapter as they outperform their counterparts with larger fixed depth levels.

2.4. Valid Inequalities and Separation for the LFATP

In this section, we introduce valid inequalities for the LFATP. For simplicity, we use *LFATP-S* notation throughout the section; however, all inequalities are valid for path-based reformulations with appropriate transformations of x variables to $\bar{x}$ and z variables.

We first consider three trivial valid inequalities for LFATP. In a feasible LFATP solution, the degree of a vertex on the tour is two, making the following inequalities valid for all vertices in V:

$$\sum_{j:(i,j)\in A} x_{ij} + \sum_{l:(l,i)\in A} x_{li} \leq 2, \qquad i \in V. \tag{2.9}$$

Using linking constraints (2.1j) and basic locking arguments, specific subsets of x and y variables can be eliminated. The following inequalities are valid for arcs incoming to or outgoing from $q \in Q$:

$$x_{qj} \leq 0, \quad (q,j) \in A; \qquad x_{iq} \leq 0, \quad (i,q) \in A. \tag{2.10}$$

From the definition of the LFATP, it is clear that $q \in Q$ is locked when all $q-b$ paths intersect with the tour and this trivially holds when q itself is on the tour.

Let $V_M \subset V$ be the set of vertices where $i, j \in V_M$ if $e_{ij} \in M$. The following inequalities hold for lock feasibility flow path variables passing through $v \in V_M$:

$$y_{vj} \leq 0, \quad (v,j) \in A; \qquad y_{iv} \leq 0, \quad (i,v) \in A. \tag{2.11}$$

Since a feasible LFATP tour visits each edge in M, each vertex in V_M must be on the tour. From constraints (2.1j), incoming and outgoing arcs for vertices in V_M cannot carry lock feasibility flow. All trivial inequalities are added to the initial model at the root node of the branching tree and they are never separated during branching.

2.4.1. Subtour Elimination and Matching Inequalities

Subtour elimination and 2-matching inequalities from the OP literature are adapted to the LFATP. Since the adaptation of these inequalities is straightforward, we refer the reader to Fischetti et al. (1998) for details. We focus on the inequalities from OP literature rather than those from ARP literature as the restriction of a simple tour in the LFATP and OP have shown more promise.

2.4.1.1. Separation. Subtour elimination inequalities are identified with the parametric connectivity heuristic (Applegate et al., 2011) with a small modification. The heuristic is designed to identify subtour elimination inequalities in classical TSP; in the LFATP, subtour elimination inequalities are not valid for connected components that are large enough to contain a feasible LFATP solution. At an integer solution, it is trivial to identify whether a subtour is a feasible solution or not: If the subtour length is smaller than L, eliminate the subtour. For fractional solutions, a more conservative approach is taken. Given a connected component induced by vertex set, $S \subseteq V$, the length of the longest $|S|$ arcs within the connected component is denoted L_S. If $L_S < L$, this connected component cannot contain a feasible LFATP solution, and a subtour elimination inequality can be written for S. 2-matching inequalities are identified with the heuristic procedure introduced by Fischetti et al. (1998).

2.4.2. Vertex Degree Lock Elimination Inequalities

Vertex Degree Lock Elimination Inequalities (VDLEIs) are derived by identifying necessary conditions for a subset of vertices to be lock-free. VDLEIs limit the total number of arcs that can be used from $A(S)$ and $\delta(S)$ for a subset of vertices $S \subseteq V$ by using basic vertex degree arguments to maintain *free* vertices for lock feasibility flow paths. Let $\omega(S)$ represent a lower bound on the number of required free vertices in S, which can be obtained from the following free vertex minimization (FVM) problem for S:

$$\omega(S) \leq \min_{p_q \in \mathcal{P}_q, \forall q \in Q} \Big| \Big(\bigcup_{q \in Q} V(p_q) \Big) \cap S \Big|, \tag{2.12}$$

where $\mathcal{P}_q$ is the set of all feasible $q-b$ paths for vertex $q \in Q$ and $V(p_q)$ is the set of vertices in path $p_q \in \mathcal{P}_q$. Therefore, $\omega(S)$ is the minimum number of vertices from set S needed to construct $q-b$ paths. A tour using more than $|S| - \omega(S)$ vertices from set S is not feasible as the lock feasibility flow paths intersect with the vertices in the tour.

Lemma 2.3. *The following VDLEIs are valid for the LFATP:*

$$2 \sum_{(i,j) \in A(S)} x_{ij} + \sum_{(i,j) \in \delta(S)} x_{ij} \leq 2\Big(|S| - \omega(S)\Big), \qquad |S| \geq 2,\ S \subseteq V. \tag{2.13}$$

Proof of Lemma 2.3. Let $\beta(v)$ be the number of active arcs incoming to and emanating from vertex $v \in V$ in solution x. For a tour T corresponding to feasible solution x, $\beta(v) = 2$ if $v \in V(T)$ and $\beta(v) = 0$ otherwise. From the definition of $\omega(S)$, a feasible tour can have at most $|S| - \omega(S)$ vertices from set S. Therefore, the following must hold for any feasible solution:

$$\sum_{v \in S} \beta(v) \leq 2\Big(|S| - \omega(S)\Big). \tag{2.14}$$

Each arc in $A(S)$ contributes to vertex degrees of two vertices in S and each arc in $\delta(S)$ contributes to one vertex from S. Therefore, $\sum_{v \in S} \beta(v)$ can be written as follows:

$$\sum_{v \in S} \beta(v) = 2 \sum_{(i,j) \in A(S)} x_{ij} + \sum_{(i,j) \in \delta(S)} x_{ij}. \tag{2.15}$$

Combining (2.14) and (2.15), 2.13s are valid for the LFATP. □

The lock-free subset requirement is also used to derive two additional sets of valid inequalities. The first set strengthens subtour elimination inequalities for vertex subsets containing critical vertices and the second set imposes lock-free restrictions on arc subsets instead of vertex subsets. However, these additional set of inequalities are not found to be effective; thus, they are not discussed in detail.

2.4.2.1. Complexity of Separation. Identification of vertex degree lock elimination inequalities has two components: (1) Finding set S that potentially violates lock elimination inequalities and (2) determining the corresponding value of $\omega(S)$. Assuming that S is identified, $\omega(S)$ can be found by solving *FVM*. When there is no critical vertex in the underlying network; i.e. $|Q| = 0$, $\omega(S)$ is trivially 0. When $|Q| = 1$, $\omega(S)$ can be calculated in polynomial time with a slight modification of Dijkstra's algorithm as the underlying problems are vertex-weighted shortest path problems. For instances with $|Q| \geq 2$, finding $\omega(S)$ through *FVM* is NP-Hard as these problems become strongly related to the Steiner Tree Problem (STP). In an undirected graph $G(V, E)$, the STP finds a minimum cost tree which connects a subset of *terminal* vertices, $U \subseteq V$, by using the edges in E. For $|U| = 1$ the problem is trivial, for $|U| = 2$ it becomes the shortest path problem and for $U = V$ it is the minimum spanning tree problem, all of which can be solved in polynomial time. For subsets $3 \leq |U| < |V|$, the problem is NP-Hard (Karp, 1972). Consider a STP variant

where vertices in U and edges in E have no cost and vertices in $V \setminus U$ have unit cost. The Unit Vertex-Weighted Steiner Tree Problem ($UVWSTP$) finds a tree connecting vertices in U by using the minimum number of vertices from $V \setminus U$.

Theorem 2.4. *For a given vertex subset S, FVM is NP-Hard when $|Q| \geq 2$.*

Proof of Theorem 2.4. In order to prove Theorem 2.4 we first show that *UVWSTP* is NP-Hard, then transform *UVWSTP* to *FVM*.

Lemma 2.5. *UVWSTP is NP-Hard.*

Proof of Lemma 2.5. See the set covering transformation discussed by Klein and Ravi (1995). □

Let *FVMU* be the undirected version of *FVM* for undirected graphs. Consider a *UVWSTP* instance in undirected graph $G(V, E)$ for a subset U, $3 \leq |U| < |V|$. Let G^*_{ST} be the subgraph representing the optimal *UVWSTP* solution with the optimal value of f^*_{ST}. The *UVWSTP* instance can be transformed into an *FVMU* instance as follows: Let $u \in U$ be an arbitrary vertex chosen from U. Let $B = \{u\}$ be the set of border vertices and $Q = U \setminus \{u\}$ be the set of critical vertices for *FVMU*. Solve *FVMU* for $S = V \setminus (Q \cup B) = V \setminus U$. With this transformation, *FVMU* constructs $q - u$ paths for $q \in Q$ with the minimum number of vertices from S. Let G^*_Ω be the subgraph formed by the union of $q - u$ paths used in the optimal *FVMU* solution with the optimal value of $\omega(S)$.

We first show that $f^*_{ST} = \omega(S)$. Note that both f^*_{ST} and $\omega(S)$ represent the number of vertices from S in subgraphs G^*_{ST} and G^*_Ω, respectively. Assume that $\omega(S) > f^*_{ST}$.

Since G^*_{ST} is a connected graph, one can find paths connecting each pair of vertices in G^*_{ST} including the vertices in $U = Q \cup B$. Therefore, G^*_{ST} contains a feasible solution to $FVMU$ with at most f^*_{ST} vertices from S, contradicting the optimality of G^*_Ω. Therefore, $\omega(S) \leq f^*_{ST}$. Now, assume that $\omega(S) < f^*_{ST}$. Since G^*_Ω is the union of paths connecting each $q \in Q$ to the single border vertex $u \in B$, G^*_Ω is a connected graph which contains vertices in $Q \cup \{u\} = U$; therefore, G^*_Ω contains a Steiner tree solution with at most $\omega(S)$ vertices from S, contradicting the optimality of G^*_{ST}. Therefore, $\omega(S) \geq f^*_{ST}$. Since both $\omega(S) \leq f^*_{ST}$ and $\omega(S) \geq f^*_{ST}$ hold, $\omega(S) = f^*_{ST}$.

$\omega(S) = f^*_{ST}$ indicates that both $FVMU$ and $UVWSTP$ use the same number of vertices. Furthermore, a solution to $FVMU$ can be converted to a $UVWSTP$ solution by using a polynomial time spanning tree algorithm on G^*_Ω. Thus, $FVMU$ is NP-Hard. Since $FVMU$ is NP-Hard, the directed version, FVM, is NP-Hard. □

2.4.2.2. Heuristic Separation Algorithm. The heuristic separation algorithm is only used at fractional solutions, as integer feasible solutions at nodes cannot contain violated VDLEIs (see constraints (2.1j)). The heuristic separation algorithm identifies sets S_q (if any) that lock q, for each critical vertex $q \in Q$. The algorithm simultaneously calculates lower bounds for the optimal value of $\omega(S_q)$, denoted $\underline{\omega}(S_q)$. Let V^* be the set of active vertices at a fractional solution and $\delta^{out}(W)$ be the set of outgoing arcs from vertex set $W \subseteq V$. Algorithm 2.1 iteratively attempts to push flow from critical vertex q to available adjacent vertices until the flow reaches a border vertex or identifies a lock. Let $\mathcal{V}_{con}$ be the set of already considered vertices, $\mathcal{V}_{av}$ be the set of available vertices at the beginning of a loop and $\mathcal{V}_{new}$ be the set of new available vertices identified in the loop.

Algorithm 2.1 *Heuristic Separation Algorithm for Fractional Solutions.*

$S_q \leftarrow \emptyset$, $\underline{\omega}(S_q) \leftarrow 0$, $\mathcal{V}_{con} \leftarrow \{q\}$, $\mathcal{V}_{new} \leftarrow \{q\}$
while $\mathcal{V}_{new} \neq \emptyset$ **do**
  $\mathcal{V}_{av} \leftarrow \mathcal{V}_{new}$, $\mathcal{V}_{new} \leftarrow \emptyset$
  for each arc $(i, j) \in \delta^{out}(\mathcal{V}_{av})$ **do**
    if $j \in \mathcal{V}_{con}$ **then**
      SKIP.
    else if $j \in B$ and $j \notin V^*$ **then**
      STOP. q has a feasible connection.
      return $S_q \leftarrow \emptyset$, $\underline{\omega}(S_q) \leftarrow 0$
    else
      $\mathcal{V}_{con} \leftarrow \mathcal{V}_{con} \cup \{j\}$
      if $j \in V^*$ **then**
        $S_q \leftarrow S_q \cup \{j\}$
      else
        $\mathcal{V}_{new} \leftarrow \mathcal{V}_{new} \cup \{j\}$
      end if
    end if
  end for
end while
return S_q, $\underline{\omega}(S_q) \leftarrow 1$

In Algorithm 2.1, vertices adjacent to critical vertex q are searched to identify new available vertices that can be used to form lock feasibility flow paths in the current fractional solution. A vertex v is available if $v \notin V^*$ and unavailable otherwise. The exploration procedure is repeated for newly identified available vertices until one of two stopping conditions is satisfied: (1) the algorithm identifies an available vertex $v \in B$ such that a feasible $q - b$ path exists or (2) the algorithm cannot identify any new available vertices indicating that set S_q locks critical vertex q. When S_q locks critical vertex q, at least one vertex in S_q must be inactive in a feasible solution; therefore, the algorithm reports $\underline{\omega}(S_q) = 1$ as a valid lower bound for $\omega(S_q)$. If the algorithm terminates in the second condition, the following valid inequality is added if violated by the fractional solution:

$$2 \sum_{(i,j)\in A(S_q)} x_{ij} + \sum_{(i,j)\in \delta(S_q)} x_{ij} \leq 2(|S_q| - \underline{\omega}(S_q)). \tag{2.16}$$

Proposition 2.6. *Algorithm 2.1 is polynomial with $\mathcal{O}(|A||V|)$ time.*

Proof of Proposition 2.6. In the worst case, the while loop is repeated for each vertex $v \in V$ and the algorithm treats each arc $(i, j) \in A$, thus the inner for loop is repeated $\mathcal{O}(|A|)$ times. The operations within the for loop require searches over vertex sets which are subsets of V, which can be completed in $\mathcal{O}(|V|)$ time, thus the algorithm takes $\mathcal{O}(|A||V|)$ time for each critical vertex. □

Corollary 2.7. *Since Algorithm 2.1 is repeated for each critical vertex $q \in Q$, the separation procedure is completed in $\mathcal{O}(|Q||A||V|)$ time.*

2.5. Numerical Experiments

We conduct three sets of numerical experiments. The first set compares formulations *LFATP-S*, *LFATP-R2* and *LFATP-R3* on randomly generated LFATP instances along with a discussion of the effects of VDLEIs. The second set compares these three formulations under more generalized settings. The last set derives practical insights on course design based on BACM, using a more detailed network. The experiments are conducted on a computer with a 2.50 GHz CPU and 8 GB memory running under 64-bit Windows 10 operating system. MILP models are solved by using Gurobi 5.6 (Gurobi Optimization, 2013) in C# environment of Microsoft Visual Studio 2010. In all experiments, formulations are given a 2-hour time limit.

2.5.1. Formulation Comparisons on the LFATP Instances

In this section, we compare *LFATP-S* and reformulations on LFATP instances and discuss the effectiveness of VDLEIs. We choose average medical distance minimization as our arc-additive objective function for two reasons. First, with this objective, the LFATP finds a tour whose arcs are closest to medical facilities (critical vertices) in the network while satisfying visit, locking and length budget requirements which is critical in marathon setting as runners seeking medical treatment may be transported to medical facilities. Second, average medical distance is a challenging arc-additive objective function from optimization perspective due to its clustered cost structure in the network and its trade-off with locking restriction, which will be discussed shortly.

The experiments are carried out on two sets of 12×20 grid networks which consist of 240 vertices and 896 arcs. The first set, *M12Q4*, contains $|M| = 12$ must-visit edges and $|Q| = 4$ critical vertices, and the second, *M24Q4* contains $|M| = 24$ must-visit edges and $|Q| = 4$ critical vertices. Each arc has a unit length. The locations of edges in M and vertices in Q are assigned randomly. Since having critical vertices near the center of must-visit edges makes instances notably difficult or infeasible, endpoints of must-visit edges are chosen at least 2 units away from the rectangular perimeter of grid networks and critical vertices are chosen within 3 units from the perimeter. Instances with varying difficulties are introduced when testing VDLEIs (see Section 2.5.1.2). The vertices on perimeters of grid networks are considered border vertices. The objective contribution for arc (i, j), c_{ij}, is calculated as follows:

$$c_{ij} = \frac{m_{ij} d_{ij}}{L} \qquad (i, j) \in A, \tag{2.17}$$

where m_{ij} is the rectilinear distance from the midpoint of arc (i, j) to the closest medical facility (critical vertex). c_{ij} represents the length-weighted contribution of arc (i, j) to the average medical distance. In this case, arcs near critical vertices are more preferable than those that are far away, resulting in clusters of preferable arcs around critical vertices, encouraging subtour formation near these vertices. Furthermore, the tour tries to pass as close as possible by the critical vertices without locking them which creates a conflict between the objective and the locking restriction. As the length budget increases, the problem is more challenging as there is more room for subtour and lock formations. We consider three tour length restrictions where the length of a feasible tour must be within $[95\%, 110\%]$, $[110\%, 125\%]$ and $[125\%, 140\%]$ of the minimum tour length (calculated by the tour length minimization problem, see Section 2.5.2.1). Restriction $[95\%, 110\%]$ is effectively $[100\%, 110\%]$ since the length of a feasible tour cannot be less than minimum tour length; however, 95% is provided as a weak lower bound on the length budget.

2.5.1.1. Reformulation Analysis. Table 2.1 provides a summary of the results. The first five columns report the average values over 10 instances of the corresponding network. These columns include counts on subtour elimination inequalities (*Sbtr*) and VDLEIs, time spent (in seconds) generating the cuts (*CutGen*) and solving the formulation (*Total*), and the optimality gap (*Gap*). The last three columns provide a breakdown of 10 instances where *Opt*, *Feas* and *NoSn* denote the number of instances for which an optimal solution is found (*Opt*), a feasible solution is found but not proven to be optimal (*Feas*) and no solution is found (*NoSn*), respectively. Table 2.1 does not include 2-matching inequality counts as these inequalities rarely identified in our experiments.

Table 2.1. Results for LFATP instances.

Setting	Formulation	Count		Duration (s)		Performance			
		Sbtr	*VDLEI*	*CutGen*	*Total*	*Gap (%)*	*Opt*	*Feas*	*NoSn*
M12Q4 [95% - 110%]	*LFATP-R2*	23	0	4	9	0.0	10	0	0
	LFATP-R3	27	1	4	8	0.0	10	0	0
	LFATP-S	3369	39	555	1727	0.5	9	1	0
M12Q4 [110% - 125%]	*LFATP-R2*	105	8	16	25	0.0	10	0	0
	LFATP-R3	107	11	18	26	0.0	10	0	0
	LFATP-S	3209	70	360	1349	0.5	9	1	0
M12Q4 [125% - 140%]	*LFATP-R2*	411	67	63	140	0.0	10	0	0
	LFATP-R3	496	58	69	227	0.0	10	0	0
	LFATP-S	4632	179	684	2459	1.0	7	3	0
M24Q4 [95% - 110%]	*LFATP-R2*	294	6	51	153	0.0	10	0	0
	LFATP-R3	214	5	32	147	0.0	10	0	0
	LFATP-S	2125	20	313	955	0.5	9	1	0
M24Q4 [110% - 125%]	*LFATP-R2*	1053	36	223	958	0.0	10	0	0
	LFATP-R3	1108	37	221	1831	0.1	8	2	0
	LFATP-S	3460	118	506	1591	0.7	9	1	0
M24Q4 [125% - 140%]	*LFATP-R2*	3100	274	826	5192	0.6	4	6	0
	LFATP-R3	2123	182	428	4198	0.4	5	5	0
	LFATP-S	7598	400	1203	5474	1.1	5	5	0

In network *M12Q4* experiments, reformulations are significantly faster, and require fewer subtour elimination inequalities and VDLEIs compared to *LFATP-S*. Each reformulation solves all instances to optimality whereas only 83% are solved optimally by *LFATP-S*. In network *M24Q4* experiments, reformulations outperform *LFATP-S* when length budget is $[95\%, 110\%]$. For higher length budgets, reformulations still yield lower optimality gaps and fewer valid inequalities than *LFATP-S*; however, solution times are closer as more instances reach the time limit of 2 hours. The number of instances solved to optimality by *LFATP-R2*, *LFATP-R3* and *LFATP-S* are 24, 23 and 23 out of 30, respectively and approximately 10% of VDLEIs are binding at these solutions. The results of *M24Q4* indicate that reformulations start to lose their effectiveness when the length

budget is high. This is an expected result as the portion consumed by reformulations decreases with increasing length budget.

2.5.1.2. Effects of using VDLEIs. In Section 2.5.1.1, formulations are tested on randomly generated LFATP instances where the locations of critical vertices are within 3 units from the perimeter. In this section, we vary restrictions on the locations of critical vertices to test the effectiveness of VDLEIs. We generate five sets of instances as follows: For *Set-*γ the critical vertices must be within γ units of the perimeter where $\gamma \in \{1, 2, 3, 4, 5\}$. As γ increases, instances become more difficult to solve as critical facilities are allowed to be closer to the center. Each set contains 10 instances with 12 must-visit edges and 4 critical vertices.

Table 2.2. Solution time averages (in seconds) for VDLEI experiments.

Reformulation	**Budget**	**VDLEI**	**Set-1**	**Set-2**	**Set-3**	**Set-4**	**Set-5**
LFATP-R2	[-5% - 10%]	*N*	**11**	**12**	24	**30**	**756**
		Y	13	13	**14**	39	776
	[10% - 25%]	*N*	75	**27**	46	**123**	805
		Y	**57**	39	**44**	126	**791**
	[25% - 40%]	*N*	**191**	805	**188**	683	1672
		Y	365	**586**	299	**568**	**1563**
	Average	*N*	**93**	281	**86**	279	1078
		Y	145	**213**	119	**244**	**1043**
LFATP-R3	[-5% - 10%]	*N*	**6**	**11**	**6**	**48**	**744**
		Y	**6**	13	7	66	754
	[10% - 25%]	*N*	**23**	**46**	**28**	207	**658**
		Y	95	66	37	**108**	775
	[25% - 40%]	*N*	204	1151	1026	1035	1452
		Y	**183**	**355**	**434**	**846**	**1236**
	Average	*N*	**77**	403	353	430	952
		Y	95	**145**	**159**	**340**	**922**

We use reformulations to test the effectiveness of VDLEIs as *LFATP-S* is found to be underperforming in Section 2.5.1.1. Table 2.2 provides average solution times (in seconds) for each set of instances and length-budget pair. As expected, solution times tend to increase from *Set-1* to *Set-5* and from low length budget to high. VDLEI column specifies whether these inequalities are used (Y) or not (N). Between N and Y rows of the same setting, better solution times are written in bold. *Average* multirows provide reformulation averages of all instances solved with and without VDLEIs for each set. As seen in Table 2.2, it is not trivial to claim that using VDLEIs improves or deteriorates solution procedure. Looking at the solution times in detail, when we use VDLEIs we either have a significant improvement in the solution time or a slight deterioration. In these experiments, there are 26 instances with a solution time of more than 20 minutes. Among these 26 instances, solution times of 13 instances can be improved more than 25% by using VDLEIs whereas the solution times of only 3 instances can be improved more than 25% without using VDLEIs. For the remaining 10 instances the difference between using and not using VDLEIs is not significant.

In summary, the experiments in this section indicate that VDLEIs are effective in many instances, especially in more challenging ones. Even though using VDLEIs increases solution times of some instances slightly, benefits outweigh such drawbacks.

2.5.2. Formulation Comparisons on Other ARPs

This section provides numerical experiments that compare formulations *LFATP-S*, *LFATP-R2* and *LFATP-R3* on two additional problems. The first problem is a tour length minimization problem where the length budget constraints are removed. The aim is to find

a minimum length tour which satisfies visit and locking requirements. In this setting, we compare formulations when the lower and upper bounds of length budget are not explicitly stated and come implicitly from the objective function. The second problem is a random coefficient minimization problem where the additive coefficients of the objective function are assigned randomly and the aim is to find a minimum cost tour which satisfies visit and length requirements while ignoring locking. In this setting, we compare formulations when solving a generic tour finding problem with respect to various length budgets. These two problems are less structured compared to the LFATP and capture a majority of arc routing problems with budget and visit requirements, potentially generalizing the results to those problems.

For these runs, we introduce two additional sets of 12×20 grid networks with no critical vertices, *M12Q0* and *M24Q0*, which consist of $|M| = 12$ and $|M| = 24$ must-visit edges, respectively.

2.5.2.1. Tour Length Minimization Problem. In this problem, we remove constraints (2.1c) and (2.2g) from *LFATP-S* and reformulations, respectively. Each arc has a unit length and the aim is to find the minimum length tour which visits all edges in M. All networks are used in these experiments and Table 2.3 summarizes the results.

In network *M12Q0*, *M24Q0* and *M12Q4* experiments, reformulations significantly outperform *LFATP-S* both in solution times and in the number of subtour elimination inequalities needed to converge. *M12Q4* and *M24Q4* are relatively easier to solve in this problem setting since must-visit edges are clustered around the center and the objective is to minimize the tour length. *LFATP-S* becomes competitive against reformulations in the instances of *M24Q4*; however, when all instances are considered, both reformulations

Table 2.3. Results for tour length minimization problem.

Setting	Formulation	Count		Duration (s)		Performance			
		Sbtr	*VDLEI*	*CutGen*	*Total*	*Gap (%)*	*Opt*	*Feas*	*NoSn*
M12Q0	*LFATP-R2*	369	-	23	40	0.0	10	0	0
	LFATP-R3	349	-	20	59	0.0	10	0	0
	LFATP-S	10951	-	1480	4949	9.4	4	6	0
M24Q0	*LFATP-R2*	609	-	68	227	0.0	10	0	0
	LFATP-R3	833	-	57	432	0.0	10	0	0
	LFATP-S	8813	-	988	4284	3.3	5	5	0
M12Q4	*LFATP-R2*	59	1	5	9	0.0	10	0	0
	LFATP-R3	100	2	11	21	0.0	10	0	0
	LFATP-S	2037	21	289	905	0.4	9	1	0
M24Q4	*LFATP-R2*	509	23	53	162	0.0	10	0	0
	LFATP-R3	399	14	39	223	0.0	10	0	0
	LFATP-S	1049	24	72	166	0.0	10	0	0

converge to optimal solutions in 40 out of 40 instances whereas this number reduces to 28 out of 40 for *LFATP-S* with varying optimality gaps. Between the reformulations, *LFATP-R2* performs slightly better than *LFATP-R3*, implying that using two farthest edges is sufficient in consuming the length budget. These results suggest that reformulations are extremely effective in reducing the solution efforts even in the absence of explicit length budget constraints.

2.5.2.2. Random Coefficient Minimization Problem. In random coefficient objective, each arc has a random objective contribution between 0 and 1. The aim is to minimize the random objective contribution while visiting all edges in M and satisfying the tour length restrictions. In this problem, we ignore locking restrictions to test the effect of budget constraints on a more generic routing setting; therefore, the experiments are carried out on networks without critical vertices, *M12Q0* and *M24Q0*. The same tour length restrictions from Section 2.5.1 are used. Table 2.4 summarizes the results. Since

locking restrictions are not considered, VDLEIs are not used and Table 2.4 does not have the corresponding *VDLEI* column.

Table 2.4. Results for random coefficient minimization problem.

Setting	**Formulation**	**Count**	**Duration (s)**		**Performance**			
		Sbtr	*CutGen*	*Total*	*Gap (%)*	*Opt*	*Feas*	*NoSn*
M12Q0 [95% - 110%]	*LFATP-R2*	123	14	35	0.0	10	0	0
	LFATP-R3	162	33	83	0.0	10	0	0
	LFATP-S	8593	1131	3633	1.7	5	3	2
M12Q0 [110% - 125%]	*LFATP-R2*	198	18	60	0.0	10	0	0
	LFATP-R3	153	14	47	0.0	10	0	0
	LFATP-S	5695	835	2644	1.2	7	2	1
M12Q0 [125% - 140%]	*LFATP-R2*	224	18	59	0.0	10	0	0
	LFATP-R3	250	21	120	0.0	10	0	0
	LFATP-S	3771	512	1889	1.1	8	2	0
M24Q0 [95% - 110%]	*LFATP-R2*	193	35	119	0.0	10	0	0
	LFATP-R3	305	70	501	0.0	10	0	0
	LFATP-S	3293	901	2407	1.2	7	3	0
M24Q0 [110% - 125%]	*LFATP-R2*	364	64	261	0.0	10	0	0
	LFATP-R3	662	97	730	0.0	10	0	0
	LFATP-S	3068	723	2167	0.7	8	2	0
M24Q0 [125% - 140%]	*LFATP-R2*	313	49	232	0.0	10	0	0
	LFATP-R3	580	73	552	0.0	10	0	0
	LFATP-S	3156	544	1816	0.5	8	2	0

Unsolved cases are not considered when calculating the average gaps.

Similar to the results in the tour length minimization problem, reformulations significantly outperform *LFATP-S* in the number of subtour elimination inequalities needed and in solution times. Both reformulations converge to optimal solutions in 60 out of 60 instances, whereas this number reduces to 43 out of 60 for *LFATP-S*. Furthermore, *LFATP-S* fails to find a feasible solution within 2-hour time limit in 3 of the remaining 17 instances. Between reformulations, *LFATP-R2* performs better than *LFATP-R3*. As

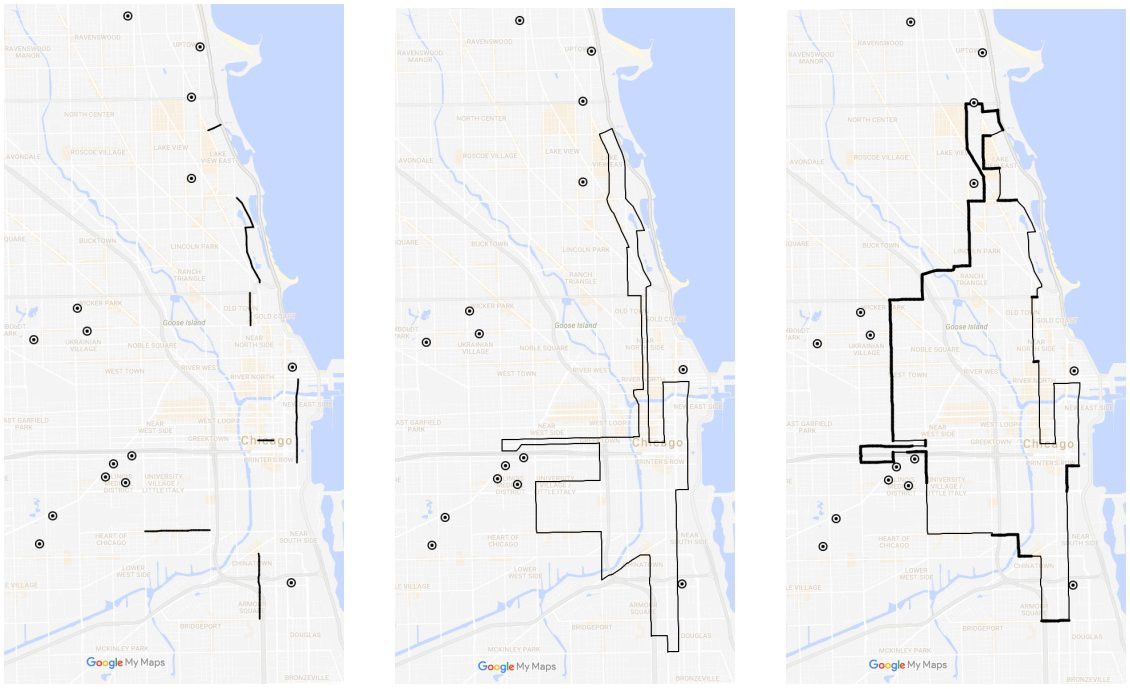
the length budget increases, the difference between solution times of *LFATP-S* and reformulations decreases; however, reformulations are still significantly better compared to *LFATP-S*.

In summary, reformulations effectively reduce solution times and subtour formations when compared to *LFATP-S* under all problem settings tested here which is promising as these settings cover a broad range of arc routing variants with length budget and locking restrictions.

2.5.3. Case Study - BACM

This section provides details regarding the application of the LFATP to BACM with the aim of minimizing average medical distance. The underlying network is generated within City of Chicago street centerline map to include all streets whose width is greater than 30 feet (City of Chicago, 2013). The network is reduced in size by eliminating dead-ends and merging 'two-neighbor' intersections. If a vertex j has only two incident vertices i and k, then arcs (i, j) and (j, k) can be merged to arc (i, k) with $c_{ik} = c_{ij} + c_{jk}$ and $d_{ik} = d_{ij} + d_{jk}$. The resulting graph consists of 435 vertices and 1492 arcs. The edges in M are specified by the race organizers of BACM. The organizers identify 7 such street segments. Some segments correspond to a sequence of multiple edges in the underlying network, in which case, the edges are merged into a single edge. The network includes 17 critical vertices which represent the hospitals in the region. The vertices on the perimeter of the network are considered border vertices. Figure 2.4a shows the locations of the identified street segments in M and the hospitals in the network (Google Maps, 2015). Two hospitals are outside the figure boundaries. The objective contribution of each arc is calculated with

equation (2.17). L is set to 26.22 miles plus an additional distance for after race events and ϵ is set to 2000 feet.



(a) Set M and hospitals. (b) Current BACM course. (c) The best LFATP result.

Figure 2.4. Locations of must-visit edges and hospitals, original course and the best LFATP solution.

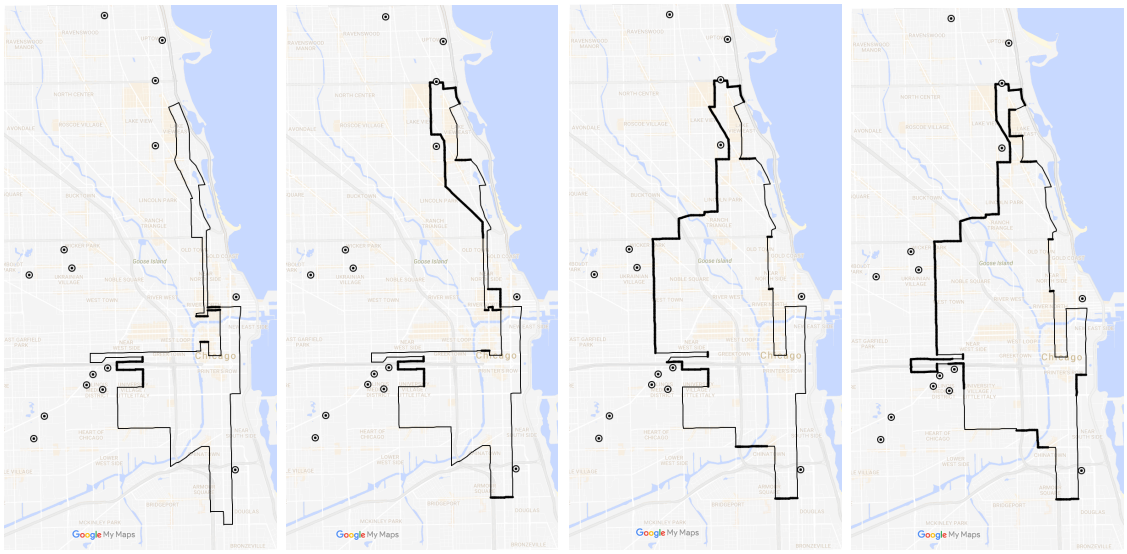
The LFATP is solved by *LFATP-R2* formulation. Using VDLEIs we obtain a solution within 1.31% optimality within 2-hour time limit. The original course and the best LFATP solution are shown in Figure 2.4b and 2.4c, respectively. Thin lines in Figure 2.4c represent the street segments common to both courses; thicker lines show the deviations of the best course from the original. The average distance from a point on the course to the closest medical facility is 5927.5 feet for the original course and the best solution found by LFATP reduces this distance to 4852.7 feet. This translates to a travel distance reduction of two city blocks in the city of Chicago (Chicago uses double block size; therefore, the reduction

is approximately four 'typical' city blocks). According to BACM medical director, Dr. George Chiampas, 'While we know on scene management is critical, we also know during the Boston Marathon bombing that rapid transports to local trauma centers led to lives being saved'. Therefore, such a reduction is critical for course design. Even though this reduction has significant effects, the resulting tour deviates from the original one, bringing noticeable operational challenges at the application level. The question then arises if we can get significant reductions by changing the course in a minimal way. For this reason, the LFATP is applied to BACM with a similarity constraint. This constraint ensures that at least a specified percent of the resulting course is the same as the current course. Specifically, $K\%$ similarity is defined as follows: The total length of the common arcs must be at least $K\%$ of the original course length. The experiments are carried out under $0, 10, 20, ..., 90\%$ similarity levels. The resulting problems are again solved by *LFATP-R2* with a time limit of 2 hours.

Table 2.5. Results with different similarity requirements.

Similarity (%)	**Duration** (s)	**Gap** (%)	**Best** (feet)	**LB** (feet)	**Improvement** (%)
100	2.5	0.0	5927.5	5927.5	0.0
90	714.3	0.0	5620.3	5620.3	5.2
80	531.3	0.0	5295.1	5295.1	10.7
70	TILIM	1.1	5140.8	5085.0	13.3
60	TILIM	0.9	4963.8	4920.3	16.3
50	TILIM	1.0	4855.2	4805.5	18.1
40	TILIM	1.2	4852.7	4792.2	18.1
30	TILIM	1.3	4852.7	4791.5	18.1
20	TILIM	1.4	4852.7	4786.2	18.1
10	TILIM	1.4	4852.7	4785.1	18.1
0	TILIM	1.3	4852.7	4789.8	18.1

Table 2.5 summarizes time spent by the solution approach along with the gap, the best objective function value and the lower bound before termination for each similarity level.



(a) 90% similarity 5.2% decrease. (b) 70% similarity 13.3% decrease. (c) 60% similarity 16.3% decrease. (d) $0-40\%$ similarity 18.1% decrease.

Figure 2.5. Tours with different similarity requirements.

The last column shows the improvement over the current course. Figure 2.5 provides plots of the best courses on Google Maps (2015) for similarity levels 90%, 70%, 60% and $0-40\%$.

The average distance from a point in the current course to the nearest medical facility is 5927.5 feet. As expected, objective function values decrease as the similarity requirement is relaxed. The best objective value of 4852.7 is observed when the similarity requirement is below 40%, indicating that the average distance to the nearest medical facility can be decreased by 18.1%. Considering the lowest lower bound among the similarity levels that are not solved to optimality, the maximum improvement that we can achieve is only slightly higher at 19.3%. The largest improvements are obtained around $90-70\%$ similarity levels, indicating that changing $10-30\%$ of the course is sufficient

to obtain a large portion of the potential improvement. For instance, the objective function value decreases by 13.3% with 70% similarity. As seen in Figure 2.5, the courses at similarity levels $90 - 70\%$ only increase the number of arcs that are used near a medical district in the southwest part and expand to north in order to get closer to the hospitals. On the other hand, the course shape changes dramatically around 60% by expanding towards west to reach the hospitals in that region. Being able to obtain significant benefits without changing the streets and overall structure significantly is an important result from practical perspective.

2.5.4. Summary

The results of the experiments can be summarized from two different perspectives. From methodological side, standard approaches may fail to solve arc routing problems when there is a lower bound on tour length or when the cost structure encourages subtour formation. In such cases, path-based reformulations perform significantly better since they consume a large portion of length budget and reduce the room for subtour formation. These reformulations make use of visit requirements and length/budget restrictions and, therefore, offer promise to many ARPs with similar requirements as suggested by our numerical experiments. Furthermore, the vertex degree lock elimination inequalities lead to improvements in solution times. From an application side, the LFATP can be used to design new courses and to improve existing courses while satisfying similarity levels. Our results show that significant improvements can be achieved even with relatively small changes.

