1.5em

AN ABSTRACT OF THE DISSERTATION OF

Pouya Barahimi for the degree of Doctor of Philosophy in Industrial Engineering presented on March 19, 2019.

Title: Reliable Integrated Planning: Applications in Hub Network Design and Airline Ground Crew Task Assignment

Abstract approved: ______________________________

Hector A. Vergara

In general, making optimal decisions is a never ending challenge that decision makers face. A comprehensive model that integrates decisions at all three levels of decision making (i.e., strategic, tactical, and operational) can help the decision maker to find solutions that best serve the organization's performance. However, as organizations expand their business, more decision problems that are larger in size and more complicated need to be considered. Thus, in practice such comprehensive models become very difficult to develop and solve. The challenge becomes even more complicated as uncertainty might have a significant effect on operations and has to be taken into account explicitly. To overcome these challenges, decision makers usually focus on individual decisions separately. They make strategic decisions first and pass the outcome to the lower levels as input. This approach fails to fully acknowledge the existing dependencies between decisions. Dealing with each decision individually can lead the decision maker to prefer an optimal solution that serves the individual planning problem but overlooks the effect of that decision all the way down to the lowest levels of operation.

In this research, we propose a Reliable Integrated Planning Framework (RIPF) as a modeling approach for highly complex systems that utilizes corrective constraints to capture interdependencies between decisions at different levels of decision making. We apply this modeling approach to a simplified airline operations framework which involves planning problems associated with hub network design at the strategic level, flight times and gate selection and ground crew determination at the tactical level, and ground crew task assignment at the operational level. To apply the RIPF in this context, we develop modeling and solution methods for the Reliable p-Hub Network Design Problem under Multiple Disruptions

(RpHND-MD) for the strategic level problem, and the Task Assignment Problem with Flexible Execution Times and Sequence Dependent Travel (TAP-FET-SDT) for the operational level problem. Extensive computational testing of the RpHND-MD and the TAP-FET-SDT problems is completed to evaluate the performance of the modeling and solution methods developed in each case and to obtain insights about these problems as different parameters are modified. Then, we formally introduce RIPF and provide details about its application in a case study of airline operations planning. We implement a simulation of failure scenarios (i.e., hub airport disruptions) affecting airline operations to compare the performance of decisions selected using a classical planning approach against decisions made using the proposed RIPF. Based on the outcomes of the comparison, it is shown that the addition of corrective constraints to capture top-to-bottom and bottom-to-top dependencies between different planning levels helps to improve the overall performance of the system in terms of fewer resource shortages (i.e., gate shortages and ground crew shortages) at active airport hubs when other hubs fail in the network. At the same time, these results suggest that the RIPF provides decision makers a systematic approach to incorporate valuable information at the tactical and operational levels when making decisions in uncertain environments even when still making use of deterministic models at these levels. At the end, we discuss some of the challenges involved in applying the RIPF and conclude by suggesting directions for future research.

Reliable Integrated Planning: Applications in Hub Network Design and Airline Ground Crew Task Assignment

by

Pouya Barahimi

A DISSERTATION

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Presented March 19, 2019
Commencement June 2019

Doctor of Philosophy dissertation of Pouya Barahimi presented on March 19, 2019.

APPROVED:

Major Professor, representing Industrial Engineering

Head of the School of Mechanical, Industrial, and Manufacturing Engineering

---

Dean of the Graduate School

I understand that my dissertation will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my dissertation to any reader upon request.

Pouya Barahimi, Author

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to my parents, brother, sister, and wife. Undoubtedly, this dissertation would have not been completed without their unsparing love and support. I would like to pay special thankfulness, warmth and appreciation to my major advisor, Professor Hector A. Vergara, who showed confidence in my work and never stopped challenging me, and helping me develop my ideas. A word of gratitude is due to my minor advisor, Professor Glencora Borradaile, and my committee members, Professor David Kim, Professor David Porter, and Professor Wade Marcum, who opened my eyes to new perspectives on science and research. In addition, I wish to present my special thanks to Dr. Seyed Alireza Mohseni whose ideas have transformed my approach to problem solving.

TABLE OF CONTENTS

Page

1 Introduction 1

1.1 Background . . . 1

1.2 Research Questions . . . 5

1.3 Research Approach . . . 5

1.4 Major Contributions . . . 7

1.5 Dissertation Organization . . . 8

2 Reliable p-Hub Network Design Under Multiple Disruptions 10

2.1 Introduction . . . 10

2.2 Literature Review . . . 12

2.3 Modeling and Solution Approach . . . 15

2.3.1 Problem Definition . . . 15

2.3.2 Connection Tree . . . 16

2.3.3 Mathematical Formulation . . . 19

2.3.4 Search Algorithm . . . 23

2.4 Numerical Study . . . 28

2.4.1 Model Validation . . . 29

2.4.2 Performance of the Proposed Search Algorithm . . . 31

2.4.3 Evaluation of Design Parameters . . . 32

2.5 Conclusions . . . 37

2.6 References . . . 38

3 Task Assignment with Flexible Execution Times and Sequence Dependent Travel 40

3.1 Introduction . . . 40

3.2 Literature Review . . . 42

3.3 Problem Description . . . 44

3.3.1 Mathematical Formulation . . . 45

3.4 Branch-and-Price Algorithm . . . 48

3.4.1 Restricted Master Problem . . . 48

3.4.2 Pricing Subproblem . . . 49

3.4.3 Branch-and-Bound Algorithm for Pricing Subproblem . . . 50

3.4.4 Great Deluge Algorithm for Pricing Subproblem . . . 52

3.5 Numerical Study . . . 53

3.5.1 Problem Instances . . . 54

3.5.2 Performance Evaluation . . . 55

3.5.3 Effects of Larger Time Windows and Fewer Flexible Tasks . . . 57

3.5.4 Worker Utilization . . . 60

3.6 Conclusions and Future Work . . . 63

3.7 References . . . 65

TABLE OF CONTENTS (Continued)

Page

3.8 Appendix 67

4 Reliable Integrated Planning Framework: An Application in Airline Operations 69

4.1 Introduction and Background 69

4.2 Reliable Integrated Planning Framework 73

4.3 Use Case of the Reliable Integrated Planning Framework in Airline Operations 75

4.3.1 Level 1 - p-Hub Network Design 77

4.3.2 Level 2 - Determining Flight Times and Gate Selection 80

4.3.3 Level 3 - Ground Crew Determination 81

4.3.4 Level 4 - Task Assignment for Ground Crew 82

4.3.5 Corrective Constraints for the Reliable Integrated Planning Framework 85

4.3.6 Simulation Setup and Performance Measurement 88

4.4 Numerical Study 89

4.4.1 Effect of Number of Hubs on Overall Performance 93

4.5 Conclusions and Future Work 96

4.6 References 98

4.7 Appendices 100

4.7.1 Appendix A: Notation for RpHND-MD 100

4.7.2 Appendix B: Notation for TAP-FET-SDT 102

5 Conclusions and Direction for Future Research 104

References 109

LIST OF FIGURES

Figure | Page

1.1 Reliable Integrated Planning Framework. 6

2.1 Example of primary and backup connections between nodes i and j 16

2.2 A connection tree for O-D pair $(1, 9)$ with $\{3, 5, 7, 8\}$ as hubs 17

2.3 A connection-tree for O-D pair $(1, 5)$ with $\{3, 5, 7, 8\}$ as hubs 18

2.4 Search algorithm for solving RpHND-MD 23

2.5 An instance of CAB dataset with 10 nodes and 3 hubs where R-MAHMP and RpHND-MD select different connections between Detroit and Houston for scenarios where Dallas-Fort Worth failed. Backup connections 1 and 2 are selected by R-MAHMP and RpHND-MD, respectively 31

2.6 Experiment results on utilization of inter-hub links 34

2.7 Experiment results on hubs dispersion 35

2.8 25 node, 5 hub instance with low failure rate ($q = 10\%$) 36

2.9 25 node, 5 hub instance with high failure rate ($q = 90\%$) 36

3.1 Steps taken to move to a new feasible neighbor. 54

3.2 Performance of proposed solution methods. 56

3.3 BP-BB and BP-GD performance comparison for additional instances. 57

3.4 Solution quality of BP-GD for the three alternative datasets as a function of task size with 40 workers. 59

3.5 Worker utilization variability for instances with 20 workers. 62

4.1 Reliable Integrated Planning Framework. 74

4.2 Classic planning framework in the context of airline operations. 75

4.3 Reliable Integrated Planning Framework in the context of airline operations. 77

4.4 A Reliable Integrated Planning Framework in the context of airline operations with three sets of corrective constraints. 86

4.5 Distribution of hub failures in terms of number of simultaneously failed hubs, $p = 5$ and $p = 7$. 94

4.6 Comparison of the distribution of the number of active hubs with gate shortages on days with failure 96

LIST OF TABLES

Table Page

2.1 Positioning of the current study in the RHND literature 15

2.2 Comparing the results obtained with R-MAHMP and RpHND-MD 29

2.3 Performance of the proposed search algorithm for RpHND-MD 32

3.1 Comparison of articles on the task assignment problem with movements. . . 44

3.2 Solution to TAP-FET-SDT with 200 tasks and 20 workers 61

3.3 Comparison of objective function values and run times in CPLEX for TAP-FET-SDT with and without workload balancing constraints and different workload imbalance allowances. 64

A.1 Objective function value and run time of three solution methods on instances of TAP-FET-SDT. 68

4.1 Sample flight times for a hub with 28 daily flights and 4 available gates. . . . 81

4.2 Problem parameters for numerical study. 89

4.3 Failure probabilities of airports in the network. 90

4.4 Distribution of number of simultaneous failed hubs, p=5. 91

4.5 Distribution of hub failures on days with at least one failure, p=5. 91

4.6 Status of the system on a simulated day under regular operations and when ORD fails. 92

4.7 Distribution of the number of active hubs with gate shortages on days with hub failures, p=5. 92

4.8 Distribution of crew shortages at active hubs under RPIF-2. 92

4.9 Distribution of crew shortages at active hubs under RPIF-3. 93

4.10 Distribution of hub failures on days with at least one failure, $p = 7$. 94

4.11 Distribution of the number of active hubs with gate shortages on days with failure, p=7. 95

LIST OF ALGORITHMS

Algorithm		Page
1	Heuristic to compute an upper bound for an optimal connection tree	25
2	Branch-and-bound procedure to obtain optimal connection tree for O-D pair ij under hub combination h	28
3	Great deluge algorithm for the solving the pricing subproblem.	53
4	Fixing an infeasible solution to the pricing subproblem.	53
5	Steps to develop a Reliable Integrated Planning Framework.	74
6	Heuristic to determine ground crew workers and their shifts.	83

Chapter 1: Introduction

1.1 Background

Decision makers face challenging problems at different planning levels. A common categorization of planning levels include strategic, tactical, and operational. The planning level of a decision is characterized by its scope, how frequently the decision has to be made, and how long its effects will last. Strategic decisions focus on long term goals and direction. For example, what markets a company will be aiming to increase its share during the next five years or where to locate major warehouses of a manufacturing company. Since strategic decisions are more long term-oriented, they involve the highest degree of uncertainty (Parnell et al., 2013). At the next level, tactical decisions help making the connection between the strategic direction of an organization and its day-to-day operations. Allocating resources and developing activities that will lead to attainment of measurable objectives fall within the tactical planning level. Finally, day-to-day operational decisions are short term and the decision context can change rapidly. Operational decisions can be a made as frequent as a few dozen times during a single day. Daily production scheduling and routing of mail trucks for delivery are operational decisions in nature. The main objective of this study is to explore a decision making framework in which reliable strategic, tactical, and operational decisions are integrated by considering the upstream and downstream interactions between the decision levels. The framework is examined in the context of the airline industry where a high degree of uncertainty exists and decisions are highly correlated. We utilize operations research techniques as decision support tools to make decisions in uncertain environments.

Most practitioners and researchers follow an analytical approach to address decision problems that affect different planning levels. One of the most common approaches is to break down (i.e., decompose) the problems into smaller and easier to solve pieces or subproblems. In other words, decisions at different planning levels are considered as individual problems or even further broken down into smaller subproblems. The outcomes of decisions made at the top planning level stream down to the lower planning levels and become the setting in which the tactical and operational decisions are considered. This helps researchers to model and solve each problem with a relatively accurate representation of the real world. However, treating each problem individually will not necessarily yield the best overall performance as the interactions between different planning levels tend to get largely overlooked, especially in cases where meaningful feedback can be extracted from lower planning levels to improve the decisions made at higher planning levels. The suboptimality of the decisions made under such approach is a major drawback.

Alternatively, as Parnell et al. (2013) state, "A good analysis must balance concerns across all three and must consider the dependencies across levels". Along this line, a group of operations research (OR) professionals in industry and academia aim at integrating multiple levels of decisions into a single comprehensive model to fully capture the interactions between different planning levels. Good examples of such efforts can mostly be found in the integrated supply chain and logistics literature. A review of different stages of supply chain design done by Erengüç et al. (1999) highlights the substantial benefits that can be achieved by integrating decisions at different levels of a supply chain.

From a different perspective, as businesses expand - both in terms of their geographical presence and the nature of their services/products - they become more susceptible to unanticipated events. For instance, the workload in a manufacturing company or the number and length of calls in a call center are almost never fixed and they are difficult to accurately forecast. The same characteristic applies to machine failures, adverse weather around an airport causing flight cancellations/delays, deviations between planned and actual available workforce, prices of raw material and/or final products, etc. (Van den Bergh et al., 2013; Janak et al., 2007). Uncertainty is an intrinsic component of almost any decision making process in business and technological environments. However, regardless of decision planning levels, most studies use a deterministic approach to solve problems. In dealing with uncertainty, it is more economical to take proactive measures rather than reactive and hasty corrections. To explicitly incorporate uncertainty as a structural part of the decision making process, one must first acknowledge the uncertainty, understand it, structure it, and finally make it part of the analysis (Kouvelis and Yu, 2013). Luce and Raiffa (2012) categorize decision environments into (i) certainty, (ii) risk, and (iii) uncertainty.

In certainty situations, the decision maker either ignores the chances of having any difference between the expected and the actual values of parameters or relies on historical data to forecast the inputs. In this context, only one set of input parameters/data is used and the best decision that fits the instance is made. In other words, the decision maker takes an optimistic approach by tuning the system that matches the most likely scenario. As mentioned before, such deterministic view of problems is relatively common and practical despite seeming naive at first glance. In risk situations, probability links inputs to outcomes, and enough data is available to estimate a probability distribution for the problem inputs. Stochastic Optimization (SO) techniques are applied in risky environments to make a decision that yields the best *expected* outcome in the long run. Finally, in uncertain environments, sufficient information on inputs is not readily available. Thus, the decision maker seeks to hedge against the worst case scenario. This approach is referred to as Robust Optimization (RO) in the OR literature. Note that insufficient historical data is not the only reason for using RO. Regardless of data availability, risk averse decision makers are more inclined towards using RO techniques as opposed to SO techniques. Also, in environments where the decision is not reversible/repeatable, RO happens to be more interesting to the decision maker (Kouvelis and Yu, 2013). However, regardless of the decision maker's preference and the availability of data, it is most reasonable in many cases to view the outcome of decisions in the long run. Therefore, reducing the expected cost or maximizing the expected gain through stochastic methods has become a very popular approach in practice. Accordingly, in this study, we limit ourselves to environments with known probability distributions for uncertainty sources and where the goal is to optimize the expected value of objective(s) in the long run (i.e., we follow a SO approach).

SO has proven to be effective and has attracted the attention of many researchers and practitioners. However, there are criticisms to these approaches as well. First, stochastic approaches require probability distributions of the inputs. In practice, estimating a probability distribution is not a trivial task for many decision makers (Kouvelis and Yu, 2013). In addition, the volume of historical data in newly established businesses is too small to fit a reliable probability distribution. In some cases, such as natural disasters, estimating an accurate probability distribution is very difficult due to highly complicated mechanisms underlying the events. In addition, input factors can be correlated which not only makes the probability distribution estimation difficult, but also increases the modeling complexity of the problem. The other major challenge in dealing with uncertain systems is that even if the decision makers have all the desired information at their disposal, the optimization model that captures all possible scenarios becomes extremely large in practice. Finding the best answer to such problems can become very time-consuming or even impossible, leaving the decision maker with no choice but to make a suboptimal decision. This undermines the whole premise of SO. This further hinders developing and solving integrated models by adding more complexity to an already difficult model.

To summarize, for better decision making support, it is more desirable to completely acknowledge the interactions between decisions at different planning levels and to incorporate uncertainty at each planning level. One can achieve the former by developing a model that comprises all of the decisions together. However, such model is not only very difficult to develop but it also becomes intractable even on the smallest practical environments. Taking a hierarchical approach while maintaining bottom-to-top as well as top-to-bottom feedback can resolve this issue. Another aspect is to proactively deal with uncertainty by explicitly incorporating it in the decision making process. It should be noted that at each level of decision making, sources of uncertainty are either *internal* or *external* with respect to the decision making scope. For example, when making operational decisions at an airport, the level of uncertainty associated with the ability to carry out operations as planned can be affected by how the airline has setup its network and flight schedules. In other words, the airline's network reliability will directly affect the level of unanticipated shifts from operational plans at each airport. This can be considered as *internal* uncertainty. On the other hand, events such as equipment breakdowns or severe weather can be considered as *external* sources of uncertainty, the effect of which is out of the scope of decisions made at other levels. As mentioned above, the lack of knowledge about the likelihood of unanticipated events undermines making proper decisions. However, an integrated view to the system can mitigate the challenge. Incorporating uncertainty at higher levels and tracing its effects at lower levels will alleviate the lack of knowledge about input fluctuations for the lower level decisions. For instance, in the context of an airline, the likelihood of service interruptions at an airport caused by congested gates can be altered by shifting the airline's network and flight schedules - the outcome of higher level decisions.

Examining the literature, we observed that there is no study focused on integrating decisions at different planning levels of airline operations. The airline industry has made significant investments in the application of OR techniques to improve its business processes. Globally, airlines were expected to generate over $800 billions in revenue in 2018 which accounts for a 250% growth since 2003 (Statista, 2018). One possible reason for the lack of integrated planning models in this industry could be that the interaction between decisions made at different planning levels of airline operations is not as explicit and straightforward as that of supply chain design and operations. For instance, including decisions on hub locations and how to connect origin destination pairs in a mathematical model alongside of decisions on the number of ground crew workers at each hub airport and assigning the workers to tasks on a daily basis can be quite the challenge. Another reason that justifies exploring the potential gains from integrating decisions at different planning levels could be that the airline industry is inherently characterized as highly unpredictable. Many sources of uncertainty such as demand fluctuations, competitors shift in operations and policies, distribution of operations across numerous parts of the world with different weather conditions makes the business subject to significant deviations from status quo. Although airlines have put a solid effort into collecting data, it should be noted that benefits can still be gained from tracking the sources of uncertainty and capturing the mechanisms through which their effects are rippled down from strategic to tactical and operational decisions.

Accordingly, the goal of this dissertation is to address the aforementioned challenges by proposing a framework to merge strategic, tactical, and operational decisions into a comprehensive model that accounts for both the downstream and the upstream interactions as well as uncertainty. The framework is examined in the context of the airline industry.

1.2 Research Questions

Reflecting on the shortcomings discussed in Section 1.1, the following research questions are to be addressed as part of this dissertation:

1. Using OR techniques as decision support tools, how to develop an integrated planning approach and make good decisions without isolating the decisions at different planning levels?

2. How tractability issues can be mitigated while solving problems in an integrated planning approach?

3. How to include uncertainty in such integrated planning approach?

4. How to quantitatively assess the effectiveness of the integrated planning approach?

5. What are the advantages and challenges associated with integrating decisions at different planning levels while considering uncertainty when applying the modeling approach to a practical case?

1.3 Research Approach

In this study, we address the challenges faced by decision makers dealing with planning problems at different levels of decision making by proposing a Reliable Integrated Planning Framework (RIPF). This framework considers a hierarchy with strategic decisions at the top level. As we go down the hierarchy, the decisions become tactical and then operational (Figure 1.1).

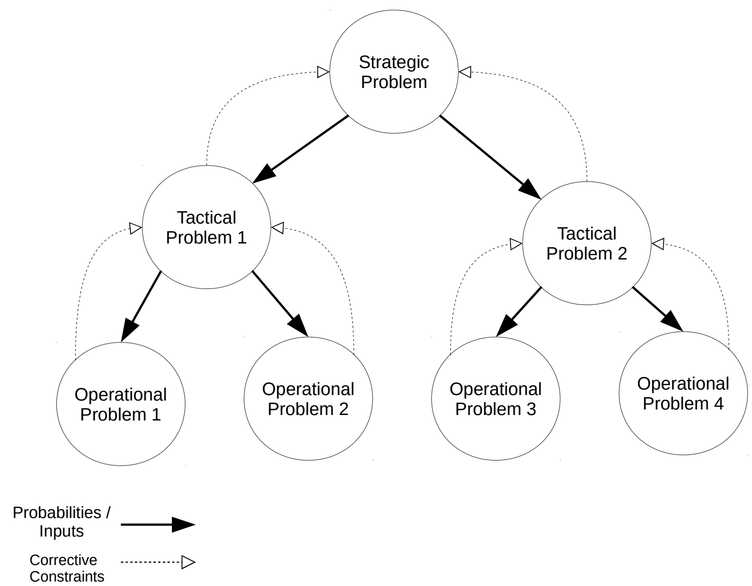


Figure 1.1: Reliable Integrated Planning Framework.

The solutions to the higher level problems form part of the input needed to model the problems at the lower levels. A critical part of the information that flows down from the top levels is how the higher level decision affects the uncertainty at lower levels. For example, consider a manufacturing company that has to decide which of its production facilities will be responding to unpredictable jumps in demand. The selection of those facilities and the rate by which they supply the additional demand will become an input to the design of those facilities such that the total expected cost of production is minimized. In other words, the major source of uncertainty for each problem lies within the decisions made at the preceding higher level(s). A realistic estimation of probability distributions for random components at the strategic level can facilitate finding probabilities for tactical and operational random components. Furthermore, in case of inadequate outcomes at the lower levels, corrective constraints can be added to the higher level decisions to complete the interaction between components. The RIPF is expected to (i) resolve the tractability issue in completely integrated models while maintaining a comprehensive view, and (ii) mitigate the inconvenience of estimating probabilities at tactical and operational levels by getting direct feedback from decisions made at higher levels.

We consider a simple planning framework in the context of airline operations consisting of four sets of decisions (levels) to demonstrate an application of the RIPF and to assess its effectiveness. The four planning decisions, in order from strategic to operational, are (i) hub network design with probability of failure of hub airports (Level-1), (ii) determining flight times & gate selection at hub airports (Level-2), (iii) ground crew determination at hub airports (Level-3), and (iv) assignment of ground crew to tasks on the day of operation (Level-4). The same set of decisions are made once under a classical planning approach and once more under the RIPF. In the classical planning approach, the outcome of each decision is passed down to the lower level decisions as input. However, RIPF incorporates feedback from both lower and higher level decisions in the form of corrective constraints. Overall performance of the system is evaluated using a simulation of airline operations. Level-1 and Level-4 of the RIPF involve relatively complicated optimization problems (i.e., Reliable p-Hub Network Design with Multiple Disruptions and Task Assignment with Flexible Execution Times and Sequence Dependent Travel, respectively). Thus, we first present mathematical models and efficient solution approaches for these problems. The proposed modeling and solution methods will then be utilized to make decisions at these levels and evaluate the effectiveness of the RIPF in the simulation of airline operations.

1.4 Major Contributions

The major contributions expected from this research are the following:

- Development of a framework to integrate decisions made at different planning levels while maintaining the interactions between decision levels, incorporating uncertainty, and mitigating tractability issues.
- A quantitative evaluation of the proposed integrated framework through simulation. The evaluation involves the setup of a sample hierarchy of decisions. In this context, we make the following related contributions to the literature as prerequisites for the simulation:
    - Development of a model and solution method to a general strategic planning problem that has application in the airline industry. An exact and efficient solution method is proposed to solve the multi-allocation reliable p-hub network design problem under multiple hub disruptions.
    - On the operational side, we develop an efficient model and solution method for the deterministic version of the ground crew task assignment problem with multiple skill requirements, flexible execution time tasks, and sequence dependent travel times.

1.5 Dissertation Organization

This dissertation has been organized in a manuscript format. Chapter 2 and Chapter 3 present modeling and solution methods for a strategic planning problem and an operational planning problem, respectively. Both planning problems are common but not exclusive to the airline industry. Chapter 4 introduces the RIPF and evaluates its performance in the context of airline operations under uncertainty. Finally, Chapter 5 summarizes the findings and suggests directions for future research.

At the strategic level, the design of a reliable network is a crucial decision with long term effects for airlines. Hence in Chapter 2, we first focus on the reliable p-hub network design problem under multiple disruptions (RpHND-MD). It should be noted that hub networks are also used in other modes of public and freight transportation, in express package shipping and postal services, and in telecommunications. In Chapter 2, we propose a decomposition procedure which is able to incorporate additional practical elements that have not been considered in previous studies of the Reliable Hub Network Design (RHND) problem. The modeling process becomes more complicated by assuming non-identical probability distributions for node failures. Chapter 2 was submitted to *Networks and Spatial Economics* and is currently undergoing minor revisions based on the feedback from the editor and the reviewers.

The second problem, included in Chapter 3, considers the task assignment problem with flexible execution times and sequence dependent travel (TAP-FET-SDT) as it applies to the assignment of an airline ground crew to tasks requiring a specific set of qualifications (i.e., skills) during their shifts. The problem exists on the other end of the planning horizon spectrum in the airline industry. In other words, it is solved multiple times on the day of operation. In Chapter 3, we contribute to the literature of the task assignment problem with multiple skills requirements by incorporating (i) sequence dependent travel time between tasks, and (ii) flexible execution time tasks. Flexible execution time tasks have a fixed length but are allowed to be executed within a given time window. The flexibility in execution time imposes a great deal of complexity into the problem. We model and solve realistic instances of this problem in a deterministic context. Chapter 3 was submitted to *Computers and Operations Research* and it is being revised based on feedback from the journal.

In Chapter 4, we formally introduce the Reliable Integrated Planning Framework(RIPF) in detail. RIPF can be considered as an extension to the classical planning framework where decisions made at the highest level (i.e., strategic) provide input to tactical and operational decisions. We evaluate the effectiveness of the proposed planning framework within the context of airline operations using a simulation approach. A comparison between the performance of a hypothetical airline under the classical planning approach versus the RIPF shows the effectiveness of the proposed framework. Chapter 4 will be submitted for review to *Transportation Research Part E: Logistics and Transportation Review.*

Chapter 2: Reliable p-Hub Network Design Under Multiple Disruptions[1]

Abstract

The design of optimal hub-and-spoke networks has been the objective of many research studies. More recently, several studies in this area have been concerned with incorporating failures of network entities (e.g., hubs and/or links) as a source of uncertainty in the formulation and solution of reliable hub networks. This study is focused on modeling and developing a solution approach for the multi-allocation reliable p-hub network design problem where more than one hub may be disrupted simultaneously. The objective is to minimize the expected cost of the network under all possible failure scenarios. A mathematical formulation is presented to select hubs and determine primary and backup connections for each origin-destination (O-D) node pair. An algorithm is suggested to generate primary and backup connections for an O-D pair for a given set of hubs. The hub selection model is then solved using a search algorithm. The computational results show that near optimal solutions for non-trivial problem size instances are obtained in a reasonable amount of time.

Keywords

Hub network; network design; reliable facility location; disruptions; hub-and-spoke; branch-and-bound

2.1 Introduction

Starting with O'Kelly (1986), numerous studies from different disciplines have addressed hub location and the design of hub networks with the objective of efficiently routing flows between many origins and destinations. The goal of the hub network design (HND) problem is to locate a number of hub nodes in a network to serve as transshipment, consolidation, or sorting points for flows between different origins and destinations using a reduced number of links. Non-hub nodes (also referred to as spokes in the literature) are assigned to hubs to complete the structure of the network, and the flows between origin-destination (O-D) nodes (which might or might not be hubs) are required to visit at least one hub to take advantage of economies of scale. Other basic assumptions have been made in the literature

[1] Under revision for publication in *Networks and Spatial Economics*

of the classic HND problem such as allowing flows to visit up to two hubs between origin and destination, having a complete network for inter-hub movements, a fixed discounted cost for inter-hub flows, infinite capacity at facilities and links, and either assigning a single hub or multiple hubs to each non-hub node (Campbell and O'Kelly, 2012).

Applications of hub networks are commonly found in telecommunications (Klincewicz, 1998) and the transportation and logistics industries such as air transportation, postal and express package carriers, less-than-truckload freight carriers, and rapid transit systems (Contreras, 2015). For extensive reviews of applications of hub-and-spoke networks, see Alumur and Kara (2008), Campbell and O'Kelly (2012), Contreras (2015), and Farahani et al. (2013).

Regardless of the application, HND problems are a challenging class of optimization problems. The integration of interrelated decisions at two levels of decision making (i.e., hub location and network design/link selection/routing) is one of the main difficulties associated with these problems. They are generally formulated as mixed integer programming models and solved using sophisticated solution algorithms, especially for large-scale instances (Contreras, 2015). However, there is a sense that most existing HND models do not really incorporate many real world elements as they are found in practice, and they rely heavily on the simplifying assumptions developed for the classic versions of the problem (Campbell and O'Kelly, 2012). Topics that are now starting to being studied include different hub network topologies, flow dependent discounted costs, capacitated models, models with uncertainty, and dynamic models among others (Contreras, 2015).

In particular, most previous studies in the HND literature assume that all entities (e.g., nodes and links) within the network operate constantly at full capacity. In other words, the chance of inefficiencies or failures of an entity is completely ignored. However, due to either natural disasters or intentional disruptions, the efficiency of a network can drastically decline imposing irreversible loss in terms of financial costs, customers' distrust, and opportunity costs. For instance, labor actions, changes of ownership, terrorist attacks (Snyder and Daskin, 2005), and adverse weather around airports (An et al., 2015) or seaports (Kim and Ryerson, 2013) are instances in which one or more parts of the hub-and-spoke network may fail. To address these instances, some recent research studies have focused on the reliability aspect of hub networks. The reliable hub network design (RHND) problem seeks to minimize the expected operating cost of a hub network while considering the failure probability of network elements (usually nodes).

In this paper, a mathematical model for the reliable p-hub network design problem under multiple disruptions (RpHND-MD) is proposed which is able to incorporate additional practical elements that have not been considered in existing models in the literature. The major contribution of this study is to propose a method to solve the RHND problem under multiple disruptions of hubs when hubs have different failure probabilities and spokes may be assigned to multiple hubs. Considering a non-identical failure probability distribution for the hub nodes complicates the modeling process as it hinders formulating a linear programming mathematical model.

The rest of this paper is organized as follows. Previous studies focusing on the RHND problem are reviewed in Section 2.2. Then, the mathematical formulation for RpHND-MD and the proposed solution procedure are presented in Section 2.3. Computational results for different instances are presented in Section 2.4. And finally, Section 2.5 shows relevant conclusions along with areas for future research.

2.2 Literature Review

The RHND problem seeks to minimize the expected operating cost of a hub network while considering the failure probability of network elements (Campbell and O'Kelly, 2012). There are multiple interpretations of the term "reliable network" in the literature. Here, we use the term reliability to refer to the "ability of the system to perform well even when parts of the system have failed" as defined by Snyder and Daskin (2005). As with the classic HND problem, applications of the RHND problem are mostly found in telecommunication networks (Kim and O'Kelly (2009), Davari et al. (2010), Kim (2012)) as well as in transportation and logistics networks (An et al. (2015), Kim and Ryerson (2013), Yıldız and Karaşan (2015)). Most previous research studies in this area have focused on developing models that explicitly incorporate potential failures in elements of the system (e.g., nodes and/or links) and their associated costs. Such models often come up with what we refer to as a backup setting for the hub network when a disruption occurs (An et al., 2015). Other approaches focus on maximizing the total network flow when considering the reliability of the O-D paths such as Kim and O'Kelly (2009) in which a method for calculating the reliability of paths in a hub network within the telecommunications context was introduced by taking into account the probability of successful communication between O-D pairs. According to An et al. (2015), the cost of ignoring potential disruptions and applying reactive strategies, like canceling and rerouting flights in the airline industry, to overcome those challenges can be dramatic. Thus, considering the probable failure of network entities and anticipating an alternative network design (i.e., backup setting) would help us to achieve a more reliable network where the operational costs arising from an unanticipated failure of an entity are minimized.

Research studies that deal with determining backup settings in hub networks can be divided into two groups. The first group includes studies that assume availability of some prior knowledge about the failing hubs. For instance, it is assumed that the most critical hubs in the network are subject to intentional disruption by a saboteur (Parvaresh et al. (2013), Parvaresh et al. (2014)) or that a set of disrupted hubs are directly given as an input for the hub network design problem as in Kim and Ryerson (2013). The second group corresponds to the studies where a failure probability is assumed to be available for hubs but no prior knowledge exists about which hubs are going to fail. An et al. (2015), Azizi et al. (2016), Tran et al. (2016), and Azizi (2017) are instances of the second group of studies. However regardless of prior knowledge or lack of knowledge about hub failures, with the exception of Tran et al. (2016), almost all related research studies solely determine what we call a first-level backup, i.e., a backup route for the case when only one hub fails at a time.