2.6. Final Remarks

As this study represents the first step towards a comprehensive modeling framework for course design, there are several lines of future research. The LFATP is valid for any arc-additive objective function such as average width of the course and number/intensity of surveillance cameras along the course. However, the quality of a course is a composite of multiple factors; future work focuses on generalizing the LFATP and solution approaches to incorporate objective functions that are not arc-additive. Many objective functions can be classified as either time- or sequence-dependent. An objective function is time-dependent if the objective coefficients of the decision variables vary over time. For example, decreasing medical distance at later stages of the race can be more important compared to early stages as the medical incident density increases over time (Kim et al., 2012). Similarly, an objective function is sequence-dependent if the objective contribution of each decision variable depends on whether another set of variables is used in the solution. Examples of sequence-dependent objective functions can be number of turns and incline change along the course. These objectives are more complex than arc-additive counterparts as time tracking and sequence identification bring an additional level of difficulty in terms of modeling and solving the problem. Extensions to these more complex objectives are discussed in Chapter 4.

Even though 2-edge and 3-edge subsets of M reduce solution efforts significantly for arc-additive objective functions, there may be instances which require more detailed visit orders with larger number of edges which can be handled by the DP-based B&C approach. Currently, the DP-based B&C approach cannot identify a feasible solution until it reaches the deepest branching level; therefore, new tour finding heuristics which are capable of

finding tours respecting a given visit order is another promising future direction to improve the quality of solutions and solution times of the DP-based B&C approach.

CHAPTER 3

An Interactive Method for Multiobjective Optimization using Weight Diversification

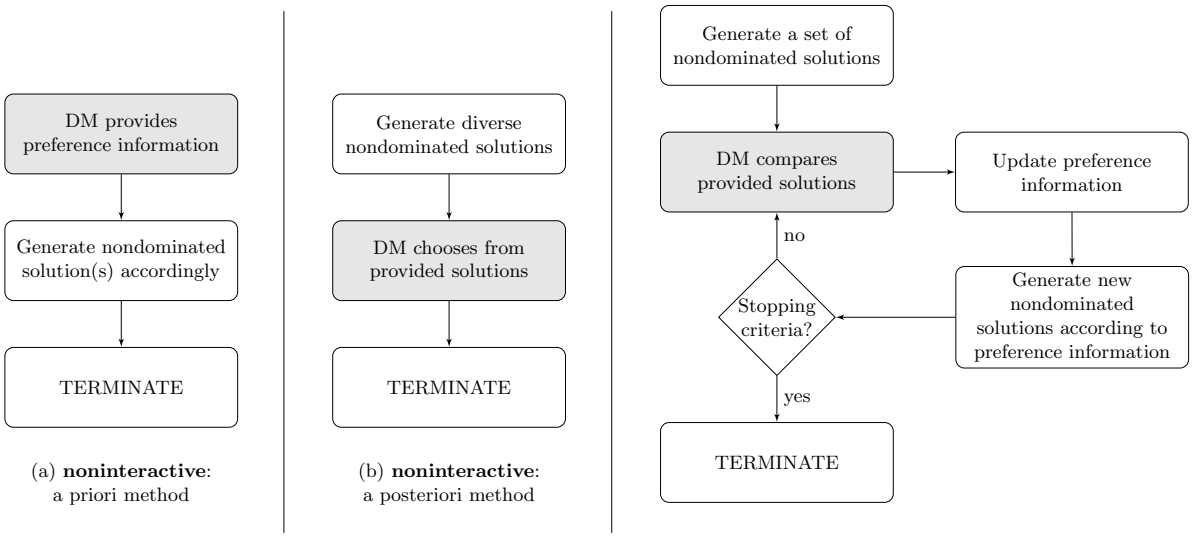
As discussed in Chapter 1 and 2, marathon course design involves multiple objectives as the course is a composite of multiple factors related to safety, health and experience. Standard approaches for multiobjective problems require decision makers (DMs) to assess the importance of each objective to find a compromise solution. In course design, a correct assessment of objectives can be extremely difficult since the underlying objectives significantly differ from standard tour finding objectives that minimize travel costs or maximize collected profits.

Similar to marathon course design, many optimization problems involve multiple objectives which may lead to conflicting solution recommendations with no unique solution which optimizes all objectives simultaneously. Consequently, it is only possible to identify a set of *nondominated (efficient, Pareto)* solutions (Miettinen, 2008) and the DM chooses a *most preferred* solution among the nondominated solutions. A straightforward approach to achieve this goal is to provide all nondominated solutions to the DM and let the DM choose. This approach is often not possible for two reasons: (i) generating all nondominated solutions is generally impractical and (ii) DMs prefer to evaluate a small number of solutions instead of the entire nondominated solution set. This chapter introduces an interactive weight region-based approach (IWRA) to address these challenges.

The algorithms proposed throughout the chapter are not specific to marathon course design as they are developed for general multiobjective linear programs (MOLPs) and multiobjective integer programs (MOIPs).

The weighted sum problem (WSP) is a commonly used problem in multiobjective optimization. It assigns a weight to each objective, depending on its importance, and optimizes the resulting weighted sum objective (Chankong and Haimes, 1983). WSP requires the optimal weight information of the DM. Many early studies focus on rating and weighting methods to elicit the DM's weight vector. Examples include the multi-attribute rating technique (Edwards, 1977), the point allocation approach (Edwards and von Winterfeldt, 1986), swing weighting (Edwards and von Winterfeldt, 1986), trade-off weighting (Keeney and Raiffa, 1993) and the analytic hierarchy process (Saaty, 1990). In many real-life applications, DMs face difficult-to-quantify objective trade-offs. In such cases, the weights provided by the DMs may not represent true priorities. Interactive approaches circumvent this problem.

Exisiting work in multiobjective optimization methods is divided into two categories based on the role of the DM during the solution process: noninteractive methods and interactive methods, see Figure 3.1. In noninteractive methods, the DM either does not take part in the solution process (*no-preference methods*) or provides preference information before (*a priori methods*) or after (*a posteriori methods*) the solution process (Miettinen, 2008). While the DM is able to provide her preference information to some extent in a priori and a posteriori methods, these methods do not allow preferential updates *during* the solution process. These methods require a global understanding of the specifications and trade-offs for the underlying problem, which can be challenging.



(a) **noninteractive**: a priori method

(b) **noninteractive**: a posteriori method

(c) **interactive** method

Figure 3.1. Schematic representation of noninteractive (a) & (b), and interactive (c) methods for multiobjective optimization problems.

Interactive methods iterate between a comparison phase and a solution generation phase to collect and refine preference information (Miettinen et al., 2008). In the literature, comparison phase and solution generation phase are also referred to as decision phase and optimization stage, respectively. In the comparison phase, the DM provides local preference information by answering questions regarding provided nondominated solutions. In the solution generation phase, new solutions are generated satisfying existing preference information. This process is repeated until a satisfactory solution is found or the DM chooses to terminate the process (see Figure 3.1c). The DM controls the process, guiding towards the most preferred solutions with local preference information. A successful interactive approach should generate solutions that enable the DM to clearly perceive the trade-offs among solutions. An ideal approach should also limit the number of interactions with the DM.

This study introduces an Interactive Weight Region-Based Approach (IWRA) to find a most preferred solution of a DM for MOLPs and MOIPs. At each iteration of IWRA, the DM is provided a new nondominated solution. The DM either compares this new solution with the current most preferred solution or inserts it in a sorted list of existing solutions based on individual preferences. This comparison/insertion step defines a set of constraints on the DM's optimal weight region. When generating the new nondominated solution for the decision maker, IWRA first finds *a most diverse* weight vector in the remaining weight region via a mixed integer linear programming (MILP), then solves a WSP for this vector to obtain a new nondominated solution.

The contributions of this study are threefold. First, different from existing approaches, we propose a weight diversification model which is capable of exploring the entire weight region. IWRA uses this model to identify the next weight vector and the corresponding nondominated solution. Obtained this way, the solutions are potentially diverse, making comparisons easier for the DM. Second, we provide two finitely converging algorithms based on WSP, IWRA-MOLP and IWRA-MOIP, for MOLPs and MOIPs respectively. Both algorithms terminate after finding a most preferred solution. With simulated experiments, we show that the algorithms find a most preferred solution with a reasonably small number of comparisons. For example, typically the DM's most preferred solution can be found after exploring just 35-45 solutions in 10-objective linear programming instances with more than a half million nondominated solutions and 20-30 solutions in 5-objective integer programming instances with more than thirty thousand nondominated solutions. We empirically show that IWRA-MOLP is computationally more efficient than the Zionts-Wallenius (ZW) method (Zionts and Wallenius, 1976) in terms of the required number of

solutions and comparisons to converge. Similar results are shown for IWRA-MOIP when compared with the cone-based method of Lokman et al. (2016) for the tested problems that use linear weighted sum. Furthermore, IWRA provides reliable results when the DM makes imperfect comparisons yielding incorrect preference information. One issue with WSP-based approaches is WSP's inability to generate unsupported nondominated solutions in nonconvex problems such as integer programs. Therefore, we extend IWRA to weighted Tchebycheff problem (WTP) setting and introduce IWRA-WTP to handle instances with unsupported nondominated solutions.

The chapter is organized as follows: Section 3.1 summarizes related work on interactive approaches with a focus on elicitation of the DM's preference information and Section 3.2 provides problem setting details. Section 3.3 introduces IWRA, and Sections 3.4 and 3.5 introduce IWRA-MOLP and IWRA-MOIP algorithms, respectively. Section 3.6 presents a detailed computational study. Section 3.7 extends IWRA to WTP setting. Section 3.8 concludes the study.

3.1. Literature Review

Toubia et al. (2003) introduce an interactive approach for multiobjective optimization which iteratively reduces the polyhedral region representing additive weights. The DM compares a pair of solutions x and y, and provides a rating a, which represents choice strength. Assuming that there is no response error and the DM prefers x to y, this comparison defines an equality constraint, $F(x)^T u - F(y)^T u = a$, on the corresponding polyhedral region U, where F is the vector representing the objective function values and $u \in U$. Toubia et al. (2004) extend this idea for the choice-based setting where the DM

only expresses preference without specifying rating information. Prefering x to y defines an inequality constraint, $F(x)^T u - F(y)^T u \geq 0$. These inequalities are used in the early interactive studies of Zionts and Wallenius (1976) and Zionts and Wallenius (1983) which solve MOLPs under linear and nonlinear value functions, respectively. The studies of Toubia et al. (2007) and Bertsimas and O'Hair (2013) extend the ideas of Toubia et al. (2003) further to handle response errors and inconsistencies.

Another way of reflecting the solution comparisons in the weight region is to focus on the neighborhoods around weight vectors that correspond to DM's most preferred solutions (see Steuer and Choo, 1983; Steuer et al., 1993; Köksalan and Karahan, 2010; Hassanzadeh et al., 2014). Underlying weight regions in these studies consist of weight vectors for WTP instead of WSP. In a related study, Hu and Mehrotra (2012) develop multicriteria robust weighted sum approach to minimize the worst-case weighted sum of objectives over a given weight region.

The quality of the preference information obtained during the comparison phase strongly depends on the selection of solution pairs (or sets) in the solution generation phase. Comparison of *quality* solution pairs reveals more information on the weight region providing faster convergence with fewer iterations. Zionts and Wallenius (1976) and Zionts and Wallenius (1983) use adjacent nondominated solution pairs, ensuring that the difference between the objective function values are above a threshold. This can be challenging due to potential similarity of adjacent solutions. Toubia et al. (2003, 2004) and Bertsimas and O'Hair (2013) find solution pairs whose comparison yields a hyperplane that passes through or near analytic center of the remaining weight region. In Toubia et al. (2003), the solution pair yields a hyperplane that is almost perpendicular to the longest

axis of the bounding hyper-ellipsoid around the remaining region. This idea is extended in Toubia et al. (2004) to generate k solution pairs using the longest k axes of the bounding hyper-ellipsoid: They identify the longest k axes, obtain $2k$ weight vectors by taking the intersection of each axis and the weight polytope, and solve WSP with those weights to generate solution pairs. The corresponding hyperplanes pass approximately through the analytic center. In their setting, the objective function values are clearly determined as they directly come from solution features whose levels are known a priori; therefore, those hyperplanes may not reduce the weight region efficiently in a more general multiobjective setting. Bertsimas and O'Hair (2013) develop a similar approach for problems with a fixed number of known solutions: Instead of selecting a pair with a nearly perpendicular hyperplane to the longest axis, they choose the pair whose hyperplane is closest to the analytic center. Although this approach is promising in reducing the polytope efficiently, direct application to general multiobjective optimization problems is challenging for two reasons: (i) nondominated solutions are not known a priori and (ii) all possible solution pairs must be enumerated.

Another way of obtaining high quality information is to provide diversified solutions to the DM from different parts of the solution space. Many studies focus on generating diversified nondominated solutions by using various measures of cardinality, region coverage and solution spacing (see Faulkenberg and Wiecek (2010) for a detailed review). A common diversification approach is the filtering approach (Steuer and Harris, 1980) which filters similar solutions from a large set of nondominated solutions, yielding a small subset of diverse solutions. In many cases, generating large sets of nondominated solutions is impractical, making the filtering approach inefficient. In addition to filtering

approaches, Das and Dennis (1998), Messac and Mattson (2002), Kim and De Weck (2006) and Masin and Bukchin (2008) develop various approaches with the aim of generating a well-distributed set of nondominated solutions in the objective space. Note that the aim of our study is neither to generate the entire efficient frontier nor a well-distributed representation of it. Rather, we focus on finding nondominated solutions whose comparison reveals more information on the DM's preferences, reducing the remaining weight region faster. Diversification efforts are centered in the remaining weight region instead of the solution or objective spaces commonly used in the literature. We refer the reader to Antunes et al. (2016) and Alves and Clímaco (2007) for reviews of interactive approaches on MOLPs and MOIPs, respectively.

IWRA finds the most diverse weight vector in the remaining weight region to generate a new nondominated solution. As discussed in Das and Dennis (1997), generating evenly distributed weight vectors in the weight region does not necessarily yield well-distributed nondominated solutions. Therefore, instead of finding the most diverse weight vector w.r.t. explored weight vectors, we find it w.r.t. weight sets of explored nondominated solutions which contain all weight vectors for which the nondominated solution is optimal to the corresponding WSP. This way, the optimal bases of WSP in previous iterations cannot be optimal in subsequent iterations, providing a finite convergence to algorithms and a methodology to explore all nondominated solutions. For MOLPs, these weight sets are obtained using optimal bases; for MOIPs, an approximation approach is used since the bases are not trivially available. To the best of our knowledge, diversification w.r.t. weight sets has not been considered in the literature. In this study, we show this

approach has significant value in limiting the number of solutions explored and in finding high quality solutions even when the DM is inconsistent in comparisons.

3.2. Problem Setting

Consider the following multiobjective optimization problem (MOP) with k objectives:

$$\text{(MOP)} \qquad \max_{x \in \mathcal{X}} \; F(x) = [f_1(x), f_2(x), ..., f_k(x)]^T,$$

where $\mathcal{X} \subset \mathbb{R}^n$ is the feasible solution set and $F : \mathbb{R}^n \to \mathbb{R}^k$ represents the objective functions.

Definition 3.1. *A solution $x^* \in \mathcal{X}$ is a strict nondominated solution if there does not exist $\bar{x} \in \mathcal{X}$ such that $f_i(\bar{x}) \geq f_i(x^*)$ for all $i \in 1, 2, ..., k$ with at least one strict inequality.*

Definition 3.2. *A solution $x^* \in \mathcal{X}$ is a weak nondominated solution if there does not exist $\bar{x} \in \mathcal{X}$ such that $f_i(\bar{x}) > f_i(x^*)$ for all $i \in 1, 2, ..., k$.*

We consider MOLPs and MOIPs with the assumption that the problems are bounded, i.e., trade-offs between the objectives are finite. This is a reasonable assumption since most practical problems are intrinsically bounded. For both settings, $F(x) = C^T x$ where C is a $n \times k$ real valued matrix. The i^{th} column of C matrix, c_i, represents the coefficients of i^{th} objective function, i.e., $f_i(x) = c_i^T x$. The feasible solution set of an MOLP is defined by a linear system of equalities $Ax = b$ and inequalities $x \geq 0$, where A is an $m \times n$ real-valued matrix and b is an m-dimensional real-valued vector. The feasible solution set of an MOIP is defined with an additional restriction $x \in \mathbb{Z}$.

3.2.1. The Weighted Sum Problem

Assuming that the DM is interested in optimizing a weighted combination of objectives, all nondominated solutions of interest can be generated by the following weighted sum problem (WSP):

$$\text{(WSP)} \qquad g(w) = \max_{x \in \mathcal{X}} \; w^T F(x) = \max_{x \in \mathcal{X}} \; \sum_{i=1}^{k} w_i f_i(x),$$

where w is a k-dimensional vector in weight region $\mathcal{W} := \{w | \sum_{i=1}^{k} w_i = 1;\, w_i \geq 0,\, i = 1, 2, ..., k\}$ or its polyhedral subset.

Proposition 3.3. *(Geoffrion, 1968) Letting x_w^* be an optimal solution to WSP for a given weight vector $w \in \mathcal{W}$, x_w^* is a nondominated solution when w is strictly positive ($w_i > 0$, $i = 1, 2, ..., k$).*

Note that x_w^* can be a weak nondominated solution when w is not strictly positive. WSP can generate all nondominated solutions when the underlying problem is convex as in MOLPs; it overlooks unsupported nondominated solutions for nonconvex problems such as MOIPs. A nondominated solution x^* is unsupported if there does not exist any $w \in \mathcal{W}$ such that the optimal solution of WSP is x^*. We first assume that unsupported nondominated solutions are not of interest to the DM, then relax this assumption in Section 3.7. We also assume that WSP is solved by the simplex method for MOLPs; i.e, the bases corresponding to identified nondominated solutions are available.

3.2.2. Weight Sets of Nondominated Solutions

A nondominated solution can be optimal for multiple weight vectors. Let weight set W_{x^*} be the set of weight vectors such that nondominated solution x^* is optimal to the corresponding WSP:

$$W_{x^*} = \{w \in \mathcal{W} \mid x^* \in arg\max_{x \in \mathcal{X}} \; w^T F(x)\}. \tag{3.1}$$

The weight set of a nondominated solution is a crucial component in IWRA to diversify the weight region. Next, we show the convexity of weight sets independent of problem setting and then prove that these sets are polyhedral for MOLPs and MOIPs.

Proposition 3.4. *W_{x^*}, as defined in* (3.1)*, is a convex set for a nondominated solution x^*.*

Proof of Proposition 3.4. Consider $w_1, w_2 \in W_{x^*}$ and assume that there exists a $\lambda \in [0, 1]$ such that $(\lambda w_1 + (1 - \lambda)w_2) \notin W_{x^*}$. For $g(\lambda w_1 + (1 - \lambda)w_2)$, there must exist a nondominated solution $\bar{x}$ such that:

$$(\lambda w_1 + (1 - \lambda)w_2)^T F(\bar{x}) > (\lambda w_1 + (1 - \lambda)w_2)^T F(x^*),$$

which is a contradiction since $w_1^T F(x^*) \geq w_1^T F(\bar{x})$ and $w_2^T F(x^*) \geq w_2^T F(\bar{x})$, thus W_{x^*} is a convex set. □

Proposition 3.5. *Let x^* be a nondominated extreme point solution of an MOLP where $\mathcal{X} := \{Ax = b, x \geq 0\}$. Then, W_{x^*}, as defined in* (3.1)*, is a polytope for a nondominated solution x^*.*

Proof of Proposition 3.5. For MOLPs, nondominated solution x^* is an extreme point of polyhedron $\mathcal{X}$ for any $w \in \mathcal{W}$. We first assume that x^* has a unique basis, then generalize the proof for solutions with multiple bases. Assuming that x^* has a unique basis $\mathcal{B}_{x^*}$, W_{x^*} can be obtained from the dual of WSP, D-WSP:

$$\text{(D-WSP)} \qquad \min \ \pi b$$
$$\text{s.t.} \qquad \pi A \geq w^T C^T.$$

where π is an m-dimensional row vector representing the dual variables. Letting $C^T_{\mathcal{B}_{x^*}}$ be the submatrix formed from the objective coefficient columns corresponding to basic variables from C^T, the dual optimal solution π^* can be written as $\pi^* = w^T C^T_{\mathcal{B}_{x^*}} \mathcal{B}^{-1}_{x^*}$. From duality, x^* is optimal and $\mathcal{B}_{x^*}$ is the corresponding optimal basis if D-WSP is feasible for π^*:

$$\pi^* A \geq w^T C^T,$$
$$w^T C^T_{\mathcal{B}_{x^*}} \mathcal{B}^{-1}_{x^*} A \geq w^T C^T,$$
$$w^T (C^T - C^T_{\mathcal{B}_{x^*}} \mathcal{B}^{-1}_{x^*} A) \leq 0, \tag{3.2}$$

which define inequalities for weight vectors for which x^* is optimal. Similar inequalities are derived by using reduced cost and duality arguments in Zeleny (1974) and Isermann (1974). Constraints (3.2) along with $\sum_{i=1}^{k} w_i = 1$ and $w \geq 0$ specify weight set $W_{x^*} \subseteq \mathcal{W}$, indicating that W_{x^*} is a polytope for MOLPs when x^* has a unique basis. If x^* has multiple bases, W_{x^*} is the union of finitely many polytopes representing the weight regions for each

basis. From Proposition 3.4 we know that this union is convex which implies that it is a polytope (see Theorem 3 and Remark 1 of Bemporad et al. (2001) for details). This indicates that W_{x^*} is a polytope for MOLPs when x^* has multiple bases, which completes the proof. □

Proposition 3.6. *Let x^* be a supported nondominated solution of an MOIP where $\mathcal{X} := \{Ax = b, x \in \mathbb{Z}^+\}$. Then, W_{x^*}, as defined in* (3.1)*, is a polytope for a nondominated solution x^*.*

Proof of Proposition 3.6. Ignoring unsupported nondominated solutions, the feasible set of solutions of any MOIP, $\mathcal{X} = \{Ax = b, x \in \mathbb{Z}^+\}$, can be replaced with an equivalent linear programming formulation, $\tilde{\mathcal{X}} = \{\tilde{A}x = \tilde{b}, x \geq 0\}$ (Wolsey and Nemhauser, 1999), by transforming the MOIP to its integer hull representation as an equivalent MOLP. Therefore, Proposition 3.5 holds for MOIPs. □

3.3. Interactive Weight Region-Based Approach

IWRA is an iterative algorithm which alternates between comparison phase (Section 3.3.1) and solution generation phase (Section 3.3.2). The comparison phase allows the DM to communicate preferences by comparing nondominated solutions. These comparisons define a set of inequalities on the weight region eliminating weight vectors that do not align with the DM's preferences. We translate preference information into the weight region as in Toubia et al. (2004): A preference of x over y defines an inequality constraint, $w^T F(x) - w^T F(y) \geq 0$, in the weight region. In the solution generation phase, the algorithm finds a new nondominated solution by solving the WSP for the most diverse weight vector which is obtained by solving an MILP that maximizes the distance of new

weight vector from already explored weight sets. Both phases are repeated until the iterative algorithm satisfies termination conditions.

3.3.1. Comparison Phase: Comparison Settings

We consider two comparison settings: (i) pairwise comparison and (ii) insertion. In pairwise comparison, the DM compares the new nondominated solution x_t^* with her current most preferred nondominated solution x^{**}. If x_t^* is preferred to x^{**}, then $w^T F(x_t^*) - w^T F(x^{**}) \geq 0$ must hold for the DM's optimal weight vector and x_t^* becomes x^{**}; else, the direction of the constraint is reversed and x^{**} remains unchanged.

In insertion, the DM maintains an ordered list of nondominated solutions based on preference, $\{x_{(1)}^*, x_{(2)}^*, ..., x_{(t-1)}^*\}$, where $x_{(1)}^*$ is the most preferred and $x_{(t-1)}^*$ is the least. At iteration t, the DM inserts x_t^* into the list based on her preference. Insertion yields a single constraint when x_t^* is inserted at the beginning of the list:

$$w^T F(x_t^*) - w^T F(x_{(1)}^*) \geq 0, \tag{3.3}$$

or at the end of the list:

$$w^T F(x_{(t-1)}) - w^T F(x_t^*) \geq 0. \tag{3.4}$$

When x_t^* is inserted between i and $i+1$, $1 \leq i \leq (t-2)$, two constraints are obtained:

$$w^T F(x_{(i)}^*) - w^T F(x_t^*) \geq 0, \tag{3.5}$$

$$w^T F(x_t^*) - w^T F(x_{(i+1)}) \geq 0. \tag{3.6}$$

Only the immediate constraints reveal information as the rest are redundant given the constraints from previous iterations. For example, when x_t^* is inserted into beginning of the list, x_t^* is also preferred to $x_{(2)}^*$, making the constraint $w^T F(x_t^*) - w^T F(x_{(2)}^*) \geq 0$ valid for the remaining weight region. This constraint is redundant when the immediate constraints of x_t^* and $x_{(1)}^*$ are considered together: $w^T F(x_t^*) - w^T F(x_{(1)}^*) \geq 0$ and $w^T F(x_{(1)}^*) - w^T F(x_{(2)}^*) \geq 0$ implies $w^T F(x_t^*) - w^T F(x_{(2)}^*) \geq 0$. The DM does not need to compare the new solution with all solutions in the list. The new solution can be inserted after a logarithmic number of comparisons with a bisection search.

Let $\mathcal{W}_t$ be the remaining weight region representing the set of feasible weight vectors at the t^{th} iteration. For a k-objective problem, IWRA starts with weight region $\mathcal{W}_0 = \{w \in \mathbb{R}^k \mid \sum_{i=1}^k w_i = 1, w \geq 0\}$ which is a unit $(k-1)$-simplex. Note that $\mathcal{W}_0$ contains all weight vector combinations since any weight vector can be optimal when there is no preference information. $\mathcal{W}_t$ is obtained by appending inequalities resulting from comparisons during the t^{th} comparison phase to $\mathcal{W}_{t-1}$.

3.3.2. Solution Generation Phase: Diverse Weight Vector Generation

In the solution generation phase, we generate the most diverse weight vector in the remaining region which is farthest from the explored weight sets. Generating the most diverse weight vector in a systematic way enables us to explore different parts of the remaining weight region. Furthermore, the corresponding nondominated solution for newly generated weight vector is potentially diverse from those already generated since newly

generated weight vector is far from explored weight sets. Comparison of such a nondominated solution with the existing nondominated solutions has potential to decrease the size of the remaining weight region significantly.

We use *constraint violation* as a measure of diversification. Constraint violation between vector w and polytope W_{x^*} returns a nonpositive value if w does not violate the constraints of W_{x^*} ($w \in W_{x^*}$) and returns the maximum of shortest l_2-norm distances between w and the hyperplanes that define the violated constraints of W_{x^*} otherwise. We refer to the violated constraint of W_{x^*} for which the maximum distance is realized as the farthest constraint of W_{x^*}. Constraint violation enables us to model the diversification problem as an MILP.

Let $W_{x^*} := \{Q_{x^*} w \leq r_{x^*}\}$ where Q_{x^*} is an $m_{x^*} \times k$ coefficient matrix and r_{x^*} is m_{x^*}-dimensional right-hand side (r.h.s.) vector. Let $q^z_{x^*}$ be the z^{th} row of Q_{x^*} matrix and $r^z_{x^*}$ be the z^{th} element of r_{x^*} vector. W.l.o.g. assume that these constraints are normalized, i.e., $||q^z_{x^*}|| = 1$ for $z = 1, 2, ..., m_{x^*}$. Letting $\mathbf{X}$ be the set of nondominated solutions identified by WSP up to and including iteration t, the most diverse weight vector for iteration $(t+1)$ is found by the following constraint violation model (CVM):

(3.7a) **CVM:** $\max \; d$

s.t.

(3.7b) $$d \leq (q^z_{x^*})^T w - r^z_{x^*} + M p_{zx^*}, \qquad \forall x^* \in \mathbf{X}, \;\; z = 1, 2, ..., m_{x^*}$$

(3.7c) $$\sum_{z=1}^{m_{x^*}} p_{zx^*} \leq m_{x^*} - 1, \qquad \forall x^* \in \mathbf{X}$$

(3.7d) $$p_{zx^*} \in \{0, 1\}, \qquad \forall x^* \in \mathbf{X}, \;\; z = 1, 2, ..., m_{x^*}$$

(3.7e) $$w \in \mathcal{W}_t,$$

where d is a continuous variable representing the distance, M is a sufficiently large number and p_{zx^*} is a binary variable which takes value of 0 if z^{th} constraint of W_{x^*} is the farthest and 1 otherwise. CVM maximizes the minimum constraint violation between w and weight sets of explored nondominated solutions in $\mathbf{X}$. Inequalities (3.7c) ensure that at least one p_{zx^*} variable is zero for each $x^* \in \mathbf{X}$, indicating that at least one constraint from each W_{x^*} polytope must be violated. When $p_{zx^*} = 0$, the r.h.s. of (3.7b) calculates the distance between w and the hyperplane defining the z^{th} constraint of W_{x^*}. If the z^{th} constraint is satisfied by w, then the r.h.s. becomes nonpositive; thus, the maximization model always sets p variables to 0 for the farthest constraint of each weight set W_{x^*}.

Proposition 3.7. *CVM is always feasible when $\mathcal{W}_t \neq \emptyset$.*

Proof of Proposition 3.7. Consider a w' which satisfies (3.7e) and an arbitrary selection of p'_{zx^*} variables such that (3.7c) and (3.7d) are satisfied (which can be done as follows: For each $x^* \in \mathbf{X}$, choose an arbitrary z' from $\{1, 2, ..., m_{x^*}\}$, set $p'_{z'x^*} = 0$ and $p'_{zx^*} = 1$ for $z = \{1, 2, ..., m_{x^*}\} \setminus \{z'\}$). Setting $d' = \min_{z,x^*}\{(q^z_{x^*})^T w' - r^z_{x^*} + Mp'_{zx^*}\}$, triplet (w', p', d') is feasible for CVM. □

Theorem 3.8. *CVM yields a strictly positive objective value if and only if remaining weight region is not fully explored; i.e., $d^* > 0$ if and only if $\mathcal{W}_t \setminus \left(\bigcup_{x^* \in \mathbf{X}} W_{x^*}\right) \neq \emptyset$.*

Proof of Theorem 3.8. First, we show that $d^* > 0 \Rightarrow \mathcal{W}_t \setminus \left(\bigcup_{x^* \in \mathbf{X}} W_{x^*}\right) \neq \emptyset$. Let w^* be the optimal solution of CVM. For any $x^* \in \mathbf{X}$, there exists at least one index

$\bar{z}$ such that $p_{\bar{z}x^*} = 0$ (due to constraints (3.7c)). In this case, constraint (3.7b) for $\bar{z}$ becomes:

$$d^* \leq (q^{\bar{z}}_{x^*})^T w^* - r^{\bar{z}}_{x^*} \qquad \Rightarrow \qquad (q^{\bar{z}}_{x^*})^T w^* - r^{\bar{z}}_{x^*} \geq d^* > 0, \tag{3.8}$$

indicating that W^* violates $\bar{z}^{th}$ constraint of W_{x^*}. Since this argument can be repeated for any W_{x^*}, $w^* \notin \left(\bigcup_{x^* \in \mathbf{X}} W_{x^*} \right)$. On the other hand, $w^* \in \mathcal{W}_t$ since it is a feasible solution to CVM. Therefore, $d^* > 0 \Rightarrow \mathcal{W}_t \setminus \left(\bigcup_{x^* \in \mathbf{X}} W_{x^*} \right) \neq \emptyset$.

Second, we show that $\mathcal{W}_t \setminus \left(\bigcup_{x^* \in \mathbf{X}} W_{x^*} \right) \neq \emptyset \Rightarrow d^* > 0$. Consider a feasible solution $\bar{w} \in \mathcal{W}_t \setminus \left(\bigcup_{x^* \in \mathbf{X}} W_{x^*} \right)$. Since $\bar{w} \notin W_{x^*}$, then $x^* \in \mathbf{X}$, $\bar{w}$ violates at least one constraint of W_{x^*} strictly. Letting $\bar{d}$ be the objective value of $\bar{w}$, $\bar{d} > 0$ can be obtained by setting the p variables for the violated constraints to 0 for each W_{x^*}, indicating that $d^* \geq \bar{d} > 0$. Hence, $\mathcal{W}_t \setminus \left(\bigcup_{x^* \in \mathbf{X}} W_{x^*} \right) \neq \emptyset \Rightarrow d^* > 0$, which completes the proof. □

Corollary 3.9. *CVM yields a nonpositive objective value if and only if remaining weight region is fully explored by the weight sets, i.e., $d^* \leq 0$ if and only if $\mathcal{W}_t \subseteq \left(\bigcup_{x^* \in \mathbf{X}} W_{x^*} \right)$.*

Proof of Corollary 3.9. Directly follows from Theorem 3.8. □

Theorem 3.8 and Corollary 3.9 enable us to check if there is a weight vector to explore, providing termination conditions for the algorithms discussed next.

The basic idea in CVM is to spread exploration of the weight region and to obtain potentially diverse solutions to compare. An alternative way to obtain new solutions is to explore adjacent weight regions to the current weight region which translates into exploration of the efficient adjacent solutions as in ZW method. In our experiments, we

compare IWRA-MOLP with ZW method to assess the value of diversification in weight region exploration when the DM makes perfect and imperfect comparisons.

3.4. IWRA-MOLP

Implementing IWRA for MOLPs is straightforward as the polyhedral representation of weight sets of nondominated solutions can be identified from the dual by using the optimal basis corresponding to that solution (see Proposition 3.5). Before we discuss the algorithm in detail, we first show that explored optimal bases of WSP cannot be optimal again for weight vector w as long as w is selected outside the weight sets defined by those bases.

Theorem 3.10. *Consider the exploration of the nondominated solutions of WSP. Let* $\mathbf{B}$ *be the set of known optimal bases corresponding to explored nondominated solutions until some iteration. Let* $W(B)$ *be the weight set corresponding to basis* $B \in \mathbf{B}$*, defined by inequalities* (3.2)*,* $\sum_{i=1}^{k} w_i = 1$ *and* $w \geq 0$*. A basis* $B \in \mathbf{B}$ *cannot be optimal for weight vector* $w \in \mathcal{W} \setminus \left(\bigcup_{B \in \mathbf{B}} W(B) \right)$*.*

Proof of Theorem 3.10. The weight vector $w \in \mathcal{W} \setminus \left(\bigcup_{B \in \mathbf{B}} W(B) \right)$ satisfies both $\sum_{i=1}^{k} w_i = 1$ and $w \geq 0$. Therefore, w must violate at least one of the inequalities (3.2) for each basis $B \in \mathbf{B}$. In this case, D-WSP becomes infeasible for all the bases in $\mathbf{B}$ implying that a basis $B \in \mathbf{B}$ cannot be optimal when $w \in \mathcal{W} \setminus \left(\bigcup_{B \in \mathbf{B}} W(B) \right)$. □

Theorem 3.10 does not guarantee that a new weight vector yields a new nondominated solution. In case of degeneracy, a nondominated solution may appear multiple times as multiple bases may correspond to the same solution; however, this cannot repeat infinitely many times since the number of bases of a polytope is finite.

Considering Theorem 3.10, one can design an iterative exploration algorithm which generates the entire efficient frontier formed by nondominated solutions of interest in finite number of iterations with a model generating weight vectors outside already explored weight sets. At each iteration of the exploration, WSP is solved for the new weight vector yielding a new optimal basis and the weight set for the corresponding optimal basis is constructed by using inequalities (3.2) with $\sum_{i=1}^{k} w_i = 1$ and $w \geq 0$. Since the number of bases is finite, this algorithm terminates after finite number of iterations. To this end, weight sets and optimal bases are used in multiobjective linear optimization (Zeleny, 1974) and ADBASE (Steuer, 2006) algorithms when enumerating the efficient frontier by generating adjacent nondominated solutions to already explored ones.

Algorithm IWRA-MOLP

Algorithm 3.1 summarizes IWRA for MOLPs. In the algorithm, t represents the iteration number, w_t is the weight vector used in iteration t to identify nondominated solution x_t^* and $\mathbf{W}$ is the set of explored weight regions.

IWRA-MOLP starts by initializing $t = 0$, $\mathcal{W}_t$ as unit $(k-1)$-simplex and w_t as a k-dimensional random initial vector. The first nondominated solution x_t^* is obtained by solving WSP for w_t and set $\mathbf{X}$ is initialized as $\{x_t^*\}$. The weight set of x_t^*, $W_{x_t^*}$, is constructed by D-WSP and set $\mathbf{W}$ is initialized as $\{W_{x_t^*}\}$. Weight vector w_{t+1} is the optimal solution of CVM over $\mathcal{W}_t$ with an optimal value of d_{t+1}^*. The while loop repeats until the optimal solution of CVM is nonpositive. At the start of each iteration, WSP returns a nondominated solution x_t^* for the weight vector w_t. New nondominated solutions are shown to the DM for preference information to update x^{**} and to reduce the weight

Algorithm 3.1 *IWRA-MOLP*

Initialize $t = 0$, $\mathcal{W}_t$ and w_t. Solve WSP for w_t to obtain x_t^*.
Set $x^{**} = x_t^*$. Initialize set $\mathbf{X} \leftarrow \{x_t^*\}$.
Construct $W_{x_t^*}$ using D-WSP. Set $\mathbf{W} \leftarrow \{W_{x_t^*}\}$.
Solve CVM over $\mathcal{W}_t$ to obtain w_{t+1}, which is at least d_{t+1}^* from $W \in \mathbf{W}$.
while $d_{t+1}^* > 0$ **do**
  $t \leftarrow t + 1$.
  Solve WSP for w_t to obtain x_t^*.
  Construct $W_{x_t^*}$ using D-WSP. $\mathbf{W} \leftarrow \mathbf{W} \cup \{W_{x_t^*}\}$
  if $x_t^* \in \mathbf{X}$ **then**
    $\mathcal{W}_t \leftarrow \mathcal{W}_{t-1}$.
  else
    Update $\mathbf{X} \leftarrow \mathbf{X} \cup \{x_t^*\}$. Ask the DM to provide preference information.
    if x_t^* is preferred to x^{**} **then**
      $x^{**} \leftarrow x_t^*$.
    end if
    Update $\mathcal{W}_{t-1}$ with the DM's preference information to obtain $\mathcal{W}_t$.
  end if
  Solve CVM over $\mathcal{W}_t$ to obtain w_{t+1}, which is at least d_{t+1}^* away from $W \in \mathbf{W}$.
end while
Return x^{**} as the most preferred nondominated solution of the DM.

region $\mathcal{W}_t$. The loop breaks when there are no more weight vectors (and nondominated solutions of interest) to explore and the algorithm returns x^{**} as the most preferred nondominated solution for the DM.

Theorem 3.11. *IWRA-MOLP terminates after finite number of iterations.*

Proof of Theorem 3.11. Theorem 3.10 states that the same basis cannot be repeated in subsequent iterations as long as the new weight vector is chosen from the unexplored region. From Theorem 3.8, we know that the new weight vector returned by CVM belongs to the unexplored region iff $d^* > 0$. Since there are finitely many bases for WSP, which are never revisited, the while loop of IWRA-MOLP cannot repeat infinitely many times. □

3.5. IWRA-MOIP

The fundamental challenge in solving MOIPs with IWRA is the identification of weight sets for which a given nondominated solution is optimal. Since duals are not easily available for MOIPs (e.g. when WSP is solved using branch & bound), the weight sets cannot be identified immediately unless the local polyhedral information around the nondominated solution is known. For this reason, we introduce the concept of partial weight sets which are subsets of the exact weight sets of nondominated solutions.

First, we discuss an important property of the extreme points of the remaining weight region which enables finite termination for IWRA-MOIP even though the weight sets are known partially.