Complementing the work of Kim and O'Kelly (2009), Kim (2012) suggested a set of potential backup hubs along with regular hubs as a means of enhancing the reliability of a system. The linear programming model introduced therein was solved to optimality for 18-node networks. It should be noted that Kim (2012) defined backup hubs differently from regular hubs. That is, in case of disruption, both regular operating hubs and potential backup hubs can serve to ship flows between nodes.

In another study, An et al. (2015) formulated a nonlinear mathematical model in the air transportation context that incorporates the disruption of only one hub at a time. Applying a Lagrangian relaxation and branch-and-bound solution approach, the authors reported results for instances of up to 25 nodes with three to seven hubs to be located. Disruption of hub airports was assumed to occur one at a time and a backup hub for a disrupted hub was determined by the model from the set of operating hub airports. Similar to An et al. (2015), Azizi et al. (2016) tackled a similar problem taking only single failure scenarios into account. Their model sought to minimize the total expected cost of the network by finding one or multiple backups for each of the located hubs. Unlike An et al. (2015) who sought to provide a backup route for any route that utilizes a failing hub, the model presented by Azizi et al. (2016) aimed at finding a backup hub for each regular hub in the network and rerouting all the flows that use the regular hub through its corresponding backup. In other words, An et al. (2015) took an O-D based approach to finding backup routes, while Azizi et al. (2016) sought a model that would replace a failed hub with its backup regardless of the origin and destination of the flows that go through it. In Azizi et al. (2016), a genetic algorithm was developed to solve instances of the RHND problem with up to 20 nodes. More recently, Azizi (2017) modeled this problem as a mixed integer quadratic program and particle-swarm optimization-based metaheuristics were proposed for its solution. In Tran et al. (2016), a mixed integer nonlinear model is proposed to find the optimal location of a predetermined number of hubs. Unlike, An et al. (2015), their model considered more than one simultaneous failure at a time. The authors linearized their model using a specialized flow network called a probability lattice and solved it using tabu search. Another application of the RHND problem with a backup setting was presented by Kim and Ryerson (2013) in the context of maritime transportation. The authors considered disruptions of one or more seaports in a network. Their model was capable of identifying backup seaports from a set consisting of operating and ad-hoc seaports. In this case, although the number of disrupted ports is not limited to one, the failed hubs are given as an input to the model to mitigate the complexity that would have arisen from not having any prior information about which hubs failed and which ones are operating.

The valuable contributions presented above have provided developments in modeling and solution of reliable versions of classic hub location problems, but there is a lack of general network design principles generated from their results. Moreover, larger problems have been primarily tackled using metaheuristic approaches. Table 2.1 shows similarities and differences between our study and some of the relevant contributions presented above. To the best of our knowledge, the number of hubs to be located is always assumed to be known and the network entities are assumed to be uncapacitated. Having to determine the optimal number of hubs and/or limiting the capacity of network entities would lead to a significant increase in the complexity of the problem. In our study, we continue to assume that a fixed number of uncapacitated hubs need to be located in a network, but each spoke can be linked to more than one hub (i.e., multi-allocation of spokes to hubs). We aim to extend the work of An et al. (2015), Azizi et al. (2016) Azizi (2017), and Tran et al. (2016) by considering that multiple simultaneous disruptions of hubs can occur when no previous knowledge of the disruptions is available in hub networks with multiple allocation of hubs to spokes.

Depending on the failure probability of nodes in a network, multiple disruptions can be a likely occurrence in practice and may affect the configuration of an optimal hub network. Also in many applications, specifically in airline networks, assignment of non-hub nodes to the hubs is very common practice and worthwhile studying. Networks design problems, as strategic decisions, have a dramatic effect on the competitiveness of companies. Therefore,

Table 2.1: Positioning of the current study in the RHND literature

	Parvaresh et al. (2013, 2014)	Kim and Ryerson (2013)	An et al. (2015)	Tran et al. (2016)	Azizi et al. (2016); Azizi (2017)	This study
Model Type	p-median	p-median	p-median	p-median	p-median	p-median
Capacity	Uncap.	Uncap.	Uncap.	Uncap.	Uncap.	Uncap.
Allocation	Multi	Multi	Single/Multi	Single	Single	Multi
Disrupted Entity	Hubs	Hubs	Hubs	Hubs	Hubs	Hubs
Disruptions	Multiple (known)	Multiple (known)	Single (unknown)	Multiple (unknown)	Single (unknown)	Multiple (unknown)
Disruption Severity	Complete	Complete	Complete	Complete	Complete	Complete
Solution Method	Metaheuristic	Exact	Exact	Metaheuristic	Metaheuristic	Exact

even the slightest improvements by obtaining an optimal solution will result in improved profits. Given the limited amount of work addressing this challenge, the focus of this study is on the reliable design of hub networks under multiple disruptions when the only knowledge regarding hub failures is their independent expected probability. The underlying assumptions and objectives of the problem are discussed in detail in the following section.

2.3 Modeling and Solution Approach

2.3.1 Problem Definition

RHND models are designed not only to minimize the fixed and operational costs of a hub network, but also to mitigate the potential costs caused by unforeseen disruption of network entities. The goal of this study is to design a hub network which, as described by Snyder and Daskin (2005), "is still able to perform well even when failure occurs in node(s) of the network." We pursue this goal while considering the possibility of simultaneous independent disruption of hubs. Hence, a hub network is to be designed optimally by (i) locating p hubs, (ii) determining a primary connection between each O-D node pair in the network to be used when no disruptions occur, and (iii) backup connections for each O-D node pair for possible disruption scenarios. Figure 2.1 illustrates an instance of primary and backup connections selected for O-D pair (i, j) such that backup connection is associated with the failure scenarios where h_1 fails and both h_2 and h_3 remain active.

Basic assumptions of the RpHND-MD problem are similar to that of most hub network

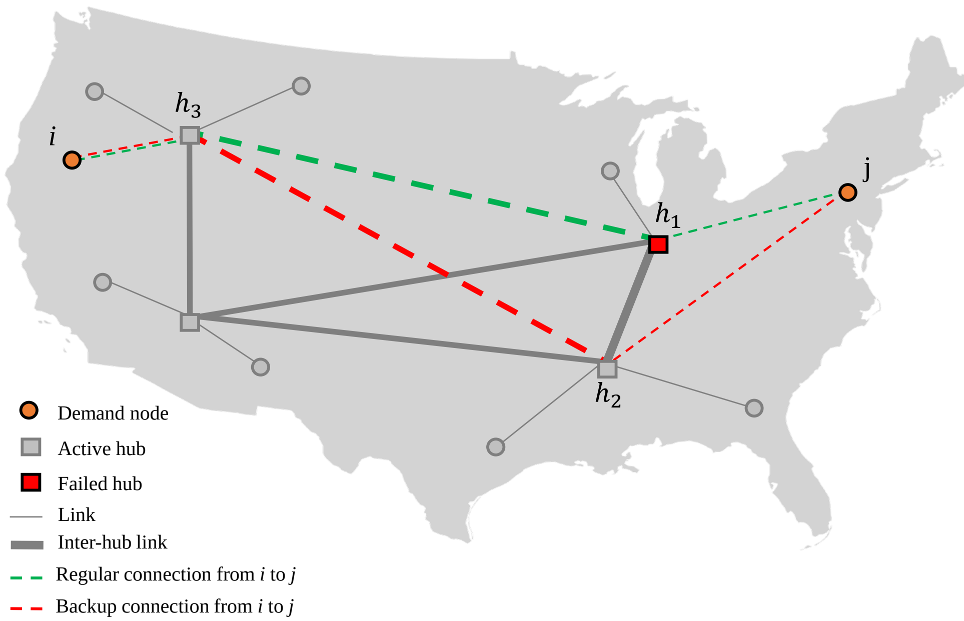


Figure 2.1: Example of primary and backup connections between nodes i and j

design studies in the literature. That is: (i) direct links between non-hub nodes are not allowed, (ii) spokes may be assigned to multiple hubs, (iii) the number of hubs to be located in the network is fixed and known, (iv) hubs form a complete graph, (v) the cost associated with establishing a hub is constant over all nodes, (vi) each node can be considered as a potential hub, (vii) the cost of shipping a unit of flow from node i to j is proportional to the distance between the two nodes and a discount factor is applied for inter-hub links reflecting economies of scale achieved by the consolidation of flows at hub nodes, (viii) hubs and links are uncapacitated; and, (ix) once a node fails, it fully loses its capacity to serve the network, i.e., partial disruptions are not applicable. In addition to this basic assumptions, in our study we allow multiple node disruptions following known independent non-identical binomial distributions. Note that the number of O-D pairs and disruption scenarios grow exponentially in the number of nodes and hubs to be located.

To formulate a mathematical model for RpHND-MD that reduces the number of variables needed, we introduce a binary tree representation of the primary and backup connections for a given O-D pair under a fixed hub combination as described below.

2.3.2 Connection Tree

Given a set of nodes, N, in a network, a subset of size p is to be selected as hubs. For each O-D pair a primary connection and a set of backup connections are to be determined such

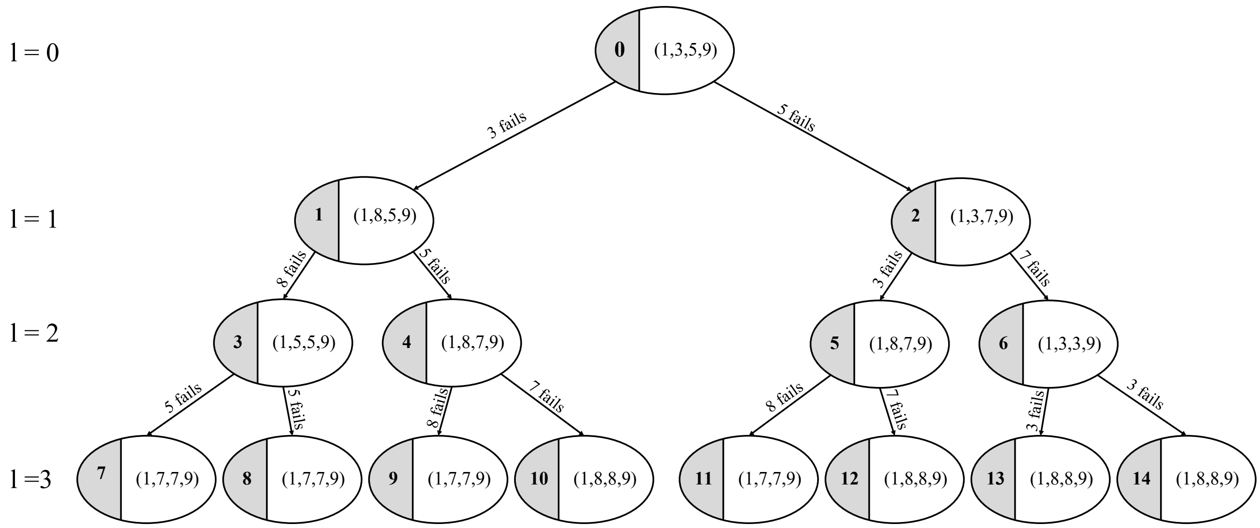


Figure 2.2: A connection tree for O-D pair $(1,9)$ with $\{3,5,7,8\}$ as hubs

that the total expected operating cost of the network is minimized under all possible failure scenarios. A connection (i,k,m,j), such that $i,k,m,j \in N$, connects origin i to destination j through hubs k and m, respectively. Here, $\mathcal{C}_{ij}$ is the set of all possible connections between nodes i and j – see Equation 2.1.

$$\mathcal{C}_{ij} = \{(i,k,m,j)\} \quad \forall k,m \in N \tag{2.1}$$

We use a binary tree structure from graph theory to represent a feasible selection of primary and backup connections for a given O-D pair. This facilitates modeling and solving the RpHND-MD problem. In addition, as we will discuss later, if one decides to limit the number of simultaneous failures, the model based on the binary tree structure will require a significant fewer number of decisions to be made, thus limiting problem size.

Define connection tree, T_{ij}, as a directed-out-tree rooted at vertex 0 such that each vertex, v, is mapped to a connection in $\mathcal{C}_{ij}$ and each edge $e_{u\to v}$ is mapped to a node in N. An edge $e_{u\to v}$ is mapped to hub $n \in N$ iff v is mapped to the backup connection for u when n fails. In other words, vertex 0 is mapped to the primary connection between ij and each of the other vertices is mapped to a connection that serves as a backup for their parent vertex. Figures 2.2 and 2.3 depict two examples of connection trees.

According to the assumptions in Section 2.3.1, there is a set of rules that determine whether a connection tree is feasible or not. The rules that make a connection tree feasible follow:

Rule 1: For any vertex u mapped to a connection with its first hub, k, different from its

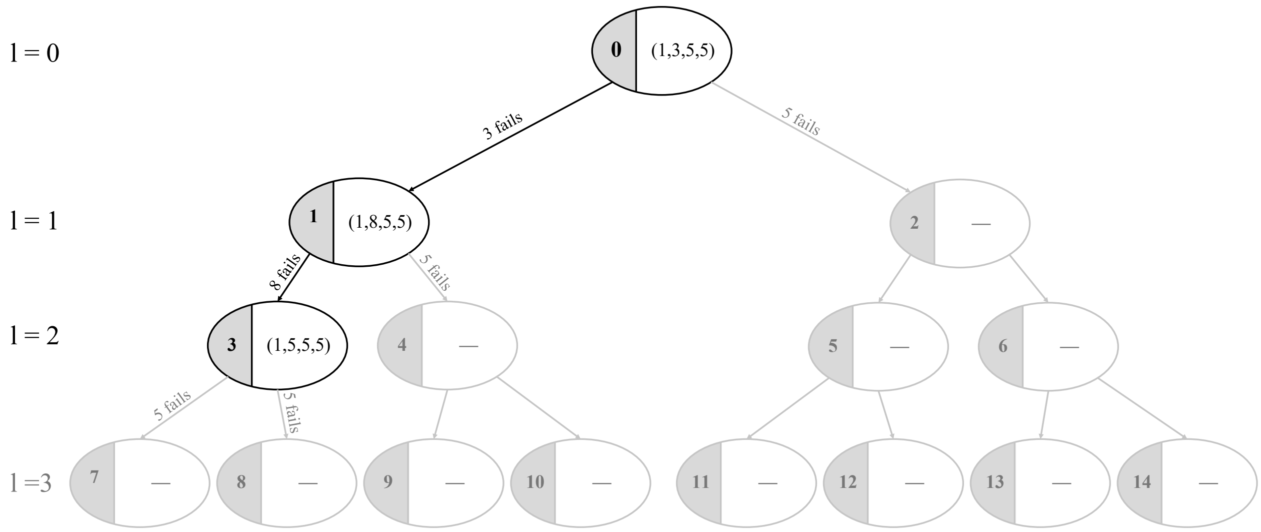


Figure 2.3: A connection-tree for O-D pair $(1, 5)$ with $\{3, 5, 7, 8\}$ as hubs

origin and destination, an edge $e_{u \to v} \to k$ exists such that v is mapped to a connection to be used when k fails. Otherwise, if k is either the origin or the destination, the flow associated with the O-D pair will be discarded as k fails. For example, consider vertex 1 as opposed to vertex 3 in Figure 2.3.

Rule 2: For any vertex u mapped to a connection with its second hub, m, different from its origin and destination, an edge $e_{u \to v} \to m$ exists such that v is mapped to a connection to be used when m fails. Otherwise, if m is either the origin or the destination, the flow associated with the O-D pair will be discarded as m fails. For example, consider vertex 0 in Figure 2.2 as opposed to vertex 0 in Figure 2.3.

Rule 3: A potential hub, k, can only be used in a vertex v iff k is not failed in the path from root to v. In other words, nodes that are pointed at by the edges from root to v cannot be used in v. For example consider vertex 10 in Figure 2.2. Hubs 3, 5, and 7 cannot be used to form a connection for this vertex as they are replaced in vertices 0, 1, and 4, respectively.

The deeper the tree gets, scenarios with more simultaneous failures are addressed. For instance, consider layer $l = 2$ in Figure 2.2. Vertices $\{0, \dots, 6\}$ address all failure scenarios with up to 2 failures: In a scenario where hubs 3 and 8 are failed, vertex 3 points to the backup connection, i.e., $(1, 5, 5, 9)$. Consider another scenario where hubs 5 and 8 are failed. In this case, vertex 2 yields the backup as the primary connection (vertex 0) cannot be used due to failure of hub 5. However, the hubs used in vertex 2 are still active and there is no need to go further down the tree. In this way, one can limit the number of simultaneous failures by setting a cap on the depth of the tree, L. Following the binary tree structure, we only need to make $O(2^L)$ decisions as opposed to dealing with $\sum_{k=0}^{L} \binom{p}{k}$ possible failure scenarios.

To exploit this structure of the problem, one has to compute the expected cost of a given connection tree. The expected cost of a connection tree is computed as the sum of expected cost of all its vertices. The expected cost of a vertex depends on the connection to which it is mapped. Equation (2.2) gives the expected cost of vertex $v \rightarrow (i, k, m, j)$ where q_k is the failure probability of node k, and P_v is the set of edges in the path from root to v.

$$\mathbb{E}(C_v) = \prod_{y \in P_v} q_y \ (1 - q_{km}) \ C_{ikmj} \tag{2.2}$$

where

$$q_{km} = \begin{cases} q_k & , \text{ if } k = m, \ v \rightarrow (i, k, m, j) \\ q_k + q_m - q_k q_m & , \text{ if } k \neq m, \ v \rightarrow (i, k, m, j) \end{cases} \tag{2.3}$$

As an example, the expected cost of vertex $4 \rightarrow (1, 8, 7, 9)$ in Figure 2.2 is calculated as:

$$\mathbb{E}(C_4) = q_3 \times q_5 \times (1 - q_8 - q_7 + q_7 q_8) \times C_{1,8,7,9}$$

2.3.3 Mathematical Formulation

Taking advantage of the suggested structure of a connection tree for a given O-D pair, T_{ij}, a non-linear integer programming model is formulated for RpHND-MD. The notation for the proposed mathematical formulation follows:

Sets:

N = set of locations in network,

V_{ij} = set of vertices in connection-tree for O-D pair ij,

V_{ij}^v = set of vertices on the path from root to vertex v of the connection-tree for O-D pair ij, without including vertex v; $V_{ij}^v \subset V_{ij}$.

Indices:

i, k, m, j point to the origin, first hub, second hub, and destination of a connection, respectively; $i, k, m, j \in N$,

u, u', v point to a vertices of a connection-tree. $u, u', v \in \{0, 1, \ldots, 2^{L+1} - 2\}$ where L is the maximum number of simultaneous failures.

Parameters:

p = number of hubs to be located,

W_{ij} = total flow between i and j,

c_{ij} = cost of sending a unit of flow directly from node i to node j,

α = discount factor for shipping flow though inter-hub links,

C_{ikmj} = cost of shipping a unit of flow through connection (i, k, m, j) regardless of failures:

$$C_{ikmj} = c_{ik} + (1 - \alpha)\, c_{km} + c_{mj} \tag{2.4}$$

q_k = failure probability of a hub located at node k,

q_{km} = failure probability of a connection that uses k and m as hubs (see Equation 2.3),

L = maximum number of simultaneous failures,

v' = v's parent vertex.

Decision Variables:

In a directed tree, each vertex has only one incoming edge. Therefore, we can refer to any edge, except the root vertex, by its tail-vertex. We define e_{ij}^v as the edge ending in vertex v in connection tree T_{ij}.

$$z_{ij}^{v} = \begin{cases} 1, & \text{if vertex } v \text{ is mapped to a connection, in connection-tree } T_{ij}, \\ 0, & \text{otherwise.} \end{cases}$$

$$x_{ikmj}^{v} = \begin{cases} 1, & \text{if vertex } v \text{ is mapped to connection } (i,k,m,j) \text{ , in connection-tree } T_{ij} \text{ ,} \\ 0, & \text{otherwise.} \end{cases}$$

$$e_{ijk}^{v} = \begin{cases} 1, & \text{if the edge with vertex } v \text{ as its tail is mapped to node } k, \text{ in connection-tree } T_{ij}, \\ 0, & \text{otherwise.} \end{cases}$$

$$y_{i} = \begin{cases} 1, & \text{if a hub is established in node } i \text{ of the network,} \\ 0, & \text{otherwise.} \end{cases}$$

Auxiliary Variables:

$$\kappa_{ij}^{v} = \begin{cases} 1, & \text{if vertex } v \text{ of connection tree } T_{ij} \text{ gets a connection with} \\ & \text{either } i \text{ or } j \text{ as its first hub,} \\ 0, & \text{otherwise.} \end{cases}$$

$$\mu_{ij}^{v} = \begin{cases} 1, & \text{if vertex } v \text{ of connection tree } T_{ij} \text{ gets a connection with} \\ & \text{either } i \text{ or } j \text{ as its second hub,} \\ 0, & \text{otherwise.} \end{cases}$$

The mathematical formulation for RpHND-MD $\mathcal{P}0$ follows:

$$\mathcal{P}0: \ Min \sum_{i} \sum_{j>i} \sum_{v} W_{ij} \left(\prod_{u \in V_{ij}^{v}} \sum_{k} q_k \ e_{ijk}^{u} \right) \cdot$$

$$\left(\sum_{k} \sum_{m} (1 - q_{km}) \ C_{ikmj} \ x_{ikmj}^{v} \right) \tag{2.5}$$

$$S.T. \ z_{ij}^{0} = 1 \qquad \forall i, j > i \in N \tag{2.6}$$

$$z_{ij}^{v} \leq \sum_{k} \sum_{m} x_{ikmj}^{v} \qquad \forall i, j > i \in N, v \in V_{ij} \tag{2.7}$$

$$z_{ij}^{v} = (1 - \kappa_{ij}^{v'}) \ z_{ij}^{v'} \qquad \forall i, j > i \in N, v_{odd} \in V_{ij} \tag{2.8}$$

$$z_{ij}^{v} = (1 - \mu_{ij}^{v'}) \ z_{ij}^{v'} \qquad \forall i, j > i \in N, v_{even} \in V_{ij} \tag{2.9}$$

$$\sum_{u \in V_{ij}^{v}} e_{ijk}^{u} \leq (1 - \frac{\sum_m x_{ikmj} + \sum_m x_{imkj}}{2}) \qquad \forall i, j > i, k \in N, v \in V_{ij} \tag{2.10}$$

$$e_{ijk}^{v} = \sum_{m} x_{ikmj}^{v'} \qquad \forall i, j > i, k \in N, v_{odd} \in V_{ij} \tag{2.11}$$

$$e_{ijk}^{v} = \sum_{k} x_{ikmj}^{v'} \qquad \forall i, j > i, m \in N, v_{even} \in V_{ij} \tag{2.12}$$

$$\sum_{m} x_{ikmj}^{v} \leq y_k \qquad \forall i, j > i, k \in N, v \in V_{ij} \tag{2.13}$$

$$\sum_{k} x_{ikmj}^{v} \leq y_m \qquad \forall i, j > i, m \in N, v \in V_{ij} \tag{2.14}$$

$$\sum_{i} y_i = p \tag{2.15}$$

$$\kappa_{ij}^{v} = \sum_{m} x_{iimj}^{v} + x_{ijjj}^{v} \qquad \forall i, j > i \in N, v \in V_{ij} \tag{2.16}$$

$$\mu_{ij}^{v} = \sum_{k} x_{ikjj}^{v} + x_{iiij}^{v} \qquad \forall i, j > i \in N, v \in V_{ij} \tag{2.17}$$

$$z_{ij}^{v}, e_{ij}^{v}, x_{ikmj}^{v}, y_i, \kappa_{ij}^{v}, \mu_{ij}^{v} \in \{1, 0\} \qquad \forall i, j > i, k, m \in N, v \in V_{ij} \tag{2.18}$$

Objective function (4.1) minimizes the total expected operating cost by summing up the expected cost of the connection trees' vertices over all O-D pairs similar to Equation (2.2). Note that we are limiting the number of failure scenarios to consider by establishing a cap on the maximum number of simultaneous failures. Therefore, Objective function (4.1) actually minimizes only the total cost of the connection tree vertices over all O-D pairs. Thus, the objective function value is less than or equal to the expected cost of the network under the considered scenarios. However, this does not affect the solution. Constraints (4.2) ensure that a primary connection is selected for each O-D pair. Constraints (4.3) make the link between x and z variables, meaning that if vertex v in connection tree T_{ij} is set to take a

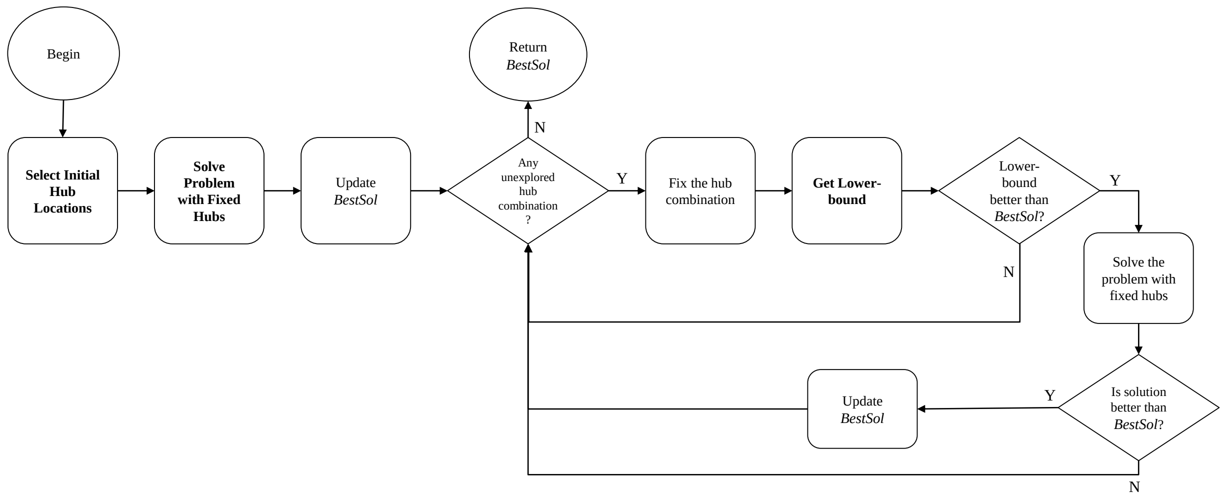


Figure 2.4: Search algorithm for solving RpHND-MD

connection, at least one connection has to be selected for it. Constraints (4.4), according to **Rule 1** (see Section (2.3.2)), dictate that vertex v has to be mapped to a connection iff (i) its parent vertex, v', is mapped to a connection, and (ii) v' does not have its first hub as origin or destination. Similarly, Constraints (4.5), according to **Rule 2**, force the selection of a backup connection for vertices that have an origin and a destination different from their second hub. Constraints (4.6), according to **Rule 3**, ensure that the hubs mapped to edges on the path from root to v are not used in v. Constraints (4.7) and (4.8) map the edges to their corresponding failed hub based on their head vertex. Constraints (4.9), (4.10), and (4.11) are common to hub location problems: Constraints (4.9) and (4.10) require a hub to be opened at a location that serves as a hub in any selected connection. Constraints (4.11) enforce that exactly p hubs are established in the network. Constraints (4.12) and p1:eq:16 assign the right value to the auxiliary variables, κ_{ij}^v and μ_{ij}^v, according to their definition. Finally, Constraints (4.14) are the variable type constraints.

2.3.4 Search Algorithm

Figure 2.4 shows a flowchart representation of a search algorithm developed for obtaining optimal solutions for RpHND-MD. The following subsections provide details on how to (i) obtain a good initial set of selected hubs (or hub combination), (ii) calculate tight upper and lower bounds; and most importantly, (iii) calculate the expected cost of the network under a fixed hub combination.

2.3.4.1 Initial Hub Locations

Instead of randomly selecting an initial set of p hubs, one can obtain a tighter upper-bound on the expected cost of the network by initiating the algorithm with a promising hub combination. The solution to the traditional hub network design (HND) problem can provide a solid starting point. However, as the HND problem is NP-Hard (Alumur and Kara, 2008), it would be more difficult to obtain solutions as larger problems are considered. To overcome this challenge, we apply the k-medoids clustering algorithm to come up with an initial hub combination. Kaufman and Rousseeuw (1990) provide an algorithm for partitioning a set of objects using a similarity function such that the objects in a partition have minimum dissimilarity from a representative for the partition. Applying this concept to a network of physical locations while considering Euclidean distances as the similarity function yields a fast yet reasonable method for selecting partition representatives, i.e., hubs. This is especially true where flow between O-D pairs have a relatively low variance.

2.3.4.2 Lower Bound and Upper Bound Under a Fixed Hub Combination

In general, the contribution of a potential connection to the expected cost of a connection tree diminishes as we go down the tree. This is due to the multiplication of hub failure probabilities on the path that starts at the root of the connection tree - first term in Equation (2.2). Hence, assigning connections with lower *absolute expected cost* AEC to the upper levels of a tree will lead to a reasonably low-cost connection tree. The AEC of a connection is given by the reliability of the connection multiplied by its cost. Equation (20) calculates the AEC of connection (i, k, m, j).

$$AEC_{(ijkm)} = \begin{cases} (1 - q_k) \times C_{ikmj}, & k = m, \\ (1 - q_k - q_m + q_k q_m) \times C_{ikmj}, & k \neq m \end{cases} \tag{20}$$

Algorithm 1 generates a near optimal connection tree for O-D pair ij under hub combination h with at most L simultaneous failures.

Algorithm 1 Heuristic to compute an upper bound for an optimal connection tree

```
procedure GetUpperBound(i, j, h, L)
    Tree ← [ ]                          ▷ initializing connection-tree as an empty array
    v ← 0                               ▷ current index in the tree
    C ← feasible connections from i to j under h
    sort C in ascending order based on absolute expected cost of its connections
    Tree[v] ← C[0]                      ▷ assign the connection with lowest exp. cost to the root
    v ← v + 1
    while v < 2^(L+1) − 1 do            ▷ while depth of the tree is not larger than L
        for n = 1 : |C| do
            if v′ requires backup & C[n] is a feasible backup then     ▷ v′ is parent of v
                Tree[v] ← C[n]
                v ← v + 1
                Break
    return Tree
```

The number of feasible connections that go from i to j under hub combination h is bounded by p^2 and can be sorted in $O(p^2 \log p)$. A connection tree with depth L has at most $2^{L+1} - 1$ vertices. Assuming L to be a constant, Algorithm 1 is polynomial in p. Hence, in a reasonable time, one can derive an upper bound on the minimum expected cost of a network under a given hub combination by running Algorithm 1 for all O-D pairs - $O(|N|^2 p^4 \log p)$. As we will see in the next subsections, the upper bound helps to expedite the search for an optimal connection tree.

Following the same reasoning, a connection tree with a single node mapped to the connection with the least AEC constitutes a lower bound on the optimal connection tree expected cost. Summation over all O-D pairs under a given hub combination h yields a lower bound on the network's expected cost. In practice, node failure probabilities are relatively low, say less than %5. Therefore, the contribution of backup connections are relatively small and the lower bound obtained in this manner provides a tight bound. In this way, we avoid computing the minimum expected operating cost of the network under a large portion of hub combinations.

2.3.4.3 Subproblem Formulation - Solving the Problem with Fixed Hub Locations

Let us restrict RpHND-MD by fixing hub locations and focusing on finding the least expected cost of shipping flows under the fixed hub combination. In this case, we can decompose the restricted problem into finding the optimal connection tree for each O-D pair ij given hub combination h. We refer to the restricted decomposed problem as the *subproblem*. The subproblem seeks the least cost connection tree for O-D pair ij under hub combination h. In the subproblem formulation $\mathcal{P}1$, the definition of parameters and decision variable are similar as before, except that all variables are now related to a specific O-D pair ij. Thus, the ij indices that distinguish a connection tree are removed. Note that potential connections are limited to the ones that go through at most two hubs in h.

$$\mathcal{P}1 : Min \sum_{v} \left(\prod_{u \in V^v} \sum_{k} q_k \; e_k^u \right) \times \left(\sum_{k} \sum_{m} (1 - q_{km}) \; C_{km} \; x_{ikmj}^v \right) \tag{4.1a}$$

$$S.T. \; z^0 = 1 \tag{4.2a}$$

$$z^v \leq \sum_{k} \sum_{m} x_{ikmj}^v \qquad \forall v \in V_{ij} \tag{4.3a}$$

$$z^v = (1 - \kappa^{v'}) \; z^{v'} \qquad \forall v_{odd} \in V \tag{4.4a}$$

$$z^v = (1 - \mu^{v'}) \; z^{v'} \qquad \forall v_{even} \in V \tag{4.5a}$$

$$\sum_{u \in V^v} e_k^u \leq (1 - \frac{\gamma_k^v + \lambda_k^v}{2}) \qquad \forall k \in h, v \in V \tag{4.6a}$$

$$e_k^v = \gamma_k^{v'} \qquad \forall k \in h, v_{odd} \in V \tag{4.7a}$$

$$e_k^v = \lambda_k^{v'} \qquad \forall k \in h, v_{even} \in V \tag{4.8a}$$

$$\kappa^v = \sum_{m} x_{iimj}^v + x_{ijjj}^v \qquad \forall v \in V \tag{4.12a}$$

$$\mu^v = \sum_{k} x_{ikjj}^v + x_{iiij}^v \qquad \forall v \in V \tag{4.13a}$$

$$z^v, e^v, x_{ikmj}^v, \kappa^v, \mu^v \in \{1, 0\} \qquad \forall k, m \in h, v \in V \tag{4.14a}$$

The objective function 4.1a minimizes the expected cost of the connection tree for O-D pair ij under hub combination h. Constraints (4.2a–4.6a) and (4.8a–4.14a) have the same implications as the ones in the original formulation, except for the fact that i and j are fixed, and $k, m \in h$ as opposed to $k, m \in N$ as in the original model.

2.3.4.4 Branch-and-bound Procedure to Solve Subproblem

As shown by Equation 2.2, the contribution of a potential connection in vertex v of a tree depends on the reliability of the connections assigned on the path from the root vertex to v. Such dependency constitutes a major obstacle to formulating a linear mathematical model or devising a divide and conquer algorithm to find the optimal connection tree in polynomial time. Hence, exhaustive enumeration seems to be the only option for finding the optimal connection tree. However, we avoid exhaustive enumeration by taking advantage of the binary structure of the tree and implementing a algorithm. To fathom the search tree, Algorithm 1 is used to obtain an upper bound on the cost of the optimal connection tree in polynomial time. The upper bound is then used to prune the search tree in Algorithm 2 which is implemented in a depth first search (DFS) manner. DFS not only decreases memory requirements but also helps tightening the bound as the search proceeds and prunning bigger portions of the search tree.

Algorithm 2 Branch-and-bound procedure to obtain optimal connection tree for O-D pair ij under hub combination h

```
procedure GetConnectionTree(i, j, h, L)
    UB ← GetUpperBound(i, j, h, L)
    OptimalTree ← null
    AllTrees ← [ ]                                    ▷ set of incomplete trees
    Connections ← feasible connections from i to j under h
    for each c ∈ Connections do
        NewTree ← [ ]
        NewTree[0] ← c
        AllTrees ← AllTrees + {NewTree}
    while AllTrees is not empty do      ▷ while depth of the tree is not larger than L
        CurrentTree ← AllTrees[last]                  ▷ last in first out
        AllTrees ← AllTrees − {CurrentTree}
        for all c ∈ Connections, such that c is a feasible backup for the left most leaf ∈ Tree do
            NewTree ← CurrentTree + c  ▷ connection c is added to the tree to make a new tree
            if Cost_NewTree < UB then
                if NewTree is complete then
                    OptimalTree ← NewTree
                    UB ← Cost_NewTree
                else
                    AllTrees ← AllTrees + {NewTree}
    return OptimalTree
```

2.4 Numerical Study

In this section, we first test our model and proposed solution algorithm by comparing results for benchmark instances with those obtained with the model by An et al. (2015) when setting the maximum number of failures to one. We then evaluate the efficiency of our approach by presenting solutions and performance measures for different instances of the RpHND-MD problem. Finally, we explore the effect of three different parameters on the configuration of reliable hub networks.

Our numerical experiments were completed on the broadly-used CAB dataset benchmark instances (O'Kelly, 1987). The search algorithm was implemented in Java and run on a HP Z640 Workstation (Intel® Xeon® CPU E5-2630 @ 2.40 GHz and 16.0 GB of RAM) in a 64-bit Windows 7 environment. As a baseline, the compact linear reformulation of the reliable multi-allocation p-hub median problem (R-MAHMP) presented by An et al. (2015) was implemented and solved using the Gurobi 6.0.5 library version for Java.

2.4.1 Model Validation

We first compare the output of the proposed search algorithm for RpHND-MD against the solution of the compact linear formulation of R-MAHMP presented by An et al. (2015). Table 2.2 presents the objective function value and selected hubs for both models on instances of CAB with 10 and 15 nodes. The difference (as a percentage) between the two objective function values is included as well. The number of hubs to be located are $p = \{3, 5, 7\}$, and the discount factor is considered at three levels, $\alpha = \{0.2, 0.4, 0.6\}$. Fixing the maximum level of failures, L, to 1 and setting the ρ coefficient in An et al. (2015)'s model to 1 makes the two models consistent in terms of objective function expression and consideration of backups in case of failures. Failure probabilities for the nodes are taken from the data provided by An et al. (2015).

Table 2.2: Comparing the results obtained with R-MAHMP and RpHND-MD

$\|N\|$	p	α	R-MAHMP Obj Val	Hubs	RpHND-MD Obj Val	Hubs	Difference
10	3	0.2	314,309,457.59	(3,5,6)	314,309,457.59	(3,5,6)	0.0000%
		0.4	295,158,855.11	(3,5,6)	295,157,278.10	(3,5,6)	**-0.0005%**
		0.6	274,075,015.21	(3,5,6)	274,065,465.51	(3,5,6)	**-0.0035%**
	5	0.2	281,620,864.50	(0,2,3,5,6)	281,613,584.25	(0,2,3,5,6)	**-0.0026%**
		0.4	243,289,466.16	(2,3,5,6,7)	243,287,889.15	(2,3,5,6,7)	**-0.0006%**
		0.6	200,437,069.68	(2,3,5,6,7)	200,398,400.80	(2,3,5,6,7)	**-0.0193%**
	7	0.2	262,694,220.01	(0,2,3,5,6,7,9)	262,694,220.01	(0,2,3,5,6,7,9)	0.0000%
		0.4	210,775,197.15	(0,1,2,3,6,7,8)	210,772,477.46	(0,1,2,3,6,7,8)	**-0.0013%**
		0.6	154,697,987.46	(0,1,2,3,6,7,8)	154,695,221.08	(0,1,2,3,6,7,8)	**-0.0018%**
15	3	0.2	1,207,892,613.94	(0,3,11)	1,207,892,613.94	(0,3,11)	0.0000%
		0.4	1,125,993,276.28	(3,6,11)	1,125,993,276.28	(3,6,11)	0.0000%
		0.6	1,036,207,334.09	(3,6,11)	1,036,207,334.09	(3,6,11)	0.0000%
	5	0.2	1,035,062,498.08	(3,5,6,11,13)	1,035,062,498.08	(3,5,6,11,13)	0.0000%
		0.4	910,896,100.77	(3,5,6,11,13)	910,894,523.77	(3,5,6,11,13)	**-0.0002%**
		0.6	770,302,982.36	(3,5,6,11,13)	770,293,432.66	(3,5,6,11,13)	**-0.0012%**
	7	0.2	972,435,244.52	(2,3,5,6,7,11,13)	972,435,244.52	(2,3,5,6,7,11,13)	0.0000%
		0.4	812,077,049.66	(2,3,5,6,7,11,13)	812,075,472.65	(2,3,5,6,7,11,13)	**-0.0002%**
		0.6	642,445,430.39	(2,3,5,6,7,11,13)	642,445,430.39	(2,3,5,6,7,11,13)	0.0000%

Both models seek to minimize the expected transportation cost. However, it is interesting to see that in some cases RpHND-MD performs slightly better - rows in Table 2.2 with a negative difference. This difference originates from the way the models handle backups. RpHND-MD considers the best available "connection" to connect origin and destination in case of a failure in any of the hubs used in the primary connection. However, An et al. (2015)'s model replaces the failed hub with the best available "hub". In other words, RpHND-MD finds backup connections for failure scenarios as opposed to finding an active hub to replace the failed hub. For connections that use two different hubs ,$\{(i,k,m,j)|k \neq m\}$, the latter approach replaces the failed hub with a backup enforcing the other hub to remain as part of the connection. However, this might not be the best available connection to link i and j. As an example, Figure 2.5 demonstrates part of the solution to an instance of CAB where $N = 10$, $p = 3$, $\alpha = 0.2$. Both models select Chicago, Cleveland, and Dallas-Fort Worth as hubs as well as (Detroit, Cleveland, Dallas-Fort Worth, Houston) as the primary connection between Detroit and Houston. However, R-MAHMP selects (Detroit, Cleveland, Cleveland, Houston) as the backup with a cost of 1,198.82 per unit flow as opposed to (Detroit, Chicago, Chicago, Houston) with a unit cost of 1,169.28, selected by RpHND-MD. Therefore, taking the latter approach in selecting backup seems to be a better option. However, depending on the application of the problem, one can switch to the former.

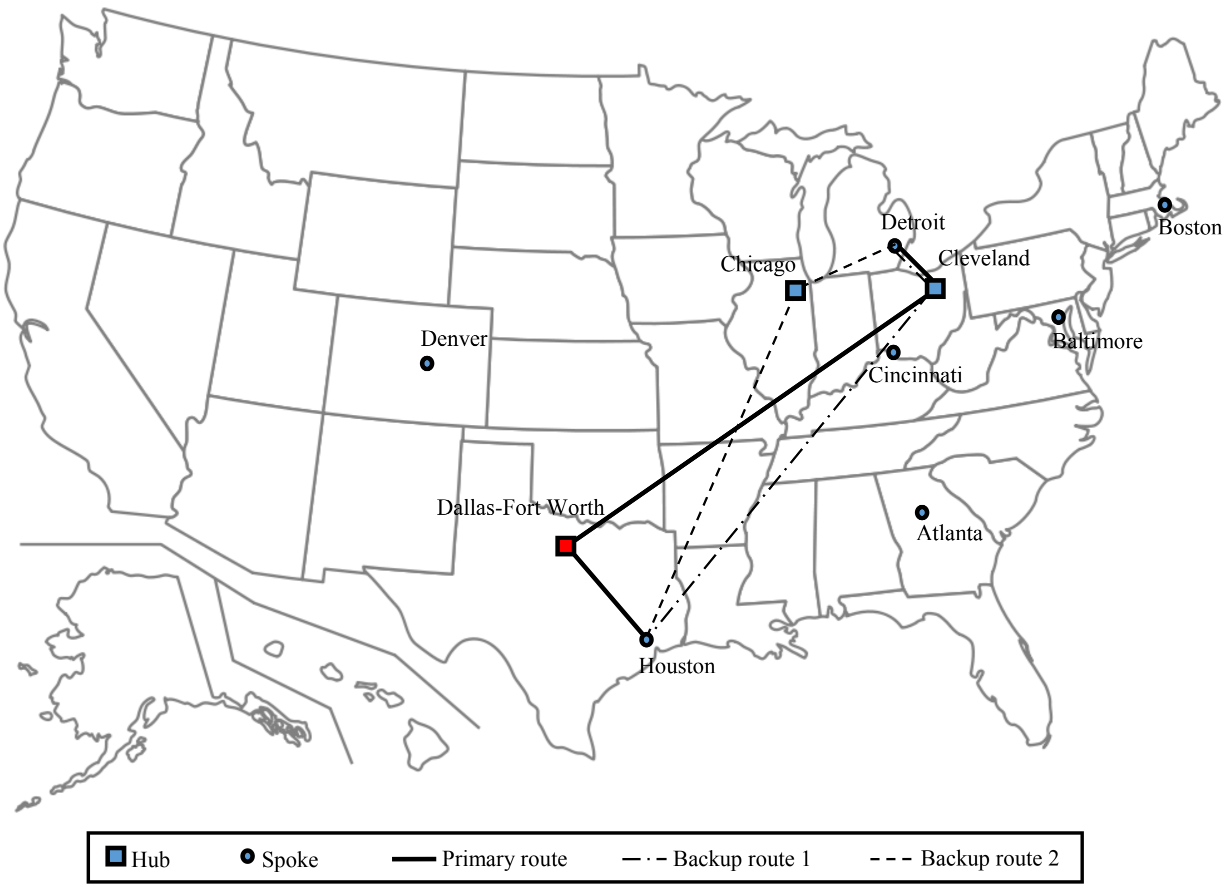


Figure 2.5: An instance of CAB dataset with 10 nodes and 3 hubs where R-MAHMP and RpHND-MD select different connections between Detroit and Houston for scenarios where Dallas-Fort Worth failed. Backup connections 1 and 2 are selected by R-MAHMP and RpHND-MD, respectively

2.4.2 Performance of the Proposed Search Algorithm

Table 2.3 shows the results of the proposed search algorithm applied to instances of CAB with $N = \{15, 20, 25\}$, $p = \{3, 5, 7\}$, $\alpha = \{0.2, 0.4, 0.6\}$, and $L = \{0, 1, 2\}$. Failure probabilities for nodes are come from An et al. (2015). Solution times do not exceed 3 minutes for the solved instances. However, the computational complexity of the solution algorithm forces the solution time to grow exponentially in the number of hub combinations, i.e., $\binom{|N|}{p}$, as well as the maximum number of simultaneous failures, L.

Table 2.3: Performance of the proposed search algorithm for RpHND-MD

L	$\|N\|$	p	α	Hubs	Obj,Val	Time (s)	L	$\|N\|$	p	alpha	Hubs	Obj,Val	Time (s)
0	15	3	0.2	(3,6,11)	1,170,580,961	1.90	1	20	5	0.6	(3,6,11,13,16)	1,730,587,573	1.91
			0.4	(3,6,11)	1,086,255,806	0.20			7	0.2	(3,6,7,11,13,16,19)	2,245,973,264	11.66
			0.6	(3,6,11)	997,519,327	0.17				0.4	(3,6,7,11,13,16,19)	1,883,474,629	11.15
		5	0.2	(3,5,6,11,13)	1,012,476,040	0.39				0.6	(0,3,5,6,11,13,16)	1,485,914,217	11.03
			0.4	(3,5,6,11,13)	885,626,067	0.33		25	3	0.2	(3,11,16)	4,404,452,134	0.45
			0.6	(3,5,6,11,13)	745,192,580	0.37				0.4	(3,11,16)	4,110,101,483	0.50
		7	0.2	(0,3,5,6,7,11,13)	953,780,000	0.66				0.6	(3,11,16)	3,738,438,392	0.51
			0.4	(2,3,5,6,7,11,13)	797,757,004	0.59			5	0.2	(3,6,11,13,16)	3,917,747,279	10.10
			0.6	(2,3,5,6,7,11,13)	628,521,527	0.54				0.4	(3,6,11,13,16)	3,474,737,011	9.85
	20	3	0.2	(3,16,18)	2,639,658,479	0.27				0.6	(3,6,11,13,16)	2,936,915,738	9.48
			0.4	(3,16,18)	2,464,382,259	0.27			7	0.2	(3,6,11,13,16,19,21)	3,666,368,132	126.71
			0.6	(3,11,16)	2,241,937,790	0.27				0.4	(3,6,11,13,16,19,21)	3,155,671,991	121.00
		5	0.2	(3,6,13,16,18)	2,317,510,586	1.90				0.6	(3,6,11,13,16,19,21)	2,555,413,856	117.54
			0.4	(3,6,11,13,16)	2,016,461,236	1.82	2	15	3	0.2	(0,3,11)	1,209,474,914	0.21
			0.6	(3,6,11,13,16)	1,666,539,508	1.73				0.4	(3,6,11)	1,126,936,846	0.18
		7	0.2	(0,3,5,6,13,16,18)	2,191,837,0.56	1.07				0.6	(3,6,11)	1,037,174,458	0.19
			0.4	(0,3,5,6,11,13,16)	1,835,816,0.85	1.11			5	0.2	(3,5,6,11,13)	1,036,040,736	0.41
			0.6	(0,3,5,6,11,13,16)	1,432,790,0.49	1.03				0.4	(3,5,6,11,13)	911,978,726	0.40
	25	3	0.2	(11,20,24)	4,198,748,0.23	0.46				0.6	(3,5,6,11,13)	771,503,611	0.42
			0.4	(3,11,16)	3,905,078,867	0.44			7	0.2	(2,3,5,6,7,11,13)	972,882,659	0.92
			0.6	(3,11,16)	3,523,623,286	0.43				0.4	(2,3,5,6,7,11,13)	812,652,413	1.02
		5	0.2	(3,6,11,16,23)	3,762,708,099	9.03				0.6	(2,3,6,7,8,11,13)	642,876,751	1.32
			0.4	(3,6,11,16,23)	3,314,273,200	9.23		20	3	0.2	(3,6,16)	2,749,934,177	0.28
			0.6	(3,6,11,13,16)	2,778,978,418	8.38				0.4	(3,11,16)	2,582,038,273	0.25
		7	0.2	(3,6,16,18,21,23,24)	3,571,563,807	119.28				0.6	(3,11,16)	2,354,549,681	0.28
			0.4	(3,6,7,11,16,23,24)	3,055,798,125	112.89			5	0.2	(3,6,11,13,16)	2,387,087,062	2.30
			0.6	(3,5,6,11,13,16,21)	2,466,825,676	107.48				0.4	(3,6,11,13,16)	2,082,169,269	2.45
1	15	3	0.2	(0,3,11)	1,207,892,614	0.22				0.6	(3,6,11,13,16)	1,733,671,845	2.30
			0.4	(3,6,11)	1,125,993,276	0.16			7	0.2	(3,6,7,11,13,16,19)	2,246,608,301	19.97
			0.6	(3,6,11)	1,036,207,334	0.16				0.4	(3,6,7,11,13,16,19)	1,884,255,634	24.39
		5	0.2	(3,5,6,11,13)	1,035,062,498	0.35				0.6	(0,3,5,6,11,13,16)	1,488,746,052	22.98
			0.4	(3,5,6,11,13)	910,894,524	0.33		25	3	0.2	(3,11,16)	4,419,802,743	0.57
			0.6	(3,5,6,11,13)	770,293,433	0.31				0.4	(3,11,16)	4,126,688,874	0.55
		7	0.2	(2,3,5,6,7,11,13)	972,435,245	0.60				0.6	(3,11,16)	3,757,210,099	0.54
			0.4	(2,3,5,6,7,11,13)	812,075,473	0.57			5	0.2	(3,6,11,13,16)	3,923,361,382	12.36
			0.6	(2,3,6,7,8,11,13)	642,445,430	0.58				0.4	(3,6,11,13,16)	3,480,980,837	12.25
	20	3	0.2	(3,6,16)	2,746,110,661	0.26				0.6	(3,6,11,13,16)	2,944,573,672	12.32
			0.4	(3,11,16)	2,573,607,594	0.26			7	0.2	(3,6,11,13,16,19,21)	3,667,744,683	174.46
			0.6	(3,11,16)	2,345,482,680	0.26				0.4	(3,6,11,13,16,19,21)	3,157,633,386	178.78
		5	0.2	(3,6,11,13,16)	2,384,807,231	2.03				0.6	(3,6,11,13,16,19,21)	2,558,023,286	178.46
			0.4	(3,6,11,13,16)	2,079,639,746	1.77							

2.4.3 Evaluation of Design Parameters

To assess the effect of parameters on network configuration for the resulting reliable hub networks and the extent to which inter-hub links are used, we conducted an experiment using the CAB dataset with $N = \{25\}$, $p = \{4, 5, 6\}$, $L = \{0, 1, 2, 3\}$, and $\alpha = \{0.2, 0.4, 0.6\}$. To examine the effect of node failure probabilities, we assumed the same failure probability, q, for all nodes such that $q = \{5\%, 10\%, 15\%, \ldots, 50\%\}$. Although failure probabilities over 20% are quite rare in real-life situations, we increased the values of q all the way up to 50% to gain a better insight on the effect of this critical factor. The following response variables were used to evaluate the configuration of the resulting reliable hub networks:

1. **Inter-hub Link Utilization**: expressed as the rate of flow that is transported over inter-hub links. This measure is calculated as the summation of flow multiplied by the length of the inter-hub link traversed by the flow over all O-D pairs divided by the total flow-distance traversed. The measure shows the degree at which the network

takes advantage of economies of scale at different levels of the factors.

2. **Hub Dispersion**: calculated as the average inter-hub distance over all hub pairs. This metric provides insight on the configuration of the network and how dispersed or concentrated the hubs get as parameters change.

Our goal is to explore the effect of the maximum number of simultaneous failures, L, the failure probability, q, and the discount factor, α, on the response variables described above.

Figures 2.6 and 2.7 summarize the observed trends for each of the performance metrics, respectively. By looking at the utilization of inter-hub links in Figure 2.6, we observe a monotonic increase for all values of $p = \{4, 5, 6\}$ as the discount factor grows larger, which is the main incentive for implementing a hub-and-spoke network topology. However, as more simultaneous failures are considered, i.e., larger values of L, inter-hub links usage reduces. The same trend is observed for the average failure probability, q. As a more conservative approach is considered by accounting for scenarios with larger number of simultaneous failures, the network shifts to using fewer hubs to connect O-D pairs - either a direct connection or only one hub in the connection are selected. The same behavior is observed as hubs become more and more susceptible to failure.

Regarding the layout of the resulting networks, Figure 2.7 shows that the discount factor significantly affects the layout of the network by locating hubs more sparsely. As one incorporates more failure scenarios, a significant jump in the average inter-hub distance is observed. This can be explained by the fact that the model selects a more balanced distribution of hubs as opposed to locating more hubs in dense parts of the network while a few hubs are established distant from all others. This is to avoid enduring a significant increase in cost caused by backups in scenarios where distant hubs fail. Notice that this behavior is observed only as L goes from 0 to 1. However, considering failure scenarios with more than one failure, i.e., $L > 1$, does not affect the layout significantly. Tran et al. (2016) report the same observation for hub networks with single allocation.

An interesting trend is noticed in hub dispersion as a function of the probability of failures, q. As illustrated in Figure 2.6, the increase in q initially does not have a significant effect on hub dispersion. However, as failure probabilities go over 40%, a jump in the average length of inter-hub links is observed. This can be explained by the fact that the model avoids inter-hub links to the extent that the topology becomes closer to a star network rather than a hub-and-spoke network. To further illustrate this behavior, Figures 2.8 and 2.9 illustrate an instance of CAB with 25 nodes and 5 hubs to be located when $L = 2$ when the probability

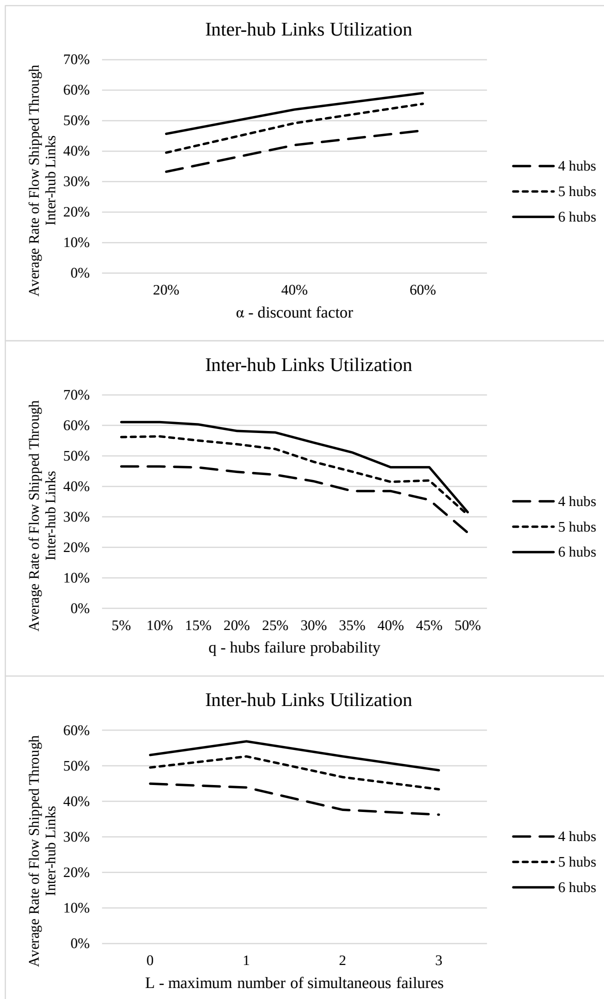


Figure 2.6: Experiment results on utilization of inter-hub links

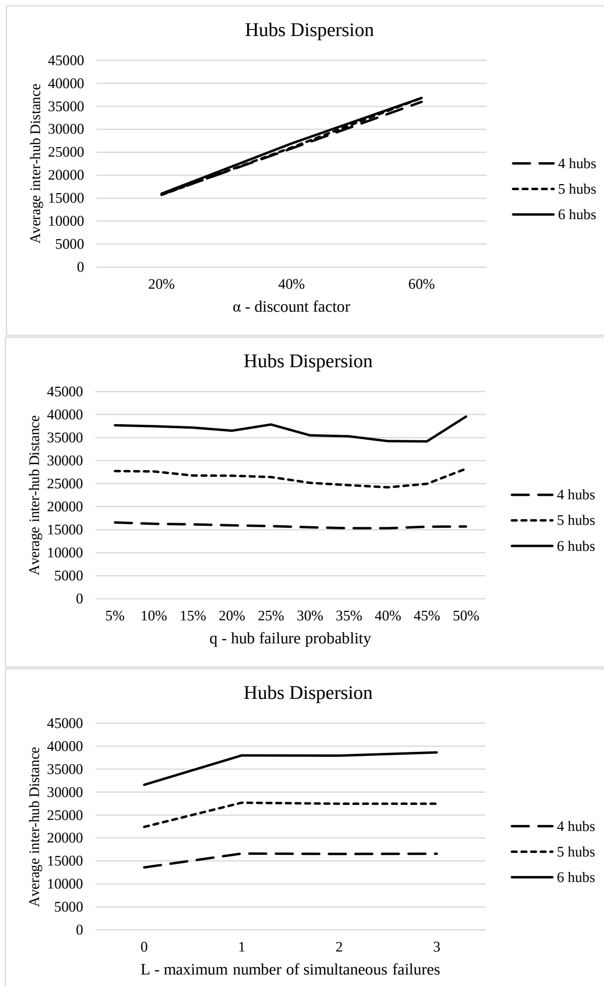


Figure 2.7: Experiment results on hubs dispersion

of failures for nodes are set to 10% and 90%, respectively. The intensity of a line represents the amount of flow that goes through the link. Inter-hub links are shown in red and the remaining ones are blue. The density of blue links is dominant in Figure 2.9. Allowing for multiple allocation of hubs to spokes makes way for such a shift in network layout and economies of scale utilization. In general, we can argue that designing a hub network with intrinsically unreliable nodes mitigates the potential benefits of using the hub-and-spoke topology.

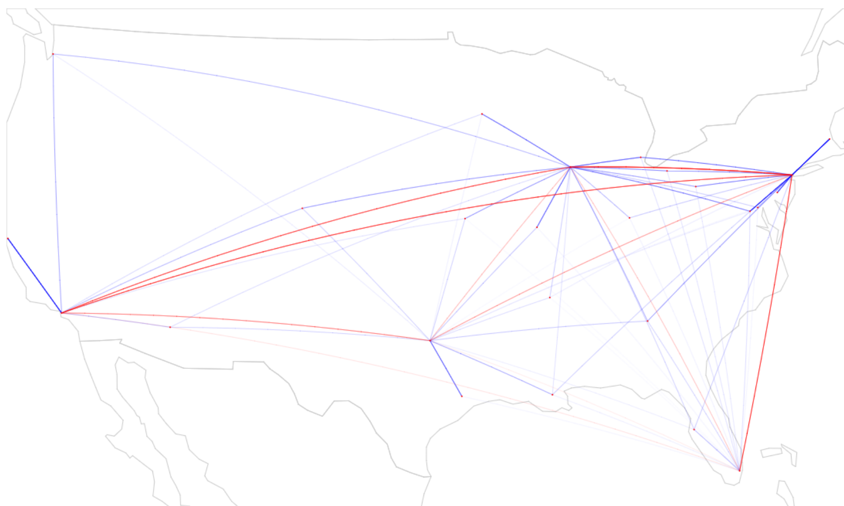


Figure 2.8: 25 node, 5 hub instance with low failure rate ($q = 10\%$)

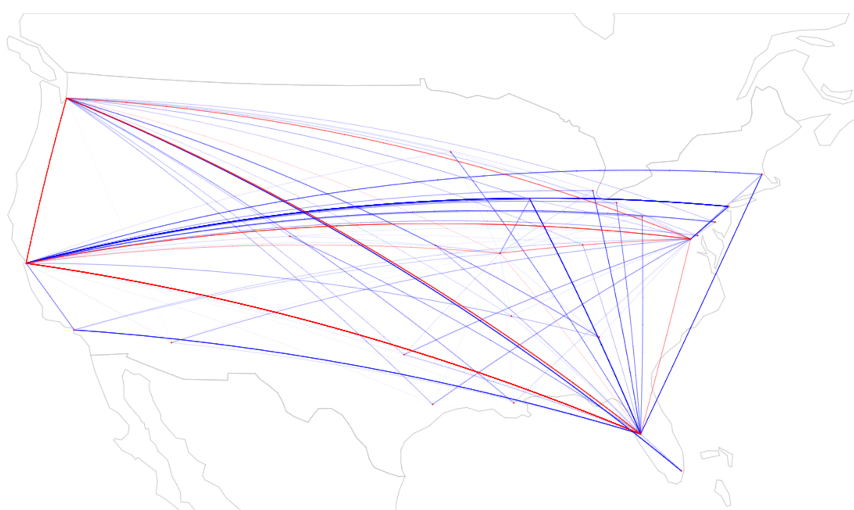


Figure 2.9: 25 node, 5 hub instance with high failure rate ($q = 90\%$)

2.5 Conclusions

Designing a reliable hub network under the risk of disruptions is a critical effort in practical applications such as transportation, logistics, and telecommunications. In this paper, we incorporated scenarios with more than one hub failure while allowing for multiple allocation of hubs to spokes. Using a binary tree structure, we developed a nonlinear mathematical model that minimizes the total expected cost of the network and determine optimal primary and backup connections to reroute flow in case of failures. We reduced the problem by fixing hubs and decomposed the reduced problem into finding the optimal primary and backup connections for each O-D pair separately. A search algorithm was developed to solve the problem efficiently. Our model showed that for cases where only one of the hubs along a connection fails, the best backup does not necessarily include the other active hub but may use a completely different hub(s) to reroute the affected flow. We also observed that higher failure probabilities reduce the benefits gained by consolidation of flow at hub nodes in a network. In general, higher failures push the network topology to a star rather than a hub-and-spoke topology.

Interestingly, we noted that considering scenarios with more than one hub failure does not significantly affect the layout of the network, i.e., the location of hubs. However, the extent to which the inter-hub links are used to make connections reduces. This implies the possibility of reducing the complexity of the problem by decomposing it into (i) locating hubs, and (ii) determining connections, especially for larger instances. In this case, hub locations can be determined only considering scenarios with one hub failure and then the connections can be easily determined while including more failure scenarios.

Similar to other studies in the literature, our research does not consider hub capacities. Moreover, in many realistic applications, reliability issues in a network are the outcome of inefficient operation of entities rather than a complete shutdown. Acknowledging this fact while designing a hub network can be challenging since (i) methods that measure or predict partial failures of network entities are complicated and highly dependent on the application, and (ii) even if such method exists, incorporating the partial failure in a hub network design model will make an already complicated problem much more difficult. However, incorporating capacity and partial disruptions must be sought as future research directions to make way for a realistic optimization of hub networks.

2.6 References

S. Alumur and B. Y. Kara. Network hub location problems: The state of the art. *European Journal of Operational Research*, 190(1):1–21, 2008.

Y. An, Y. Zhang, and B. Zeng. The reliable hub-and-spoke design problem: Models and algorithms. *Transportation Research Part B: Methodological*, 77:103–122, 2015.

N. Azizi. Managing facility disruption in hub-and-spoke networks: formulations and efficient solution methods. *Annals of Operations Research*, pages 1–27, 2017.

N. Azizi, S. Chauhan, S. Salhi, and N. Vidyarthi. The impact of hub failure in hub-and-spoke networks: Mathematical formulations and solution techniques. *Computers & Operations Research*, 65:174–188, 2016.

J. F. Campbell and M. E. O'Kelly. Twenty-five years of hub location research. *Transportation Science*, 46(2):153–169, 2012.

I. Contreras. Hub location problems. In G. Laporte, S. Nickel, and F. S. da Gama, editors, *Location science*, chapter 10, pages 266–290. Springer, Berlin, 2015.

S. Davari, M. H. F. Zarandi, and I. B. Turksen. The fuzzy reliable hub location problem. In *Fuzzy Information Processing Society (NAFIPS), 2010 Annual Meeting of the North American*, pages 1–6. IEEE, 2010.

R. Z. Farahani, M. Hekmatfar, A. B. Arabani, and E. Nikbakhsh. Hub location problems: A review of models, classification, solution techniques, and applications. *Computers & Industrial Engineering*, 64(4):1096–1109, 2013.

L. Kaufman and P. J. Rousseeuw. *Finding groups in data: an introduction to cluster analysis.* John Wiley & Sons, Brussels, Belgium, 1990.