Theorem 3.12. *Given the set of explored solutions* $\mathbf{X}$*, current most preferred solution* x^{**} *and the remaining polyhedral weight region of* $\mathcal{W}_t$*, exploring the weight vectors corresponding to the extreme points of* $\mathcal{W}_t$ *either yields an unexplored solution or results in complete exploration of* $\mathcal{W}_t$*.*

In order to prove Theorem 3.12, we first introduce Lemma 3.13.

Lemma 3.13. *Given remaining weight region* $\mathcal{W}_t$ *at iteration* t *and current most preferred solution* x^{**}*, any weight vector* $w \in \mathcal{W}_t$ *satisfies* $w^T F(x^{**}) \geq w^T F(x^*)$ *for* $x^* \in \mathbf{X}$*.*

Proof of Lemma 3.13. The polytope representing $\mathcal{W}_t$ satisfies the following inequalities (some may be redundant) as the DM prefers x^{**} to all solutions in $\mathbf{X} \setminus \{x^{**}\}$:

$$w^T F(x^{**}) \geq w^T F(x^*), \qquad x^* \in \mathbf{X} \setminus \{x^{**}\}. \tag{3.9}$$

Since $w^T F(x^{**}) = w^T F(x^{**})$ holds trivially, set $\mathbf{X} \setminus \{x^{**}\}$ can be extended to $\mathbf{X}$, indicating that explored nondominated solutions cannot yield a larger WSP objective value than $w^T F(x^{**})$ for $w \in \mathcal{W}_t$. □

Proof of Theorem 3.12. Letting w_e be a weight vector which corresponds to one of the extreme points of $\mathcal{W}_t$ and x_e be the corresponding nondominated solution of WSP for w_e, x_e must satisfy $w_e^T F(x_e) \geq w_e^T F(x^{**})$. Otherwise x^{**} is a better solution than x_e for weight vector w_e, which contradicts the definition of x_e. When $w_e^T F(x_e) = w_e^T F(x^{**})$, $w_e \in W_{x^{**}}$ since x^{**} is an optimal solution of WSP at w_e. If this condition holds for all extreme points, x^{**} is optimal for all weight vectors in the convex hull of $\mathcal{W}_t$ since $W_{x^{**}}$ is known to be a convex set from Proposition 3.4. This indicates that there are no more weight vectors to explore in $\mathcal{W}_t$. When $w_e^T F(x_e) > w_e^T F(x^{**})$, x_e is an unexplored solution, i.e., $x_e \notin \mathbf{X}$, since each explored solution $x \in \mathbf{X}$ satisfies $w_e^T F(x^{**}) \geq w_e^T F(x)$ from Lemma 3.13. □

With Theorem 3.12, one can introduce a basic algorithm: Explore the extreme points of the remaining weight region iteratively starting from the initial unit simplex. From Theorem 3.12, at each iteration the algorithm either explores the vertices of the remaining region completely or finds a new solution which yields an update in the remaining weight region. Such an algorithm terminates in finite time since the number of nondominated solutions is finite and the algorithm identifies at least one new nondominated solution at each iteration until termination. The algorithm can be inefficient since generating all extreme points of the remaining region is a difficult task. For this reason, we propose

Algorithm 3.3 which explores the extreme points of the remaining weight region only if it cannot find a new nondominated solution for certain number of trial iterations.

Algorithm IWRA-MOIP

From Proposition 3.6, the weight sets of supported nondominated solutions are polytopes for MOIPs. If nondominated solution x^* is optimal for weight vectors $w_1, w_2, ..., w_s$, then x^* is also optimal for any weight vector in $conv(w_1, w_2, ..., w_s)$ where $conv(.)$ represents the convex hull. With this observation, IWRA-MOIP uses partial weight sets which are updated at each iteration. Letting S_{x^*} be the partial weight set which contains the explored weight vectors for which the nondominated solution x^* is optimal, $conv(S_{x^*})$ is an approximation of exact weight set W_{x^*} at a given iteration. As IWRA-MOIP progresses, existing partial weight sets are updated and new ones are introduced for newly identified nondominated solutions. The weight sets of explored solutions are not identified exactly and IWRA-MOIP can return an already explored solution.

Consider a weight vector $\bar{w}$ with a corresponding solution $\bar{x}$. Weight vector $\bar{w}$ can be added to partial weight set S_{x^*} if $\bar{w}^T F(\bar{x}) = \bar{w}^T F(x^*)$ holds. This condition holds trivially when $\bar{x} = x^*$. If the equality holds when $\bar{x} \neq x^*$, then WSP has multiple optimal solutions for $\bar{w}$. In this case, $\bar{w}$ can be added to both $S_{\bar{x}}$ and S_{x^*} since $\bar{w}$ is an element of a face touched by both polytopes representing exact weight sets of $\bar{x}$ and x^*. Algorithm 3.2 formally presents the method for updating set of partial weight sets $\mathbf{S}$, $UpdatePartialWeightSets(\mathbf{S}, \mathbf{X}, \bar{w}, \bar{x})$, for a given set of explored solutions $\mathbf{X}$, current solution $\bar{x}$ and current weight vector $\bar{w}$.

Algorithm 3.2 *UpdatePartialWeightSets(***S**, **X**, $\bar{w}$,$\bar{x}$*)*

```
if x̄ ∈ X then
    S_x̄ ← S_x̄ ∪ {w̄}.
else
    Construct new partial weight set S_x̄ = {w̄}. S ← S ∪ S_x̄.
    for each x* ∈ X do
        for each w ∈ S_x* do
            if w^T F(x̄) = w^T F(x*) then
                S_x̄ ← S_x̄ ∪ {w}.
            end if
        end for
    end for
end if
for each x* ∈ X do
    if w̄^T F(x̄) = w̄^T F(x*) then
        S_x* ← S_x* ∪ {w̄}.
    end if
end for
```

Algorithm 3.3 summarizes IWRA for MOIPs. In addition to the parameters and sets defined in Section 3.4, let s be the number of consecutive iterations during which no new nondominated solution is found. IWRA-MOIP starts by initializing $t = 0$, $\mathcal{W}_t$ as unit $(k-1)$-simplex and w_t as a k-dimensional random initial vector. The first solution x_t^* is obtained by solving WSP for w_t and $\mathbf{X}$ is initialized as $\{x_t^*\}$. Initial partial weight set is constructed and CVM returns the most diverse weight vector w_{t+1} with an optimal value of d_{t+1}^*. At each iteration, the algorithm finds a solution, updates the partial weight sets and initiates a comparison if the current solution is a new nondominated solution. Instead of exact weight sets used in IWRA-MOLP, CVM finds the most diverse weight vector with respect to polytopes obtained by projecting $conv(S)$, $S \in \mathbf{S}$ via Fourier-Motzkin elimination onto k-dimensional weight space. In order to have finite convergence in this algorithm, the extreme points of the remaining weight region $\mathcal{W}_t$ are checked when the algorithm returns existing nondominated solutions for s_{max} consecutive iterations.

Theorem 3.14. *IWRA-MOIP terminates after finite number of iterations.*

Algorithm 3.3 *IWRA-MOIP*

Initialize $t = 0$, $\mathcal{W}_t$ and w_t. Initialize $s = 0$ and set $\mathbf{S} = \emptyset$.
Solve WSP for w_t to obtain x_t^*.
Set $x^{**} = x_t^*$. $UpdatePartialWeightSets(\mathbf{S}, \mathbf{X}, w_t, x_t^*)$.
Initialize set $\mathbf{X} = \{x_t^*\}$.
Solve CVM over $\mathcal{W}_t$ to obtain w_{t+1}, which is at least d_{t+1}^* from $conv(S), \forall S \in \mathbf{S}$.
while $d_{t+1}^* > 0$ **do**
  $t \leftarrow t + 1$.
  Solve WSP for w_t to obtain x_t^*.
  $UpdatePartialWeightSets(\mathbf{S}, \mathbf{X}, w_t, x_t^*)$.
  if $x_t^* \in \mathbf{X}$ **then**
    $s \leftarrow s + 1$, $\mathcal{W}_t \leftarrow \mathcal{W}_{t-1}$.
  else
    Update $\mathbf{X} \leftarrow \mathbf{X} \cup \{x_t^*\}$. Ask the DM to provide preference information.
    if x_t^* is preferred to x^{**} **then**
      $x^{**} \leftarrow x_t^*$.
    end if
    Update $\mathcal{W}_{t-1}$ with the DM's preference information to obtain $\mathcal{W}_t$.
  end if
  if $s > s_{max}$ **then**
    $s \leftarrow 0$.
    for each w corresponding to an extreme point of $\mathcal{W}_t$ **do**
      Solve WSP to obtain x^*. $UpdatePartialWeightSets(\mathbf{S}, \mathbf{X}, w, x^*)$.
      if $x^* \notin \mathbf{X}$ **then**
        Update $\mathbf{X} \leftarrow \mathbf{X} \cup \{x^*\}$. Ask the DM to provide preference information.
        if x^* is preferred to x^{**} **then**
          $x^{**} \leftarrow x^*$.
        end if
        Update $\mathcal{W}_{t-1}$ with the DM's preference information to obtain $\mathcal{W}_t$.
        break
      end if
    end for
  end if
  Solve CVM over $\mathcal{W}_t$ to obtain w_{t+1}, which is at least d_{t+1}^* away from $conv(S), \forall S \in \mathbf{S}$.
end while
Return x^{**} as the most preferred nondominated solution of the DM.

Proof of Theorem 3.14. IWRA-MOIP controls the extreme points of the remaining weight region when it cannot find a new nondominated solution for $(s_{max} + 1)$ iterations. Theorem 3.12 guarantees that exploring the extreme points identifies a new nondominated solution unless all the solutions are explored in the remaining region. Since a bounded MOIP has finitely many nondominated solutions, IWRA-MOIP cannot exceed $\phi(s_{max} + 1)$ iterations where ϕ is the number of nondominated solutions; indicating that IWRA-MOIP terminates after finite number of iterations. □

3.6. Numerical Experiments

This section presents experiments to evaluate the performance of IWRA in randomly generated instances and benchmark instances from the literature. Section 3.6.1 presents results on simulated test instances: Section 3.6.1.1 introduces our simulated test instances, Sections 3.6.1.2 and 3.6.1.3 present results of IWRA-MOLP (in comparison with ZW method) and IWRA-MOIP on these instances, respectively. Section 3.6.1.4 presents results when the DM makes imperfect comparisons. Lastly, Section 3.6.2 compares IWRA-MOIP with the cone-based interactive approach of Lokman et al. (2016) using benchmark instances. All experiments are conducted on a 64-bit computer with Windows 7 OS with Intel Core2 Duo E6600 2.40GHz processor and 8.00 GB RAM.

3.6.1. Performance on Simulated Test Instances

In experiments with simulated test instances, comparisons are made by a simulated DM (sDM) with a randomly generated optimal weight vector. Guided by this weight vector, the sDM compares given solutions and returns the comparison result. Experiments in Sections 3.6.1.2 and 3.6.1.3 do not consider response errors; therefore, sDM's comparisons are correct and consistent with its optimal weight vector. This assumption is relaxed in Section 3.6.1.4 where the sDM is allowed to make mistakes during comparisons. Without knowing the optimal weight vector, IWRA algorithms find the most preferred nondominated solution of the sDM (or a high quality solution when the sDM makes imperfect comparisons). Corresponding instances are replicated 25 times for randomly initialized optimal weight vectors of sDM and random starting weight vectors for the algorithm. The

same initial weight vectors are used in both comparison settings for consistent evaluations. The results presented are the average values of 25 replications for each instance.

From Corollary 3.9, IWRA algorithms must terminate when $d_t^* \leq 0$ to guarantee that the remaining region is explored completely; during the experiments, IWRA-MOLP and IWRA-MOIP terminate when $d_t^* \leq 10^{-4}$ to avoid precision issues. If CVM cannot return the optimal solution within the first five seconds, the current best solution is accepted as the next weight vector if its objective function value is greater than 10^{-4}. Otherwise, the optimization procedure continues until CVM yields a feasible solution with objective value greater than 10^{-4}. IWRA-MOIP checks the extreme points of the remaining region if the algorithm cannot find a new nondominated solution for k consecutive iterations where k is the number of objectives of the underlying instance.

3.6.1.1. Instance Generation. Simulated test instances include nine MOLP instances where MOLP-k is an instance with $k = 2, 3, ..., 10$ objectives and four MOIP instances where MOIP-k is an instance with $k = 2, 3, 4, 5$ objectives. To generate a test instance with n variables we first generate the following *cube* constraints defining the n-dimensional hypercube with edge length of l:

$$0 \leq x_i \leq l \qquad i = 1, 2, ..., n. \tag{3.10}$$

We add a set of *tangential* constraints which are tangential to the n-dimensional hypersphere inscribed in the hypercube defined by inequalities (3.10). The radius of this hypersphere is $\frac{l}{2}$. Letting α be a randomly generated n-dimensional row vector, the hyperplane defining the following inequality is tangential to the inscribed hypersphere:

$$\alpha^T x \leq \frac{l}{2}\Big(\sum_{i=1}^{n} \alpha_i + (\alpha^T \alpha)^{\frac{1}{2}}\Big). \tag{3.11}$$

For MOLP-k instances, we set $n = 100$ and $l = 2$; the edge length of hypercube is 2 units and the inscribed hypersphere has a radius of 1. For MOIP-k instances, we set $n = 50$ and $l = 10$. The edge length is increased by a factor of 5 to include more integer solutions in the feasible region. We generate n tangential constraints for each problem instance. The elements of the row vectors representing tangential constraints are assigned randomly from a continuous uniform distribution on the interval $[-1, 1]$. Lastly, k additional row vectors representing objective coefficients are created similar to row vectors of tangential constraints.

In order to assess the difficulty of the generated instances, each instance is solved for arbitrary weight vectors for five hours and a lower bound on the number of unique nondominated solutions is obtained. Table 3.1 provides a summary of the lower bounds for each instance. The number of unique nondominated solutions in MOLP-k instances for $k > 5$ grows quickly reaching an approximate limit over $600,000$ due to time limit on solution generation. This indicates that the lower bounds become looser as the number of objectives increases. For IP instances with 4 and 5 objectives, the number of unique nondominated solutions is over $10,000$ and $30,000$ respectively.

3.6.1.2. IWRA-MOLP Runs. Table 3.2 summarizes the results for IWRA-MOLP on MOLP-k instances, $k = 2, 3, ..., 10$, for pairwise comparison and insertion settings. Each cell reports the average and the standard deviation of the corresponding metric over 25 replications. *Count* columns provide the number of comparisons made by the DM, *Comp*,

Table 3.1. Lower bounds on the number of unique nondominated solutions.

Instance	*Lower Bound*	*Instance*	*Lower Bound*
MOLP-2	193	MOLP-9	640961
MOLP-3	13981	MOLP-10	620060
MOLP-4	320438		
MOLP-5	541499	MOIP-2	51
MOLP-6	600204	MOIP-3	1269
MOLP-7	626218	MOIP-4	10580
MOLP-8	630553	MOIP-5	30806

and the number of total solutions presented, *TSoln*. *Duration* columns provide the *Total* time spent by the algorithm along with the portions of time spent on solving WSPs and weight generation models, *WGen*. The last column provides an upper bound on the ratio of the number of explored solutions to the number of nondominated solutions, $\mathcal{R}$. Since the actual number of nondominated solutions is unknown, the ratio is overestimated by using the lower bounds in Section 3.6.1.1. In both comparison settings, IWRA-MOLP finds a most preferred solution of the sDM for each instance at each replication. In insertion, a most preferred solution is found by exploring 35-45 solutions in 10-objective linear programming instances with more than a half million nondominated solutions. The number of solutions for the same instances increases to 90-200 in pairwise setting since comparisons yield less information.

Table 3.2. IWRA-MOLP results in 'avg (stdev)' form.

	Pairwise Comparison						Insertion					
	Count		Duration (s)				Count		Duration (s)			
Problem	Comp	TSoln	Total	WSP	WGen	$\mathcal{R}$ (avg)	Comp	TSoln	Total	WSP	WGen	$\mathcal{R}$ (avg)
MOLP-2	9 (2)	10 (2)	3 (1)	1 (0)	0 (0)	5.2E-02	22 (4)	9 (1)	3 (0)	1 (0)	0 (0)	4.8E-02
MOLP-3	19 (4)	20 (4)	5 (1)	1 (0)	1 (0)	1.5E-03	45 (6)	15 (2)	4 (0)	1 (0)	0 (0)	1.1E-03
MOLP-4	33 (8)	34 (8)	16 (9)	2 (1)	8 (7)	1.1E-04	69 (10)	21 (2)	7 (1)	1 (0)	2 (1)	6.4E-05
MOLP-5	43 (11)	44 (11)	95 (85)	3 (1)	83 (83)	8.2E-05	89 (10)	25 (2)	13 (4)	2 (0)	7 (3)	4.6E-05
MOLP-6	59 (13)	60 (13)	241 (95)	4 (1)	226 (92)	9.9E-05	108 (12)	29 (2)	36 (15)	2 (0)	28 (14)	4.8E-05
MOLP-7	84 (26)	85 (26)	467 (249)	6 (2)	444 (242)	1.4E-04	132 (14)	33 (3)	78 (19)	2 (0)	69 (18)	5.3E-05
MOLP-8	91 (19)	92 (19)	570 (205)	6 (1)	546 (201)	1.5E-04	142 (14)	35 (2)	102 (23)	2 (0)	92 (23)	5.6E-05
MOLP-9	115 (21)	116 (21)	837 (196)	8 (1)	804 (191)	1.8E-04	159 (17)	38 (3)	152 (36)	3 (0)	141 (35)	6.0E-05
MOLP-10	145 (33)	146 (33)	1119 (350)	10 (2)	1077 (343)	2.4E-04	177 (16)	42 (3)	190 (36)	3 (0)	177 (35)	6.7E-05

The number of comparisons needed to find a most preferred solution increases with the number of objectives, as expected since the dimension of the weight region is directly correlated to the number of objectives and higher dimensional regions are more difficult to explore. The total number of unique solutions is one more than the total number of comparisons as the new solution is compared to the most preferred one at each iteration. Similarly, the total algorithm duration increases with the number of objectives. While WSP uses a small portion of total time, more than 90% of the total time is spent on solving CVMs for instances with 5 or more objectives. CVM duration increases with the number of explored solutions since the underlying MILP size depends on the number of explored solutions. $\mathcal{R}$ values in the last columns show the effectiveness of IWRA-MOLP, requiring exploration of a small fraction of all solutions. The increasing trend in this measure for instances with 5 or more objectives does not represent the real behavior since the upper bounds are weaker due to loose lower bounds on the number of nondominated solutions.

In insertion, the total number of comparisons is larger than the pairwise setting while the total number of unique solutions is significantly smaller. During the comparison phase, sDM requires multiple comparisons to find the location of the new solution in the sorted list and this procedure yields tighter constraints on the remaining weight region compared to those obtained by pairwise comparison. With insertion, IWRA-MOLP is able to find the most preferred solutions by exploring a smaller fraction of all solutions at the expense of more comparisons, which is beneficial when obtaining a new nondominated solution is challenging. Having fewer unique solutions results in smaller size CVMs that can be

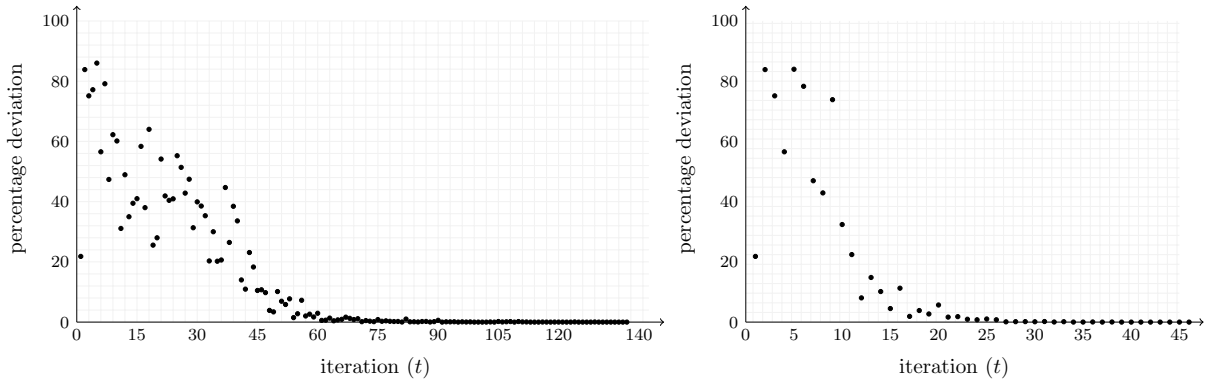
solved more efficiently. This is reflected in time spent on CVMs and on the algorithm as seen in Table 3.2.

The average number of comparisons and the average number of unique solutions follow a linear trend in both pairwise comparison and insertion settings, indicating that IWRA scales well for the linear case. Although the number of comparisons does not increase drastically in instances with many objectives, the DM may decide to stop IWRA before making all necessary comparisons for a provable most preferred solution. We explore the progression of IWRA and evaluate the quality of explored solutions by using the following metric which measures the percentage deviation ($\%\ Dev$) of current nondominated solution x_t^* from the actual most preferred solution $\mathbf{x}^*$ as follows:

$$\frac{(\mathbf{w}^*)^T F(\mathbf{x}^*) - (\mathbf{w}^*)^T F(x_t^*)}{|(\mathbf{w}^*)^T F(\mathbf{x}^*)|}, \tag{3.12}$$

where $\mathbf{w}^*$ is the actual unknown weight vector of the DM. In IWRA, $\mathbf{w}^*$ is the optimal weight vector provided to the sDM at the beginning of the algorithm. Figure 3.2 provides percentage deviations of nondominated solutions identified at each iteration of a sample IWRA-MOLP replication under pairwise comparison and insertion settings for an MOLP-10 instance. In both graphs, the solutions found in the first half of the comparisons converge to a percentage deviation less than 5% and most solutions in the second half have nearly 0% deviation. This indicates that the DM can obtain high quality solutions with half the effort: ~ 70 comparisons in the pairwise setting and ~ 20 unique solutions in insertion which takes approximately 70 to 80 comparisons to sort. The replication

provided in Figure 3.2 is representative of other replications; half of the iterations are spent closing the last $1-2\%$ deviation in most replications.



(a) pairwise comparison setting (b) insertion setting

Figure 3.2. Progression of percentage deviation.

The rest of this subsection compares IWRA-MOLP with ZW method. Similar to IWRA-MOLP, ZW method is an interactive method which iteratively shrinks the remaining weight region. At each iteration, ZW method generates a new solution by using a weight vector which is obtained by finding an arbitrary basic feasible solution of the polytope representing the remaining weight region. The newly generated solution is compared with all of its efficient adjacent solutions in a pairwise manner. In order to decrease the number of such pairwise comparisons, we follow the suggestion of Zionts and Wallenius (1976) and eliminate efficient adjacent solutions whose comparisons yield redundant inequalities for the remaining weight region.

Before discussing the results in detail, we note that the original implementation of ZW method generates trade-offs between the current solution and its efficient adjacent solutions rather than generating the efficient adjacent solutions explicitly. In theory, there is no difference between generating trade-offs and efficient adjacent solutions since each

trade-off is obtained by taking the difference between the objective function values of the current solution and one of its efficient adjacent solutions. The total number of efficient adjacent solutions (trade-offs) in ZW method is equivalent to the number of comparisons since the DM makes a comparison for each efficient adjacent solution (each trade-off).

Table 3.3 summarizes the results for ZW method on the same instances. Different from Table 3.2, Table 3.3 does not report *WGen* since the weight generation time is almost zero for all the replications; instead, Table 3.3 reports the total time spent on generating efficient adjacent solutions, *EAGen*.

Table 3.3. ZW method results in 'avg (stdev)' form.

	Count		Duration (s)			
Problem	Comp	TSoln	Total	WSP	EAGen	$\mathcal{R}$ (avg)
MOLP-2	68 (53)	136 (106)	43 (33)	14 (11)	5 (4)	3.5E-01
MOLP-3	48 (23)	66 (32)	15 (7)	4 (2)	3 (2)	3.4E-03
MOLP-4	77 (21)	93 (26)	15 (5)	3 (1)	5 (2)	2.4E-04
MOLP-5	80 (21)	92 (25)	13 (4)	2 (1)	5 (1)	1.5E-04
MOLP-6	107 (29)	119 (32)	15 (4)	2 (1)	7 (2)	1.8E-04
MOLP-7	122 (23)	132 (24)	15 (3)	2 (0)	8 (2)	1.9E-04
MOLP-8	143 (20)	153 (22)	16 (3)	2 (0)	9 (2)	2.3E-04
MOLP-9	148 (32)	157 (34)	17 (4)	2 (0)	10 (2)	2.3E-04
MOLP-10	184 (25)	194 (26)	22 (4)	2 (0)	13 (2)	3.0E-04

The results of ZW method are compared with those of IWRA-MOLP in pairwise comparison setting (see pairwise comparison columns of Table 3.2) since both methods use pairwise comparisons rather than inserting the solutions to a sorted list. As seen in Table 3.3, ZW method requires significantly more solutions and comparisons compared to IWRA-MOLP. This is an expected result since the comparisons between efficient adjacent solutions are likely to yield less information compared to the comparisons between the diverse solutions generated via CVM in IWRA-MOLP. For example, in a 2-objective setting, ZW method generates all nondominated solutions corresponding to the weight sets along the line segment connecting the initial weight vector to the DM's optimal weight

vector. On the other hand, the total algorithm duration of IWRA-MOLP is significantly larger, mainly due to the time spent on solving CVMs when generating diverse weight vectors. In summary, the large gap between the two methods indicates that diversification efforts in the weight region have significant benefits in reducing the number of comparisons and solutions in interactive approaches at the cost of increasing overall algorithm time. We have also carried out experiments with a slightly modified version of ZW method where the current solution is compared with a subset of efficient adjacent solutions instead of all; however, the results are not reported here as they are not significantly different from those presented in Table 3.3.

3.6.1.3. IWRA-MOIP Runs. Table 3.4 summarizes the results for IWRA-MOIP on MOIP-k instances, $k = 2, 3, 4, 5$. In addition to comparison and total solution count, Table 3.4 includes the number of unique solutions found during the algorithm, *USoln*, as the same solution can be found multiple times. The table also includes the number of times extreme points of the remaining region are checked, *ExtChk*. For extreme point checks, s_{max} is set to the number of objectives. In some replications, IWRA-MOIP reaches the distance precision limit to terminate before checking the extreme points. IWRA-MOIP finds a most preferred solution of the sDM for each instance at each replication in both comparison settings.

In both comparison settings, the number of unique solutions presented to the DM increases with increasing number of objectives; however, the increase in the number of unique solutions is significantly smaller than the increase in the number of nondominated solutions as reflected by $\mathcal{R}$ column. In instances with 4 and 5 objectives, both settings converge after exploring less than 0.3% of all nondominated solutions. Between the two

Table 3.4. IWRA-MOIP results in 'avg (stdev)' form.

		Count				Duration (s)			
	Problem	Comp	USoln	TSoln	ExtChk	Total	WSP	CVM	$\mathcal{R}$ (avg)
Pairwise Comp.	MOIP-2	6 (2)	7 (2)	12 (2)	0.9 (0.5)	3 (2)	3 (2)	0 (0)	1.5E-01
	MOIP-3	16 (3)	17 (3)	29 (7)	1.3 (0.6)	4 (2)	3 (1)	1 (1)	1.3E-02
	MOIP-4	28 (9)	29 (9)	66 (25)	2.9 (1.3)	56 (57)	18 (17)	28 (43)	2.7E-03
Insertion	MOIP-2	13 (4)	7 (1)	10 (1)	0.8 (0.4)	2 (1)	2 (1)	0 (0)	1.3E-01
	MOIP-3	33 (5)	12 (1)	20 (2)	0.9 (0.4)	2 (1)	2 (1)	0 (0)	9.6E-03
	MOIP-4	47 (10)	16 (2)	32 (6)	1.1 (0.3)	11 (8)	7 (8)	2 (1)	1.5E-03
	MOIP-5	69 (10)	21 (2)	52 (8)	1.3 (0.5)	69 (64)	15 (9)	18 (11)	6.8E-04

comparison settings, insertion requires a smaller fraction of all solutions in concordance with the results for unique solution count.

In the pairwise setting of IWRA-MOIP, the number of unique solutions and the number of total solutions increase with objective function count similar to the linear case. Values in *ExtChk* column show that IWRA-MOIP needs a few extreme point checks to converge to a most preferred solution. Both WSP and CVM become significant components in total algorithm times as opposed to linear case where CVM is dominant.

MOIP-5 is not included in the runs with pairwise comparison since the algorithm does not terminate within one hour. Similar to IWRA-MOLP, pairwise comparisons require more nondominated solutions to converge which also increases the number and the size of partial weight sets in the algorithm. Partial sets containing large number of weight vectors require more effort when projecting their convex hull to k-dimensional weight space and slow down the overall algorithm significantly. This issue becomes more visible in instances with large number of objectives, increasing the duration of the overall algorithm significantly.

In insertion, the behavior of IWRA-MOIP is similar to the linear case. Compared to the pairwise setting, the algorithm finds a most preferred solution with fewer nondominated solutions; however, it requires more comparisons. For IWRA-MOIP, converging with fewer solutions is critical since increased number of solutions decreases the efficiency of projections. As seen in Table 3.4, IWRA-MOIP with insertion takes 69 seconds on average to find a most preferred solution of the sDM for MOIP-5 while the pairwise setting cannot converge within an hour. In addition, IWRA-MOIP requires fewer extreme point checks in insertion, yielding a faster convergence. Similar to the pairwise setting, both WSP and CVM are dominant components in the total duration.

The scalability of IWRA-MOIP strongly depends on the effectiveness of the projection of convex hulls to the weight region and extreme point generation. Between projections and extreme point generation, the former requires more effort as the projection of a weight set needs to be reevaluated once a weight vector (which is not an element of existing projection) is added to the weight set. Since the dimension of the weight space increases with the number of objectives, the projections and extreme point generation become more challenging for instances with more objectives. Furthermore, finding 20-30 unique solutions might be challenging when the underlying MOIP is difficult to solve. In this case, approaches designed to carry over information during the iterations of solution procedure may help (Alves and Clímaco, 2000).

3.6.1.4. Imperfect Comparisons. Experiments in the previous subsections assume that the DM makes perfect comparisons; i.e., given two solutions, she always sorts them correctly. This section provides empirical results on the performance of IWRA when the DM's comparisons are imperfect. We compare the performance of IWRA and ZW

method in MOLPs to evaluate the effect of the diversification model in the imperfect setting. Imperfect comparisons are simulated by adding Gaussian noise, $\varepsilon \sim \mathcal{N}(0, \sigma^2)$, to the calculated value function where σ is a preset parameter (López-Ibáñez and Knowles, 2015). For a given solution pair, x and y, in the imperfect comparison setting, sDM compares $(\mathbf{w}^*)^T F(x) + \varepsilon_x$ and $(\mathbf{w}^*)^T F(y) + \varepsilon_y$ instead of $(\mathbf{w}^*)^T F(x)$ and $(\mathbf{w}^*)^T F(y)$ and occasionally makes mistakes. Additional preference non-idealities occur when objective functions are not modeled correctly (Stewart, 1999); however, we assume this is not the case in our setting.

In the experiments, the standard deviation of the error, σ, is set to 5% and 10% of the best value function $(\mathbf{w}^*)^T F(\mathbf{x}^*)$ where $\mathbf{x}^*$ is a most preferred solution obtained by using the optimal weight vector $\mathbf{w}^*$. Table 3.5 reports average percentage deviations as calculated by (3.12) over 25 replications of IWRA-MOLP and ZW method on MOLP-k test instances, and IWRA-MOIP on MOIP-k test instances under both error settings.

Table 3.5. Percentage deviations of IWRA solutions with imperfect comparisons.

			MOLP									MOIP			
	Setting	σ perc	k = 2	k = 3	k = 4	k = 5	k = 6	k = 7	k = 8	k = 9	k = 10	k = 2	k = 3	k = 4	k = 5
IWRA	Pairwise Comp.	5%	0.8	1.3	2.6	3.1	4.3	4.5	5.1	5.2	5.3	1.0	2.0	2.3	-
		10%	2.2	4.6	5.8	6.2	6.9	6.7	7.9	12.6	10.3	2.5	3.4	3.8	-
	Insertion	5%	0.8	2.9	4.2	3.9	7.1	4.1	5.5	6.0	7.7	1.1	1.9	2.9	2.3
		10%	2.2	5.7	7.0	9.3	9.3	7.9	8.6	11.6	15.2	2.8	5.6	5.4	7.3
ZW	-	5%	12.1	20.1	36.3	28.7	21.8	22.3	32.5	20.4	26.0	—			
		10%	12.1	23.3	36.7	28.7	22.8	22.3	32.0	20.4	26.1				

Statistics regarding comparison, unique and total solution counts are not reported in Table 3.5 since the behavior of IWRA algorithms does not change: These counts increase with more objectives and the insertion setting requires fewer unique and total solutions compared to pairwise comparison setting to converge. Interestingly, IWRA algorithms

converge faster as the standard deviation of error increases. As discussed earlier, in the runs with perfect comparisons almost half the search effort is spent eliminating the last $1-5\%$ deviation from the optimal value function. Since the solutions in $1-5\%$ neighborhood have relatively similar objective function values, comparisons tend to be more random during this stage; leading to early terminations.

With IWRA, percentage deviations from the optimal value function increase with increasing σ and more objectives. Percentage deviations are within $10-15\%$ of the most preferred solutions in settings with large number of objectives and relatively large σ values, indicating that IWRA is a successful approach even when comparisons are imperfect. As seen in Table 3.5, the pairwise comparison setting yields better percentage deviations compared to the insertion. This is expected since insertion requires clearer comparisons to insert the solution to the correct position in the list. Comparisons become more difficult as solutions near the correct position are generally similar to the new solution. In the pairwise comparison setting, the new solution is compared to the current best solution, which is relatively easier particularly in the early stages of the algorithm. This indicates that pairwise comparisons yield better results when comparisons are imperfect despite requiring more solutions to be identified.

IWRA outperforms ZW method in finding high-quality solutions. The percentage deviations of IWRA-MOLP increases linearly, reaching $10-15\%$ for MOLP-10 whereas using ZW method increases the deviations rapidly to $25-35\%$ starting from MOLP-3. In these experiments, ZW method prematurely converges to a potentially low-quality solution after a few comparisons as efficient adjacent solutions yield almost identical objective values due to the proximity of underlying weight sets, causing the DM to make incorrect

comparisons more frequently and at the early stages of the algorithm. This indicates that diversification provides significant benefits in percentage deviations when the DM makes imperfect comparisons.

3.6.2. Performance on Benchmark Instances

In a recent study, Lokman et al. (2016) propose a cone-based interactive method which utilizes 2-point cones. These cones are obtained from pairwise comparisons of the DM and are added to the original formulation iteratively, restricting the solution space by eliminating *cone-dominated* solutions. The cone-based method is a more general method which is capable of handling any value function as long as the function is nondecreasing and quasiconcave (Ramesh et al., 1989). This method does not assume a specific value function as the actual value function does not have an impact on the way 2-point cones are defined: preferring solution x to y yields the same cone irrespective of the value function. In IWRA, however, the assumption on the value function determines how the preferences are reflected in the remaining weight region: for WSP setting with a linear value function, the preferences define a set of linear inequalities and for WTP setting with a Tchebycheff value function, the preferences define a union of polytopes (see Section 3.7). Therefore, the results provided in this subsection should be considered as a comparison of how IWRA-MOIP and the cone-based method perform under the assumption that the value function is a weighted sum rather than an overall comparison of both methods.

Lokman et al. (2016) tests the cone-based method on multiobjective assignment problems (MOAP), knapsack problems (MOKP) and shortest path problems (MOSSP) with

varying instance sizes. In our experiments, we use the same generation schemes as Lokman et al. (2016): For MOAP and MOKP instances, we follow the random generation scheme of Özpeynirci and Köksalan (2010); and for MOSPP instances, we follow the random generation scheme of Köksalan and Lokman (2009) on special random graphs (Lokman, 2007). We generate three-objective MOAP instances with sizes 10, 20 and 30 where assignment costs are randomly selected integers from the interval $[1, 20]$; three-objective MOKP instances with 25, 50 and 100 items where profit and weight coefficients are randomly selected integers from the interval $[10, 100]$; and three-objective MOSSP instances on 25, 50, 100 and 200-node graphs where arc costs are selected as integers from the interval $[10, 100]$. Similar to Lokman et al. (2016), ten instances are generated for each problem-size pair and these instances are solved for three optimal weight vectors $w_1^* = (0.7, 0.2, 0.1)$, $w_2^* = (0.1, 0.6, 0.3)$ and $w_3^* = (0.333, 0.333, 0.333)$.

Table 3.6 summarizes the comparisons for weighted sum value function. We use pairwise comparisons for IWRA-MOIP as in the cone-based method. Our computing environment is similar (slightly slower) to that of Lokman et al. (2016); therefore, the results (solution times) are comparable. Table 3.6 provides averages of the number of comparisons until termination (*Comp*), the total time spent by the algorithm (*Time*) and the number comparisons until the best solution is reached (*CuB*) over ten instances for each problem-size pair and weight configuration. The first three columns summarize the results of IWRA-MOIP and columns 4, 5 and 6 are directly taken from Lokman et al. (2016). The last three columns provide the ratio of cone-based method results to IWRA results where values greater than 1 indicate that our findings are better for the corresponding metric and values less than 1 favors Lokman et al. (2016).

Table 3.6. Comparison of IWRA and cone-based method under weighted sum assumption.

			IWRA-MOIP			Cone-based			Ratio		
Problem	Size	Weight	Comp	Time (s)	CuB	Comp	Time (s)	CuB	Comp	Time (s)	CuB
MOAP	10x10	w1	7.5	1.3	4.3	21.5	3.6	4.4	2.9	2.8	1.0
		w2	7.2	0.9	4.5	30.2	7.5	4.4	4.2	8.3	1.0
		w3	9.8	1.8	3.8	41.3	22.4	2.9	4.2	12.4	0.8
	20x20	w1	9.7	2.0	5.5	50.1	53.8	6.9	5.2	26.9	1.3
		w2	11.9	2.3	6.7	50.2	74.6	7.1	4.2	32.4	1.1
		w3	12.8	2.1	7.4	82.0	1296.1	5.9	6.4	617.2	0.8
	30x30	w1	14.8	6.6	7.6	74.1	536.5	8.4	5.0	81.3	1.1
		w2	13.5	5.8	8.3	76.2	621.7	8.8	5.6	107.2	1.1
		w3	14.7	4.4	11.8	113.6	9490.7	5.7	7.7	2157.0	0.5
MOKP	25 items	w1	6.4	1.1	2.4	12.5	1.6	3.0	2.0	1.5	1.3
		w2	7.5	1.1	3.7	17.2	2.7	1.6	2.3	2.5	0.4
		w3	8.8	1.4	1.0	22.6	3.6	2.5	2.6	2.6	2.5
	50 items	w1	7.2	1.3	5.1	32.2	19.2	4.4	4.5	14.8	0.9
		w2	9.1	1.3	5.9	33.9	26.0	3.7	3.7	20.0	0.6
		w3	11.6	2.0	5.2	57.0	139.4	4.8	4.9	69.7	0.9
	100 items	w1	9.7	1.8	5.6	50.4	233.6	5.2	5.2	129.8	0.9
		w2	10.8	2.0	5.5	69.2	723.9	8.0	6.4	362.0	1.5
		w3	14.2	2.4	8.2	93.1	3894.2	6.4	6.6	1622.6	0.8
MOSSP	25 nodes	w1	7.0	1.2	1.3	13.6	1.3	2.6	1.9	1.1	2.0
		w2	6.9	0.8	3.2	19.6	2.4	3.9	2.8	3.0	1.2
		w3	8.0	1.0	3.5	22.7	2.9	0.9	2.8	2.9	0.3
	50 nodes	w1	7.1	1.4	3.3	26.7	12.9	4.1	3.8	9.2	1.2
		w2	8.2	1.3	5.6	28.1	13.4	3.4	3.4	10.3	0.6
		w3	9.1	1.7	5.3	40.2	31.2	4.8	4.4	18.4	0.9
	100 nodes	w1	9.3	9.7	4.9	36.5	83.9	2.7	3.9	8.6	0.6
		w2	8.6	9.1	5.1	43.2	117.9	6.1	5.0	13.0	1.2
		w3	10.0	9.2	6.7	53.5	359.9	4.6	5.4	39.1	0.7
	200 nodes	w1	9.5	162.9	4.8	47.3	626.4	5.1	5.0	3.8	1.1
		w2	9.4	160.6	5.3	48.6	850.3	6.1	5.2	5.3	1.2
		w3	12.7	186.2	4.6	62.2	2957.7	3.6	4.9	15.9	0.8

The results in Table 3.6 show that IWRA-MOIP is able to find a most preferred solution with fewer comparisons, providing around $49 - 87\%$ improvement. The inequalities on the remaining weight region, derived from the comparisons of the DM throughout IWRA-MOIP, restrict larger portions of the objective space compared to 2-point cones in the cone-based method. In terms of total duration, IWRA-MOIP significantly outperforms cone-based method (up to 99% improvement) due to their difference in solution generation phase. At each iteration, IWRA-MOIP uses the new weight vector to form the

aggregated objective function of the WSP and solves the same problem with a different objective while cone-based method appends 2-point cones to the solution space of the original problem making solution generation potentially more difficult in later iterations. Furthermore, IWRA's performance improves more with increasing problem size. Both approaches reach the best solution after a few comparisons, where the cone-based method slightly outperforms IWRA, and spend the remaining iterations to prove optimality.

3.7. Weighted Tchebycheff Problem Setting

As noted earlier, WSP does not necessarily generate all nondominated solutions for nonconvex problems. Thus, we extend our approach using the weight region of augmented Tchebycheff problem which is capable of generating the entire nondominated solution set (Steuer and Choo, 1983). A commonly used weight-based approach to generate nondominated solutions of multiobjective problems is the method of weighted norm (Miettinen, 2008). This approach finds the nondominated solution that is closest to some reference point, α^*, based on the specified weighted norm. Assuming all the objectives are maximized, the method of weighted norm solves the following problem for $1 \leq p < \infty$:

$$\min_{x \in \mathcal{X}} \left(\sum_{i=1}^{k} w_i \Big(\alpha_i^* - f_i(x) \Big)^p \right)^{\frac{1}{p}}, \tag{3.13}$$

where $w \in \mathcal{W}$. Problem (3.13) minimizes the weighted distance of the objective function values from the reference point. Among potential norms, Tchebycheff metric (l_∞-norm) is commonly used since it can be modeled as a single MILP and is capable of generating all nondominated solutions (including unsupported ones) when α^* is set to be a *utopian objective vector* whose entries are the maximum values of each objective function plus a

small nonnegative constant. The weighted Tchebycheff problem (WTP) can be written as follows:

$$\text{(WTP)} \qquad \min_{x \in \mathcal{X}} \max_{i=1,\ldots,k} w_i |\alpha_i^* - f_i(x)|$$

WTP minimizes the maximum weighted deviation from α^*. In this setting, WTP is capable of generating all nondominated solutions and weakly nondominated solutions can be avoided by the augmented Tchebycheff problem (ATP) with a slight modification to WTP:

$$\text{(ATP)} \qquad \min_{x \in \mathcal{X}} \max_{i=1,\ldots,k} \Big(w_i |\alpha_i^* - f_i(x)| \Big) + \rho \sum_{i=1}^{k} (\alpha_i^* - f_i(x))$$

where ρ is a sufficiently small scalar. Constructed this way, ATP can find any nondominated solution with finite trade-off, as we assume here. The absolute value sign can be removed in WTP and ATP since $\alpha_i^* \geq f_i(x)$ for any nondominated solution x in our maximization setting.

3.7.1. IWRA-WTP Decision Phase: Comparison Settings

The decision phase of IWRA-WTP uses the same comparison settings with WSP-based IWRA: Pairwise comparison and insertion. The main difference is the way the preference information is translated into the weight region. In WSP-based algorithms, preferring x to y yields an inequality constraint $w^T F(x) - w^T F(y) \geq 0$. In IWRA-WTP, a more preferred solution yields a lower objective value; therefore, preferring x to y indicates the following:

$$\max_{i=1,...,k} w_i(\alpha_i^* - f_i(x)) \leq \max_{i=1,...,k} w_i(\alpha_i^* - f_i(y)) \tag{3.14}$$

Equivalently, inequality (3.14) can be represented as k inequalities of the following type:

$$w_j(\alpha_j^* - f_j(x)) \leq \max_{i=1,...,k} w_i(\alpha_i^* - f_i(y)), \qquad j \in \{1, ..., k\} \tag{3.15}$$

Since $\max_{i=1,...,k} w_i(\alpha_i^* - f_i(y))$ depends on the weight vector, we can represent inequalities (3.15) with k disjoint cases where case j assumes that $\max_{i=1,...,k} w_i(\alpha_i^* - f_i(y)) = w_j(\alpha_j^* - f_j(y))$; i.e., the maximum weighted distance of the r.h.s. is realized for the j^{th} objective function. Let P_j be the polytope representing the weight vectors satisfying the preference information when the weighted distance of j^{th} objective is maximum for nondominated solution y. P_j can be written as follows:

$$P_j: \quad w_i(\alpha_i^* - f_i(y)) \leq w_j(\alpha_j^* - f_j(y)) \qquad i \in \{1, ..., k\} \tag{3.16}$$

$$w_i(\alpha_i^* - f_i(x)) \leq w_j(\alpha_j^* - f_j(y)) \qquad i \in \{1, ..., k\} \tag{3.17}$$

$$w \in \mathcal{W} \tag{3.18}$$

Inequalities (3.16) indicate that the weighted distance of j^{th} objective of nondominated solution y is the maximum of the weighted distances of the remaining objective functions for the same solution and inequalities (3.17) ensure that this value is greater than or equal to the maximum weighted distance of nondominated solution x. Using P_j polytopes,

preferring x to y can be represented as $\bigcup_{j=1}^{k} P_j$ which is a union of finite number of polytopes (Dell and Karwan, 1990).

For IWRA-WTP, as the DM completes the decision phase of the t^{th} iteration, the remaining weight region, $\mathcal{W}_t$, is constructed with the intersection of $\mathcal{W}_{t-1}$ with $\bigcup_{j=1}^{k} P_j^t$. Since the intersection of two polytope unions is a polytope union, the remaining weight region at a given iteration is always a polytope union (initial weight region, $\mathcal{W}_0$, is a unit $(k-1)$-simplex which is a polytope union with a single element). As the remaining weight regions are represented as polytope unions, they are not necessarily convex for IWRA-WTP.

3.7.2. IWRA-WTP Optimization Stage: Diverse Weight Vector Generation

The optimization stage of IWRA-WTP uses CVM to identify the most diverse weight vector where CVM becomes a disjunctive program since constraint (3.7e) is a polytope union instead of a single polytope. The most diverse weight vector can be found by solving CVM separately for each polytope in the union and taking the weight vector for which the objective is maximum.

3.7.3. IWRA-WTP Implementation

We implement a heuristic algorithm for IWRA-WTP, shown in Algorihtm 3.4, since the weight sets of nondominated solutions are not convex. There are three main differences between IWRA-WTP and WSP-based IWRA algorithms: (i) IWRA-WTP does not construct exact or approximate weight sets and treats the singletons of each explored weight

vector as a separate set, (ii) the weight region updates involve the intersection of polytope unions as opposed to introducing constraints, and (iii) CVM is solved over all the polytopes in the remaining weight region to find the most diverse weight vector. IWRA-WTP can be terminated early if the same solution is repeated for a certain number of consecutive iterations.

Algorithm 3.4 *IWRA-WTP*

Initialize $t = 0$, $s = 0$, $\mathcal{W}_t$ and w_t. Solve WTP for w_t to obtain x_t^*.
Set $x^{**} = x_t^*$. Initialize sets $\mathbf{X} \leftarrow \{x_t^*\}$ and $\mathbf{W} \leftarrow \{w_t\}$.
Solve CVM over $\mathcal{W}_0$ to obtain w_{t+1}, which is at least d_{t+1}^* away from $w \in \mathbf{W}$.
while $d_{t+1}^* > d_{min}$ **do**
  $t \leftarrow t + 1$.
  Solve WTP for w_t to obtain x_t^*. Update $\mathbf{W} \leftarrow \mathbf{W} \cup \{w_t\}$.
  if $x_t^* \in \mathbf{X}$ **then**
    $s \leftarrow s + 1$, $\mathcal{W}_t \leftarrow \mathcal{W}_{t-1}$.
  else
    $s \leftarrow 0$. Update $\mathbf{X} \leftarrow \mathbf{X} \cup \{x_t^*\}$. Ask the DM to provide preference information.
    if x_t^* is preferred to x^{**} **then**
      $x^{**} \leftarrow x_t^*$.
    end if
    Update $\mathcal{W}_{t-1}$ with the DM's preference information to obtain $\mathcal{W}_t$.
  end if
  if $s > s_{max}$ **then**
    break
  end if
  Solve CVM for each polytope in $\mathcal{W}_t$ to obtain w_{t+1}, which is at least d_{t+1}^* away from $w \in \mathbf{W}$.
end while
Return x^{**} as an approximate to a most preferred nondominated solution of the DM.

Algorithm 3.4 starts by initializing the counters and the weight region as unit $(k-1)$-simplex. It finds the first nondominated solution for an arbitrary weight vector, w_0 and initializes $\mathbf{W} = \{w_0\}$. CVM is solved over $\mathcal{W}_0$ to obtain the most diverse weight vector. For the rest of the algorithm, CVM is solved individually for all polytopes within the polytope union representing $\mathcal{W}_t$ to obtain the most diverse weight vector, w_{t+1}, for the next iteration. At each iteration, the nondominated solution, x_t^*, corresponding to w_t is identified and w_t is added to $\mathbf{W}$ as a singleton. If x_t^* is an unexplored solution, it is shown to the DM for comparison, otherwise a new iteration starts. There are two

termination conditions for Algorithm 3.4: (i) no new nondominated solution is found in s_{max} consecutive iterations, where s_{max} is set to the number of objectives, or (ii) CVM's optimal value d^*_{t+1} is not greater than a predetermined minimum distance $d_{min} = 10^{-4}$. When one of these termination conditions hold, the algorithm terminates returning x^{**} as an approximate most preferred solution.

3.7.4. IWRA-WTP Experiments

IWRA-WTP experiments are conducted on test and benchmark instances with an sDM which provides perfect comparisons.

3.7.4.1. Performance on Simulated Test Instances. This section summarizes the results for IWRA-WTP on MOIP-k instances, $k = 2, 3, 4$, for insertion and pairwise comparison settings in Table 3.7. Table 3.7 reports averages and standard deviations of count, duration and performance related metrics over 25 replications. Since IWRA-WTP is a heuristic algorithm, we report average percentage deviation of the resulting solution from a most preferred solution. Letting α^* be the fixed utopian vector, $\mathbf{w}^*$ be the optimal weight vector of sDM, f^* be the objective vector of a most preferred solution of the sDM and f be the objective vector of returned solution, the percentage deviation is calculated as follows:

$$D = 100 \times \frac{max_i\{\mathbf{w}_i^*(\alpha_i^* - f_i)\} - max_i\{\mathbf{w}_i^*(\alpha_i^* - f_i^*)\}}{|max_i\{\mathbf{w}_i^*(\alpha_i^* - f_i)\}|} \tag{3.19}$$

We report the average number of feasible polytopes (*Feas Poly*) in polytope unions throughout the iterations of IWRA-WTP.

Table 3.7. IWRA-WTP results in 'avg (stdev)' form.

		Count				Duration (s)			Performance	
Setting	Problem	Comp	USoln	TSoln	Feas Poly	Total	WTP	CVM	% Dev	$\mathcal{R}$
Pairwise Comp.	MOIP-2	11 (2)	12 (2)	14 (2)	1.0 (0.0)	5 (2)	4 (2)	0 (0)	0.0 (0.0)	2.4E-01
	MOIP-3	21 (7)	22 (7)	26 (7)	3.2 (0.9)	16 (9)	9 (7)	4 (3)	0.1 (0.3)	1.7E-02
	MOIP-4	34 (9)	35 (9)	42 (9)	18.3 (13.7)	335 (281)	81 (96)	120 (102)	0.1 (0.2)	3.3E-03
Insertion	MOIP-2	30 (4)	11 (1)	13 (1)	1.0 (0.0)	4 (2)	3 (1)	0 (0)	0.0 (0.0)	2.2E-01
	MOIP-3	56 (16)	18 (4)	22 (3)	2.2 (0.5)	12 (6)	7 (5)	2 (1)	0.0 (0.1)	1.4E-02
	MOIP-4	77 (26)	22 (6)	29 (7)	4.0 (1.3)	89 (83)	62 (76)	13 (10)	0.2 (0.5)	2.1E-03

As with the WSP results, IWRA-WTP with insertion requires fewer solutions as sDM makes more comparisons each time the algorithm finds a new nondominated solution. Since the comparisons in pairwise setting may not reveal quality information, the number of polytopes in the union representing the remaining weight region increases faster. This slows down the process of finding the next weight vector significantly as we solve CVM for each polytope in the union. IWRA-WTP yields high quality solutions within 2.2% (with an average of 0.2%) of the actual most preferred solution in MOIP-2, MOIP-3 and MOIP-4 instances.

3.7.4.2. Performance on Benchmark Instances. Table 3.8 provides a comparison of IWRA-WTP with the cone-based method of Lokman et al. (2016) under the assumption that the value function is weighted Tchebycheff. Table 3.8 uses the same format with Table 3.6 of Section 3.6.2 in which we report *Comp*, *Time* and *CuB* values over ten instances for each problem-size pair and weight configuration. Similar to the results in Section 3.6.2, IWRA-WTP is capable of finding *a most preferred solution* with fewer comparisons and significantly shorter amount of time under Tchebycheff setting as accumulation of 2-point cones makes solution generation more difficult at later stages of the cone-based method. Note that IWRA-TCH is a heuristic procedure and 17 out of 300 instances are solved

near optimally with 2-3% deviation from the optimal Tchebycheff function value whereas the results from the cone-based method are optimal for all the instances.

Table 3.8. Comparison of IWRA and cone-based method under weighted Tchebycheff metric.

			IWRA-WTP			Cone-based			Ratio		
Problem	Size	Weight	Comp	Time (s)	CuB	Comp	Time (s)	CuB	Comp	Time (s)	CuB
MOAP	10x10	w1	8.8	2.4	4.9	24.1	4.3	5.7	2.7	1.8	1.2
		w2	7.0	1.4	4.6	34.0	7.4	14.1	4.9	5.3	3.1
		w3	13.1	6.4	9.1	42.3	14.0	14.5	3.2	2.2	1.6
	20x20	w1	14.1	6.4	11.2	46.0	42.3	19.9	3.3	6.6	1.8
		w2	13.7	5.4	9.5	48.6	43.2	25.2	3.5	8.0	2.7
		w3	16.5	8.8	13.1	82.6	647.3	29.1	5.0	73.6	2.2
	30x30	w1	20.2	18.1	11.9	61.4	130.2	26.1	3.0	7.2	2.2
		w2	17.5	13.4	14.4	59.8	105.5	28.6	3.4	7.9	2.0
		w3	21.2	28.1	17.7	102.9	4766.6	21.8	4.9	169.6	1.2
MOKP	25 items	w1	7.0	1.9	4.3	13.7	1.6	3.2	2.0	0.8	0.7
		w2	7.8	2.3	4.8	18.1	2.5	5.0	2.3	1.1	1.0
		w3	9.3	5.4	6.7	21.6	3.7	5.8	2.3	0.7	0.9
	50 items	w1	11.0	3.7	6.3	30.0	14.9	11.0	2.7	4.0	1.7
		w2	11.0	3.8	7.2	30.6	13.8	11.6	2.8	3.6	1.6
		w3	13.4	7.2	10.0	59.9	100.6	18.8	4.5	14.0	1.9
	100 items	w1	12.4	3.9	9.7	46.3	116.5	22.2	3.7	29.9	2.3
		w2	12.6	5.9	9.6	63.4	438.1	33.2	5.0	74.3	3.5
		w3	20.3	13.8	17.2	87.6	4938.0	39.6	4.3	357.8	2.3
MOSSP	25 nodes	w1	7.7	2.3	5.3	15.6	1.6	2.8	2.0	0.7	0.5
		w2	8.4	3.7	3.9	17.5	1.8	7.0	2.1	0.5	1.8
		w3	10.1	10.0	5.1	21.0	2.8	5.1	2.1	0.3	1.0
	50 nodes	w1	9.1	3.1	5.4	27.7	13.7	13.7	3.0	4.4	2.5
		w2	8.9	2.9	5.9	25.8	11.6	12.2	2.9	4.0	2.1
		w3	12.1	5.3	8.3	41.7	30.7	13.2	3.4	5.8	1.6
	100 nodes	w1	12.4	15.0	9.4	31.4	58.8	14.8	2.5	3.9	1.6
		w2	10.4	11.6	6.3	35.6	55.5	21.1	3.4	4.8	3.3
		w3	14.3	16.4	10.9	48.4	171.6	18.2	3.4	10.5	1.7
	200 nodes	w1	12.4	167.9	9.2	47.8	530.7	18.0	3.9	3.2	2.0
		w2	13.5	181.2	9.6	39.8	399.1	13.8	2.9	2.2	1.4
		w3	16.5	198.4	13.9	60.5	2999.5	19.7	3.7	15.1	1.4

3.8. Concluding Remarks

In this study, we develop an interactive approach, IWRA, to find the DM's most preferred solution for multiobjective optimization problems. In WSP setting, we develop two finitely converging algorithms, IWRA-MOLP and IWRA-MOIP, for MOLPs and MOIPs, respectively. The numerical experiments show that both algorithms require the DM to

compare a reasonable number of nondominated solutions to find her most preferred solution. The experiments with perfect and imperfect comparisons show that diversification efforts in the weight region provide significant benefits in reducing this number. Furthermore, the number of comparisons increases linearly with the number of objectives. We extend IWRA to WTP setting and provide a heuristic algorithm, IWRA-WTP, which is capable of yielding solutions within $0-3\%$ of the actual most preferred solution.

Further research on diversification models can provide additional benefits for IWRA. Currently, IWRA uses the weight vectors returned by CVM to find the new nondominated solutions. CVM slows down as the algorithms progress, mainly due to increased number of weight sets. Two potential directions are replacing CVM with heuristic models or introducing more efficient diversification models with different distance metrics. A diversification model which does not require projections of convex hulls of partial sets to the weight space can eliminate one of the most critical bottlenecks in IWRA-MOIP enabling it to solve instances with more objectives.

CHAPTER 4

Lock-Free Arc Tour Problem with Complex Objectives in a Multiobjective Setting

This chapter has two main purposes: First, we revisit the LFATP formulations provided in Chapter 2 and discuss various extensions to solve the marathon course design problem for more 'complex' objectives. The term 'complex' refers to sequence- and time-dependent objectives where the objective contribution of an arc cannot be represented in additive format with constant coefficients. Second, we apply IWRA to multiobjective LFATP to design an interactive catalog of solutions that can be presented to race organizers of the Bank of America Chicago Marathon.

In Chapter 2, a standard formulation, *LFATP-S*, and a family of reformulations derived from a disjunctive programming (DP) formulation, *LFATP-P*, have been introduced to solve the LFATP. We refer to these formulations as arc-additive formulations throughout this chapter. The numerical experiments in Section 2.5 show that the reformulations generally outperform the standard formulation in the arc-additive setting. In this chapter, we extend arc-additive formulations to solve sequence- and time-dependent objectives and compare their performances to analyze their strengths and weaknesses further. Between sequence- and time-dependent objectives, extending arc-additive formulations to sequence-dependent objectives is relatively straightforward as it involves introduction of

additional variables and linking constraints. On the other hand, extensions to time-dependent objectives depend on the level of granularity required in terms of timekeeping. When discussing the new formulations, we use the same notations from Chapter 2 unless otherwise is stated.

After developing the extensions for complex objectives, we solve LFATP in a multiobjective setting in order to design a catalog of solutions for the race organizers. We develop two combined formulations, one based on *LFATP-S* and the other based on *LFATP-P*, which are capable of solving problems with all types of objectives and their weighted sum combinations. These combined formulations are used within IWRA-MOIP in generating nondominated solutions based on the weights provided by the algorithm. In the interactive setting, we prepopulate an extensive set of solutions which cover all preference scenarios for multiple iterations instead of interacting with the race organizers at each iteration of IWRA-MOIP. The rest of the chapter is organized as follows: Section 4.1 provides extensions of arc-additive LFATP formulations to sequence- and time-dependent objectives. Section 4.2 tests provided formulations under various objectives. Section 4.3 combines LFATP and IWRA to solve marathon course design problem of BACM in a multiobjective setting. Section 4.4 concludes the chapter.

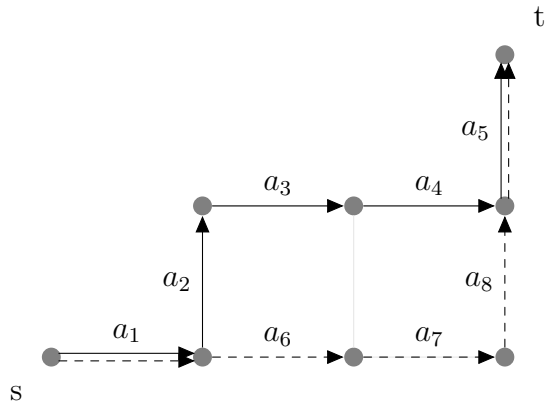
4.1. LFATP Extensions

4.1.1. Extending LFATP Formulations to Sequence-dependent Objectives

In tour finding problems, sequence-dependent objectives measure the contribution resulted from the *sequence* of the arcs used in the solution. The most common examples of such objectives are the turn related ones where the aim is to minimize the total number

of certain type of turns (sharp turns, U-turns, left turns or all turns) and the incline related ones where the aim is to minimize the absolute incline change throughout the route. Different from the arc-additive objectives, sequence-dependent objectives require the knowledge of the *sequence*; i.e., the adjacent arcs in the solution. Figure 4.1 provides a turn detection example by using two paths, $path_1$ (represented by solid arcs) and $path_2$ (represented by dashed arcs), that start from vertex s and end at vertex t. Both paths use arc a_1; however, a_1 leads to a 90 degree turn in $path_1$ due to arc a_2 whereas the same does not hold for $path_2$ due to arc a_6. Thus, each arc's contribution depends on the sequence it creates with arcs that emanate from its terminal vertex.

Figure 4.1. Two sequences with the same length and different number of turns.



Sequence-dependent objectives are generally considered in the scope of shortest route finding problems with turn penalties or turn prohibition. These problems are commonly addressed by three solution approaches (Vanhove and Fack, 2012): (i) line graph transformation, (ii) node splitting and (iii) specialized algorithms. The first two approaches transform the original network so that the resulting problem can be solved by using existing route finding algorithms while the last approach focuses on developing problem

specific algorithms. In the line graph transformation, the vertices of transformed network represent the arcs of the original network and the arcs of the transformed network represent the sequences. For example, arcs (i, j) and (k, l) form a sequence in the original network when $j = k$; therefore, the transformed network includes an arc from the vertex representing (i, j) to the vertex representing (k, l). Examples of line graph transformations can be found in the studies of Caldwell (1961), Anez et al. (1996), Winter (2002) and Fidler and Einhoff (2004). In node splitting, the vertices in the original network are split for each of their incoming and emanating arcs, and the split vertices in the transformed graph are connected to represent the sequences. Examples of node splitting can be found in the studies of Kirby and Potts (1969) and Speičvcys et al. (2003). Between the two transformations, Winter (2002) advises against node splitting as it increases the network size significantly. Apart from node transformations, studies of Ziliaskopoulos and Mahmassani (1996), Boroujerdi and Uhlmann (1998) and Gutiérrez and Medaglia (2008) develop algorithms for shortest path problems with turn restrictions while studies of Clossey et al. (2001), Micó and Soler (2011) and Bräysy et al. (2011) develop methods that are applicable to more general routing problems such as Chinese Postman Problem. See Vanhove and Fack (2012) for a computational comparison of routing algorithms with turn costs and restrictions.

In order to extend the arc-additive LFATP formulations to sequence-dependent objectives, we introduce a new set of variables that mimic line graph transformation. Let v_{ijk} be a binary variable which takes value 1 if arcs $(i, j) \in A$ and $(j, k) \in A$ are used together and let θ_{ijk} be the objective contribution when the arcs (i, j) and (j, k) are used together. The standard sequence-dependent formulation *LFATP-S-SD* can be written as follows:

$$\text{(4.1a)} \qquad \min \sum_{\substack{i,j,k \\ (i,j),(j,k)\in A}} \theta_{ijk} v_{ijk}$$

s.t.

$$\text{(4.1b)} \qquad v_{ijk} \geq x_{ij} + x_{jk} - 1 \qquad (i,j),(j,k) \in A,$$

$$\text{(4.1c)} \qquad v_{ijk} \leq x_{ij} \qquad (i,j),(j,k) \in A,$$

$$\text{(4.1d)} \qquad v_{ijk} \leq x_{jk} \qquad (i,j),(j,k) \in A,$$

$$\text{(4.1e)} \qquad 0 \leq v_{ijk} \leq 1 \qquad (i,j),(j,k) \in A,$$

$$(2.1b) - (2.1m).$$

LFATP-S-SD is obtained by adding constraints (4.1b) - (4.1e) to the feasible region of arc-additive standard formulation, *LFATP-S*, which is defined by constraints (2.1b) - (2.1m). Constraints (4.1b) - (4.1e) ensure that v_{ijk} takes value 1 when arcs (i,j) and (j,k) are used together. Although we do not transform the underlying network explicitly, considering v_{ijk} as an *arc* that connects *node transformations* of arcs (i,j) and (j,k), these constraints mimic line graph transformation. Note that the binary restriction on v_{ijk} variables are relaxed in constraints (4.1e) as v_{ijk} variables can only take values 0 or 1 for any combination of x_{ij} and x_{jk} variables. Furthermore, when $\theta_{ijk} \geq 0$ for all $(i,j),(j,k) \in A$, constraints (4.1c) and (4.1d) become redundant. These constraints ensure that v_{ijk} is set to 0 when any one of x_{ij} or x_{jk} variables is 0; i.e., arcs (i,j) and (j,k) are not used together in the solution. However, with $\theta_{ijk} \geq 0$, the problem sets the value of v_{ijk} variables to 0 due to the minimization setting, making constraints (4.1c) and

(4.1d) redundant. Note that when $\theta_{ijk} = 0$ and these redundant constraints are ignored, the optimal value of v_{ijk} can be set to 1 even if both arcs (i, j) and (j, k) are not used together; however, this does not affect the optimization problem as the corresponding coefficient $\theta_{ijk} = 0$.

Similar to *LFATP-S-SD*, sequence-dependent extension of disjunctive programming formulation, *LFATP-P-SD*, can be written as follows:

$$\text{(4.2a)} \qquad \min \sum_{\substack{i,j,k \\ (i,j),(j,k)\in A}} \theta_{ijk} v_{ijk}$$

s.t.

$$\text{(4.2b)} \qquad v_{ijk} \geq \sum_{m=1}^{|M|} \bar{x}_{ijm} + \sum_{m=1}^{|M|} \bar{x}_{jkm} - 1 \qquad (i,j) \in \bar{A}, (j,k) \in \bar{A},$$

$$\text{(4.2c)} \qquad v_{ijk} \geq z_{ij} + \sum_{m=1}^{|M|} \bar{x}_{jkm} - 1 \qquad (i,j) \in A \setminus \bar{A}, (j,k) \in \bar{A},$$

$$\text{(4.2d)} \qquad v_{ijk} \geq \sum_{m=1}^{|M|} \bar{x}_{ijm} + z_{jk} - 1 \qquad (i,j) \in \bar{A}, (j,k) \in A \setminus \bar{A},$$

$$\text{(4.2e)} \qquad v_{ijk} \geq z_{ij} + z_{jk} - 1 \qquad (i,j) \in A \setminus \bar{A}, (j,k) \in A \setminus \bar{A},$$

$$\text{(4.2f)} \qquad v_{ijk} \leq \sum_{m=1}^{|M|} \bar{x}_{ijm} \qquad (i,j) \in \bar{A}, (j,k) \in A,$$

$$\text{(4.2g)} \qquad v_{ijk} \leq z_{ij} \qquad (i,j) \in A \setminus \bar{A}, (j,k) \in A,$$

$$\text{(4.2h)} \qquad v_{ijk} \leq \sum_{m=1}^{|M|} \bar{x}_{jkm} \qquad (i,j) \in A, (j,k) \in \bar{A},$$

$$\text{(4.2i)} \qquad v_{ijk} \leq z_{jk} \qquad (i,j) \in A, (j,k) \in A \setminus \bar{A},$$

$$(4.1e), (2.2b) - (2.2m),$$

$$(2.1g) - (2.1i), (2.1m), (2.1l),$$

where constraints (4.2b)-(4.2i) replace x_{ij} variables with $\sum_{m=1}^{|M|} \bar{x}_{ijm}$ when $(i,j) \in \bar{A}$ and with z_{ij} when $(i,j) \in A \setminus \bar{A}$ in constraints (4.1b) - (4.1d). Similar to the standard formulation case, these constraints and constraints (4.1e) are added to the feasible region of *LFATP-P* which is defined by constraints (2.2b)-(2.2m), (2.1g)-(2.1i), (2.1m) and (2.1l).

4.1.2. Extending LFATP Formulations to Time-dependent Objectives

Finishing a marathon generally takes around 2 hours for elite runners and 4 to 7 hours for runners with average to low speeds. As marathons span over a long period of time, the objective coefficients for certain objective functions can differ significantly throughout the race. Considering the medical proximity objective mentioned in Chapter 2, a decision maker may prefer to minimize the distance to medical facilities at the later stages of the race rather than the earlier stages since the likelihood of hospital transfers is significantly higher at the later stages of the race (Kim et al., 2012). In this case, the objective contribution of each arc depends on 'when' the arc is visited, which we refer to as time-dependent objectives.

With recent developments in technology and availability of historic data, time-dependent routing problems have gained increasing interest over the last few decades. While time-dependence has been considered in a wide range of routing problems such as traveling salesman problem and vehicle routing problem (see Picard and Queyranne (1978), Fox et al. (1980), Malandraki and Daskin (1992) for early examples), we briefly summarize

time-dependence from orienteering problem (OP) and arc routing problem (ARP) perspective as these problems are more relevant to the LFATP setting. We refer the reader to the recent review paper by Gendreau et al. (2015) for a broader discussion on time-dependent routing problems.

Time-dependent problems can be divided into four categories based on the type of dependency on travel times and profit (or cost) functions. The first category is time-windowed problems with constant travel times and profits. In these problems, the vertices in the network must be visited during a specific time window, otherwise, the system incurs a penalty (can be infinity when the time window is strict). While time tracking is necessary in these problems, they are weakly time-dependent since system attributes do not change over time. Kantor and Rosenwein (1992), Vansteenwegen et al. (2009), Labadie et al. (2011) and Gunawan et al. (2017) develop heuristics to solve OP and team OP (TOP) with time windows. The second category consists of studies with time-dependent travel times and constant profit functions. In these studies, the time spent when traversing an arc depends on the arrival/departure times or periods. Li et al. (2010) and Li (2011) use time discretization to keep track of time-dependent travel times when solving OPs and TOPs. Similarly Gunawan et al. (2014) use multiple time periods with constant travel times where travel time is a constant that depends on the period during which the arc is traversed. Verbeeck et al. (2014) and Verbeeck et al. (2017) assume piecewise linear travel times with varying slopes and intercepts. Mei et al. (2016) consider a general travel time function for their multiobjective OP and solve the resulting nonlinear model with evolutionary algorithms. The studies of Garcia et al. (2010) and Gavalas et al. (2015) combine the first two categories to solve TOPs with time-dependent

travel times and time windows. The third category includes studies with constant travel times and time-dependent profits. The studies of Erkut and Zhang (1996) and Ekici and Retharekar (2013) utilize time-dependent profit functions which decrease linearly with time when solving OP and TOP, respectively. Tagmouti et al. (2007) and Tagmouti et al. (2011) consider piecewise linear service costs in the scope of standard and dynamic capacitated ARPs. The study of Erdoğan and Laporte (2013) can also be considered in this category as it assumes that the collected profit depends not on the arrival time but the amount of time spent on a vertex. The last category combines second and third categories with time-dependent travel times and time-dependent profits. The study of Jossé et al. (2016) is the only example in this category where the authors develop an approximate dynamic programming algorithm to solve twofold time-dependent arc OP. In terms of solution approaches, most time-dependent studies focus on developing efficient heuristics, evolutionary algorithms or approximation schemes since exact solution approaches do not scale for networks with more than a few hundred vertices.

In our setting, a time-dependent objective replaces the arc-additive objective coefficient c_{ij} with a time-dependent cost function $c_{ij}(\tau_{ij})$ where τ_{ij} represents the arrival time to arc $(i,j) \in A$ (or departure time from arc $(i,j) \in A$). Without loss of generality, we assume that τ_{ij} represents the arrival time to arc (i,j) for the rest of the discussion. Thus, time-dependent formulations must (i) keep track of arrival times and (ii) effectively calculate the cost function value at a given time. In the next subsections, we first discuss common time tracking and cost calculation practices which can be utilized to extend arc-additive LFATP formulations to time-dependent ones. As the formulations obtained via these common practices are found to be remarkably difficult to solve, we take a less

granular approach in time tracking and cost calculation to develop a relatively simplified LFATP formulation for time-dependent objectives.

4.1.2.1. Time Tracking. Before discussing the details of time tracking, we must note that time tracking discussions in this section assume 'no-waiting' due to the underlying marathon setting. In some routing problems, the moving entity is allowed to wait before traveling to the next destination in the network; i.e., the arrival time to the next destination must be greater than or equal to the arrival time to the current destination plus the travel time. Considering our marathon setting where the runners do not wait between destinations, we enforce the arrival time to the next destination to be equal to the arrival time to the current destination plus the travel time, leaving no room for waiting.

For general travel time functions on the arcs of the network, the resulting time tracking constraints involve nonlinearities which further complicate the underlying optimization problem. Therefore, most studies in the literature assume that travel times are either constant or linear functions of arrival time. Since linear travel time functions cover constant travel times as well, we assume that the travel time through arc (i, j) is a linear function of the arrival time to (i, j) and it is represented by the following expression:

$$\zeta_{ij}\tau_{ij} + \eta_{ij} \qquad (i,j) \in A. \tag{4.3}$$

Given linear travel time functions as in (4.3), time tracking constraints for the routing problems can be written in two different ways: (i) node-based time tracking and (ii) arc-based time tracking. In the node-based time tracking, the arrival time is tracked by s_j variables where $s_j > 0$ is a continuous variable representing the time of arrival to vertex j. Assuming that the route starts from vertex $0 \in V$ at start time $\bar{s}$, the arrival times

can be tracked with the following constraints:

$$s_0 = \bar{s}, \tag{4.4}$$

$$-M(1-x_{ij}) + \zeta_{ij}s_i + \eta_{ij} \leq s_j \leq M(1-x_{ij}) + \zeta_{ij}s_i + \eta_{ij} \qquad (i, j \neq 0) \in A. \tag{4.5}$$

where M is a large number. Time balance constraints (4.5) ensure that $s_j = \zeta_{ij}s_i + \eta_{ij}$ when $x_{ij} = 1$; i.e., when arc $(i, j) \in A$ is traversed.

In the arc-based time tracking, the arrival time is tracked by w_{jk} variables where $w_{jk} > 0$ is a continuous variable representing the time of arrival to vertex j when arc (j, k) is traversed. Again, assuming that the route starts from vertex $0 \in V$ at start time $\bar{s}$, the arrival times can be tracked as follows:

$$w_{0k} = \bar{s}x_{0k} \qquad (0, k) \in A, \tag{4.6}$$

$$\sum_{k:(k,j)\in A} \left(w_{kj} + \zeta_{kj}w_{kj} + \eta_{kj}x_{kj}\right) = \sum_{k:(j,k)\in A} w_{jk} \qquad j \neq 0 \in V, \tag{4.7}$$

$$w_{jk} \leq Mx_{jk} \qquad (j, k) \in A. \tag{4.8}$$

In constraints (4.7), s_j variables are replaced with $\sum_k w_{jk}$ variables while constraints (4.8) ensure that $w_{jk} = 0$ when arc $(j, k) \in A$ is not visited. Different from node-based time tracking, variable restrictions (Big-M restrictions) are not embedded in time balance constraints as they are specified separately in constraints (4.8).

4.1.2.2. Cost Calculation. Similar to time tracking, the cost calculation strongly depends on the properties of the cost function. As general cost functions lead to nonlinearities, most studies in the literature assume that cost functions are either linear functions

or continuous piecewise linear functions of arrival time. When discussing the details of cost calculations, we use node-based time tracking from Section 4.1.2.1 with s_j variables satisfying constraints (4.4) and (4.5); however, the results can be generalized to arc-based time tracking as well.

Letting the objective function of the underlying routing problem be the following:

$$\min \sum_{(i,j)\in A} c_{ij}(t)x_{ij}, \tag{4.9}$$

where $c_{ij}(t)$ is a linear function defined as:

$$c_{ij}(t) = \pi_{ij}t + \rho_{ij} \qquad (i,j) \in A, \tag{4.10}$$

and assuming that $c_{ij}(t) \geq 0$ for $t \in [0, T_{max}]$ where T_{max} is the upper limit on time, the cost contribution of arc (i,j), f_{ij}, can be calculated by using the following constraints:

$$f_{ij} \geq -M(1 - x_{ij}) + \pi_{ij}s_i + \rho_{ij} \qquad (i,j) \in A, \tag{4.11}$$

$$f_{ij} \geq 0, \tag{4.12}$$

where M is a large number. When $x_{ij} = 1$; i.e., arc (i,j) is used in the solution, constraint (4.11) indicates that $f_{ij} \geq \pi_{ij}s_i + \rho_{ij}$ and f_{ij} is set to $\pi_{ij}s_i + \rho_{ij}$ due to minimization setting. Otherwise, this constraint becomes redundant. With these constraints, the time-dependent objective function (4.9) can be replaced by an arc-additive function $\sum_{(i,j)\in A} f_{ij}$.

When $c_{ij}(t)$ is a continuous piecewise linear function with $n_{max} \in \mathbb{Z}^+$ time intervals where the start and end times for interval $n \in \{1, 2, ..., n_{max}\}$, are T_{n-1} and T_n respectively,

$c_{ij}(t)$ can be written as follows:

$$c_{ij}(t) = \Big\{ \pi_{ij}^n t + \rho_{ij}^n, \qquad t \in [T_{n-1}, T_n],\ n \in \{1, 2, ..., n_{max}\},\ (i,j) \in A, \tag{4.13}$$

where π_{ij}^n and ρ_{ij}^n are interval-dependent coefficients. In this case, if $c_{ij}(t)$ is also convex for each $(i,j) \in A$, cost calculation can be achieved by repeating constraints (4.11) over each time interval as the value of a convex piecewise linear function is the pointwise maximum of the linear functions representing the pieces:

$$f_{ij} \geq -M(1 - x_{ij}) + \pi_{ij}^n s_i + \rho_{ij}^n \qquad (i,j) \in A, n \in \{1, 2, ..., n_{max}\}, \tag{4.14}$$

(4.12).