H. Kim. p-hub protection models for survivable hub network design. *Journal of Geographical Systems*, 14(4):437–461, 2012.

H. Kim and M. E. O'Kelly. Reliable p-hub location problems in telecommunication networks. *Geographical Analysis*, 41(3):283–306, 2009.

H. Kim and M. S. Ryerson. Ad-hoc port assignment due to a port outage. In *Logistics Society Workshop*, page 83, 2013.

J. G. Klincewicz. Hub location in backbone/tributary network design: a review. *Location Science*, 6(1):307–335, 1998.

M. E. O'Kelly. The location of interacting hub facilities. *Transportation science*, 20(2): 92–106, 1986.

M. E. O'Kelly. A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research*, 32(3):393–404, 1987.

F. Parvaresh, S. H. Golpayegany, S. M. Husseini, and B. Karimi. Solving the p-hub median problem under intentional disruptions using simulated annealing. *Networks and Spatial Economics*, 13(4):445–470, 2013.

F. Parvaresh, S. M. Husseini, S. H. Golpayegany, and B. Karimi. Hub network design problem in the presence of disruptions. *Journal of Intelligent Manufacturing*, 25(4):755–774, 2014.

L. V. Snyder and M. S. Daskin. Reliability models for facility location: the expected failure cost case. *Transportation Science*, 39(3):400–416, 2005.

T. H. Tran, J. R. OHanley, and M. P. Scaparra. Reliable hub network design: Formulation and solution techniques. *Transportation Science*, 2016.

B. Yıldız and O. E. Karaşan. Regenerator location problem and survivable extensions: A hub covering location perspective. *Transportation Research Part B: Methodological*, 71: 32–55, 2015.

Chapter 3: Task Assignment with Flexible Execution Times and Sequence Dependent Travel[2]

Abstract

Task scheduling on the day of the operation is a common problem in labor intensive industries. In this study we consider the problem where each task may require multiple workers with different skills. A task is defined as an uninterruptible piece of work with a fixed duration which can be executed within a given time window. The travel times between tasks are different. The objective is to maximize the reward gained by executing tasks given a limited team of workers. We propose a branch-and-price algorithm with two heuristics for the pricing subproblem and solve instances of up to 400 tasks and 40 workers. We present the results of a numerical study to assess the effect of inputs on run time and quality of the solutions obtained. The problem is presented in the airline industry context where ground crew personnel located at an airport perform tasks to get flights ready for departure.

Keywords

Task assignment; sequence dependent travel; flexible execution times; integer programming; branch-and-price; great deluge

3.1 Introduction

Our study is motivated by the challenges associated with workforce scheduling in the airline industry. As airlines grow bigger and business rules become more complicated, finding a good schedule becomes more difficult, in particular when there is more activity at an airport during particular hours of the day. In some environments, even finding a feasible schedule can be a significant challenge. The vast majority of studies on airline workforce scheduling are focused on scheduling flight crews (i.e. pilots and flight attendants). However, airlines also have limited ground crew located at each airport to help with tasks associated with boarding and deplaning of passengers, and preparing the plane for departure. An airline, depending on its size can have a few dozens of flights arriving to and departing from a given airport. Given the limited number of gates available to an airline, there is a desire for the

[2]Under revision for publication in *Computers & Operations Research*

shortest idle times for planes between arrival and departure. A critical factor to achieve this goal is having an effective assignment of ground crew to the required tasks.

In this paper, we assume that the ground crew personnel located at an airport are already assigned to shifts with fixed start and end times. Each worker has a set of skills making them eligible to execute certain tasks. As flights arrive, a set of tasks with multiple skill requirements are needed to be completed. A few examples of those tasks are loading and unloading bags, deplaning and boarding passengers, fueling, cabin cleaning, and performing safety inspections. In practice, all tasks associated with incoming and outgoing flights have to be executed. Due to unanticipated changes in flight schedules and fluctuations in arrival/departure rates throughout the day, execution of all tasks is not possible unless some workers stay overtime. The decision on how to assign workers to stay overtime is subject to several business rules that are outside of the scope of this research. The goal of this study is to assign tasks to a limited number of workers on the day of operation so that as many tasks with the highest priorities are executed by workers with the required skills.

The realization of scheduled arrivals and departures in the airline industry is hardly ever 100% as severe weather and many other factors cause delays and cancellations. The cancellation or delay of a flight will affect task sequences assigned to ground crew workers. As these unanticipated changes are prone to happen several times during the day of operation, we require a reasonably fast and efficient solution approach to address the changes to the original schedule and promptly generate an updated task assignment plan. Therefore, our goal is to propose modeling and solution approaches that produce good quality solutions for realistically sized instances of this problem in acceptable computational times.

This paper makes the following contributions. We propose a linear integer programming model for the task assignment problem with flexible execution times and sequence dependent travel (TAP-FET-SDT). Then, we decompose the formulation to apply a branch-and-price algorithm. Two heuristics are developed for the subproblem; one is based on the branch-and-bound (BB) algorithm and the second is a great deluge (GD) heuristic. The performance of the heuristics is then evaluated and we further explore the effect of parameters such as the level of flexibility in task execution times on the performance of the solution methods and solution characteristics.

The remainder of this paper is organized as follows. In Section 3.2, we review the literature on the task assignment problem with worker movement. Section 3.3 includes the problem description along with the mathematical formulation. In Section 3.4, we propose a branch-and-price algorithm and two methods to solve the pricing subproblem. Our numerical study and findings on the performance of the proposed methods are discussed in Section 5. Conclusions and future research directions are presented in Section 6.

3.2 Literature Review

The literature on staff scheduling is vast. A very useful review of studies in this area can be found in Ernst et al. (2004). More recent studies on the topic are thoroughly categorized from different perspectives by Van den Bergh et al. (2013). They classified over 300 published studies under the staff scheduling umbrella based on several criteria such as personnel characteristics, types of decisions to be made, whether the studies consider time or skill requirements as soft or hard constraints, solution method, and uncertainty incorporation. As Lequy et al. (2012a) argue, the staff scheduling process is too complicated to be addressed all at once. As a result, several studies in the literature decompose the problem into multiple levels and tackle each one separately. A sequential view of the problem includes:

- *Personnel budgeting*: determines the staff size,
- *Days off scheduling*: assigns days off to personnel over a period of time (on a weekly or monthly basis),
- *Shift scheduling*: creates daily shifts to satisfy aggregated demand,
- *Shift assignment*: assignment of shifts to the available staff,
- *Task assignment*: assignment of tasks to be done on the day of operation to the staff.

Except for the task assignment problem (TAP), all other staff scheduling problems are solved at least a few days in advance to operation. The number of studies on the TAP is significant. However, only a few include travel time or transition of workers between physical locations of tasks as a part of their model. Therefore, we focus on studies that consider movements between physical locations when assigning tasks to workers. Bard and Wan (2006), presented the TAP with the objective of minimizing total transition between workstations by a homogeneous group of workers in a mail processing and distribution center. They assumed fixed duration and start time for jobs. The problem was modeled as a multi-commodity network flow problem with workers as commodities. The authors solved large instances of the problem with tabu search. In another study, Al-Yakoob and Sherali (2007) focused on assigning employees to serve at different gas stations. Their model takes the assignment process further by integrating decisions on assigning employees to shifts and day-offs as well. The proposed multi-objective model aims at minimizing the number of employees, maximizing overall satisfaction of personnel preferences for locations, shifts, and on/off days. The last component of their objective consist of balancing preference satisfaction over all employees. The authors suggested a two-stage heuristic method to solve instances of up to 29 locations. However, employee transitions between gas station is not expressed as a part of their model. Eveborn et al. (2009) considered a home care service where the staff are scheduled to visit elderly people in need of support at their homes. The goal is to have minimum transition of staff between clients as well as conforming to several business rules. Although they did not present a mathematical formulation, they proposed a repeated matching process which starts with a feasible assignment of staff to visits and repeatedly merges and splits the assignments to find feasible improvements.

Later, Lequy et al. (2012a) incorporated an interesting perspective on the TAP. In addition to tasks, they defined activities as interruptible pieces of work that can be broken into pieces. Each piece can be executed by a different personnel. The authors applied this consideration in the retail service business where long activities such as operating a cash register could take several hours. The authors considered personnel with multiple skill levels. In an extension of this work, Lequy et al. (2012b) suggested a two stage optimization model for the problem introduced in Lequy et al. (2012a). Furthermore, Lequy et al. (2013) extended their previous work by allowing for the assignment of multiple personnel to tasks. Their latter model puts more emphasis on assignment of activities than tasks. Lequy et al. (2013) reported results on instances with 40 tasks.

As an application in the airline industry, Kuo et al. (2014) worked on the TAP for customer service agents at large airports. They introduced multiple levels of skills for the agents with lunch and rest-break requirements. They defined minimum and target levels for the number of agents required for each task and assumed a fixed travel time between locations. Their goal was to meet the highest level of requirements in terms of the number and proficiency of agents required for each task. The travel times between locations are fixed as well as the task start times. They solved their model by introducing efficient cuts to a branch-and-bound search algorithm.

Although the literature on the TAP is extensive, we were not able to find to the best of our ability a study that explicitly includes all of the characteristics that we intend to address in an optimization model. No previous studies consider sequence dependent travel times, variability in execution times, and heterogeneous workers, all at the same time. Table 3.1 helps to better position our study in the related literature. Although considering movement between physical locations of tasks is a major characteristic of the mentioned studies, they all assume a fixed transition time/cost between locations. However, this does not apply to airline ground crew scheduling as the tasks are associated with flights arriving at and departing from different gates or even different zones within an airport. Depending on the size of the airport, the travel times can vary from two minutes to over 30 minutes.

Table 3.1: Comparison of articles on the task assignment problem with movements.

	Bard and Wan (2006)	Al-Yakoob and Sherali (2007)	Eveborn et al. (2009)	Lequy et al. (2013)	Kuo et al. (2014)	This paper
Heterogeneous workers	✗	✓	✓	✓	✓	✓
Multi-skill requirements	✗	✓	✓	✓	✗	✓
Sequence dependent movements	✗	✗	✓	✗	✗	✓
Variable task start times	✗	✗	✓	✓	✗	✓
Worker preferences	✗	✓	✓	✗	✗	✗
Mathematical model	✓	✓	✗	✓	✓	✓

3.3 Problem Description

The task assignment problem with flexible execution times and sequence dependent travel (TAP-FET-SDT) involves assigning tasks with specific skill requirements to ground crew personnel who are already assigned to shifts with fixed start and end times. Each worker has a certain set of skills and each task may require one or more workers with a given skill. For example, loading bags may require two workers who are certified to operate vehicles on airport runways. To accommodate multi-skill requirements, each task is mapped to a vector of numbers such that each element of the vector represents the number of required workers with that skill. Tasks may be located at different locations. Thus, assigning workers to tasks that are located closest to each other minimizes non-value added time and increases workforce utilization. Sequence dependent travel times significantly increase the complexity of the problem. All tasks are assumed to have deterministic duration. Some tasks, like boarding passengers, have to be executed at a fixed time, i.e., the arrival time of the flight. On the other hand, tasks such as fueling can be completed within a time window after an arrival. Accordingly, we categorize tasks into fixed versus variable tasks. Allowing for flexible tasks further increases the complexity of the problem. Tasks are not all the same in terms of importance. Thus, they are given rewards to represent their priority. Under a limited number of workers, the objective is to gain the highest possible reward by getting more high priority tasks assigned to workers. A reward associated with a task is achieved if and only if all the skill requirements of the task are satisfied, which means that enough workers with the required skill set are assigned to the task within the allowed time window. Assigning any number of workers that is less than the requirement leaves the task as unassigned. In practice, the ground crew supervisor asks employees to stay overtime to execute unassigned tasks such that all tasks are completed. Ad-hoc assignment of workers to overtime shifts follows a completely separate set of business rules and is out of the scope of this study. For simplicity purposes, we assume that unassigned tasks are discarded and their reward is lost.

Also in practice, the ground crew schedules are generated on the day of operation. Therefore, an ideal optimization process has to be able to plan for at least the 12 hours ahead. Normally, workers are split into workgroups that focus on a specific set of tasks. Workgroup size rarely exceeds 30 people and the number of tasks that each work-group has to complete goes up to a few hundred in big airports. Task duration may be as short as ten minutes and can go up to an hour which requires splitting the time horizon into very short time periods. Another property of TAP-FET-SDT which adds complexity is frequent changes in schedules as a result of cancellations, early arrivals, and delays. With every change in flight schedule, the ground crew schedule has to be updated which also entails the need for very short run times.

3.3.1 Mathematical Formulation

In this section we propose a binary integer programming (BIP) formulation for TAP-FET-SDT. But first, we introduce the notation needed for the BIP formulation.

Sets

- T: Set of tasks,
- W: Set of workers,
- T_w: Set of tasks that can be executed by worker w,
- $T_t^{'}$: Set of tasks succeeding task t. Task $t^{'}$ succeeds task t iff the latest start time of $t^{'}$ is later than the earliest finish time of t,
- $T_t^{''}$: Set of tasks that conflict with task t. Task t'' conflicts with task t iff both tasks cannot be executed by the same worker due to a time limitation. In other words, t and t'' are overlapping tasks,
- W_t: Set of workers who can execute task t, i.e., they have at least one of the skills required to execute t,

- Q: Set of qualifications, i.e., skills,
- P: Set of time periods in the planning horizon,
- P_t: Set of time periods in which task t can be started,
- P_{tw}: Set of time periods in which task t can be started by worker w,.

Indices

- $t, t', t'' \in T$ task indices,
- $w \in W$ worker index,
- $q \in Q$ qualification index,
- $p, p' \in P$ time period indices,

Parameters

- α_t: reward associated with executing task t,
- β_{wq}: 1 if worker w has qualification q, 0 otherwise,
- γ_{tq}: number of workers with qualification q required to execute task t,
- δ_t: execution length of task t,
- $\theta_{tt'}$: the travel time between tasks t and t',
- M: a very large number.

Decision Variables

- x_{tp}: equals to 1 if the execution of task t is set to be started at period p; 0 otherwise,
- y_{twp}: equals to 1 if task t is assigned to worker w and is to start in time period p; 0 otherwise,
- $z_{tt'wp}$: equals to 1 if both tasks t and t' are assigned to worker w such that task t starts at period p and task t' starts in any period after p; 0 otherwise.

Mathematical Model

$$Max \sum_{t\in T}\sum_{p\in P_t} \alpha_t \; x_{tp} \tag{3.1}$$

$$S.T. \sum_{p\in P_t} x_{tp} \leq 1 \qquad \forall t\in T \tag{3.2}$$

$$\sum_{t\in T} y_{twp} \leq 1 \qquad \forall w\in W, \forall p\in P \tag{3.3}$$

$$\sum_{w\in W_t}\sum_{p\in P_{tw}} \beta_{wq} \; y_{twp} \geq \gamma_{tq} \sum_{p\in P_t} x_{tp} \qquad \forall t\in T, \forall q\in Q \tag{3.4}$$

$$\sum_{t'\in T}\sum_{p'\in P_{tw}} y_{t'wp'} \leq M(1-y_{twp}) \qquad \forall w\in W, \forall t\in T_w, \forall p\in P_{tw} \tag{3.5}$$

$$\sum_{\substack{p'\in P_{t'w}\\ p'>p}} p' \; y_{t'wp'} - p \; y_{twp} \geq \delta_t + \theta_{tt'} - M(1-z_{tt'wp}) \qquad \forall w\in W, \forall t\in T_w, \forall t'\in T_w\cap T'_t, \forall p\in P_{tw} \tag{3.6}$$

$$z_{tt'wp} \geq y_{twp} + \sum_{\substack{p'\in P_{t'w}\\ p'>p}} y_{t'wp'} - 1 \qquad \forall w\in W, \forall t\in T_w, \forall t'\in T_w\cap T'_t, \forall p\in P_{tw} \tag{3.7}$$

$$2 \; z_{tt'wp} \leq y_{twp} + \sum_{\substack{p'\in P_{t'w}\\ p'>p}} y_{t'wp'} \qquad \forall w\in W, \forall t\in T_w, \forall t'\in T_w\cap T'_t, \forall p\in P_{tw} \tag{3.8}$$

$$z_{tt'wp} \leq 0 \qquad \forall w\in W, \forall t\in T_w, \forall t'\in T_w\cap T''_t, \forall p\in P_{tw} \tag{3.9}$$

$$\sum_{w\in W_t} y_{twp} \leq M \; x_{tp} \qquad \forall t\in T, \forall p\in P_t \tag{3.10}$$

$$x_{tp}, y_{twp}, z_{tt'wp} \in \{1,0\} \qquad \forall t,t'\in T, \forall w\in W, \forall p\in P \tag{3.11}$$

Objective function (4.15) maximizes the total reward gained by assigning tasks to the required number of workers with the appropriate qualifications. Constraints (4.16) prevent a task from being executed more than once. Constraints (4.17) enforce that each worker can only be assigned to execute at most one task at each time period. Constraints (4.18) specify that a task is covered if and only if enough workers with the required skills are assigned to execute the task. Constraints (4.19) guarantee that if a worker is assigned to start executing task t in period p, then the worker is not assigned to any task after p for the duration of task t. Sequence dependent travel times are represented by constraints (4.20), i.e., if a worker is assigned to execute task t' after t, the worker cannot be assigned to any task for the duration of t and the travel time between t and t'. Constraints (4.21) and (4.22) assign the value of 1 to the auxiliary variable $z_{tt'wp}$ if worker w is assigned to start executing task t in period p and start task t' in any period after that, and 0 otherwise. Constraints (4.23) enforce that overlapping tasks cannot be assigned to the same worker. These constraints act as effective cuts and are not required in the formulation as Constraints (4.20) already prevent the assignment of conflicting tasks to a worker. Constraints (4.24) enforce that a task can be assigned to workers if and only if the task is selected to be executed. These constraints also ensure that the workers who are assigned to the same task begin execution at the same time. Constraints (4.25) enforce binary decision variables.

3.4 Branch-and-Price Algorithm

TAP-FET-SDT can be formulated by enumerating all possible task sequences for each worker. A sequence-based model will easily become intractable due to the large number of feasible sequences that may exist in a realistic problem instance. Therefore, we take a branch-and-price approach to solve medium to large instances of the problem.

3.4.1 Restricted Master Problem

Constraints (4.18) are complicating as they link together tasks and workers satisfying the qualification (i.e., skill) requirements. We now propose a decomposition of the TAP-FET-SDT. The restricted master problem (RMP) includes a subset of feasible task sequences $\tilde{S}_w \subset S_w$ for each worker $w \in W$. At a high level, the RMP seeks to select one task sequence, s_l, per worker such that the highest overall reward is gained. The RMP can be solved directly using a commercial solver.

$$RMP: \; Max \sum_{t \in T} \sum_{p \in P} \alpha_t x_{tp} \tag{4.15}$$

$$S.T \sum_{p \in P_t} x_{tp} \leq 1 \qquad \forall t \in T \tag{4.16}$$

$$\sum_{l \in \tilde{S}_p} s_l = 1 \qquad \forall w \in W \tag{3.12}$$

$$\sum_{w \in P} \sum_{l \in \tilde{S}_w} \psi_{tpql} \; s_l \geq \gamma_{tq} \; x_{tp} \qquad \forall t \in T, \forall q \in Q, \forall p \in P_t \tag{3.13}$$

$$x_{tp}, s_l \in \{0, 1\} \qquad \forall t \in T, \forall w \in W, \forall l \in \tilde{S}_w \tag{3.14}$$

Constraints (3.12) select exactly one sequence l, for each worker w. Constraints (3.13) are equivalent to Constraints (4.18). Indicator parameters ψ_{tpql} equal to 1 if (i) sequence l includes task t, (ii) task t is executed in period p, and (iii) the worker associated with sequence l has qualification q. Constraints (3.14) are the variable type constraints.

3.4.2 Pricing Subproblem

The pricing subproblem SP_w seeks to generate a feasible sequence/column for worker $w \in W$ that can improve the RMP solution. Dual variables for the linear programming relaxation of the RMP $u4.16_t$, $u3.12_w$ and $u3.13_{tqp}$, obtained from constraints (4.16), (3.12) and (3.13) respectively, are part of the objective function of our pricing problem.

$$SP_w: \ Min \ u3.12_w + \sum_{t \in T_w} \sum_{p \in P_{tw}} u4.16_t \ y_{twp} + \sum_{t \in T_w} \sum_{p \in P_{tw}} \sum_{q \in Q} u3.13_{tqp} \ y_{twp} \tag{3.15}$$

$$S.T. \ \sum_{t} y_{twp} \leq 1 \qquad \forall p \in P_w \tag{4.17}$$

$$\sum_{t' \in T} \sum_{p' \in P'_{tp}} y_{t'wp'} \leq M(1 - y_{twp}) \qquad \forall t \in T_w, \forall p \in P_{tw} \tag{4.19}$$

$$\sum_{\substack{p' \in P_{t'w} \\ p' > p}} p' \ y_{t'wp'} - p \ y_{twp} \geq \delta_t + \theta_{tt'} - M(1 - z_{tt'wp}) \quad \forall w \in W, \forall t \in T_w, t' \in T_w \cap T'_t, \forall p \in P_{tw} \tag{4.20}$$

$$z_{tt'wp} \geq y_{twp} + \sum_{\substack{p' \in P_{t'w} \\ p' > p}} y_{t'wp'} - 1 \qquad \forall w \in W, \forall t \in T_w, t' \in T_w \cap T'_t, \forall p \in P_{tw} \tag{4.21}$$

$$2 \ z_{tt'wp} \leq y_{twp} + \sum_{\substack{p' \in P_{t'w} \\ p' > p}} y_{t'wp'} \qquad \forall w \in W, \forall t \in T_w, t' \in T_w \cap T'_t, \forall p \in P_{tw} \tag{4.22}$$

$$z_{tt'wp} \leq 0 \qquad \forall w \in W, \forall t \in T_w, \forall t' \in T_w \cap T''_t \tag{4.23}$$

$$y_{twp}, z_{tt'wp} \in \{1, 0\} \qquad \forall t, t' \in T_w, \forall p \in P_{tw} \tag{3.16}$$

Objective function (3.15) maximizes the value by which the RMP can be improved through entering a new sequence for worker w. Constraints (4.17) and Constraints (4.19) to (4.23) serve the same purpose as explained above. Constraints (16) are variable type constraints. The pricing subproblem is an integer programming model with many constraints to generate a feasible sequence. As solving the model directly can be computationally expensive, we propose a branch-and-bound algorithm to solve reasonable instances of the problem efficiently.

3.4.3 Branch-and-Bound Algorithm for Pricing Subproblem

We convert the pricing subproblem into a maximization problem and propose a branch-and-bound algorithm to solve it to optimality. $-f(SP_w)$ can be interpreted as a knapsack problem with additional constraints where fixed tasks (y_{twp}) are the items. We define fixed task (FT) as a task t that is mapped to a fixed start period p. For instance, assume we decide to break down the planning horizon into five minute periods. Then, we can break down a 15 minute flexible task with execution start time between 9:00 and 9:30 into 6 FTs with the same duration and start times at 9:00, 9:05, 9:10, ..., and 9:30. Each node of the search tree represents a sequence of FTs which has the following attributes:

- *Assigned set*: set of FTs that are included in the sequence,

- *Unassigned set*: set of FTs that can be added to the sequence, meaning that they do not conflict with any FT in *Assigned set* individually,

- *Actual value*: summation of pricing problem objective function coefficients over all FTs in *Assigned set*,

- *Upper-bound*: *Actual value* of the node plus maximum achievable value from *Unassigned set*. Maximum achievable value of a set of FTs, G, is calculated by Equation (3.17), where c_{tp} is the pricing problem objective function coefficient of the FT associated with task t and period p. V_G is the set of tasks that have at least one FT in G.

$$\text{Max Achievable Value}(G) = \sum_{t \in V_G} \max_p c_{tp} \tag{3.17}$$

To improve the search, the algorithm starts by enumerating FTs for all tasks, calculating a score for each, and sorting them in descending order of the scores. Equation (3.18) gives the score for a FT denoted by x. Intuitively, a fixed task that has a larger objective function coefficient and conflicts with a smaller number of FTs will get a higher score. Such FT improves the objective function value more and prevents smaller number of other FTs from joining the sequence. The root node is initialized with an empty *Assigned set*. The sorted list of FTs forms the root's *Unassigned set*. *Actual value* and *Upper-bound* are calculated accordingly. The branching process is done by removing the first FT from the *Unassigned set* and adding it to the Assigned set in the left child or discarding it in the right child. In the left child, all FTs in *Unassigned set* that conflict the newly added FT must be removed. A node is fathomed if its *Unassigned set* is empty or if its *Upper-bound* is worse than the best *Actual value* found so far.

$$Score(x) = \frac{\text{obj. fun. coefficient of } x}{\text{average obj. fun. coefficient of FTs that conflict } x} \tag{3.18}$$

Limiting the depth of the search tree will turn the algorithm into a heuristic with high quality solutions. We will discuss the performance of the heuristic in Section 3.5.

3.4.4 Great Deluge Algorithm for Pricing Subproblem

As an alternative solution method to solve large instances of TAP-FET-SDT, we developed a great deluge algorithm (GD) to solve each iteration of the pricing subproblem extremely fast. Dueck (1993) introduced GD as a simple yet powerful heuristic algorithm that compared to well-known heuristics like genetic algorithms and tabu search needs considerably less computing time. GD is a single parameter Monte Carlo heuristic that yields good quality solutions with a good choice of only one parameter as opposed to a series of parameters required by many popular meta-heuristics. This characteristic makes the method more robust. The branch-and-price algorithm requires solving a pricing subproblem numerous times. Thus, a heuristic that is (i) very fast and (ii) produces relatively good solutions will complement the branch-and-price procedure by introducing more columns to the RMP compared to more sophisticated heuristics that are prone to yield a higher quality solution in a single run. We follow the same pseudocode presented by Dueck (1993) with a slight modification to obtain a solution to the pricing subproblem. Algorithm 3 demonstrates the steps to solve the pricing subproblem. We represent a solution/sequence as a vector of binary values. Each element of the array is associated with a FT. FTs that are included in the sequence are set to 1 and otherwise 0. To start from a good initial solution, we enumerate and sort all FTs as explained in Section 3.4.3. Starting from the top, we add the FT with the highest score to the sequence and remove all conflicting FTs from the list. The same steps are repeated until the list is empty. The random move to a neighborhood involves flipping a random element in the solution vector. It should be noted that the chance of finding a feasible sequence by randomly searching the solution space is very small. To improve the algorithm, we introduce a procedure to ensure the feasibility of the solutions obtained after random moves.

Algorithm 4 ensures feasibility of the newly created solution in Algorithm 3 (see Algorithm 3: Line 10). Figure 3.1 demonstrates two examples of fixing an infeasible sequence.

Algorithm 3 Great deluge algorithm for the solving the pricing subproblem.

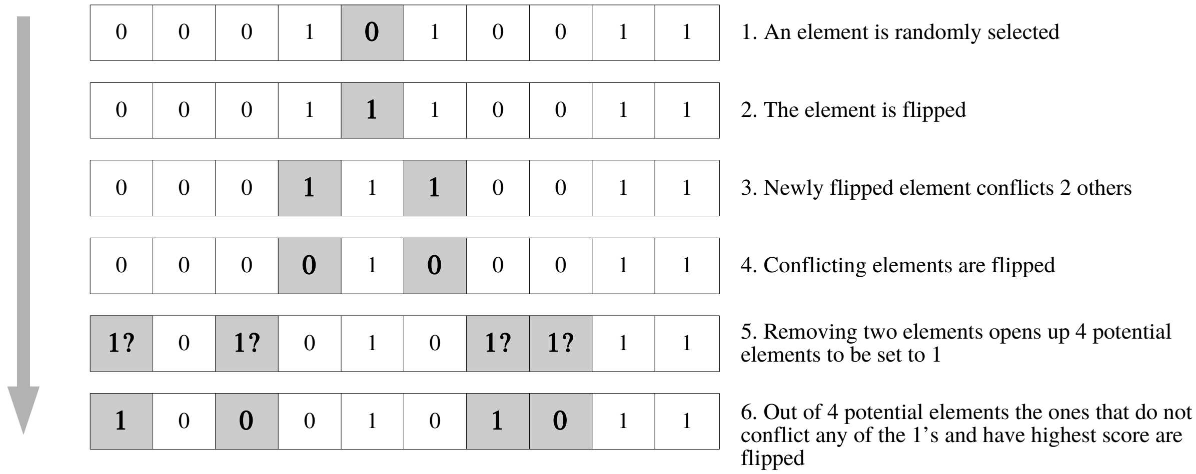
```
procedure GD
    Choose an initial solution
    BEST-SOLUTION ← initial solution
    WATER-LEVEL ← value of the initial solution
    RAIN SPEED ← 0.01
    OLD-SOLUTION ← initial solution
    while stopping criterion is not met do
        NEW-SOLUTION ← randomly flip one of the elements in the OLD-SOLUTION
        if NEW-SOLUTION is infeasible then
            make NEW-SOLUTION feasible
        if value of NEW-SOLUTION is better than WATER-LEVEL then
            OLD-SOLUTION ← NEW-SOLUTION
            WATER-LEVEL ← (1+RAIN-SPEED)× WATER-LEVEL
            if NEW-SOLUTION is better than BEST-SOLUTION then
                BEST-SOLUTION ← NEW-SOLUTION
return BEST-SOLUTION
```

Algorithm 4 Fixing an infeasible solution to the pricing subproblem.

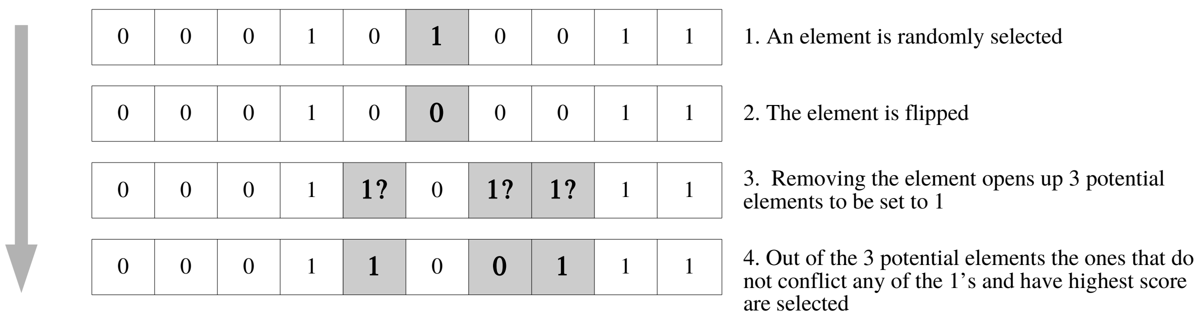
```
 1: procedure Fix
 2:     if the newly flipped element is set to 1 then
 3:         set all elements that conflict the changed element to 0
 4:         POTENTIAL-SET ← all elements with value 0
 5:         while POTENTIAL-SET is not empty do
 6:             NEW-ELEMENT ← the element that has the highest score in POTENTIAL-SET
    and does not conflict any of the elements with value 1
 7:             set NEW-ELEMENT to 1
 8:             remove NEW-ELEMENT and all its conflicting elements from POTENTIAL-SET
 9:     else
10:         go to line 4
```

3.5 Numerical Study

In this section, we evaluate the performance of the proposed solution methods. One method is branch-and-price with branch-and-bound for the pricing subproblem (BP-BB), and the other method is branch-and-price with great deluge for the pricing subproblem (BP-GD). The evaluation is done with respect to the best bound obtained by CPLEX for small to medium size instances of a realistically generated dataset. We further expand our experiments by examining the effect of changes in input, other than problem size, on the performance of our heuristics as well as the solutions to the TAP-FET-SDT. The proposed algorithms were



(a) Example 1



(b) Example 2

Figure 3.1: Steps taken to move to a new feasible neighbor.

developed in Java 8 and use CPLEX 12.7.1.0 as solver. All experiments were run on a 64-bit Windows 10 workstation (Intel Xeon CPU E3-1240 @ 3.40 GHz, 12.0 GB of RAM).

3.5.1 Problem Instances

A dataset was generated based on realistic requirements observed in the airline industry. The dataset consists of 400 tasks and 40 workers. The rewards gained by executing tasks are uniformly distributed in the range (0, 1]. 20% of the tasks are fixed, i.e., they have to begin precisely at a given time. The remaining 80% of tasks have execution time windows, i.e., time intervals in which a task has to begin, that are specified in minutes and belong to the set $\{5, 10, 15, 20, 25\}$. Tasks are distributed in four location zones with average travel times of $\{5, 10, 15, 20\}$ minutes. The major characteristics of the data are:

3.5.2 Performance Evaluation

In practice, due to cancellations, delays, and early arrivals, the TAP-FET-SDT has to be solved multiple times on a daily basis. Therefore, for both heuristics, we limit the total run time to three minutes by stabilizing the column generation process after two minutes and report the integer solution to the RMP within one minute. The solution obtained from BP-BB and BP-GD are compared to the integer solution and the best bound obtained by directly solving the BIP model presented in Section 3.1 using CPLEX with a one hour limit on run time. We solved 56 instances resulting from number of tasks in increments of 10 from 30 to 100, and number of workers in increments of 5 from 10 to 40. Table A.1 in the Appendix shows the run time and optimality gap for each of the three solution methods (i.e., CPLEX, BP-BB, and BP-GD) over all instances. On average, solving the problem with CPLEX considering a one hour run time limitation yields a 0.48% optimality gap as opposed to 1.81% and 0.94% observed from the application of BP-BB and BP-GD, respectively.