For a general continuous piecewise linear function, the time interval during which arc (i,j) is visited needs to be tracked for accurate cost calculation. Thus, x_{ij} variables are replaced with $\tilde{x}_{ij}^n$ variables where $\tilde{x}_{ij}^n$ takes value 1 when arc (i,j) is traversed during time interval n. With $x_{ij} = \sum_{n=1}^{n_{max}} \tilde{x}_{ij}^n$, the cost contribution can be calculated as follows:

$$\sum_{(i,j) \in A} \sum_{n=1}^{n_{max}} T_{n-1} \tilde{x}_{ij}^n \leq s_i \leq \sum_{(i,j) \in A} \sum_{n=1}^{n_{max}} T_n \tilde{x}_{ij}^n \qquad i \in V, \tag{4.15}$$

$$f_{ij} \geq -M(1 - \tilde{x}_{ij}^n) + \pi_{ij}^n s_i + \rho_{ij}^n \qquad (i,j) \in A, n \in \{1, 2, ..., n_{max}\}, \tag{4.16}$$

(4.12).

Constraints (4.15) link s_i and $\tilde{x}_{ij}^n$ variables to ensure $T_{n-1} \leq s_i \leq T_n$ is satisfied when (i,j) is traversed during interval n and constraints (4.16) become binding if and only if arc (i,j) is traversed during interval n. Note that constraints (4.12), (4.15) and (4.16)

can be used in piecewise functions with jump discontinuities such as step functions when the function value is equal to the minimum of the one-sided limits at the point of jump discontinuity.

4.1.2.3. Simplified LFATP Formulation for Time-Dependent Objectives. We have obtained various time-dependent formulations by using the time tracking and cost calculation constraints discussed in Sections 4.1.2.1 and 4.1.2.2; however, these formulations are found to be expectedly difficult to solve even in relatively small instances. For this reason, we take a less granular approach and introduce 'segment-based' formulations which imitate the time-dependent behavior. In these formulations, the route is divided into segments of specific lengths and the objective contribution of each arc depends on the segment it is used in. For example, a marathon course can be divided into four equal segments, representing quarters of the race, and each arc can be assigned four cost coefficients where n^{th} coefficient specifies the objective contribution of that arc when it is used during n^{th} quarter of the race. This way, the time notion is embedded to length-based segments which do not require detailed time tracking at the cost of losing granularity; however, this level of granularity is appropriate in this setting as marathon related objectives do not require precise time tracking.

Let $\tilde{c}_{ij}^{n}$ be the objective contribution of arc $(i,j) \in A$ if (i,j) is used within segment $n \in \{1,2,...,n_{max}\}$ where n_{max} specifies the number of segments. Let $\mathcal{L}_{min}^{n}$ and $\mathcal{L}_{max}^{n}$ be the minimum and maximum segment lengths for segment n. Similar to tour length, specifying a fixed segment length leads to infeasibilities in most cases; therefore, segment length is specified by interval $[\mathcal{L}_{min}^{n}, \mathcal{L}_{max}^{n}]$. Letting $\tilde{x}_{ij}^{n}$ be a binary variable which takes value 1 when (i,j) is used within segment n, the standard time-dependent formulation,

LFATP-S-TD, can be written as follows:

$$\text{(4.17a)} \qquad \min \sum_{(i,j)\in A} \sum_{n=1}^{n_{max}} \tilde{c}_{ij}^{n} \tilde{x}_{ij}^{n}$$

s.t.

$$\text{(4.17b)} \qquad x_{ij} = \sum_{n=1}^{n_{max}} \tilde{x}_{ij}^{n} \qquad (i,j) \in A,$$

$$\text{(4.17c)} \qquad \mathcal{L}_{min}^{n} \leq \sum_{(i,j)\in A} d_{ij} \tilde{x}_{ij}^{n} \leq \mathcal{L}_{max}^{n} \qquad n \in \{1,2,...,n_{max}\},$$

$$\text{(4.17d)} \qquad \sum_{j:(j,i)\in A} \tilde{x}_{ji}^{1} \geq \sum_{j:(i,j)\in A} \tilde{x}_{ij}^{1} \qquad i \in V \setminus \{0\},$$

$$\text{(4.17e)} \qquad \sum_{j:(j,i)\in A} \left(\tilde{x}_{ji}^{n-1} + \tilde{x}_{ji}^{n}\right) \geq \sum_{j:(i,j)\in A} \tilde{x}_{ij}^{n} \qquad i \in V \setminus \{0\},\ n \in \{2,3,...,n_{max}\},$$

$$\text{(4.17f)} \qquad \tilde{x}_{ij}^{n} \in \{0,1\} \qquad (i,j) \in A,\ n \in \{1,2,...,n_{max}\},$$

$$(2.1b) - (2.1m).$$

LFATP-S-TD is obtained by adding constraints (4.17b) - (4.17f) to the feasible region of arc-additive standard formulation, *LFATP-S*, which is defined by constraints (2.1b) - (2.1m). Constraints (4.17b) link the original routing variables x_{ij} to $\tilde{x}_{ij}^{n}$. Note that these constraints are introduced for notational convenience as x_{ij} variables in constraints (2.1b) - (2.1m) can be replaced by $\sum_{n=1}^{n_{max}} \tilde{x}_{ij}^{n}$ and removed from the current formulation. Constraints (4.17c) ensure that the length restrictions are satisfied for each segment. Constraints (4.17d) and (4.17e) guarantee segment balance; i.e., if the incoming arc to a vertex (except the start vertex 0) belongs to segment n, then emanating arc must belong to either segment n or $n+1$.

Similar to *LFATP-S-TD*, *LFATP-P-TD* can be written as follows:

$$\text{(4.18a)} \qquad \min \sum_{(i,j)\in A} \sum_{n=1}^{n_{max}} \tilde{c}_{ij}^{n} \tilde{x}_{ij}^{n}$$

s.t.

$$\text{(4.18b)} \qquad \sum_{m=1}^{|M|} \bar{x}_{ijm} = \sum_{n=1}^{n_{max}} \tilde{x}_{ij}^{n} \qquad (i,j) \in \bar{A},$$

$$\text{(4.18c)} \qquad z_{ij} = \sum_{n=1}^{n_{max}} \tilde{x}_{ij}^{n} \qquad (i,j) \in A \setminus \bar{A},$$

$$(4.17c) - (4.17f), (2.2b) - (2.2m),$$

$$(2.1g) - (2.1i), (2.1m), (2.1l),$$

where constraints (4.17b) are replaced by constraints (4.18b) and (4.18c) to link $\bar{x}_{ijm}$ and z_{ij} variables to $\tilde{x}_{ij}^{n}$ variables. The remaining segment constraints (4.17c)-(4.17f) stay the same and are directly taken from *LFATP-S-TD*. Similar to the standard formulation case, these constraints are added to the feasible region of *LFATP-P* which is defined by constraints (2.2b)-(2.2m), (2.1g)-(2.1i), (2.1m) and (2.1l).

4.2. Numerical Experiments

This section compares standard and DP formulations under arc-additive, sequence-dependent and time-dependent objective functions for marathon course design. When solving DP formulations, we use *LFATP-R2* reformulation which utilizes 2-edge subsets of must visit edges in M. While the formulations have already been tested in Chapter 2 for some of the arc-additive objective functions, we rerun those experiments in this chapter due to changes in computational settings. The experiments are conducted on a computer

with a 2.30 GHz CPU and 16 GB memory running under 64-bit Windows 8 operating system. MILP models are solved by using Gurobi 7.5 (Gurobi Optimization, 2018) in C# environment of Microsoft Visual Studio 2010. In all experiments, formulations are given a 2-hour time limit.

The experimental setting in this section is the same as that of the case study in Section 2.5.3. The formulations are applied to BACM instance with 7 must-visit edges and use the network generated from City of Chicago street centerline map to include all streets whose widths are greater than 30 feet (City of Chicago, 2013). In Sections 4.2.2, 4.2.3 and 4.2.4, each formulation and objective pair is solved with an additional $K\%$ similarity constraint which ensures that the total length of the common arcs, used in the current solution and in the original course, must be at least $K\%$ of the original course length. In this case, 100% similarity becomes the original course and 0% similarity becomes the unrestricted case from similarity perspective. When solving for different similarity levels, we start from the highest similarity level and solve for the instances in a descending order of similarity levels. This way, the optimal/near optimal solution of the current instance can be used as a starting solution for the subsequent instance. Details of the BACM case and similarity discussions can be found in Section 2.5.3.

4.2.1. Objective Functions for Marathon Course Design

In the numerical experiments, we consider several objective functions related to marathon course design to compare the standard and DP formulations under various settings.

- **Average medical proximity** is an arc-additive objective function which aims to shorten travel times for hospital transports by minimizing the average distance from a

point on the course to a nearest hospital. Letting m_{ij} be the rectilinear distance (in feet) from the midpoint of arc (i, j) to the closest hospital, the objective contribution of each arc is constructed as:

$$c_{ij} = \frac{m_{ij} d_{ij}}{L} \qquad (i, j) \in A. \tag{4.19}$$

where c_{ij} represents the length-weighted contribution of arc (i, j) to average medical proximity. Note that this objective function has been explored in Chapter 2 in detail; however, the experiments are rerun due to changes in the computational setting.

• **Average transport proximity** is an arc-additive objective function which aims to maximize accessibility of the course from spectator perspective by minimizing the average distance from a point on the course to the nearest public transport station. The objective contribution of arc (i, j) is calculated similar to that of average medical proximity objective:

$$c_{ij} = \frac{t_{ij} d_{ij}}{L} \qquad (i, j) \in A. \tag{4.20}$$

where t_{ij} is the rectilinear distance (in feet) from the midpoint of arc (i, j) to the closest public transport station. c_{ij} represents the length-weighted contribution of arc (i, j) to average transport proximity.

• **Average street width** is an arc-additive objective function which aims to reduce runner congestion by maximizing the average width of the road segments used in the course. Similar to the previous arc-additive objectives, the objective contribution of arc

(i, j) is calculated as:

$$c_{ij} = \frac{w_{ij} d_{ij}}{L} \qquad (i, j) \in A. \tag{4.21}$$

where w_{ij} is the width of arc (i, j) (in feet). c_{ij} represents the length-weighted contribution of arc (i, j) to average street width.

• **Total turn angle** is a sequence-dependent objective function which aims to increase runner speed and decrease runner congestion by minimizing total turn angle along the course. Letting $\beta_{ij}^{\angle}$ be the total turn angle (in degree) within arc (i, j) and $\theta_{ijk}^{\angle}$ be the turn angle (in degree) when moving from arc (i, j) to (j, k), the total turn angle of a route can be written as follows:

$$\sum_{(i,j)\in A} \beta_{ij}^{\angle} x_{ij} + \sum_{\substack{i,j,k \\ (i,j),(j,k)\in A}} \theta_{ijk}^{\angle} v_{ijk} \tag{4.22}$$

Note that $\beta_{ij}^{\angle}$ coefficients are needed since the road segments are not necessarily straight lines and preprocessing operations may merge some road segments into a single arc.

• **Total sharpness count** is a sequence-dependent objective function which aims to increase runner speed and decrease runner congestion by minimizing the sharp turns along the course. For this purpose, each turn in the network is assigned an integer sharpness

count. At a given point:

$$\text{sharpness count} = \begin{cases} 0, & 0^\circ \leq \text{turn angle} < 30^\circ \\ 1, & 30^\circ \leq \text{turn angle} < 60^\circ \\ 2, & 60^\circ \leq \text{turn angle} < 120^\circ \\ 3, & 120^\circ \leq \text{turn angle} \leq 180^\circ \end{cases}$$

Letting $\bar{\beta}_{ij}$ be the total sharpness count within arc (i, j) and $\bar{\theta}_{ijk}$ be the sharpness count when moving from arc (i, j) to (j, k), the total sharpness count of a route can be written as follows:

$$\sum_{(i,j)\in A} \bar{\beta}_{ij} x_{ij} + \sum_{\substack{i,j,k \\ (i,j),(j,k)\in A}} \bar{\theta}_{ijk} v_{ijk} \tag{4.23}$$

- **Time-dependent average medical proximity** is a time-dependent objective function which aims to minimize average medical proximity of a marathon course based on cardiac arrest incidence rates. The study of Kim et al. (2012) investigates outcomes of 59 cardiac arrest incidents during marathon / half-marathon races in the United States between 2000 and 2010, and their study indicates that 10 out of 59 cardiac arrests occurred within the first two quarters of the race while the remaining 49 occurred within the third and last quarters. Thus we divide the marathon course into two equal segments and update the c_{ij} values from equation (4.19) with incident rates of $\frac{10}{59}$ and $\frac{49}{59}$ for the first and second segment, respectively. This objective function favors courses whose later stages are closer to the hospitals.

• **Time-dependent average street width** is a time-dependent objective function which aims to maximize average street width in order to minimize early stage congestion. The runner density on the streets decreases with time as the spread of runner speeds leads to a greater geographic dispersion of runners; therefore, maximizing average street width at the earlier stages is more important. For this reason, the course is divided into two equal segments and c_{ij} values from equation (4.21) are updated in such a way that the width contribution of an arc in the first segment is twice as much as its width contribution in the second segment.

4.2.2. Numerical Experiments for Arc-additive Objectives

Tables 4.1, 4.2 and 4.3 summarize the *LFATP-S* and *LFATP-P* results at similarity levels $0\%, 10\%, ..., 90\%, 100\%$ for average medical proximity, average transport proximity and average street width objectives, respectively. The tables report the number of subtour elimination inequalities (*SBTR*) and VDLEI inequalities (*VDLEI*), time spent on cut generation (*CutGen*) and solving the instance (*Total*), optimality gap (*% Gap*) and the objective value for the optimal/near-optimal solution for each formulation and similarity pair. Figure 4.2 shows the locations of hospitals and public transport stations along with the current marathon course. For public transport stations, we use Chicago Transit Authority's train stops data which consists of 300 station locations, obtained from City of Chicago (2017).

For all objectives, instances with high similarity levels ($70 - 100\%$) are easier to solve as they are more restricted compared to instances with lower similarity levels. For average medical proximity, the formulations fail to prove optimality when similarity level is below

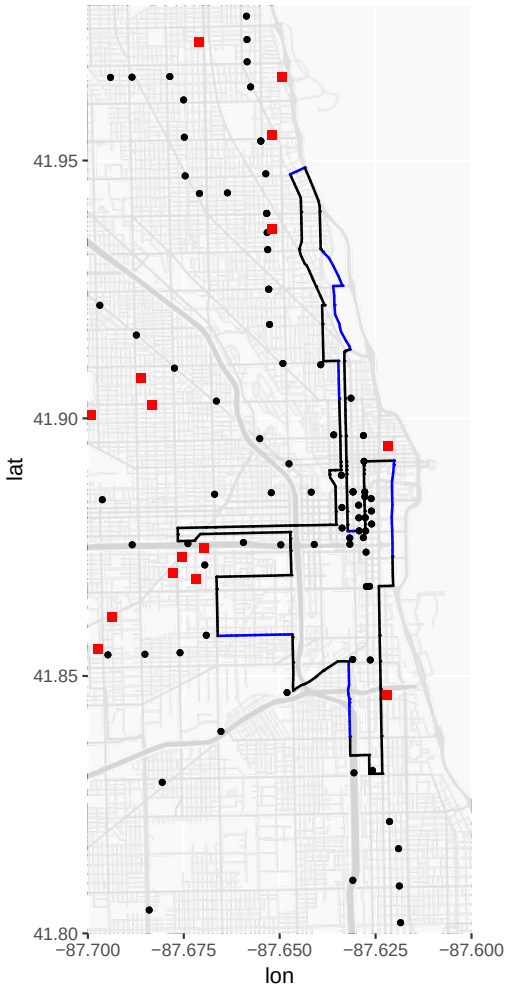


Figure 4.2. Locations of hospitals (red squares) and public transport stations (black circles) along with the current BACM course.

$70 - 80\%$. The same behavior is observed for average transport proximity and average street width objectives when similarity levels are below 40% and 70%, respectively. Similar to the results of the BACM case study in Section 2.5.3, changing around 30% of the course is sufficient to achieve more than $65 - 70\%$ of maximum improvement.

For average medical and transport proximity objectives, *LFATP-P* performs significantly better than *LFATP-S* in terms of optimality gaps. While both formulations fail

Table 4.1. Results of *LFATP-S* and *LFATP-P* for average medical proximity objective.

	LFATP-S						**LFATP-P**					
Sim.	**Count**		**Duration (s)**		**Performance**		**Count**		**Duration (s)**		**Performance**	
%	*SBTR*	*VDLEI*	*CutGen*	*Total*	*% Gap*	*ObjVal*	*SBTR*	*VDLEI*	*CutGen*	*Total*	*% Gap*	*ObjVal*
100	0	0	0.0	0.1	0.0	5928	0	0	0.0	1.7	0.0	5928
90	1076	17	29.4	112.2	0.0	5620	176	18	4.8	44.0	0.0	5620
80	1162	79	60.8	259.4	0.0	5295	336	51	13.1	135.4	0.0	5295
70	5012	1061	452.4	TILIM	2.6	5141	1305	430	100.3	1574.8	0.0	5132
60	8244	1549	410.8	TILIM	4.1	5017	3535	2237	236.0	TILIM	0.5	4964
50	9407	1070	371.7	TILIM	6.7	4929	2995	2581	325.7	TILIM	0.7	4855
40	6284	2219	397.3	TILIM	6.3	4853	3498	3815	257.5	TILIM	1.1	4853
30	6420	1323	580.9	TILIM	6.1	4853	3381	3858	272.0	TILIM	1.1	4853
20	4676	1664	470.6	TILIM	6.5	4853	3550	2641	293.1	TILIM	1.3	4853
10	4622	1643	460.6	TILIM	6.5	4853	3151	3782	264.8	TILIM	1.1	4853
0	4528	1619	445.6	TILIM	6.5	4853	3225	4464	246.2	TILIM	1.1	4853

Table 4.2. Results of *LFATP-S* and *LFATP-P* for average transport proximity objective.

	LFATP-S						**LFATP-P**					
Sim.	**Count**		**Duration (s)**		**Performance**		**Count**		**Duration (s)**		**Performance**	
%	*SBTR*	*VDLEI*	*CutGen*	*Total*	*% Gap*	*ObjVal*	*SBTR*	*VDLEI*	*CutGen*	*Total*	*% Gap*	*ObjVal*
100	0	0	0.0	0.1	0.0	2669	0	0	0.0	1.8	0.0	2669
90	254	4	4.7	21.7	0.0	2511	68	3	1.7	17.0	0.0	2511
80	301	11	4.3	32.5	0.0	2314	121	13	2.7	31.3	0.0	2314
70	1571	26	47.3	246.0	0.0	2196	156	12	4.2	28.5	0.0	2196
60	2678	52	87.2	637.1	0.0	2119	66	9	1.5	12.7	0.0	2119
50	9148	140	352.5	4266.3	0.0	2074	195	68	11.0	65.4	0.0	2074
40	9958	294	390.7	TILIM	3.2	2060	2803	415	297.2	TILIM	1.0	2053
30	13445	282	384.8	TILIM	2.6	2050	3201	387	337.3	TILIM	1.1	2043
20	11699	330	598.5	TILIM	2.9	2050	4142	365	302.3	TILIM	1.1	2043
10	11616	325	594.1	TILIM	2.9	2050	5298	183	291.4	TILIM	1.2	2043
0	11473	318	580.5	TILIM	2.9	2050	4373	474	440.3	TILIM	1.0	2043

to prove optimality at low similarity levels, optimality gaps of *LFATP-P* are around $0.5 - 1.2\%$ whereas they are around $2.5 - 6.5\%$ for *LFATP-S*, indicating that *LFATP-P* is a better choice for these two objectives. As expected, *LFATP-P* uses fewer subtour elimination inequalities than *LFATP-S*; however, the former uses more VDLEIs than the latter. In terms of cut generation times, *LFATP-P* spends less time when searching for

Table 4.3. Results of *LFATP-S* and *LFATP-P* for average street width objective.

	LFATP-S						**LFATP-P**					
Sim.	**Count**		**Duration (s)**		**Performance**		**Count**		**Duration (s)**		**Performance**	
%	*SBTR*	*VDLEI*	*CutGen*	*Total*	*% Gap*	*ObjVal*	*SBTR*	*VDLEI*	*CutGen*	*Total*	*% Gap*	*ObjVal*
100	0	0	0.0	0.2	0.0	47.9	0	0	0.0	1.9	0.0	47.9
90	1278	21	35.1	144.1	0.0	49.7	1320	21	45.2	621.3	0.0	49.7
80	856	12	32.9	150.2	0.0	51.6	1426	26	53.8	869.1	0.0	51.6
70	6630	846	420.5	TILIM	0.5	53.2	4557	226	219.6	TILIM	1.4	53.1
60	6696	616	468.6	TILIM	0.9	54.9	5482	306	244.7	TILIM	3.2	54.1
50	3031	1668	458.9	TILIM	0.9	56.1	4196	592	239.7	TILIM	1.3	56.1
40	2989	1709	429.6	TILIM	1.0	56.1	3610	644	329.7	TILIM	1.3	56.1
30	3254	1512	490.2	TILIM	1.1	56.1	3852	594	345.6	TILIM	1.2	56.1
20	3432	1646	511.9	TILIM	1.1	56.1	3502	480	297.9	TILIM	1.2	56.1
10	3369	1613	500.8	TILIM	1.1	56.1	2256	1155	272.7	TILIM	1.1	56.1
0	3263	1555	486.6	TILIM	1.1	56.1	3322	898	341.3	TILIM	1.1	56.1

the cuts but the difference is not significant especially in instances for which the time limit is reached.

For average street width objective, *LFATP-S* becomes competitive with faster solution times for instances that are solved to optimality and slightly better optimality gaps for instances that reach the time limit. The main difference between proximity objectives and average street width objective is the structure of arc costs. In proximity objectives, the aim is to be as close as possible to a predetermined set of vertices in the network; therefore, the preferable arcs are clustered around those vertices, resulting in excessive subtour formation. In such cases, *LFATP-P* is expected to perform better as it leaves a small length budget for subtour formation. In width maximization, on the other hand, wide streets are stretched along the entire map and these streets are not clustered. Thus, *LFATP-P* loses its benefit and standard formulation performs relatively better.

4.2.3. Numerical Experiments for Sequence-dependent Objectives

Tables 4.4 and 4.5 summarize the *LFATP-S-SD* and *LFATP-P-SD* results at similarity levels $0\%, 10\%, ..., 90\%, 100\%$ for total turn angle and total sharpness count objective, respectively. For total turn angle objective, *LFATP-S-SD* performs better than *LFATP-P-SD* as it solves each instance significantly faster. *LFATP-P-SD* fails to solve 60% similarity instance to optimality with a 3.3% gap and terminates at a sub-optimal solution. The results for total sharpness count objective are similar: Both formulations solve each instance to optimality while the standard formulation is significantly faster.

Table 4.4. Results of *LFATP-S-SD* and *LFATP-P-SD* for total turn angle objective.

	LFATP-S-SD						**LFATP-P-SD**					
Sim.	**Count**		**Duration (s)**		**Performance**		**Count**		**Duration (s)**		**Performance**	
%	*SBTR*	*VDLEI*	*CutGen*	*Total*	*% Gap*	*ObjVal*	*SBTR*	*VDLEI*	*CutGen*	*Total*	*% Gap*	*ObjVal*
100	0	0	0.0	0.1	0.0	3691	0	0	0.0	1.8	0.0	3691
90	14	0	1.5	35.7	0.0	3194	12	3	4.5	156.3	0.0	3194
80	81	6	7.5	201.5	0.0	2920	46	9	12.0	830.6	0.0	2920
70	262	50	44.1	986.1	0.0	2708	272	45	36.1	4229.3	0.0	2708
60	304	92	38.8	906.9	0.0	2454	344	113	67.6	TILIM	3.3	2503
50	87	24	11.4	245.9	0.0	2322	77	32	19.9	2111.5	0.0	2322
40	92	26	13.0	321.7	0.0	2314	39	20	11.7	1166.6	0.0	2314
30	94	31	15.4	363.2	0.0	2314	53	33	33.8	2993.1	0.0	2314
20	109	27	15.2	318.2	0.0	2314	37	30	13.2	1361.4	0.0	2314
10	109	27	15.1	316.1	0.0	2314	60	44	31.6	2692.3	0.0	2314
0	109	27	14.8	313.8	0.0	2314	40	25	16.6	1916.4	0.0	2314

Despite having an additional set of (v_{ijk}) variables and additional sets of constraints, sequence-dependent formulations are found to be easier to solve than arc-additive ones for turn related objectives as these objectives implicitly discourage subtour formation. In a standard routing problem, relaxations generally favor subtour formation as subtours tend to decrease total travel distances or increase collected profit. This is not the case in a turn minimization setting as subtours introduce more turns, worsening the objective

Table 4.5. Results of *LFATP-S-SD* and *LFATP-P-SD* for total sharpness count objective.

	LFATP-S-SD						**LFATP-P-SD**					
Sim.	**Count**		**Duration (s)**		**Performance**		**Count**		**Duration (s)**		**Performance**	
%	*SBTR*	*VDLEI*	*CutGen*	*Total*	*% Gap*	*ObjVal*	*SBTR*	*VDLEI*	*CutGen*	*Total*	*% Gap*	*ObjVal*
100	0	0	0.0	0.1	0.0	70	0	0	0.0	1.9	0.0	70
90	8	0	0.8	20.2	0.0	60	6	1	2.6	97.7	0.0	60
80	20	1	1.6	76.8	0.0	52	9	4	5.1	395.0	0.0	52
70	62	13	9.2	308.8	0.0	49	75	12	19.1	1953.9	0.0	49
60	99	13	12.4	307.8	0.0	45	94	23	20.9	1935.6	0.0	45
50	39	11	3.1	108.4	0.0	43	5	7	7.7	1007.1	0.0	43
40	217	50	34.7	971.3	0.0	43	48	14	12.1	1355.0	0.0	43
30	241	55	35.0	914.3	0.0	43	15	22	14.6	1085.6	0.0	43
20	240	64	39.6	1116.1	0.0	43	27	27	14.3	1753.7	0.0	43
10	240	64	40.1	1120.5	0.0	43	72	34	20.6	1972.1	0.0	43
0	240	64	39.8	1127.1	0.0	43	18	14	12.1	1286.3	0.0	43

value. Thus, sequence-dependent formulations with turn related objectives do not require extensive efforts to eliminate subtours (as seen in *SBTR* columns of Tables 4.4 and 4.5) and converge to optimal solutions faster.

4.2.4. Numerical Experiments for Time-dependent Objectives

Tables 4.6 and 4.7 summarize the *LFATP-S-TD* and *LFATP-P-TD* results at similarity levels $0\%, 10\%, ..., 90\%, 100\%$ for time-dependent average medical proximity and average street width objectives, respectively. As the results in Tables 4.6 and 4.7 indicate, time-dependent objectives are the most difficult to optimize even with the simplified segment-based formulations. For time-dependent average medical proximity objective, *LFATP-S-TD* reaches up to 30% optimality gaps while *LFATP-P-TD* reaches up to 14%. This is an expected result since the underlying ordering logic behind DP formulation reflects the time notion better. While *LFATP-P-TD* is significantly better in reducing optimality gaps, the objective values for low similarity levels are slightly worse than those of the

standard formulation. The results for time-dependent average street width objective are very similar to those of medical proximity; however, the optimality gaps for both formulations are significantly better.

Table 4.6. Results of *LFATP-S-TD* and *LFATP-P-TD* for time-dependent average medical proximity objective.

	LFATP-S-TD						**LFATP-P-TD**					
Sim.	**Count**		**Duration (s)**		**Performance**		**Count**		**Duration (s)**		**Performance**	
%	*SBTR*	*VDLEI*	*CutGen*	*Total*	*% Gap*	*ObjVal*	*SBTR*	*VDLEI*	*CutGen*	*Total*	*% Gap*	*ObjVal*
100	0	0	0.0	2.1	0.0	5641	0	0	0.0	4.5	0.0	5641
90	3135	21	122.4	3680.2	0.0	5145	622	14	25.9	1360.5	0.0	5145
80	3917	76	227.2	TILIM	6.7	4729	2050	190	112.0	TILIM	4.2	4729
70	3696	151	156.9	TILIM	18.4	4629	1451	220	135.1	TILIM	7.4	4599
60	4248	243	153.4	TILIM	23.6	4573	297	84	97.4	TILIM	9.4	4554
50	3972	379	128.8	TILIM	29.1	4529	272	108	102.4	TILIM	12.9	4534
40	4124	349	133.4	TILIM	29.6	4481	371	61	88.3	TILIM	13.7	4534
30	5021	508	172.2	TILIM	29.0	4481	412	124	145.8	TILIM	13.1	4501
20	4220	260	177.2	TILIM	30.5	4481	402	95	103.9	TILIM	13.1	4501
10	4149	258	170.0	TILIM	30.5	4481	293	87	141.1	TILIM	11.0	4501
0	4042	257	164.7	TILIM	30.5	4481	320	53	85.0	TILIM	13.8	4501

Table 4.7. Results of *LFATP-S-TD* and *LFATP-P-TD* for time-dependent average street width objective.

	LFATP-S-TD						**LFATP-P-TD**					
Sim.	**Count**		**Duration (s)**		**Performance**		**Count**		**Duration (s)**		**Performance**	
%	*SBTR*	*VDLEI*	*CutGen*	*Total*	*% Gap*	*ObjVal*	*SBTR*	*VDLEI*	*CutGen*	*Total*	*% Gap*	*ObjVal*
100	0	0	0.0	2.2	0.0	48.4	0	0	0.0	10.2	0.0	48.4
90	2867	26	140.0	TILIM	2.0	49.6	1830	15	88.7	TILIM	2.3	49.6
80	4716	46	207.8	TILIM	4.9	51.2	1384	18	117.8	TILIM	4.1	51.1
70	4940	94	176.5	TILIM	6.6	52.6	1590	83	118.1	TILIM	3.3	53.6
60	3623	189	164.6	TILIM	5.5	54.2	1972	105	111.3	TILIM	4.0	54.7
50	2763	205	136.3	TILIM	6.8	54.6	1179	154	93.4	TILIM	3.0	55.9
40	2944	240	154.6	TILIM	7.8	54.6	854	101	83.3	TILIM	3.0	56.1
30	2447	216	161.3	TILIM	4.9	56.2	1005	103	92.3	TILIM	2.9	56.1
20	1900	224	170.5	TILIM	4.6	56.2	1011	58	87.6	TILIM	2.7	56.1
10	1892	224	168.2	TILIM	4.6	56.2	1321	82	108.7	TILIM	2.8	56.1
0	1882	223	165.8	TILIM	4.6	56.2	1195	126	98.2	TILIM	2.2	56.1

Among all the objectives tested in this section, time-dependent average medical proximity objective is the most difficult one as it combines the time-dependence with the clustered cost structure. To improve optimality gaps, we used the DP-based B&C method

to solve *LFATP-P* with higher fixed depth levels as visit orders with more must-visit edges provide a better understanding of *when* the route visits certain parts of the map. However, this approach could not improve the optimality gaps for most of the instances.

In summary, performances of *LFATP-P* and *LFATP-S* depend strongly on the degree of subtour formation imposed by the underlying objective function. When the subtour formation is encouraged, as in average medical proximity, average transport proximity and time-dependent average medical proximity objectives, *LFATP-P* outperforms *LFATP-S* since path-based approach exploits length restrictions to reduce the length budget for subtour formation. When subtour formation is not necessarily encouraged, as in average street width and time-dependent average street width objectives, the performances of *LFATP-P* and *LFATP-S* do not differ significantly. When subtour formation is discouraged, as in total turn angle and total sharpness degree objectives, *LFATP-S* outperforms *LFATP-P* since the latter loses its strength when subtours are not prevalent.

4.3. Case Study - Multiobjective LFATP for Bank of America Chicago Marathon

This section combines our findings from Chapters 2, 3 and 4 to generate a catalog of guided solution sets for the Bank of America Chicago Marathon. The solution sets, representing a wide range of marathon course designs, are generated by using IWRA-MOIP on the multiobjective version of LFATP. Instead of conducting real-time interactive sessions with race organizers in which new constraints are added through comparisons at each iteration of IWRA-MOIP, we generate diverse solution sets that cover all possible

outcomes of the first four IWRA iterations. The rest of the section is organized as follows: Section 4.3.1 introduces two combined formulations, standard and DP, which are capable of solving LFATP for any combination of arc-additive, sequence-dependent and time-dependent objective functions. Section 4.3.2 provides the details of objective functions used in the case study. Section 4.3.3 discusses the application of IWRA-MOIP to multiobjective LFATP and evaluates the generated solutions.

4.3.1. Combined Formulations

In order to solve LFATP in a multiobjective setting, a *combined formulation* which is capable of finding optimal solutions for any combination of arc-additive, sequence-dependent and time-dependent objectives is needed. This section provides two such combined formulations, *LFATP-S-C* and *LFATP-P-C*, based on standard and DP formulations, respectively. *LFATP-S-C* and *LFATP-P-C* are comprehensive formulations that integrate all variables and constraints from arc-additive, sequence-dependent and time-dependent formulations.

Without loss of generality, let c_{ij}, $\tilde{c}^n_{ij}$ and θ_{ijk} be the weighted sum objective coefficients of standard routing (x_{ij}), sequence (v_{ijk}) and segment-based routing ($\tilde{x}^n_{ij}$) variables, respectively, for a weight vector $\mathbf{w}$. Then the combined standard formulation, *LFATP-S-C*, can be written as follows:

$$\text{(4.24a)} \qquad \min \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{\substack{i,j,k:\\ (i,j),(j,k)\in A}} \theta_{ijk}v_{ijk} + \sum_{n=1}^{n_{max}} \sum_{(i,j)\in A} \tilde{c}^n_{ij}\tilde{x}^n_{ij}$$

s.t.

$$(2.1b)-(2.1m),(4.1b)-(4.1e),(4.17b)-(4.17f).$$

Constraints (2.1b) - (2.1m) define the feasible region of *LFATP-S* formulation for arc-additive objectives, and constraints (4.1b) - (4.1e) and (4.17b)-(4.17f) are added for sequence- and time-dependent extensions, respectively.

Similar to *LFATP-S-C*, combined DP formulation, *LFATP-P-C*, can be written as follows:

$$\text{(4.25a)} \quad \min \sum_{m=1}^{|M|} \sum_{(i,j)\in\bar{A}} c_{ij}\bar{x}_{ijm} + \sum_{(i,j)\in A\setminus\bar{A}} c_{ij}z_{ij} + \sum_{\substack{i,j,k:\\(i,j),(j,k)\in A}} \theta_{ijk}v_{ijk} + \sum_{n=1}^{n_{max}} \sum_{(i,j)\in A} \tilde{c}_{ij}^{n}\tilde{x}_{ij}^{n}$$

s.t.

$$(2.2b)-(2.2m),(2.1g)-(2.1i),(2.1m),(2.1l),$$

$$(4.2b)-(4.2i),(4.1e),$$

$$(4.18b),(4.18c),(4.17c)-(4.17f).$$

Constraints (2.2b)-(2.2m), (2.1g)-(2.1i), (2.1m),(2.1l) define the feasible region of *LFATP-P* formulation for arc-additive objectives, and constraints (4.2b)-(4.2i), (4.1e) and (4.18b), (4.18c), (4.17c)-(4.17f) are added for sequence- and time-dependent extensions, respectively.

4.3.2. Objective Functions

In the case study, we focus on solving 4-objective LFATP which optimizes: (i) average medical proximity, (ii) average transport proximity, (iii) similarity and (iv) total sharpness

count objectives. Different from Section 4.2, similarity to the existing course is no longer a constraint but an arc-additive objective function to be maximized: the higher the similarity, the easier to implement the new solution. Thus, the multiobjective setting includes three arc-additive and one sequence-dependent objective function. Note that time-dependent objective functions are not included in this case study considering the results discussed in Section 4.2.4. Street width related objectives are not included either since the BACM organizers are able to control runner density (and congestion) with staggered starts by corral.

From the results in Section 4.2, the optimal/best objective values for average medical proximity, average transport proximity, similarity and total sharpness count objectives are 4853 feet, 2050 feet, 100% and 43, respectively. Among these four objectives, similarity is the only maximization-type objective, therefore, its coefficients are multiplied by -1 to make all objectives minimization-type. Furthermore, to have a more uniform weight representation in the weight space, the objective functions are normalized with their optimal values; i.e., their coefficients are divided by their individual optimal values and the LFATP is solved for these normalized values. For example, if the normalized LFATP yields the objective value vector $(1.10, 1.12, -0.73, 2.21)$ for a given weight vector $\mathbf{w}$, then the actual objective value vector is:

$$(1.10 \times 4853, 1.12 \times 2050, -0.73 \times 100 \times (-1), 2.21 \times 43) = (5338, 2296, 73, 95).$$

4.3.3. Multi-Iteration Comparison Set Generation with IWRA-MOLP

As discussed in Chapter 3, IWRA-MOIP iterates between solution generation and comparison phases. During the solution generation phase, we first generate diverse weight vectors via constraint violation model (see Section 3.3.2) and solve the resulting weighted sum problem with an aggregated objective function to find a potentially diverse nondominated solution. Note that this case study ignores unsupported nondominated solutions as LFATP is not a convex problem and IWRA-MOIP is not capable of finding unsupported solutions. Letting weight vector $\mathbf{w} = (w_1, w_2, w_3, w_4)$ represent the weights corresponding to normalized average medical proximity, average transport proximity, similarity and total sharpness count objectives, respectively, we use the following logic when solving the resulting weighted sum problem: When $w_1 + w_2 \geq w_3 + w_4$, we assume that the average proximity objectives are the dominant structure in the resulting weighted sum problem; therefore, the problem is solved by using *LFATP-P-C* as the DP formulation performs better for average proximity objectives (see Section 4.2.2). Otherwise, we use *LFATP-S-C* as the standard formulation performs better for similarity objective and sequence-dependent objectives (see Section 4.2.3).

During the comparison phase, the decision maker (DM) is presented a set of solutions and asked to compare these solutions. Chapter 3 discusses two comparison methods: (i) pairwise comparison of the new solution with the current best and (ii) inserting the new solution to the preference-sorted list of already identified solutions. After the comparison phase, the weight region is updated based on the preference information of the decision maker and a new iteration starts with solution generation phase. In this case study, we have generated all possible comparison sets for the first four iterations of IWRA. This

way, a single interaction with the DM (the race organizers in this case) enables us to iterate IWRA for multiple iterations, reducing the number of interactions at the cost of enumerating the solutions for all possible scenarios.

4.3.3.1. Comparison Set Generation. In our setting, comparison sets consist of four LFATP solutions and the DM is asked to choose the best solution of a given comparison set. Based on her selection, a new set is shown to DM and the process is repeated for four iterations. In order to be able to provide these sets offline (without solving LFATP in real time), all preference scenarios are considered and the subsequent sets are generated accordingly. Let $\mathcal{S}_\kappa$ represent solution set κ and $\ell_{\kappa i}$ be the i^{th} solution in $\mathcal{S}_\kappa$. The initial set $\mathcal{S}$ (without κ index) consists of four solutions ℓ_1, ℓ_2, ℓ_3 and ℓ_4, which are constructed by minimizing each objective individually; i.e., by using weight vectors $(1,0,0,0)$, $(0,1,0,0)$, $(0,0,1,0)$ and $(0,0,0,1)$ – corners of unit 3-simplex. At the end of iteration 1, there are four possible scenarios where the DM selects ℓ_i, $i = 1,2,3,4$, as the most preferred solution. For the second iteration, we generate four solution sets $\mathcal{S}_1 = \{\ell_{11}, \ell_{12}, \ell_{13}, \ell_{14}\}$, $\mathcal{S}_2 = \{\ell_{21}, \ell_{22}, \ell_{23}, \ell_{24}\}$, $\mathcal{S}_3 = \{\ell_{31}, \ell_{32}, \ell_{33}, \ell_{34}\}$ and $\mathcal{S}_4 = \{\ell_{41}, \ell_{42}, \ell_{43}, \ell_{44}\}$ where $\mathcal{S}_i$ assumes that solution ℓ_i is selected as the most preferred solution from comparison set $\mathcal{S}$. Therefore, the weight region of set $\mathcal{S}_i$ satisfies $w^T F(\ell_i) \leq w^T F(\ell_j)$, $j = 1,2,3,4$, $j \neq i$, where $F(\ell)$ is the objective value vector of the normalized LFATP for solution ℓ. The first solution at each $\mathcal{S}_i$ is set to ℓ_i, $(\ell_{i1} \leftarrow \ell_1)$, then, three additional solutions ℓ_{i2}, ℓ_{i3} and ℓ_{i4} are generated by using weight vectors obtained via the constraint violation model using the weight region of $\mathcal{S}_i$. The same procedure is repeated once more for 16 scenarios to generate sets $\mathcal{S}_{ij} = \{s_{ij1}, s_{ij2}, s_{ij3}, s_{ij4}\}$, $1 \leq i,j \leq 4$, and finally for 64 scenarios to generate sets $\mathcal{S}_{ijk} = \{s_{ijk1}, s_{ijk2}, s_{ijk3}, s_{ijk4}\}$, $1 \leq i,j,k \leq 4$. For example, in a scenario where

the DM stops with a final solution s_{1413} at the end of the fourth iteration, the iteration progression for that DM is as follows: In iteration 1, the DM chooses the first solution s_1 from set $\mathcal{S}$ and is directed to comparison set $\mathcal{S}_1$. In iteration 2, the DM chooses the fourth solution s_{14} from set $\mathcal{S}_1$ and is directed to comparison set $\mathcal{S}_{14}$. In iteration 3, the DM chooses the first solution s_{141} from set $\mathcal{S}_{14}$ and is directed to comparison set $\mathcal{S}_{141}$. Lastly, in iteration 4, the DM chooses the third solution s_{1413} from set $\mathcal{S}_{141}$, which becomes her most preferred solution at the end of four IWRA-MOIP iterations. See Figure 4.3 for a schematic of comparison sets and the iteration progress for solution s_{1413}.