Figure 3.2 shows a closer look at the performance of the three approaches for instances with 10, 15 and 20 workers. There are a few noteworthy observations from Figure 3.2. First, one would expect an increase in computational complexity as the problem size grows. However, counter-intuitively, we observed a significant decrease in computational run time for instances with more workers. Unfortunately, we were not able to identify the underlying reason causing this behavior in TAP-FET-SDT. Second, as we move towards a larger number of tasks and workers, the optimality gap of both heuristics reduces considerably which shows the heuristics effectiveness. Note that one reason behind larger optimality gaps in instances with small number of workers is that the optimal number of executable tasks is normally small in those instances. Hence, missing one or few tasks leads to a substantial increase in the optimality gap. Third, as the number of tasks and workers exceed 80 and 30, respectively, both BP-BB and BP-GD begin to outperform CPLEX. In almost all instances, BP-GD performs very close to CPLEX, if not better. Interestingly, BP-BB rarely outperforms BP-GD. To confirm this observation, we solved a different set of large instances with up to 400 tasks and 40 workers using both heuristics. Figure 3.3 compares the quality of the solutions provided by the two heuristics for these new instances. In instances with fewer workers, the BP-GD yields significantly higher objective function values. Out of 133 instances with tasks in $\{40, 60, 80, \ldots, 400\}$ and workers in $\{10, 15, \ldots, 40\}$, BP-GD outperformed BP-BB in 94 cases by an average of 1.39%. BP-BB performed better in 27 cases with an average of 1.15%. The two methods yielded the same solution in 12 cases.

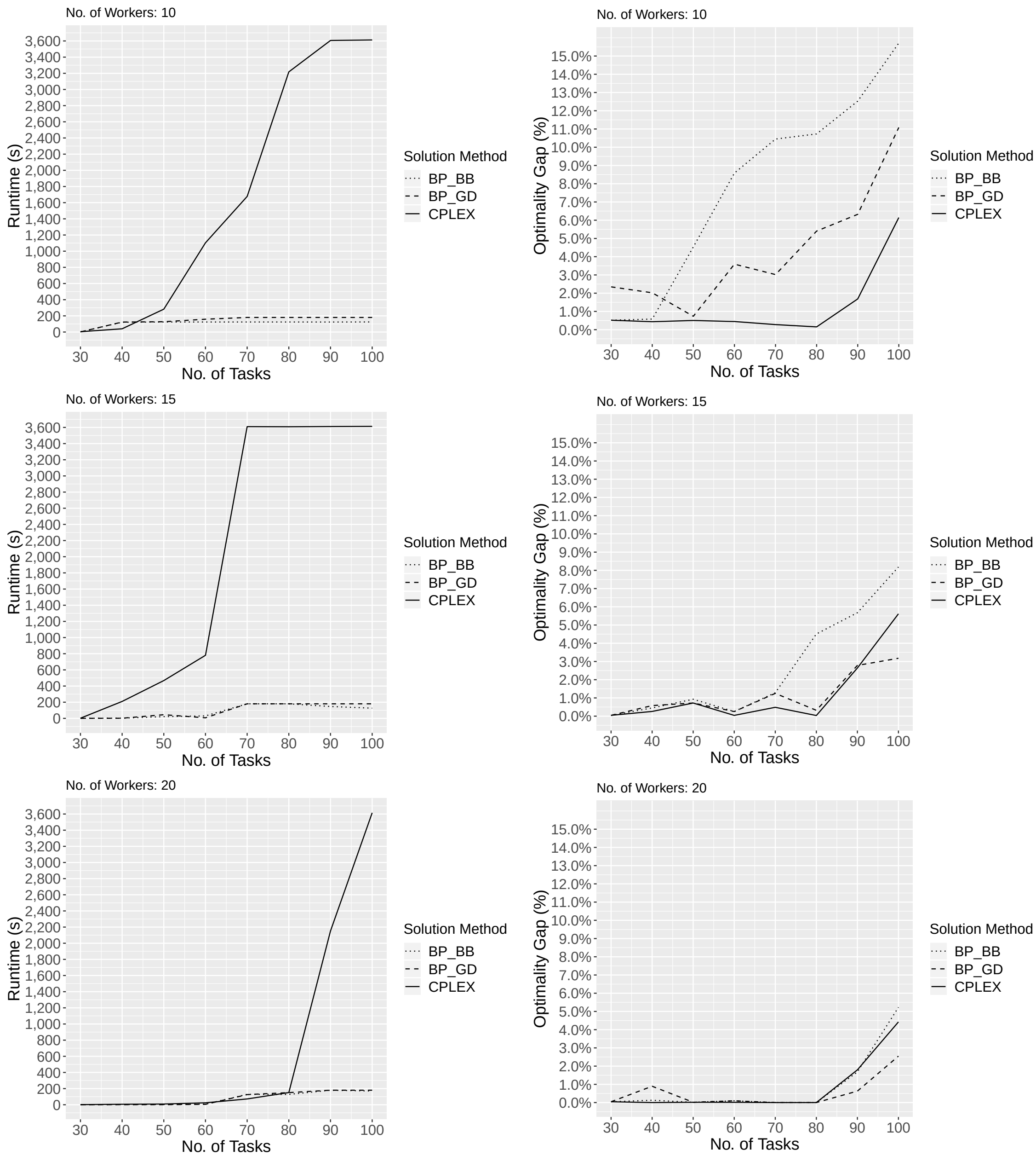


Figure 3.2: Performance of proposed solution methods.

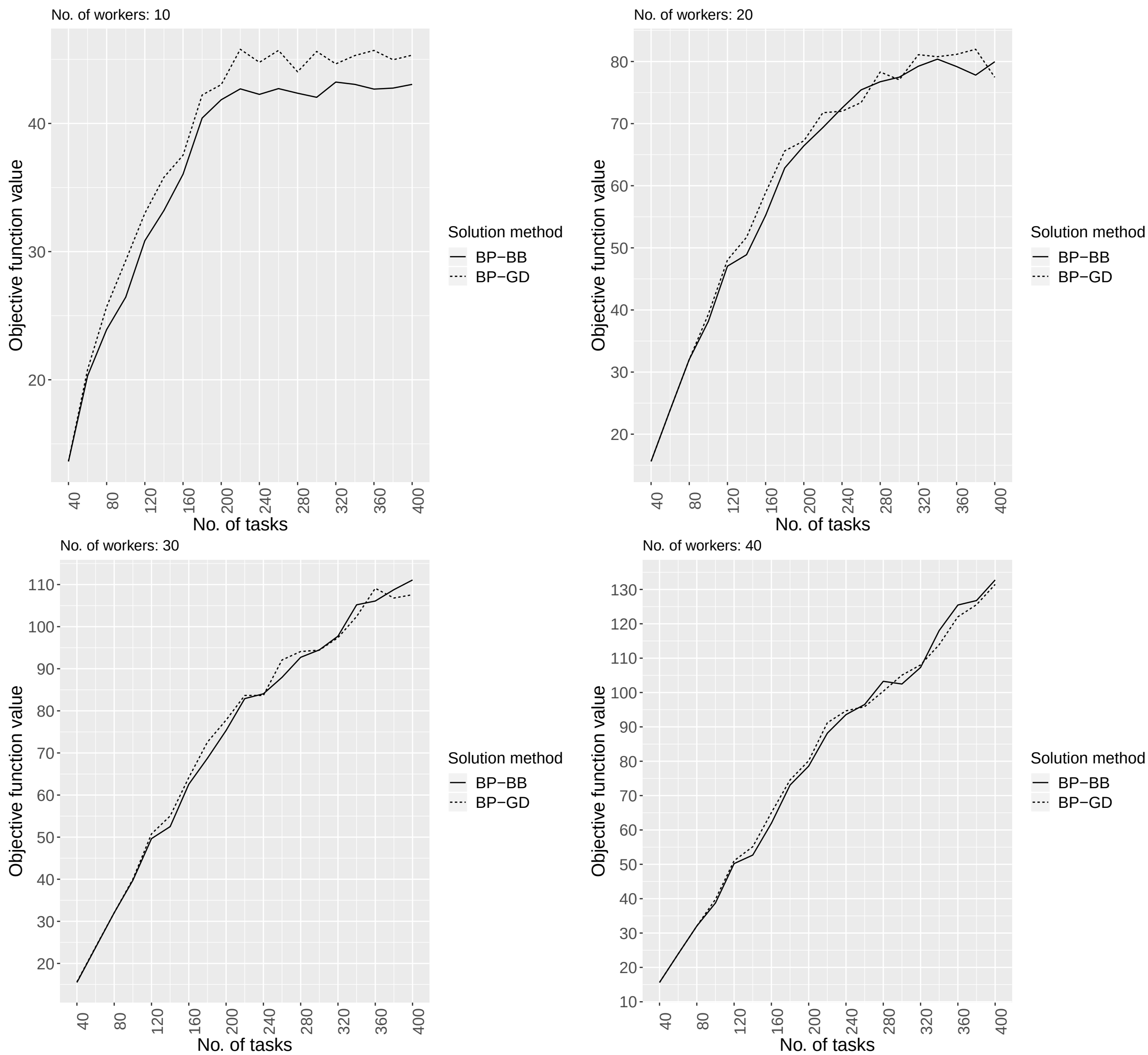


Figure 3.3: BP-BB and BP-GD performance comparison for additional instances.

3.5.3 Effects of Larger Time Windows and Fewer Flexible Tasks

To gain more insights about TAP-FET-SDT, we modified the dataset to capture the effect of having larger time windows and fewer flexible tasks on problem complexity and solutions. In the original dataset, 80% of the tasks can be executed within a given time window, i.e., they are flexible tasks. The minimum, maximum, mean, and standard deviation of the time windows used in the original dataset are 5, 25, 12.41, and 5.92 minutes, respectively. To assess the effect of execution time flexibility, we modified the original dataset to create two new datasets:

1. *Larger Time Windows*: the execution time window, i.e., time window in which a task

needs to begin, for each flexible task is doubled, and

2. *Fewer Flexible Tasks*: the number of flexible tasks are reduced by half by randomly selecting 50% of the flexible tasks in the original dataset and fixing them to a start time equal to the lower bound of the original time window.

Figure 3.4a shows the percentage of assigned tasks and Figure 3.4b shows the objective function value for the three alternative datasets, i.e., the original dataset as well as the two alternative datasets. As one would expect, having larger time windows to execute tasks should lead to a schedule with more executed tasks and higher objective function values. However, looking at Figure 3.4a, this is only met on the smaller instances.

(a) Solution quality in terms of percentage of assigned tasks.

(b) Solution quality in terms of objective function value.

Figure 3.4: Solution quality of BP-GD for the three alternative datasets as a function of task size with 40 workers.

As the number of tasks grows over 100, fewer tasks are assigned for execution and the objective function value decreases. Larger time windows make way for better schedules, however they further increase the complexity of TAP-FET-SDT. Therefore, finding a solution that takes advantage of the flexibility requires more time. For the same reason, in larger instances with over 160 tasks, the objective function value and the percentage of assigned tasks for instances with larger time windows are less than that of instances with fewer flexible tasks.

3.5.4 Worker Utilization

From a different perspective, we examined the workload balance across workers for solutions obtained in our computational testing. Table 3.2 presents the solution to an instance of TAP-FET-SDT with 200 tasks and 20 workers. The "Utilization" column shows the utilization of each worker which is computed by considering the ratio of the total time the worker spends executing tasks over the length of the worker's shift. The variability in the worker utilization values shows that workload is highly unbalanced across workers, which in practice will lower worker satisfaction levels. This trend is observed across almost all instances of TAP-FET-SDT. Figure 3.5 further highlights this point by showing box plots of worker utilization for instances with different number of tasks.

Table 3.2: Solution to TAP-FET-SDT with 200 tasks and 20 workers

Worker ID	Shift Length	Total Exec. Time	Utilization	# of Assigned Tasks	Execution Times	Travel Times	Idle Times
0	240	125	52.08%	11	10-10-10-10-10-10-25-10-10-10-10	0-10-10-0-10-5-0-0-0-0	30-0-10-15-0-0-10-5-0-0-10-0
1	240	95	39.58%	5	25-10-10-25-25	10-10-15-0	15-0-40-50-5-0
2	510	210	41.18%	12	10-10-25-25-10-10-25-10-25-25-10-25	0-0-15-0-0-0-15-0-10-5-0	0-5-55-75-5-10-5-0-0-15-75-5-5
3	510	150	29.41%	9	10-25-10-25-25-10-10-25-10	0-10-10-0-10-0-0-5	25-10-5-120-20-10-60-60-5-10
4	510	210	41.18%	12	10-25-25-10-25-25-25-25-10-10-10-10	10-10-15-15-15-15-0-0-15-0-5	15-15-10-45-15-0-10-45-0-35-10-0-0
5	510	165	32.35%	7	25-10-25-60-10-25-10	10-10-0-10-0-5	45-5-165-15-5-60-5-10
6	510	250	49.02%	19	10-10-10-25-10-10-25-10-10-10-10-10-10-10-10-10-25-25-10	0-10-5-0-0-0-15-0-0-0-0-10-5-5-5-0-15-0	0-15-30-55-30-5-0-0-0-5-0-0-0-25-0-35-0-0-0–10
7	240	80	33.33%	5	25-10-25-10-10	10-0-15-5	20-0-0-45-40-25
8	240	160	66.67%	13	10-10-10-10-25-10-10-10-10-10-25-10-10	10-0-0-15-0-0-0-0-0-0-0-0	5-35-0-0-0-5-0-0-0-10-0-10-0–10
9	240	185	77.08%	14	10-25-10-10-10-25-10-10-10-10-10-25-10-10	10-0-0-0-15-0-0-0-0-0-0-5-5	5-5-5-0-0-0-5-0-0-0-10-0-0-0–10
10	240	105	43.75%	4	60-10-25-10	10-15-0	10-0-0-80-20
11	240	120	50.00%	9	10-10-10-25-10-10-25-10-10	0-0-5-0-0-5-10-0	10-0-0-55-30-5-0-0-0-0
12	510	295	57.84%	14	10-10-25-25-25-10-25-25-25-25-10-10-10-60	5-15-0-0-10-5-5-5-0-0-0-0-0	25-65-35-20-0-0-0-0-5-5-0-0-0-0-15
13	510	180	35.29%	12	25-10-25-10-10-10-25-10-25-10-10-10	0-10-0-5-5-0-10-10-10-15-0	0-10-175-0-20-0-0-0-0-10-0-5-45
14	510	350	68.63%	16	25-60-25-10-10-10-10-25-10-25-25-25-10-10-10-60	0-15-0-5-0-10-0-10-5-5-5-0-0-0-0	0-10-25-20-0-0-0-0-0-0-0-5-30-0-0-0-15
15	510	240	47.06%	8	60-25-10-60-25-25-25-10	15-0-15-0-15-0-0	30-50-35-20-10-10-0-0-70
16	510	105	20.59%	6	25-25-10-10-10-25	0-0-0-0-0	95-45-10-5-35-110-105
17	510	115	22.55%	7	25-25-25-10-10-10-10	15-0-15-0-15-15	0-55-30-50-0-100-25-75
18	510	250	49.02%	9	25-10-10-60-10-25-25-25-60	0-0-0-5-5-5-10-0	35-15-25-15-0-5-15-0-25-100
19	510	255	50.00%	21	10-10-10-10-10-25-10-10-10-10-10-10-25-10-10-25-10-10-10-10-10	0-0-0-0-0-15-0-0-0-0-10-0-0-5-0-0-5-0-5-15	0-0-0-0-10-0-0-0-0-5-0-10-0-10-0-0-0-0-5-5-0-155

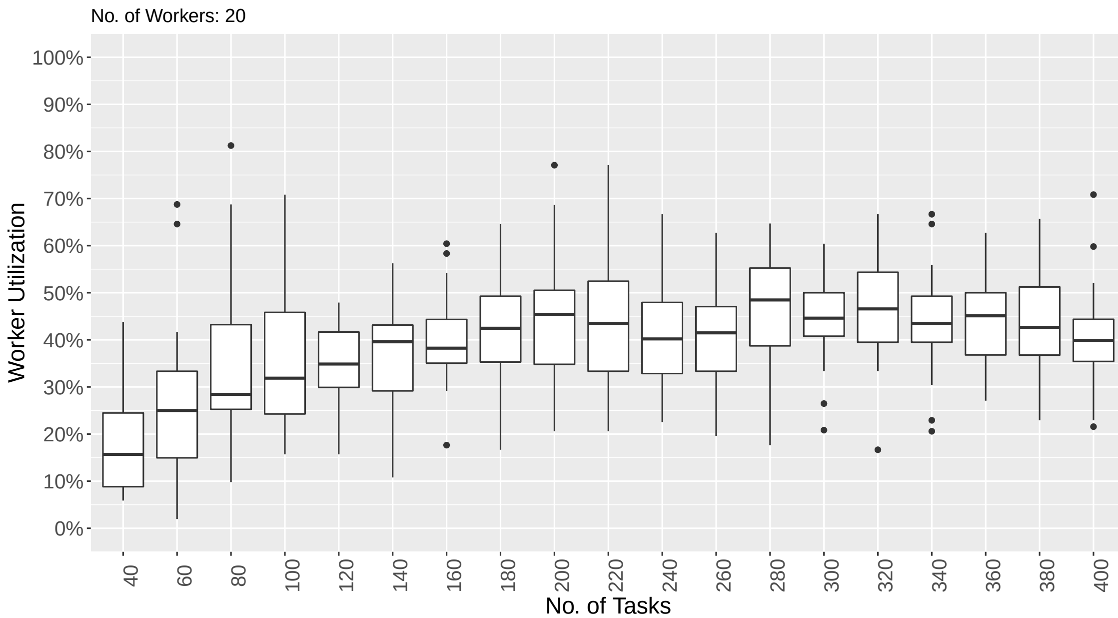


Figure 3.5: Worker utilization variability for instances with 20 workers.

Incorporating workload balancing constraints to the formulation of TAP-FET-SDT would improve the practicality of solutions obtained by reducing the variance in worker utilization over all workers. This comes at the cost of a decrease in the overall workforce utilization. To reduce workload imbalance among workers, we can modify the formulation of TAP-FET-SDT to enforce a cap on the difference on worker utilization between each pair of workers. By adding Constraints (3.19) to the formulation presented in Section 3.3.1 we ensure that the maximum workload imbalance does not exceed a certain threshold, B. Note that ψ_w denotes total availability of worker w.

$$\left| \frac{1}{\psi_w} \sum_t \sum_p \delta_t . y_{twp} - = \frac{1}{\psi_{w'}} \sum_t \sum_p \delta_t . y_{tw'p} \right| \leq B \qquad \forall w, w^{'} \in W \qquad (3.19)$$

To linearize the formulation, we replace Constraints (3.19) with Constraints (3.20) and (3.21).

$$\frac{1}{\psi_w}\sum_t\sum_p \delta_t.y_{twp} - \frac{1}{\psi_{w'}}\sum_p\sum_t \delta_t.y_{tw'p} \leq B \qquad \forall w, w' \in W \tag{3.20}$$

$$\frac{1}{\psi_w}\sum_t\sum_p \delta_t.y_{twp} - \frac{1}{\psi_{w'}}\sum_p\sum_t \delta_t.y_{tw'p} \geq -B \qquad \forall w, w' \in W \tag{3.21}$$

We used CPLEX to solve the modified model on instances with tasks in $\{30, 40, \ldots, 100\}$ and workers in $\{10, 15, 20\}$ to gain insights on how enforcing the workload balancing constraints would affect the complexity and the solution for TAP-FET-SDT.

Analyzing the results in Table 3.3, we conclude that the trade-off of enforcing workload balancing constraints can be desirable, as these constraints seem to reduce the solution space (i.e., act as effective cuts), with only a minimal average reduction in the objective function value (less that 1%). This claim only holds when the workload imbalance allowance, $B \in [0, 1]$, is not too tight (20% and larger). It is worth mentioning that, with a slight modification, we can still use both BP-BB and BP-GD to solve TAP-FET-SDT with workload balancing constraints. The incorporation of workload balance constraints only affects the RMP and does not significantly change the structure of the pricing problem.

3.6 Conclusions and Future Work

Assigning available workers to predefined tasks with execution time windows that are spread across multiple locations can be a complicated problem faced by businesses, especially in the airline industry. Airlines aim at efficient assignment of their ground crew personnel to tasks that need to be completed on flights in order to shorten their turnaround. In this paper, we incorporate tasks with flexible execution times, multiple skill requirements, and multiple locations. We developed a linear integer programming model that minimizes the total reward gained by executing tasks with workers with the required set of skills. To solve large instances of this problem, we proposed a branch-and-price approach with two heuristics to solve the pricing problem which focuses on finding new task sequences for each worker. The first heuristic is a branch-and-bound search algorithm. The second heuristic is based on the great deluge algorithm which is easy to implement.

Computational testing on different instances shows that even though the branch-and-bound heuristic generates better columns at each iteration, the solution obtained by using

Table 3.3: Comparison of objective function values and run times in CPLEX for TAP-FET-SDT with and without workload balancing constraints and different workload imbalance allowances.

No. of Tasks	No. of Workers	Workload imbalance allowance N/A Obj	N/A Run time (s)	10% Obj	10% Run time (s)	10% Gap	20% Obj	20% Run time (s)	20% Gap	30% Obj	30% Run time (s)	30% Gap
30	10	10.88	3.84	10.54	2.31	3.13%	10.79	4.527	0.83%	10.79	4.11	0.83%
30	15	11.05	1.89	10.94	3.09	1.00%	11.05	2.672	0.00%	11.05	2.78	0.00%
30	20	11.05	1.7	9.4	1.89	14.93%	10.94	2.36	1.00%	11.05	2.33	0.00%
40	10	13.83	40.06	13.08	64.61	5.42%	13.72	73.164	0.80%	13.83	51.77	0.00%
40	15	15.48	209.61	15.17	57.69	2.00%	15.48	258.46	0.00%	15.48	45.46	0.00%
40	20	15.62	6.19	12.39	3.55	20.68%	14.95	5.92	4.29%	15.62	7.91	0.00%
50	10	17.28	282.83	17.04	193.21	1.39%	17.15	457.417	0.75%	17.28	255.84	0.00%
50	15	19.27	467.65	19.27	175.97	0.00%	19.27	76.313	0.00%	19.27	294.61	0.00%
50	20	19.41	8.95	19.1	71.26	1.60%	19.41	22.642	0.00%	19.41	12.04	0.00%
60	10	21.53	1,103.65	**21.19**	**3,603.70**	**1.58%**	21.44	633.575	0.42%	21.53	1,135.94	0.00%
60	15	23.77	780.4	23.77	1,995.03	0.00%	23.77	288.762	0.00%	23.77	343.11	0.00%
60	20	23.91	24.08	23.91	61.91	0.00%	23.91	61.204	0.00%	23.91	52.68	0.00%
70	10	25.1	1,676.78	**24.87**	**3,604.11**	**0.92%**	25.1	1,074.597	0.00%	25.1	1,719.79	0.00%
70	15	**28.67**	**3,604.74**	28.67	1,519.65	0.00%	28.67	1,556.089	0.00%	28.67	692.18	0.00%
70	20	29.02	71.92	28.88	739.46	0.48%	29.02	152.77	0.00%	29.02	52.69	0.00%
80	10	26.62	3,217.49	**26.42**	**3,604.56**	**0.75%**	26.62	2,154.86	0.00%	26.62	2,437.79	0.00%
80	15	**31.51**	**3,608.47**	**31.17**	**3,607.99**	**1.08%**	**31.08**	**3,611.27**	**1.36%**	**31.51**	**3,607.91**	**0.00%**
80	20	32.04	151.14	**31.56**	**3,609.86**	**1.50%**	32.04	507.36	0.00%	32.04	242.63	0.00%
90	10	**28.2**	**3,605.99**	**27.8**	**3,605.29**	1.42%	**28.13**	**3,605.33**	**0.25%**	**28.2**	**3,605.4**	**0.00%**
90	15	**34.02**	**3,610.98**	**32.11**	**3,609.85**	5.61%	**33.77**	**3,609.76**	**0.73%**	**33.78**	**3,609.83**	**0.71%**
90	20	35.39	2,152.36	**32.39**	**3,612.36**	8.48%	**34.42**	**3,612.29**	**2.74%**	**35.46**	**3,612.01**	**-0.20%**
100	10	**29.83**	**3,612.33**	**30.44**	**3,606.10**	**-2.04%**	**30.62**	**3,606.20**	**-2.65%**	**30.56**	**3,606.01**	**-2.45%**
100	15	**36.38**	**3,613.04**	**32.5**	**3,611.72**	**10.67%**	**36.08**	**3,611.53**	**0.82%**	**36.63**	**3,611.63**	**-0.69%**
100	20	**38.19**	**3,615.08**	**29.41**	**3,614.87**	**22.99%**	**37.05**	**3,614.60**	**2.99%**	**33.52**	**3,614.64**	**12.23%**
Average		24.09	1,477.97	23.00	1,857.50	4.32%	23.94	1,358.49	0.60%	23.92	1,359.21	0.43%

the great deluge algorithm is better on average. This is due to significantly shorter run times of the great deluge algorithm which leads to the introduction of more columns to the restricted master problem. We observed that, in real size instances, the complexity of the problem under the same number of tasks reduces as the number of available workers grows. Furthermore, we observed a relative high variability in worker utilization across workers assigned to tasks. The results after enforcing workload balancing constraints, indicate that as long as the cap on maximum difference between each pair of workers' utilization is not too tight, enforcing workload balancing constraints does not significantly affect the level of overall utilization for all workers. Also, in some cases such constraint may even improve computational tractability. Regarding the effect of execution time windows, one would expect that an increase in time window length would result in a more productive schedule. However, in applications such as the airline industry, the TAP-FET-SDT has to solved frequently, i.e., every time a change in flight schedule occurs. Therefore, a new task assignment cannot take more than a few minutes. Under such circumstances, the potential benefit from having more flexible tasks is undermined by the growth in complexity for large instances. As a result, further effort should go into explicitly incorporating uncertainty into the formulation of TAP-FET-SDT. Early arrivals and delayed arrivals/departures are common events that increase the rate by which TAP-FET-SDT has to be solved. Having a robust task assignment requires less frequent solving the problem providing the decision maker with more time to improve assignments and benefit more from the flexibility in executing tasks. As another direction for future research, one can explore the effect of the reward system used for the objective function with respect to computational performance and the solutions obtained. In this study, we assumed satisfaction of skill set requirements as a hard constraint on achieving the reward associated with the task. However, other reward systems such as allowing for under-covering task requirements may be worth exploring. Also, examining the effect of different reward value distributions (besides uniform) can shed more light on the behavior of TAP-FET-SDT.

3.7 References

S. M. Al-Yakoob and H. D. Sherali. Mixed-integer programming models for an employee scheduling problem with multiple shifts and work locations. *Annals of Operations Research*, 155(1):119–142, 2007.

J. F. Bard and L. Wan. The task assignment problem for unrestricted movement between workstation groups. *Journal of Scheduling*, 9(4):315–341, 2006.

G. Dueck. New optimization heuristics: The great deluge algorithm and the record-to-record travel. *Journal of Computational Physics*, 104(1):86–92, 1993.

A. T. Ernst, H. Jiang, M. Krishnamoorthy, and D. Sier. Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, 153(1):3–27, 2004.

P. Eveborn, M. Rönnqvist, H. Einarsdóttir, M. Eklund, K. Lidén, and M. Almroth. Operations research improves quality and efficiency in home care. *Interfaces*, 39(1):18–34, 2009.

Y.-H. Kuo, J. M. Leung, and C. A. Yano. Scheduling of multi-skilled staff across multiple locations. *Production and Operations Management*, 23(4):626–644, 2014.

Q. Lequy, M. Bouchard, G. Desaulniers, F. Soumis, and B. Tachefine. Assigning multiple activities to work shifts. *Journal of Scheduling*, 15(2):239–251, 2012a.

Q. Lequy, G. Desaulniers, and M. M. Solomon. A two-stage heuristic for multi-activity and task assignment to work shifts. *Computers & Industrial Engineering*, 63(4):831–841, 2012b.

Q. Lequy, G. Desaulniers, and M. M. Solomon. Assigning team tasks and multiple activities to fixed work shifts. *INFOR: Information Systems and Operational Research*, 51(2):64–75, 2013.

J. Van den Bergh, J. Beliën, P. De Bruecker, E. Demeulemeester, and L. De Boeck. Personnel scheduling: A literature review. *European Journal of Operational Research*, 226(3):367–385, 2013.

3.8 Appendix

Table A.1 shows the objective function values and run times for instances of TAP-FET-SDT solved using CPLEX, BP-BB, and BP-GD. Instances with bold font where terminated due to one hour time limit on CPLEX.

Table A.1: Objective function value and run time of three solution methods on instances of TAP-FET-SDT.

Tasks	Workers	CPLEX- Best Integer Solution	CPLEX Best Bound	CPLEX Run time(s)	BP_BB Obj. Func. Val.	BP_BB Run time(s)	BP_GD - Obj. Func. Val.	BP_GD Run time(s)	CPLEX Gap	BP_BB Gap	BP_GD Gap
30	10	10.88	10.94	3.84	10.88	1.52	10.68	1.67	0.52%	0.52%	2.35%
30	15	11.05	11.06	1.89	11.05	0.69	11.05	0.49	0.05%	0.05%	0.05%
30	20	11.05	11.06	1.7	11.05	0.53	11.05	0.11	0.05%	0.05%	0.05%
30	25	11.05	11.06	1.76	11.05	0.63	11.05	0.13	0.05%	0.05%	0.05%
30	30	11.05	11.06	1.75	11.05	0.59	11.05	0.2	0.05%	0.05%	0.05%
30	35	11.05	11.06	1.8	11.05	0.67	11.05	0.19	0.05%	0.05%	0.05%
30	40	11.05	11.06	1.83	11.05	0.58	11.05	0.14	0.05%	0.05%	0.05%
40	10	13.83	13.89	40.06	13.81	121.27	13.61	123.25	0.44%	0.58%	2.02%
40	15	15.48	15.52	209.61	15.45	2.56	15.43	1.53	0.26%	0.45%	0.58%
40	20	15.62	15.62	6.19	15.6	2.52	15.48	1.33	0.00%	0.13%	0.90%
40	25	15.62	15.62	6.25	15.62	1.92	15.6	0.78	0.00%	0.00%	0.13%
40	30	15.62	15.62	6.22	15.62	1.8	15.62	0.7	0.00%	0.00%	0.00%
40	35	15.62	15.62	6.34	15.48	1.42	15.51	0.56	0.00%	0.90%	0.70%
40	40	15.62	15.62	6.3	15.62	1.67	15.62	1.47	0.00%	0.00%	0.00%
50	10	17.28	17.37	282.83	16.58	124.08	17.24	127.37	0.50%	4.53%	0.73%
50	15	19.27	19.41	467.65	19.23	21.57	19.27	45.32	0.72%	0.93%	0.72%
50	20	19.41	19.41	8.95	19.41	3.52	19.41	1.33	0.02%	0.02%	0.02%
50	25	19.41	19.41	25.21	19.41	2.94	19.41	1.83	0.02%	0.02%	0.02%
50	30	19.41	19.41	25.39	19.41	3.03	19.39	1.66	0.02%	0.02%	0.12%
50	35	19.41	19.41	25.48	19.41	3.16	19.37	2.48	0.02%	0.02%	0.22%
50	40	19.41	19.41	25.14	19.41	3.47	19.41	1.13	0.02%	0.02%	0.02%
60	10	21.53	21.63	1103.65	19.77	123.91	20.85	158.49	0.45%	8.59%	3.59%
60	15	23.77	23.78	780.4	23.72	30.47	23.72	6.72	0.04%	0.25%	0.25%
60	20	23.91	23.91	24.08	23.89	6.53	23.89	4.97	0.01%	0.10%	0.10%
60	25	23.91	23.91	25.39	23.73	4.86	23.91	2.19	0.01%	0.76%	0.01%
60	30	23.91	23.91	25.91	23.77	4.56	23.71	5.35	0.01%	0.59%	0.84%
60	35	23.91	23.91	18.21	23.91	5.19	23.91	1.45	0.01%	0.01%	0.01%
60	40	23.91	23.91	18.66	23.91	5.44	23.91	0.95	0.01%	0.01%	0.01%
70	10	25.1	25.17	1676.78	22.54	124.11	24.41	180.21	0.28%	10.45%	3.02%
70	**15**	**28.67**	**28.81**	**3,604.74**	**28.44**	**180.33**	**28.45**	**180.26**	**0.49%**	**1.28%**	**1.25%**
70	20	29.02	29.02	71.92	29.02	128.9	29.02	125.62	0.00%	0.00%	0.00%
70	25	29.02	29.02	44.78	29.02	123.6	29.02	122.98	0.00%	0.00%	0.00%
70	30	29.02	29.02	44.67	29.02	123.3	29.02	121.56	0.00%	0.00%	0.00%
70	35	29.02	29.02	44.28	29.02	122.3	29.02	123.58	0.00%	0.00%	0.00%
70	40	29.02	29.02	45.11	29.02	122.45	29.02	125.55	0.00%	0.00%	0.00%
80	10	26.62	26.66	3217.49	23.8	124.04	25.22	180.2	0.15%	10.73%	5.40%
80	**15**	**31.51**	**31.52**	**3,608.47**	**30.1**	**180.33**	**31.42**	**180.26**	**0.03%**	**4.51%**	**0.32%**
80	20	32.04	32.04	151.14	32.04	128.21	32.04	150.24	0.00%	0.00%	0.00%
80	25	32.04	32.04	343.93	32.04	125.23	32.04	125.88	0.01%	0.01%	0.01%
80	30	32.04	32.04	344.17	32.04	126.69	32.04	124.51	0.01%	0.01%	0.01%
80	35	32.04	32.04	343.49	32.04	125.85	32.04	122.99	0.01%	0.01%	0.01%
80	40	32.04	32.04	343.8	32.04	130.28	32	129.45	0.01%	0.01%	0.13%
90	**10**	**28.2**	**28.68**	**3,605.99**	**25.09**	**123.65**	**26.87**	**180.13**	**1.69%**	**12.53%**	**6.32%**
90	**15**	**34.02**	**34.94**	**3,610.98**	**32.96**	**146.89**	**33.97**	**180.13**	**2.64%**	**5.67%**	**2.78%**
90	20	35.39	36.04	2152.36	35.43	180.29	35.81	180.26	1.79%	1.68%	0.63%
90	25	36.12	36.12	1179.57	36.12	146.16	36.05	180.25	0.00%	0.00%	0.19%
90	30	36.12	36.12	1180.42	36.12	153.74	36.12	132.22	0.00%	0.00%	0.00%
90	35	36.12	36.12	1182.14	36.12	136.17	36.1	180.24	0.00%	0.00%	0.06%
90	40	36.12	36.12	1184.51	36.12	129.58	36.12	150.29	0.00%	0.00%	0.00%
100	**10**	**29.83**	**31.78**	**3,612.33**	**26.79**	**124.74**	**28.26**	**180.09**	**6.15%**	**15.71%**	**11.09%**
100	**15**	**36.38**	**38.54**	**3,613.04**	**35.39**	**126.16**	**37.32**	**180.2**	**5.61%**	**8.18%**	**3.18%**
100	**20**	**38.19**	**39.96**	**3,615.08**	**37.87**	**170.2**	**38.94**	**180.23**	**4.43%**	**5.23%**	**2.55%**
100	**25**	**40.06**	**40.1**	**3,619.13**	**39.37**	**150.06**	**39.9**	**180.27**	**0.10%**	**1.82%**	**0.50%**
100	**30**	**40.06**	**40.1**	**3,619.12**	**39.7**	**147**	**39.85**	**180.29**	**0.10%**	**1.00%**	**0.62%**
100	**35**	**40.06**	**40.1**	**3,619.26**	**39.3**	**133.71**	**39.81**	**180.31**	**0.10%**	**2.00%**	**0.72%**
100	**40**	**40.06**	**40.1**	**3,618.95**	**39.29**	**156.81**	**39.91**	**180.32**	**0.10%**	**2.02%**	**0.47%**

Chapter 4: Reliable Integrated Planning Framework: An Application in Airline Operations[3]

Abstract

As organizations expand their business, their operations become more complex. Without fully grasping the system, an optimal decision within a limited scope can have drastic unanticipated effects at multiple planning levels. Making good decisions becomes even more challenging in highly non-deterministic environments, such as those experienced by the airline industry. Therefore, integrating different planning problems at different levels in one single model would be ideal. However, full integration of strategic, tactical, and operational decisions in a mathematical model is not practical due to limitations in computational power to handle the complexity of such large-scale model. In this study we propose a reliable integrated planning framework which considers interactions between decisions made at different planning levels (strategic, tactical, and operational) in an uncertain environment. We apply this modeling approach to a simplified airlines operations framework and compare the performance achieved by a classical planning approach to that of the reliable integrated planning framework. A numerical study is completed and a discussion about the advantages and challenges of using this modeling approach when making complex inter-related decisions is presented.

Keywords

Integrated decision making; uncertainty; failure scenarios; simulation; airline operations

4.1 Introduction and Background

Planning decisions faced by organizations are commonly classified in three levels: strategic, tactical, and operational. A planning decision falls within one of these categories based on the length of its planning horizon and the frequency in which it will be revisited. Strategic decisions apply over long term planning horizons and affect decisions made at other levels. The long term nature of strategic decisions comes with the highest degree of uncertainty affecting decision making (Parnell et al., 2013). At an intermediate period range, tactical

[3]To be submitted for review to *Transportation Research Part E: Logistics and Transportation Review*

decisions provide a basis to link strategic decisions with daily operations. Planning for resources and activities needed to deliver tangible value to users or customers is the main characteristic of tactical planning decisions which apply from a few weeks to a few months. At the lowest level, operational planning decisions are made more frequently and are geared towards executing and delivering services or products on a day-to-day basis.

An analytical approach to decision making is very common among practitioners and researchers who address decision problems that affect different planning levels. The analytical approach is characterized by decomposing (i.e., breaking down or reducing) systems and considering smaller problems individually. The outcomes of decisions made at the top planning level stream down to the lower levels and form the basis to consider tactical and operational decisions. This approach seeks to model and solve each problem with a high degree of accuracy with respect to the real world. However, as Chan et al. (2002) argue, treating each problem individually does not necessarily results in the best overall performance as the interactions between different planning levels tend to get overlooked. This criticism becomes even more relevant where meaningful feedback can be extracted from lower or higher planning levels to improve the decisions. The suboptimal decision made under purely analytical approaches reflects a major disadvantage.

A proper analysis must acknowledge the dependencies and balance trade-offs across all three planning levels (Parnell et al., 2013). A group of operations research (OR) professionals in industry and academia aim at integrating multiple levels of decisions into a single comprehensive model to fully capture the interactions between different planning levels. Examples of such efforts are common in the integrated supply chain and logistics literature. A review of different stages of supply chain design done by Erengüç et al. (1999) highlights the substantial benefits that can be achieved by integrating decisions at different levels of a supply chain. Goetschalckx et al. (2002) present the same argument and report case studies to justify integration of strategic and tactical decisions over a hierarchical approach to designing systems. A long list of studies that have integrated multiple levels of decisions into one single model is provided by Goetschalckx et al. (2002). Another review of supply chain planning literature that integrates facility location problems (as the strategic decision) with other decisions common to supply chain design was completed by Melo et al. (2009). The authors justify the necessity of integrating all decisions into one model by arguing that a step-by-step approach to design of a supply chain cannot capture the optimal setting. However, some related studies emphasize the computational limitation of fully integrating strategic, tactical, and operational planning decisions. For example, Talluri and Baker (2002) propose a three stage optimization process that starts with the evaluation of suppliers, manufacturers, and distributors in the first stage, facility location decisions at the second stage, and transshipment and deployment decisions in the third stage. The results of each stage affect lower level decisions. In this way, they maintain some level of integration in their approach while addressing the exponential growth in complexity of a completely integrated approach. Somewhat similar to Talluri and Baker (2002), Sabri and Beamon (2000) addressed the integration of decisions at different planning levels by taking a step by step approach also. However, the inter-level interactions are not limited to cascading down the results of higher level decisions to the lower levels, but also the decisions made at the lower levels turn into a feedback to alter the higher level decisions, if need be. We were not able to find many other studies that take a similar approach to that of Sabri and Beamon (2000). A recent study by Zheng et al. (2019) also focuses on the integration of multiple decisions within the context of supply chain management and exploited real-world limitations as effective cuts that would improve their model's tractability.

Based on the reviewed literature, decision makers are normally faced with the dilemma of choosing between (i) decomposing the problems into smaller components at the cost of neglecting their interactions or (ii) developing a comprehensive model with extremely large number of variables and constraints that easily becomes intractable.

From a different perspective, businesses grow rapidly and continuously extend their offering of products and/or services. In this environment, businesses become more susceptible to natural uncertainty and unanticipated events. For instance, forecasting the number and length of calls arriving to a call center or the weekly demand for a product can be a very cumbersome effort. The same argument is relevant to predicting machine failures, adverse weather around an airport causing flight cancellations/delays, deviations between planned and actual available workforce, prices of raw material and/or final products, etc. (Van den Bergh et al., 2013; Janak et al., 2007). Although uncertainty affects a large range of decisions, many studies follow a deterministic approach. Taking proactive measures to address deviations from status quo is more cost effective as opposed to taking reactive and hasty corrections. However, estimating a probability distribution that would fit the uncertainty in parameters needed to make those proactive decision is not a trivial task for many decision makers (Kouvelis and Yu, 2013). In addition, the volume of historical data in newly established businesses is very small for fitting a reliable probability distribution. In some cases where uncertainty is significant, such as natural disasters, estimating an accurate probability distribution is very difficult due to highly complex mechanisms underlying the events. Even if a dependable estimation of probabilities is at disposal, the optimization model that captures all possible scenarios becomes extremely large in practice. Finding the best answer to such problems can become very time-consuming or even impossible. This leaves the decision maker with no choice but to make a suboptimal decision diminishing the effectiveness of stochastic optimization approaches for dealing with these uncertain problems. This further hinders developing and solving integrated planning models by adding more complexity to an already difficult model.

Accordingly, accounting for the dependencies between decisions as well as uncertainty is a better pathway to improve overall performance. One can achieve the former by developing a model that comprises all of the decisions together. However, such model is difficult to develop and even more difficult to solve even in the smallest cases in practice. Taking a hierarchical approach while maintaining bottom-to-top as well as top-to-bottom feedback can overcome the challenge. Another aspect is to proactively deal with uncertainty by explicitly incorporating it in the decision making process. When dealing with a decision making problem, sources of uncertainty can be either *internal* or *external* with respect to the decision making scope. For example, when making operational decisions at an airport, the level of uncertainty associated with the ability to carry out operations as planned can be affected by how the airline has setup its network and flight schedules. In other words, the airline's network reliability will directly affect the level of unanticipated shifts from plans at each airport. If the decision maker follows an integrated approach, this can be considered as *internal* uncertainty. On the other hand, events such as equipment breakdowns or severe weather can be considered as somewhat *external* sources of uncertainty, the effect of which is out of the scope of decisions made at other levels. As mentioned earlier, the lack of information about the likelihood of unanticipated events undermines the idea of making better decisions by considering uncertainty. However, one can alleviate this shortcoming by following an integrated approach to decision making while incorporating uncertainty at each level. Thus, we can keep track of the effect of *internal* uncertainty that cascades from higher level decisions to the lower levels. For instance, in the context of an airline, the likelihood of service interruptions at an airport caused by gate shortage can be altered by shifting the airline's network and the timing of flight arrivals and departures - the outcome of higher level decisions.

Although the airline industry is one of the pioneers in the application of OR techniques, examining the literature, we observed that no study has focused on integrating decisions at different planning levels of airline operations. Another reason that justifies exploring the potential gains from integrating decisions at different planning levels could be that the airline industry is inherently characterized as highly unpredictable. Multiple sources of uncertainty such as demand fluctuations, competitors shift in operations and policies, distribution of operations across numerous parts of the world with different weather conditions exposes the airline companies to significant deviations from status quo. Although airlines have made a solid effort of collecting data, they can still gain benefits from tracking sources of uncertainty and capturing the mechanisms through which their effects are rippled down from strategic to tactical and operational decisions.

The goal of this study is to address the aforementioned challenges by proposing a framework to merge strategic, tactical, and operational decisions into a comprehensive model that accounts for both the downstream and the upstream interactions between decisions at different levels as well as uncertainty. The proposed framework is examined in the context of the airline industry. The rest of the paper is organized as follows. The Reliable Integrated Planning Framework is explained in Section 4.2. Section 4.3 presents details of the implementation of proposed framework in a simplified use case based in airline operations. A numerical study of the case in airline operations is presented in Section 4.4. Section 4.5 provides a discussion on the main findings and suggests directions for future research.

4.2 Reliable Integrated Planning Framework

The Reliable Integrated Planning Framework (RIPF) is a modeling approach that aims at the integration of interacting decisions at different levels of decision making. RIPF not only considers the direct effect of decisions on each other but explores propagation of uncertainty across different decisions. Quantitative techniques common in OR are unable of fully integrating components of complex systems in one single model. RIPF addresses this issue by sketching potential direct interactions between decisions and then extracting potential feedback to improve/modify other decisions with the objective of overall performance improvement. As shown in Figure 4.1, the decision interactions are not limited to having the output of one decision problem become the input to another. In this framework, corrective constraints that simply are signals extracted from other decisions may be incorporated to modify solutions for interrelated decision problems. Optimal solutions obtained for decision problems considered in isolation might not necessarily point in the direction of improving the overall performance of the system. In simple terms, corrective constraints have the effect of preventing decision makers from falling into local optima. This comes at the cost of deviating from a level-specific optimal solution for the sake of overall performance improvement.

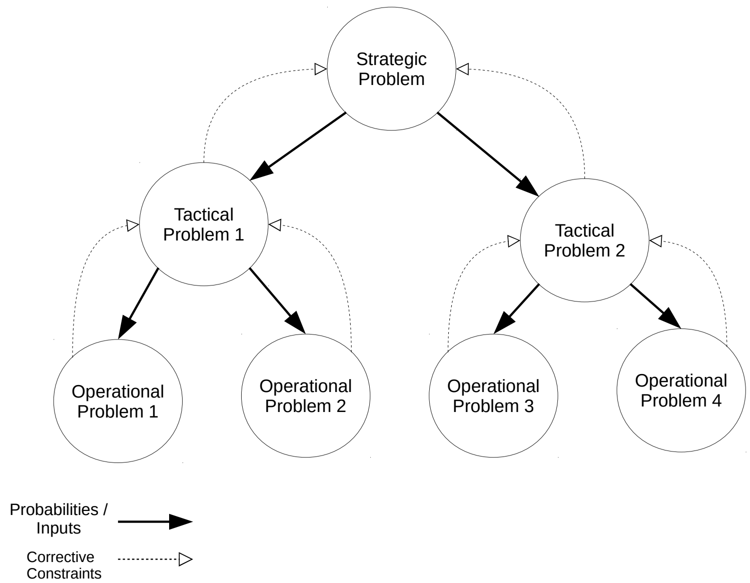


Figure 4.1: Reliable Integrated Planning Framework.

The initial step in the RIPF requires establishing an initial framework based on the classical planning approach where the output of higher level decision problems become the input for lower level decision problems. A simulation of events related to the system is used to evaluate the overall performance of the previously obtained decisions and identify negative effects that propagate through the framework. The negative effects are then addressed in terms of corrective constraints to higher or lower levels of the framework. This process is repeated until a desired overall performance for the system is achieved. Algorithm 5 provides a high-level description of the process.

Algorithm 5 Steps to develop a Reliable Integrated Planning Framework.

```
1: procedure
2:     Develop initial framework with output of higher level decisions becoming input to lower
   levels
3:     Run simulation based on the initial framework decisions
4:     Evaluate the overall performance
5:     while Overall performance does not meet the requirements do
6:         Identify sources of under-performance
7:         Introduce corrective constraints and update the framework
8:         Evaluate the overall performance
```

In the next section, a case is presented to illustrate how RIPF can be applied to complex interrelated problems and its capability for capturing hard-to-predict effects of decisions made in highly uncertain contexts.

4.3 Use Case of the Reliable Integrated Planning Framework in Airline Operations

To illustrate the application and evaluate the effectiveness of the RIPF, we consider a planning framework in the context of airline operations with decisions at multiple levels (strategic, tactical, and operational). We apply a simulation approach to evaluate the airline overall performance when decisions are made over a long period of time. Planning decisions are made once within a classical planning framework and once again within the RIPF. The performance under the two sets of decisions are then compared to evaluate the effectiveness of the RIPF. Although airline operations involve numerous highly inter-related decisions, for simplicity, we consider a classical planning framework with four levels. The highest level being a strategic decision and the lowest level involving an operational decision (Figure 4.2).

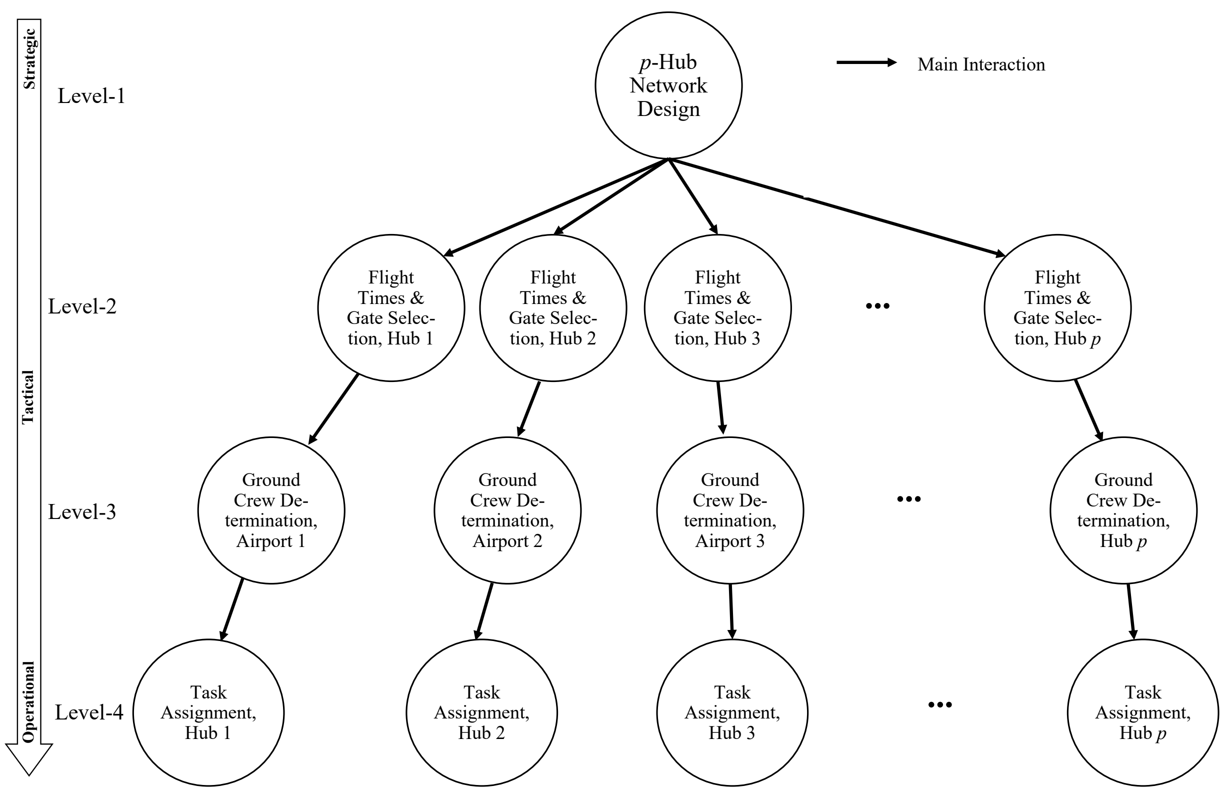


Figure 4.2: Classic planning framework in the context of airline operations.

In the first level (Level-1), we focus on designing the hub network that the airline will be operating on for the next few years. This is a strategic decision which affects and is affected by the other three sets of decisions to be made at the hub airport level. Then, selecting the number of gates to make available for an airline at each hub airport and determining the timing of flight arrivals and departures at the hub airport are the decisions made at Level-2 of the planning framework. In Level-3, determining the size of the ground crew to make available and the timing of their shifts at hub airports are tactical decisions with a planning horizon of a few months. Finally, at the daily operational level (Level-4), tasks that are required to be executed for flights to depart to their next destination are assigned to the ground crew workers available at the hub airports based on their skill levels and availability.

In this context, the potential of experiencing a hub airport shutdown/failure is a source of uncertainty introduced in the problem at Level-1. However, failures affect operations all the way down to the lowest level. As discussed earlier, besides having a mere top-down approach to inter-related decisions, failing to acknowledge uncertainty and its rippling effects is a major drawback of the classical planning approach to making inter-related decisions. RIPF intends to address these shortcomings. Accordingly, we implement a simulation in which we evaluate the overall performance of the system when decisions are made first using a classic planning framework from strategic to operational (i.e., from Level-1 down to Level-4) with the output of a higher level decision becoming the input to the next lower level. Figure 4.2 illustrates the classic planning framework. Then, the simulation is repeated to evaluate the overall system performance for the same series of decisions using the RIPF presented in Figure 4.3 where decision makers acknowledge feedback from both higher and lower levels to adjust and improve their decisions.

In the assessment of the performance of the two frameworks, we use optimization techniques to make decisions at each level. The decisions within the RIPF are made not only based on the information passed downstream from a higher level, but also taking into account corrective constraints dictated from both higher and lower levels. In contrast, in the classical planning framework, the inter-level interactions are mostly limited to a single level top-to-bottom flow of information. After decisions are made at all levels, a simulation is run to evaluate the overall system performance over a relatively long time period. The following subsections describe the modeling and solution methods used to make decisions for the planning problems at different levels in the context of airline operations.

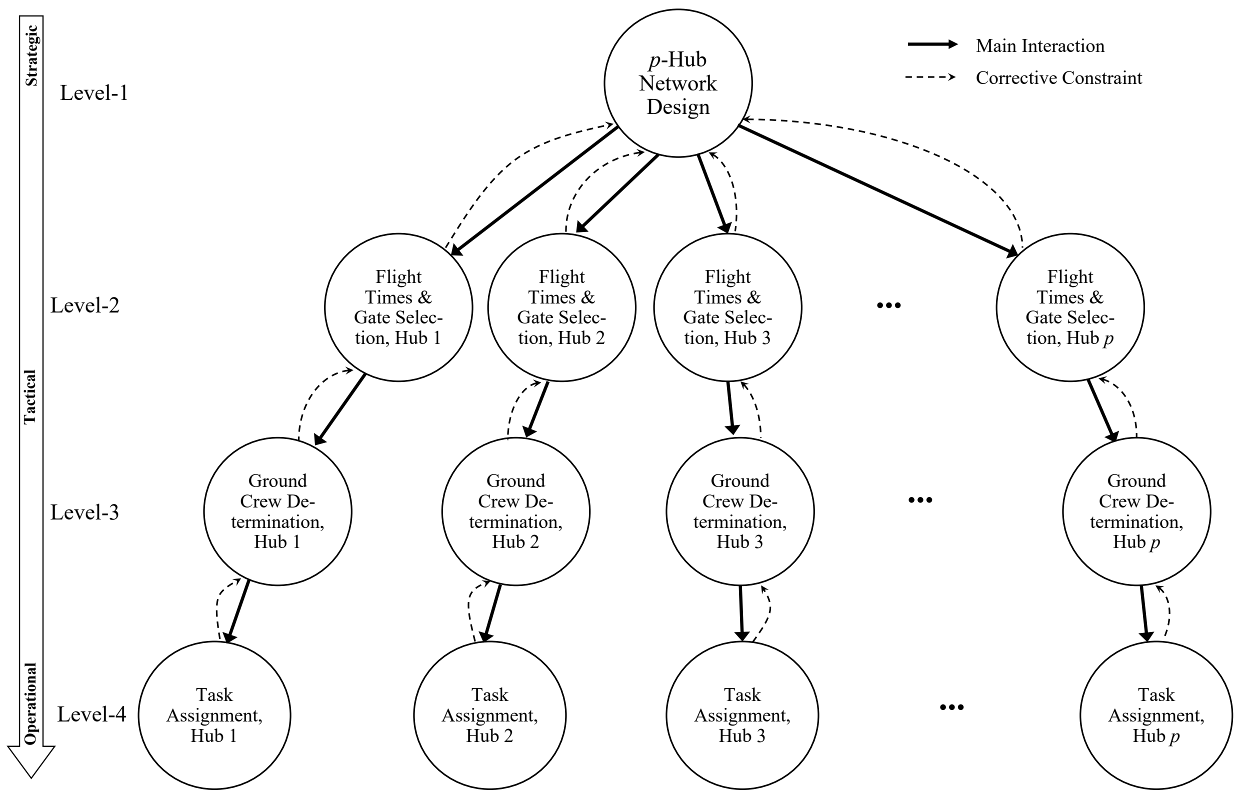


Figure 4.3: Reliable Integrated Planning Framework in the context of airline operations.

4.3.1 Level 1 - p-Hub Network Design

Airlines utilize hub networks where a subset of airports, namely hubs, operate as consolidation and distribution points to take advantage of economies of scale when transporting passengers from multiple origins to multiple destinations. Each origin-destination (OD) in the network is connected through at least one and at most two hubs. Normally, the outcome of the selecting hub airports and connections between OD pairs remain in effect for a long time. The goal is to keep the operating cost of the network as low as possible. Alumur and Kara (2008) and Farahani et al. (2013) present a thorough review of the hub network design problem.

However, the airline industry is significantly affected by unanticipated events (e.g., severe weather, security lock-downs, etc.) that are likely to result in reduced capacity or complete unavailability. Therefore, airlines seek to design a reliable hub network with the lowest expected operating cost. In this setting, the decisions are (i) selecting a specific number of airports as hubs, (ii) determining a primary connection between each OD pair to be used when no failures occur, and (iii) determining backup connections for each OD pair for possible disruption scenarios. A formulation and an exact solution approach for the Reliable p-Hub Network Design Problem with Multiple Disruptions (RpHND-MD) is presented in Chapter 2. In this problem, given a set of nodes, N, in a network, a subset of size p is to be selected as hubs. For each OD pair a primary connection and a set of backup connections are to be determined such that the total expected operating cost of the network is minimized under possible failure scenarios. Basic assumptions of the RpHND-MD are (i) direct links between non-hub nodes are not allowed, (ii) spokes may be assigned to multiple hubs, (iii) the number of hubs to be located in the network is fixed and known, (iv) hubs form a complete graph, (v) the cost associated with establishing a hub is constant over all nodes, (vi) each airport can be considered as a potential hub, (vii) the cost of transporting a passenger from airport i to j is proportional to the distance between the two airports and a discount factor is applied for inter-hub links reflecting economies of scale, (viii) hubs and links are uncapacitated, (ix) once an airport fails, it fully loses its capacity to serve the network (i.e., no partial disruptions are allowed), and (x) flows have to be shipped through at least one and at most two hubs. In addition to these basic assumptions, the RpHND-MD allows multiple disruptions following known probabilities. We assume that failure of hubs occur independent of each other. The strategic decisions made at this level will affect almost all other decisions at the hub airport level (i.e., Level-2, Level-3, and Level-4).

As discussed in Section 2.3.2, a connection (i, k, m, j), such that $i, k, m, j \in N$, connects origin i to destination j through hubs k and m, respectively. Given the structure of the problem, a feasible selection of primary (i.e., regular operation) and backup routes for a given OD pair can be presented in a binary tree. The tree depth is limited by the maximum number of simultaneous failures that are considered. This is called a "connection tree" and it is used to develop a mathematical model for the problem. Each vertex of the tree is mapped to a route. The root node maps to the primary route used when no failure has occurred. The left child of the root node is mapped to the backup route that is to be used when the first hub in the primary route fails. Similarly, the right child points to a backup route which is utilized when the second hub in the primary route fails. To ensure connection tree feasibility, several rules have to be satisfied. For a detailed explanation of the rules and examples of connection trees, see Section 2.3.2. Based on this concept, we developed a nonlinear integer programming model for RpHND-MD that requires four sets of decision variables. z_{ij}^v takes a value of 1 if vertex v in the connection tree associated with origin destination ij is mapped to a connection, and 0 otherwise. x_{ikmj}^v is set to 1 if vertex v of the connection tree ij is mapped to the connections that connect i to j through hubs k and m, respectively. Each edge of the tree may point to a failed hub. Accordingly, e_{ij}^v equals to 1 if the edge with vertex v as its tail in connection tree ij is mapped to hub k. Finally, y_i takes the value of 1 if a hub is established in node i of the network. The model involves a few sets of auxiliary variables as well. The full notation for the mathematical formulation $\mathcal{P}1$ can be found in Appendix A.

$$\mathcal{P}1: \ Min \sum_{i \in N} \sum_{\substack{j \in N \\ j>i}} \sum_{v \in V_{ij}} B_{ij} \left(\prod_{u \in V_{ij}^v} \sum_k q_k \ e_{ijk}^u \right) . \left(\sum_{k \in N} \sum_{m \in N} (1 - q_{km}) \ C_{ikmj} \ x_{ikmj}^v \right) \tag{4.1}$$

$$S.T. \ z_{ij}^0 = 1 \qquad \forall i, j > i \in N \tag{4.2}$$

$$z_{ij}^v \leq \sum_{k \in N} \sum_{m \in N} x_{ikmj}^v \qquad \forall i, j > i \in N, v \in V_{ij} \tag{4.3}$$

$$z_{ij}^v = (1 - \kappa_{ij}^{v'}) \ z_{ij}^{v'} \qquad \forall i, j > i \in N, v_{odd} \in V_{ij} \tag{4.4}$$

$$z_{ij}^v = (1 - \mu_{ij}^{v'}) \ z_{ij}^{v'} \qquad \forall i, j > i \in N, v_{even} \in V_{ij} \tag{4.5}$$

$$\sum_{u \in V_{ij}^v} e_{ijk}^u \leq (1 - \frac{\sum_m x_{ikmj} + \sum_{m \in N} x_{imkj}}{2}) \qquad \forall i, j > i, k \in N, v \in V_{ij} \tag{4.6}$$

$$e_{ijk}^v = \sum_{m \in N} x_{ikmj}^{v'} \qquad \forall i, j > i, k \in N, v_{odd} \in V_{ij} \tag{4.7}$$

$$e_{ijk}^v = \sum_{k \in K} x_{ikmj}^{v'} \qquad \forall i, j > i, m \in N, v_{even} \in V_{ij} \tag{4.8}$$

$$\sum_{m \in M} x_{ikmj}^v \leq y_k \qquad \forall i, j > i, k \in N, v \in V_{ij} \tag{4.9}$$

$$\sum_{k \in K} x_{ikmj}^v \leq y_m \qquad \forall i, j > i, m \in N, v \in V_{ij} \tag{4.10}$$

$$\sum_{i \in N} y_i = p \tag{4.11}$$

$$\kappa_{ij}^v = \sum_{m \in N} x_{iimj}^v + x_{ijjj}^v \qquad \forall i, j > i \in N, v \in V_{ij} \tag{4.12}$$

$$\mu_{ij}^v = \sum_{k \in N} x_{ikjj}^v + x_{iiij}^v \qquad \forall i, j > i \in N, v \in V_{ij} \tag{4.13}$$

$$z_{ij}^v, e_{ij}^v, x_{ikmj}^v, y_i, \kappa_{ij}^v, \mu_{ij}^v \in \{1, 0\} \qquad \forall i, j > i, k, m \in N, v \in V_{ij} \tag{4.14}$$

Objective function (4.1) minimizes the total expected operating cost calculated as the sum of expected cost of connections selected for nodes of the routing trees. Constraints (4.2) enforce selection of a primary connection for each OD. Constraints (4.3) link x variables to z so that a vertex is either mapped to one connection or none. Constraints (4.4) enforce that vertex v has to be mapped to a connection if and only if its parent vertex, v' is mapped to a connection for which its first hub is neither the origin nor the destination. Similarly, Constraints (4.5) enforce mapping of a connection to vertex v if and only if its parent vertex v' is mapped to a connection and the second hub of the connection is neither the origin nor the destination. Constraints (4.6) ensure that hubs that are used at a vertex are not failed hubs on the path from the root node. Constraints (4.7) and (4.8) map the edges to their corresponding failed hub based on their head vertex. Constraints (4.9) and (4.10), common in hub-and-spoke models, guarantee that the connections selected by the model are limited to the ones that use open hubs. Constraints (4.11) enforce that exactly p hubs should be opened. Constraints (4.12) and (4.13) assign the right value to auxiliary variables in the formulation. Constraints (4.14) define all variables a binary.

We developed an efficient search algorithm to find the exact solution to the RpHND-MD as formulated above (Section 2.3.4). The two stage algorithm begins with fixing hub locations. Then, it takes advantage of the connection tree structure to calculate tight bounds on the solution to the problem with fixed hubs. Efficient bounds are then used to avoid unnecessary computations of non-promising hub combinations.