The process of generating all comparison sets requires solving LFATP 256 times. For this reason, we limit the maximum solution time to 60 minutes; however, terminate the optimization if optimality gap is below 1.5% and more than 20 minutes is spent already. The solutions are generated in 76 hours with an average optimality gap to 0.60% and maximum optimality gap of 2.53%. Out of 256 problems, 112 are solved to optimality, 131 are terminated as they reach 1.5% optimality gap between minute 20 and 60 (time limit), and 13 is terminated due to time limit of 60 minutes. Note that some solutions are found multiple times; therefore, the total number of unique solutions is 140. Table 4.8 provides the details of solutions for sets $\mathcal{S}$, $\mathcal{S}_1$, $\mathcal{S}_2$, $\mathcal{S}_3$ and $\mathcal{S}_4$ (see Appendix B.1 for a list of all sets and solutions). Figure 4.4 provides the comparison page for set $\mathcal{S}$.

4.3.3.2. Quality of the Comparison Sets. The quality of the comparison sets are tested by using 50 random weight vectors. For each randomly generated weight vector, we first calculate the actual most preferred solution, $\mathbf{x}^*$ (see Appendix B.2 for details). Then, the same weight vector is provided to a simulated DM which completes four IWRA-MOIP iterations by using generated comparison sets and converging to best IWRA solution,

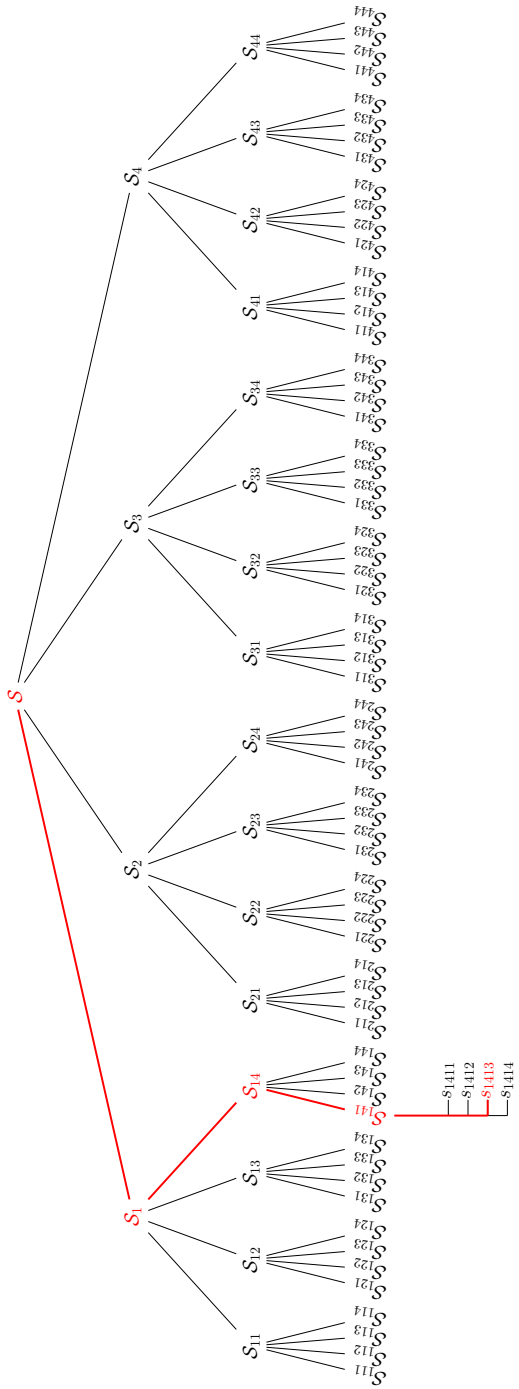
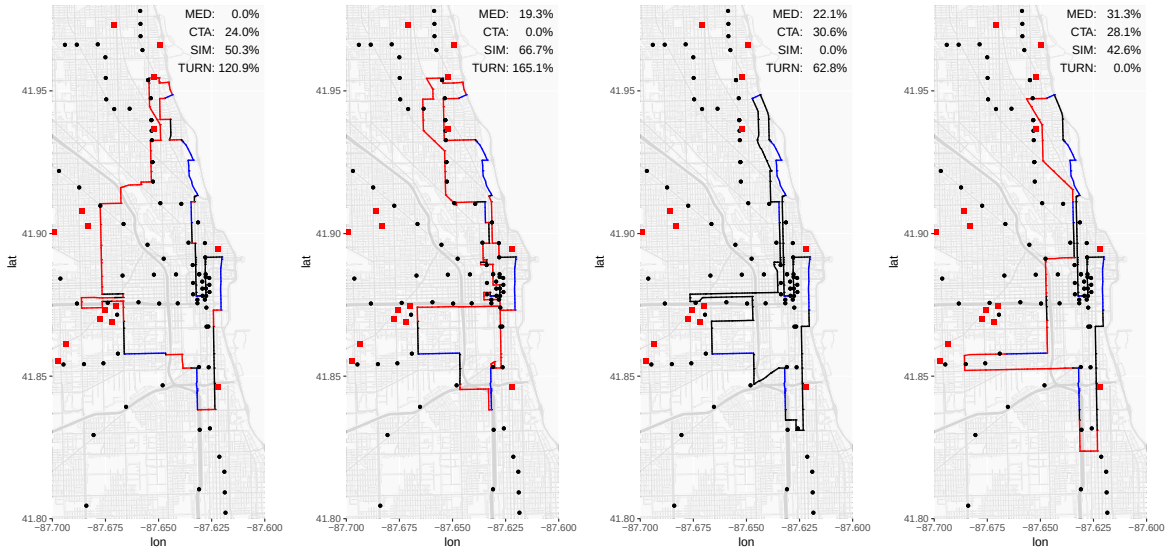


Figure 4.3. Comparison sets and the iteration progression for solution s_{1413}.

Table 4.8. List of LFATP solutions for sets $\mathcal{S}$, $\mathcal{S}_1$, $\mathcal{S}_2$, $\mathcal{S}_3$ and $\mathcal{S}_4$.

Set	**Solution**	**Weight**	**Objective Values**
$\mathcal{S}$	ℓ_1	(1.0000, 0.0000, 0.0000, 0.0000)	(4852.7, 2534.1, 49.7, 95)
	ℓ_2	(0.0000, 1.0000, 0.0000, 0.0000)	(5787.6, 2043.1, 33.3, 114)
	ℓ_3	(0.0000, 0.0000, 1.0000, 0.0000)	(5927.5, 2669.3, 100.0, 70)
	ℓ_4	(0.0000, 0.0000, 0.0000, 1.0000)	(6370.3, 2617.1, 57.4, 43)
$\mathcal{S}_1$	ℓ_{11}	(1.0000, 0.0000, 0.0000, 0.0000)	(4852.7, 2534.1, 49.7, 95)
	ℓ_{12}	(0.3889, 0.3889, 0.2222, 0.0000)	(5334.6, 2260.2, 72.3, 95)
	ℓ_{13}	(0.6578, 0.0467, 0.2956, 0.0000)	(5295.1, 2568.1, 80.2, 88)
	ℓ_{14}	(0.7044, 0.2956, 0.0000, 0.0000)	(4982.3, 2272.1, 41.1, 97)
$\mathcal{S}_2$	ℓ_{21}	(0.0000, 1.0000, 0.0000, 0.0000)	(5787.6, 2043.1, 33.3, 114)
	ℓ_{22}	(0.3561, 0.4306, 0.2133, 0.0000)	(5334.6, 2260.2, 72.3, 95)
	ℓ_{23}	(0.0240, 0.6680, 0.3080, 0.0000)	(5947.5, 2164.4, 67.7, 95)
	ℓ_{24}	(0.2847, 0.7153, 0.0000, 0.0000)	(5450.0, 2066.2, 33.8, 108)
$\mathcal{S}_3$	ℓ_{31}	(0.0000, 0.0000, 1.0000, 0.0000)	(5927.5, 2669.3, 100.0, 70)
	ℓ_{32}	(0.2643, 0.2643, 0.2643, 0.2070)	(5417.8, 2502.3, 64.5, 49)
	ℓ_{33}	(0.0000, 0.3678, 0.6322, 0.0000)	(5927.5, 2669.3, 100.0, 70)
	ℓ_{34}	(0.1136, 0.6322, 0.1771, 0.0771)	(5741.6, 2261.5, 75.1, 63)
$\mathcal{S}_4$	ℓ_{41}	(0.0000, 0.0000, 0.0000, 1.0000)	(6370.3, 2617.1, 57.4, 43)
	ℓ_{42}	(0.2500, 0.2500, 0.2500, 0.2500)	(5417.8, 2502.3, 64.5, 49)
	ℓ_{43}	(0.2505, 0.6250, 0.0000, 0.1245)	(5344.7, 2207.7, 40.3, 62)
	ℓ_{44}	(0.3750, 0.0000, 0.0000, 0.6250)	(5406.8, 2582.5, 58.3, 47)

$\mathbf{x}^{IWRA}$. Table 4.9 reports the objective function values of the actual most preferred solution and the corresponding IWRA-MOIP solution along with element-wise percentage deviations between these objective vectors for each instance. As seen in Table 4.9, 4-iteration IWRA-MOIP converges to the actual most preferred solution for 28 out of 50 instances and provides solutions with low deviations in most cases. Only 7 out of 50 instances include a percentage deviation greater than 15%.



(a) Solution s_1. (b) Solution s_2. (c) Solution s_3. (d) Solution s_4.

Figure 4.4. Comparison page for set $\mathcal{S} = \{s_1, s_2, s_3, s_4\}$.

Table 4.9. Comparison of actual solutions and 4-iteration IWRA solutions.

$F(\mathbf{x}^*)$	$F(\mathbf{x}^{IWRA})$	**% Deviations**
(5952, 2659, 98.2, 66)	s_{3124}: (5952, 2659, 98.2, 66)	(0.0, 0.0, 0.0, 0.0)
(5652, 2174, 44.2, 61)	s_{2441}: (5735, 2186, 44.4, 59)	(1.5, 0.6, 0.5, 3.3)
(6500, 2511, 59.6, 43)	s_{4141}: (6500, 2511, 59.6, 43)	(0.0, 0.0, 0.0, 0.0)
(5174, 2589, 50.7, 52)	s_{1123}: (5174, 2589, 50.7, 52)	(0.0, 0.0, 0.0, 0.0)
(5413, 2594, 60.5, 47)	s_{4211}: (5418, 2502, 64.5, 49)	(0.1, 3.5, 6.6, 4.3)
(6500, 2511, 59.6, 43)	s_{4141}: (6500, 2511, 59.6, 43)	(0.0, 0.0, 0.0, 0.0)
(5927, 2669, 100, 70)	s_{3111}: (5927, 2669, 100, 70)	(0.0, 0.0, 0.0, 0.0)
(5229, 2562, 54.9, 51)	s_{4424}: (5411, 2491, 62.3, 49)	(3.5, 2.8, 13.5, 3.9)
(5709, 2333, 79.0, 61)	s_{3413}: (5709, 2333, 79.0, 61)	(0.0, 0.0, 0.0, 0.0)
(5927, 2669, 100.0, 70)	s_{3111}: (5927, 2669, 100, 70)	(0.0, 0.0, 0.0, 0.0)
(5869, 2307, 64.9, 53)	s_{4212}: (5709, 2333, 79.0, 61)	(2.7, 1.1, 21.7, 15.1)
(5204, 2606, 56.1, 51)	s_{4421}: (5293, 2597, 58.9, 49)	(1.7, 0.3, 5.0, 3.9)
(5927, 2669, 100, 70)	s_{3111}: (5927, 2669, 100, 70)	(0.0, 0.0, 0.0, 0.0)
(6503, 2493, 56.2, 43)	s_{4131}: (6503, 2493, 56.2, 43)	(0.0, 0.0, 0.0, 0.0)
(5719, 2430, 77.2, 55)	s_{3213}: (5752, 2554, 82.2, 55)	(0.6, 5.1, 6.5, 0.0)
(5138, 2475, 62.6, 61)	s_{1321}: (5228, 2523, 63.0, 53)	(1.8, 1.9, 0.6, 13.1)
(6099, 2189, 45.7, 57)	s_{4343}: (5807, 2132, 46.9, 65)	(4.8, 2.6, 2.6, 14.0)

continued ...

. . . continued

$F(\mathbf{x}^*)$	$F(\mathbf{x}^{IWRA})$	**% Deviations**
(5878, 2680, 93.0, 60)	s_{3121}: (5878, 2680, 93.0, 60)	(0.0, 0.0, 0.0, 0.0)
(5300, 2608, 61.1, 49)	s_{4423}: (5300, 2608, 61.1, 49)	(0.0, 0.0, 0.0, 0.0)
(5418, 2502, 64.5, 49)	s_{4211}: (5418, 2502, 64.5, 49)	(0.0, 0.0, 0.0, 0.0)
(6500, 2511, 59.6, 43)	s_{4141}: (6500, 2511, 59.6, 43)	(0.0, 0.0, 0.0, 0.0)
(5949, 2661, 96.6, 64)	s_{3121}: (5878, 2680, 93.0, 60)	(1.2, 0.7, 3.7, 6.3)
(5927, 2669, 100, 70)	s_{3111}: (5927, 2669, 100, 70)	(0.0, 0.0, 0.0, 0.0)
(5008, 2293, 55.0, 89)	s_{1444}: (5034, 2390, 52.4, 65)	(0.5, 4.2, 4.7, 27.0)
(5709, 2333, 79.0, 61)	s_{3413}: (5709, 2333, 79.0, 61)	(0.0, 0.0, 0.0, 0.0)
(5418, 2502, 64.5, 49)	s_{4211}: (5418, 2502, 64.5, 49)	(0.0, 0.0, 0.0, 0.0)
(5927, 2669, 100, 70)	s_{3111}: (5927, 2669, 100, 70)	(0.0, 0.0, 0.0, 0.0)
(5407, 2583, 58.3, 47)	s_{4411}: (5407, 2583, 58.3, 47)	(0.0, 0.0, 0.0, 0.0)
(5162, 2515, 64.6, 61)	s_{3234}: (5228, 2523, 63.0, 53)	(1.3, 0.3, 2.5, 13.1)
(5220, 2577, 56.5, 51)	s_{4421}: (5293, 2597, 58.9, 49)	(1.4, 0.8, 4.2, 3.9)
(5571, 2501, 56.8, 47)	s_{4411}: (5407, 2583, 58.3, 47)	(2.9, 3.3, 2.6, 0.0)
(5539, 2467, 77.1, 57)	s_{3221}: (5878, 2680, 93.0, 60)	(6.1, 8.6, 20.6, 5.3)
(5120, 2365, 58.8, 64)	s_{1221}: (5014, 2305, 57.2, 89)	(2.1, 2.5, 2.7, 39.1)
(5719, 2430, 77.2, 55)	s_{4211}: (5418, 2502, 64.5, 49)	(5.3, 3.0, 16.5, 10.9)
(5927, 2669, 100, 70)	s_{3111}: (5927, 2669, 100, 70)	(0.0, 0.0, 0.0, 0.0)
(5418, 2502, 64.5, 49)	s_{4211}: (5418, 2502, 64.5, 49)	(0.0, 0.0, 0.0, 0.0)
(6500, 2511, 59.6, 43)	s_{4141}: (6500, 2511, 59.6, 43)	(0.0, 0.0, 0.0, 0.0)
(5878, 2680, 93.0, 60)	s_{3121}: (5878, 2680, 93.0, 60)	(0.0, 0.0, 0.0, 0.0)
(5418, 2502, 64.5, 49)	s_{4211}: (5418, 2502, 64.5, 49)	(0.0, 0.0, 0.0, 0.0)
(5348, 2198, 62.1, 81)	s_{2231}: (5370, 2241, 71.3, 89)	(0.4, 2.0, 14.8, 9.9)
(5368, 2329, 62.7, 57)	s_{4211}: (5418, 2502, 64.5, 49)	(0.9, 7.4, 2.9, 14.0)
(6500, 2511, 59.6, 43)	s_{4141}: (6500, 2511, 59.6, 43)	(0.0, 0.0, 0.0, 0.0)
(5695, 2403, 57.5, 49)	s_{4213}: (6503, 2493, 56.2, 43)	(14.2, 3.7, 2.3, 12.2)
(6503, 2493, 56.2, 43)	s_{4131}: (6503, 2493, 56.2, 43)	(0.0, 0.0, 0.0, 0.0)
(5418, 2502, 64.5, 49)	s_{4211}: (5418, 2502, 64.5, 49)	(0.0, 0.0, 0.0, 0.0)
(5202, 2503, 71.9, 71)	s_{3234}: (5228, 2523, 63.0, 53)	(0.5, 0.8, 12.4, 25.4)
(6500, 2511, 59.6, 43)	s_{4213}: (6503, 2493, 56.2, 43)	(0.0, 0.7, 5.7, 0.0)
(6500, 2511, 59.6, 43)	s_{4214}: (6526, 2552, 69.2, 47)	(0.4, 1.6, 16.1, 9.3)
(5952, 2659, 98.2, 66)	s_{3113}: (5952, 2659, 98.2, 66)	(0.0, 0.0, 0.0, 0.0)
(5878, 2680, 93.0, 60)	s_{3121}: (5878, 2680, 93.0, 60)	(0.0, 0.0, 0.0, 0.0)

The results of Table 4.9 can be evaluated from various angles. Low objective deviations for a random weight vector **w** indicate that the generated solution list contains a nondominated solution corresponding to a weight set near **w**. Therefore, we can empirically claim that IWRA-MOIP generates diverse solutions which provide a good representation of the solution space. The results also show the success of IWRA-MOIP in solving multiobjective optimization problems as the DM evaluates only 13 solutions in four iterations to reach her most preferred/near most preferred solution. Considering the results in Section 3.6.1.3, running IWRA-MOIP for a few additional iterations, after including the preference information obtained during the first four iterations, must be sufficient to terminate the algorithm with a guaranteed most preferred solution.

4.4. Conclusions and Future Work

This chapter provides extensions to the arc-additive LFATP formulations to solve marathon course design problem for sequence- and time-dependent objective functions. The standard and DP formulations for arc-additive objectives and their extensions to sequence- and time-dependent objectives are tested by using a wide range of objective functions. The numerical experiments indicate that DP formulations perform better for average proximity objectives as these formulations are better in eliminating excessive subtour formation caused by proximity objectives. For sequence-dependent objectives, which focus on minimizing turns along the route, standard formulations perform better since turn minimization implicitly discourages subtour formation. Both formulations show a relatively poor performance for time-dependent objectives despite our simplification efforts in time tracking and cost calculation. While DP formulations are remarkably

better in reducing the optimality gaps for time-dependent objectives, they are still far from proving optimality. Considering the time-dependent studies in the literature, these results are not surprising as time-dependence becomes more difficult to handle with increasing network and tour size.

The extensions are integrated to develop combined standard and DP formulations which are capable of solving any combination of arc-additive, sequence-dependent and time-dependent objectives. These formulations are used jointly with IWRA-MOIP to solve the marathon course design problem in a multiobjective setting. Different form standard iteration-based application of IWRA-MOIP, we generate comparison sets (and solutions) corresponding to all possible preference scenarios within the first four iterations of IWRA-MOIP. Additional experiments with a simulated decision maker show that even four iterations can be sufficient to find the DM's most preferred solutions in more than half of the instances and the generated solutions provide a good representation of the solution space. One potential direction to speed up the multi-iteration set generation procedure is to share solutions and cut information when solving the combined LFATP formulations for different weight vectors. Currently, the system treats each problem individually; however, already identified solutions and valid inequalities can be used to warm start subsequent problems.

CHAPTER 5

Conclusions

This study develops models and solution approaches to design marathon courses with various objectives. As courses must visit certain neighborhoods in the city due to traditions and/or sponsorship restrictions, marathon course design problem is considered as a tour finding problem which visits a predetermined set of street segments in the city. Marathon routes differ from standard vehicle routes as they stay in the network over a five-to-seven hour span. While a standard vehicle route does not affect the underlying network (apart from its contribution to traffic congestion), marathon routes block access to certain districts in the city for a relatively long period of time. For this reason, we introduce a novel tour finding problem, LFATP, which takes access related restrictions into account, and develop standard and disjunctive programming formulations to solve the LFATPfor a wide range of objectives.

The study also focuses on the multiobjective nature of the marathon course design and introduces an interactive multiobjective optimization approach (IWRA) to find a most preferred solution of a decision maker in the presence of multiple objectives. IWRA differs from existing interactive methods as it provides a systematic way of generating diverse weight vectors via a mixed integer linear programming formulation and separates weight diversification and solution generation efforts. In Chapter 4, we combine LFATP formulations and IWRA to successfully generate a guided set of marathon course designs for the Bank of America Chicago Marathon in a 4-objective setting with medical

proximity minimization, public transport proximity minimization, similarity (ease of implementation) maximization and sharp turn minimization objectives. The rest of this chapter briefly summarizes our contributions to routing, multiobjective optimization and marathon course design literature.

5.1. Advancements in Routing and Future Work

This study contributes to the routing literature from various perspectives. From the application perspective, access related restrictions have never been considered in the routing literature and the LFATP is the first problem to address this issue. In addition to this unique restriction, LFATP is solved to optimize a novel class of proximity minimization functions which aim to minimize the average distance to a predetermined set of vertices in the network. With proximity minimization objectives, the underlying cost structure becomes clustered, leading to excessive subtour formation.

When solving standard routing formulations with exponentially many subtour elimination constraints, the standard B&C approach relaxes most of these constraints at the beginning of the optimization and reintroduces them to the formulation when they are violated during the optimization process. However, when the problem encourages excessive subtour formation, the standard B&C approach has difficulty in finding feasible solutions and converging to optimality. To overcome this issue, we introduce a disjunctive programming formulation for the LFATP that exploits the tour length bounds to efficiently solve the LFATP by enforcing visit order of must-visit arcs. The proposed DP formulation is found to be effective in various tour finding problems with similar requirements.

The study extends the standard and disjunctive programming formulations to solve tour finding problems for a wide range of objective functions including sequence- and time-dependent ones. While the extensions are remarkably successful in solving problems with sequence-dependent objectives, solving those with time-dependent objectives remains a challenge. In our preliminary experiments with different time-dependent models, we observe that the underlying models become more difficult to solve when costs depend on time rather than travel times. Considering ease of access to historic data and technology, further efforts on time-dependent problems, especially those with time-dependent service costs/times are needed.

5.2. Advancements in Multiobjective Optimization and Future Work

This study contributes to multiobjective optimization literature by introducing a novel interactive weight region based approach, IWRA, which is capable of finding the decision maker's most preferable solution for multiobjective linear problems up to 10 and multi-objective integer problems up to 5 objectives. Different from existing approaches, IWRA separates the weight diversification and solution generation efforts, and provides a mixed integer linear programming formulation which is capable of exploring the entire weight region in a systematic way. Separating weight diversification and solution generation efforts enables us to scale IWRA to a large number of objectives.

The numerical experiments in Chapter 3 and the case study in Chapter 4 show that IWRA requires a reasonably small number of comparisons to identify the most preferred solution of the decision maker. Multi-iteration results from Chapter 4 show that IWRA is successful not only in identifying the most preferred solutions but also generating a

guided set of diverse solutions in the absence of preference information. Furthermore, IWRA provides reliable results when the decision maker makes imperfect comparisons yielding incorrect preference information. Considering its success in various multiobjective problems, extending IWRA to more general utility settings (currently supports weighted sum and weighted Tchebycheff functions) is a promising future direction. Furthermore, considering the results of multi-iteration comparison set generation experiments in Section 4.3.3, exploring the applicability of IWRA in generating representative solution sets via guided solution set logic is another promising future direction.

5.3. Advancements in Marathon Course Design and Future Work

This study, to the best of our knowledge, is the first optimization-based attempt to design marathon courses. Our optimization-based approach to marathon course design has two key benefits. First, as marathons are increasing in popularity, this approach allows race organizers to design new marathon courses and test a wide range of objectives when designing a course. Second, it allows race organizers to evaluate the quality of existing marathon courses and provides improvement suggestions while ensuring ease of implementation with similarity constraints and objectives.

Our case studies from the Bank of America Chicago Marathon indicate that slight changes in the existing courses may lead to significant improvements for objectives related to safety, health and experience. A potential future direction in marathon course design is to develop more comprehensive models that integrate various aspects of marathon planning, such as manpower planning and resource management operations, when designing a course.

References

Alves, M. J. and Clímaco, J. (2000). An interactive reference point approach for multiobjective mixed-integer programming using branch-and-bound. *European Journal of Operational Research*, 124(3):478–494.

Alves, M. J. and Clímaco, J. (2007). A review of interactive methods for multiobjective integer and mixed-integer programming. *European Journal of Operational Research*, 180(1):99–115.

Anez, J., De La Barra, T., and Pérez, B. (1996). Dual graph representation of transport networks. *Transportation Research Part B: Methodological*, 30(3):209–216.

Antunes, C. H., Alves, M. J., and Clímaco, J. (2016). *Multiobjective linear and integer programming*. Springer.

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

Aráoz, J., Fernández, E., and Franquesa, C. (2009a). The clustered prize-collecting arc routing problem. *Transportation Science*, 43(3):287–300.

Aráoz, J., Fernández, E., and Franquesa, C. (2013). Grasp and path relinking for the clustered prize-collecting arc routing problem. *Journal of Heuristics*, 19(2):343–371.

Aráoz, J., Fernández, E., and Meza, O. (2009b). Solving the prize-collecting rural postman problem. *European Journal of Operational Research*, 196(3):886–896.

Aráoz, J., Fernández, E., and Zoltan, C. (2006). Privatized rural postman problems. *Computers & Operations Research*, 33(12):3432–3449.

Archetti, C., Bianchessi, N., Speranza, M. G., and Hertz, A. (2014). The split delivery capacitated team orienteering problem. *Networks*, 63(1):16–33.

Archetti, C., Corberán, Á., Plana, I., Sanchis, J. M., and Speranza, M. G. (2015). A matheuristic for the team orienteering arc routing problem. *European Journal of Operational Research*, 245(2):392–401.

Archetti, C., Feillet, D., Hertz, A., and Speranza, M. G. (2010). The undirected capacitated arc routing problem with profits. *Computers & Operations Research*, 37(11):1860–1869.

Archetti, C., Speranza, M. G., Corberán, Á., Sanchis, J. M., and Plana, I. (2013). The team orienteering arc routing problem. *Transportation Science*, 48(3):442–457.

Awerbuch, B., Azar, Y., Blum, A., and Vempala, S. (1998). New approximation guarantees for minimum-weight k-trees and prize-collecting salesmen. *SIAM Journal on Computing*, 28(1):254–262.

Başdere, M., Caniglia, G., Collar, C., Rozolis, C., Chiampas, G., Nishi, M., and Smilowitz, K. (2018). SAFE: a comprehensive data visualization system. Submitted to Interfaces.

Balas, E. (2005). Projection, lifting and extended formulation in integer and combinatorial optimization. *Annals of Operations Research*, 140(1):125–161.

Bartolini, E., Cordeau, J.-F., and Laporte, G. (2013a). An exact algorithm for the capacitated arc routing problem with deadheading demand. *Operations Research*, 61(2):315–327.

Bartolini, E., Cordeau, J.-F., and Laporte, G. (2013b). Improved lower bounds and exact algorithm for the capacitated arc routing problem. *Mathematical Programming*, 137(1-2):409–452.

Başdere, M., Ross, C., Chan, J. L., Mehrotra, S., Smilowitz, K., and Chiampas, G. (2014). Acute incident rapid response at a mass-gathering event through comprehensive planning systems: a case report from the 2013 Shamrock Shuffle. *Prehospital and Disaster Medicine*, 29(03):320–325.

Bemporad, A., Fukuda, K., and Torrisi, F. D. (2001). Convexity recognition of the union of polyhedra. *Computational Geometry*, 18(3):141–154.

Bertsimas, D. and O'Hair, A. (2013). Learning preferences under noise and loss aversion: an optimization approach. *Operations Research*, 61(5):1190–1199.

Black, D., Eglese, R., and Wøhlk, S. (2013). The time-dependent prize-collecting arc routing problem. *Computers & Operations Research*, 40(2):526–535.

Boroujerdi, A. and Uhlmann, J. (1998). An efficient algorithm for computing least cost paths with turn constraints. *Information Processing Letters*, 67(6):317–321.

Bräysy, O., Martínez, E., Nagata, Y., and Soler, D. (2011). The mixed capacitated general routing problem with turn penalties. *Expert Systems with Applications*, 38(10):12954–12966.

Caldwell, T. (1961). On finding minimum routes in a network with turn penalties. *Communications of the ACM*, 4(2):107–108.

Chankong, V. and Haimes, Y. Y. (1983). *Multiobjective decision making: theory and methodology*. North-Holland, New York, USA.

Chiampas, G. and Jaworski, C. A. (2009). Preparing for the surge: perspectives on marathon medical preparedness. *Current Sports Medicine Reports*, 8(3):131–135.

City of Chicago (2013). Street center line. `http://www.cityofchicago.org/city/en/depts/doit/dataset/street_center_line.html`. [Online; accessed 27-Apr-2013].

City of Chicago (2017). Chicago Transit Authority - System information - List of 'L' stops. `https://data.cityofchicago.org/Transportation/CTA-System-Information-List-of-L-Stops/8pix-ypme`. [Online; accessed 01-Apr-2017].

Clossey, J., Laporte, G., and Soriano, P. (2001). Solving arc routing problems with turn penalties. *Journal of the Operational Research Society*, 52(4):433–439.

Corberán, Á., Fernández, E., Franquesa, C., and Sanchis, J. M. (2011). The windy clustered prize-collecting arc-routing problem. *Transportation Science*, 45(3):317–334.

Corberán, Á. and Laporte, G., editors (2015). *Arc routing: problems, methods, and applications*, volume 20. SIAM.

Corberán, A., Martí, R., Martínez, E., and Soler, D. (2002). The rural postman problem on mixed graphs with turn penalties. *Computers & Operations Research*, 29(7):887–903.

Corberán, A. and Prins, C. (2010). Recent results on arc routing problems: an annotated bibliography. *Networks*, 56(1):50–69.

Das, I. and Dennis, J. E. (1997). A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural Optimization*, 14(1):63–69.

Das, I. and Dennis, J. E. (1998). Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM*

Journal on Optimization, 8(3):631–657.

Deitch, R. and Ladany, S. P. (2000). The one-period bus touring problem: solved by an effective heuristic for the orienteering tour problem and improvement algorithm. *European Journal of Operational Research*, 127(1):69–77.

Dell, R. F. and Karwan, M. H. (1990). An interactive MCDM weight space reduction method utilizing a Tchebycheff utility function. *Naval Research Logistics (NRL)*, 37(2):263–277.

Dror, M. (2000). *Arc routing: theory, solutions, and applications*. Springer.

Edwards, W. (1977). How to use multiattribute utility measurement for social decision-making. *Systems, Man and Cybernetics, IEEE Transactions on*, 7(5):326–340.

Edwards, W. and von Winterfeldt, D. (1986). *Decision analysis and behavioral research*. Cambridge University Press, Cambridge, UK.

Ekici, A. and Retharekar, A. (2013). Multiple agents maximum collection problem with time dependent rewards. *Computers & Industrial Engineering*, 64(4):1009–1018.

Erdoğan, G. and Laporte, G. (2013). The orienteering problem with variable profits. *Networks*, 61(2):104–116.

Erkut, E. and Zhang, J. (1996). The maximum collection problem with time-dependent rewards. *Naval Research Logistics (NRL)*, 43(5):749–763.

Faulkenberg, S. L. and Wiecek, M. M. (2010). On the quality of discrete representations in multiple objective programming. *Optimization and Engineering*, 11(3):423–440.

Feillet, D., Dejax, P., and Gendreau, M. (2005). The profitable arc tour problem: solution with a branch-and-price algorithm. *Transportation Science*, 39(4):539–552.

Few, L. (1955). The shortest path and the shortest road through n points. *Mathematika*, 2(2):141–144.

Fidler, M. and Einhoff, G. (2004). Routing in turn-prohibition based feed-forward networks. In *International Conference on Research in Networking*, pages 1168–1179. Springer.

Fischetti, M., Gonzalez, J. J. S., and Toth, P. (1998). Solving the orienteering problem through branch-and-cut. *INFORMS Journal on Computing*, 10(2):133–148.

Fox, K. R., Gavish, B., and Graves, S. C. (1980). An n-constraint formulation of the (time-dependent) traveling salesman problem. *Operations Research*, 28(4):1018–1021.

Garcia, A., Arbelaitz, O., Vansteenwegen, P., Souffriau, W., and Linaza, M. T. (2010). Hybrid approach for the public transportation time dependent orienteering problem with time windows. In *International Conference on Hybrid Artificial Intelligence Systems*, pages 151–158. Springer.

Gavalas, D., Konstantopoulos, C., Mastakas, K., Pantziou, G., and Vathis, N. (2015). Heuristics for the time dependent team orienteering problem: application to tourist route planning. *Computers & Operations Research*, 62:36–50.

Gendreau, M., Ghiani, G., and Guerriero, E. (2015). Time-dependent routing problems: a review. *Computers & operations research*, 64:189–197.

Geoffrion, A. M. (1968). Proper efficiency and the theory of vector maximization. *Journal of Mathematical Analysis and Applications*, 22(3):618 – 630.

Golden, B. L., Levy, L., and Vohra, R. (1987). The orienteering problem. *Naval Research Logistics*, 34(3):307–318.

Google Maps (2015). `https://www.google.com/maps`. [Online; accessed 27-Apr-2015].

Gunawan, A., Lau, H. C., Vansteenwegen, P., and Lu, K. (2017). Well-tuned algorithms for the team orienteering problem with time windows. *Journal of the Operational Research Society*, 68(8):861–876.

Gunawan, A., Yuan, Z., and Lau, H. C. (2014). A mathematical model and metaheuristics for time dependent orienteering problem. PATAT.

Gurobi Optimization (2013). Gurobi optimizer version 5.6. `http://www.gurobi.com/`. [Online; accessed 27-Apr-2015].

Gurobi Optimization (2018). Gurobi optimizer version 7.5. `http://www.gurobi.com/`. [Online; accessed 18-Apr-2018].

Gutiérrez, E. and Medaglia, A. L. (2008). Labeling algorithm for the shortest path problem with turn prohibitions with application to large-scale road networks. *Annals of Operations Research*, 157(1):169–182.

Hanken, T., Young, S., Smilowitz, K., Chiampas, G., and Waskowski, D. (2016). Developing a data visualization system for the Bank of America Chicago Marathon (Chicago, Illinois USA). *Prehospital and Disaster Medicine*, 31(5):572–577.

Hassanzadeh, F., Nemati, H., and Sun, M. (2014). Robust optimization for interactive multiobjective programming with imprecise information applied to R&D project portfolio selection. *European Journal of Operational Research*, 238(1):41–53.

Hu, J. and Mehrotra, S. (2012). Robust and stochastically weighted multiobjective optimization models and reformulations. *Operations Research*, 60(4):936–953.

IAAF (2015). Rules & Regulations. `http://www.iaaf.org/about-iaaf/documents/rules-regulations`. [Online; accessed 27-Apr-2015].

Irnich, S. (2008). Solution of real-world postman problems. *European Journal of Operational Research*, 190(1):52–67.

Isermann, H. (1974). Technical note - proper efficiency and the linear vector maximum problem. *Operations Research*, 22(1):189–191.

Jossé, G., Lu, Y., Emrich, T., Renz, M., Shahabi, C., Demiryurek, U., and Schubert, M. (2016). Scenic routes now: efficiently solving the time-dependent arc orienteering problem. *arXiv preprint arXiv:1609.08484*.

Kantor, M. G. and Rosenwein, M. B. (1992). The orienteering problem with time windows. *Journal of the Operational Research Society*, 43(6):629–635.

Karloff, H. J. (1989). How long can a euclidean traveling salesman tour be? *SIAM Journal on Discrete Mathematics*, 2(1):91–99.

Karp, R. M. (1972). *Reducibility among combinatorial problems*. Springer.

Keeney, R. L. and Raiffa, H. (1993). *Decisions with multiple objectives: preferences and value trade-offs*. Cambridge University Press, Cambridge, UK.

Kim, I. Y. and De Weck, O. (2006). Adaptive weighted sum method for multiobjective optimization: a new method for Pareto front generation. *Structural and Multidisciplinary Optimization*, 31(2):105–116.

Kim, J. H., Malhotra, R., Chiampas, G., d'Hemecourt, P., Troyanos, C., Cianca, J., Smith, R. N., Wang, T. J., Roberts, W. O., Thompson, P. D., et al. (2012). Cardiac arrest during long-distance running races. *New England Journal of Medicine*, 366(2):130–140.

Kirby, R. F. and Potts, R. B. (1969). The minimum route problem for networks with turn penalties and prohibitions. *Transportation Research*, 3(3):397–408.

Klein, P. and Ravi, R. (1995). A nearly best-possible approximation algorithm for node-weighted Steiner trees. *Journal of Algorithms*, 19(1):104–115.

Köksalan, M. and Karahan, I. (2010). An interactive territory defining evolutionary algorithm: iTDEA. *Evolutionary Computation, IEEE Transactions on*, 14(5):702–722.

Köksalan, M. and Lokman, B. (2009). Approximating the nondominated frontiers of multi-objective combinatorial optimization problems. *Naval Research Logistics*, 56(2):191–198.

Labadie, N., Melechovskỳ, J., and Calvo, R. W. (2011). Hybridized evolutionary local search algorithm for the team orienteering problem with time windows. *Journal of Heuristics*, 17(6):729–753.

Laporte, G. (1997). Modeling and solving several classes of arc routing problems as traveling salesman problems. *Computers & Operations Research*, 24(11):1057–1061.

Laporte, G. and Martello, S. (1990). The selective travelling salesman problem. *Discrete Applied Mathematics*, 26(2):193–207.

Li, J. (2011). Model and algorithm for time-dependent team orienteering problem. In *Advanced Research on Computer Education, Simulation and Modeling*, pages 1–7. Springer.

Li, J., Wu, Q., Li, X., and Zhu, D. (2010). Study on the time-dependent orienteering problem. In *E-Product E-Service and E-Entertainment (ICEEE), 2010 International Conference on*, pages 1–4. IEEE.

Lokman, B. (2007). Approaches for multi-objective combinatorial optimization problems. Master's thesis, Middle East Technical University, Ankara.

Lokman, B., Köksalan, M., Korhonen, P. J., and Wallenius, J. (2016). An interactive algorithm to find the most preferred solution of multi-objective integer programs. *Annals of Operations Research*, 245(1-2):67–95.

López-Ibáñez, M. and Knowles, J. (2015). Machine decision makers as a laboratory for interactive EMO. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 295–309. Springer.

Malandraki, C. and Daskin, M. S. (1992). Time dependent vehicle routing problems: formulations, properties and heuristic algorithms. *Transportation Science*, 26(3):185–200.

Masin, M. and Bukchin, Y. (2008). Diversity maximization approach for multiobjective optimization. *Operations Research*, 56(2):411–424.

McCarthy, D. M., Chiampas, G. T., Malik, S., Cole, K., Lindeman, P., and Adams, J. G. (2011). Enhancing community disaster resilience through mass sporting events. *Disaster Medicine and Public Health Preparedness*, 5(04):310–315.

Mei, Y., Salim, F. D., and Li, X. (2016). Efficient meta-heuristics for the multi-objective time-dependent orienteering problem. *European Journal of Operational Research*, 254(2):443–457.

Messac, A. and Mattson, C. A. (2002). Generating well-distributed sets of Pareto points for engineering design using physical programming. *Optimization and Engineering*, 3(4):431–450.

Micó, J. C. and Soler, D. (2011). The capacitated general windy routing problem with turn penalties. *Operations Research Letters*, 39(4):265–271.

Miettinen, K. (2008). Introduction to multiobjective optimization: noninteractive approaches. In *Multiobjective Optimization*, pages 1–26. Springer.

Miettinen, K., Ruiz, F., and Wierzbicki, A. P. (2008). Introduction to multiobjective optimization: interactive approaches. In *Multiobjective Optimization*, pages 27–57. Springer.

Özpeynirci, Ö. and Köksalan, M. (2010). An exact algorithm for finding extreme supported nondominated points of multiobjective mixed integer programs. *Management Science*, 56(12):2302–2315.

Picard, J.-C. and Queyranne, M. (1978). The time-dependent traveling salesman problem and its application to the tardiness problem in one-machine scheduling. *Operations Research*, 26(1):86–110.

Ramesh, R., Karwan, M. H., and Zionts, S. (1989). Interactive multicriteria linear programming: an extension of the method of Zionts and Wallenius. *Naval Research Logistics (NRL)*, 36(3):321–335.

Ross, C., Başdere, M., Chan, J. L., Mehrotra, S., Smilowitz, K., and Chiampas, G. (2015). Data value in patient tracking systems at racing events. *Medicine and Science in Sports and Exercise*.

Saaty, T. L. (1990). How to make a decision: the analytic hierarchy process. *European Journal of Operational Research*, 48(1):9–26.

Souffriau, W., Vansteenwegen, P., Vanden Berghe, G., and Van Oudheusden, D. (2011). The planning of cycle trips in the province of east flanders. *Omega*, 39(2):209–213.

Speičvcys, L., Jensen, C. S., and Kligys, A. (2003). Computational data modeling for network-constrained moving objects. In *Proceedings of the 11th ACM International*

Symposium on Advances in Geographic Information Systems, pages 118–125. ACM.

Steuer, R. E. (2006). ADBASE: a multiple objective linear programming solver for efficient extreme points and unbounded efficient edges. *Terry College of Business, University of Georgia, Athens, Georgia, USA*.

Steuer, R. E. and Choo, E.-U. (1983). An interactive weighted Tchebycheff procedure for multiple objective programming. *Mathematical Programming*, 26(3):326–344.

Steuer, R. E. and Harris, F. W. (1980). Intra-set point generation and filtering in decision and criterion space. *Computers & Operations Research*, 7(1):41–53.

Steuer, R. E., Silverman, J., and Whisman, A. W. (1993). A combined Tchebycheff/aspiration criterion vector interactive multiobjective programming procedure. *Management Science*, 39(10):1255–1260.

Stewart, T. J. (1999). Evaluation and refinement of aspiration-based methods in MCDM. *European Journal of Operational Research*, 113(3):643–652.

Stout, R. J., Cannon-Bowers, J. A., Salas, E., and Milanovich, D. M. (1999). Planning, shared mental models, and coordinated performance: an empirical link is established. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 41(1):61–71.

Suozzo, A. (2002). The chicago marathon and urban renaissance. *The Journal of Popular Culture*, 36(1):142–159.

Tagmouti, M., Gendreau, M., and Potvin, J.-Y. (2007). Arc routing problems with time-dependent service costs. *European Journal of Operational Research*, 181(1):30–39.

Tagmouti, M., Gendreau, M., and Potvin, J.-Y. (2011). A dynamic capacitated arc routing problem with time-dependent service costs. *Transportation Research Part C: Emerging*

Technologies, 19(1):20–28.

Toubia, O., Hauser, J., and Garcia, R. (2007). Probabilistic polyhedral methods for adaptive choice-based conjoint analysis: theory and application. *Marketing Science*, 26(5):596–610.

Toubia, O., Hauser, J. R., and Simester, D. I. (2004). Polyhedral methods for adaptive choice-based conjoint analysis. *Journal of Marketing Research*, 41(1):116–131.

Toubia, O., Simester, D. I., Hauser, J. R., and Dahan, E. (2003). Fast polyhedral adaptive conjoint estimation. *Marketing Science*, 22(3):273–303.

Vanhove, S. and Fack, V. (2012). Route planning with turn restrictions: a computational experiment. *Operations Research Letters*, 40(5):342–348.

Vansteenwegen, P., Souffriau, W., Berghe, G. V., and Van Oudheusden, D. (2009). Iterated local search for the team orienteering problem with time windows. *Computers & Operations Research*, 36(12):3281–3290.

Vansteenwegen, P., Souffriau, W., and Oudheusden, D. V. (2011). The orienteering problem: a survey. *European Journal of Operational Research*, 209(1):1–10.

Vecchietti, A., Lee, S., and Grossmann, I. E. (2003). Modeling of discrete/continuous optimization problems: characterization and formulation of disjunctions and their relaxations. *Computers & Chemical Engineering*, 27(3):433–448.

Verbeeck, C., Sörensen, K., Aghezzaf, E.-H., and Vansteenwegen, P. (2014). A fast solution method for the time-dependent orienteering problem. *European Journal of Operational Research*, 236(2):419–432.

Verbeeck, C., Vansteenwegen, P., and Aghezzaf, E.-H. (2017). The time-dependent orienteering problem with time windows: a fast ant colony system. *Annals of Operations*

Research, 254(1-2):481–505.

Winter, S. (2002). Modeling costs of turns in route planning. *GeoInformatica*, 6(4):345–361.

Wolsey, L. A. and Nemhauser, G. L. (1999). *Integer and combinatorial optimization*. Wiley-Interscience.

Zeleny, M. (1974). *Lecture notes in economics and mathematical systems: linear multiobjective programming*, volume 95. Springer-Verlag, New York, USA.

Ziliaskopoulos, A. K. and Mahmassani, H. S. (1996). A note on least time path computation considering delays and prohibitions for intersection movements. *Transportation Research Part B: Methodological*, 30(5):359–367.

Zionts, S. and Wallenius, J. (1976). An interactive programming method for solving the multiple criteria problem. *Management Science*, 22(6):652–663.

Zionts, S. and Wallenius, J. (1983). An interactive multiple objective linear programming method for a class of underlying nonlinear utility functions. *Management Science*, 29(5):519–529.

APPENDIX A

Appendix for Chapter 2

A.1. Numerical Results for DP-based B&C Approach

This section provides numerical experiments for DP-based B&C approach with varying fixed depth level, φ, values from 2 to 5. When $|M_g| < \varphi$, we solve subproblems without constraints (2.7) with a time limit of 5 seconds to obtain quick lower bounds and detect infeasibilities. When $|M_g| = \varphi$, we solve subproblems with constraints (2.7) with a time limit of 2 hours to obtain a feasible solution at that level, and never go below φ^{th} level in the tree. When $\varphi \geq 3$, the root note starts with M_3 as three edges form a single order when costs are symmetric. We also test $\varphi = 2$ to see the effect of having fewer edges than the unique order. Note that for $\varphi = 2$ and $\varphi = 3$, we solve a single formulation at the root node and this formulation is *LFATP-R2* when $\varphi = 2$ and *LFATP-R3* when $\varphi = 3$.

We use problem settings and instances from Section 2.5.1 and 2.5.2 to compare DP-based B&C with varying levels of φ and *LFATP-S*. Tables A.1, A.2 and A.3 summarize the results for the LFATP, tour length minimization and random coefficient minimization instances, respectively. In the tables, each row provides a summary of 10 instances for a given problem setting. *Sbtr* and *VDLEI* columns represent the average number of subtour elimination inequalities and VDLEIs identified during B&C. For a given $\varphi \geq 3$, the total number of orders at φ^{th} level is $(\varphi - 1)!/2$ and the maximum number of nodes the corresponding B&C tree can have is $\sum_{i=3}^{\varphi}(\varphi - 1)!/2$ (if $\varphi = 2$, then both values are

Table A.1. DP-based B&C results for LFATP instances.

Setting		**Count**				**Duration**		**Performance**			
Network	φ	*Sbtr*	*VDLEI*	*Orders*	*Nodes*	*CutGen*	*Total*	*Gap (%)*	*Opt*	*Feas*	*NoSn*
M12Q4 [95% - 110%]	2	23	0	1 / 1	1 / 1	4	9	0.0	10	0	0
	3	27	1	1 / 1	1 / 1	4	8	0.0	10	0	0
	4	14	6	2 / 3	3 / 4	1	10	0.0	10	0	0
	5	33	12	4.1 / 12	7.1 / 16	5	28	0.0	10	0	0
	-*S*	3369	39	-	-	555	1727	0.5	9	1	0
M12Q4 [110% - 125%]	2	105	8	1 / 1	1 / 1	16	25	0.0	10	0	0
	3	107	11	1 / 1	1 / 1	18	26	0.0	10	0	0
	4	106	12	2.2 / 3	3.2 / 4	18	35	0.0	10	0	0
	5	203	41	5.5 / 12	8.7 / 16	38	89	0.0	10	0	0
	-*S*	3209	70	-	-	360	1349	0.5	9	1	0
M12Q4 [125% - 140%]	2	411	67	1 / 1	1 / 1	63	140	0.0	10	0	0
	3	496	58	1 / 1	1 / 1	69	227	0.0	10	0	0
	4	351	48	2.7 / 3	3.7 / 4	58	150	0.0	10	0	0
	5	1189	92	6.6 / 12	10.3 / 16	266	1309	0.1	9	1	0
	-*S*	4632	179	-	-	684	2459	1.0	7	3	0
M24Q4 [95% - 110%]	2	294	6	1 / 1	1 / 1	51	153	0.0	10	0	0
	3	214	5	1 / 1	1 / 1	32	147	0.0	10	0	0
	4	137	8	3 / 3	4 / 4	22	102	0.0	10	0	0
	5	178	22	10.2 / 12	14.2 / 16	35	208	0.0	10	0	0
	-*S*	2125	20	-	-	313	955	0.5	9	1	0
M24Q4 [110% - 125%]	2	1053	36	1 / 1	1 / 1	223	958	0.0	10	0	0
	3	1108	37	1 / 1	1 / 1	221	1831	0.1	8	2	0
	4	714	35	3 / 3	4 / 4	109	851	0.0	10	0	0
	5	988	55	11.9 / 12	15.9 / 16	160	1834	0.1	8	2	0
	-*S*	3460	118	-	-	506	1591	0.7	9	1	0
M24Q4 [125% - 140%]	2	3100	274	1 / 1	1 / 1	826	5192	0.6	4	6	0
	3	2123	182	1 / 1	1 / 1	428	4198	0.4	5	5	0
	4	2108	162	3 / 3	4 / 4	367	4500	0.3	5	5	0
	5	1783	203	12 / 12	16 / 16	352	4596	0.3	6	4	0
	-*S*	7598	400	-	-	1203	5474	1.1	5	5	0

1). When solving the DP-based B&C, not all visit orders or nodes are processed due to fathoming. *Orders* column provides the ratio of average number of visit orders for which the subproblems are solved to the total number of visit orders and *Nodes* column provides the ratio of average number of explored nodes to the maximum number of nodes. *CutGen* and *Total* columns present average cut generation and solution times in seconds. *Gap* column reports the average optimality gaps of solved instances in percentages while *Opt*, *Feas* and *NoSn* provide a breakdown of optimally solved, sub-optimally solved and unsolved instances, respectively.

Table A.2. DP-based B&C results for tour length minimization problem.

Setting		**Count**				**Duration (s)**		**Performance**			
Network	φ	*Sbtr*	*VDLEI*	*Orders*	*Nodes*	*CutGen*	*Total*	*Gap (%)*	*Opt*	*Feas*	*NoSn*
M12Q0	2	369	0	1 / 1	1 / 1	23	40	0.0	10	0	0
	3	349	0	1 / 1	1 / 1	20	59	0.0	10	0	0
	4	286	0	2.9 / 3	3.9 / 4	136	248	0.0	10	0	0
	5	133	0	5.8 / 12	9.7 / 16	13	101	0.0	10	0	0
	-S	10951	0	-	-	1480	4949	9.4	4	6	0
M24Q0	2	609	0	1 / 1	1 / 1	68	227	0.0	10	0	0
	3	833	0	1 / 1	1 / 1	57	432	0.0	10	0	0
	4	2168	0	3 / 3	4 / 4	177	1475	0.7	9	1	0
	5	1617	0	10.8 / 12	14.8 / 16	102	1378	0.0	10	0	0
	-S	8813	0	-	-	988	4284	3.3	5	5	0
M12Q4	2	59	1	1 / 1	1 / 1	5	9	0.0	10	0	0
	3	100	2	1 / 1	1 / 1	11	21	0.0	10	0	0
	4	13	0	3 / 3	3.9 / 4	2	16	0.0	10	0	0
	5	26	1	7.6 / 12	9.7 / 16	7	64	0.0	10	0	0
	-S	2037	21	-	-	289	905	0.4	9	1	0
M24Q4	2	509	23	1 / 1	1 / 1	53	162	0.0	10	0	0
	3	399	14	1 / 1	1 / 1	39	223	0.0	10	0	0
	4	382	9	3 / 3	4 / 4	43	222	0.0	10	0	0
	5	342	8	10.8 / 12	14.8 / 16	49	390	0.0	10	0	0
	-S	1049	24	-	-	72	166	0.0	10	0	0

In most instances, $\varphi = 2$ and $\varphi = 3$ yield the best solution times and gaps. As the length budget increases in random coefficient and average medical distance minimization problems, $\varphi = 4$ and $\varphi = 5$ start to lose their competitiveness. In two instances of random coefficient minimization, $\varphi = 4$ and $\varphi = 5$ fail to find a solution to the problem whereas this is never the case for $\varphi = 2$ and $\varphi = 3$. Even though $\varphi = 4$ and $\varphi = 5$ are almost always better than *LFATP-S*, they cannot compete against $\varphi = 2$ and $\varphi = 3$ when the solution times and the sizes of the underlying subproblems are considered.

Table A.3. DP-based B&C results for random coefficient minimization problem.

Setting			**Count**		**Duration**		**Performance**			
Network	φ	*Sbtr*	*Orders*	*Nodes*	*CutGen*	*Total*	*Gap (%)*	*Opt*	*Feas*	*NoSn*
M12Q0 [95% - 110%]	2	123	1 / 1	1 / 1	14	35	0.0	10	0	0
	3	162	1 / 1	1 / 1	33	83	0.0	10	0	0
	4	148	2.1 / 3	3.1 / 4	33	103	0.0	10	0	0
	5	36	4.6 / 12	7.7 / 16	9	33	0.0	10	0	0
	-S	8593	-	-	1131	3633	1.7	5	3	2
M12Q0 [110% - 125%]	2	198	1 / 1	1 / 1	18	60	0.0	10	0	0
	3	153	1 / 1	1 / 1	14	47	0.0	10	0	0
	4	237	2.7 / 3	3.7 / 4	25	104	0.0	10	0	0
	5	228	6.8 / 12	10.5 / 16	23	117	0.0	10	0	0
	-S	5695	-	-	835	2644	1.2	7	2	1
M12Q0 [125% - 140%]	2	224	1 / 1	1 / 1	18	59	0.0	10	0	0
	3	250	1 / 1	1 / 1	21	120	0.0	10	0	0
	4	361	2.9 / 3	3.9 / 4	37	220	0.0	10	0	0
	5	207	7.9 / 12	11.8 / 16	27	118	0.0	10	0	0
	-S	3771	-	-	512	1889	1.1	8	2	0
M24Q0 [95% - 110%]	2	193	1 / 1	1 / 1	35	119	0.0	10	0	0
	3	305	1 / 1	1 / 1	70	501	0.0	10	0	0
	4	453	2.9 / 3	3.9 / 4	143	1443	0.3	9	1	0
	5	415	9.6 / 12	13.5 / 16	105	1034	0.4	9	1	0
	-S	3293	-	-	901	2407	1.2	7	3	0
M24Q0 [110% - 125%]	2	364	1 / 1	1 / 1	64	261	0.0	10	0	0
	3	662	1 / 1	1 / 1	97	730	0.0	10	0	0
	4	536	3 / 3	4 / 4	123	1227	0.3	9	1	0
	5	648	11.7 / 12	15.7 / 16	108	1420	0.2	9	1	0
	-S	3068	-	-	723	2167	0.7	8	2	0
M24Q0 [125% - 140%]	2	313	1 / 1	1 / 1	49	232	0.0	10	0	0
	3	580	1 / 1	1 / 1	73	552	0.0	10	0	0
	4	1434	3 / 3	4 / 4	544	3371	0.4	8	1	1
	5	1793	12 / 12	16 / 16	1504	8821	0.2	9	0	1
	-S	3156	-	-	544	1816	0.5	8	2	0

APPENDIX B

Appendix for Chapter 4

B.1. List of Multiobjective LFATP Solutions

Table B.1 lists sets of solutions generated for the first 4 iterations of IWRA when solving the multiobjective LFATP. The list includes 85 sets and 340 solutions (including the repeated ones in multiple sets). For each solution, the table provides the weight vector (corresponding to the normalized problem) and the objective values of the corresponding solution for average medical proximity, average transport proximity, similarity and total sharpness count objectives, respectively.

Table B.1. List of LFATP solutions for the first 4 iterations of IWRA.

Set	Solution	Weight	Objective Values
$\mathcal{S}$	ℓ_1	(1.0000, 0.0000, 0.0000, 0.0000)	(4852.7, 2534.1, 49.7, 95)
	ℓ_2	(0.0000, 1.0000, 0.0000, 0.0000)	(5787.6, 2043.1, 33.3, 114)
	ℓ_3	(0.0000, 0.0000, 1.0000, 0.0000)	(5927.5, 2669.3, 100.0, 70)
	ℓ_4	(0.0000, 0.0000, 0.0000, 1.0000)	(6370.3, 2617.1, 57.4, 43)
$\mathcal{S}_1$	ℓ_{11}	(1.0000, 0.0000, 0.0000, 0.0000)	(4852.7, 2534.1, 49.7, 95)
	ℓ_{12}	(0.3889, 0.3889, 0.2222, 0.0000)	(5334.6, 2260.2, 72.3, 95)
	ℓ_{13}	(0.6578, 0.0467, 0.2956, 0.0000)	(5295.1, 2568.1, 80.2, 88)
	ℓ_{14}	(0.7044, 0.2956, 0.0000, 0.0000)	(4982.3, 2272.1, 41.1, 97)
$\mathcal{S}_2$	ℓ_{21}	(0.0000, 1.0000, 0.0000, 0.0000)	(5787.6, 2043.1, 33.3, 114)
	ℓ_{22}	(0.3561, 0.4306, 0.2133, 0.0000)	(5334.6, 2260.2, 72.3, 95)
	ℓ_{23}	(0.0240, 0.6680, 0.3080, 0.0000)	(5947.5, 2164.4, 67.7, 95)
	ℓ_{24}	(0.2847, 0.7153, 0.0000, 0.0000)	(5450.0, 2066.2, 33.8, 108)

continued ...

. . . continued

Set	Solution	Weight	Objective Values
$\mathcal{S}_3$	ℓ_{31}	(0.0000, 0.0000, 1.0000, 0.0000)	(5927.5, 2669.3, 100.0, 70)
	ℓ_{32}	(0.2643, 0.2643, 0.2643, 0.2070)	(5417.8, 2502.3, 64.5, 49)
	ℓ_{33}	(0.0000, 0.3678, 0.6322, 0.0000)	(5927.5, 2669.3, 100.0, 70)
	ℓ_{34}	(0.1136, 0.6322, 0.1771, 0.0771)	(5741.6, 2261.5, 75.1, 63)
$\mathcal{S}_4$	ℓ_{41}	(0.0000, 0.0000, 0.0000, 1.0000)	(6370.3, 2617.1, 57.4, 43)
	ℓ_{42}	(0.2500, 0.2500, 0.2500, 0.2500)	(5417.8, 2502.3, 64.5, 49)
	ℓ_{43}	(0.2505, 0.6250, 0.0000, 0.1245)	(5344.7, 2207.7, 40.3, 62)
	ℓ_{44}	(0.3750, 0.0000, 0.0000, 0.6250)	(5406.8, 2582.5, 58.3, 47)
$\mathcal{S}_{11}$	ℓ_{111}	(1.0000, 0.0000, 0.0000, 0.0000)	(4852.7, 2534.1, 49.7, 95)
	ℓ_{112}	(0.7395, 0.0351, 0.0351, 0.1902)	(5204.2, 2606.5, 56.1, 51)
	ℓ_{113}	(0.8098, 0.1054, 0.0847, 0.0000)	(4879.7, 2490.5, 53.1, 95)
	ℓ_{114}	(0.6168, 0.2002, 0.0180, 0.1651)	(5189.6, 2560.4, 51.0, 52)
$\mathcal{S}_{12}$	ℓ_{121}	(0.3889, 0.3889, 0.2222, 0.0000)	(5334.6, 2260.2, 72.3, 95)
	ℓ_{122}	(0.5770, 0.2348, 0.1881, 0.0000)	(5014.4, 2304.7, 57.2, 89)
	ℓ_{123}	(0.4306, 0.3748, 0.0755, 0.1191)	(5225.0, 2281.0, 53.1, 61)
	ℓ_{124}	(0.5316, 0.2308, 0.0987, 0.1389)	(5227.8, 2523.4, 63.0, 53)
$\mathcal{S}_{13}$	ℓ_{131}	(0.6578, 0.0467, 0.2956, 0.0000)	(5295.1, 2568.1, 80.2, 88)
	ℓ_{132}	(0.6252, 0.1094, 0.1094, 0.1559)	(5227.8, 2523.4, 63.0, 53)
	ℓ_{133}	(0.5542, 0.1814, 0.2432, 0.0212)	(5317.0, 2387.5, 77.4, 77)
	ℓ_{134}	(0.5272, 0.2316, 0.1037, 0.1375)	(5227.8, 2523.4, 63.0, 53)
$\mathcal{S}_{14}$	ℓ_{141}	(0.7044, 0.2956, 0.0000, 0.0000)	(4982.3, 2272.1, 41.1, 97)
	ℓ_{142}	(0.4847, 0.3773, 0.0000, 0.1380)	(5335.5, 2220.3, 45.8, 61)
	ℓ_{143}	(0.6352, 0.1941, 0.0000, 0.1708)	(5189.6, 2560.4, 51.0, 52)
	ℓ_{144}	(0.6307, 0.2267, 0.1426, 0.0000)	(5011.1, 2305.5, 57.0, 93)
$\mathcal{S}_{21}$	ℓ_{211}	(0.0000, 1.0000, 0.0000, 0.0000)	(5787.6, 2043.1, 33.3, 114)
	ℓ_{212}	(0.1399, 0.8601, 0.0000, 0.0000)	(5688.6, 2044.7, 34.9, 102)
	ℓ_{213}	(0.1163, 0.7675, 0.1163, 0.0000)	(5651.0, 2063.5, 43.2, 113)
	ℓ_{214}	(0.0775, 0.8450, 0.0775, 0.0000)	(5677.9, 2050.8, 38.5, 132)
$\mathcal{S}_{22}$	ℓ_{221}	(0.3561, 0.4306, 0.2133, 0.0000)	(5334.6, 2260.2, 72.3, 95)
	ℓ_{222}	(0.3661, 0.5076, 0.0156, 0.1107)	(5335.5, 2220.3, 45.8, 61)
	ℓ_{223}	(0.1901, 0.5493, 0.2606, 0.0000)	(5370.4, 2241.3, 71.3, 89)
	ℓ_{224}	(0.2135, 0.5727, 0.1181, 0.0957)	(5394.6, 2224.2, 56.8, 61)

continued . . .

. . . continued

Set	Solution	Weight	Objective Values
$\mathcal{S}_{23}$	ℓ_{231}	(0.0240, 0.6680, 0.3080, 0.0000)	(5947.5, 2164.4, 67.7, 95)
	ℓ_{232}	(0.1130, 0.7346, 0.0225, 0.1299)	(5735.2, 2185.6, 44.4, 59)
	ℓ_{233}	(0.1770, 0.6475, 0.1755, 0.0000)	(5574.3, 2130.9, 57.6, 101)
	ℓ_{234}	(0.0000, 0.8635, 0.0075, 0.1289)	(6105.8, 2158.6, 36.9, 61)
$\mathcal{S}_{24}$	ℓ_{241}	(0.2847, 0.7153, 0.0000, 0.0000)	(5450.0, 2066.2, 33.8, 108)
	ℓ_{242}	(0.5551, 0.4449, 0.0000, 0.0000)	(5064.1, 2204.7, 39.9, 117)
	ℓ_{243}	(0.3651, 0.5199, 0.0000, 0.1150)	(5335.5, 2220.3, 45.8, 61)
	ℓ_{244}	(0.1104, 0.7534, 0.0000, 0.1362)	(5735.2, 2185.6, 44.4, 59)
$\mathcal{S}_{31}$	ℓ_{311}	(0.0000, 0.0000, 1.0000, 0.0000)	(5927.5, 2669.3, 100.0, 70)
	ℓ_{312}	(0.0000, 0.0000, 0.5957, 0.4043)	(5877.7, 2679.6, 93.0, 60)
	ℓ_{313}	(0.3733, 0.0000, 0.6267, 0.0000)	(5927.5, 2669.3, 100.0, 70)
	ℓ_{314}	(0.6322, 0.0838, 0.2512, 0.0329)	(5150.4, 2442.7, 69.7, 73)
$\mathcal{S}_{32}$	ℓ_{321}	(0.2643, 0.2643, 0.2643, 0.2070)	(5417.8, 2502.3, 64.5, 49)
	ℓ_{322}	(0.0742, 0.0000, 0.5451, 0.3807)	(5877.7, 2679.6, 93.0, 60)
	ℓ_{323}	(0.6244, 0.0000, 0.1697, 0.2059)	(5299.8, 2608.5, 61.1, 49)
	ℓ_{324}	(0.0000, 0.3408, 0.4010, 0.2582)	(6299.0, 2473.0, 73.9, 51)
$\mathcal{S}_{33}$	ℓ_{331}	(0.0000, 0.3678, 0.6322, 0.0000)	(5927.5, 2669.3, 100.0, 70)
	ℓ_{332}	(0.0000, 0.0000, 0.5957, 0.4043)	(5877.7, 2679.6, 93.0, 60)
	ℓ_{333}	(0.3733, 0.0000, 0.6267, 0.0000)	(5927.5, 2669.3, 100.0, 70)
	ℓ_{334}	(0.6322, 0.0838, 0.2512, 0.0329)	(5150.4, 2442.7, 69.7, 73)
$\mathcal{S}_{34}$	ℓ_{341}	(0.1136, 0.6322, 0.1771, 0.0771)	(5741.6, 2261.5, 75.1, 63)
	ℓ_{342}	(0.5622, 0.1717, 0.2406, 0.0254)	(5317.0, 2387.5, 77.4, 77)
	ℓ_{343}	(0.0000, 0.3476, 0.3971, 0.2553)	(6299.0, 2473.0, 73.9, 51)
	ℓ_{344}	(0.3477, 0.4548, 0.0986, 0.0989)	(5386.0, 2226.4, 56.8, 61)
$\mathcal{S}_{41}$	ℓ_{411}	(0.0000, 0.0000, 0.0000, 1.0000)	(6370.3, 2617.1, 57.4, 43)
	ℓ_{412}	(0.0000, 0.0000, 0.5813, 0.4187)	(6652.1, 2677.0, 80.0, 52)
	ℓ_{413}	(0.0000, 0.5710, 0.0000, 0.4290)	(6503.1, 2493.5, 56.2, 43)
	ℓ_{414}	(0.0000, 0.2219, 0.3313, 0.4468)	(6499.9, 2511.2, 59.6, 43)
$\mathcal{S}_{42}$	ℓ_{421}	(0.2500, 0.2500, 0.2500, 0.2500)	(5417.8, 2502.3, 64.5, 49)
	ℓ_{422}	(0.0742, 0.0000, 0.5451, 0.3807)	(5877.7, 2679.6, 93.0, 60)
	ℓ_{423}	(0.6264, 0.0268, 0.1530, 0.1937)	(5299.8, 2608.5, 61.1, 49)
	ℓ_{424}	(0.4604, 0.3498, 0.0000, 0.1898)	(5242.1, 2335.3, 46.1, 58)

continued . . .

. . . continued

Set	Solution	Weight	Objective Values
$\mathcal{S}_{43}$	ℓ_{431}	(0.2505, 0.6250, 0.0000, 0.1245)	(5344.7, 2207.7, 40.3, 62)
	ℓ_{432}	(0.5730, 0.2697, 0.0000, 0.1572)	(5112.4, 2436.9, 46.6, 58)
	ℓ_{433}	(0.0000, 0.6880, 0.0000, 0.3120)	(6498.0, 2345.0, 48.6, 49)
	ℓ_{434}	(0.3577, 0.4097, 0.0957, 0.1369)	(5386.0, 2226.4, 56.8, 61)
$\mathcal{S}_{44}$	ℓ_{441}	(0.3750, 0.0000, 0.0000, 0.6250)	(5406.8, 2582.5, 58.3, 47)
	ℓ_{442}	(0.6181, 0.1389, 0.0000, 0.2431)	(5293.4, 2597.0, 58.9, 49)
	ℓ_{443}	(0.1441, 0.4641, 0.0000, 0.3918)	(6503.1, 2493.5, 56.2, 43)
	ℓ_{444}	(0.4654, 0.0000, 0.2403, 0.2943)	(5413.2, 2594.0, 60.5, 47)
$\mathcal{S}_{111}$	ℓ_{1111}	(1.0000, 0.0000, 0.0000, 0.0000)	(4852.7, 2534.1, 49.7, 95)
	ℓ_{1112}	(0.8946, 0.0000, 0.1054, 0.0000)	(4861.6, 2531.1, 53.0, 91)
	ℓ_{1113}	(0.9049, 0.0951, 0.0000, 0.0000)	(4868.9, 2466.2, 42.7, 101)
	ℓ_{1114}	(0.9121, 0.0072, 0.0175, 0.0632)	(4925.7, 2762.5, 50.9, 73)
$\mathcal{S}_{112}$	ℓ_{1121}	(0.7395, 0.0351, 0.0351, 0.1902)	(5204.2, 2606.5, 56.1, 51)
	ℓ_{1122}	(0.6739, 0.1203, 0.1597, 0.0460)	(5012.2, 2434.6, 58.3, 69)
	ℓ_{1123}	(0.8698, 0.0000, 0.0000, 0.1302)	(5173.8, 2589.4, 50.7, 52)
	ℓ_{1124}	(0.7275, 0.0000, 0.1538, 0.1187)	(5140.5, 2561.8, 58.9, 55)
$\mathcal{S}_{113}$	ℓ_{1131}	(0.8098, 0.1054, 0.0847, 0.0000)	(4879.7, 2490.5, 53.1, 95)
	ℓ_{1132}	(0.6643, 0.1856, 0.1501, 0.0000)	(4999.0, 2306.1, 55.7, 101)
	ℓ_{1133}	(0.7866, 0.0000, 0.2134, 0.0000)	(4932.5, 2639.9, 58.9, 90)
	ℓ_{1134}	(0.7071, 0.1929, 0.0474, 0.0527)	(5005.8, 2423.1, 56.2, 69)
$\mathcal{S}_{114}$	ℓ_{1141}	(0.6168, 0.2002, 0.0180, 0.1651)	(5189.6, 2560.4, 51.0, 52)
	ℓ_{1142}	(0.8632, 0.0130, 0.0000, 0.1237)	(5099.2, 2806.7, 51.2, 57)
	ℓ_{1143}	(0.6941, 0.1640, 0.0901, 0.0518)	(5040.2, 2401.9, 54.5, 65)
	ℓ_{1144}	(0.7968, 0.1081, 0.0000, 0.0951)	(5068.0, 2399.6, 46.9, 62)
$\mathcal{S}_{121}$	ℓ_{1211}	(0.3889, 0.3889, 0.2222, 0.0000)	(5334.6, 2260.2, 72.3, 95)
	ℓ_{1212}	(0.4936, 0.3395, 0.1669, 0.0000)	(5328.6, 2263.3, 72.3, 99)
	ℓ_{1213}	(0.4932, 0.2561, 0.2507, 0.0000)	(5293.3, 2318.2, 74.4, 103)
	ℓ_{1214}	(0.4359, 0.4364, 0.1277, 0.0000)	(5207.1, 2208.0, 58.8, 103)
$\mathcal{S}_{122}$	ℓ_{1221}	(0.5770, 0.2348, 0.1881, 0.0000)	(5014.4, 2304.7, 57.2, 89)
	ℓ_{1222}	(0.5080, 0.3808, 0.1112, 0.0000)	(5004.2, 2288.6, 53.6, 97)
	ℓ_{1223}	(0.4130, 0.4513, 0.1117, 0.0241)	(5297.9, 2178.3, 55.4, 82)
	ℓ_{1224}	(0.4830, 0.2772, 0.2040, 0.0359)	(5331.6, 2285.1, 68.6, 69)

continued . . .

. . . continued

Set	Solution	Weight	Objective Values
$\mathcal{S}_{123}$	ℓ_{1231}	(0.4306, 0.3748, 0.0755, 0.1191)	(5225.0, 2281.0, 53.1, 61)
	ℓ_{1232}	(0.3429, 0.4999, 0.0553, 0.1019)	(5364.2, 2207.2, 51.4, 62)
	ℓ_{1233}	(0.5246, 0.3288, 0.1106, 0.0360)	(5144.5, 2262.7, 53.4, 69)
	ℓ_{1234}	(0.4354, 0.4490, 0.0890, 0.0266)	(5126.1, 2250.6, 52.2, 73)
$\mathcal{S}_{124}$	ℓ_{1241}	(0.5316, 0.2308, 0.0987, 0.1389)	(5227.8, 2523.4, 63.0, 53)
	ℓ_{1242}	(0.5919, 0.1878, 0.1486, 0.0717)	(5131.8, 2397.1, 58.5, 60)
	ℓ_{1243}	(0.5064, 0.2481, 0.1817, 0.0639)	(5261.8, 2335.8, 62.2, 60)
	ℓ_{1244}	(0.5771, 0.1808, 0.2026, 0.0395)	(5150.4, 2442.7, 69.7, 73)
$\mathcal{S}_{131}$	ℓ_{1311}	(0.6578, 0.0467, 0.2956, 0.0000)	(5295.1, 2568.1, 80.2, 88)
	ℓ_{1312}	(0.7703, 0.0000, 0.2297, 0.0000)	(4981.3, 2529.4, 62.5, 91)
	ℓ_{1313}	(0.6988, 0.0855, 0.2101, 0.0056)	(4986.0, 2523.6, 63.0, 89)
	ℓ_{1314}	(0.6289, 0.1134, 0.2578, 0.0000)	(5171.9, 2394.3, 72.2, 89)
$\mathcal{S}_{132}$	ℓ_{1321}	(0.6252, 0.1094, 0.1094, 0.1559)	(5227.8, 2523.4, 63.0, 53)
	ℓ_{1322}	(0.6747, 0.0981, 0.1831, 0.0441)	(5012.2, 2434.6, 58.3, 69)
	ℓ_{1323}	(0.7290, 0.0000, 0.1975, 0.0735)	(5003.0, 2800.5, 58.9, 65)
	ℓ_{1324}	(0.6005, 0.1816, 0.1561, 0.0618)	(5012.2, 2434.6, 58.3, 69)
$\mathcal{S}_{133}$	ℓ_{1331}	(0.5542, 0.1814, 0.2432, 0.0212)	(5317.0, 2387.5, 77.4, 77)
	ℓ_{1332}	(0.6721, 0.1038, 0.1857, 0.0384)	(5012.2, 2434.6, 58.3, 69)
	ℓ_{1333}	(0.7475, 0.0140, 0.2075, 0.0309)	(4954.2, 2756.1, 58.9, 75)
	ℓ_{1334}	(0.6307, 0.1152, 0.2541, 0.0000)	(5171.9, 2394.3, 72.2, 89)
$\mathcal{S}_{134}$	ℓ_{1341}	(0.5272, 0.2316, 0.1037, 0.1375)	(5227.8, 2523.4, 63.0, 53)
	ℓ_{1342}	(0.6747, 0.0981, 0.1831, 0.0441)	(5012.2, 2434.6, 58.3, 69)
	ℓ_{1343}	(0.7290, 0.0000, 0.1975, 0.0735)	(5003.0, 2800.5, 58.9, 65)
	ℓ_{1344}	(0.6005, 0.1816, 0.1561, 0.0618)	(5012.2, 2434.6, 58.3, 69)
$\mathcal{S}_{141}$	ℓ_{1411}	(0.7044, 0.2956, 0.0000, 0.0000)	(4982.3, 2272.1, 41.1, 97)
	ℓ_{1412}	(0.5688, 0.3714, 0.0597, 0.0000)	(4996.5, 2268.5, 49.2, 107)
	ℓ_{1413}	(0.8277, 0.1723, 0.0000, 0.0000)	(4878.8, 2436.2, 43.2, 98)
	ℓ_{1414}	(0.7306, 0.1985, 0.0158, 0.0551)	(5033.8, 2390.5, 52.4, 65)
$\mathcal{S}_{142}$	ℓ_{1421}	(0.4847, 0.3773, 0.0000, 0.1380)	(5335.5, 2220.3, 45.8, 61)
	ℓ_{1422}	(0.5990, 0.3225, 0.0372, 0.0413)	(5144.5, 2262.7, 53.4, 69)
	ℓ_{1423}	(0.5004, 0.3551, 0.1038, 0.0406)	(5144.5, 2262.7, 53.4, 69)
	ℓ_{1424}	(0.4750, 0.4503, 0.0498, 0.0249)	(5126.1, 2250.6, 52.2, 73)

continued . . .