4.3.2 Level 2 - Determining Flight Times and Gate Selection

A portion of an airline's operating cost comes from landing fees that airports charge. Airports have a limited number of gates to accommodate inbound and outbound flights. Each airline, depending on its level of operations as well as many other business rules, is assigned a certain number of gates at an airport. At this level, an airline's goal is to reduce its operating costs by limiting the number of gates it occupies while maintaining a high level of service and safety for all arrivals and departures dictated by the network plans and flight schedules.

The main decisions considered at this level are determining the number of gates at each hub airport and establishing the timing of arrivals and departures for flights at each hub airport. The number of gates to occupy is minimized while maintaining a certain level of service. Regarding the timing of arrivals and departures, we assume no specific limitation except for the minimum turnaround between flights. It should be noted that any failure in one or more hubs in the network causes fluctuations in the number of passengers/flights connecting at active hubs. High fluctuations can lead to resource shortages at a hub airport. In the classical planning framework, the level of activity is determined by the flights going through the hub airport under regular operation. However in RIPF, potential traffic fluc-

Table 4.1: Sample flight times for a hub with 28 daily flights and 4 available gates.

Gate 1		Gate 2		Gate 3		Gate 4	
Flight #	Time	Flight #	Time	Flight #	Time	Flight #	Time
1	08:00:00 AM	9	08:15:00 AM	17	08:30:00 AM	25	08:45:00 AM
2	09:00:00 AM	10	09:15:00 AM	18	09:30:00 AM	26	09:45:00 AM
3	10:00:00 AM	11	10:15:00 AM	19	10:30:00 AM	27	10:45:00 AM
4	11:00:00 AM	12	11:15:00 AM	20	11:30:00 AM	28	11:45:00 AM
5	12:00:00 PM	13	12:15:00 PM	21	12:30:00 PM	-	-
6	01:00:00 PM	14	01:15:00 PM	22	01:30:00 PM	-	-
7	02:00:00 PM	15	02:15:00 PM	23	02:30:00 PM	-	-
8	03:00:00 PM	16	03:15:00 PM	24	03:30:00 PM	-	-

tuations affect the level of activity and consequently the number of gates that will be required.

The solution to the hub network design problem at Level-1 provides input to the decisions at Level-2. The assumption is that the average annual flow that goes through a hub is determined based on the solution to Level-1. This flow is then translated into a fixed number of flights for each hub airport. Flight scheduling is a highly complex process subject to numerous rules and regulations. However, without loss of generality, we make a set of simplistic assumptions to define the timing of arrivals and departures at a hub airport. Assuming (i) a fixed 1-hour turnaround for each flight, (ii) 8 hours of operation during the day, and (iii) no other restrictions on scheduling of flights, we can determine the timing of flight arrivals and departures and the number of gates to occupy at a hub airport. In practice, 1 hour is roughly the minimum turnaround time possible for aircraft arriving to a gate. For the duration of operations, we consider 8 hours of operations to be consistent with a full-time shift for ground crew workers. As an example, consider a hub airport that is selected in Level-1 according to the solution obtained for RpHND-MD. This airport is expected to handle 1,498,220 units of flow (i.e., passengers) annually. Given 365 days in a year and a standard capacity of 150 passengers for an aircraft, results in 27.36 ($\sim$ 28) flights per day. Note that, for simplicity, we do treat all flights the same regardless of them being an inbound or an outbound flight. Given the minimum turnaround of 1 hour and 8 hours of operation, each gate can handle up to 8 flights. Therefore, the airline needs to have at least 4 gates for service at the hub airport. To avoid having simultaneous flight arriving/departing at the hub, the flights are staggered over the available gates. Table 4.1 shows flight times for the four available gates in the case of an operation between 8 AM to 4 PM.

4.3.3 Level 3 - Ground Crew Determination

Ground crew workers at each airport are responsible for getting incoming flights ready for their next departure. Airlines need to roster their ground crew with the objective of minimizing ground crew staffing cost while avoiding delays due to crew shortages. In general, shorter turnarounds are preferable for airlines as they give them more flexibility when scheduling flights.

In our case, we assume only one type of aircraft being used by the airline in the network which requires a fixed set of tasks while on the ground. Loading and unloading bags, deplaning, boarding, and fueling are a few examples of common tasks that are needed for each flight. Each task can require multiple workers with different skills. Therefore, given a sequence of flight arrivals and departures at a hub airport for a day, we use a simple heuristic to determine the number of ground crew workers needed and their shifts at a hub airport.

According to the flight times determined at Level-2 and the set of tasks required to be done on each flight, we generate the task requirements for each day of operation. Given a set of ground crew worker templates (i.e., workers with different skill sets) and the number of tasks to be executed on a given day, we can determine the ground crew and their shifts during the day using the heuristic described in Algorithm 6. For simplicity, we determine the ground crew according to the flight sequence that goes through each hub airport on a day with no failures in the network. In other words, we take a deterministic approach to generate ground crew shifts that are repeated everyday.

4.3.4 Level 4 - Task Assignment for Ground Crew

As flights arrive to a hub airport, a set of tasks with multiple skill requirements are needed to be completed before the next departure. A few examples of those tasks are loading and unloading bags, disembarking and boarding passengers, fueling, cabin cleaning, and performing safety inspections. Workers possess different skill sets making them eligible to execute certain tasks for each flight. Changes in flight schedules are highly probable making it difficult to get all tasks done with a limited number of ground crew workers. Therefore, reactive measures such as paying a few workers overtime can be very common and also costly. Airlines aim at assigning tasks to a limited number of workers on the day of operation so that as many tasks with the highest priorities are executed by qualified workers.

The task assignment problem with flexible execution times and sequence dependent travel (TAP-FET-SDT) focuses on finding the best assignment of ground crew workers with a given set of skills to tasks that need to be executed on an aircraft during its turnaround time. Tasks

Algorithm 6 Heuristic to determine ground crew workers and their shifts.

1: **procedure** GETCREW(*workerTemplates*, *Tasks*, *skills*, *planningHorizon*)
2: Sort *skills* from most restricting to least restricting ▷ The skill possessed by the least number of worker templates is considered most restrictive
3: **for all** *period* $\in$ *planningHorizon* **do**
4: **for all** *skill* $\in$ *skills* **do**
5: Calculate the number of skill requirements for the period by adding the number of tasks that require *skill* and have to be executed in *period*
6: **for all** *skill* $\in$ *skills* **do**
7: $x \leftarrow$ maximum number of workers required over the planning horizon who have *skill*
8: Add x workers from the template that has *skill* to *GroundCrew*
9: Set the start time and end time of each newly added worker equal to the beginning and the end time of *planningHorizon*
10: Update maximum skill requirements ▷ As new workers are added, their skills can be used to satisfy skill requirements throughout the planning horizon. So, to update skill requirements after adding new workers, subtract the skills of newly added workers from total skill requirements calculated in line 5.
11: Assign tasks to workers
12: **for all** *worker* $\in$ *groundCrew* **do**
13: Remove idle times from the beginning and the end of *worker*'s shift
14: **return** *groundCrew*

are not all the same in terms of priority. Therefore, they come with a reward associated with their execution that reflects task priority. The goal is to execute as many high priority tasks as possible. A task is considered executed if all the skill requirements are satisfied (i.e., enough workers with the required skill set are assigned to the task within the allowed time window). We present an integer programming model $\mathcal{P}2$ for the TAP-FET-SDT. Three sets of decision variables are defined in $\mathcal{P}2$: l_{tb} takes a value of 1 if task t is set to be executed at period b, and 0 otherwise; m_{twb} equals to 1 if task t is assigned to worker w and is to start in time period b, and 0 otherwise; and, to control the sequence dependent travel, $h_{tt'wb}$ is defined such that it is set to 1 if both tasks t and t' are assigned to worker w such that task t starts at period b and task t' starts in any period after b, and 0 otherwise. Refer to Appendix B for the notation of sets, parameters, and decision variables used in $\mathcal{P}2$.

$$\mathcal{P}2: Max \sum_{t \in T} \sum_{b \in B_t} \alpha_t \; l_{tb} \tag{4.15}$$

$$S.T. \sum_{b \in B_t} l_{tb} \leq 1 \qquad \forall t \in T \tag{4.16}$$

$$\sum_{t \in T} m_{twb} \leq 1 \qquad \forall w \in W, \forall b \in B_w \tag{4.17}$$

$$\sum_{w \in W_t} \sum_{b \in B_{tw}} \beta_{ws} \; m_{twp} \geq \gamma_{ts} \sum_{b \in B_t} l_{tb} \qquad \forall t, \forall s \tag{4.18}$$

$$\sum_{t' \in T} \sum_{b' \in B'_{tb}} m_{t'wb'} \leq M(1 - m_{twb}) \qquad \forall w \in, \forall t \in T_w, \forall b \in B_{tw} \tag{4.19}$$

$$\sum_{\substack{b' \in B_{t'w} \\ b' > b}} b' \; m_{t'wb'} - b \; m_{twb} \geq \delta_t + \theta_{tt'} - M(1 - h_{tt'wb}) \qquad \forall w \in W, \forall t \in T_w, \forall t' \in T_w \cap T'_t, \forall b \in B_{tw} \tag{4.20}$$

$$h_{tt'wb} \geq m_{twb} + \sum_{\substack{b' \in B_{t'w} \\ b' > b}} m_{t'wb'} - 1 \qquad \forall w \in W, \forall t \in T_w, \forall t' \in T_w \cap T'_t, \forall b \in B_{tw} \tag{4.21}$$

$$2 \; h_{tt'wb} \leq m_{twb} + \sum_{\substack{b' \in B_{t'w} \\ b' > b}} m_{t'wb'} \qquad \forall w \in W, \forall t \in T_w, \forall t' \in T_w \cap T'_t, \forall b \in B_{tw} \tag{4.22}$$

$$h_{tt'wb} \leq 0 \qquad \forall w \in W, \forall t \in T_w, \forall t' \in T_w \cap T''_t, \forall b \in B_{tw} \tag{4.23}$$

$$\sum_{w \in W_t} m_{twb} \leq M \; l_{tb} \qquad \forall t \in T, \forall b \in B_t \tag{4.24}$$

$$l_{tb}, m_{twb}, h_{tt'wb} \in \{1, 0\} \qquad \forall t, t', \forall w, \forall b \in B \tag{4.25}$$

The objective function (4.15) seeks to maximize the sum of rewards gained as a result of assigning tasks to workers. Constraints (4.16) enforce that a task can be executed not more than once. Constraints (4.17) prevent workers from being assigned to multiple tasks at a time. Constraints (4.18) require that a task be considered executed only if the workers assigned to it can collectively satisfy the skill requirements. Constraints (4.19) guarantee that a worker who is assigned to start executing task t in period b is not assigned to any other task for the duration of t. Constraints (4.20) dictate that if worker w is assigned to execute task t' after t, w cannot be assigned to any task for the duration of t plus the travel time from t to t'. Constraints (4.21) and (4.22) assign the value of 1 to the auxiliary variable $h_{tt'wb}$ if worker w is assigned to start executing task t in period b and start task t' in any period after that, and 0 otherwise. Constraints (4.23) prevent the assignation of overlapping tasks to the same worker. Constraints (4.24) require that a task can be assigned to workers if and only if it is selected to be executed. Constraints (4.24) are also synchronizing the assignment of workers to a given task (i.e., workers assigned to the same task have to start executing the task at the same time period). Constraints (4.25) are the variable type constraints. Section 3.4 discusses a branch-and-price algorithm for solving TAP-FET-SDT with a random search heuristic to solve the pricing problem. We use the same solution method to make operational decisions at Level-4.

The timing of flight arrivals and departures generated in Level-2 and the available ground crew determined in Level-3 provide the input to make decisions related to ground crew task assignment in Level-4 and to evaluate the overall system performance on each day of operation.

4.3.5 Corrective Constraints for the Reliable Integrated Planning Framework

We consider three sets of *corrective constraints* in the implementation of the RIPF. Figure 4.4 shows the interactions between decision planning levels that are affected by the corrective constraints. The corrective constraints are:

- *Corrective Constraint 1*: passes along to Level-2 the information available on failure probabilities of hub airports at Level-1. The probabilities along with the backup connections selected in Level-1 can provide information on traffic fluctuations at each hub airport. In this way, a decision maker can establish the number of gates to make available not only based on the flow that goes through hub airports under regular operation, but also based on fluctuations in the number of rerouted flights that will visit the hub. For the numerical study presented in Section 4.4, this constraint ensures that the airline can handle up to a 20% increase in traffic at active hubs when other hubs are disrupted.

- *Corrective Constraint 2*: establishes an upstream feedback from Level-2 to Level-1. This constraint enforces a redistribution of rerouted flows from disrupted hub airports so that the not too much traffic is directed through an active hub. This would result in a reduction of gate shortages at active hubs. An integer programming model was developed (see Section 4.3.5.1) which can easily be solved using a commercial solver. This model modifies the backup setting originally determined in Level-1 to limit the number of flights going through active hubs that are highly prone to experience resource shortages. Decisions at Level-1 are then modified to minimize the increase in the network's total expected cost.

- *Corrective Constraint 3*: increases in flow traffic at an active hub due to failures experienced by other hubs could lead to a large number of additional flights and consequently tasks causing a ground crew shortage. Corrective Constraint 3 enforces redistribution of rerouted flights so that the number of additional flights going through active hubs experiencing resource shortages (i.e., gate shortages or ground crew shortages) is reduced. This is achieved by taking a portion of the flow that goes through the active hubs with resource shortages and rerouting them through other active hubs. Same to the previous corrective constraint, this modification is done with minimum increase in the network cost using the model in Section 4.3.5.1.

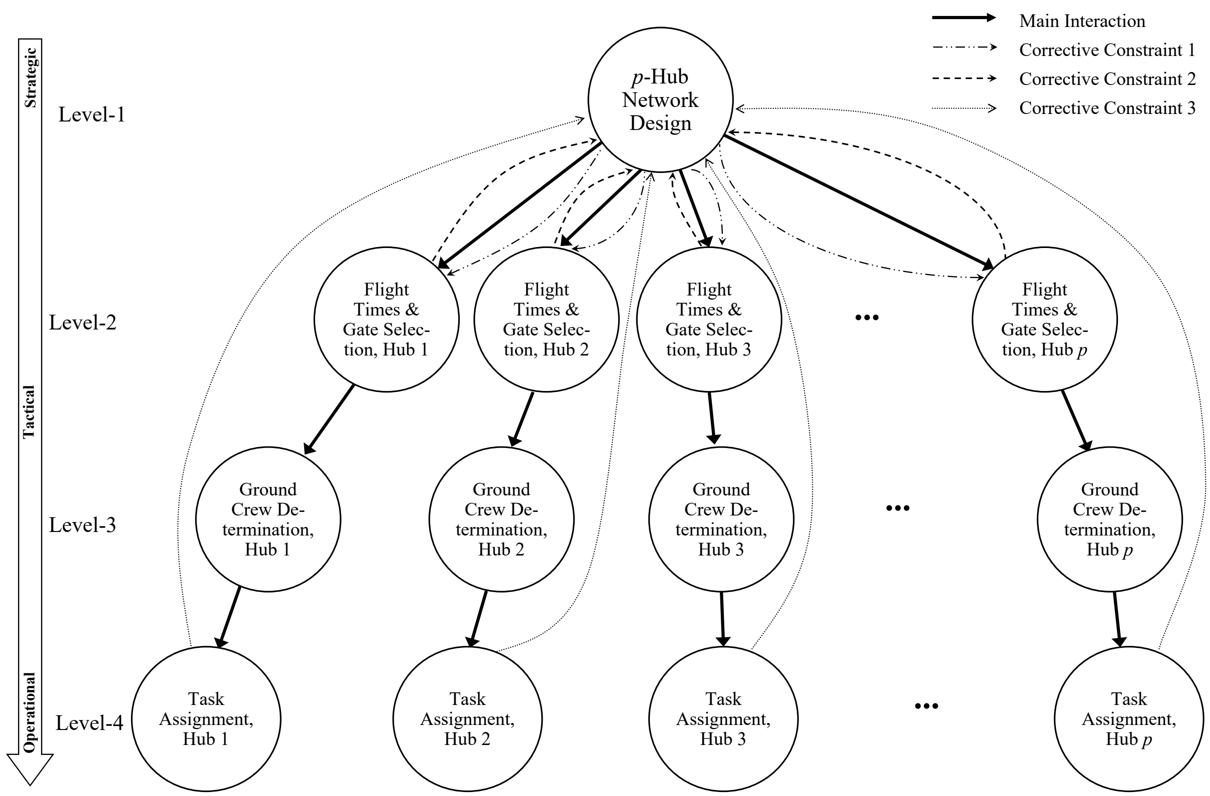


Figure 4.4: A Reliable Integrated Planning Framework in the context of airline operations with three sets of corrective constraints.

4.3.5.1 Mathematical Model to Modify RpHND-MD

To avoid resource shortages (i.e., gate shortages or ground crew shortages) at an active hub node under a given failure scenario, some of the anticipated additional flow that goes through the hub with resource shortages should be rerouted through other active hubs with available resources. The following integer programming model $\mathcal{P}3$ efficiently reroutes the affected flow. The sets, parameters, and decision variables for the model are presented below and are complemented by the notation for the RpHND-MD presented in Appendix A.

Sets:

- N: nodes in the network,
- H: hubs,
- R_{od}: set of routes that can replace the current route in the routing tree for origin o to destination d,

Parameters:

- c_r: cost of directing a unit of flow through route r,
- f_{od}: flow associated with origin destination od,
- ω_h: maximum flow that can be handled on hub h,
- ρ_{rh}: 1 if route r goes through hub h, 0 otherwise,
- ψ: flow that has to be reduced from the hub experiencing resource shortages,

Decision Variables:

- n_{odr}: 1 if route r replaces current route under failure scenario, 0 otherwise.

$$\mathcal{P}3: \quad Min. \sum_{o \in N} \sum_{\substack{d \in N \\ o \neq d}} \sum_{r \in R_{od}} c_r \ f_{od} \ n_{odr} \tag{4.26}$$

$$S.T.: \sum_{r \in R_{od}} n_{odr} \leq 1 \qquad \forall o, d \in N, o \neq d \tag{4.27}$$

$$\sum_{o \in N} \sum_{\substack{d \in N \\ o \neq d}} \sum_{r \in R_{od}} \rho_{rh} \ . \ f_{od} \ . \ n_{odr} \leq \omega_h \qquad \forall h \in H \tag{4.28}$$

$$\sum_{o \in N} \sum_{\substack{d \in N \\ o \neq d}} \sum_{r \in R_{od}} \rho_{rh} \ . \ f_{od} \ . \ n_{odr} \geq \psi \tag{4.29}$$

$$n_{odr} \in \{1, 0\} \qquad \forall o, d \in N, r \in R_{od} \tag{4.30}$$

Objective function (4.26) minimizes the cost increase for the RpHND-MD by rerouting flows from an active hub experiencing resource shortages through other active hubs that are the closest. Constraints (4.27) require that the flow associated with an OD pair be rerouted using only one other route and no more. Constraints 4.28 limit the total flow that is being rerouted through active hubs with no resource shortages. This constraint ensures that rerouting flows from a hub will not lead to shortages at another. Constraint (4.29) enforces that the sum of all flows that are going to be directed through other routes must exceed a minimum that is ψ. Constraints (4.30) are variable type constraints.

4.3.6 Simulation Setup and Performance Measurement

A comparison of the performance of the classical planning framework and the RIPF is completed by simulating daily airline operations over a 3-year time period. In other words, based on the decisions made under either the classic planning framework or the RIPF, airline operations are simulated on a day by day basis over 3 years. For each day, a random failure scenario is generated according to hub failure probabilities. A failure scenario basically reflects which hubs have failed on the day of operation. Since failure probabilities are normally small ($< 5\%$), regular operation occurs on most days, that is no hub fails and the system performance is considered acceptable. Regarding failure scenarios with one failure or more, operations may change drastically. We calculate the number of flights that have to be rerouted through active hubs for the day. For each active hub, rerouted flights are then added to the flights that use the hub under regular operation. This reflects the changes in traffic caused by random failures. The change in traffic at active hubs may lead to resource shortages. A "gate shortage" occurs when the number of flights going through a hub will require the use of a number of gates that is larger than the number of available gates at the hub. For instance, given the 8 hours of operation and at least 1 hour turnarounds, an airline with 3 gates at a hub can manage at most 24 (3×8) flights. Any additional flight results in a gate shortage. Also, if the performance of the ground crew in terms of task execution is reduced by more than 15% from regular operation, the hub experiences a "ground crew shortage". Although there are many other measures to assess performance, we focus on hub resource shortages as the main criteria for comparing the performance of the classical planning framework versus the RIPF. For a fair comparison between the outcomes of the simulations for the classical planning approach and the RIPF, we use the same random seed for each run to generate identical failure scenarios over the 3 years of the simulation. The simulation is run once for the classical planning framework and once for each of the variants of the RIPF as presented in Section 4.4. Considering a total of 1,095 days in one simulation run should provide a dependable measurement of the overall performance for the system.

4.4 Numerical Study

Table 4.2 summarizes the problem parameters used for the case application of the RIPF. At Level-1, we solve the Reliable p-Hub Network Design problem under Multiple Disruptions (RpHND-MD) briefly described in Section 4.3.1. We utilize the efficient solution method proposed by in Section 2.3.4. The Civil Aeronautics Board (CAB) dataset is used as input. CAB includes data on passenger flows between 25 cities in the United States and it has been extensively used in hub-and-spoke models. We consider a network with 25 nodes in which we want to locate 5 hubs (i.e., p = 5). The inter-hub discount factor is set at 20%. Each node/airport in the network has a known failure probability as shown in Table 4.3. The failure probabilities are obtained from An et al. (2015). A random number generator is used to determine if an airport is subject to failure on a given day. Failure scenarios are limited to up to two simultaneous failures. Solving the RpHND-MD leads to the selection of the following airports as hubs:

1. ORD - Chicago
2. NYC - New York
3. LAX - Los Angeles
4. DFW - Dallas Fort-Worth
5. MIA - Miami

Table 4.2: Problem parameters for numerical study.

Parameter	Value
Number of Nodes	25 Cities
Number of Hubs	5 Hubs
Inter-hub Discount Factor	20%
Maximum Turnaround Length	1 Hour
Operating Hours per Day	8 Hours
Maximum Number of Simultaneous Failures	2
Aircraft Capacity	150 Passengers

Four simulations are run to evaluate RIPF and compare against the classical approach. The first simulation is based on the classical planning approach with merely a top-down flow

Table 4.3: Failure probabilities of airports in the network.

#	City	Failure Probability	#	City	Failure Probability
1	Atlanta	2.3%	14	Miami	2.7%
2	Baltimore	1.7%	15	Minneapolis	1.3%
3	Boston	4.7%	16	New Orleans	1.9%
4	Chicago	4.1%	17	New York	5.0%
5	Cincinnati	2.6%	18	Philadelphia	2.4%
6	Cleveland	4.7%	19	Phoenix	4.5%
7	Dallas-Fort Worth	1.2%	20	Pittsburgh	1.2%
8	Denver	1.5%	21	St. Louis	3.5%
9	Detroit	3.5%	22	San Francisco	4.3%
10	Houston	2.6%	23	Seattle	2.0%
11	Kansas City	1.8%	24	Tampa	3.6%
12	Los Angeles	4.9%	25	Washington DC	5.0%
13	Memphis	2.4%			

of information (as shown in Figure 4.2). Three versions of RIPF are run, namely, RIPF-1, RIPF-2, and RIPF-3 based on the use of the corrective constraints described in Section 4.3.5. RIPF-1 includes the first corrective constraint, RIPF-2 includes the first and the second corrective constraints, and RIPF-3 includes all three corrective constraints. All simulations are run with the same random seed for a fair comparison.

The output to Level-1 is translated into flow that goes through each hub during regular operation as well as during operation under hub failures. Assuming standard equipment (i.e., an aircraft) with 150 passenger capacity, we calculate the number of flights that will arrive and depart from a hub airport. Given the number of flights, a minimum one hour turnaround, and 8 hours of daily operations, we calculate the flight times and the number of gates required at every hub airport. We pass along the flight times and gate information obtained in Level-2 to Level-3 to determine the ground crew deployed at each hub using Algorithm 6. Lastly, the task assignment problem is solved at the airport level for each iteration (i.e., each day in the 3 years of the simulation). The simulations are implemented in Java and run on a 64-bit Windows 10 workstation (Intel Xeon CPU E3-1240 @ 3.40 GHz, 12.0 GB of RAM). To improve run time, task assignment problems at Level-4 were solved in parallel with a 4 minute run time limit. Each simulation run with 5 hubs lasted about 14 hours.

Table 4.4 shows the distribution of the number of simultaneous failed hubs in the network throughout 3 years (1,095 days) after a simulation is run. Note that failures occur during 180 days out of the total number of days in the simulation run. 4.5 shows the distribution

Table 4.4: Distribution of number of simultaneous failed hubs, p=5.

# of Failed Hubs	Frequency (days)	Rate
0	915	83.56%
1	168	15.34%
2	12	1.10%

Table 4.5: Distribution of hub failures on days with at least one failure, p=5.

Failed Hub(s)	Frequency (days)	Rate
ORD	54	30.00%
LAX	45	25.00%
NYC	40	22.22%
MIA	22	12.22%
DFW	7	3.89%
NYC, ORD	3	1.67%
ORD, LAX	3	1.67%
NYC, LAX	2	1.11%
NYC, MIA	1	0.56%
NYC, DFW	1	0.56%
ORD, MIA	1	0.56%
DFW, MIA	1	0.56%

of hub failures during those 180 days.

We assume that the same ground crew workers are available at each hub airport everyday throughout the simulation length. When failures occur, rerouted flights will vary the workload (i.e., number of flights arriving and departing) at active hubs. Fluctuations in workload at a given hub may lead to crew shortages given the limited number of ground crew workers available on site. Table 4.6 provides an example of the status of the system for a day in the simulation under regular operations and under a failure scenario in which the hub at ORD fails. As a result of the failure, the number of flights and consequently the workload increases at active hubs. Also, the percentage of rewards gained by executing tasks at DFW drops more than 19% causing a crew shortage at this hub airport.

Table 4.7 shows the distribution of the number of active hubs affected by a gate shortage on days with hub failures for the simulations under the classic framework, RIPF-1, and RIPF-2. Accounting for a 20% increase in flow traffic when determining the number of gates to make available at each hub (i.e., as in RIPF-1) increases the proportion of days with no gate shortages at active hubs from 37% to 65%. Moreover, the reconfiguration of backup connections (i.e., as in RIPF-2) increases the proportion of days with no gate shortages at active hubs to over 95%. This latter result is achieved at the expense of a 0.0046% increase

Table 4.6: Status of the system on a simulated day under regular operations and when ORD fails.

	ORD Regular Operation	ORD Failure at ORD	DFW Regular Operation	DFW Failure at ORD	LAX Regular Operation	LAX Failure at ORD	MIA Regular Operation	MIA Failure at ORD	NYC Regular Operation	NYC Failure at ORD
# of gates	5	-	2	2	3	3	2	2	5	5
# of flights	32	-	11	16	16	18	9	10	32	37
# of ground crew workers	25	-	15	15	18	18	15	15	25	25
# of tasks	256	-	88	128	128	144	72	80	256	296
Percentage of tasks rewards gained	60%	-	**85%**	**66%**	80%	72%	92%	86%	58%	52%

Table 4.7: Distribution of the number of active hubs with gate shortages on days with hub failures, p=5.

# of Active Hubs with Gate Shortages	Classic Framework Frequency (days)	Classic Framework Rate	RIPF-1 Frequency (days)	RIPF-1 Rate	RIPF-2 Frequency (days)	RIPF-2 Rate
0	68	37.78%	118	65.55%	172	95.55%
1	51	28.33%	59	32.77%	5	2.78%
2	4	2.22%	3	1.67%	3	1.67%
3	57	31.67%	0	0.00%	0	0.00%

in the expected cost of the network.

Table 4.8 provides additional information about the performance of RIPF-2 by showing the frequency and rate of crew shortages that occur at each hub as a result of single-failure scenarios. We observe 14 incidents where a failure at ORD led to a crew shortage at DFW given a reduction of more than 15% in the task execution rate at this active hub airport if it were to handle the additional flow dictated by the solution at Level-1. Reconfiguring backup routes in Level-1 and shifting some of the flow from DFW to other active hubs when ORD fails (i.e., as in RIPF-3) reduce the number crew shortage incidents to zero as shown in Table 4.9. This reduction slightly increases the chance of a crew shortage at ORD when NYC fails. One may consider increasing the number of the ground crew workers deployed at ORD as another corrective constraint to resolve the issue.

Table 4.8: Distribution of crew shortages at active hubs under RPIF-2.

Failed Hub	Active Hub with Crew Shortage: ORD Freq. (days)	ORD Rate	DFW Freq. (days)	DFW Rate	LAX Freq. (days)	LAX Rate	MIA Freq. (days)	MIA Rate	NYC Freq. (days)	NYC Rate	Total (Row) Freq. (days)	Total (Row) Rate
ORD	-	-	**14**	**30%**	**1**	**2%**	0	0%	0	0%	15	33%
DFW	0	0%	-	-	0	0%	0	0%	0	0%	0	0%
LAX	0	0%	0	0%	-	-	0	0%	0	0%	0	0%
MIA	0	0%	0	0%	0	0%	-	-	0	0%	0	0%
NYC	**31**	**67%**	0	0%	0	0%	0	0%	-	-	31	67%
Total (Col.)	31	67%	14	30%	1	2%	0	0%	0	0%	46	100%

Table 4.9: Distribution of crew shortages at active hubs under RPIF-3.

Failed Hub	Active Hub with Crew Shortage										Total (Row)	
	ORD		DFW		LAX		MIA		NYC			
	Freq. (days)	Rate	Freq. (days)	Rate	Freq. (days)	Rate	Freq. (days)	Rate	Freq. (days)	Rate	Freq. (days)	Rate
ORD	-	-	**0**	**0%**	**1**	**2%**	0	0%	0	0%	1	2%
DFW	0	0%	-	-	0	0%	0	0%	0	0%	0	0%
LAX	0	0%	0	0%	-	-	0	0%	0	0%	0	0%
MIA	0	0%	0	0%	0	0%	-	-	0	0%	0	0%
NYC	**37**	**96%**	0	0%	0	0%	**1**	**2%**	-	-	38	98%
Total (Col.)	37	96%	0	0%	1	2%	1	2%	0	0%	39	100%

The results presented above show how one can improve the overall performance of the system by implementing inter-level dependencies between decisions. Accounting for fluctuations in the number of flights going through each hub (an outcome of Level-1) helped making better decisions at Level-2. Allowing for at least 20% increase in number of flights when selecting the number of gates at a hub caused a drastic reduction in gate shortages. Also, we observed that balancing rerouted flights across active hubs comes with an additional performance improvement at a slight cost in the network operating costs at Level-1.

4.4.1 Effect of Number of Hubs on Overall Performance

We also ran the same set of simulations increasing the number of hubs to be selected in Level-1 from 5 to 7. The following cities were selected as hubs in this case:

1. ORD - Chicago
2. NYC - New York
3. LAX - Los Angeles
4. DFW - Dallas Fort-Worth
5. MIA - Miami
6. PIT - Pittsburgh
7. SFO - San Francisco

Figure 4.5 and Table 4.10 provide an overview of how hub failures are distributed throughout the 3 years of the simulation. Intuitively, having more hubs in the system will increase the probability of having at least one hub failure in the system (the rate of single hub failures increases from 15% for p=5 to 20% for p=7).

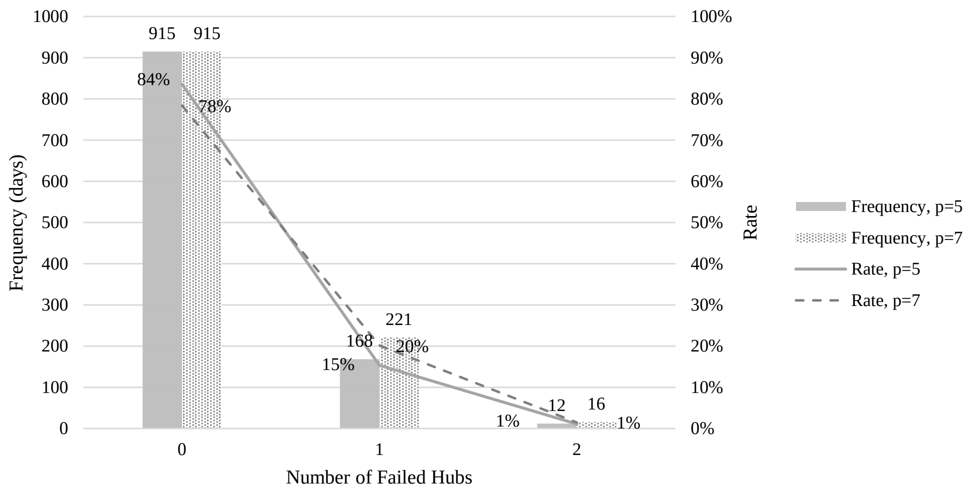


Figure 4.5: Distribution of hub failures in terms of number of simultaneously failed hubs, $p = 5$ and $p = 7$.