. . . continued

Set	Solution	Weight	Objective Values
$\mathcal{S}_{143}$	ℓ_{1431}	(0.6352, 0.1941, 0.0000, 0.1708)	(5189.6, 2560.4, 51.0, 52)
	ℓ_{1432}	(0.7652, 0.1793, 0.0000, 0.0554)	(5033.8, 2390.5, 52.4, 65)
	ℓ_{1433}	(0.6770, 0.2073, 0.0543, 0.0614)	(5005.8, 2423.1, 56.2, 69)
	ℓ_{1434}	(0.6081, 0.2203, 0.1027, 0.0689)	(5131.8, 2397.1, 58.5, 60)
$\mathcal{S}_{144}$	ℓ_{1441}	(0.6307, 0.2267, 0.1426, 0.0000)	(5011.1, 2305.5, 57.0, 93)
	ℓ_{1442}	(0.5674, 0.3637, 0.0689, 0.0000)	(4996.5, 2268.5, 49.2, 107)
	ℓ_{1443}	(0.4543, 0.4430, 0.0951, 0.0077)	(5001.4, 2274.3, 50.6, 91)
	ℓ_{1444}	(0.7453, 0.1851, 0.0157, 0.0539)	(5033.8, 2390.5, 52.4, 65)
$\mathcal{S}_{211}$	ℓ_{2111}	(0.0000, 1.0000, 0.0000, 0.0000)	(5787.6, 2043.1, 33.3, 114)
	ℓ_{2112}	(0.0000, 0.9524, 0.0476, 0.0000)	(5719.6, 2043.6, 35.9, 110)
	ℓ_{2113}	(0.0315, 0.9604, 0.0081, 0.0000)	(5720.3, 2043.5, 35.2, 106)
	ℓ_{2114}	(0.0047, 0.9732, 0.0208, 0.0013)	(5720.3, 2043.5, 35.2, 106)
$\mathcal{S}_{212}$	ℓ_{2121}	(0.1399, 0.8601, 0.0000, 0.0000)	(5688.6, 2044.7, 34.9, 102)
	ℓ_{2122}	(0.0056, 0.9225, 0.0000, 0.0719)	(5686.8, 2078.7, 38.4, 76)
	ℓ_{2123}	(0.0000, 0.8558, 0.1320, 0.0122)	(5666.9, 2069.1, 42.4, 87)
	ℓ_{2124}	(0.0000, 0.9279, 0.0721, 0.0000)	(5800.7, 2051.8, 39.9, 124)
$\mathcal{S}_{213}$	ℓ_{2131}	(0.1163, 0.7675, 0.1163, 0.0000)	(5651.0, 2063.5, 43.2, 113)
	ℓ_{2132}	(0.0000, 0.8529, 0.1471, 0.0000)	(6036.1, 2069.0, 49.1, 113)
	ℓ_{2133}	(0.0582, 0.8062, 0.1357, 0.0000)	(6000.1, 2062.5, 47.0, 112)
	ℓ_{2134}	(0.0288, 0.8712, 0.0985, 0.0016)	(5739.2, 2051.0, 38.8, 105)
$\mathcal{S}_{214}$	ℓ_{2141}	(0.0775, 0.8450, 0.0775, 0.0000)	(5677.9, 2050.8, 38.5, 132)
	ℓ_{2142}	(0.0000, 0.9225, 0.0775, 0.0000)	(6007.9, 2060.5, 45.9, 116)
	ℓ_{2143}	(0.0208, 0.8658, 0.1134, 0.0000)	(6000.1, 2062.5, 47.0, 112)
	ℓ_{2144}	(0.1282, 0.8145, 0.0573, 0.0000)	(5634.8, 2048.7, 36.2, 114)
$\mathcal{S}_{221}$	ℓ_{2211}	(0.3561, 0.4306, 0.2133, 0.0000)	(5334.6, 2260.2, 72.3, 95)
	ℓ_{2212}	(0.4737, 0.4391, 0.0873, 0.0000)	(5040.9, 2233.9, 45.8, 109)
	ℓ_{2213}	(0.4030, 0.4628, 0.1342, 0.0000)	(5207.1, 2208.0, 58.8, 103)
	ℓ_{2214}	(0.2940, 0.4750, 0.2310, 0.0000)	(5370.4, 2241.3, 71.3, 89)
$\mathcal{S}_{222}$	ℓ_{2221}	(0.3661, 0.5076, 0.0156, 0.1107)	(5335.5, 2220.3, 45.8, 61)
	ℓ_{2222}	(0.4414, 0.4633, 0.0567, 0.0386)	(5126.1, 2250.6, 52.2, 73)
	ℓ_{2223}	(0.3902, 0.4982, 0.0516, 0.0600)	(5364.2, 2207.2, 51.4, 62)
	ℓ_{2224}	(0.3554, 0.5219, 0.0482, 0.0745)	(5364.2, 2207.2, 51.4, 62)

continued . . .

. . . continued

Set	Solution	Weight	Objective Values
$\mathcal{S}_{223}$	ℓ_{2231}	(0.1901, 0.5493, 0.2606, 0.0000)	(5370.4, 2241.3, 71.3, 89)
	ℓ_{2232}	(0.4799, 0.4407, 0.0777, 0.0017)	(5146.8, 2193.0, 47.1, 108)
	ℓ_{2233}	(0.3823, 0.5129, 0.0959, 0.0089)	(5265.9, 2156.2, 51.1, 97)
	ℓ_{2234}	(0.2425, 0.6196, 0.1379, 0.0000)	(5478.0, 2117.5, 52.6, 101)
$\mathcal{S}_{224}$	ℓ_{2241}	(0.2135, 0.5727, 0.1181, 0.0957)	(5394.6, 2224.2, 56.8, 61)
	ℓ_{2242}	(0.4675, 0.4480, 0.0709, 0.0136)	(5103.4, 2236.1, 49.6, 86)
	ℓ_{2243}	(0.3775, 0.4942, 0.1078, 0.0205)	(5222.4, 2201.8, 54.3, 79)
	ℓ_{2244}	(0.2955, 0.5742, 0.0981, 0.0322)	(5551.4, 2119.8, 50.4, 75)
$\mathcal{S}_{231}$	ℓ_{2311}	(0.0240, 0.6680, 0.3080, 0.0000)	(5947.5, 2164.4, 67.7, 95)
	ℓ_{2312}	(0.0000, 0.8364, 0.1636, 0.0000)	(6013.2, 2069.9, 49.2, 111)
	ℓ_{2313}	(0.0146, 0.7254, 0.1969, 0.0631)	(6055.9, 2166.2, 62.8, 65)
	ℓ_{2314}	(0.1067, 0.6352, 0.2582, 0.0000)	(5702.2, 2211.4, 71.7, 91)
$\mathcal{S}_{232}$	ℓ_{2321}	(0.1130, 0.7346, 0.0225, 0.1299)	(5735.2, 2185.6, 44.4, 59)
	ℓ_{2322}	(0.0000, 0.8171, 0.1351, 0.0478)	(6055.9, 2162.9, 62.1, 65)
	ℓ_{2323}	(0.0146, 0.7054, 0.1963, 0.0838)	(6055.9, 2166.2, 62.8, 65)
	ℓ_{2324}	(0.2125, 0.6227, 0.0509, 0.1139)	(5364.2, 2207.2, 51.4, 62)
$\mathcal{S}_{233}$	ℓ_{2331}	(0.1770, 0.6475, 0.1755, 0.0000)	(5574.3, 2130.9, 57.6, 101)
	ℓ_{2332}	(0.0200, 0.8301, 0.1499, 0.0000)	(6036.1, 2069.0, 49.1, 113)
	ℓ_{2333}	(0.1160, 0.7435, 0.1405, 0.0000)	(5678.4, 2063.8, 43.8, 113)
	ℓ_{2334}	(0.1683, 0.6801, 0.1121, 0.0395)	(5569.8, 2131.9, 51.6, 71)
$\mathcal{S}_{234}$	ℓ_{2341}	(0.0000, 0.8635, 0.0075, 0.1289)	(6105.8, 2158.6, 36.9, 61)
	ℓ_{2342}	(0.0000, 0.8360, 0.1201, 0.0438)	(6019.4, 2156.0, 61.5, 67)
	ℓ_{2343}	(0.0000, 0.8144, 0.0746, 0.1110)	(5807.2, 2132.3, 46.9, 65)
	ℓ_{2344}	(0.0497, 0.8058, 0.0045, 0.1400)	(6098.7, 2188.9, 45.7, 57)
$\mathcal{S}_{241}$	ℓ_{2411}	(0.2847, 0.7153, 0.0000, 0.0000)	(5450.0, 2066.2, 33.8, 108)
	ℓ_{2412}	(0.1346, 0.8499, 0.0143, 0.0013)	(5688.6, 2044.7, 34.9, 102)
	ℓ_{2413}	(0.4111, 0.5889, 0.0000, 0.0000)	(5165.1, 2165.2, 35.8, 117)
	ℓ_{2414}	(0.2346, 0.6439, 0.1216, 0.0000)	(5475.5, 2116.0, 52.3, 109)
$\mathcal{S}_{242}$	ℓ_{2421}	(0.5551, 0.4449, 0.0000, 0.0000)	(5064.1, 2204.7, 39.9, 117)
	ℓ_{2422}	(0.4404, 0.4725, 0.0871, 0.0000)	(5062.1, 2218.1, 44.2, 113)
	ℓ_{2423}	(0.4680, 0.5320, 0.0000, 0.0000)	(5405.7, 2081.2, 33.6, 112)
	ℓ_{2424}	(0.3663, 0.5265, 0.1073, 0.0000)	(5265.6, 2147.7, 51.3, 105)

continued . . .

. . . continued

Set	Solution	Weight	Objective Values
$\mathcal{S}_{243}$	ℓ_{2431}	(0.3651, 0.5199, 0.0000, 0.1150)	(5335.5, 2220.3, 45.8, 61)
	ℓ_{2432}	(0.2144, 0.6547, 0.1014, 0.0294)	(5551.4, 2119.8, 50.4, 75)
	ℓ_{2433}	(0.4556, 0.4560, 0.0693, 0.0191)	(5222.4, 2201.8, 54.3, 79)
	ℓ_{2434}	(0.3479, 0.6170, 0.0000, 0.0351)	(5370.9, 2106.9, 37.4, 82)
$\mathcal{S}_{244}$	ℓ_{2441}	(0.1104, 0.7534, 0.0000, 0.1362)	(5735.2, 2185.6, 44.4, 59)
	ℓ_{2442}	(0.0067, 0.8809, 0.0461, 0.0663)	(5674.5, 2103.9, 42.0, 70)
	ℓ_{2443}	(0.1101, 0.8411, 0.0000, 0.0488)	(5674.4, 2068.3, 38.4, 80)
	ℓ_{2444}	(0.0764, 0.7854, 0.1036, 0.0346)	(5645.9, 2080.1, 42.8, 79)
$\mathcal{S}_{311}$	ℓ_{3111}	(0.0000, 0.0000, 1.0000, 0.0000)	(5927.5, 2669.3, 100.0, 70)
	ℓ_{3112}	(0.1091, 0.4853, 0.4055, 0.0000)	(5666.2, 2331.0, 81.5, 81)
	ℓ_{3113}	(0.0000, 0.0000, 0.7791, 0.2209)	(5951.7, 2659.4, 98.2, 66)
	ℓ_{3114}	(0.4273, 0.0414, 0.4224, 0.1089)	(5880.3, 2678.3, 94.6, 62)
$\mathcal{S}_{312}$	ℓ_{3121}	(0.0000, 0.0000, 0.5957, 0.4043)	(5877.7, 2679.6, 93.0, 60)
	ℓ_{3122}	(0.0000, 0.2421, 0.4997, 0.2582)	(5877.7, 2679.6, 93.0, 60)
	ℓ_{3123}	(0.3200, 0.0000, 0.4343, 0.2458)	(5877.7, 2679.6, 93.0, 60)
	ℓ_{3124}	(0.0000, 0.0000, 0.7689, 0.2311)	(5951.7, 2659.4, 98.2, 66)
$\mathcal{S}_{313}$	ℓ_{3131}	(0.3733, 0.0000, 0.6267, 0.0000)	(5927.5, 2669.3, 100.0, 70)
	ℓ_{3132}	(0.1091, 0.4853, 0.4055, 0.0000)	(5666.2, 2331.0, 81.5, 81)
	ℓ_{3133}	(0.0000, 0.0000, 0.7791, 0.2209)	(5951.7, 2659.4, 98.2, 66)
	ℓ_{3134}	(0.4273, 0.0414, 0.4224, 0.1089)	(5880.3, 2678.3, 94.6, 62)
$\mathcal{S}_{314}$	ℓ_{3141}	(0.6322, 0.0838, 0.2512, 0.0329)	(5150.4, 2442.7, 69.7, 73)
	ℓ_{3142}	(0.4485, 0.2680, 0.2836, 0.0000)	(5311.7, 2330.3, 75.6, 99)
	ℓ_{3143}	(0.4989, 0.1347, 0.2965, 0.0699)	(5514.4, 2477.4, 79.0, 61)
	ℓ_{3144}	(0.6546, 0.0000, 0.3454, 0.0000)	(5295.1, 2568.1, 80.2, 88)
$\mathcal{S}_{321}$	ℓ_{3211}	(0.2643, 0.2643, 0.2643, 0.2070)	(5417.8, 2502.3, 64.5, 49)
	ℓ_{3212}	(0.3786, 0.4164, 0.0995, 0.1055)	(5386.0, 2226.4, 56.8, 61)
	ℓ_{3213}	(0.4092, 0.0000, 0.3508, 0.2400)	(5751.5, 2554.5, 82.2, 55)
	ℓ_{3214}	(0.4641, 0.1998, 0.2707, 0.0655)	(5483.9, 2431.7, 80.2, 67)
$\mathcal{S}_{322}$	ℓ_{3221}	(0.0742, 0.0000, 0.5451, 0.3807)	(5877.7, 2679.6, 93.0, 60)
	ℓ_{3222}	(0.3196, 0.0000, 0.4336, 0.2468)	(5877.7, 2679.6, 93.0, 60)
	ℓ_{3223}	(0.4627, 0.1984, 0.2729, 0.0660)	(5483.9, 2431.7, 80.2, 67)
	ℓ_{3224}	(0.1741, 0.1741, 0.4158, 0.2360)	(5877.7, 2679.6, 93.0, 60)

continued . . .

. . . continued

Set	Solution	Weight	Objective Values
$\mathcal{S}_{323}$	ℓ_{3231}	(0.6244, 0.0000, 0.1697, 0.2059)	(5299.8, 2608.5, 61.1, 49)
	ℓ_{3232}	(0.4249, 0.0000, 0.3058, 0.2693)	(5413.2, 2594.0, 60.5, 47)
	ℓ_{3233}	(0.6987, 0.0000, 0.2613, 0.0399)	(5065.8, 2776.3, 63.3, 67)
	ℓ_{3234}	(0.5512, 0.0722, 0.2860, 0.0906)	(5227.8, 2523.4, 63.0, 53)
$\mathcal{S}_{324}$	ℓ_{3241}	(0.0000, 0.3408, 0.4010, 0.2582)	(6299.0, 2473.0, 73.9, 51)
	ℓ_{3242}	(0.1362, 0.3306, 0.3330, 0.2003)	(5719.0, 2429.6, 77.2, 55)
	ℓ_{3243}	(0.0750, 0.2556, 0.3985, 0.2709)	(5751.5, 2554.5, 82.2, 55)
	ℓ_{3244}	(0.0681, 0.3600, 0.3436, 0.2284)	(5719.0, 2429.6, 77.2, 55)
$\mathcal{S}_{331}$	ℓ_{3311}	(0.0000, 0.3678, 0.6322, 0.0000)	(5927.5, 2669.3, 100.0, 70)
	ℓ_{3312}	(0.1091, 0.4853, 0.4055, 0.0000)	(5666.2, 2331.0, 81.5, 81)
	ℓ_{3313}	(0.0000, 0.0000, 0.7791, 0.2209)	(5951.7, 2659.4, 98.2, 66)
	ℓ_{3314}	(0.4273, 0.0414, 0.4224, 0.1089)	(5880.3, 2678.3, 94.6, 62)
$\mathcal{S}_{332}$	ℓ_{3321}	(0.0000, 0.0000, 0.5957, 0.4043)	(5877.7, 2679.6, 93.0, 60)
	ℓ_{3322}	(0.0000, 0.2421, 0.4997, 0.2582)	(5877.7, 2679.6, 93.0, 60)
	ℓ_{3323}	(0.3200, 0.0000, 0.4343, 0.2458)	(5877.7, 2679.6, 93.0, 60)
	ℓ_{3324}	(0.0000, 0.0000, 0.7689, 0.2311)	(5951.7, 2659.4, 98.2, 66)
$\mathcal{S}_{333}$	ℓ_{3331}	(0.3733, 0.0000, 0.6267, 0.0000)	(5927.5, 2669.3, 100.0, 70)
	ℓ_{3332}	(0.1091, 0.4853, 0.4055, 0.0000)	(5666.2, 2331.0, 81.5, 81)
	ℓ_{3333}	(0.0000, 0.0000, 0.7791, 0.2209)	(5951.7, 2659.4, 98.2, 66)
	ℓ_{3334}	(0.4273, 0.0414, 0.4224, 0.1089)	(5880.3, 2678.3, 94.6, 62)
$\mathcal{S}_{334}$	ℓ_{3341}	(0.6322, 0.0838, 0.2512, 0.0329)	(5150.4, 2442.7, 69.7, 73)
	ℓ_{3342}	(0.4485, 0.2680, 0.2836, 0.0000)	(5311.7, 2330.3, 75.6, 99)
	ℓ_{3343}	(0.4989, 0.1347, 0.2965, 0.0699)	(5514.4, 2477.4, 79.0, 61)
	ℓ_{3344}	(0.6546, 0.0000, 0.3454, 0.0000)	(5295.1, 2568.1, 80.2, 88)
$\mathcal{S}_{341}$	ℓ_{3411}	(0.1136, 0.6322, 0.1771, 0.0771)	(5741.6, 2261.5, 75.1, 63)
	ℓ_{3412}	(0.1437, 0.4632, 0.3931, 0.0000)	(5661.5, 2332.9, 81.6, 81)
	ℓ_{3413}	(0.1586, 0.4736, 0.2169, 0.1509)	(5708.7, 2332.8, 79.0, 61)
	ℓ_{3414}	(0.0000, 0.6099, 0.3901, 0.0000)	(5966.3, 2237.5, 74.8, 75)
$\mathcal{S}_{342}$	ℓ_{3421}	(0.5622, 0.1717, 0.2406, 0.0254)	(5317.0, 2387.5, 77.4, 77)
	ℓ_{3422}	(0.4092, 0.2931, 0.2977, 0.0000)	(5311.7, 2330.3, 75.6, 99)
	ℓ_{3423}	(0.2001, 0.4270, 0.3728, 0.0000)	(5641.3, 2304.2, 79.8, 81)
	ℓ_{3424}	(0.2778, 0.4851, 0.2371, 0.0000)	(5370.4, 2241.3, 71.3, 89)

continued . . .

. . . continued

Set	Solution	Weight	Objective Values
$\mathcal{S}_{343}$	ℓ_{3431}	(0.0000, 0.3476, 0.3971, 0.2553)	(6299.0, 2473.0, 73.9, 51)
	ℓ_{3432}	(0.1390, 0.2979, 0.3797, 0.1833)	(5831.3, 2537.0, 84.5, 57)
	ℓ_{3433}	(0.1006, 0.3985, 0.2993, 0.2015)	(5719.0, 2429.6, 77.2, 55)
	ℓ_{3434}	(0.0648, 0.3535, 0.4071, 0.1745)	(5952.5, 2491.2, 85.2, 59)
$\mathcal{S}_{344}$	ℓ_{3441}	(0.3477, 0.4548, 0.0986, 0.0989)	(5386.0, 2226.4, 56.8, 61)
	ℓ_{3442}	(0.2144, 0.4989, 0.1643, 0.1223)	(5401.0, 2235.7, 59.0, 61)
	ℓ_{3443}	(0.4263, 0.2731, 0.2180, 0.0826)	(5385.6, 2317.0, 66.8, 59)
	ℓ_{3444}	(0.3792, 0.3765, 0.1993, 0.0451)	(5355.8, 2275.2, 66.8, 65)
$\mathcal{S}_{411}$	ℓ_{4111}	(0.0000, 0.0000, 0.0000, 1.0000)	(6370.3, 2617.1, 57.4, 43)
	ℓ_{4112}	(0.0773, 0.0000, 0.0955, 0.8273)	(6370.3, 2617.1, 57.4, 43)
	ℓ_{4113}	(0.2225, 0.0000, 0.2751, 0.5024)	(5420.0, 2594.3, 60.5, 47)
	ℓ_{4114}	(0.2261, 0.0000, 0.0237, 0.7502)	(6370.3, 2617.1, 57.4, 43)
$\mathcal{S}_{412}$	ℓ_{4121}	(0.0000, 0.0000, 0.5813, 0.4187)	(6652.1, 2677.0, 80.0, 52)
	ℓ_{4122}	(0.0000, 0.1683, 0.4996, 0.3322)	(5877.7, 2679.6, 93.0, 60)
	ℓ_{4123}	(0.1184, 0.0319, 0.4449, 0.4048)	(5751.5, 2554.5, 82.2, 55)
	ℓ_{4124}	(0.0142, 0.0723, 0.4772, 0.4363)	(6525.9, 2551.8, 69.2, 47)
$\mathcal{S}_{413}$	ℓ_{4131}	(0.0000, 0.5710, 0.0000, 0.4290)	(6503.1, 2493.5, 56.2, 43)
	ℓ_{4132}	(0.0000, 0.3149, 0.0806, 0.6046)	(6493.5, 2499.7, 57.4, 43)
	ℓ_{4133}	(0.0521, 0.1170, 0.0289, 0.8021)	(6493.5, 2499.7, 57.4, 43)
	ℓ_{4134}	(0.1858, 0.2872, 0.0000, 0.5270)	(6503.1, 2493.5, 56.2, 43)
$\mathcal{S}_{414}$	ℓ_{4141}	(0.0000, 0.2219, 0.3313, 0.4468)	(6499.9, 2511.2, 59.6, 43)
	ℓ_{4142}	(0.0000, 0.0000, 0.3032, 0.6968)	(6499.9, 2511.2, 59.6, 43)
	ℓ_{4143}	(0.0000, 0.1968, 0.0532, 0.7500)	(6499.9, 2511.2, 59.6, 43)
	ℓ_{4144}	(0.0000, 0.4586, 0.2367, 0.3047)	(6273.0, 2432.4, 64.3, 47)
$\mathcal{S}_{421}$	ℓ_{4211}	(0.2500, 0.2500, 0.2500, 0.2500)	(5417.8, 2502.3, 64.5, 49)
	ℓ_{4212}	(0.0000, 0.5390, 0.2877, 0.1733)	(5708.7, 2332.8, 79.0, 61)
	ℓ_{4213}	(0.1428, 0.4645, 0.0031, 0.3896)	(6503.1, 2493.5, 56.2, 43)
	ℓ_{4214}	(0.0064, 0.2954, 0.3784, 0.3198)	(6525.9, 2551.8, 69.2, 47)
$\mathcal{S}_{422}$	ℓ_{4221}	(0.0742, 0.0000, 0.5451, 0.3807)	(5877.7, 2679.6, 93.0, 60)
	ℓ_{4222}	(0.0000, 0.4751, 0.3242, 0.2007)	(5708.7, 2332.8, 79.0, 61)
	ℓ_{4223}	(0.0000, 0.2414, 0.4578, 0.3008)	(5834.1, 2590.1, 86.2, 57)
	ℓ_{4224}	(0.2959, 0.0000, 0.3939, 0.3103)	(5751.5, 2554.5, 82.2, 55)

continued . . .

. . . continued

Set	Solution	Weight	Objective Values
$\mathcal{S}_{423}$	ℓ_{4231}	(0.6264, 0.0268, 0.1530, 0.1937)	(5299.8, 2608.5, 61.1, 49)
	ℓ_{4232}	(0.4157, 0.0000, 0.2992, 0.2851)	(5413.2, 2594.0, 60.5, 47)
	ℓ_{4233}	(0.4668, 0.1303, 0.1361, 0.2668)	(5413.2, 2594.0, 60.5, 47)
	ℓ_{4234}	(0.5324, 0.2306, 0.0295, 0.2075)	(5411.4, 2490.8, 62.3, 49)
$\mathcal{S}_{424}$	ℓ_{4241}	(0.4604, 0.3498, 0.0000, 0.1898)	(5242.1, 2335.3, 46.1, 58)
	ℓ_{4242}	(0.3166, 0.4520, 0.0000, 0.2314)	(5576.1, 2229.4, 53.4, 57)
	ℓ_{4243}	(0.3596, 0.4052, 0.1188, 0.1164)	(5386.0, 2226.4, 56.8, 61)
	ℓ_{4244}	(0.4553, 0.3265, 0.0956, 0.1226)	(5334.6, 2332.9, 60.6, 57)
$\mathcal{S}_{431}$	ℓ_{4311}	(0.2505, 0.6250, 0.0000, 0.1245)	(5344.7, 2207.7, 40.3, 62)
	ℓ_{4312}	(0.0705, 0.7696, 0.0279, 0.1320)	(5807.2, 2132.3, 46.9, 65)
	ℓ_{4313}	(0.1880, 0.5435, 0.0000, 0.2685)	(5826.8, 2207.3, 42.7, 56)
	ℓ_{4314}	(0.4760, 0.3880, 0.0000, 0.1361)	(5344.7, 2207.7, 40.3, 62)
$\mathcal{S}_{432}$	ℓ_{4321}	(0.5730, 0.2697, 0.0000, 0.1572)	(5112.4, 2436.9, 46.6, 58)
	ℓ_{4322}	(0.4654, 0.3463, 0.0000, 0.1884)	(5411.4, 2490.8, 62.3, 49)
	ℓ_{4323}	(0.5245, 0.3288, 0.0000, 0.1467)	(5217.5, 2285.6, 53.1, 61)
	ℓ_{4324}	(0.4983, 0.3161, 0.0493, 0.1363)	(5217.5, 2285.6, 53.1, 61)
$\mathcal{S}_{433}$	ℓ_{4331}	(0.0000, 0.6880, 0.0000, 0.3120)	(6498.0, 2345.0, 48.6, 49)
	ℓ_{4332}	(0.0000, 0.6551, 0.1610, 0.1839)	(5900.2, 2265.3, 63.2, 55)
	ℓ_{4333}	(0.1539, 0.5677, 0.0000, 0.2784)	(5826.8, 2207.3, 42.7, 56)
	ℓ_{4334}	(0.0000, 0.8060, 0.0204, 0.1736)	(6098.7, 2188.9, 45.7, 57)
$\mathcal{S}_{434}$	ℓ_{4341}	(0.3577, 0.4097, 0.0957, 0.1369)	(5386.0, 2226.4, 56.8, 61)
	ℓ_{4342}	(0.0000, 0.6461, 0.2265, 0.1274)	(5741.6, 2261.5, 75.1, 63)
	ℓ_{4343}	(0.0668, 0.7622, 0.0428, 0.1283)	(5807.2, 2132.3, 46.9, 65)
	ℓ_{4344}	(0.1738, 0.5297, 0.1745, 0.1220)	(5741.6, 2261.5, 75.1, 63)
$\mathcal{S}_{441}$	ℓ_{4411}	(0.3750, 0.0000, 0.0000, 0.6250)	(5406.8, 2582.5, 58.3, 47)
	ℓ_{4412}	(0.3811, 0.3456, 0.0000, 0.2733)	(5411.4, 2490.8, 62.3, 49)
	ℓ_{4413}	(0.4108, 0.1384, 0.0330, 0.4177)	(5406.8, 2582.5, 58.3, 47)
	ℓ_{4414}	(0.5839, 0.0000, 0.0000, 0.4161)	(5406.8, 2582.5, 58.3, 47)
$\mathcal{S}_{442}$	ℓ_{4421}	(0.6181, 0.1389, 0.0000, 0.2431)	(5293.4, 2597.0, 58.9, 49)
	ℓ_{4422}	(0.7945, 0.0000, 0.0000, 0.2055)	(5173.8, 2589.4, 50.7, 52)
	ℓ_{4423}	(0.6308, 0.0000, 0.1654, 0.2038)	(5299.8, 2608.5, 61.1, 49)
	ℓ_{4424}	(0.5070, 0.2808, 0.0000, 0.2122)	(5411.4, 2490.8, 62.3, 49)

continued . . .

. . . continued

Set	Solution	Weight	Objective Values
$\mathcal{S}_{443}$	ℓ_{4431}	(0.1441, 0.4641, 0.0000, 0.3918)	(6503.1, 2493.5, 56.2, 43)
	ℓ_{4432}	(0.2188, 0.2659, 0.0000, 0.5153)	(6503.1, 2493.5, 56.2, 43)
	ℓ_{4433}	(0.2655, 0.1329, 0.0155, 0.5861)	(6493.5, 2499.7, 57.4, 43)
	ℓ_{4434}	(0.1597, 0.3168, 0.1136, 0.4099)	(6499.9, 2511.2, 59.6, 43)
$\mathcal{S}_{444}$	ℓ_{4441}	(0.4654, 0.0000, 0.2403, 0.2943)	(5413.2, 2594.0, 60.5, 47)
	ℓ_{4442}	(0.1960, 0.0000, 0.3507, 0.4534)	(5413.2, 2594.0, 60.5, 47)
	ℓ_{4443}	(0.4208, 0.1973, 0.0775, 0.3044)	(5406.8, 2582.5, 58.3, 47)
	ℓ_{4444}	(0.5103, 0.0063, 0.0493, 0.4341)	(5413.2, 2594.0, 60.5, 47)

B.2. List or Random Weight Vectors and Corresponding Solutions

Table B.2 lists the random weight vectors used in Section 4.3.3.2 along with corresponding normalized and actual objective function values for average medical proximity, average transport proximity, similarity and total sharpness count objectives, respectively.

Table B.2. List of random weight vectors and corresponding objective values.

Weight	Normalized Obj. Values	Actual Obj. Values
(0.1708, 0.2201, 0.5, 0.1402)	(1.23, 1.30, 1.0, 1.53)	(5952, 2659, 98.2, 66)
(0.1592, 0.7073, 0.0, 0.1210)	(1.16, 1.06, 0.4, 1.42)	(5652, 2174, 44.2, 61)
(0.0306, 0.0530, 0.4, 0.5278)	(1.34, 1.23, 0.6, 1.00)	(6500, 2511, 59.6, 43)
(0.8111, 0.0193, 0.0, 0.1550)	(1.07, 1.27, 0.5, 1.21)	(5174, 2589, 50.7, 52)
(0.3454, 0.1018, 0.2, 0.3170)	(1.12, 1.27, 0.6, 1.09)	(5413, 2594, 60.5, 47)
(0.0252, 0.4579, 0.1, 0.3782)	(1.34, 1.23, 0.6, 1.00)	(6500, 2511, 59.6, 43)
(0.1878, 0.2406, 0.5, 0.0746)	(1.22, 1.31, 1.0, 1.63)	(5927, 2669, 100.0, 70)
(0.5229, 0.2571, 0.0, 0.2134)	(1.08, 1.25, 0.6, 1.19)	(5229, 2562, 54.9, 51)
(0.1421, 0.4120, 0.3, 0.1710)	(1.18, 1.14, 0.8, 1.42)	(5709, 2333, 79.0, 61)
(0.3206, 0.2232, 0.4, 0.0695)	(1.22, 1.31, 1.0, 1.63)	(5927, 2669, 100.0, 70)
(0.0531, 0.4847, 0.2, 0.2259)	(1.21, 1.13, 0.7, 1.23)	(5869, 2307, 64.9, 53)
(0.6743, 0.0593, 0.0, 0.2333)	(1.07, 1.28, 0.6, 1.19)	(5204, 2606, 56.1, 51)
(0.3749, 0.0437, 0.5, 0.0999)	(1.22, 1.31, 1.0, 1.63)	(5927, 2669, 100.0, 70)
(0.1377, 0.4116, 0.0, 0.4283)	(1.34, 1.22, 0.6, 1.00)	(6503, 2493, 56.2, 43)

continued . . .

. . . continued

Weight	Normalized Obj. Values	Actual Obj. Values
(0.2621, 0.2555, 0.3, 0.1830)	(1.18, 1.19, 0.8, 1.28)	(5719, 2430, 77.2, 55)
(0.6794, 0.0495, 0.2, 0.0627)	(1.06, 1.21, 0.6, 1.42)	(5138, 2475, 62.6, 61)
(0.0596, 0.7254, 0.1, 0.1634)	(1.26, 1.07, 0.5, 1.33)	(6099, 2189, 45.7, 57)
(0.2039, 0.0379, 0.5, 0.2766)	(1.21, 1.31, 0.9, 1.40)	(5878, 2680, 93.0, 60)
(0.6212, 0.0239, 0.1, 0.2334)	(1.09, 1.28, 0.6, 1.14)	(5300, 2608, 61.1, 49)
(0.2584, 0.2517, 0.2, 0.3241)	(1.12, 1.22, 0.6, 1.14)	(5418, 2502, 64.5, 49)
(0.0976, 0.1120, 0.3, 0.5397)	(1.34, 1.23, 0.6, 1.00)	(6500, 2511, 59.6, 43)
(0.0456, 0.1301, 0.6, 0.2281)	(1.23, 1.30, 1.0, 1.49)	(5949, 2661, 96.6, 64)
(0.1993, 0.2927, 0.4, 0.0586)	(1.22, 1.31, 1.0, 1.63)	(5927, 2669, 100.0, 70)
(0.5775, 0.2898, 0.1, 0.0327)	(1.03, 1.12, 0.6, 2.07)	(5008, 2293, 55.0, 89)
(0.3350, 0.3425, 0.3, 0.0693)	(1.18, 1.14, 0.8, 1.42)	(5709, 2333, 79.0, 61)
(0.3754, 0.2386, 0.2, 0.2038)	(1.12, 1.22, 0.6, 1.14)	(5418, 2502, 64.5, 49)
(0.1670, 0.3063, 0.5, 0.0371)	(1.22, 1.31, 1.0, 1.63)	(5927, 2669, 100.0, 70)
(0.5005, 0.2063, 0.0, 0.2780)	(1.11, 1.26, 0.6, 1.09)	(5407, 2583, 58.3, 47)
(0.6602, 0.0471, 0.2, 0.0681)	(1.06, 1.23, 0.7, 1.42)	(5162, 2515, 64.6, 61)
(0.5996, 0.1673, 0.0, 0.1952)	(1.08, 1.26, 0.6, 1.19)	(5220, 2577, 56.5, 51)
(0.2671, 0.2678, 0.1, 0.3990)	(1.15, 1.22, 0.6, 1.09)	(5571, 2501, 56.8, 47)
(0.3879, 0.1842, 0.3, 0.1201)	(1.14, 1.21, 0.8, 1.33)	(5539, 2467, 77.1, 57)
(0.5700, 0.2477, 0.1, 0.0379)	(1.06, 1.16, 0.6, 1.49)	(5120, 2365, 58.8, 64)
(0.1670, 0.3448, 0.3, 0.2251)	(1.18, 1.19, 0.8, 1.28)	(5719, 2430, 77.2, 55)
(0.1826, 0.0326, 0.8, 0.0278)	(1.22, 1.31, 1.0, 1.63)	(5927, 2669, 100.0, 70)
(0.2103, 0.2117, 0.2, 0.3346)	(1.12, 1.22, 0.6, 1.14)	(5418, 2502, 64.5, 49)
(0.0527, 0.1355, 0.1, 0.7388)	(1.34, 1.23, 0.6, 1.00)	(6500, 2511, 59.6, 43)
(0.1913, 0.2165, 0.4, 0.1888)	(1.21, 1.31, 0.9, 1.40)	(5878, 2680, 93.0, 60)
(0.3390, 0.1056, 0.3, 0.2947)	(1.12, 1.22, 0.6, 1.14)	(5418, 2502, 64.5, 49)
(0.2465, 0.5758, 0.2, 0.0207)	(1.10, 1.08, 0.6, 1.88)	(5348, 2198, 62.1, 81)
(0.4156, 0.3154, 0.1, 0.1539)	(1.11, 1.14, 0.6, 1.33)	(5368, 2329, 62.7, 57)
(0.0013, 0.0244, 0.4, 0.5544)	(1.34, 1.23, 0.6, 1.00)	(6500, 2511, 59.6, 43)
(0.1778, 0.4906, 0.0, 0.3044)	(1.17, 1.18, 0.6, 1.14)	(5695, 2403, 57.5, 49)
(0.1238, 0.3534, 0.0, 0.4939)	(1.34, 1.22, 0.6, 1.00)	(6503, 2493, 56.2, 43)
(0.4191, 0.1063, 0.2, 0.2618)	(1.12, 1.22, 0.6, 1.14)	(5418, 2502, 64.5, 49)
(0.6120, 0.0747, 0.3, 0.0543)	(1.07, 1.23, 0.7, 1.65)	(5202, 2503, 71.9, 71)
(0.1423, 0.2782, 0.2, 0.3646)	(1.34, 1.23, 0.6, 1.00)	(6500, 2511, 59.6, 43)
(0.0518, 0.3209, 0.3, 0.3205)	(1.34, 1.23, 0.6, 1.00)	(6500, 2511, 59.6, 43)
(0.0193, 0.1491, 0.6, 0.1916)	(1.23, 1.30, 1.0, 1.53)	(5952, 2659, 98.2, 66)

Vita

Mehmet Başdere

Education

Northwestern University, PhD candidate in Industrial Engineering and Management Sciences, Expected grad. June 2018.

- PhD Thesis: Models and Approaches to Multiobjective Arc Tour Problems with an Application to Marathon Course Design.
- Advisors: Prof. Karen Smilowitz & Prof. Sanjay Mehrotra.

Bogazici University, MS in Industrial Engineering, Feb 2010 - June 2012.

- MS Thesis: Exact and Heuristic Approaches to Operational Aircraft Maintenance Routing Problem.
- Advisor: Prof. Ümit Bilge.

Bogazici University, BS in Industrial Engineering, Feb 2006 - Feb 2010.

- Graduation Project Topic: Bilevel Sensor Network Problem.

Publications

- Başdere, M., Mehrotra, S., and Smilowitz, K. (2018). An interactive method for multiobjective optimization using weight diversification. *Submitted.*

- Başdere, M., Smilowitz, K., and Mehrotra, S. (2018). A study of the lock-free arc tour problem and path-based reformulations. *In final preparation.*
- Başdere, M., Caniglia, G., Collar, C., Rozolis, C., Chiampas, G., Nishi, M., and Smilowitz, K. (2018). SAFE: A comprehensive data visualization system. *Submitted.*
- Ansari, S., Başdere, M., Li, X., Ouyang, Y., and Smilowitz, K. (2017). Advancements in continuum approximation models for logistics and transportation systems: 1996 - 2016. *Transportation Research Part B: Methodological. In press.*
- Ross, C., Başdere, M., Chan, J.L., Mehrotra, S., Smilowitz, K., and Chiampas, G. (2015). Data value in patient tracking systems at racing events. *Medicine and Science in Sports and Exercise*, 47(10), 2014-2023.
- Başdere, M., and Bilge, Ü. (2014). Operational aircraft maintenance routing problem with remaining time consideration. *European Journal of Operational Research*, 235(1), 315-328.
- Başdere, M., Ross, C., Chan, J. L., Mehrotra, S., Smilowitz, K., and Chiampas, G. (2014). Acute incident rapid response at a mass-gathering event through comprehensive planning systems: a case report from the 2013 Shamrock Shuffle. *Prehospital and Disaster Medicine*, 29(03), 320-325.
- Başdere, M., Afşar, S., Aras, N., and Altınel, K. (2013). A leader-follower game for the point coverage problem in wireless sensor networks. *European Journal of Industrial Engineering*, 7(5), 635 - 656.

Awards & Honor

- 2018 - **INFORMS Innovative Applications in Analytics Award**, awarded by INFORMS Analytics Society to recognize creative and unique applications. Project: *SAFE (Situational Awareness for Events): A Data Visualization System.*
- 2017 - **Dissertation Year Fellowship**, awarded by the Transportation Center at Northwestern University to fund outstanding dissertation projects.
- 2015 - **Nemhauser Best Student Paper Award**, awarded to the best graduate student paper at Northwestern University IEMS Department. Paper: *The Lock-Free Arc Tour Problem with an Application to Marathon Course Design*, co-authored by Karen Smilowitz and Sanjay Mehrotra.
- 2013 - **Arthur P. Hurter Award**, awarded to the top first-year graduate students at Northwestern University IEMS Department.
- 2012 - **Northwestern University Scholarship** for 2012-2013 academic year to pursue doctoral studies.
- 2010 - **Received TUBITAK Scholarship**, awarded to outstanding graduate students in Turkey.
- 2005 - **Prime Ministry Scholarship**, given to the most successful 100 students in University Entrance Exam of Turkey.
- 2005 - Ranked 31st among 1.7 million students in the nationwide University Entrance Exam 2005, Turkey.