Table 4.10: Distribution of hub failures on days with at least one failure, $p = 7$.

#	Failed Hub(s)	Frequency (days)	Rate	#	Failed Hub(s)	Frequency (days)	Rate
1	NYC	53	22.36%	10	NYC, DFW	2	0.84%
2	SFO	46	19.41%	11	ORD, LAX	2	0.84%
3	LAX	43	18.14%	12	LAX, MIA	1	0.42%
4	ORD	34	14.35%	13	NYC, MIA	1	0.42%
5	MIA	21	8.86%	14	NYC, PIT	1	0.42%
6	DFW	18	7.59%	15	NYC, SFO	1	0.42%
7	NYC, LAX	6	2.53%	16	NYC, DFW	1	0.42%
8	ORD, SFO	3	1.27%	17	ORD, PIT	1	0.42%
9	SFO, LAX	3	1.27%				

Table 4.11: Distribution of the number of active hubs with gate shortages on days with failure, p=7.

# of Active Hubs with Gate Shortages	Classic Framework		RIPF-1		RIPF-2	
	Frequency (days)	Rate	Frequency (days)	Rate	Frequency (days)	Rate
0	152	64.14%	229	96.62%	235	99.15%
1	79	33.33%	8	3.38%	2	0.85%
2	6	2.53%	0	0.0%	0	0.0%
3	0	0.0%	0	0.0%	0	0.0%

Regarding the distribution of the number of active hubs affected by gate shortages, we compare Table 4.7 and Table 4.11. Figure 4.6 presents the comparison graphically in terms of the rate of hubs with gate shortages for the cases with p=5 and p=7. Interestingly, although the number of failures grows with the number of hubs in the system, we observe a significant reduction in the rate of gate shortages. In the classic planning approach, with no corrective constraints considered, the number of failures that do not lead to an active hub being affected by gate shortages jumps from 38% to 64% when number of hubs is increased from 5 to 7. Under RIPF-1, the rate of no active hubs with gate shortages goes up to 97% for 7 hubs as opposed to only 66% with 5 hubs. Figure 4.6 clearly shows that the system with more hubs is more robust. As a result, less effort is needed for modifications such as increasing gate allowances or having a larger ground crew in place to avoid resource shortages. On the other hand, increasing the number of hubs in the network could increase the cost associated with the decisions made at Level-1, but the desirable effect of such decisions at lower levels might justify decisions at Level-1 that may not look optimal when examined in isolation as opposed to as part of an integrated model. Capturing such interactions is not easy unless the system is examined within the context of the RIPF.

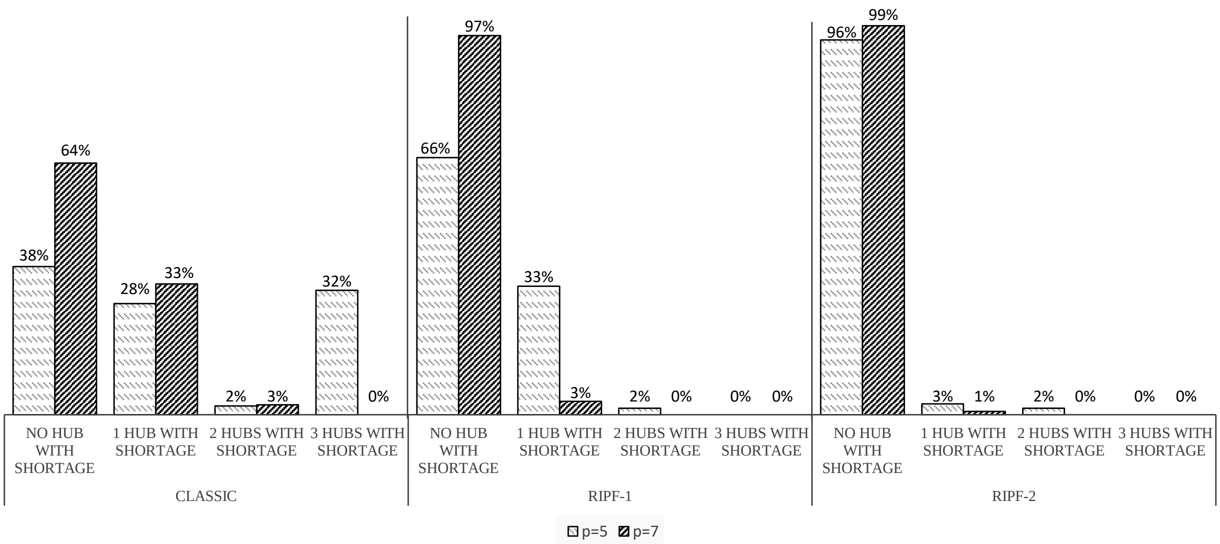


Figure 4.6: Comparison of the distribution of the number of active hubs with gate shortages on days with failure

4.5 Conclusions and Future Work

Organizations deal with highly complicated systems involving numerous strategic, tactical, and operational decisions. Traditionally, a top to bottom approach cascades the outcome of decisions made at each level to the next. Very often, this approach leads to sub-optimal performance of the system as whole. This becomes more evident in uncertain environments. To overcome this challenge, we propose a reliable integrated planning framework that, unlike the classic top to bottom approach, considers each decision individually but iteratively incorporates corrective constraints signaled by other related decisions from both higher and lower levels. These corrective constraints may imply fluctuations in flow/demand due to uncertainty in the system or any type of limitations dictated by other interrelated decisions. We tested this approach in the context of the highly uncertain airline industry. Evaluating the overall system performance using simulation showed how optimal decisions made in isolation at one level can reduce the system's overall performance. In fact, a proper measurement of overall performance, regardless of the application, is a prerequisite to making effective decisions for the system. Going back to our airline operations case, we saw that a decision that minimizes the total expected operating cost of the network at Level-1 does not necessarily optimize the overall performance of the system (i.e., by having as few resource shortages as possible). Therefore, our proposed approach provides a more comprehensive measurement of system performance than traditional, single-measure approaches do as they overlook the effect of a local optimal decision on the overall performance.

Sometimes a single decision at a given level of decision making, regardless of its effect on other planning levels, can be very complex. However, RIPF could potentially relax the need to explicitly consider some complicating constraints. For instance, to the best of our knowledge no study has yet proposed an exact solution for the capacitated version of RpHND-MD, a complicated problem. However, in the case study presented above (see corrective constraints 2 and 3), we were able to significantly reduce the chances of hubs being affected by resource shortages (i.e., gates and ground crew workers not being available to handle increased flow) without the need to explicitly address the capacitated version of the RpHND-MD. This characteristic of RIPF would help developing more tractable models.

As observed, an integrated approach can significantly facilitate capturing the ripple effect of decisions affecting different levels of decision making. The comparison of the cases where 5 and 7 hubs were placed in the network shows that a more expensive decision at the strategic level (e.g., choosing 7 hubs over 5), significantly improves the robustness of operations and saves time and effort when addressing large fluctuations in daily operations that lead to hubs experiencing resource shortages.

Moreover, when addressing uncertainty, one of the challenges is data availability. This is especially true when a new system is to be designed. Sometimes, information related to uncertainty that is available at higher levels of the organization can help to capture potential fluctuations at lower levels of decision making. This would not be possible if the different decisions at different levels are considered in isolation. Additionally, planning and controlling complex decision systems in the context of a RIPF helps to capture the effect of internal sources of uncertainty. For instance in our numerical study, the task assignment problem solved on the day of operation involves a random search which yields a certain level of variation in the performance of the ground crew workers. Performance evaluation through simulation automatically takes into account such internal sources of uncertainty.

RIPF is relatively easy to understand as it clearly shows the corrective constraints and how decisions are related to other decisions made in the system. This characteristic facilitates communication between technical practitioners and senior management at organizations.

Although it is easy to develop and justify the introduction of corrective constraints in RIPF, in practice, these constraints can grow exponentially in the size of the system and number of decisions to make. Many of the required corrective constraints might not be known in the first place. The iterative nature when developing a RIPF allows for identifying such constraints. On the other hand, running multiple simulations to identify new corrective constraints or modifying existing ones can be heavily time consuming. This is specifically true as the number of decisions and their complexity grow. Overall, RIPF reduces the computational complexity by focusing on each decision individually while considering corrective constraints. However, this improvement in tractability is limited and cannot completely overcome the never-ending challenge in decision making: finding the global optimal decision.

Another challenge facing decision makers who would utilize RIPF lies in performance measurement. Developing an objective function to optimize an individual problem is fairly straightforward. However, when it comes to evaluating a complicated system with multiple inter-dependent optimization problems, more effort has to be put into devising a quantitative measure for the overall system. As the system gets more complicated, more conflicting criteria have to be kept under the radar. As an example, in this study we focused on alleviating resource shortages as the performance measure. However, improving this criteria comes at the cost of establishing more hubs or rerouting flights to more distant hubs.
In this context, it seems that more practical studies aiming to capture the actual interaction of decisions at different planning levels in different applications would be very worthwhile. At the same time, applying common multi-objective decision methods as well as suggesting innovative approaches to performance measurement can bring the literature and practice even closer together.

4.6 References

S. Alumur and B. Y. Kara. Network hub location problems: The state of the art. *European Journal of Operational Research*, 190(1):1–21, 2008.

Y. An, Y. Zhang, and B. Zeng. The reliable hub-and-spoke design problem: Models and algorithms. *Transportation Research Part B: Methodological*, 77:103–122, 2015.

P. Barahimi and H. A. Vergara. Reliable p-Hub Network Design Under Multiple Disruptions. August 2018a.

P. Barahimi and H. A. Vergara. Task Assignment with Flexible Execution Times and Sequence Dependent Travel. Technical report, Oregon State University, Department of Mechanical, Industrial, and Manufacturing Engineering, November 2018b.

F. T. Chan, N. K. Tang, H. Lau, and R. Ip. A simulation approach in supply chain management. *Integrated Manufacturing Systems*, 13(2):117–122, 2002.

Ş. S. Erengüç, N. C. Simpson, and A. J. Vakharia. Integrated production/distribution planning in supply chains: An invited review. *European Journal of Operational Research*, 115(2):219–236, 1999.

R. Z. Farahani, M. Hekmatfar, A. B. Arabani, and E. Nikbakhsh. Hub location problems: A review of models, classification, solution techniques, and applications. *Computers & Industrial Engineering*, 64(4):1096–1109, 2013.

M. Goetschalckx, C. J. Vidal, and K. Dogan. Modeling and design of global logistics systems: A review of integrated strategic and tactical models and design algorithms. *European Journal of Operational Research*, 143(1):1–18, 2002.

S. L. Janak, X. Lin, and C. A. Floudas. A new robust optimization approach for scheduling under uncertainty: Ii. uncertainty with known probability distribution. *Computers & Chemical Engineering*, 31(3):171–195, 2007.

P. Kouvelis and G. Yu. *Robust discrete optimization and its applications*, volume 14. Springer Science & Business Media, 2013.

M. T. Melo, S. Nickel, and F. Saldanha-Da-Gama. Facility location and supply chain management–a review. *European Journal of Operational Research*, 196(2):401–412, 2009.

G. S. Parnell, M. Terry Bresnick, S. N. Tani, and E. R. Johnson. *Handbook of decision analysis*, volume 6. John Wiley & Sons, 2013.

E. H. Sabri and B. M. Beamon. A multi-objective approach to simultaneous strategic and operational planning in supply chain design. *Omega*, 28(5):581–598, 2000.

S. Talluri and R. Baker. A multi-phase mathematical programming approach for effective supply chain design. *European Journal of Operational Research*, 141(3):544–558, 2002.

J. Van den Bergh, J. Beliën, P. De Bruecker, E. Demeulemeester, and L. De Boeck. Personnel scheduling: A literature review. *European Journal of Operational Research*, 226(3):367–385, 2013.

X. Zheng, M. Yin, and Y. Zhang. Integrated optimization of location, inventory and routing in supply chain network design. *Transportation Research Part B: Methodological*, 121: 1–20, 2019.

4.7 Appendices

4.7.1 Appendix A: Notation for RpHND-MD

Sets, parameters, and decision variables defined by Barahimi and Vergara (2018a) to develop the mathematical model for RpHND-MD are as follows:

Sets:

N = set of locations in network,

V_{ij} = set of vertices in the connection-tree for O-D pair ij,

V_{ij}^{v} = set of vertices on the path from root to vertex v of the connection-tree for O-D pair ij, without including vertex v; $V_{ij}^{v} \subset V_{ij}$.

Indices:

i, k, m, j point to the origin, first hub, second hub, and destination of a connection, respectively; $i, k, m, j \in N$,

u, u', v point to a vertices of a connection tree. $u, u', v \in \{0, 1, \ldots, 2^{L+1} - 2\}$ where L is the maximum number of simultaneous failures.

Parameters:

p = number of hubs to be located,

B_{ij} = total flow between i and j,

c_{ij} = cost of sending a unit of flow directly from node i to node j,

α = discount factor for shipping flow though inter-hub links,

C_{ikmj} = cost of shipping a unit of flow through connection (i, k, m, j) regardless of failures:

$$C_{ikmj} = c_{ik} + (1 - \alpha)\ c_{km} + c_{mj}$$

q_k = failure probability of a hub located at node k,

q_{km} = failure probability of a connection that uses k and m as hubs,

L = maximum number of simultaneous failures,

$v' = v$'s parent vertex.

Decision Variables:

In a directed tree, each vertex has only one incoming edge. Therefore, any edge, except the root vertex, can be pointed at by its tail-vertex. Define e^v_{ij} as the edge ending in vertex v in connection tree T_{ij}.

$$z^v_{ij} = \begin{cases} 1, & \text{if vertex } v \text{ is mapped to a connection, in connection tree } T_{ij}, \\ 0, & \text{otherwise.} \end{cases}$$

$$x^v_{ikmj} = \begin{cases} 1, & \text{if vertex } v \text{ is mapped to connection } (i,k,m,j) \text{ , in connection tree } T_{ij} \text{ ,} \\ 0, & \text{otherwise.} \end{cases}$$

$$e^v_{ijk} = \begin{cases} 1, & \text{if the edge with vertex } v \text{ as its tail is mapped to node } k, \text{ in connection tree } T_{ij}, \\ 0, & \text{otherwise.} \end{cases}$$

$$y_i = \begin{cases} 1, & \text{if a hub is established in node } i \text{ of the network,} \\ 0, & \text{otherwise.} \end{cases}$$

Auxiliary Variables:

$$\kappa^v_{ij} = \begin{cases} 1, & \text{if vertex } v \text{ of connection tree } T_{ij} \text{ gets a connection with} \\ & \text{either } i \text{ or } j \text{ as its first hub,} \\ 0, & \text{otherwise.} \end{cases}$$

$$\mu^v_{ij} = \begin{cases} 1, & \text{if vertex } v \text{ of connection tree } T_{ij} \text{ gets a connection with} \\ & \text{either } i \text{ or } j \text{ as its second hub,} \\ 0, & \text{otherwise.} \end{cases}$$

4.7.2 Appendix B: Notation for TAP-FET-SDT

Sets, parameters, and decision variables defined by Barahimi and Vergara (2018b) to develop the mathematical model for TAP-FET-SDTare as follows:
Sets:

- T: Set of tasks,
- W: Set of workers,
- T_w: Set of tasks that can be executed by worker w,
- $T_t^{'}$: Set of tasks succeeding task t. Task $t^{'}$ succeeds task t iff the latest start time of $t^{'}$ is later than the earliest finish time of t,
- T_t'': Set of tasks that conflict with task t. Task t'' conflicts with task t iff both tasks cannot be executed by the same worker due to a time limitation. In other words, t and t'' are overlapping tasks,
- W_t: Set of workers who can execute task t, i.e., they have at least one of the skills required to execute t,
- S: Set of qualifications, i.e., skills,
- B: Set of time periods in the planning horizon,
- B_t: Set of time periods in which task t can be started,
- B_{tw}: Set of time periods in which task t can be started by worker w,.

Indices:

- $t, t', t'' \in T$ task indices,
- $w \in W$ worker index,
- $s \in S$ qualification index,
- $b, b' \in B$ time period indices,

Parameters:

- α_t: reward associated with executing task t,
- β_{ws}: 1 if worker w has skill s, 0 otherwise,

- γ_{ts}: number of workers with skill s required to execute task t,
- δ_t: execution length of task t,
- $\theta_{tt'}$: the travel time between tasks t and t',
- M: a very large number.

Decision Variables:

- l_{tb}: equals to 1 if the execution of task t is set to be started at period b; 0 otherwise,
- m_{twb}: equals to 1 if task t is assigned to worker w and is to start in time period b; 0 otherwise,
- $h_{tt'wb}$: equals to 1 if both tasks t and t' are assigned to worker w such that task t starts at period b and task t' starts in any period after b; 0 otherwise.

Chapter 5: Conclusions and Direction for Future Research

Making optimal decisions has always been the ultimate objective of decision makers and researchers. Considering strategic, tactical and operational decisions in a single model to capture all dependencies between different levels of decision making would allow overall performance optimization. However, the exponential increase in computational complexity prevents us from considering all decisions in a single quantitative model. At the same time, as planning problems increase in size, they become more susceptible to uncertainty associated with unanticipated events which affects the effectiveness of the planning decisions in practice. Incorporating uncertainty in large decision planning models adds even more computational complexity. To overcome these challenges, we propose the application of a Reliable Integrated Planning Framework(RIPF) that focuses on each planning decision individually, but at the same time considers feedback from the outcome of other decisions in the form of corrective constraints. In this way, decisions made at each level of decision making not only improve the performance of the affected problem, but also contribute to the overall performance of the system. In other words, the decision made at each sub-system attempts to improve the overall performance rather than merely optimizing the sub-system. To illustrate the application of the proposed approach, we developed, implemented and evaluated a RIPF in the context of airline operations planning. In this case, the decision making framework is composed of four interdependent sets of decisions.

At the top level (i.e., Level-1), we considered the Reliable p-Hub Network Design Problem with Multiple Disruptions (RpHND-MD) as a strategic decision which affects the operations of a hypothetical airline at each of its established hub airports. The location of hub airports, and the connection of origins and destinations through those hubs under different scenarios are the major outcomes of Level-1. At Level-2, the goals are to determine flight times and select the number of gates required at each hub airport. Decisions at this level are affected by the output to the hub network design problem at Level-1. A few basic assumptions on flight turnarounds and fixed operation hours made the decision at this level simple. At Level-3, we proposed a heuristic to determine the ground crew to make available at each hub airport based on the number of flights that need to be handled. At the lowest level in the framework (Level-4), we considered a task assignment problem that focuses on assigning ground crew to tasks required to be executed on flights between arrivals and departures. The problems considered at Level-1 and Level-4 are complicated enough and required detailed study on suitable modeling and solution approaches.

For RpHND-MD, we suggested a nonlinear mathematical model that is based on the concept of connection trees (Chapter 2). Using an efficient search algorithm which is based on the well-known branch-and-bound method, we solved instances of the problem with up to 25 nodes and 7 hubs. The results showed that an increase in the maximum number of simultaneous failures affects the location of hubs and the utilization of inter-hub links. However, the locations of hubs do not change as the maximum number of failures grows larger than one. From this finding, for large networks, we can limit the maximum failures to one and solve the problem to find the location of hubs. By fixing the hubs, the problem becomes easier to solve to optimality. We also observed that growth in the average failure rate of nodes lead to a decline in utilization of inter-hub links. Therefore, in networks with relatively high failure rates, it is suggested that using a hub-and-spoke topology might not be very appropriate.

The Task Assignment Problem with Flexible Execution Times and Sequence Dependent Travel (TAP-FET-SDT), studied for Level-4, is a combinatorial optimization problem that has to be solved multiple times during the day at each hub airport (Chapter 3). The goal in TAP-FET-SDT is to execute the largest possible number of high priority tasks with a limited number of ground crew workers to shorten flight turnarounds. The computational runtime cannot exceed a few minutes as any cancellation or delay in flights would require an update of the task assignment. Therefore, we developed a branch-and-price algorithm to obtain quality solutions within computational runtime limitations. Two heuristics were developed to provide fast solutions to the pricing problem. The first heuristic is a modified branch-and-bound search. The second heuristic is a modified Great Deluge (GD) search. Computational testing on different instances of the problem showed that even though branch-and-bound outperforms GD in generating higher quality column for the master problem, GD is faster in generating reasonably good columns. Therefore, the output of the branch-and-price search turned out to be better on average when the pricing problem is solved with our proposed GD algorithm. We also observed that for instances with the same number of tasks, the problem complexity declines as more workers are available. This contrasts with our initial intuition. Furthermore, we observed a relatively high variability in worker utilization across workers assigned to tasks. To mitigate these shortcoming, we introduced workload balancing constraints to the model. The results indicated that as long as the cap on the maximum difference between each pair of workers utilization is not too tight, enforcing workload balancing constraints does not significantly affect the level of overall utilization for all workers. Also, in some cases, such constraint may even improve computational tractability. As another noteworthy point, we saw that increasing execution time windows does not necessarily lead to a better assignment given the limitations in computational runtime. This is due to the significant increase in complexity of the problem. Therefore, the potential benefit from having more flexible tasks is undermined by the growth in complexity for large instances when we have limited time to solve the problem.

Finally, in Chapter 4, we modeled the proposed airline operations framework once using the classical modeling approach with a top-to-bottom flow of decision interactions, and another time with the RIPF. A set of simulations of failure scenarios affecting the availability of hub airports was used to compare the airline's overall performance under the two approaches. To evaluate the overall performance, we focused on resource shortages (i.e., gate shortages and ground crew shortages) at active hubs caused by failure in one or more hub airports. We showed a few examples on how after explicitly introducing uncertainty to the higher level of decision planning hierarchy (i.e., strategic decisions), an apparently good strategic decision can still produce an undesirable effect on system performance at tactical and operational levels. The results of the computational study showed that introducing simple corrective constraints to the planning framework rather than merely passing the solution of a higher level decision on to the lower level can significantly improve overall performance. Furthermore, it is suggested that the RIPF can be used to address the challenge related to lack of historical data when incorporating uncertainty in decision making. We showed that if decisions at different levels of an organization are integrated in one comprehensive model such as the RIPF, the decision maker can easily track the fluctuations in demand or availability of resources from the top level all the way down to the operational levels where lack of historical data is most recognized. For demonstration purposes, we only accounted for a 20% increase in traffic at hubs. This reflects other limitations that emerge in real cases where we do not have complete freedom in selecting the number of gates to have available at a hub. We showed how we can address this issue by incorporating additional corrective constraints (i.e., RIPF-2) to modify the decisions at the network design level. Additional corrective constraints will further reduced gate shortages and improve the overall performance.

Reflecting on the existing gaps in the literature as well as the findings and limitations of this study, we suggest focusing on the following areas to further extend the literature.

- Regarding the Reliable p-Hub Network Design problem, to the best of our knowledge, no study has addressed the capacitated version of the problem which is applicable to almost all real-world problems. In addition, considering "dependent" failure probabilities and how they would affect the modeling and solution approaches as well as the network configuration is definitely worth exploring. We also noticed a gap in the reliable hub network design literature due to the lack of studies focusing on failures of links or any other network entities rather than just the nodes. It should be noted that link failures in a hub network might not be relevant to airline operations. However, other applications of hub networks such as rail transportation and telecommunication networks can benefit from studies that consider failure of links. With regards to failure distribution of hubs, we assumed independent probabilities. In real-world, geographical proximity of nodes entails dependent failures which may drastically affect the modeling and solution approaches to RpHND-MD. Finally, in Chapter 2, we conducted computational experiments to assess the effect of different parameters such as inter-hub discount factor and failure rates on hub network configuration. It would be interesting to further explore the effect of these and other network design parameters on the overall performance of airline operations through the RIPF.

- Airline operations are inherently spread across broad geographic areas making them highly susceptible to unanticipated events such as severe weather. Delays and cancellations are almost unavoidable in airline operations. Such events force the airline operations team to reschedule tasks at the airport level leading to frequent changes in the assignment of ground crew to tasks. Frequent changes to the ground crew work schedule can negatively affect employees' satisfaction leading to high turnover rates. It seems that further effort should go into explicitly incorporating uncertainty into the formulation of the TAP-FET-SDT and generate robust work schedules. A robust task assignment requires less frequent changes in the assignments. As another direction for future research, one can explore the effect of the reward system used for the objective function with respect to computational performance and the solutions obtained. In this dissertation, we assumed satisfaction of skill set requirements as a hard constraint on achieving the reward associated with the task. However, other reward systems such as allowing for under-covering task requirements may be worth exploring. Also, examining the effect of different reward value distributions (besides uniform) can shed more light on the behavior of the TAP-FET-SDT.

- One of the challenges faced in implementing the RIPF relates to the overall performance measurement of the framework. As the decision maker moves towards integrating decisions at multiple levels of the organization, developing a practical performance measure that actually reflects the overall performance becomes more challenging. Several conflicting objectives may arise making it difficult to decide on the right criteria to consider when assessing overall performance. It seems that more comprehensive research on innovative mechanisms to capture overall system performance in highly complicated systems with numerous dependent decisions would be worthwhile. Finally, it seems that more practical studies aiming at capturing the actual interaction of decisions at different planning levels in different applications would be very valuable. In this study we addressed a simplified and limited version of airline operations to demonstrate the advantages and disadvantages of the RIPF. However, applying the same approach to an actual real-world problem will shed light on unexplored specifics of the subject matter. Many practitioners and academics will definitely benefit from these findings. Lastly, we recommend exploring the effect of other sources of uncertainty such as fluctuations in flows associated with origin-destination pairs as another direction for future research.

References

S. M. Al-Yakoob and H. D. Sherali. Mixed-integer programming models for an employee scheduling problem with multiple shifts and work locations. *Annals of Operations Research*, 155(1):119–142, 2007.

S. Alumur and B. Y. Kara. Network hub location problems: The state of the art. *European Journal of Operational Research*, 190(1):1–21, 2008.

Y. An, Y. Zhang, and B. Zeng. The reliable hub-and-spoke design problem: Models and algorithms. *Transportation Research Part B: Methodological*, 77:103–122, 2015.

N. Azizi. Managing facility disruption in hub-and-spoke networks: formulations and efficient solution methods. *Annals of Operations Research*, pages 1–27, 2017.

N. Azizi, S. Chauhan, S. Salhi, and N. Vidyarthi. The impact of hub failure in hub-and-spoke networks: Mathematical formulations and solution techniques. *Computers & Operations Research*, 65:174–188, 2016.

P. Barahimi and H. A. Vergara. Reliable p-Hub Network Design Under Multiple Disruptions. August 2018a.

P. Barahimi and H. A. Vergara. Task Assignment with Flexible Execution Times and Sequence Dependent Travel. Technical report, Oregon State University, Department of Mechanical, Industrial, and Manufacturing Engineering, November 2018b.

J. F. Bard and L. Wan. The task assignment problem for unrestricted movement between workstation groups. *Journal of Scheduling*, 9(4):315–341, 2006.

J. F. Campbell and M. E. O'Kelly. Twenty-five years of hub location research. *Transportation Science*, 46(2):153–169, 2012.

F. T. Chan, N. K. Tang, H. Lau, and R. Ip. A simulation approach in supply chain management. *Integrated Manufacturing Systems*, 13(2):117–122, 2002.

I. Contreras. Hub location problems. In G. Laporte, S. Nickel, and F. S. da Gama, editors, *Location science*, chapter 10, pages 266–290. Springer, Berlin, 2015.

S. Davari, M. H. F. Zarandi, and I. B. Turksen. The fuzzy reliable hub location problem. In *Fuzzy Information Processing Society (NAFIPS), 2010 Annual Meeting of the North American*, pages 1–6. IEEE, 2010.

G. Dueck. New optimization heuristics: The great deluge algorithm and the record-to-record travel. *Journal of Computational Physics*, 104(1):86–92, 1993.

Ş. S. Erengüç, N. C. Simpson, and A. J. Vakharia. Integrated production/distribution planning in supply chains: An invited review. *European Journal of Operational Research*, 115(2):219–236, 1999.

A. T. Ernst, H. Jiang, M. Krishnamoorthy, and D. Sier. Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, 153(1):3–27, 2004.

P. Eveborn, M. Rönnqvist, H. Einarsdóttir, M. Eklund, K. Lidén, and M. Almroth. Operations research improves quality and efficiency in home care. *Interfaces*, 39(1):18–34, 2009.

R. Z. Farahani, M. Hekmatfar, A. B. Arabani, and E. Nikbakhsh. Hub location problems: A review of models, classification, solution techniques, and applications. *Computers & Industrial Engineering*, 64(4):1096–1109, 2013.

M. Goetschalckx, C. J. Vidal, and K. Dogan. Modeling and design of global logistics systems: A review of integrated strategic and tactical models and design algorithms. *European Journal of Operational Research*, 143(1):1–18, 2002.

S. L. Janak, X. Lin, and C. A. Floudas. A new robust optimization approach for scheduling under uncertainty: Ii. uncertainty with known probability distribution. *Computers & Chemical Engineering*, 31(3):171–195, 2007.

L. Kaufman and P. J. Rousseeuw. *Finding groups in data: an introduction to cluster analysis.* John Wiley & Sons, Brussels, Belgium, 1990.

H. Kim. p-hub protection models for survivable hub network design. *Journal of Geographical Systems*, 14(4):437–461, 2012.

H. Kim and M. E. O'Kelly. Reliable p-hub location problems in telecommunication networks. *Geographical Analysis*, 41(3):283–306, 2009.

H. Kim and M. S. Ryerson. Ad-hoc port assignment due to a port outage. In *Logistics Society Workshop*, page 83, 2013.

J. G. Klincewicz. Hub location in backbone/tributary network design: a review. *Location Science*, 6(1):307–335, 1998.

P. Kouvelis and G. Yu. *Robust discrete optimization and its applications*, volume 14. Springer Science & Business Media, 2013.

Y.-H. Kuo, J. M. Leung, and C. A. Yano. Scheduling of multi-skilled staff across multiple locations. *Production and Operations Management*, 23(4):626–644, 2014.

Q. Lequy, M. Bouchard, G. Desaulniers, F. Soumis, and B. Tachefine. Assigning multiple activities to work shifts. *Journal of Scheduling*, 15(2):239–251, 2012a.

Q. Lequy, G. Desaulniers, and M. M. Solomon. A two-stage heuristic for multi-activity and task assignment to work shifts. *Computers & Industrial Engineering*, 63(4):831–841, 2012b.

Q. Lequy, G. Desaulniers, and M. M. Solomon. Assigning team tasks and multiple activities to fixed work shifts. *INFOR: Information Systems and Operational Research*, 51(2):64–75, 2013.

R. D. Luce and H. Raiffa. *Games and decisions: Introduction and critical survey*. Courier Corporation, 2012.

M. T. Melo, S. Nickel, and F. Saldanha-Da-Gama. Facility location and supply chain management–a review. *European Journal of Operational Research*, 196(2):401–412, 2009.

M. E. O'Kelly. The location of interacting hub facilities. *Transportation science*, 20(2): 92–106, 1986.

M. E. O'Kelly. A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research*, 32(3):393–404, 1987.

G. S. Parnell, M. Terry Bresnick, S. N. Tani, and E. R. Johnson. *Handbook of decision analysis*, volume 6. John Wiley & Sons, 2013.

F. Parvaresh, S. H. Golpayegany, S. M. Husseini, and B. Karimi. Solving the p-hub median problem under intentional disruptions using simulated annealing. *Networks and Spatial Economics*, 13(4):445–470, 2013.

F. Parvaresh, S. M. Husseini, S. H. Golpayegany, and B. Karimi. Hub network design problem in the presence of disruptions. *Journal of Intelligent Manufacturing*, 25(4):755–774, 2014.

E. H. Sabri and B. M. Beamon. A multi-objective approach to simultaneous strategic and operational planning in supply chain design. *Omega*, 28(5):581–598, 2000.

L. V. Snyder and M. S. Daskin. Reliability models for facility location: the expected failure cost case. *Transportation Science*, 39(3):400–416, 2005.

Statista. Airline industry revenue worldwide 2017 — statista. `https://www.statista.com/statistics/278372/revenue-of-commercial-airlines-worldwide/`, 2018. (Accessed on 04/20/2018).

S. Talluri and R. Baker. A multi-phase mathematical programming approach for effective supply chain design. *European Journal of Operational Research*, 141(3):544–558, 2002.

T. H. Tran, J. R. OHanley, and M. P. Scaparra. Reliable hub network design: Formulation and solution techniques. *Transportation Science*, 2016.

J. Van den Bergh, J. Beliën, P. De Bruecker, E. Demeulemeester, and L. De Boeck. Personnel scheduling: A literature review. *European Journal of Operational Research*, 226(3):367–385, 2013.

B. Yıldız and O. E. Karaşan. Regenerator location problem and survivable extensions: A hub covering location perspective. *Transportation Research Part B: Methodological*, 71: 32–55, 2015.

X. Zheng, M. Yin, and Y. Zhang. Integrated optimization of location, inventory and routing in supply chain network design. *Transportation Research Part B: Methodological*, 121: 1–20, 2019.